



ЩЕРБАНЬ В.Ю.
КОЛИСКО О.З.
ЩЕРБАНЬ Ю.Ю.
МЕЛЬНИК Г.В.
КОЛИСКО М.І.
КАЛАШНИК В.Ю.



Міністерство освіти і науки
України



Київський національний
університет технологій та
дизайну



Кафедра комп'ютерних
наук та технологій

АЛГОРИТМІЧНЕ ТА МАТЕМАТИЧНЕ
ЗАБЕЗПЕЧЕННЯ ПРИ КОМП'ЮТЕРНОМУ
ПРОЕКТУВАННІ СКЛАДНИХ СИСТЕМ



АЛГОРИТМІЧНЕ ТА
МАТЕМАТИЧНЕ
ЗАБЕЗПЕЧЕННЯ ПРИ
КОМП'ЮТЕРНОМУ
ПРОЕКТУВАННІ
СКЛАДНИХ СИСТЕМ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ

ЩЕРБАНЬ В.Ю., КОЛИСКО О.З., ЩЕРБАНЬ Ю.Ю., МЕЛЬНИК Г.В.,
КОЛИСКО М.І., КАЛАШНИК В.Ю.

**АЛГОРИТМІЧНЕ ТА
МАТЕМАТИЧНЕ
ЗАБЕЗПЕЧЕННЯ ПРИ
КОМП'ЮТЕРНОМУ
ПРОЕКТУВАННІ СКЛАДНИХ
СИСТЕМ**

Київ – 2021

УДК 677.024.3
ББК 65.9(4Укр)306.4-6
Щ 610

*Рекомендовано Вченою радою Київського національного університету технологій та дизайну для широкого кола викладачів, науковців, аспірантів, магістрів та студентів профільних вищих навчальних закладів, інженерно-технічних працівників комп'ютерної галузі
(Протокол №9 від 26 травня 2021 року)*

Колектив авторів:

ЩЕРБАНЬ В. Ю. – лауреат Державної премії України в галузі науки і техніки, академік Міжнародної академії комп'ютерних наук та систем, доктор технічних наук, професор, завідувач кафедрою комп'ютерних наук та технологій Київського національного університету технологій та дизайну;

КОЛИСКО О.З. - кандидат технічних наук, доцент кафедри комп'ютерних наук та технологій Київського національного університету технологій та дизайну;

ЩЕРБАНЬ Ю.Ю. – лауреат Державної премії України в галузі науки і техніки, доктор технічних наук, професор, заступник директора Київського фахового коледжу прикладних наук;

МЕЛЬНИК Г.В. - кандидат технічних наук, доцент кафедри комп'ютерних наук та технологій Київського національного університету технологій та дизайну;

КОЛИСКО М.І. - кандидат технічних наук, доцент кафедри комп'ютерних наук та технологій Київського національного університету технологій та дизайну;

КАЛАШНИК В.Ю. - кандидат технічних наук, асистент кафедри комп'ютерних наук та технологій Київського національного університету технологій та дизайну;

Рецензенти:

ЧЕПЕЛЮК О.В. – д.т.н., професор, завідувач кафедру, Херсонський національний технічний університет;

ОПАНАСЕНКО В.М. – д.т.н., професор, провідний науковий співробітник, Інститут кібернетики НАН України;

ЩУЦЬКА Г. В. – д.т.н., директор коледжу, Київський фаховий коледж прикладних наук.

Щ 610 Щербань В.Ю. Алгоритмічне та математичне забезпечення при комп'ютерному проектуванні складних систем / В.Ю.Щербань, О.З.Колиско, Ю.Ю.Щербань, Г.В.Мельник, М.І.Колиско, В.Ю.Калашник. – К.: Освіта України, 2021. – 930 с.

ISBN 978-617-7993-98-7

Наведені результати досліджень з питань розробки алгоритмічних, математичних та комп'ютерних програмних компонентів систем автоматизованого проектування, спрямованих на розвиток наукових засад та методології створення нових, ефективних технологій. Отримані нові алгоритми та програмне забезпечення для удосконалення існуючих та розробку нових технологій. Призначена для широкого кола викладачів, науковців, аспірантів, магістрів та студентів спеціальності 122 – комп'ютерні науки, профільних вищих навчальних закладів, інженерно-технічних працівників промисловості

УДК 677.024.3
ББК 65.9(4Укр)306.4-6

ISBN 978-617-7993-98-7

©В.Ю.Щербань, 2021
©Освіта України, 2021

ЗМІСТ

1. Загальні питання САПР	6
1.1. Характеристика стадій проектування	9
1.2. Характеристика функціональних схем САПР, їх підсистем і компонентів	31
1.3 Класифікація САПР	40
2. Об'єкти проектування в САПР легкої і текстильної промисловості	45
2.1. Структурно - логічна схема САПР устаткування і технологічних процесів легкої і текстильної промисловості	47
2.2. Структурно - логічна схема ПС САПР му-1 «трикотажна машина» і «ткацький верстат»	62
2.3. Структурно - логічна схема ПС САПР ТП му-2 «ткацтво»	65
3. Математичні моделі в САПР і їх класифікація	67
3.1. Математичні моделі і обчислювальний експеримент.....	68
3.2. Класифікація математичних моделей	80
4. Математичні моделі, що реалізуються з використанням чисельних методів	86
4.1. Системи лінійних рівнянь	86
4.2. Трансцендентні і алгебраїчні рівняння	98
4.3. Диференціальні рівняння	114
4.4. Чисельне інтегрування	126
4.5. Інтерполяція, екстраполяція і апроксимація	133
4.6. Математичні моделі, отримані шляхом обробки експериментальних даних	144
4.7. Питання оптимізації	166
5. Удосконалення технологічних процесів текстильної промисловості з урахуванням нелінійних характеристик деформації та тертя в зоні контакту нитки з напрямною	187

5.1. Взаємодія нитки з направляючими поверхнями малої кривизни	188
5.2. Дослідження взаємодії комплексних ниток і пряжі з циліндричною напрямною поверхнею	206
5.3. Експериментальне дослідження умов взаємодії ниток з напрямними поверхнями	238
5.4. Удосконалення технологічних процесів текстильного виробництва на основі мінімізації натягу пружної системи заправки	273
6. Сучасні САПР легкої промисловості, їх можливості і складові підсистеми	306
6.1. Апроксимація контурів деталей	318
6.2. Алгоритмічне та програмне забезпечення з підготовки вхідної інформації для автоматичного розкрою деталей шкіргалантерейних виробів	341
6.3. Оптимізація розміщення деталей шкіргалантерейних виробів при побудові розкрійних схем	364
6.4. Алгоритмічне та програмне забезпечення для генерації розкрійних схем дрібної шкіргалантереї на рулонних матеріалах	373
6.5. Інтерактивне коригування розкрійних схем, що побудовані в автоматичному режимі	392
7. Удосконалення технологічних процесів в'язань та ткацтва з урахуванням анізотропії фрикційних властивостей нитки	401
7.1. Теоретичне дослідження поперечного руху ниток з урахуванням анізотропії тертя	401
7.2. Теоретичне дослідження впливу матеріалу нитки і анізотропії тертя на її натяг і форму осі	433
7.3. Експериментальне дослідження процесу взаємодії ниток з направляючими з урахуванням анізотропії тертя	475
7.4. Дослідження технологічних процесів трикотажного та ткацького виробництв з урахуванням анізотропії тертя ниток	512
8. Удосконалення пружної системи заправки ниток на круглов'язальних машинах на основі стабілізації натягу	533
8.1. Математичні, алгоритмічні та програмні компоненти системи проектування форми лінії заправки ниток на круглов'язальних машинах	533
8.2. Удосконалення конструктивних елементів системи ниткоподачі круглов'язальних машин	574
8.3. Експериментальне дослідження впливу елементів системи подачі нитки на її натяг	617

8.4. удосконалення системи подачі ниток круглов'язальних машин	644
9. Комп'ютерний програмний комплекс для визначення технологічних зусиль на основі рекурсії	680
9.1. Структура комп'ютерної програми реалізації алгоритму рекурсії для визначення технологічних зусиль	680
9.2. Програмні модулі комп'ютерної програми реалізації алгоритму рекурсії для випадку змінного вхідного натягу	688
9.3. Комп'ютерна реалізація алгоритму рекурсії для випадку змінного діаметру сировини	694
10. Алгоритмічне та програмне забезпечення процесу взаємодії нитки з направляючими трикотажних машин з урахуванням їх форми та конструктивних параметрів	701
10.1. Геометричні, кінематичні та динамічні компоненти точок нитки при її взаємодії з напрямною великої кривизни	702
10.2. Взаємодія ниток з направляючими структурними елементами великої кривизни системи подачі нитки круглов'язальної машини	710
10.3. Визначення приведенного коефіцієнту тертя для структурних елементів системи подачі нитки круглов'язальних машин у формі тора	717
10.4. Алгоритмічне та комп'ютерне програмне забезпечення для визначення натягу нитки після напрямної великої кривизни	721
10.5. Алгоритмічне та комп'ютерне програмне забезпечення для визначення приведенного коефіцієнту тертя для структурних елементів системи подачі нитки круглов'язальних машин у формі тора	727
Література.....	734
Додатки	739

1. ЗАГАЛЬНІ ПИТАННЯ САПР

В даний час системи автоматизованого проектування знаходять найширше застосування в різних областях техніки. Це пов'язано з бурхливим розвитком обчислювальної техніки, програмного забезпечення, а також завдяки тим вигодам, які отримують проектувальники і проектні організації в цілому. Поняття “автоматизація” має на увазі під собою заміну дій людини в процесі проектування енергією неживої природи [18]. Таким чином, однією з перших переваг є скорочення тривалості циклів проектування. Останнє з перерахованих переваг обумовлено підвищенням точності проектування і мінімальним числом помилок, пов'язаних з виконанням ручних операцій на різних стадіях проектування. Використання сучасної обчислювальної техніки і адаптованого до проблем проектування програмного забезпечення дозволяють забезпечити внесення змін до конструкції проектного виробу, забезпечувати динамічнішу реакцію САПР на запити проектувальників за рахунок використання існуючих баз даних (БД), які інтегровані у використовувану САПР і є її невід'ємною частиною.

Широкий розвиток інформаційних моделей (дозволяють описувати об'єкт проектування вербально, графічно, табличний, математично), у відмінності від використання матеріальних (натурних) моделей, дозволяють забезпечити процедуру аналізу взаємодій різних компонентів проектованих виробів і спрощують завдання аналізу функціонування проектованих виробів. Це приводить до скорочення об'ємів випробувань дослідних зразків проектного виробу. Цілком очевидно, що це достатньо досягатиметься, коли на всіх рівнях проектних процедур використовуватимуться програмні продукти, які засновані на одній ідеології і на єдиній інформаційній моделі. Такий підхід пояснюється складністю систем, що вивчаються і проектованих, що природно

приводить до необхідності розробки прикладного програмного забезпечення для імітації – відтворення на ЕОМ математичних моделей функціонування проєктованого об'єкту. Дане програмне забезпечення може характеризуватися як сукупність універсальних пакетів блокового моделювання, фізичного моделювання і пакетів для наочного опису проєктованих об'єктів з складною логікою взаємозв'язків між окремими складовими.

Пакети блокового моделювання дозволяють будувати на основі принципів об'єктно-орієнтованого моделювання функціональні багаторівневі ієрархічні системи. У них повинні бути закладені можливості для візуалізації і анімації.

Постійний розвиток і вдосконалення систем управління базами даних (СУБД), комп'ютерній периферії, графічних прикладних нальотів дозволяють автоматизувати процес підготовки технічної документації на основі використання фактографічних БД (у них у вигляді каталогів, карток містяться короткі відомості про об'єкти проєктування в строго певному форматі), документальних БД (у них містяться файли документів з текстовим, графічним, мультимедійним розширенням).

Використання сучасних САПР дозволяє здійснювати стандартизацію проєктних рішень і на цій основі покращувати якість проєктних розробок.

Декілька слів необхідно сказати про ті вигоди, які може отримати виробник конкретних об'єктів, якщо разом з використанням САПР для їх проєктування він використовуватиме САПР технологічних процесів для їх виготовлення.

САПР технологічних процесів покликана вирішувати наступні завдання: імпорт конструкторської інформації, її ідентифікація, класифікація і кодування; декомпозиція (розбиття на вузли) спроектованого виробу; систематизація зберігання і пошуку конструкторської і технологічної документації; автоматизована генерація

принципово нових технологічних процесів; створення нових технологічних процесів шляхом інтерактивного редагування аналогічної інформації; автоматизованої технологічної підготовки виробництва; випуск комплектів технологічної документації; планування і управління процесом технологічної підготовки виробництва; експорт інформації в систему виробничого планування і управління проектами.

Зберігання і обробка даних здійснюється з використанням СУБД у вигляді даних реляційного типу.

Процес проектування структури САПР технологічного процесу орієнтована на процедури синтезу, моделювання і аналізу, що реалізують графічну імітацію технологічного процесу в динаміці і вибір якнайкращого варіанту структури технологічного процесу (ТП). У САПР технологічного процесу можливе завдання різних процедур синтезу структури ТП, що визначається кваліфікацією експерта-технолога, вибором і використанням одного або комбінації механізмів логічного виводу.

До основних переваг тут можна віднести підвищення продуктивності процесу проектування робочих інструментів і оснащення, підготовку програм для верстатів з ЧПУ, автоматизований контроль якості виготовлення спроектованих об'єктів. САПР технологічних процесів виготовлення спроектованих об'єктів дозволяє на базі автоматизованого планування виробництва здійснити скорочення тривалості виробничих циклів.

Декілька слів потрібно сказати про рівень автоматизації проектних процедур. Навіть зараз існує думка, що заміна кульмана електронною креслярською дошкою все одно вимагає наявності проектувальника, який повинен інтелектуально впливати на процес проектування і залишатися головною фігурою, що діє, на всіх стадіях проектування. Крім того, як можна співвіднести по рівню автоматизації проектних процедур у разі

двох організацій, в одній з яких знаходяться декілька ЕОМ, об'єднаних в єдиний комплекс засобів автоматизації проектування з відповідним програмним забезпеченням, БД, периферійним устаткуванням і організацією з одним комп'ютером, на якому встановлений Auto Cad 14? Відповідь може бути тільки одним. І в першому і в другому випадку можна говорити про автоматизацію процесу проектування. Але говорити про систему автоматизованого проектування в другому випадку говорити дуже складно. Хоча тут необхідно провести ту межу, яка при певному відсотку автоматизації проектних процедур дозволить говорити про наявність САПР. Існує велика кількість об'єктів, при проектуванні яких ще широко використовується “ручна” праця проектувальників. Дане твердження з повною підставою може бути віднесене до устаткування і технологічних процесів легкої і текстильної промисловості, які по своїй складності займають одне з перших місць. Крім того, як наголошувалося в першому розділі, навіть на автомобілебудівних фірмах США, Італії, які характеризуються високим відсотком автоматизації проектних процедур, ще немає повної 100% САПР, яка дозволила б повністю виключити участь проектувальника в процесі розробки нової моделі автомобіля. Отже, можна говорити тільки про ступінь автоматизації конкретних проектних процедур, де використання комплексу засобів автоматизації може дати істотний економічний ефект.

1.1.ХАРАКТЕРИСТИКА СТАДІЙ ПРОЕКТУВАННЯ

Згідно існуючим нормативним документам проектування будь-якого технічного виробу повинне складатися з наступних стадій (див. мал. 1.1 а): перед проектні дослідження; технічне завдання; ескізний проект; технічний проект; робочий проект (проведення випробувань); введення в

експлуатацію (впровадження). Після введення виробу в експлуатацію передбачається дослідна і промислова експлуатація.

Перед проектні дослідження є початковою стадією проектування, які дозволяють виявити потреби певної галузі у виробі, що розробляються. На цьому етапі відбувається оцінка актуальності проведення проектних досліджень, принциповій вирішуваній поставленого завдання, проводяться необхідні науково-дослідні роботи, вивчаються можливі підходи до рішення поставленої задачі. На цій стадії відбувається вибір необхідних методів інструментарію. Визначається концептуальна основа проектного об'єкту.

Технічне завдання, як одна із стадій проектування, дозволяє формулювати технічну вимогу до проектного об'єкту на основі проведених раніше передпроектних досліджень, патентної інформації.

Стадії ескізного і технічного проекту близькі між собою. Для ескізного проектування характерне ухвалення попередніх проектних рішень на основі аналізу інформації, отриманої при проведенні передпроектних досліджень і формуванні технічного завдання. При реалізації технічного проекту проводять коректування, уточнення рішень прийнятих при ескізному проектуванні. Таким чином, дані дві стадії дуже близькі між собою за змістом і характеру виконаних проектних процедур. Виходячи з цього при проектуванні іноді опускають стадію ескізного проектування (мал. 1.1 б). Роботи, передбачені на цій стадії, виконуються при реалізації технічного проекту. У разі наявності прототипів виробу або типових проектів технічний і робочий проект об'єднуються в техно-робочий проект (мал. 1.1 в).

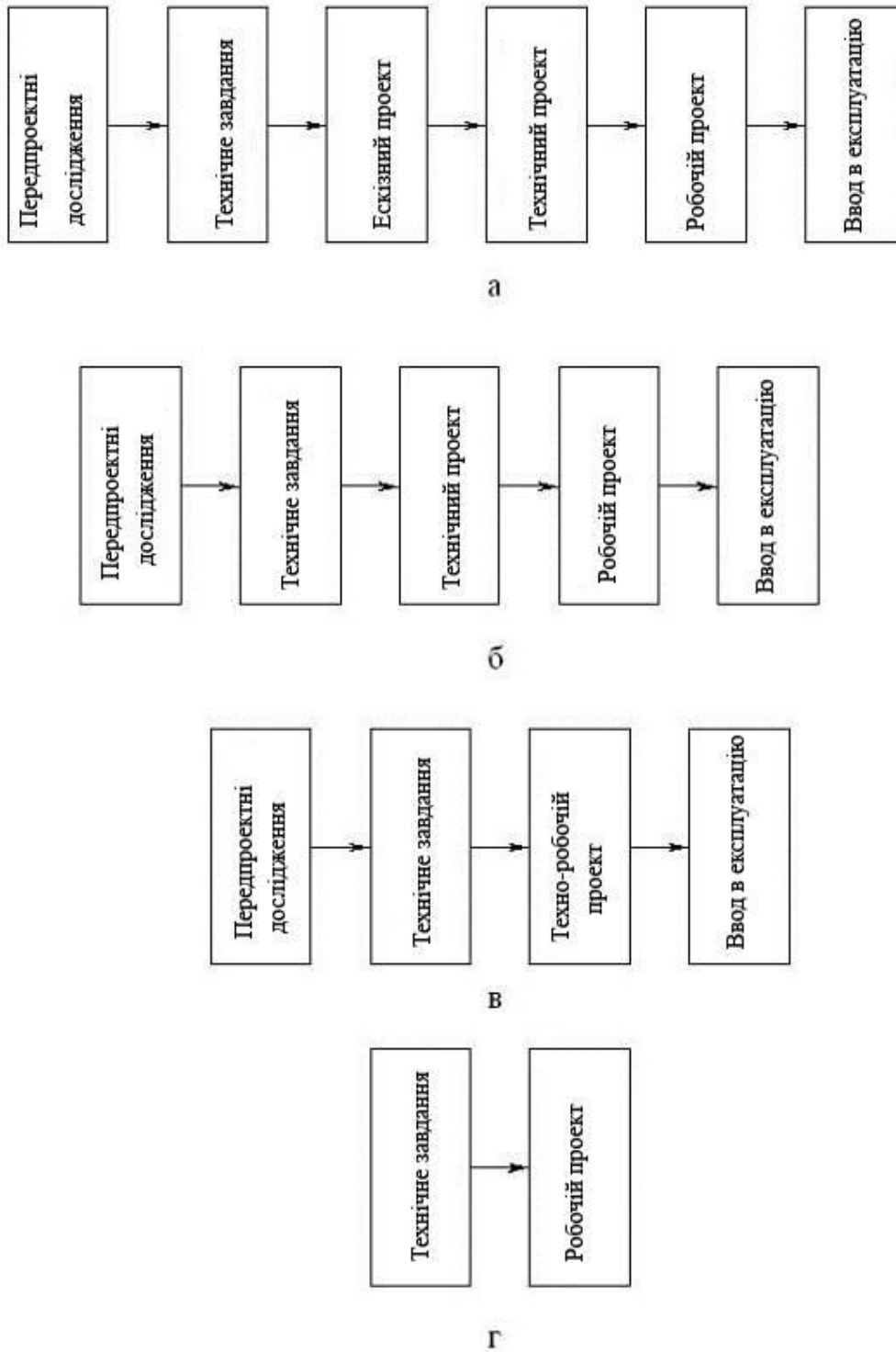


Рис.1.1

При проектуванні простих технічних об'єктів допускається реалізація двох стадій: технічне завдання і робочий проект (мал. 1.1г).

Описані вище стадії проектування дають можливість оцінити характер виконуваних конструкторських робіт на кожній стадії. Зупинимося тепер на розгляді основних операцій, які виконуються колективом конструкторів для реалізації технічного завдання. Вони отримали назву проектних процедур – сукупності дій над модельним представленням об'єкту проектування, що приводять до ухвалення рішення і характеризуються повторюваністю, відтворюваністю і інваріантністю до моделі об'єкту. Основними проектними процедурами є синтез, аналіз, оптимізація, перетворення і ідентифікація. На думку авторів даної книги останні дві проектні процедури можна віднести до підкласу проектних процедур доповнюючих процедуру синтезу основного класу проектних процедур.

Перейдемо до аналізу кожній з вказаних проектних процедур. Почнемо з процедури синтезу. У загальному випадку синтез характеризується як метод дослідження якого-небудь предмету, явища, що полягає в пізнанні його як єдиного цілого, в єдності і взаємному зв'язку його частин. Для обладнання і механізмів легкої промисловості синтез трактується як проектування схеми пристрою по заданих його властивостях і включає вибір структурної схеми пристрою по заданих його властивостях.

Проектна процедура синтезу підрозділяється на структурний синтез і параметричний синтез. Структурний синтез забезпечує опис об'єкту проектування у вигляді структури і сукупності значень конструктивних параметрів і включає вибір структури, вибір технічних режимів і складання відповідної документації. Основна початкова інформація визначається технічним завданням на проектування. Параметричний синтез виконується з метою установки значень параметрів спроектованого об'єкту із заданою структурою. Таким чином, можна стверджувати, що параметричний синтез можна проводити, якщо вже проведена процедура

структурного синтезу об'єкту проектування або якщо структура об'єкту, що синтезується, обумовлена в технічному завданні.

Побудова проектної процедури синтезу безпосередньо пов'язана з вибором технології проектування. Розрізняють дві технології проектування: низхідна технологія проектування і висхідна технологія проектування. Низхідна технологія проектування (дедуктивне проектування) характеризується послідовним або паралельним переходом від більш загального об'єкту до його складових частин (наприклад, виріб → пристрої вузли → пристрої деталі). Дані процедури перетворення виконуються до тих пір, поки проектант не дійдуть до елементарних складових в ієрархії проектованого виробу (наприклад, деталей). При низхідній технології проектування починають з уніфікації маршрутів проектування, які після розбиваються на проектні процедури і останні уніфікуються.

Висхідна технологія проектування (індуктивне проектування) характеризується композиційним переходом від елементарних складових в ієрархії проектованого виробу до складовим, які займають вищий рівень. При такому підході просто створюється інструментарій для реалізації окремих процедур. Низхідна технологія проектування характерна для процесів проектування технологічних процесів виготовлення виробів, коли на першому етапі розробляється технологія виготовлення окремих деталей, потім вузлів, потім пристроїв. Всі вони інтегруються, на останній стадії, в технологію виготовлення виробу.

Наявність двох технологій проектування диктує умови побудови різних структурних схем проектної процедури синтезу. На мал. 1.2 приведена одна з них, яка дозволяє реалізувати процедуру синтезу при дедуктивному підході до проектування. Як було відмічено вищим, процес починається із структурного синтезу.

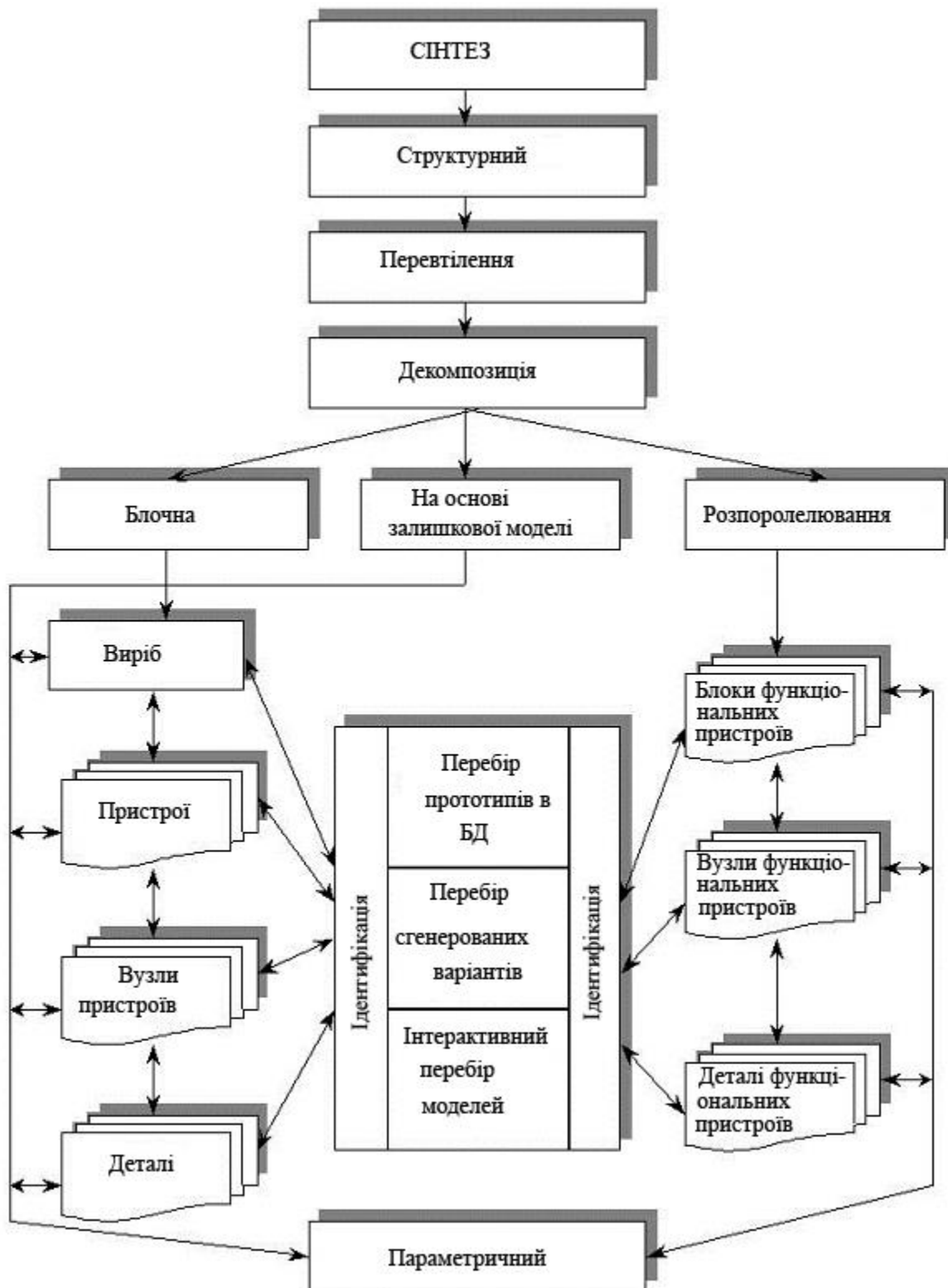


Рис.1.2

Здійснення перетворення на основі декомпозиції об'єкту виробу проектування дозволяє здійснити розбиття його на простіші складові. Декомпозиція здійснюється по окремих блоках виробу, на основі

надмірної моделі або шляхом розпаралелювання. До блоків виробу, з урахуванням положення в ієрархії структури виробу, відносяться пристрої, вузли пристроїв, деталі. Наприклад, для швейної машини (виріб) пристроями є привід, трансмісія (передає рух від приводу до робочих механізмів), механізм голки, механізм петлителя, механізм переміщення матеріалу, система подачі нитки, система мастила. Кожен з вказаних пристроїв включає сукупність вузлів, які дозволяють виконувати певні функціональні дії. Всі вузли складаються з кінематично зв'язаних ланок (деталей). Необхідно відзначити, що кінематичний зв'язок здійснюється не тільки між ланками, але і із старизною швейної машини. Тут декілька слів необхідно сказати про проектування об'єктів з різною фізичною природою. Цілком очевидно, що проектування механізму голки відрізнятиметься від проектування системи мастила. Про ці особливості буде сказано при розгляді і класифікації об'єктів проектування.

На кожному етапі блокового синтезу здійснюється ідентифікація пристроїв, вузлів пристроїв і деталей. На підставі вимог до кожного з них і до виробу в цілому, викладених в технічному завданні на проектування, проводиться вибір відповідного структурного образу. Це може бути реалізовано або шляхом перебору прототипів в БД, або шляхом перебору варіантів що згенерували над безліччю базових компонент, або шляхом інтерактивного перебору моделей.

Деталізація моделей проектування на основі розпаралелювання має на увазі під собою структурний синтез тільки окремих функціональних пристроїв, вузлів функціональних пристроїв. При цьому мається на увазі, що розроблені нові функціональні пристрої, їх вузли і деталі зажадають внесення змін до структурних зв'язків між ними і існуючими пристроями. Наприклад, розробка нових зіво утворюючих механізмів для виробництва багатощаровий технічних тканин вимагає внесення змін до структурних

зв'язків між цими механізмами і механізмами натягнення і подачі основи і відведення тканини і ін.

Декілька слів необхідно сказати про структурний синтез на основі декомпозиції з введенням надмірної моделі об'єкту проектування. Веденням надмірності є розширення мінімального складу і мінімальної області (діапазону, набору пристроїв, вузлів і ін.) значень параметрів проектованого виробу. Тут під мінімальним складом зміни значень параметрів виробу розуміється такий склад (або структура), які необхідні і достатні для реалізації виробу з вузько вираженою спеціалізацією. Наприклад, швейна машина для пришиття гудзиків. Введення надмірної моделі дозволяє виключити необхідність заміни якій-небудь частині виробу на іншу, що володіє іншими функціональними можливостями. Такий підхід дозволив би розробити “універсальну” швейну машину, яка могла б виконувати практично всі операції: сточувати деталі, обметувати краї тканини, виконувати фігурні строчки, пришивати гудзики і ін. Цілком очевидно, що такої необхідності немає. І при організації швейних потоків виникає необхідність використовувати вузькоспеціалізоване устаткування на кожному переході. Використання декомпозиції на основі надмірної моделі раціонально використовувати у тому випадку, коли можливо вводити до складу системи набір елементів, які можуть використовуватися поодиночі. В цьому випадку зміна структури досягається перемиканням цих елементів і введенням в дію одного елемента замість іншого.

Наступним етапом проектної процедури синтезу є параметричний синтез. Метою його проведення є встановлення значень параметрів проектованого об'єкту (вироби) з певною (заданою) структурою. Як видно із структурної схеми (мал. 1.2) дана процедура здійснюється по кожному блоку з урахуванням його місця в ієрархічній структурі виробу. Цілком очевидно, що це впливатиме і на якісну характеристику параметрів проектування для кожного окремого блоку.

На мал. 1.3 показана структурна схема проектної процедури синтезу при використанні висхідної технології проектування. Її характерна відмінність від структурної схеми, приведеної на мал. 1.2, полягає в реалізації структурного індуктивного синтезу на основі перетворення структури з використанням композиції. Процедура композиції може мати послідовний, паралельний або послідовно-паралельний характер.

Наступною відмінністю є зміна положення блоків в ієрархічній структурі. На перше місце стають блоки деталей (деталей функціональних пристроїв). Завершує ієрархічну структуру вибір. Необхідно відзначити, що реалізація подібної структурної схеми синтезу вимагає проведення глибоких перед проектних досліджень (НДР). При цьому аналізуються основні принципи побудови, проводяться обстеження всіх літературних джерел на дану тему, проводиться повне патентне дослідження, аналізуються всі подібні системи.

Наступною проектною процедурою є аналіз. Процедура аналізу розуміється як процес визначення вихідних характеристик об'єкту проектування при відомих вхідних параметрах або початкових умовах по якому-небудь опису об'єкту проектування (модельному уявленню). В даному випадку можна говорити про інформаційні моделі об'єктів і процесів, які можуть бути математичними, табличними, графічними і вербальними. Розділяють одно варіантний і багатоваріантний аналіз. Одно варіантний аналіз проводиться з метою встановлення відповідності вихідних характеристик вимогам технічного завдання. Багатоваріантний аналіз направлений на встановлення в явному, наочному вигляді зв'язку між вхідними даними, вихідними характеристиками і конструктивними параметрами.

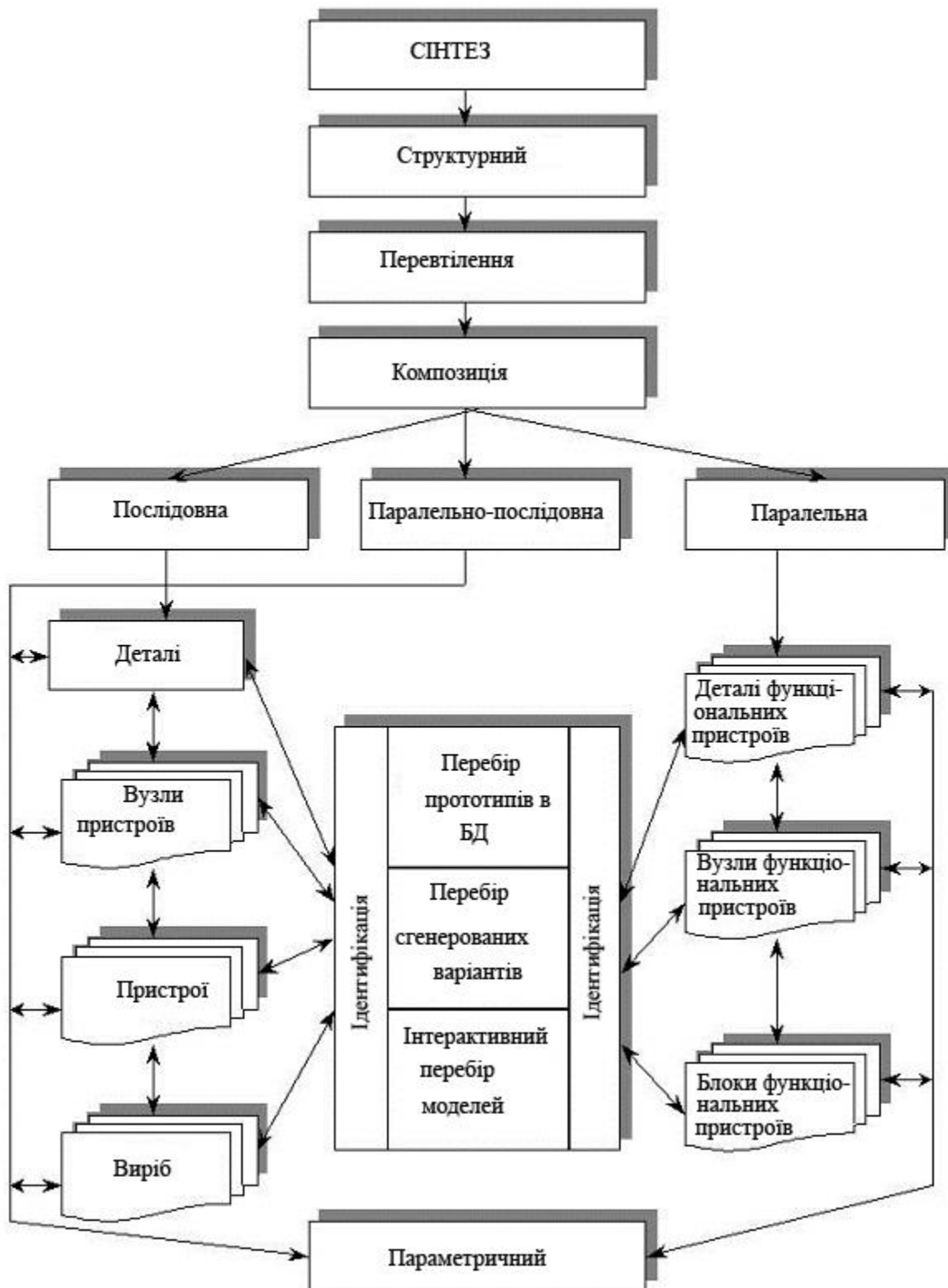


Рис.1.3

Залежно від виду представлення інформаційної моделі на цьому етапі проектної процедури можуть реалізовуватися різні чисельні методи

для вирішення систем диференціальних, лінійних, трансцендентних рівнянь.

Останньою з даних проектних процедур є оптимізація. Необхідність введення даної проектної процедури не викликає сумнівів. Оцінка ефективності проведення проектних операцій повинна базуватися на принципах оптимального проектування, при якому вдасться максимізувати (мінімізувати) значення цільових функціоналів, для реалізації яких і розробляється виріб (об'єкт проектування).

Розрізняють параметричну і структурну оптимізацію, які розглядаються як проектні процедури. Параметрична оптимізація направлена на визначення значень керованих параметрів проектованого об'єкту, якнайкращих з позиції вибраного критерію, за умови дотримання заданих обмежень і при фіксованій структурі об'єкту. З цього виходить, що параметрична оптимізація проводиться після структурної оптимізації. По числу параметрів, що оптимізуються, завдання оптимізації може бути однопараметричним і багатопараметричним, а по числу критеріїв (цільових функцій) – однокритеріальна і багатокритеріальна. Успішна реалізація процедури параметричної оптимізації безпосередньо залежить від вибору ефективного пошукового методу, який визначається на підставі аналізу конкретних особливостей цільової функції і системи обмежень.

Перейдемо до опису структурної оптимізації. З інженерної точки зору різні структури проектованого об'єкту відрізняються числом елементів, самими елементами, їх компоновкою, характером з'єднання між елементами. Завдання структурної оптимізації полягає в знаходженні глобально-оптимальної структури і глобально-оптимальних значень змінних усередині цієї структури. Майже завжди в завданнях структурної оптимізації присутні і дискретні, і безперервні змінні, тобто завдання структурної оптимізації в загальному випадку відносяться до змішаних завдань математичного програмування. При структурних перетвореннях

змінюються число і характер змінних функції. Що стосується характеру багато зв'язкової області пошуку, то окремі під області або мають різну розмірність або утворені різними наборами змінних.

Як наголошувалося вище, успішна реалізація проектної процедури оптимізації залежить від вибору ефективного пошукового методу. На мал. 1.4 показана структурна схема методів оптимізації (Дана схема не претендує на абсолютну повноту. У неї включені тільки найбільш поширені методи, для яких розроблено відповідне програмне забезпечення).

Як видно з мал. 1.4 оптимізація може бути безумовною і умовною. Кожен з цих підкласів, відповідно до класифікації Аоки, підрозділяється на три групи залежно від порядку похідних. До першої групи відносяться прямі методи (методи нульового порядку), в яких використовуються тільки значення цільових функцій. До другої групи відносяться методи першого порядку, в яких використовуються перші похідні. До третьої групи відносяться методи, в яких використовуються другі похідні.

Залежно від числа параметрів розрізняють одно параметричні завдання і багато параметричні завдання.

Ряд завдань умовної оптимізації можна зводити до завдань безумовної оптимізації за допомогою реалізації спеціальних методів.

Перейдемо до короткого аналізу вказаних методів. Методи виключення інтервалів дозволяють визначити екстремум функції одного змінного шляхом послідовного виключення под інтервалів. Різновидом даного методу є метод дихотомії, метод золотого перетину (інтервал ділиться на дві нерівні частини так, що відношення довжини великого відрізаня до довжини всього інтервалу рівне відношенню довжини меншого відрізаня до довжини більшого відрізаня), метод Фібоначчі.

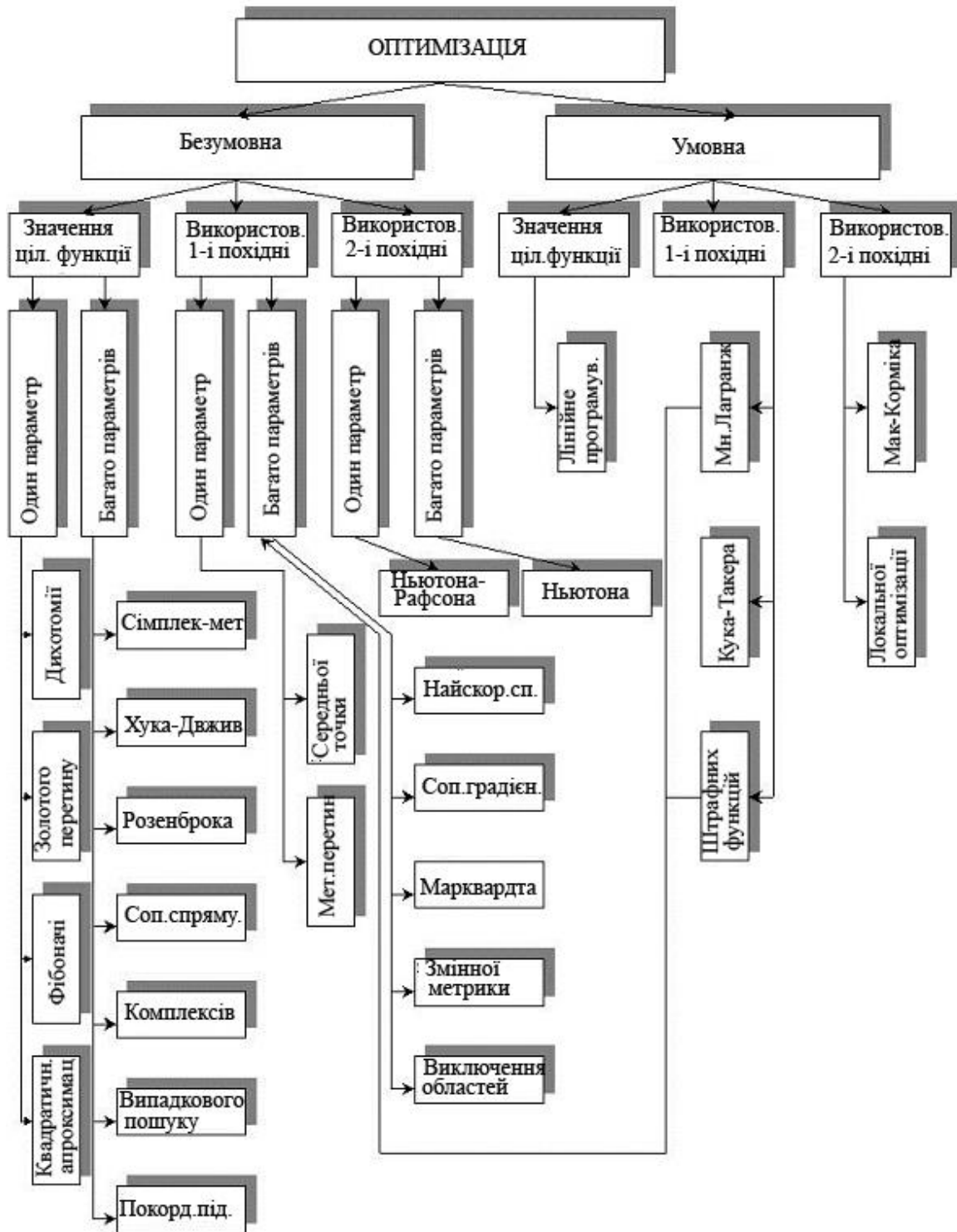


Рис.1.4

Метод квадратичної апроксимації пов'язаний з можливістю апроксимації гладкої функції поліномом і подальшого використання координати точки оптимуму.

До методів 0-го порядку для багатьох параметрів відносяться симплекс-методи, метод Хука-Джівса, метод конфігурацій Розенброка, метод зв'язаних напрямів, метод комплексів, метод випадкового пошуку, метод по координатного підйому.

До методів безумовної оптимізації 1-го порядку відносяться: для одного параметра – метод середньої крапки і метод січних; для багатьох параметрів – метод Коші (швидкого спуску), метод Марквардта, метод зв'язаних градієнтів, метод Девідона-Флетчера-Пауела (змінної метрики), метод виключення областей.

Необхідно відзначити, що використання методів 1-го і 2-го порядків при умовній і безумовній оптимізації вимагає визначення значень першою і другою похідних, ускладнює процес пошуку екстремуму цільових функціоналів.

Таким чином, три основні проектні процедури (синтез, аналіз, оптимізація) є тією конструкторською базою, яка дозволяє реалізувати процес проектування виробу. На мал. 1.5 представлена структурна схема, яка ілюструє послідовність розташування вказаних проектних процедур. На етапі синтезу здійснюється структурний і параметричний синтез конкретного компонента або вузла проектованої системи. Після цього здійснюється аналіз, який направлений на визначення вихідних характеристик об'єкту проектування при відомих вхідних параметрах або початкових умовах по якому-небудь опису об'єкту проектування. В процесі аналізу здійснюється удосконалення проектованого об'єкту, що показане стрілкою на мал. 1.5. Наступним етапом є оптимізація структури і параметрів проектованого об'єкту і здійснюється перевірка відповідності отриманих результатів технічному завданню. У разі не відповідності необхідно повернутися на перший етап. Цей цикл повторюється до тих пір, поки не буде отримано проектне рішення, оптимальне в умовах проектних обмежень.

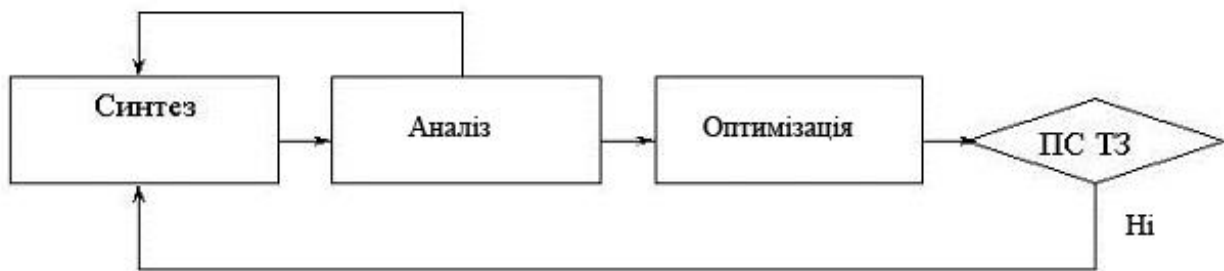


Рис.1.5

Представлений на мал. 1.5 ітераційний блок надалі називатимемо інтегрованим ітераційним проектним модулем ІПМ, який присутній в ієрархічній системі проектування на всіх рівнях, – починаючи від мета рівня і кінчаючи мікро рівнем (при проектуванні технологічних процесів маємо зворотний порядок – від мікро рівня до мета рівня). Таке послідовне використання ІПМ дозволяє синтезувати спроектовані компоненти і підсистеми в рамках остаточного проектного рішення по всій проектованій системі в цілому.

На мал. 1.6 і 1.7 представлені загальні структурні схеми ієрархічних систем проектування для устаткування легкої і текстильної промисловості (мал. 1.6) і технологічних процесів (мал. 1.7).

Перейдемо до аналізу представлених схем. Почнемо із структурної схеми на мал. 2.6. Необхідна початкова інформація формулюється в технічному завданні (ТЗ) на проектування. Для устаткування легкої і текстильної промисловості об'єктом проектування мета рівня (ОПМЕУ) є швейні, трикотажні, текстильні машини, машини підготовчого і обробних виробництв і ін. Використовуючи дедуктивну технологію низхідного проектування, здійснюється перехід від ОПМЕУ до об'єктів проектування макрорівня (ОПМАУ).

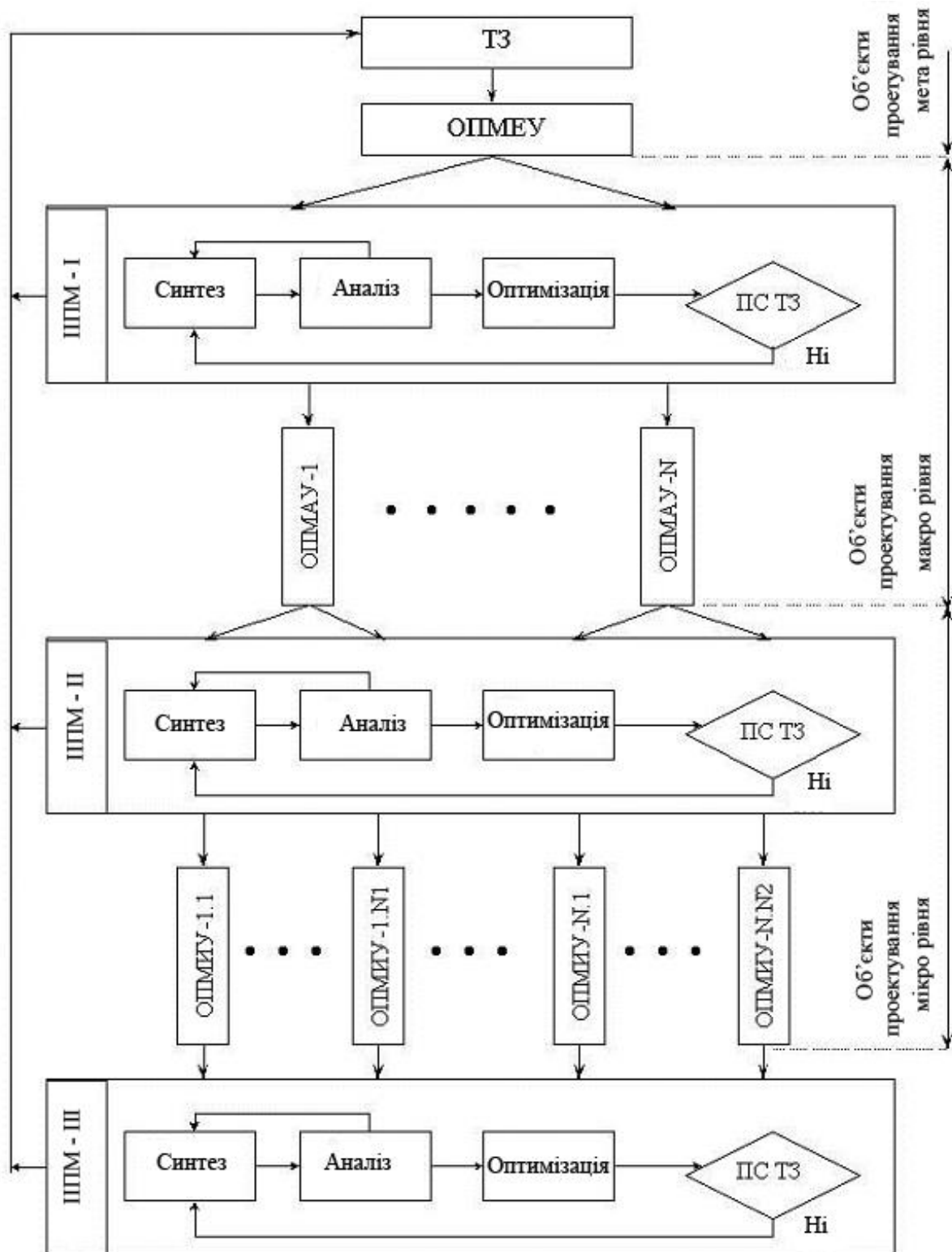


Рис.1.6

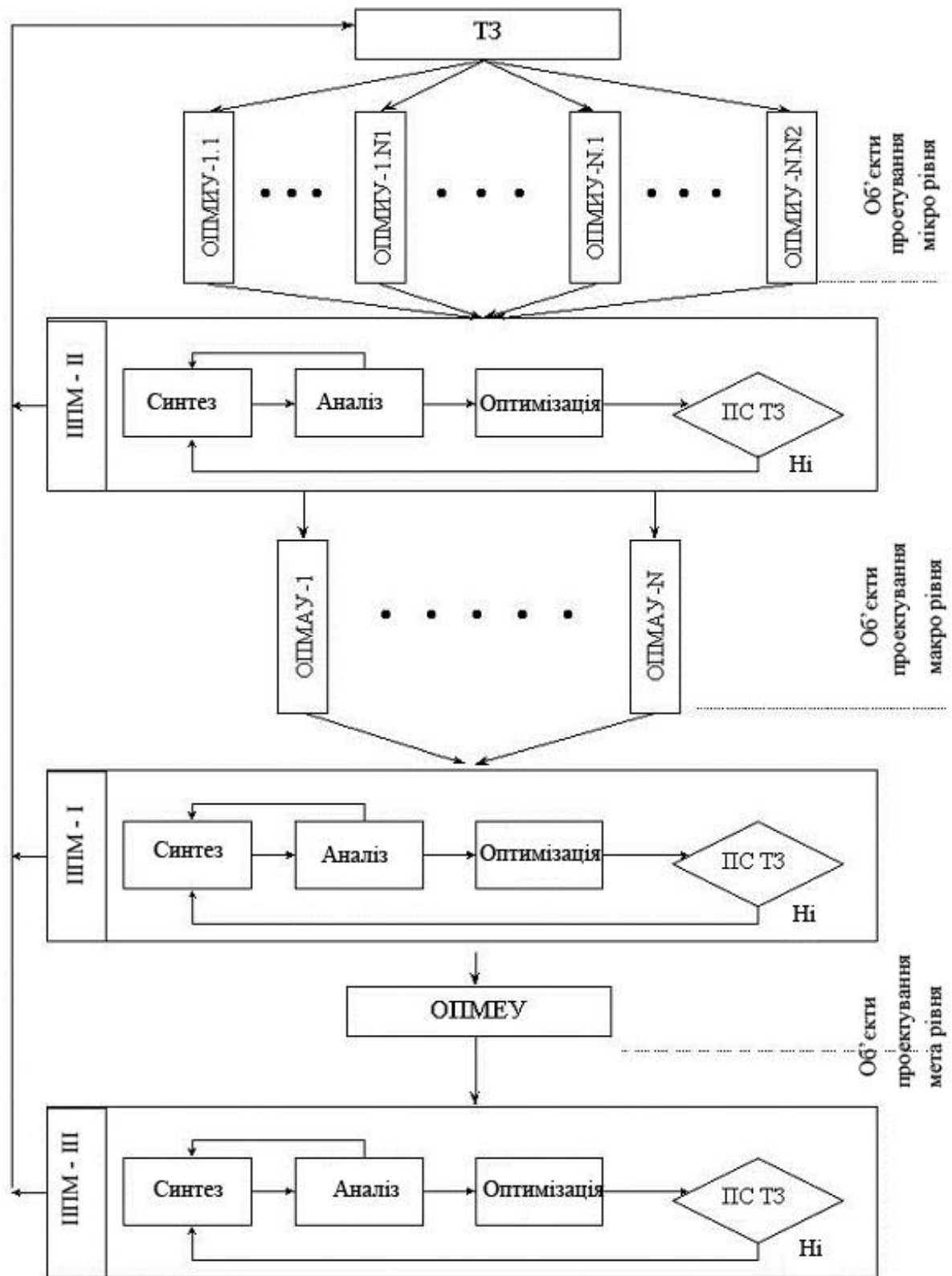


Рис.1.7

До таких об'єктів можна віднести основні механізми і пристрої машин легкої промисловості. У загальному випадку вони утворюють деяку множину $\{ \text{ОПМАУ-1}, \text{ОПМАУ-N} \}$, де N – загальне число основних механізмів і пристроїв проектованого устаткування. Основним елементом на цьому етапі є модуль ІППМ-І.

Тут необхідно відмінити специфічну реалізацію проектних процедур синтезу, аналізу і оптимізації модуля ІППМ для різних стадій проектування (див. рис.1.1). Так для стадії ескізного проекту характерні роботи, пов'язані з перевіркою коректності і реалізовується основних принципів, закладених в ТЗ. Для стадії технічного проекту характерне всестороннє опрацювання всіх частин проекту і деталізація всіх технічних рішень.

Виходячи з цього можна рекомендувати на цьому етапі при реалізації ІППМ-І на стадії синтезу реалізувати структурний синтез. Питання, пов'язані з аналізом, повинні бути орієнтовані на загальний опис динамічних, енергетичних і ін. характеристиках проектованого об'єкту. Як видно з схеми в процесі реалізації ІППМ-, можливе коректування початкових даних на проектування, приведені в ТЗ. Необхідно відзначити, що ОПМАУ можуть бути складними механічними системами, які необхідно розбивати на простіші вузли. Таким чином, структурна схема розширюватиметься за рахунок додаткових підрівнів. Дані підрівні включатимуть проектні модулі розташовані нижче в ієрархічній системі об'єкту проектування, чим об'єкти множини $\{ \text{ОПМАУ-1}, \text{ОПМАУ-N} \}$.

Наступний етап полягає в переході від ОПМАУ до об'єктів проектування мікро рівня (ОПМІУ). До даних об'єктів відносяться прості деталі, з яких складаються вузли і пристрої. Сюди відносяться і стандартні елементи (різні сполучні деталі, ущільнювачі, перехідники і ін.). Дані об'єкти утворюють деяку систему множин $\{ \text{ОПМІУ-1.1}, \dots, \text{ОПМІУ-1.N1} \} \cdot \{ \text{Опміу-n.1}, \dots, \text{опміу-n. N2} \}$, де $N1$ – загальне число об'єктів проектування мікро рівня з яких полягає об'єкт макрорівня ОПМАУ-1; $N2$ – загальне

число об'єктів проектування мікро рівня з яких полягає об'єкт макрорівня ОПМАУ- N.

Тут необхідно відзначити, що під об'єктами проектування мікро рівня мається на увазі об'єкти, які вимагають оригінального опрацювання (різні функціональні елементи механізмів машин легкої і текстильної промисловості, нитко-, ткане спрямовувачі і ін.). Стандартні елементи вибираються з відповідних баз даних (де приведені їх основні характеристики). Дані стандартні елементи утворюють окрему множину (на мал. 2.1 не показано), яка може об'єднуватися з множиною { ОПМІУ-1.1., ОПМІУ-1.N1 }. { Опміу-п.1., опміу-п. N2 }. В процесі реалізації ІІІМ-ІІ можливе коректування початкових даних на проектування які приведені в ТЗ.

На завершальній стадії відбувається реалізація ІІІМ-ІІІ. Це дозволяє синтезувати спроектовані компоненти і підсистеми в рамках остаточного проектного рішення по всій спроектованій системі в цілому. В цьому випадку також можливе коректування початкових даних на проектування в ТЗ.

На мал. 1.7 представлена загальна структурна схема ієрархічної системи проектування для технологічних процесів легкої і текстильної промисловості.

Необхідна початкова інформація формулюється в технічному завданні (ТЗ) на проектування. У відмінності від схеми на мал. 1.6 (для устаткування) процес проектування починається з розгляду об'єктів мікро рівня, які утворюють деяку систему множин { ОПМІУ-1.1., ОПМІУ-1.N1 }. { Опміу-п.1., опміу-п. N2 }, де N1 – загальне число об'єктів мікрорівня однієї структурної або фізичної природи; N – загальне число синтезованих об'єктів ОПМАУ для даного технологічного процесу; N2 – загальне число об'єктів мікро рівня по кожному з ОПМАУ (у загальному випадку $N1 \neq N2$). Під даними об'єктами маються на увазі різні нитки,

пряжа, нитки різних систем (основні, утокові), різні наповнювачі і композиційні матеріали (ватин, тканини, вставки багат шарових тканин і ін.), тканини (для технологічних процесів швейного виробництва), шліхтувальні і фарбувальні склади і ін.

Використовуючи індуктивну технологію висхідного проектування здійснюється перехід від ОПМІУ до ОПМАУ. На цьому етапі реалізується ШПМ-II. На стадії синтезу реалізується його складові елементи – структурний і параметричний синтез об'єктів проектування макрорівня. В процесі аналізу, на основі інформаційної моделі визначаються основні співвідношення між параметрами об'єктів ОПМАУ. На стадії оптимізації і перевірки відповідності ТЗ відбувається оцінка отриманих рішень і коректуються (якщо необхідно) початкові дані ТЗ на проектування. До об'єктів ОПМАУ, які складають деяку множину { ОПМАУ-1., ОПМАУ-N} відносяться бобіни, ткацькі навої, котушки основов'язальних машин, утокові шпулі, пакети матеріалів і ін.

На наступному етапі, при реалізації ШПМ-I, синтезується ОПМЕУ, який і є технологічним процесом. На цьому етапі при реалізації проектних процедур синтезу, аналізу і оптимізації здійснюється структурний і параметричний синтез ОПМЕУ, проводиться всебічний аналіз впливу різних параметрів, на значення цільових функціоналів, що описують фізико - механічні властивості об'єктів, які переробляються на технологічному устаткуванні. Оптимізація вхідних параметрів дозволяє мінімізувати (максимізувати) значення цільових функціоналів. На цьому етапі також передбачено коректування початкових даних в ТЗ.

На завершальній стадії відбувається реалізація ШПМ- IIIз метою синтезування спроектованих об'єктів в рамках остаточного проектного рішення по всій спроектованій системі в цілому.

Декілька слів необхідно сказати про положення ієрархічних об'єктів проектування в структурній схемі проектування, яка передбачена

відповідними нормативними документами (мал. 1.1). На мал. 1.8 представлена схема, яка визначає приналежність (визначає відповідність) ієрархічних об'єктів до окремих етапів процес проектування устаткування і технологічних процесів.

Як видно з мал. 2.8, для устаткування, ієрархічні об'єкти ОПМЕУ і ОПМАУ відповідають етапу ескізного проектування, на якому здійснюється перевірка коректності і реалізовується основних принципів закладених в ТЗ. Даний етап характерний ухваленням попередніх проектних рішень.

На етапі технічного проекту здійснюється детальне опрацювання окремих елементів, що відповідає реалізації ІППМ- ІІ для об'єктів мікро рівня.

Для технологічних процесів характерна висхідна індуктивна технологія проектування. В зв'язку з цим етапу ескізного проекту відповідають структурні ієрархічні об'єкти мікро- і макрорівня (ОПМІУ і ОПМАУ). Синтезування отриманих рішень на етапі технічного проекту дозволяє завершити процес розробкою об'єкту мета рівня (ОПМЕУ) – технологічного процесу.

Як зауваження необхідно відзначити, що залежно від ступеня опрацювання проекту на стадії перед проектних досліджень (визначення наявності прототипів, типових проектів, оцінка складності конструкції), як наголошувалося раніше. Можливе об'єднання окремих етапів. Так, для простих деталей, допускається об'єднання етапів ескізного і технічного проекту (техніко - робочий проект). У такому разі проектування ієрархічних об'єктів ОПМЕУ, ОПМАУ і ОПМІУ здійснюється на етапі техніко - робочого проекту.

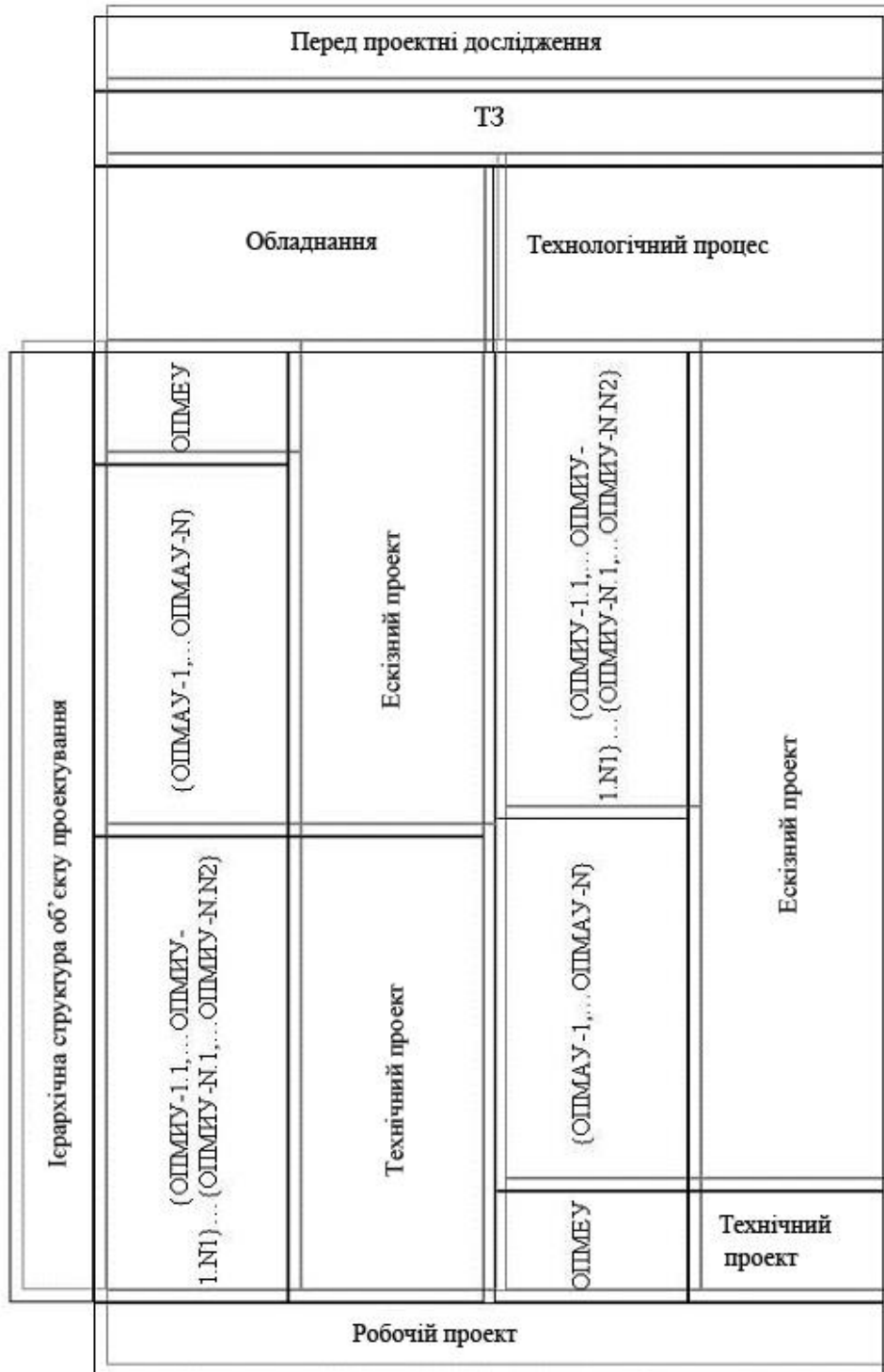


Рис.1.8

1.2. ХАРАКТЕРИСТИКА ФУНКЦІОНАЛЬНИХ СХЕМ

САПР, ЇХ ПІДСИСТЕМ І КОМПОНЕНТІВ

Приведена вище характеристика стадій проектування дозволяє прослідкувати весь розробки об'єкту (проектування) починаючи від перед проектних досліджень і закінчуючи робочим проектом. Розвиток обчислювальних засобів, ЕОМ, програмного забезпечення створило передумови для автоматизації окремих стадій проектування. На перших порах автоматизувалися проектні процедури, пов'язані з виконанням розрахунків і підготовкою конструкторської документації, креслень.

На мал. 1.9 представлена структурна схема процесу проектування, запропонована Shigley I.E., на якій показані процедури автоматизованого проектування. Зупинимося на її докладному аналізі. Автор, розглядаючи процес проектування, відразу зупиняється на трьох основних проектних процедурах: синтезі, аналізі і оптимізації. Але тут необхідно відразу відзначити, що дані процедури присутні і на стадії ескізного проектування і на стадії технічного проектування. Даний недолік приведеної схеми спричиняє за собою наступні неточності. Так, розглядаючи синтез проектних рішень, автор як процедура автоматизованого проектування указує геометричне моделювання. Але з цим твердженням важко погодитися. Загально відомо, що синтез включає дві складові. Це структурний синтез (куди відноситься геометричне моделювання як одна з складових) і параметричний синтез.

Викликає сумнів (з погляду автоматизації виконання) можливість автоматизованого виконання процедури огляду і оцінки проектних рішень. Таким чином, можна стверджувати, що по суті автоматизація процесу проектування охоплюватиме питання пов'язані з проведенням розрахунків і підготовкою текстової і графічної документації. Як наголошувалося раніше, в даному випадку не можна говорити про систему автоматизованого проектування, а тільки можна затверджувати про автоматизацію окремих етапів.

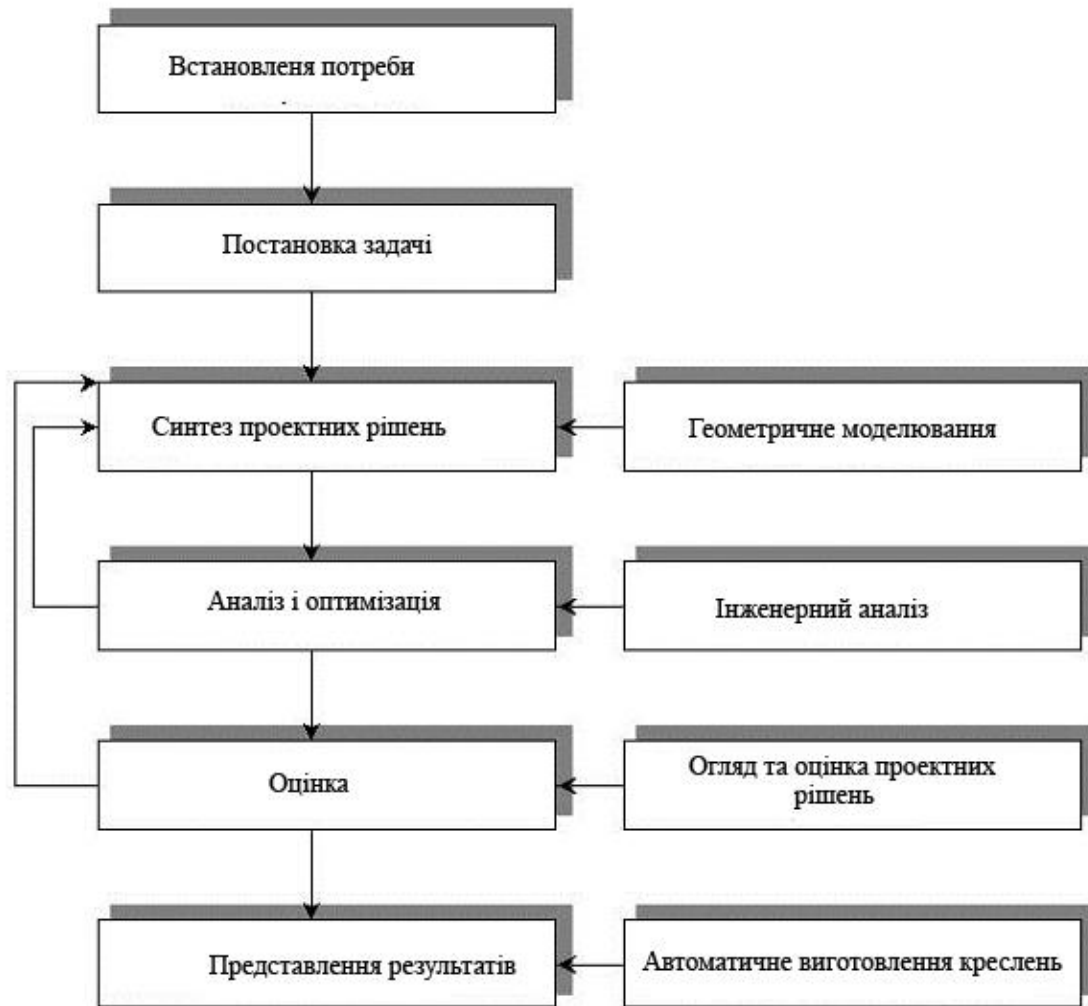


Рис.1.9

Враховуючи ієрархічну структуру об'єкту проектування (див. мал. 1.8) можна стверджувати, що оцінку рівня автоматизації проектування необхідно проводити шляхом оцінки рівня автоматизації ІПМ-I, ІПМ-II, ІПМ-III на стадіях проектування об'єктів дозволить оцінити рівень автоматизації і на стадії ескізного проекту і на стадії технічного і робочого проектів.

Переходячи до розгляду автоматизації проектування як системи необхідно перш за все зупинитися на аналізі проектуючих і обслуговуючих підсистем САПР, які складаються з відповідних підсистем. Як наголошувалося вище, що проектують підсистеми включають ІПМ-I,II,III,

які послідовно реалізуються на стадії проектування об'єктів мета-, макро- і мікро рівня виробу.

Визначення обслуговуючих підсистем витікає з характеристики самої САПР, як сукупності об'єднаних в пакети бібліотеки і модулі програм, що дозволяють реалізувати певні алгоритми в послідовності, визначеній програмами – диспетчерами, які здійснюють зв'язок з автоматизованими робочими місцями і периферією, призначеною для випуску конструкторської документації [6]. Отже, обслуговуючі підсистеми призначені для зберігання і пошуку інформації, для організації обчислювального процесу, для обробки графічної інформації і формування проектної текстової документації.

Описані вище підсистеми, які є самостійними частинами САПР, також підрозділяються на компоненти. Відповідною нормативною документацією виділяється мережа таких компонентів, які присутні в будь-якого вигляду САПР: методичне забезпечення; математичне забезпечення; лінгвістичне забезпечення; програмне забезпечення; організаційне забезпечення.

Методичне забезпечення включає документи, що визначають порядок використання САПР при проектуванні виробу. Дані документи носять інструктивний характер, їх розробка базується цілком на творчому підході і насилу піддається автоматизації.

Математичне забезпечення вдає із себе сукупність математичних методів, модулів і алгоритмів проектування технічного об'єкту, які з достатнім ступенем точності дозволяють описувати структурні і функціональні характеристики самого технічного об'єкту і його складових частин. Математичне забезпечення в явному вигляді в САПР не використовується, а є базою для розробки певного програмного забезпечення, яке дозволяє автоматизувати даний етап на базі використання сучасних ЕОМ і відповідної комп'ютерної периферії.

Другою складовою математичного забезпечення є математичне забезпечення, яке дозволяє описати технологію автоматизованого проектування, де враховуються не тільки традиційні проектні процедури, але і специфіка їх реалізації з використанням технічних засобів САПР. Все це дозволяє зробити вивід про те, що системний підхід до проектування ставить перед математичним забезпеченням завдання опису у взаємозв'язку об'єкту, засобів і процесу автоматизації проектування.

Лінгвістичне забезпечення полягає у використанні спеціальних проблемно-орієнтованих мов (ПОМ) в процесі проектування які використовуються для забезпечення діалогу проектанта з технічними і програмними засобами автоматизації даного процесу. ПОМ орієнтовані на певні проектні процедури і об'єкти проектування. До лінгвістичного забезпечення відносяться бази даних специфічної термінології, які формуються на стадії підготовки методичного забезпечення САПР.

Одній з сучасних підсистем САПР є система графічного моделювання AUTOCAD [17], яка є успішною реалізацією ПОМ в прикладному програмному пакеті. До складу основних компонентів ПОМ входить: інтерпретатор ПОЯ; засоби синтаксичного аналізу; набір термінальних символів ПОЯ; засоби пакування директив; засоби організації режимів; бібліотека базових функцій.

Одним з основних компонентів є програмне забезпечення, яке можна розділити на загальносистемне і спеціалізоване. До загальносистемного програмного забезпечення відносяться операційні системи (здійснюють: управління процесом обчислення; введення і виведення інформації; управління виконанням всіх інших програм; забезпечують ефективність використання апаратури ЕОМ), бібліотеки стандартних програм, системи управління базами даних (СУБД). До спеціалізованого програмного забезпечення відносяться різні пакети прикладних програм конкретних проектних рішень.

Інформаційне забезпечення САПР складається з даних (БД) на електронних носіях проектно - конструкторської документації. Дані бази містять інформацію про прототипи, як самого проєктованого виробу, так і про його складові, що комплектують. У них зберігається інформація про матеріали, які використовуються для виготовлення окремих елементів. Особливе місце займає інформація, яка відноситься до технологічного процесу виготовлення виробу, використовуваному при цьому інструменту технологічним режимам.

Відповідна нормативно-технічна документація регламентує правила і норми проектування, правила документування отримуваних результатів.

Автоматизація інформаційного забезпечення САПР пред'являє підвищені вимоги до створення баз даних. Цей процес охоплює теоретична побудова інформаційної моделі, формування структури таблиці в процесі роботи в середовищі СУБД, відкриття файлу (вказівка імен, типів і форматів виведення всіх полів таблиці), заповнення бази даних. При цьому особливу увагу необхідно приділяти рівню нормалізації таблиці без наявності збитковості.

Технічне забезпечення САПР включає комплекс обчислювальних засобів, набір периферійних пристроїв (дисплей, клавіатури, ручні маніпулятори, принтери, сканери, графічні пристрої), які призначені для забезпечення діалогового режиму роботи, випуску конструкторської документації.

Залежно від ієрархічної структури САПР технічною забезпечення може змінюватися. Наприклад, для дворівневої ієрархічної структури окрім окремих автоматизованих робочих місць (АРМ) передбачається наявність високопродуктивної ЕОМ з великим об'ємом оперативної і зовнішньої пам'яті. Рівень комплектації АРМ залежить від складності вирішуваних завдань. Широкий розвиток персональних ЕОМ, збільшення

їх швидкодії, об'єму пам'яті привело до того, що вони знаходять все більше застосування в САПР.

Організаційне забезпечення САПР визначає і регламентує взаємодія між різними структурними підрозділами проектної організації на основі комплексу директивних документів. Вони також визначають правила підготовки конструкторської документації, доступу до баз даних встановлюють пріоритети користування програмними і технічними засобами САПР структурними підрозділами і окремими конструкторами. Види, комплектність і позначення документів при створенні автоматизованих систем проектування (відповідно до ГОСТ 34.201-89) приведені в додатку.

Після розгляду підсистем САПР і компонентів, що становлять, перейдемо до аналізу структурно - функціональних схем останніх, які показані на мал. 1.10 і 1.11.

На мал. 1.10 представлена структурно - функціональна схема одно рівневої ієрархічної структури САПР, яка побудована на основі комплексу автоматизованих робочих місць (АРМ). Кожне автоматизоване робоче місце АРМ-1, АРМ-2.,АРМ-N включає з погляду технічного забезпечення системні блоки ЕОМ-1, ЕОМ-2.,ЭВМ-N, відповідне периферійне устаткування ПРО-1, ПРО-2.,ПРО-N, яке включає монітори, ручний маніпулятор, клавіатуру, принтер, сканер, графічний пристрій і ін.

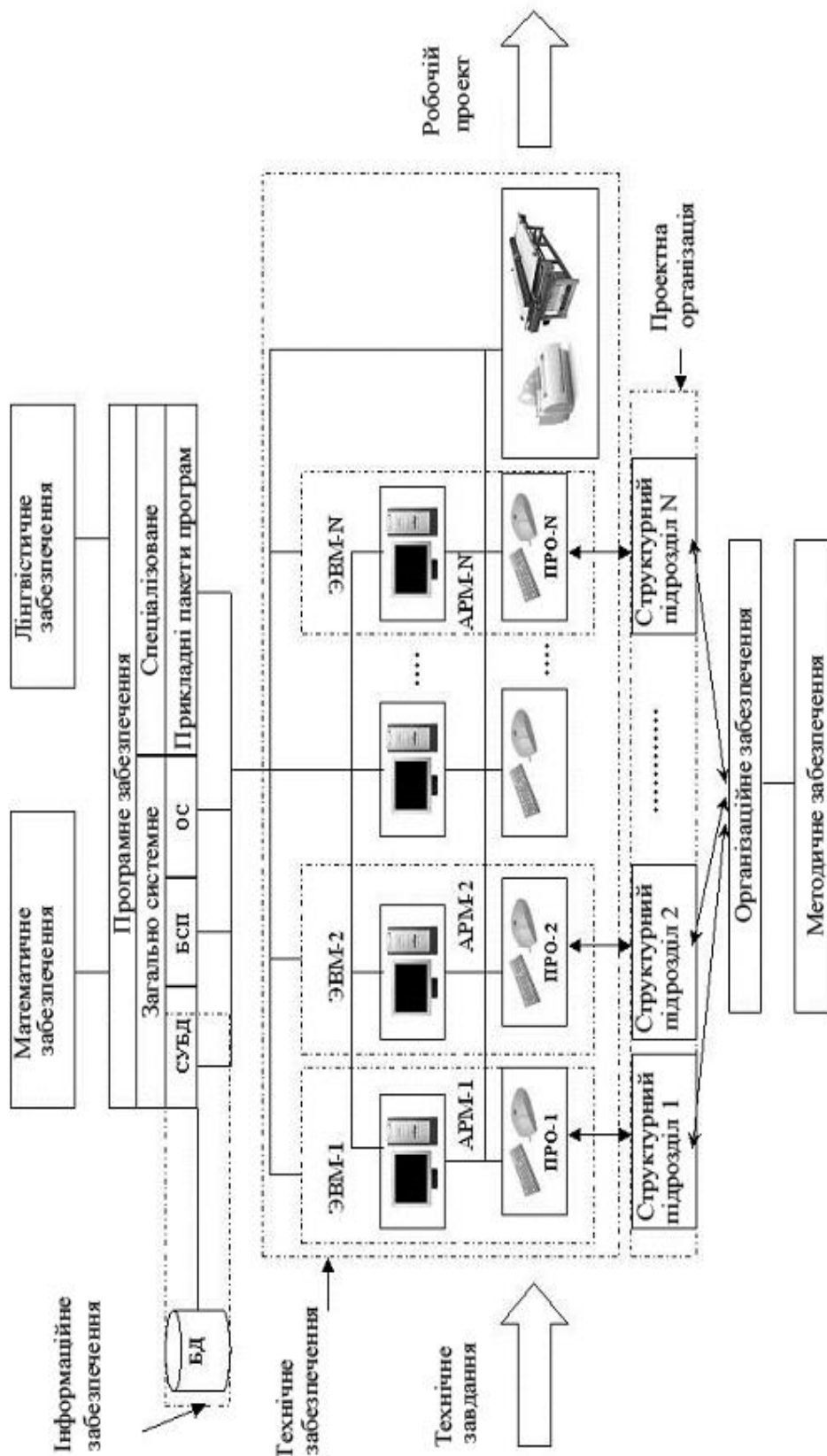


Рис.1.10

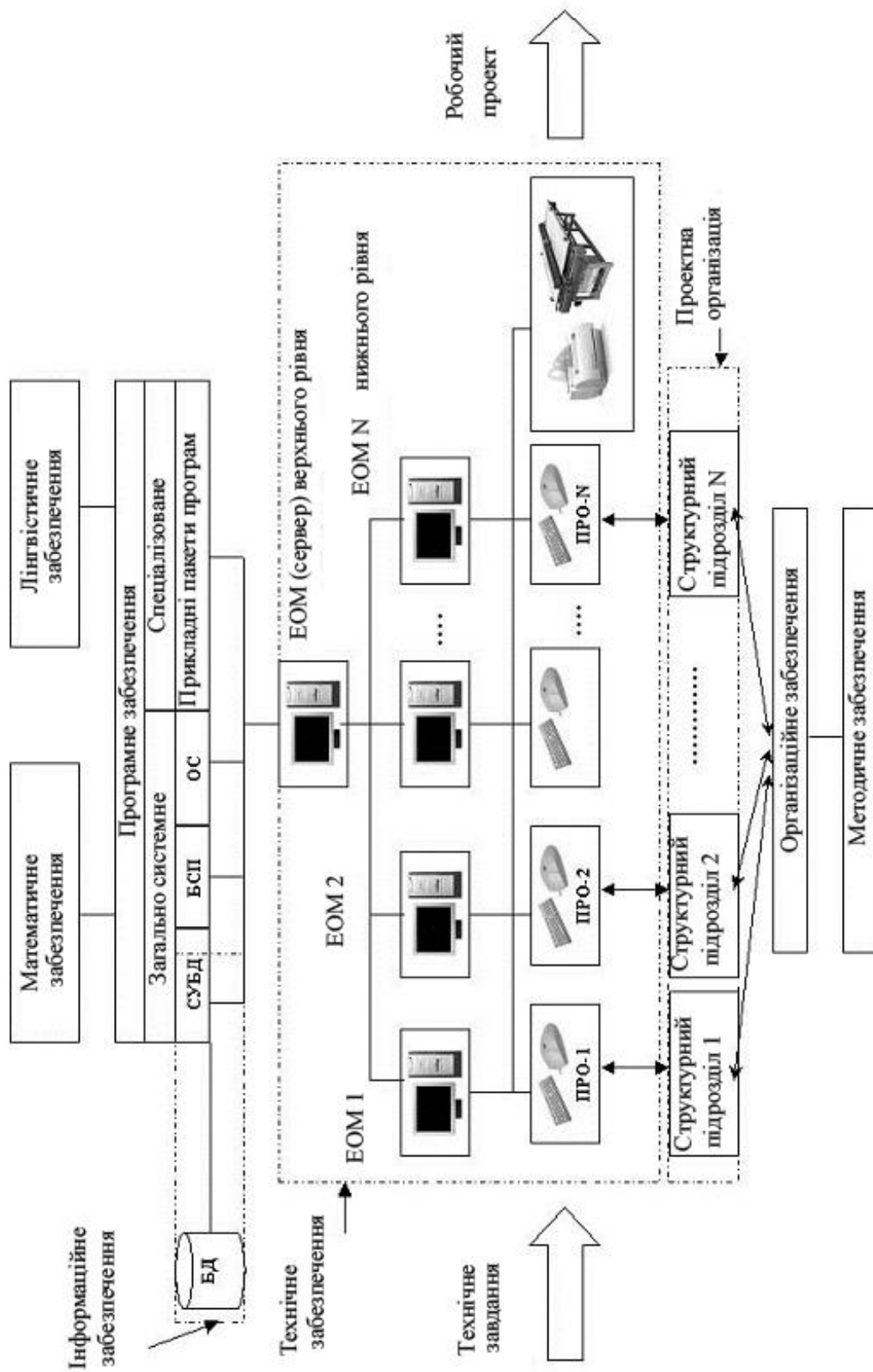


Рис.1.11

Необхідно відзначити, що організація локальної комп'ютерної мережі дозволяє упорядкувати використання засобів комп'ютерної периферії і скоротити її чисельність, а це приведе до зниження вартості самої САПР.

Кожне АРМ оснащується відповідним математичним, програмним і лінгвістичним забезпеченням у вигляді програм, пакетів програм. Програмне забезпечення складається із загальносистемного і спеціалізованого ПО. Загальносистемне включає систему управління базами даних, бібліотеку. Спеціалізоване ПО складається з прикладних пакетів програм. Інформаційне забезпечення включає безпосередньо бази даних (БД) і СУБД.

На вході в системі маємо технічне завдання проектування, а на виході робочий проект.

Проектна організація складається із структурних підрозділів 1,2,..,N, кожне з яких (або навіть окремий конструктор-проектувальник) виконує певний вид роботи. Види даних робіт, а також взаємодія між окремими структурними проектними підрозділами визначається відповідним організаційним забезпеченням. Кожен структурний підрозділ (або певний конструктор-проектувальник) має доступ через ПРО до відповідного АРМ.

Документи експлуатації, що регламентують порядок, САПР складають методичне забезпечення.

На мал. 1.11 представлена структурно - функціональна схема дворівневої ієрархічної структури САПР з ЕОМ великої продуктивності верхнього рівня. Нижній рівень складають АРМ, об'єднані в мережу. ЕОМ верхнього рівня може виконувати роль сервера локальної мережі і надавати проектантам можливість використовувати своїми ресурсами при реалізації проектних процедур.

Дворівневі САПР володіють великими функціональними можливостями, є продуктивнішими в порівнянні з одно рівневими, які

були розглянуті вище. Вони призначаються для проектування складних об'єктів, які характеризуються великим числом ОПМАУ і ОПМІУ. У легкій і текстильній промисловості вони можуть використовуватися при проектуванні таких складних об'єктів як трикотажні машини, ткацькі верстати, технологічні процеси підготовчого і обробного виробництва.

1.3. КЛАСИФІКАЦІЯ САПР

Згідно існуючої нормативної документації (ГОСТ 23501.8-80) в основу САПР покладені тип об'єктів, їх складність, рівень автоматизації проектних процедур, комплектність, характер проектних документів, що випускаються, продуктивність.

По типах об'єктів можна розрізняти САПР виробів легкого і текстильного машинобудування і технологічних процесів. По складності можна виділити САПР: простих об'єктів (до 102 складових частин); об'єктів середньої складності (від 102 до 103 складових частин); складних об'єктів (від 103 до 104 складових частин); дуже складних об'єктів (від 104 до 106 складових частин); об'єктів дуже високої складності (від 106 складових частин).

Рівень автоматизації проектних процедур може бути низьким (рівень автоматизації процедур складає 25% від загальної кількості), середнім (рівень автоматизації 50% від загальної кількості), високим (рівень автоматизації процедур складає понад 50% від загальної кількості).

Ступінь комплектності САПР може бути підрозділена на одно етапні САПР, багатоетапних і комплексних. Характер проектних документів, що випускаються, дозволяє провести наступну класифікацію: документація може бути текстовою і графічною; документи можуть зберігатися на якомусь одному типі носія даних або на декількох. У загальному випадку інформація може зберігатися на всіх типах носіїв даних.

САПР можуть бути малій, середній і високій продуктивності (визначається кількістю проектних документів тих, що випускаються за

рік). По кількості рівнів в структурі технічного забезпечення розрізняють одно рівневі САПР (побудовані на базі ЕОМ і наборі периферійних пристроїв), дворівневі (побудовані на базі ЕОМ і одного або декількох АРМ), три рівневі (побудовані на базі ЕОМ, одного або декількох АРМ і периферійного програмно-керованого устаткування).

Необхідно відзначити, що САПР необхідно розглядати як нерозривна ланка “проектувальники – технічне забезпечення – програмне забезпечення”. Виходячи з цього, а також враховуючи решту компонентів підсистем проектування класифікувати САПР можна по наступних характеристиках: загальні; програмні; технічні; ергономічні. Останні оцінюють ефективність взаємодії користувача з програмно-технічними засобами САПР.

До загальних можна віднести класифікацію за способом організації інформаційних потоків: індивідуальні автоматизовані робочі місця; розподілена одно рівнева система; розподілена багаторівнева система; інтегрована багаторівнева система; інтегрована система управління підприємством.

Перша є суттю справи АРМ, які створюються на базі окремих ПК з відповідними ПО і програмним забезпеченням.

Другі є однорівневими системами, в яких робочі станції або ПК об'єднуються в локальну мережу. Їх функціональні можливості в цьому випадку будуть рівноправними в плані виконання проектно-конструкторських функцій.

Розподілена багаторівнева система характеризується об'єднаними в локальну мережу однієї або декількох робочих станцій і декілька ПК. Причому, на робочих станціях здійснюється розробка вузлів макро- і метарівня на основі вже розроблених об'єктів мікро рівня, які розробляються на ПК з використанням відповідного ПО.

Інтегрована багаторівнева система призначається для проектування складних виробів і є сукупністю декількох розподілених багаторівневих систем.

Інтегрована система управління підприємством включає САПР як окрему структурну складову в тому випадку, якщо підприємство займається розробкою нової техніки без залучення проектних організацій.

По програмних характеристиках класифікацію САПР можна здійснювати по наступних напрямках: по спеціалізації програмних засобів; за способом організації внутрішньої структури САПР; по можливості функціонального розширення системи користувачем; по можливості обміну інформацією; за способом створення змінних прототипів; по методах моделювання функцій створюваних виробів.

Перші додатково можна підрозділити на вузько спеціалізоване ПО (призначено для проглядання файлів, перекладу їх з одного формату в інший), спеціалізовані системи (призначені для підготовки виробництва на верстатах з ЧПУ, розробки програм для контрольно-вимірювальних систем), універсальні САПР (призначені для проектування виробів найширшого профілю), комплексні системи (призначені для проектування високо складних виробів).

За способом організації внутрішньої структури САПР розділяють нерозширювані системи і масштабовані модульні системи. Перші використовують стандартний набір взаємозв'язаних модулів, які дозволяють реалізовувати основні функції САПР. Другі можуть функціонально змінюватися шляхом доповнення можливостей до базового ядра (доповнення обумовлені специфічними особливостями проектованого виробу) або шляхом горизонтального розширення САПР за рахунок організації доступу до зовнішніх застосувань, обміну даними із зовнішніми системами. В даному випадку інтегруючим ядром є диспетчер призначеного для користувача середовища.

По можливості функціонального розширення системи користувачем розділяють: закриті системи; системи з системою інтерфейсу, що налаштовується, користувачем; системи з пакетною обробкою команд; системи з вбудованою макро мовою і бібліотекою функцій; системи з можливістю підключення зовнішніх модулів; інструменти розробника САПР.

Перші і другі системи не вимагають особливих коментарів. Системи з пакетною обробкою команд надають можливість виконання послідовності команд САПР, сформованих в текстовому пакетному файлі, створеному зовнішньою програмою. Системи з вбудованою макро мовою і бібліотекою функцій дозволяють здійснювати запис макрокоманд і створювати нові функції користувача. Інструменти розробника САПР дають проектувальникові можливість, використовуючи набір стандартних бібліотек функцій, створити свої власні застосування САПР.

По можливості обміну інформацією розрізняють замкнуті системи (не дозволяють обмінюватися інформацією з іншими системами), системи з текстовими файлами обміну інформацією (зберігають і прочитують інформацію у вигляді масивів цифр, розділених пропусками і комами), системи із стандартними засобами обміну інформацією (дозволяють зберігати і прочитувати інформацію про об'єкт проектування в спеціальному текстовому або двійковому форматі, наприклад .DXF.)

За способом створення змінених прототипів класифікують на САПР, де використовується: незмінні готові блоки (готові об'єкти вставляються з БД в проєктований виріб); елементи, що програмно формуються в зовнішніх модулях (створюються спеціальними програмами); елементи, що параметрично задаються (графічні об'єкти, розміри яких зв'язані між собою функціонально і зміну одну з яких приводить до зміни останніх); адаптивний змінні елементи (коректування об'єктів здійснюється

проектантом); комбіновані методи (об'єднують параметричну і адаптивну технології).

По методах моделювання функцій створюваних виробів класифікуються: без спеціальних методів (параметри проєктованого об'єкту визначаються традиційними методами); перевірочні розрахунки з використанням методу кінцевих елементів (використовуються дані про інформаційну модель об'єкту і результати обчислювального експерименту); спеціалізовані підсистеми моделювання (спеціалізовані пакети прикладних програм).

По технічних характеристиках САПР класифікують: по використовуваних засобах обчислювальної техніки (ПК, робочі станції, міні ЕОМ, мейнфрейми); за способом об'єднання технічних засобів (автономні робочі станції, багато термінальні ЕОМ, локальна мережа, локальна мережа з виділеним сервером, гетерогенна мережа з складною структурою); по використовуваних технічних засобах і периферійній освіті (САПР мінімальної конфігурації – монітор, клавіатура, миша, принтер, графічний пристрій, пристрій для зберігання інформації; технічно розвинені САПР – один або декілька моніторів, клавіатура, миша, дигитайзер (цифровий планшет), сканер, принтер, графічні пристрої (плоттери), пристрої зберігання інформації).

По ергономічних характеристиках можна класифікувати САПР: за способом організації діалогу системи з користувачем (за допомогою командного рядка, ієрархічних меню, об'єктно-орієнтованого інтерфейсу); по зручності діалогу системи з користувачем (простий, складний); по підтримці тривимірного моделювання (двовимірні системи, тривимірні каркасні, тривимірні з видаленням прихованих ліній, тривимірні зі світлотіньовим розфарбовуванням, тривимірні з фотореалістичним відображенням).

2. ОБ'ЄКТИ ПРОЕКТУВАННЯ В САПР ЛЕГКОЇ І ТЕКСТИЛЬНОЇ ПРОМИСЛОВОСТІ

Як об'єкти проектування вибираються з одного боку устаткування, а з іншого боку технологічні процеси легкої і текстильної промисловості. Дані об'єкти тісно взаємозв'язані між собою. Це виявляється в тому, що відповідне технологічне устаткування завжди розробляється або удосконалюється під конкретний технологічний процес. Кожна зміна технологічного ланцюжка приводить до обов'язкової модернізації устаткування, яке використовується на даному переході. Дані факти дозволяють оцінювати об'єкти проектування як єдину систему, у якій присутні тісні зв'язки між об'єктами. Причому, дані зв'язки мають двосторонню спрямованість. Наприклад, недосконалість робочих органів технологічного устаткування може привести до серйозних порушень при виконанні певної технологічної операції.

Науково-технічний прогрес в легкій і текстильній промисловості має на увазі під собою вдосконалення всіх аспектів виробництва на основі використання новітніх досягнень науки і техніки. Це виявляється в застосуванні сучасних систем автоматизованого проектування (САПР) устаткування і технологічних процесів легкої і текстильної промисловості. У свою чергу, одним з основних компонентів будь-якого вигляду САПР є математичне забезпечення, яке нерозривно пов'язане з програмним забезпеченням. Зупинимося на цьому детальніше. Будь-який об'єкт проектування може бути представлений у вигляді інформаційної моделі, яка може мати вигляд вербальний, математичний, графічний або табличний. З погляду автоматизації процесу проектування нас більш цікавлять математичні і табличні інформаційні моделі. Для отримання даних моделей використовуються цілі комплекси дисциплін (про це буде сказано нижче).

Для побудови інформаційних моделей об'єктів проектування в легкій і текстильній промисловості необхідно враховувати однорідність властивостей наочних областей і їх характер інженерної спрямованості. Виходячи з сказаного, можна стверджувати, що стосовно легкої і текстильної промисловості основними спостережуваними фізичними процесами і явищами будуть механічні і теплові. Також мають місце фізико-хімічні і хімічні процеси. Зважаючи на специфіку отримання хімічних волокон, нас цікавитимуть фізико-хімічні процеси, що відбуваються при отриманні полімерів. При проектуванні устаткування основне місце займатимуть механічні процеси і явища. Специфіка отримання готової продукції (тканини, трикотаж, готові вироби і ін.) потребує необхідності вивчення механічної взаємодії між робочими органами технологічного устаткування і різними системами натуральних і хімічних ниток, з іншими матеріалами (неткані полотна, шкіра, хутро і ін.). Для отримання кількісних описів механічних, теплових і інших процесів використовуються класичні дисципліни. Основними з них (зважаючи на специфіку питання) є: теоретична механіка; теорія механізмів і машин; опір матеріалів; деталі машин; механіка нитки; механічна технологія волокнистих матеріалів; матеріалознавство швейного і шкіряно-взуттєвого виробництва і ін.

Навіть такий короткий перелік дисциплін показує, що отримання інформаційних моделей об'єктів в легкій і текстильній промисловості зв'язане із значними труднощами. Їх подальша модернізація приводитиме до збільшення витрат часу на проектування, робитиме процес проектування складнішим і таким, що важко піддається формалізації.

Використання сучасних інформаційних технологій дозволяє значно спростити процес проектування, зробити його мобільнішим до вимог, що постійно змінюються, до якості продукції, що виготовляється. Створення

сучасних САПР дозволить оперативно здійснювати модернізацію існуючого устаткування і технологічних процесів або розробляти нові.

В зв'язку з цим як об'єкти проектування доречно говорити про деякий комплекс, куди входить безпосередньо об'єкт (устаткування або технологічний процес) і розроблена підсистема САПР для його конструкторської реалізації. Такий підхід створить передумови для реалізації єдиною САПР устаткування і технологічних процесів легкої і текстильної промисловості.

2.1. СТРУКТУРНО-ЛОГИЧЕСЬКАЯ СХЕМА САПР УСТАТКУВАННЯ І ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ЛЕГКОЇ І ТЕКСТИЛЬНОЇ ПРОМИСЛОВОСТІ

Єдина САПР устаткування і технологічних процесів легкої і текстильної промисловості повинна включати дві підсистеми (ПС) для устаткування ПС САПР Устаткування і ПС САПР Технологічний процес (ТП) (див. рис.2.1). Різний колір тек указує на різне поєднання фізичних процесів і явищ, які враховуються при проектуванні. Так для устаткування основними будуть механічні і теплові процеси. Як впливає з рис.2.1 між двома підсистемами існує зв'язок двосторонньої спрямованості.

Необхідно відзначити, що дана САПР за ознакою «Складність об'єкту проектування» відноситиметься до САПР дуже складних об'єктів з числом складових частин від 104 до 106 [6].

По рівню автоматизації проектування вона відноситиметься до середньо автоматизованих САПР (від 25 до 50%). За ознакою «Комплексність автоматизації проектування» вона відноситиметься до комплексних систем.

САПР обладнання та технологічних процесів легкої та текстильної промисловості

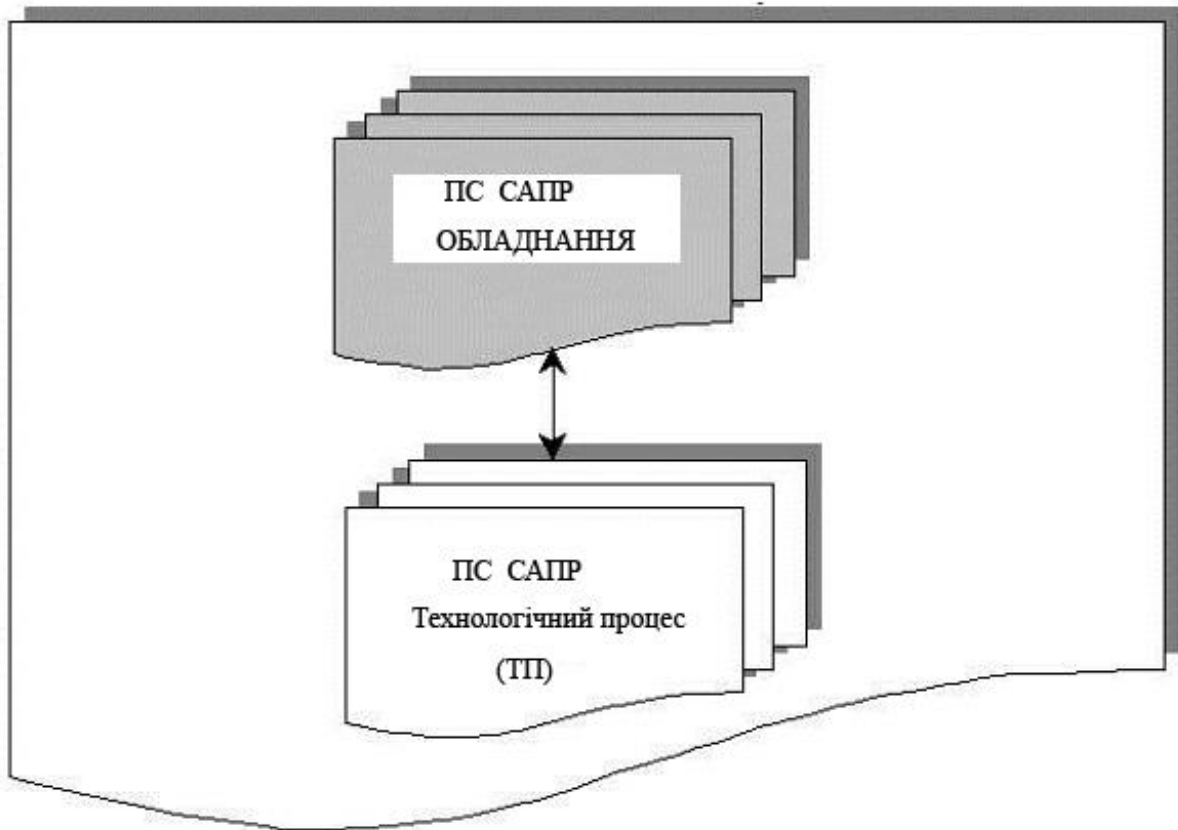


Рис.2.1

ПС САПР Устаткування (рис.2.2) включає 17 підсистем мета рівня першого класу (МУ-1). Перша підсистема охоплює сегмент машинного парку, який пов'язаний з машинами прядильного виробництва. Друга (ПС САПР МУ-1) охоплює сегмент мотальних машин. Третя ПС САПР охоплює сегмент уточно-мотальних автоматів. Четверта ПС САПР мета рівня 1-го класу охоплює сегмент устаткування для формування полотна для нетканих полотен.

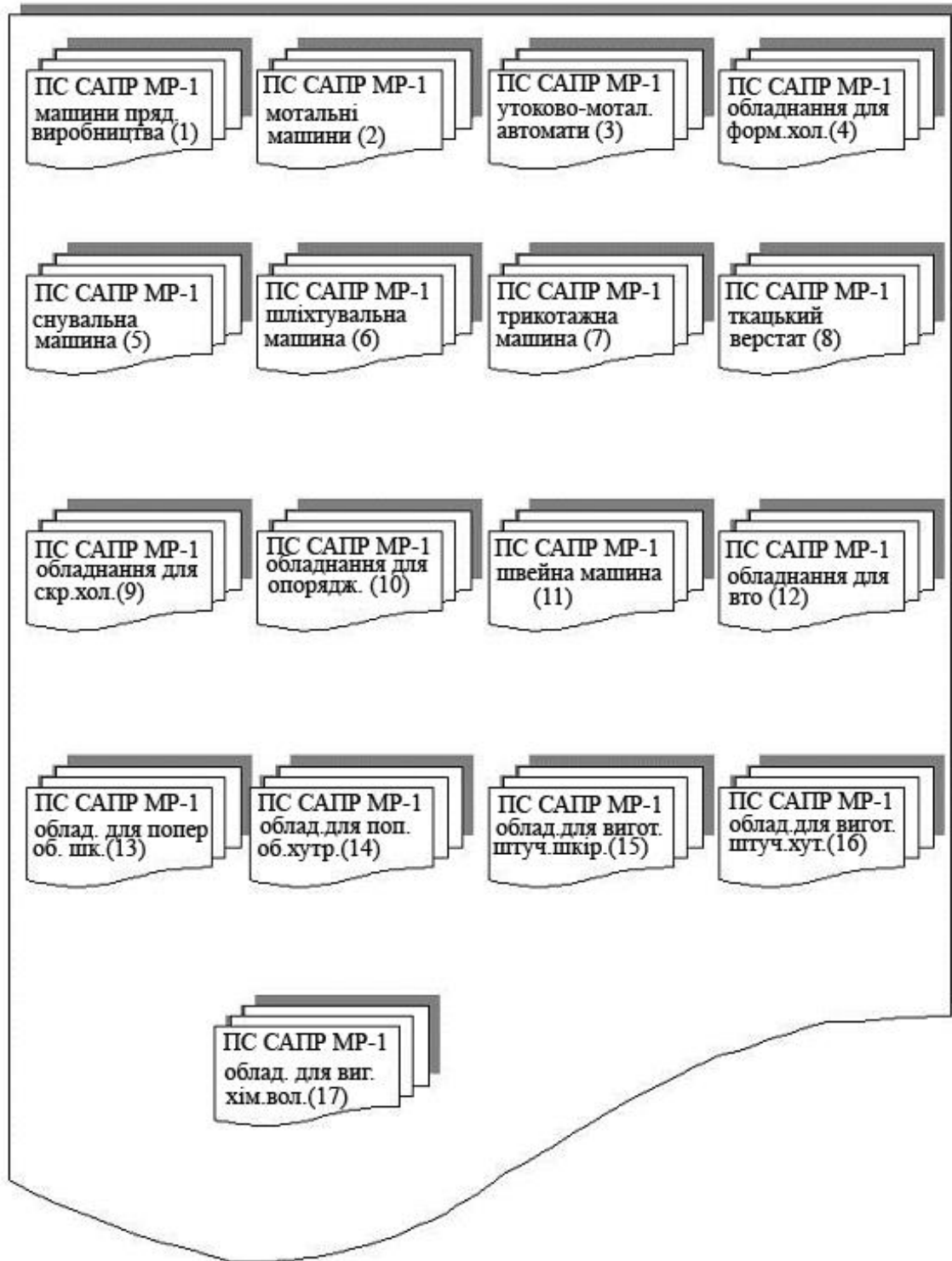


Рис.2.2

П'ята ПС САПР охоплює сегмент снувальних машин. Шоста ПС САПР охоплює сегмент шліхтувальних машин.

Дані підсистеми направлені на проектування устаткування, яке забезпечує підготовче виробництво. Сюди слід також віднести і ПС САПР (17), яка охоплює сегмент машинного парку, пов'язаного з устаткуванням для отримання хімічних волокон.

Наступний ряд підсистем направлений на проектування устаткування для отримання тканини, трикотажу і нетканого полотна. ПС САПР (7) охоплює сегмент трикотажних машин. Восьма ПС САПР МУ-1 охоплює сегмент ткацьких верстатів. Дев'ята ПС САПР охоплює сегмент устаткування для того, що скріплює полотна (систем ниток, тканини, їх комбінацій) при отриманні нетканих полотен. Приведені десять підсистем охоплюють весь процес отримання напівфабрикату від сировини до його виходу з технологічної машини. Докладніше на цьому зупинимося при розгляді ПС САПР Технологічний процес. Десята ПС САПР мета рівня 1-го класу охоплює устаткування для розбраковування і обробки тканин, трикотажних і нетканих полотен.

Дві ПС САПР МУ-1 (рис.2.2) тринадцята і чотирнадцята охоплюють сегмент устаткування для попередньої обробки шкіри і шкірок. П'ятнадцята і шістнадцята ПС САПР МУ-1 відносяться до устаткування для виробництва штучної шкіри і хутра.

З отриманих напівфабрикатів виготовляються вироби (швейні, трикотажні, взуттєві, шкіргалантерейні і ін.). На цьому етапі використовуються машини розкроїв, швейних, устаткування для волого-теплової обробки (ВТО). Для їх проектування використовуються ПС САПР «Швейна машина» (11) і «Устаткування для ВТО» (12).

Другий крупною ПС САПР загальною САПР устаткування і технологічних процесів легкої і текстильної промисловості є підсистема «Технологічний процес (ТП)» (рис.2.3). Необхідно відзначити, що специфіка побудови даної підсистеми має на увазі використання висхідної системи проектування. Початковими об'єктами для неї є натуральна

сировина (волокна), сировина для отримання штучних і синтетичних волокон, шкіра, хутро.

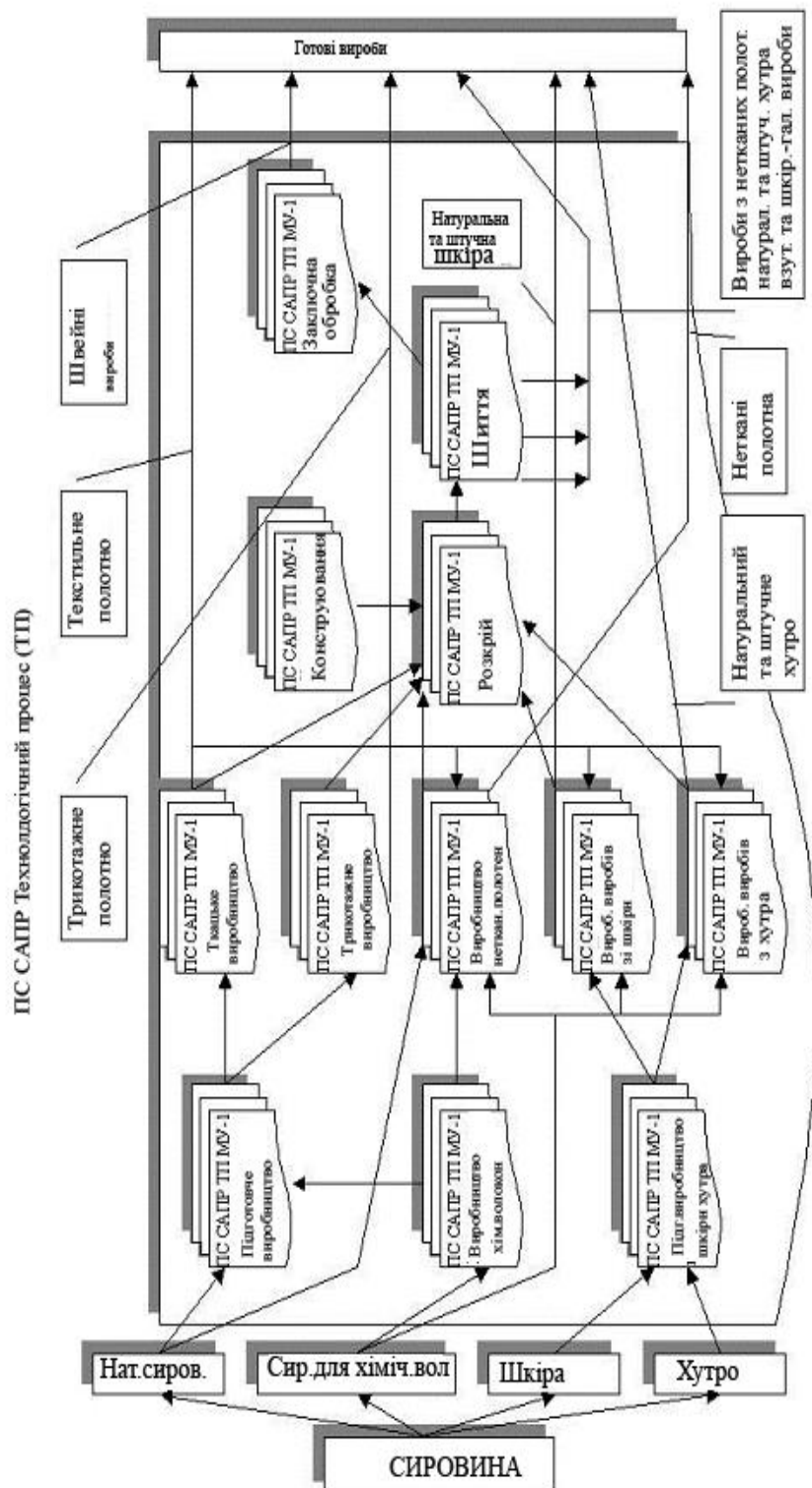


Рис.2.3

На виході отримують готові вироби: текстильні і трикотажні полотна; швейні вироби; натуральні і штучну шкіру і хутро; неткані полотна;

вироби з нетканих полотен; вироби з натурального і штучного хутра; взуттєві і шкіргалантерейні вироби з натуральної і штучної шкіри.

Першими трьома ПС САПР МУ-1 є «Підготовче виробництво», «Виробництво хімічних волокон» (22), «Підготовче виробництво шкіри і хутра». Наступні п'ять ПС САПР МУ-1 складають підсистеми «Ткацьке виробництво», «Трикотажне виробництво», «Виробництво нетканих полотен», «Виробництво виробів з шкіри», «Виробництво виробів з хутра». Для ПС САПР МУ-1 «Ткацьке виробництво» характерне те, що отримувані напівфабрикати можуть безпосередньо використовуватися як готові вироби і можуть служити до якості композитів для підсистем «Виробництво нетканих полотен», «Виробництво виробів з шкіри», «Виробництво виробів з хутра». Вхідними компонентами для даної ПС САПР будуть вихідні компоненти ПС САПР ТП «Підготовче виробництво».

Підсистема САПР «Трикотажне виробництво» також дозволяє отримувати напівфабрикати, які безпосередньо можуть використовуватися як готові вироби і можуть служити як композити для вказаних вище підсистем. Як вхідні компоненти можуть використовуватися готові вироби підсистеми САПР «Підготовче виробництво» - бобіни, навої.

ПС САПР МУ-1 «Виробництво нетканих полотен» об'єднує в собі два етапи: підготовка складників нетканого полотна (волокнистого полотна, тканини, трикотажного полотна, систем ниток, їх різних комбінацій); скріплюють складників. Вхідними даними для даної ПС будуть вихідні компоненти ПС «Ткацьке виробництво», «Трикотажне виробництво», «Виробництво хімічних волокон (22)». Крім того, при розробці даної ПС САПР використовуються початкові (вхідні) компоненти для всієї ПС САПР «Технологічний процес (ТП)»: натуральна сировина; сировина для штучних і синтетичних волокон. Вихідними компонентами для даної ПС буде неткане полотно, яке може використовуватися безпосередньо як

готовий виріб, так і поступати як вхідний компонент в ПС САПР «Розкрій».

ПС САПР ТП мета рівня 1-го класу «Виробництво виробів з шкіри» і «Виробництво виробів з хутра» є основними при формуванні готових виробів. До них можна віднести натуральну і штучну шкіру і натуральне і штучне хутро. Крім того, можливий і другий шлях, коли напівфабрикати використовуються при виготовленні: виробів з натурального і штучного хутра; взуттєвих виробів; шкіргалантерейних виробів; виробів з натуральної і штучної шкіри.

ПС САПР ТП МУ-1 «Підготовче виробництво шкіри і хутра» включає ПС САПР ТП мета рівня 2-го класу, які направлені на забезпечення виконання наступних технологічних процесів: консервації; розконсервації; міздріння (для хутра необхідне підшкірне міздріння); дублення; жирування; сушки; обробки. Вхідними компонентами для даної ПС будуть шкіра і хутро.

Як наголошувалося вище, до готових виробів (рис.2.3) відносяться не тільки напівфабрикати (трикотажні і ткані полотна, натуральна і штучна шкіра і хутро, неткані полотна), але і конкретні вироби. Технологічний процес їх виготовлення (з вказаних напівфабрикатів) забезпечується ПС САПР ТП МУ-1 «Конструювання (18)» і «Розкрій (19)». Тут навіть доречніше говорити про проектування технологічного процесу їх виготовлення з використанням вказаних підсистем САПР. ПС САПР «Конструювання (18)» дозволяє розробляти нові моделі виробів з використанням сучасних графічних пакетів в інтерактивному режимі. Дана ПС дозволяє також отримувати готові викрійки. ПС САПР «Розкрій (19)» направлена на вдосконалення технологічного процесу розкрою з погляду мінімізації відходів матеріалу.

Наступна ПС САПР ТП МУ-1 «Шиття (20)» направлена на проектування ТП сточування елементів криючи для отримання готового виробу.

Для виробів з тканини і трикотажу передбачається ще одна ПС САПР ТП МУ-1 «Завершальна обробка (21)». Дана ПС призначена для проектування ТП волого-теплової обробки виробів.

Зупинимося докладніше на утриманні приведених вище ПС САПР мета рівня 1-го класу. На рис.2.4 показана ПС САПР МУ-1 «Підготовче виробництво».

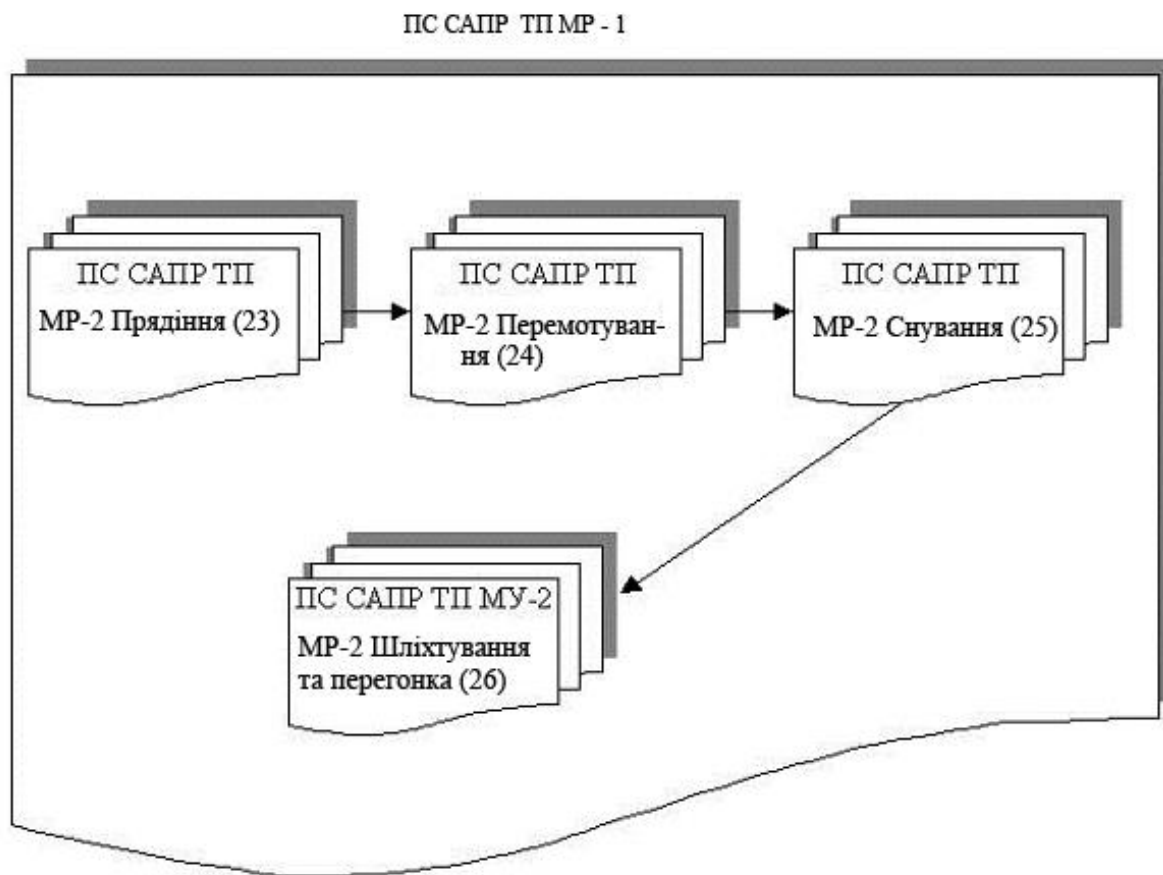


Рис.2.4

Вона включає чотири ПС САПР мета рівня 2-го класу. До них відносяться ПС САПР ТП МУ-2 «Прядіння (23)», ПС САПР ТП МУ-2 «Перемотування (24)», ПС САПР ТП МУ-2 «Снування (25)», ПС САПР ТП МУ-2 «Шліхтування і перегонка (26)». Перша направлена на проектування

ТП прядіння, коли з початкової сировини отримують готові нитки. Друга направлена на проектування ТП перемотування ниток, коли відбувається очищення засміченої пряжі низької якості з великим числом пороків і її перемотування на мотальні пакування, які зручно використовувати для подальшого процесу снування. При перемотуванні ниток відбувається збільшення їх довжини. Для цього нитки з декількох пакувань перемотуються на одну бобіну. Кінці ниток при цьому зв'язуються.

Третя ПС направлена на проектування ТП снування, коли на одне пакування (навой) намотується певне число ниток заздалегідь певної довжини.

Четверта ПС САПР ТП МУ-2 «Шліхтування і перегонка» направлена на проектування технологічних процесів шліхтування і перегонки. Операція шліхтування (покриття ниток склеювальною речовиною) направлена на зменшення сил тертя між нитками і робочими органами ткацького верстата. Процес шліхтування дозволяє підвищити міцність і стійкість ниток до стираючої дії. Кручена і шерстяна пряжа апаратного прядіння, синтетичні нитки піддаються перегонці, коли нитки з декількох бобін з'єднуються на загальному навої.

Наступна ПС САПР ТП МУ-1 «Ткацьке (трикотажне) виробництво» включає три ПС САПР ТП мета рівня 2-го класу (див. рис.2.5). Вона включає наступні ПС: «Ткацтво»; «В'язання»; «Розбракування і обробка». ПС САПР МУ-2 «Ткацтво» направлена на проектування технологічного процесу ткацтва. При цьому враховується вид переплетення, фізико - механічні властивості ниток, щільність тканини по основі і витчу, особливості пристрою конкретного технологічного устаткування.

ПС САПР ТП МУ-2 «В'язання» направлена на проектування технологічного процесу того, що в'яже ниток на трикотажних машинах.

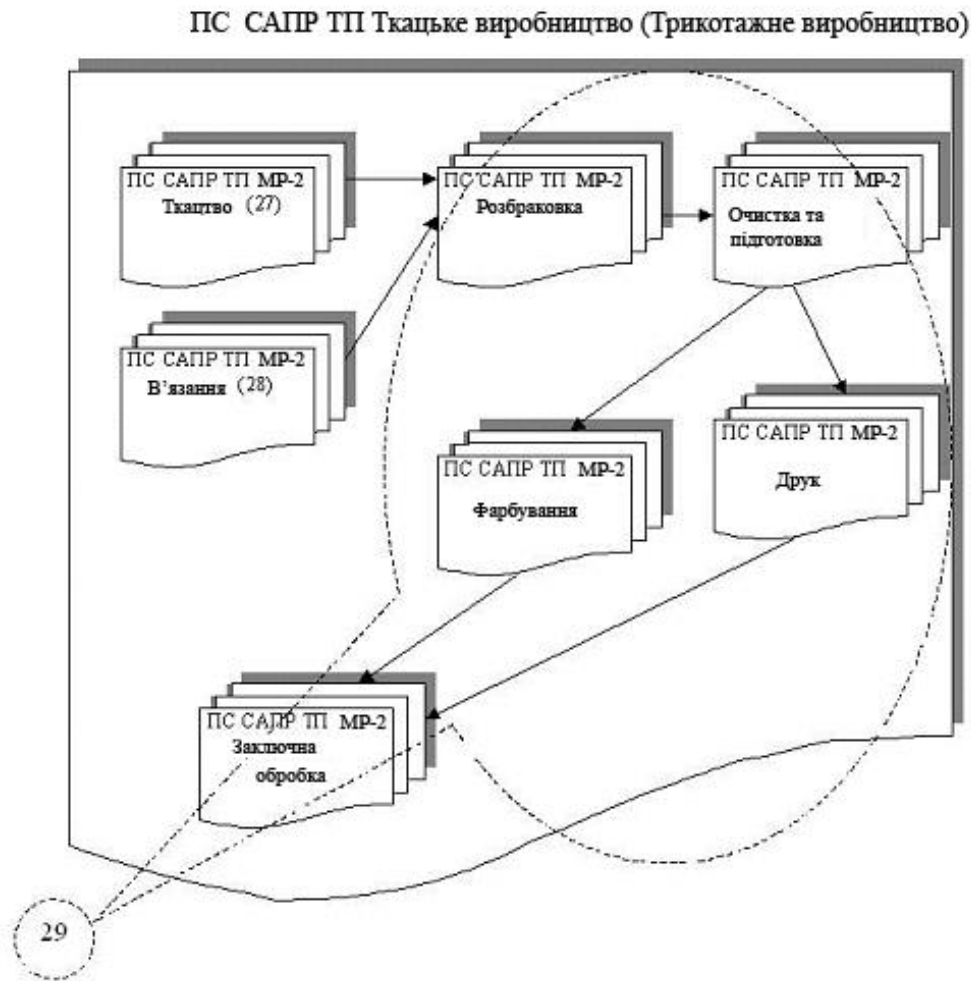


Рис.2.5

Тут враховується вигляд і клас трикотажної машини, фізико - механічні властивості ниток, що переробляються, кількість петельних рядків і стовпчиків на одиницю площі, модуль петлі і ін.

Наступна ПС САПР ТП МУ-2 (29) включає п'ять підсистем САПР МУ-2: «Розбравування»; «Очищення і підготовка»; «Фарбування»; «Друкування»; «Завершальна обробка».

На стадії проектування ПС САПР ТП МУ-2 «Очищення і підготовка» розглядаються технологічні процеси обпалення, розшліхтування, відварювання, білення, мерсеризації, ворсування, промивання, карбонізує, заварювання, валяння, стабілізації. Їх поєднання і послідовність виконання

вибираються залежно від складу і будови ниток, з яких виготовляються тканини і трикотажні полотна.

ПС САПР ТП МУ-2 «Завершальна обробка» призначена для проектування технологічних процесів апретування, ширення, глядіння, проти усадкової обробки, не мнучкої обробки, стрижки, пресування, декатирування. Склад і послідовність виконання цих процесів знову ж таки диктуються хімічною природою використовуваних матеріалів і будовою готового продукту.

На рис.2.6 представлена ПС САПР ТП МУ-1 «Підготовче виробництво шкіри і хутра». Вона включає дві ПС САПР ТП мета рівня 2-го класу: «Попередня обробка шкіри»; «Попередня обробка шкірок». При розробці даних ПС САПР розглядаються технологічні процеси консервації, розконсервації, міздріння, дублення (фарбування), жирування, сушки, обробки (фарбування). Для штучної шкіри передбачається своя послідовність технологічних операцій, яка обумовлена особливостями отримання початкового матеріалу.

На рис.2.7 представлена структурно - логічна схема ПС САПР ТП МУ-1 «Виробництво виробів з штучної шкіри». Вона складається з двох підсистем САПР мета рівня 2-го класу: «Штучна шкіра»; «Розбраковування і обробка». Аналогічна будова має ПС САПР ТП МУ-1 «Виробництво виробів з штучного хутра» (рис.2.8). У неї входять дві ПС САПР ТП МУ-2: «Штучне хутро»; «Розбраковування і обробка».

ПС САПР ТП МР - 2 Підготовче виробництво шкіри та хутра

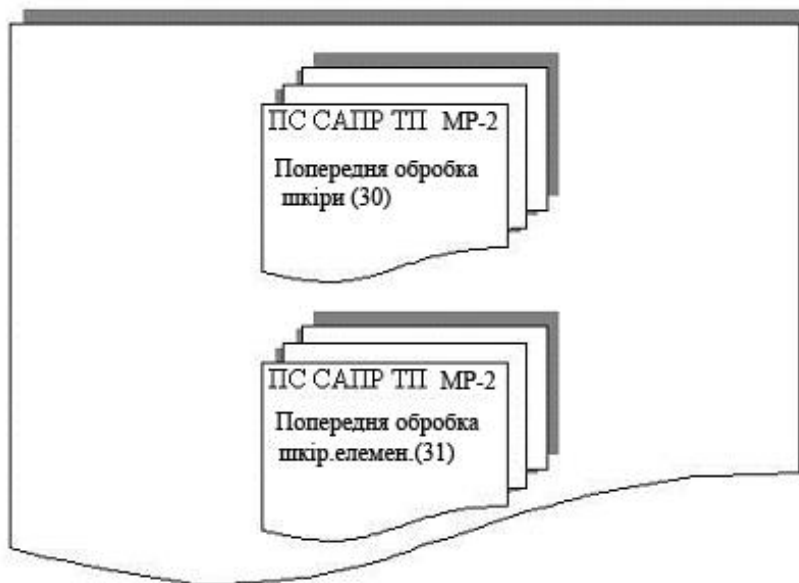


Рис.2.6

ПС САПР МР - 2 Виробництво виробів зі шкіри

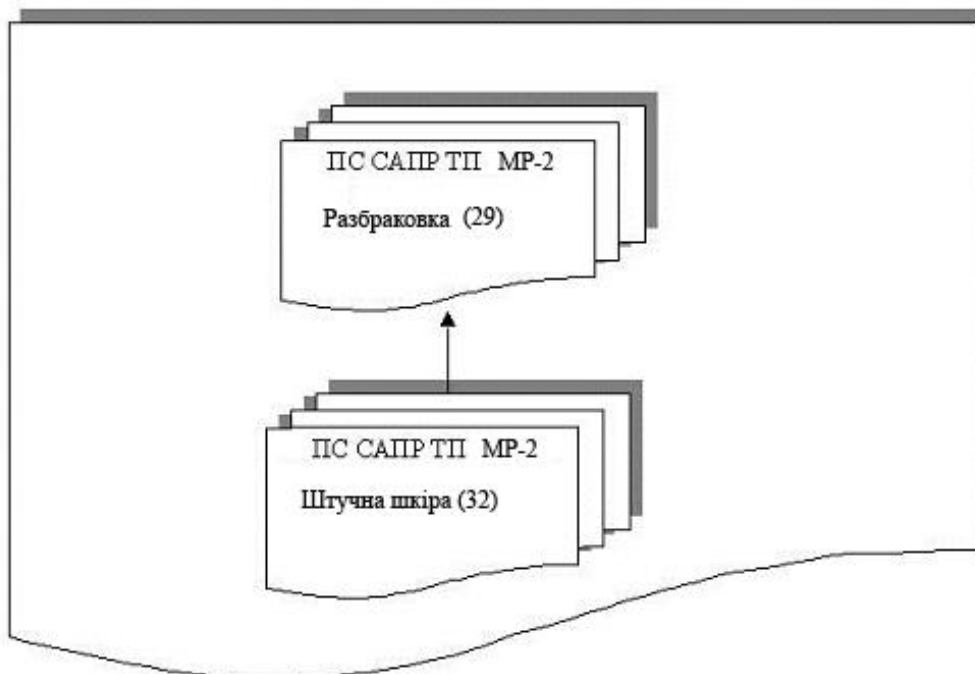


Рис.2.7

ПС САПР МР - 2 Виробництво виробів з хутра

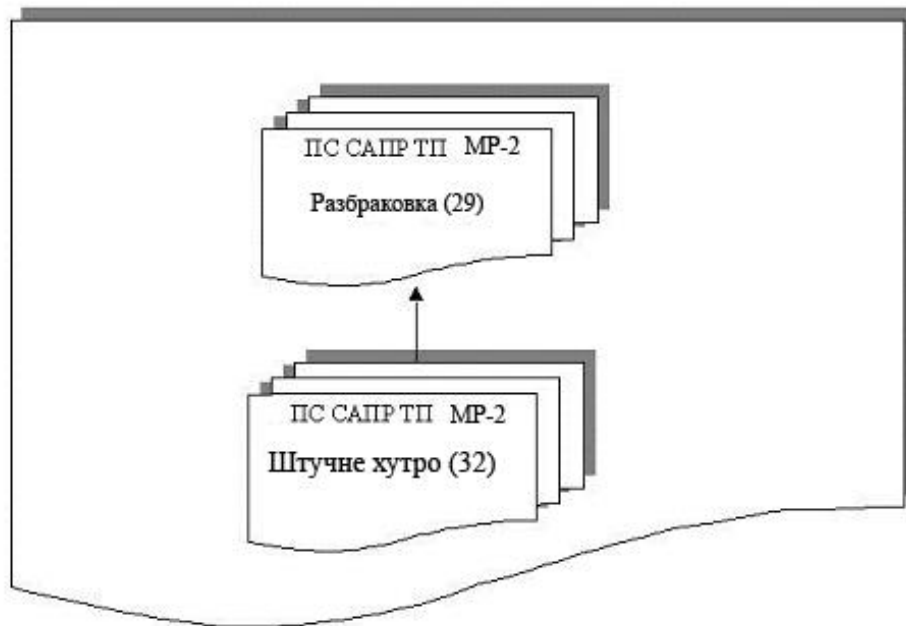


Рис.2.8

На закінчення розглянемо ПС САПР ТП МУ-1 «Виробництво нетканих полотен» (рис.2.9). Дана ПС САПР містить три ПС САПР ТП мета рівня 2-го класу: «Волокнисте полотно»; «Неткане полотно»; «Розбраковування і обробка». Перша ПС дозволяє проектувати технологічний процес отримання волокнистого полотна залежно від хімічного складу початкових компонентів. Друга ПС дозволяє проектувати технологічний процес отримання нетканого полотна. Остання ПС направлена на проектування технологічних процесів розбраковування і обробки.

Таким чином, розглянуті вище дві ПС САПР «Устаткування» і «Технологічний процес» дозволили побудувати загальну структурно - логическую схему САПР устаткування і технологічних процесів легкої і текстильної промисловості, яка приведена на рис.2.10. Дана схема дозволяє прослідкувати порядок послідовних переходів при проектуванні

устаткування і технологічних процесів для отримання готових виробів з початкової сировини.

Необхідно відзначити особливості взаємин між різними ПС САПР, які можуть носити як взаємозв'язаний характер, так і одно напрямлений. У першому випадку вихідні параметри ПС будуть вхідними для пов'язаних з ними ПС і в той же час будуть залежати від вихідних параметрів останніх. Причому, пріоритетне значення матимуть вихідні параметри ПС, направлених на проектування технологічних процесів.

У другому випадку здійснюється послідовний перехід від однієї ПС до наступної, коли вихідні параметри попередньої ПС однозначно будуть вхідними для подальшої.

ПС САПР МР - 1 Виробництво нетканних полотен

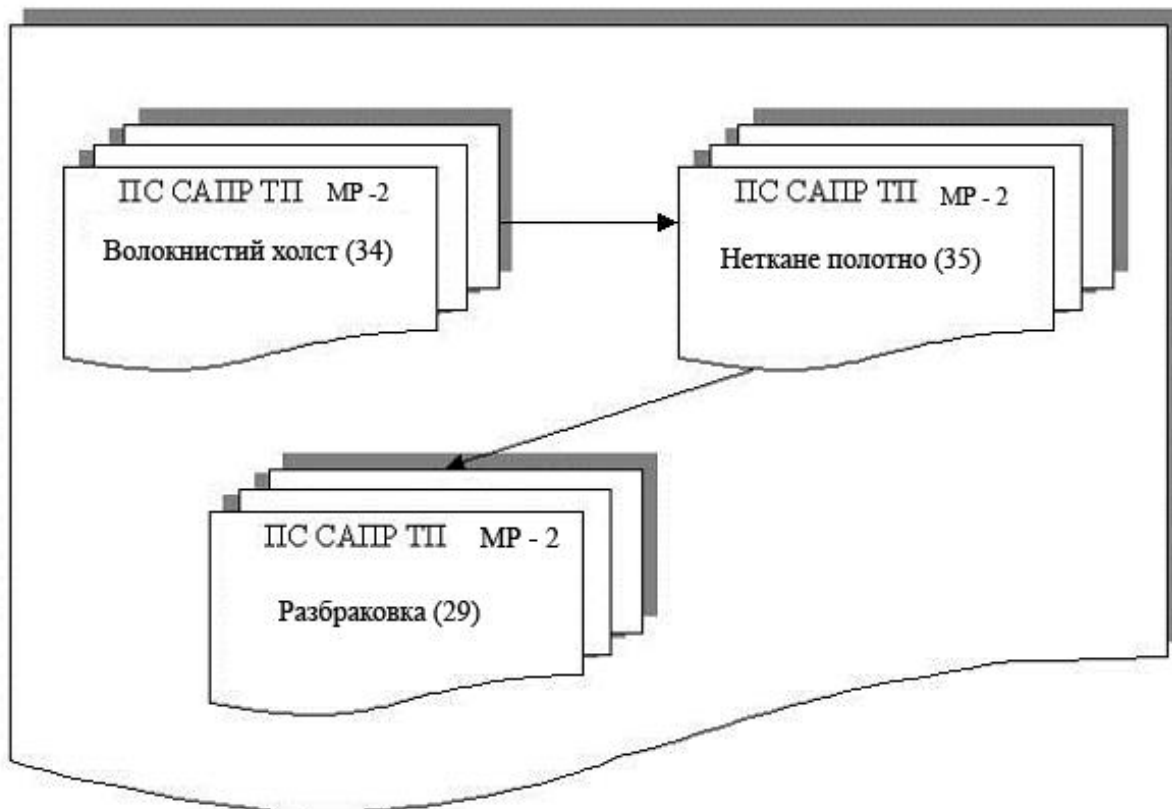


Рис.2.9

Структурно-логічна схема САПР обладнання та технологічних процесів легкої та текстильної промисловості

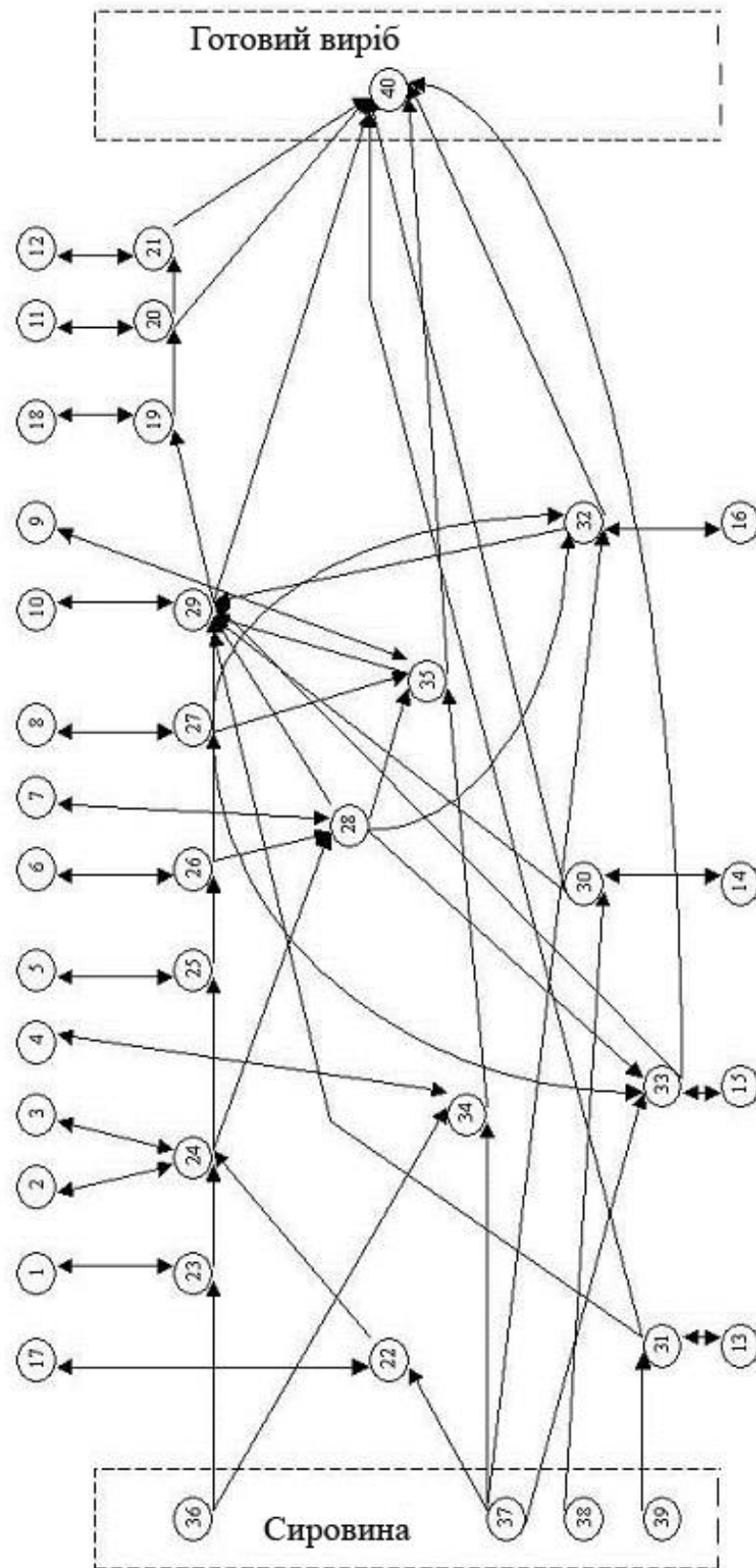


Рис.2.10

2.2. СТРУКТУРНО-ЛОГИЧЕСЬКАЯ СХЕМА ПС САПР МУ-1 «ТРИКОТАЖНА МАШИНА» І «ТКАЦЬКИЙ ВЕРСТАТ»

Підсистеми САПР мета рівня 1-го класу «Трикотажна машина» і «Ткацький верстат» є компонентами, що становлять, загальною САПР «Устаткування» (см.рис.2.1). Дані підсистеми призначені для проектування технологічного устаткування, яке використовується при виготовленні тканин і трикотажних полотен.

На рис.2.11 представлена структурно - логічна схема ПС САПР МУ-1 «Трикотажна машина», яка включає п'ять ПС САПР мета рівня 2-го класу: «Основов'язальна машина»; «Плосков'язальна машина»; «Котонна машина»; «Панчішний автомат»; «Кругло трикотажна машина». Дані п'ять ПС САПР охоплюють практично весь парк трикотажного устаткування, яке використовується на сучасних підприємствах.

Аналогічну структурно - логічну схему можна скласти і для ткацьких верстатів (рис.2.12). ПС САПР МУ-1 «Ткацький верстат» складається з двох ПС САПР мета рівня 2-го класу: «Човниковий ткацький верстат»; «Без човниковий ткацький верстат». Остання ПС включає чотири підсистеми мета рівня 3-го класу: «Мікро човниковий ткацький верстат»; «Рапірний»; «Пневматичний і гідравлічний»; «Пневмо рапірний». Така композиція ПС САПР будувалася на обліку принципу прокладення качка в зів на ткацькому верстаті. Повніша ПС САПР МУ-1 може бути отримана при обліку поєднання технологічних операцій. Тоді побудову вказаної ПС необхідно будувати на основі принципу утворення тканини, який дозволяє здійснити як циклічне, послідовне виконання операцій, так і поєднання технологічних операцій (ткацькі машини з декількома хвилевими зівами, кругло ткацька машина).

Зупинимося докладніше на структурно - логічній схемі ПС САПР МУ-2 «Човниковий ткацький верстат» (рис.2.13). Початковим компонентом тут буде ТЗ.

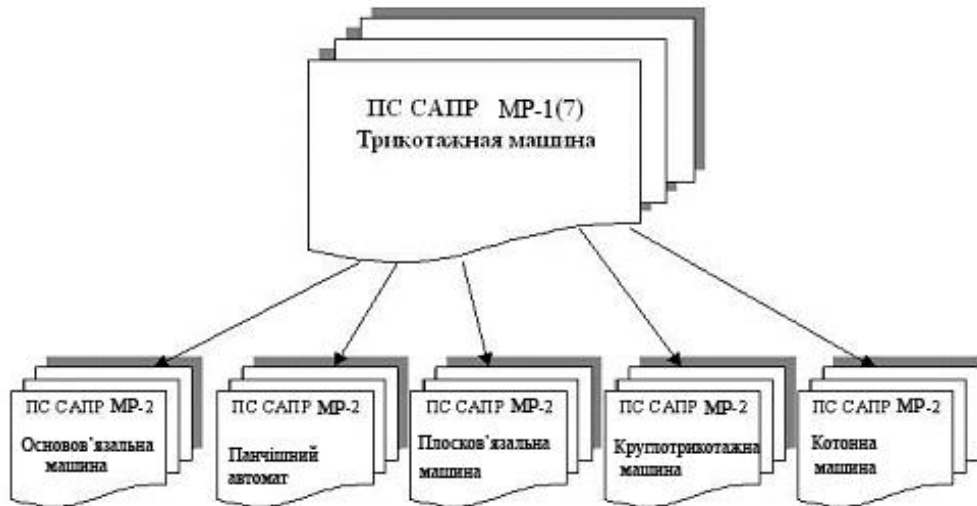


Рис.2.11

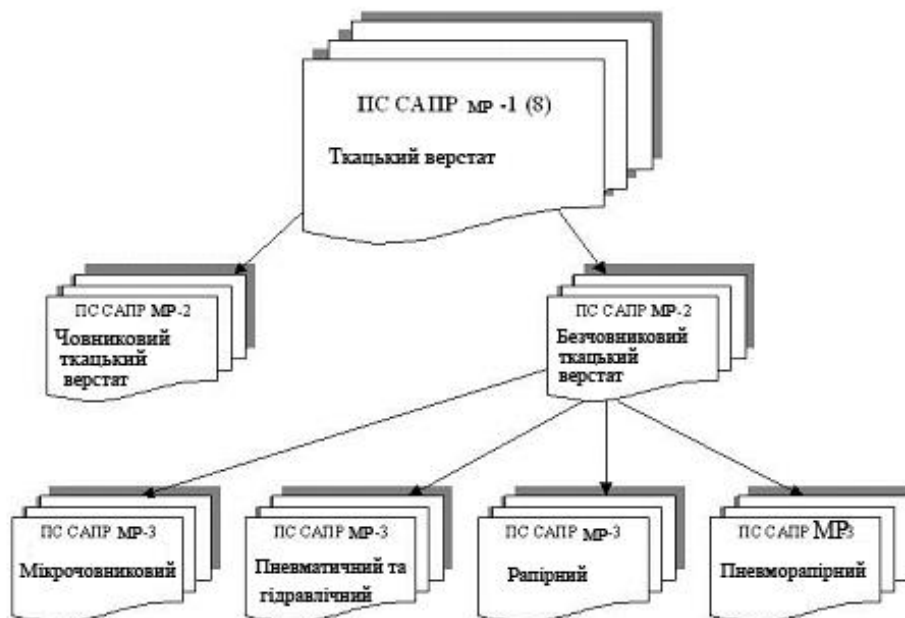


Рис.2.12

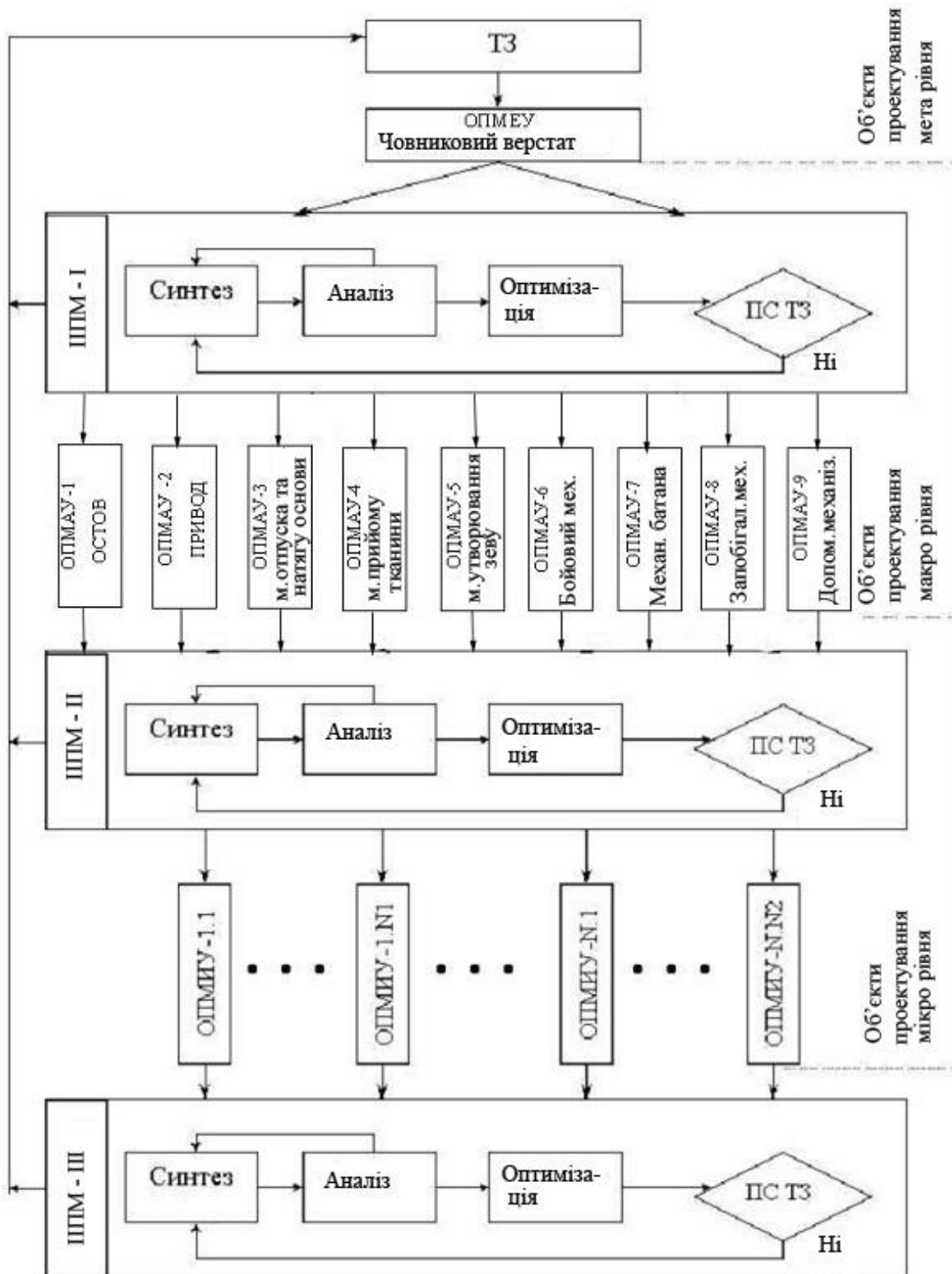


Рис.2.13

Об'єктом проектування мета рівня є човниковий ткацький верстат. Результатом реалізації ШПМ – I буде перехід до об'єктів проектування макро рівня. Для човникового ткацького верстата це будуть: ОПМАУ-1

«Остов»; ОПМАУ-2 «Привід»; ОПМАУ-3 «Механізм відпустки і натягнення основи»; ОПМАУ- 4 «Механізм прийому тканини»; ОПМАУ- 5 механізм «Зево утворювальний»; ОПМАУ- 6 «Бойовий механізм»; ОПМАУ- 7 «Механізм батана»; ОПМАУ- 8 «Запобіжних механізмів»; ОПМАУ- 9 «Допоміжних механізмів». До останніх слід віднести різних напрямників, ремізні рамки, грудницю, механізм скало, шпарутки, спрямовувачі ниток і ін.

При реалізації ІІІМ- і і ІІІМ- іі на стадії синтезу і аналізу, для ОПМАУ- 3 . ОПМАУ- 7, використовується спеціальне програмне забезпечення для кінематичного і динамічного дослідження плоских механізмів з урахуванням тертя в шарнірних парах (див. Додаток).

На наступному етапі, при реалізації ІІІМ- іі, відбувається перехід до об'єктів проектування мікро рівня (ОПМІУ). Вони є різними деталями, з яких складаються вказані вище механізми і пристрої. Математичне і програмне забезпечення, яке служить для цього основою, приведене в 4 розділі і в додатку. У п'ятому розділі будуть розглянуті окремі приклади, які дозволять прослідкувати послідовність виконання конкретних проектних процедур.

На завершальній стадії, при реалізації ІІІМ- ІІІ, здійснюється синтез спроектованих компонентів і підсистем в рамках остаточного проектного рішення по всій проектованій системі в цілому. Тут також можливе коректування початкових даних на проектування в ТЗ.

2.3. СТРУКТУРНО-ЛОГИЧЕСЬКАЯ СХЕМА ПС САПР ТП МУ-2 «ТКАЦТВО»

На рис.2.14 представлена структурно - логічна схема ПС САПР технологічного процесу мета рівня 2-го класу «Ткацтво». Дана підсистема призначена для проектування технологічного процесу ткацтва. В даному

випадку об'єктами проектування мікро рівня будуть ткацькі навої і уточні нитки. Тут необхідно відзначити, що вказані об'єкти і технологія їх виготовлення реалізуються на стадії виконання ПС САПР ТП МУ-1 «Підготовче виробництво». У нашому випадку вони будуть вхідними компонентами для ПС САПР ТП МУ-2 «Ткацтво».

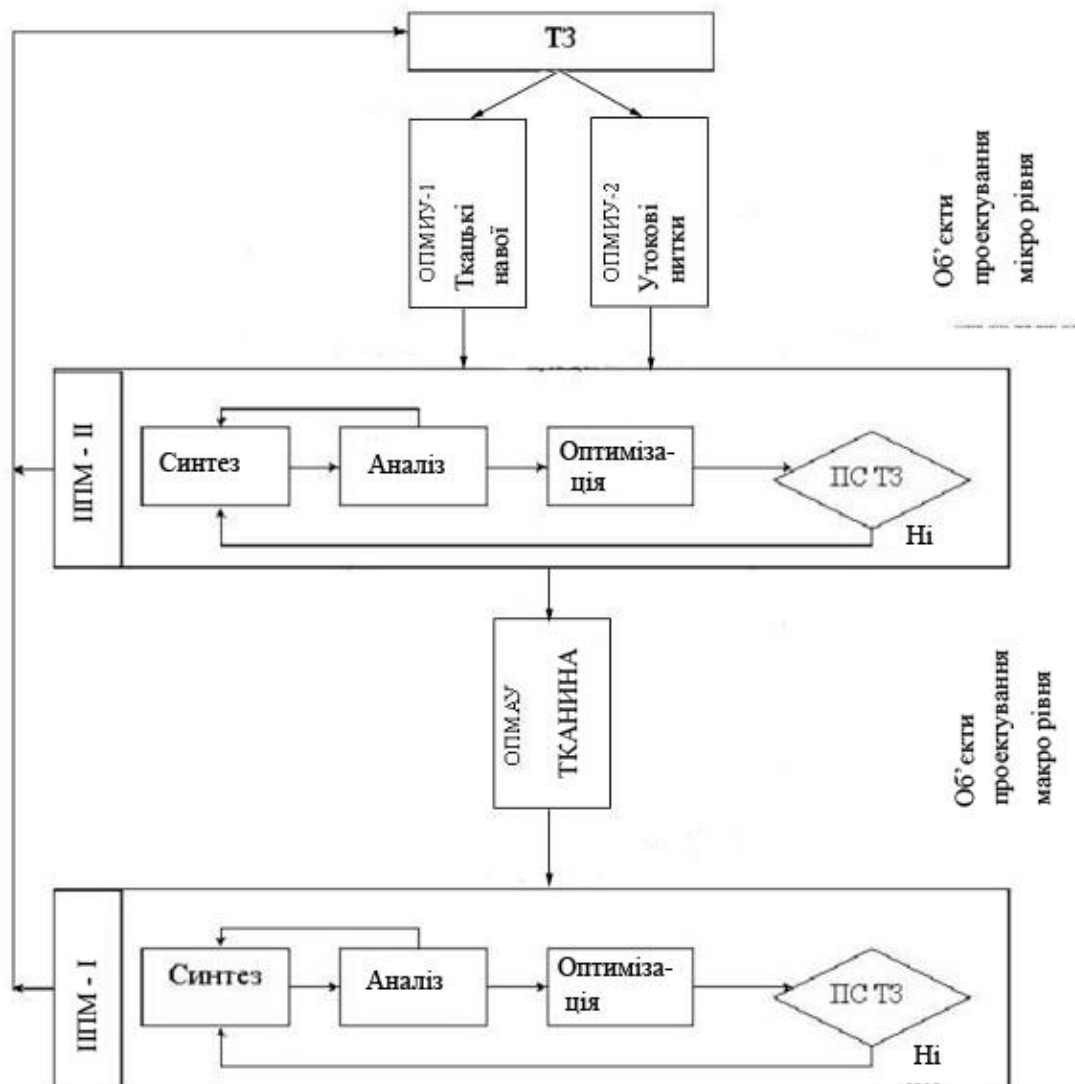


Рис.2.14

Реалізація ШПМ- II дозволяє перейти до об'єкту проектування макрорівня – тканини. Слід зазначити, що при реалізації проектних процедур синтезу, аналізу і оптимізації відбувається вибір оптимальних параметрів пружної системи заправки, величини заправного натягнення,

заступа і різно натягнутості зіву. Це дозволяє здійснювати формування тканини в найбільш прийнятних умовах, при мінімально необхідній силі прибою.

На стадії реалізації ШПМ- I здійснюється синтез підсистем в рамках остаточного проектного рішення.

3. МАТЕМАТИЧНІ МОДЕЛІ В САПР І ЇХ КЛАСИФІКАЦІЯ

Збільшення випуску продукції підприємствами легкої і текстильної промисловості, підвищення її якості безпосередньо залежить від машинобудівників, технологів і конструкторів, які створюють нове високопродуктивне устаткування, удосконалюють ті, що існують і розробляють нові технологічні процеси.

Вирішення вказаних завдань вимагає проведення великого об'єму обчислювальних операцій, тому розробка систем автоматизованого проектування (САПР) є найважливішим завданням і чинником, що багато в чому визначає технічний рівень, якість і ефективність сучасної техніки і технологічних процесів легкої і текстильної промисловості. Застосування математичних методів і ЕОМ істотно скорочує терміни розробки і впровадження нової техніки і технологічних процесів, дозволяє економити матеріали і отримувати оптимальну конструкцію машини (наприклад, як критерії оптимізації можуть бути вибрані мінімальні габаритні розміри, маса, приведений момент інерції, максимальна величина ККД) або оптимальний технологічний процес (тут як критерії оптимізації можуть бути вибрані наступні: мінімальний сумарний час протікання технологічного процесу, мінімальні енерговитрати, мінімізація відходів, збільшення випуску продукції і ін.). Крім того, автоматизація процесу проектування, із застосуванням ЕОМ і пакетів прикладних програм, дозволяє виключити грубі помилки при виконанні розрахунків, які можуть

приводити до порушення стійкості (статичною і динамічною) при роботі технологічного устаткування і при протіканні технологічних процесів.

3.1. МАТЕМАТИЧНІ МОДЕЛІ І ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ

Математичною моделлю називається наближений опис основних закономірностей якого-небудь класу об'єктів, систем або процесів за допомогою математичної символіки у вигляді рівнянь, нерівностей і ін. Поява ЕОМ привела до подальшого розвитку методу математичного моделювання для дослідження поведінки складних технічних систем. У загальному випадку процес отримання математичної моделі може бути розділений на чотири основні етапи [14].

На першому етапі, коли проектуваний об'єкт представляє складну інтегровану технічну систему, на основі принципу декомпозиції відбувається його розбиття на основні елементи, що становлять, і об'єкти. При цьому, наприклад при фізичному моделюванні, аналізуючи взаємозв'язок між окремими структурними елементами проектуваного об'єкту формулюються основні закономірності про зв'язки між ними в математичних термінах, які дозволяють здійснювати не тільки якісний, але і кількісний аналіз.

На другому етапі здійснюється рішення прямої або зворотної задачі проектування (тут доречно провести аналогію з першим і другим завданнями динаміки, завданнями аналізу і синтезу в теорії механізмів і машин). При рішенні прямої задачі по відомих зовнішніх і внутрішніх параметрах структурного елемента визначаються його вихідні параметри. Зворотне завдання проектування є складнішим. Тут по відомих зовнішніх і вихідних параметрах визначаються внутрішні параметри. Зворотне завдання відповідає в інженерній практиці проектувальному розрахунку, коли варіювання значень внутрішніх параметрів приводить до оптимізації

деякої цільової функції (наприклад завдання синтезу механізмів, коли як цільова функція можна використовувати мінімальні габаритні розміри механізму або його масу). Зворотне завдання є складним ще і тому, що не завжди відома інформація про сформульовані якісні уявлення про внутрішні параметри проєктованого об'єкту (наприклад процеси, що протікають в зоні формування тканин при взаємному переміщенні і деформації основних і утокових ниток; процеси протікають при транспортуванні пакету матеріалів на швейній машині; процеси тепло - масо переносу при шліхтуванні основи в підготовчому виробництві і ін.). В даному випадку, коли зовнішні параметри (характеризуючи умови функціонування проєктованого об'єкту) відомі і відомі відповідні їм вихідні параметри необхідно використовувати імітаційні моделі, побудовані за принципом чорного ящика. Особливу роль грають процедури синтезу при проєктуванні складних технічних об'єктів. Після рішення прямої або зворотної задачі для кожного конкретного структурного елемента, з яких складається технічна система, необхідно здійснити інтегрований параметричний синтез, який полягатиме в стиковці зовнішніх і вихідних параметрів кожного з об'єктів, що становлять. Аналогія цьому процесу простежується наприклад при розгляді рівноваги системи, що складається з n матеріальних тіл. Для плоскої системи сил, для кожного з тіл тих, що входять в систему можна скласти 3 рівняння рівноваги. Таким чином, для визначення всіх внутрішніх реакцій зв'язків необхідно буде скласти і спільно вирішити $3n$ рівнянь рівноваги.

Третій етап включає аналіз отриманих результатів і порівняння їх з існуючими результатами, які виходять шляхом проведення експерименту на макеті або при натурному експерименті. Даний етап є дуже важливим, оскільки дозволяє встановити чи задовольняє прийнята гіпотетична модель критерію практики, чи порівняна в межах точності отримана інформація про об'єкт з результатами спостережень явищ, що вивчаються. Вельми

істотним при цьому є вибір необхідної точності значення цільової функції, що набуває. Основним критерієм є абсолютна величина або діапазон зміни значень вихідних параметрів (наприклад, істотно різні вимоги до точності пред'являються при проектуванні окремих вузлів і механізмів у важкому і точному машинобудуванні).

Результати, отримані на третьому етапі, є основою для проведення четвертого, завершального етапу. У разі, коли результати аналізу математичної моделі співпадають в межах заданої точності з результатами експериментальних спостережень, то можна вважати, що запропонована математична модель задовольняє вимогам, що пред'являються. Інакше виникає необхідність модернізації математичної моделі. Таким чином, тут доречно говорити про її ускладнення з погляду обліку додаткових зовнішніх і внутрішніх параметрів, виявленню додаткового взаємозв'язку між ними. Отримувана ускладнена математична модель, відповідно до принципів об'єктно-орієнтованого моделювання, успадковує властивості свого прототипу. Дана ієрархія математичних моделей (як структурна, так і функціональна) дозволяє істотно розширити і збагатити знання про проєктований об'єкт.

Розробка математичних моделей технічних систем і технологічних процесів легкої і текстильної промисловості в першу чергу направлена на проведення процедури обчислювального експерименту. В зв'язку з цим є актуальним аналіз послідовності проведення окремих етапів процесу побудови математичної моделі і організації обчислювального експерименту. На рис.3.1 представлена спрощена блок-схема, яка визначає послідовність проведення проєктувальних розрахунків технічних систем (ТС) і технологічних процесів легкої і текстильної промисловості на початковій стадії проектування. Дана спрощена блок-схема дозволяє враховувати як структурні характеристики проєктованих технічних систем, так і властивості їх вхідних, вихідних і внутрішніх параметрів.

На початковому етапі проводиться аналіз спроектованої технічної системи на предмет можливості її ділення на окремі елементарні технічні об'єкти. Професор Зарубін В.С. характеризує їх як об'єкти мікро рівня.

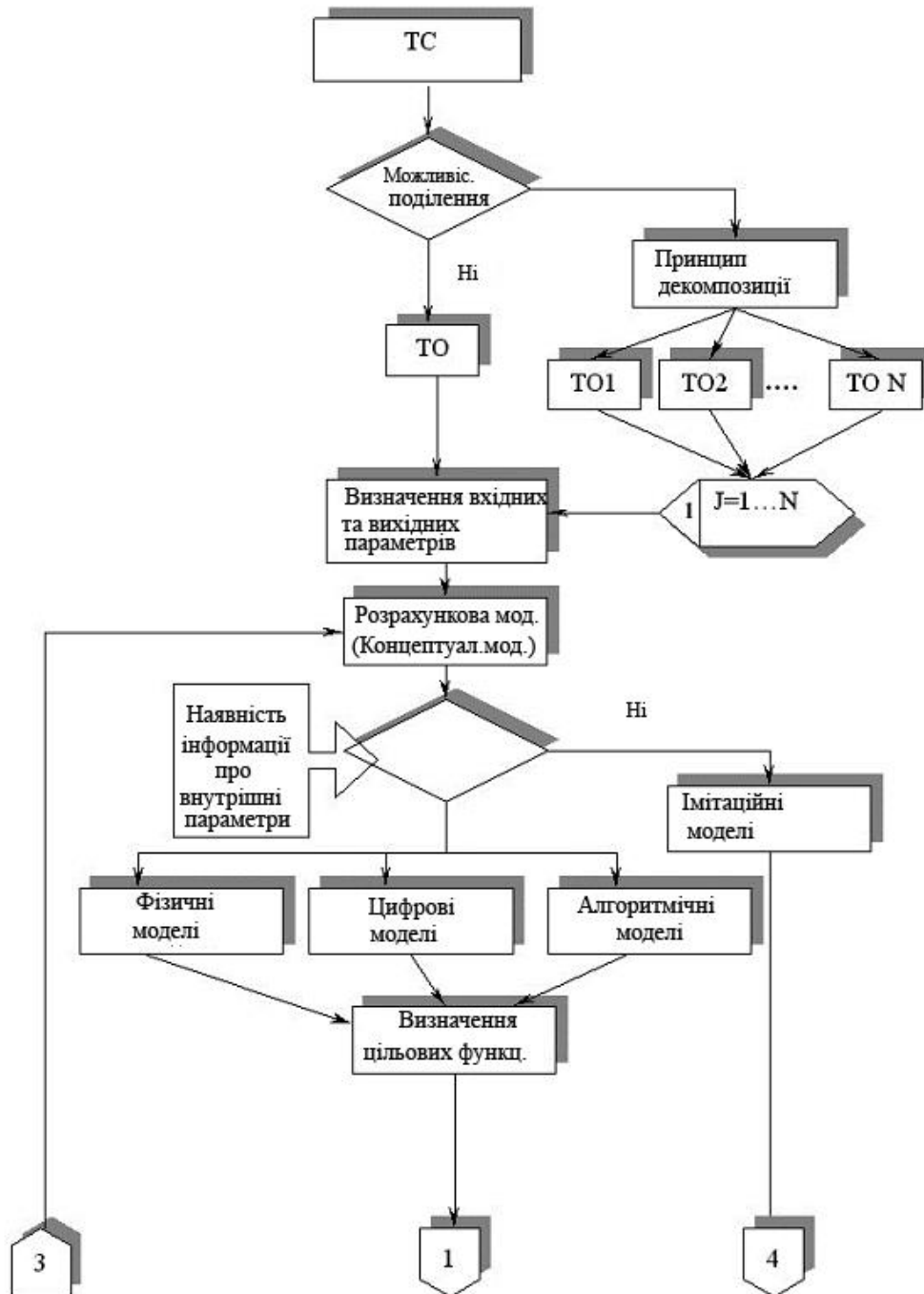
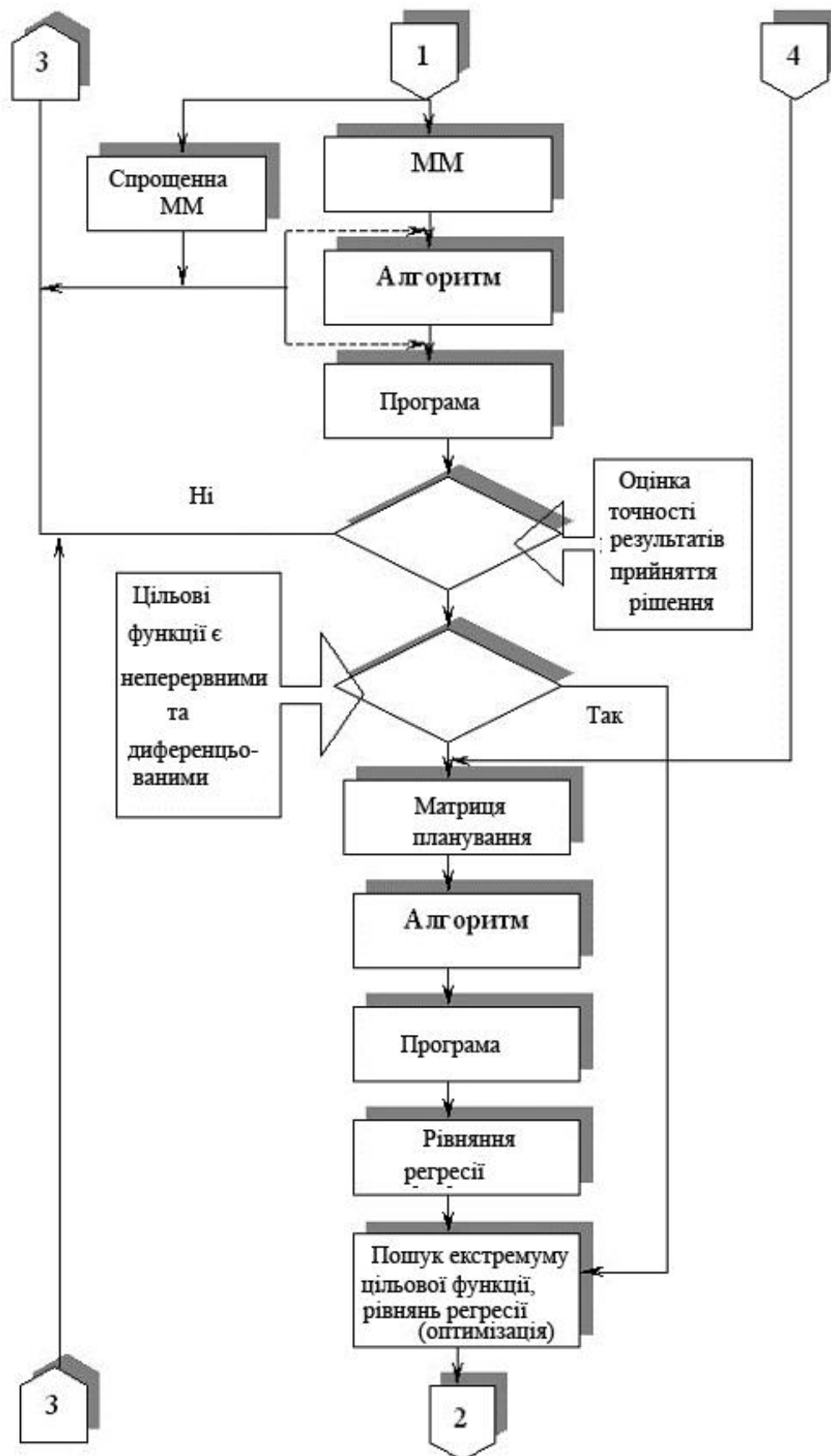
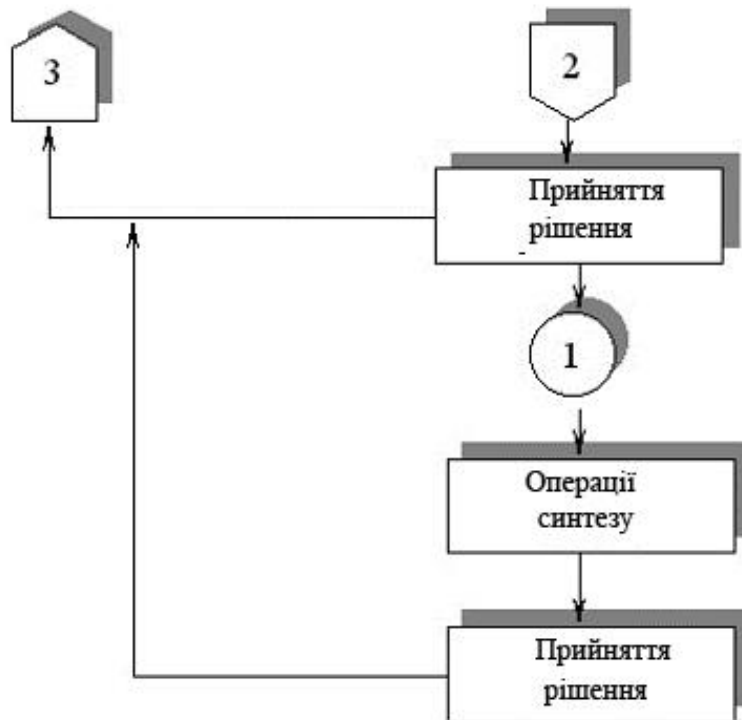


Рис.3.1. Блок-схема послідовності проведення проектувальних розрахунків



Продовження рис.3.1



Продовження рис.3.1

Детальніше питання структурної класифікації будуть розглянуті нижче, а на цьому етапі вважатимемо, що структурна ієрархія технічної системи може включати як об'єкти мікро-, так і об'єкти макрорівня.

У випадку якщо можливість ділення відсутня або недоцільна, то подальшому аналізу піддається технічний об'єкт, який буде еквівалентний технічній системі.

У випадку якщо можливість ділення відсутня або недоцільна, то подальшому аналізу піддається технічний об'єкт, який буде еквівалентний технічній системі. Інакше, використовуючи принцип декомпозиції, розбиваємо технічну систему на ряд технічних об'єктів $TO1, TO2, \dots, TON$, де N - число технічних об'єктів мікро рівня, з яких складається технічна система (наприклад, будь-який плоский механізм можна розбити на групи Асура, ступінь рухливості яких дорівнює нулю).

Наступним важливим етапом (см.рис.3.1) є визначення вхідних і вихідних (цільових функцій) параметрів, які впливають на технічний

об'єкт або є наслідком його реакції на дію вхідних і внутрішніх параметрів. Необхідно відзначити, що число вхідних і внутрішніх параметрів визначається ступенем їх впливу на вихідні параметри і необхідною точністю. Для технічної системи, що складається з ряду технічних об'єктів, для кожного з них може бути свій набір вхідних і внутрішніх параметрів, що пояснюється специфікою їх будови, фізичною природою процесів що протікають в них.

Визначившись з вибором вхідних і вихідних параметрів (для цього можна використовувати результати попередніх експериментів, експертні оцінки і ін.) можна приступати до побудови розрахункової схеми (концептуальній моделі) з використанням спеціальних прийомів і символів наочного графічного зображення (наприклад кінематичні і електричні схеми). Паралельно з цим визначається наявність інформації про внутрішні параметри технічного об'єкту. У разі, коли дана інформація відсутня, для подальшого аналізу необхідно використовувати імітаційні моделі, які побудовані за принципом чорного ящика. Для кожного вхідного в систему об'єкту в пам'яті ЕОМ підтримується таблиця його параметрів. Кожна елементарна взаємодія в системі реалізується у вигляді одного або декількох алгоритмів, що дозволяють розраховувати невідомі параметри взаємодії по заданих. Отриманий набір підпрограм доповнюється провідною програмою, що управляє процесом звернення до підпрограм відповідно до заданої структури системи [14].

За наявності інформації про внутрішні параметри можливе використання фізичних, цифрових і аналогових моделей. Фізичні моделі більш повно відтворюють властивості і поведінку досліджуваного технічного об'єкту. Дана модель широко використовується при дослідженні механізмів, машин і технологічних процесів легкої і текстильної промисловості. Це пояснюється тим, що науковою основою для вивчення властивостей і поведінки перерахованих вище об'єктів є

теоретична механіка, теорія механізмів і машин, механіка нитки, опір матеріалів, деталі машин, механічна технологія волокнистих матеріалів, які є по суті продовженням і розвитком окремих розділів курсу фізики.

Використання аналогових обчислювальних машин (АВМ) дозволило вивчати поведінку складних технічних систем на моделі, що має іншу фізичну природу чим у реального об'єкту або процесу. Тут можна відзначити широко відомі електромеханічні, електромагнітні аналогії, які дозволяють моделювати поведінку складних систем з використанням електричних ланцюгів. Аналогові моделі можуть бути як прямій, так і непрямій аналогії. По-перше поведінка різних по фізичній природі об'єктів описується подібними функціональними залежностями (наприклад другий закон Ньютона і закон Ома і ін.). У другому випадку, з використанням принципу декомпозиції, технічна система розбивається на об'єкти мікрорівня, поведінка яких моделюється певними функціональними залежностями. Вирішення отриманих трансцендентних, алгебри і диференціальних рівнянь здійснюється різними електронними вирішальними пристроями АВМ, які утворюють схеми аналогічні структурі даних рівнянь.

У разі, коли параметри стану технічної системи представляються у вигляді сукупності дискретних значень використовують цифрові моделі, які вирішуються на ЕОМ з використанням чисельних методів. Точність отримуваних результатів, в даному випадку, значно вище чим при використанні АВМ.

На наступному етапі, виходячи з проектного завдання, визначаються цільові функції, які можна розглядати як функціональні залежності загального вигляду між вхідними, внутрішніми і вихідними параметрами. Дана математична формалізація не дозволяє ще здійснювати якісний і кількісний аналіз. На цьому етапі відбувається вибір прямого або зворотного завдання проектування і визначаються параметри, які істотно

впливають на поведінку технічної системи. Даний етап закінчується побудовою математичної моделі з використанням отриманої раніше інформації і проектного завдання (см.рис.3.1).

Необхідно декілька слів сказати про основні властивості математичних моделей, які повинні задовольняти певним властивостям. Огляд літературних джерел дозволяє виділити наступні основні властивості математичних моделей: повнота, точність, адекватність, економічність, робастність, продуктивність, наочність. Зупинимося на аналізі даних показників.

Під повнотою математичної моделі мається на увазі ступінь її відповідності проектуваному технічному об'єкту щодо вихідного параметра оптимізації, який визначається технічним завданням. Наприклад, в процесі проектування нас цікавить розробка такого технічного об'єкту, функціональні параметри якого істотно важливіші за його структурні параметри (отримання необхідної траєкторії руху робочого органу механізму, при цьому обмеження на масу, геометричні розміри і число ланок відсутні або грають другорядну роль і ін.).

Під точністю математичної моделі мається на увазі необхідність прийнятного збігу результатів визначення значення цільових функцій (вихідних параметрів), отриманих при виконанні обчислювального експерименту, із значеннями, визначеними за допомогою експерименту.

Адекватність математичної моделі полягає в її здатності відображати основні властивості технічного об'єкту у визначених технічним завданням межах зміни вхідних і внутрішніх параметрів.

Під економічністю математичної моделі маються на увазі витрати часу на виконання обчислювальних операцій і ефективність використання пам'яті ЕОМ. Даний показник безпосередньо пов'язаний з проблемою ефективності запропонованого алгоритму рішення задачі. Сучасні можливості обчислювальної техніки цілком задовольняють, з погляду

продуктивності, тому кругу завдань, які виникають при проектуванні машин і технологічних процесів легкої і текстильної промисловості.

Робасність визначає стійкість математичної моделі до різного роду погрішностям, які виникають при виконанні обчислювального експерименту. Ці погрішності виникають при діленні числа на число близьке до нуля, при відніманні близьких по модулю величин. Вирішення цієї проблеми також пов'язане з умінням дослідника грамотно розробляти обчислювальний алгоритм.

Продуктивність математичної моделі залежить від якісно підготовленої початкової інформації, яка необхідна при виконанні обчислювального експерименту. Це дозволить ефективно проводити ті тільки якісний, але і кількісний аналіз явищ, що вивчаються, і процесів.

Останньою властивістю є наочність математичної моделі. Дана властивість дозволяє дослідникові на початковій стадії проектування оцінювати результати і своєчасно вносити корективи при проведенні обчислювального експерименту.

Подальші три блоки (см.рис.3.1) визначають собою попередній обчислювальний експеримент. Сюди входять блоки ММ (математична модель), алгоритм, програма. Окремо представлений блок – спрощена модель. Необхідність проведення попереднього обчислювального експерименту обумовлена вимогами, що пред'являються до точності обчислення дискретних значень цільової функції. Набутих значень порівнюються з відомими значеннями цільової функції, які можуть бути визначені в кінцевому числі точок області визначення цільового функціонала. У випадку якщо точність отриманого результату (для вибраних крапок) задовольняє пред'явленим вимогам, то тоді математична модель може бути прийнята і використана для подальших обчислень. Інакше необхідно її удосконалити починаючи з розрахункової схеми. На початковому етапі можна зробити спробу аналізу технічного об'єкту з

використанням спрощеної математичної моделі, яка включає мінімальне число вхідних і внутрішніх параметрів. Процедура отримання кінцевого результату і його порівняння з відомими даними аналогічна випадку як і для математичних моделей вищого ієрархічного порядку. У разі низької точності також відбувається перехід до блоку, що відповідає за розробку розрахункової схеми, яку необхідно уточнити.

Блок, що включає алгоритм рішення, є відомими методами вирішення отриманих математичних залежностей для цільової функції (це можуть бути чисельні методи вирішення трансцендентних, диференціальних рівнянь, систем лінійних рівнянь і ін.). Для виконання відповідного алгоритму необхідна програма, яка дозволяє реалізувати його на ЕОМ.

Цілком очевидно, що якщо вихідні параметри пов'язані з вхідними і внутрішніми параметрами явним чином і цільовими функціоналами є безперервні функції, що диференціюються, на вибраних інтервалах варіювання вхідних і внутрішніх параметрів, то завдання пошуку екстремуму цільових функцій (оптимізація) не викликає великих утруднень. На блок-схемі (см.рис.3.4) це враховується оператором умовного порівняння.

Серйозніше завдання виникатиме тоді, коли вихідні параметри не можна безпосередньо виразити через вхідні і внутрішні параметри. Наприклад, якщо цільовий функціонал представлений трансцендентним рівнянням або диференціальним рівнянням, яке дуже складно проінтегрувати. В цьому випадку необхідно використовувати відповідні чисельні методи, які дозволяють визначити значення цільового функціонала для кінцевого числа дискретних значень варіюваних змінних. В цьому випадку необхідно проводити основний обчислювальний експеримент з використанням методів активного планування експерименту. Тоді на основі проектного завдання складається відповідний план проведення експерименту, визначаються інтервали варіювання

вхідних і внутрішніх параметрів (змінних), визначаються значення цільових функцій для відповідних змінних з використанням розроблених алгоритмів і програмного забезпечення. Аналіз сказаного ще раз підкреслює важливість проведення попереднього експерименту, на стадії якого математична модель уточнюється до рівня відповідного вимогам, що пред'являються до результатів.

На наступному етапі проводиться визначення коефіцієнтів в рівнянні регресії, визначаються значущі коефіцієнти, проводиться оцінка адекватності отриманого регресійного рівняння. Такий підхід дозволяє перейти від дискретних значень цільової функції до рівняння, яке з певною довірчою вірогідністю описує технічний об'єкт, що вивчається. При цьому отримувана функція, на заданому інтервалі варіювання змінних, буде безперервною і такою, що диференціюється. Після цього можна визначити екстремуми (глобальні або локальні) і проводити оптимізацію.

У випадку якщо отримувані результати не задовольняють проектному завданню, необхідно повернутися до коректування розрахункової схеми.

Для визначення значення цільової функції в характерних точках плану необхідно також розробляти алгоритм і програму або використовувати вже відомі, які були отримані при проведенні попереднього обчислювального експерименту.

При проведенні основного обчислювального експерименту виникають певні труднощі, які пов'язані з тим, що паралельні чисельні дослідження, що виконуються в одній точці факторного простору, не дозволяють визначити математичне очікування і дисперсію значень цільового функціонала в характерних точках плану. За даними літературних джерел доцільно розглядати помилки дослідів як випадкові величини, створюючи множину, кількість елементів якого рівна кількості дослідів, і використовувати їх для оцінки адекватності рівняння регресії. Кожен елемент безлічі помилок визначається як різниця між обчисленими

на ЕОМ і отриманими по рівнянню регресії значенням цільового функціонала в досвіді.

Для технічних систем після блоку ухвалення рішення, з використанням операції циклу, відбувається перехід до технічного об'єкту *To2* і повторюється вся послідовність процедур описаних вище. Даний процес виконується до тих пір, поки не буде розглянутий останній технічний об'єкт *TON* (см.рис.3.1). Потім слідує операції синтезу отриманих рішень по кожному технічному об'єкту і ухвалюється рішення щодо всієї технічної системи.

3.2. КЛАСИФІКАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ

Класифікація всього різноманіття математичних моделей, які описують різні явища і об'єкти, є завданням вельми складної. Зважаючи на специфіку даної книги зупинимося на класифікації математичних моделей, які використовуються для дослідження технологічних процесів, а також для виконання аналізу і синтезу механізмів і машин легкої і текстильної промисловості. У основу даної класифікації можна покласти різні особливості і ознаки математичних моделей, на які указує проф. Зарубін В.С. Основними з них є ступінь деталізації технічного об'єкту, характер властивостей, що відображаються, і методи отримання і представлення математичної моделі.

На рис.3.2 представлена класифікація математичних моделей залежно від ступеня їх деталізації і з урахуванням характеру властивостей, що відображаються. Тут необхідно зробити декілька попередніх зауважень, які стосуються питань структурної ієрархії математичних моделей. Цілком очевидно, що елементарним технічним об'єктом може вважатися деяка система з розподіленими параметрами. Сукупність таких систем можна характеризувати як деяка множина, клас якої визначає його потужність.

Даний параметр може характеризуватися кардиналом (кардинальним числом) даної множини. Інтеграція даних систем дозволяє отримати новий об'єкт вищого класу. Наприклад, для технологічного устаткування елементарним технічним об'єктом може бути ланка (тверде тіло, що входить до складу механізму). Об'єктом вищого класу є механізм (система тіл, призначена для перетворення руху одного або декількох твердих тіл в необхідні рухи інших твердих тіл). Нарешті до вищого класу відноситься машина (пристрій, що виконує механічний рух для перетворення енергії, матеріалів і інформації з метою заміни або полегшення фізичної і розумової праці людини). На думку проф. Зарубіна В.С., для забезпечення найбільш загальної уніфікації, тут доречно говорити про моделі мікро-, макро- і мета рівня.

Як видно з рис.3.2 математичні моделі властивостей, що по характеру відображаються, можна класифікувати як моделі структурні і функціональні. Структурні моделі класифікуються на топологічні моделі $I...N$ (N - число топологічних моделей даного технічного об'єкту) і геометричні моделі $II..IN, NI...NN$. Топологічні моделі відносяться до вищого класу в структурній ієрархії і призначаються для з'ясування і уточнення взаємозв'язку між окремими елементами технічного об'єкту (представляються у вигляді графів, структурних схем, матриць, списків і ін.). Геометричні моделі відносяться до нижчого класу структурної ієрархії і містять відомості про форму і розміри елементів технічного об'єкту і про їх взаємне розташування.

Функціональні математичні моделі, як наголошувалося вище, можна розглядати як сукупність моделей мета- ($I...N$), макро- ($II..IN, NI...NN$) і мікро рівня ($III...IIN, INI...INN, NII...NIN, NNI...NNN$). При визначенні місця математичної моделі в даній класифікації, на попередньому етапі, визначається клас математичної моделі.

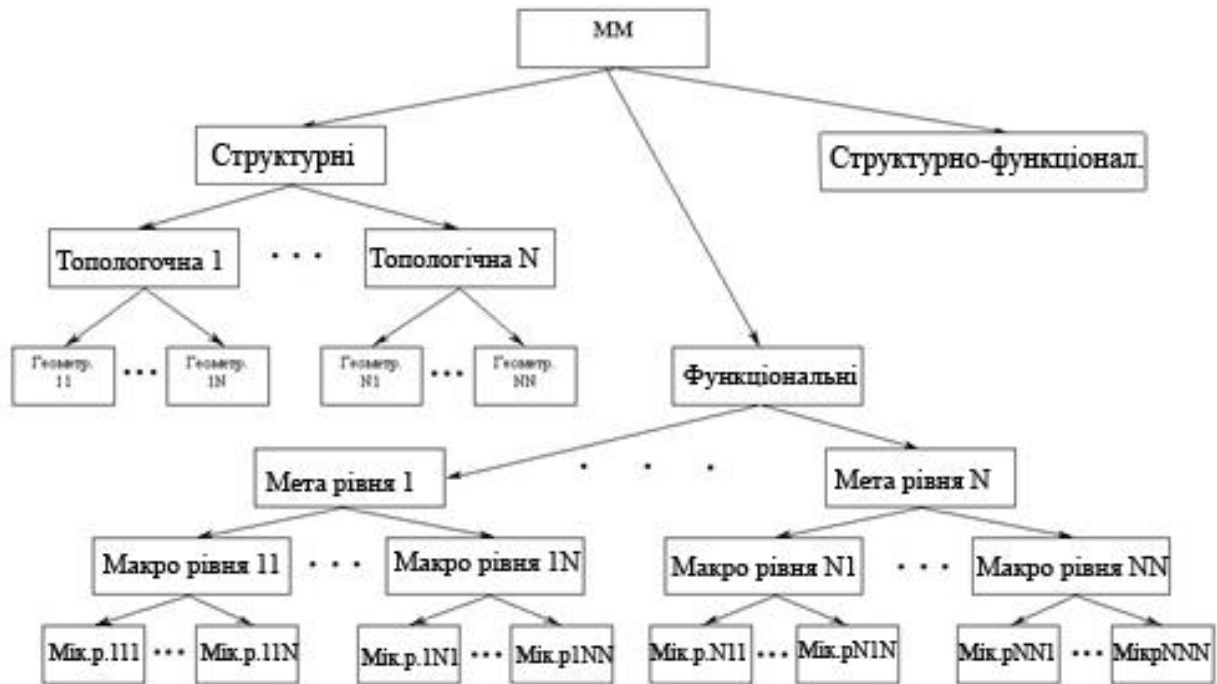


Рис.3.2. Класифікація математичних моделей залежно від ступеня їх деталізації

Потім визначається можливість додаткової деталізації на кожному з рівнів з відповідним кардинальним числом. Функціональні математичні моделі відображають фізичні, механічні, хімічні і ін. процеси, що відбуваються в технічних об'єктах.

Цілком очевидно, що для складних по своїй структурі об'єктів можливі інтегровані математичні моделі структурно-функціонального характеру. Такі моделі можуть використовуватися для опису технічних об'єктів на макро- і мета рівнях.

На рис.3.3 представлена класифікація математичних моделей по способах отримання і уявлення. Тут можна виділити теоретичні,

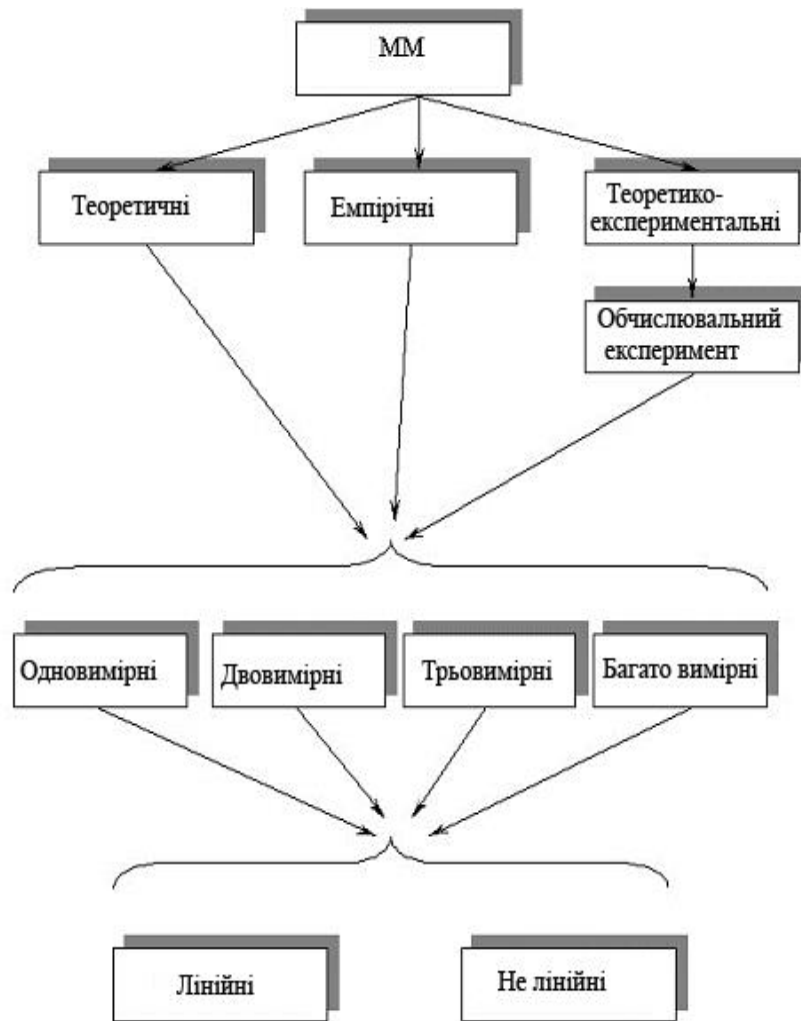


Рис.3.3. Класифікація математичних моделей по способах отримання і уявлення емпіричні (досвідчені, експериментальні) і теоретико-експериментальні моделі. Окремим випадком отримання теоретико-експериментальної моделі можна вважати обчислювальний експеримент (про нього мовилося вище). Для кожного вказаного вище виду математичної моделі отримувана цільова функція може залежати від одного (лінійне завдання), два (плоске завдання), три (просторове завдання) і багатьох чинників (вхідних і внутрішніх параметрів). Крім того, дані залежності можуть носити лінійний і нелінійний характер.

Нарешті розглянемо класифікацію математичних моделей по характеру властивостей вхідних, вихідних і внутрішніх параметрів (см.рис.3.4).



Рис.3.4. Класифікація математичних моделей по характеру властивостей вхідних, вихідних і внутрішніх параметрів

По характеру даних можна виділити стохастичні (серед даних, що представляються, містяться випадкові величини) і детерміновані (визначені) математичні моделі. Можливий варіант, коли містяться дані двох типів, тоді говорять про змішані моделі. Змішані моделі можуть відноситися тільки до моделей макро- і метарівня.

По характеру зміни параметрів від часу розрізняють статичні і нестаціонарні моделі. Останні підрозділяються на динамічних і

квазістатичних. У разі статичних моделей параметри не залежать від часу. Коли враховуються інерційні властивості технічних об'єктів необхідно використовувати динамічні моделі. Квазістатичні моделі відповідають випадку, коли параметри змінюються від часу з дуже малою швидкістю (наприклад, зміна діаметру навою основов'язальних і текстильних машин).

Значень цільових функціоналів моделі, що по характеру набувають, підрозділяються на моделі, де дані носять безперервний характер і на моделі з дискретними значеннями вхідних параметрів. На макро- і мета рівні можуть бути випадки, коли моделі носитимуть змішаний характер.

4. МАТЕМАТИЧНІ МОДЕЛІ, ЩО РЕАЛІЗОВУЮТЬСЯ З ВИКОРИСТАННЯМ ЧИСЕЛЬНИХ МЕТОДІВ

4.1. СИСТЕМИ ЛІНІЙНИХ РІВНЯНЬ

При проведенні статичних і динамічних досліджень механічних систем доводиться стикатися з необхідністю вирішення систем лінійних рівнянь. Тут можна говорити про їх використання не тільки при вивченні рівноваги механічних об'єктів, але і при вирішенні систем диференціальних рівнянь як звичайних, так і в приватних похідних механічних систем, що описують поведінку, з великим числом мір свободи.

У загальному випадку систему лінійних рівнянь можна представити таким чином [10]

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k + \dots + a_{2n}x_n &= b_2, \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mk}x_k + \dots + a_{mn}x_n &= b_m, \end{aligned} \quad (4.1.1)$$

де $a_{11} \dots a_{mn}$ – коефіцієнти при невідомих величинах; $x_1 \dots x_n$ – невідомі (при вирішенні завдань статички і динаміки це можуть бути реакції зв'язків, узагальнені координати, швидкості і прискорення); $b_1 \dots b_m$ – постійні величини (необхідно відзначити, що враховуючи фізичну суть досліджуваних статичних і динамічних процесів розмірність парних творів в лівій частині рівності системи (4.1.1) повинна співпадати з розмірністю постійних $b_1 \dots b_m$); n – число невідомих, які входять в систему; m – число рівнянь.

Систему рівнянь (4.1.1) зручно представити у векторній формі

$$A\bar{x} = \bar{b}, \quad (4.1.2)$$

де A – матриця коефіцієнтів $m \times n$; \bar{x} – шуканий n – компонентний вектор; \bar{b} – заданий m – компонентний вектор.

Враховуючи, що нижче розглядатимуться тільки статично визначні завдання (число невідомих дорівнює числу можливих рівнянь $m=n$), то систему лінійних рівнянь (4.1.1) можна представити у вигляді (матриця коефіцієнтів перетвориться до квадратного вигляду)

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k + \dots + a_{2n}x_n &= b_2, \\ \dots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nk}x_k + \dots + a_{nn}x_n &= b_n. \end{aligned} \tag{4.1.3}$$

Визначник (детермінант) матриці коефіцієнтів A визначимо рівністю

$$\Delta = \det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{vmatrix} \tag{4.1.4}$$

Тут необхідно відзначити, що у разі, коли $\Delta \neq 0$, то матриця називається не виродженою. У разі, коли заданий компонентний вектор $\bar{b} = 0$ – система (4.1.3) називається однорідною і має ненульове рішення при $\Delta = 0$. Якщо система рівності (4.1.3) має рішення, то вона називається сумісною. Інакше вона називатиметься суперечливою або несумісною. За наявності єдиного вирішення системи (4.1.3) можна говорити про те, що система є визначеною. За наявності двох і більш за рішення система буде невизначеною [11].

Випадок, коли детермінант $\Delta \neq 0$ забезпечує існування єдиного рішення.

Перейдемо до визначення методів вирішення системи рівнянь (4.1.3). Точне рішення в явному виді даної системи може бути отримане з використанням формули Крамера. Суть методу полягає в послідовному діленні перетвореного визначника (у якого відповідний стовпець коефіцієнтів лівої частини системи замінюється стовпцем $b_1 \dots b_n$) на

початкового визначника, що складається з коефіцієнтів лівої частини системи рівнянь. Для визначення невідомих $x_1 \dots x_n$ необхідно виконати n перестановок

$$\begin{aligned}
 x_1 &= \frac{\begin{vmatrix} b_1 & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{vmatrix}}, \\
 \dots & \\
 x_k &= \frac{\begin{vmatrix} a_{11} & \dots & a_{1k-1} & b_1 & a_{1k+1} & \dots & a_{1n} \\ a_{21} & \dots & a_{2k-1} & b_2 & a_{2k+1} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nk-1} & b_n & a_{nk+1} & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{vmatrix}}, \\
 \dots & \\
 x_n &= \frac{\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & b_1 \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & b_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & b_n \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{vmatrix}}.
 \end{aligned} \tag{4.1.5}$$

Даний метод з обчислювальної точки зору є не ефективним. Для визначення всього коріння $x_1 \dots x_n$ необхідно виконати з використанням системи (4.1.5) $(n^2 - 1) \cdot (n!)$ множень і n ділень. Для системи складається з 50 рівнянь необхідно буде виконати $7,610^{67}$ операцій множення. Виконання такого числа операцій в рамках реальної тимчасової протяжності недоступно сучасним ЕОМ. Як відомо, рівновага одного тіла під дією просторової системи сил описується 6 рівняннями. Для дослідження рівноваги механічної системи з 9 тіл буде потрібно 54 рівняння рівноваги з 54 невідомими. А сучасні машини легкої і текстильної промисловості і їх окремі вузли складаються із значно більшого числа ланок.

Виходячи з цього ефективними методами визначення невідомих в системі рівнянь (4.1.3) можна вважати прямі і ітераційні методи. Використання перших дозволяє за кінцеве число дій отримати точне вирішення системи, якщо всі величини задані і обчислення проводяться точно, без помилок при округленні результату. Ітераційні методи полягають в отриманні наближеного вирішення системи лінійних рівнянь із заданою точністю шляхом побудови ітераційної послідовності, що сходиться до шуканого рішення.

Вибір того або іншого методу залежить від характеристик ЕОМ, числа рівнянь системи, характеристик матриці коефіцієнтів A . Наприклад, при складанні рівнянь рівноваги твердого тіла деякі коефіцієнти a_{ij} дорівнюють нулю. У таких випадках використовувати метод Гауса послідовного виключення невідомих і метод Гауса з вибором головного елемента не можна. Тому, що при реалізації алгоритму на одному з етапів обчислень відбувається ділення попереднього результату на 0. Машина видає повідомлення про виконання некоректної операції.

Нижче ми розглянемо декілька різних методів на конкретних прикладах і постараємося показати як сильні, так і слабкі сторони кожного з них.

Метод послідовного виключення невідомих Гауса заснований на послідовному пониженні порядку системи (4.1.3) за рахунок виключення невідомих $x_1 \dots x_{n-1}$ в лінійних рівняннях. Це приводить до отримання матриці коефіцієнтів a_{ij} трикутного вигляду. Ця послідовність процедур характеризується як прямий хід виключення змінних. Для його здійснення необхідно послідовно віднімати перше рівняння з подальших, умножуючи його ліву і праву частини на постійні коефіцієнти, що представляють приватне від ділення постійних коефіцієнтів $m_{ji}=a_{ji}/a_{ii}$, де $i = 1, 2, \dots, n-1; j= i + 1, i + 2, \dots, n$. Кінцева система рівнянь на прямому ході має вигляд

$$\begin{aligned}
 &a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k + \dots + a_{1n}x_n = b_1, \\
 &a_{22}^{(1)}x_2 + \dots + a_{2k}^{(1)}x_k + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\
 &\dots\dots\dots \\
 &a_{kk}^{(k-1)}x_k + \dots + a_{kn}^{(k-1)}x_n = b_k^{(k-1)}, \\
 &\dots\dots\dots \\
 &a_{nn}^{(n-1)}x_n = b_n^{(n-1)},
 \end{aligned}
 \tag{4.1.6}$$

де

$$\begin{aligned}
 a_{22}^{(1)} &= a_{22} - \frac{a_{21}}{a_{11}}a_{12}; \dots a_{2k}^{(1)} = a_{2k} - \frac{a_{21}}{a_{11}}a_{1k}; \dots a_{2n}^{(1)} = a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}; b_2^{(1)} = b_2 - \frac{a_{21}}{a_{11}}b_1; \\
 a_{kk}^{(k-1)} &= a_{kk}^{(k-2)} - \frac{a_{kk-1}^{(k-2)}}{a_{k-1k-1}^{(k-2)}}a_{k-1k}^{(k-2)}; \dots a_{kn}^{(k-1)} = a_{kn}^{(k-2)} - \frac{a_{kk-1}^{(k-2)}}{a_{k-1k-1}^{(k-2)}}a_{k-1n}^{(k-2)}; b_k^{(k-1)} = b_k^{(k-2)} - \frac{a_{kk-1}^{(k-2)}}{a_{k-1k-1}^{(k-2)}}b_{k-1}^{(k-2)}; \\
 a_{nn}^{(n-1)} &= a_{nn}^{(n-2)} - \frac{a_{nn-1}^{(n-2)}}{a_{n-1n-1}^{(n-2)}}a_{n-1n}^{(n-2)}; b_n^{(n-1)} = b_n^{(n-2)} - \frac{a_{nn-1}^{(n-2)}}{a_{n-1n-1}^{(n-2)}}b_{n-1}^{(n-2)}.
 \end{aligned}$$

Блок-схема програми реалізації методу Гауса послідовного виключення невідомих показана на рис.4.1.1. Програма дозволяє реалізувати даний алгоритм (*PROGRAM_S_G_PE*) приведена в Додатку.

Декілька слів необхідно сказати про недоліки даного методу. Розглянемо систему лінійних рівнянь

$$\begin{aligned}
 1 \cdot 10^{-11} x_1 - 3x_2 + 2x_3 &= 1, \\
 1 \cdot 10^{10} x_1 - 5x_2 + 4x_3 &= 2, \\
 10x_1 - 7x_2 + 3x_3 &= 9.
 \end{aligned}
 \tag{4.1.7}$$

Вирішення (4.1.7) з використанням методу Гауса послідовного виключення невідомих дає (обмежуємося чотирма значущими цифрами після коми)

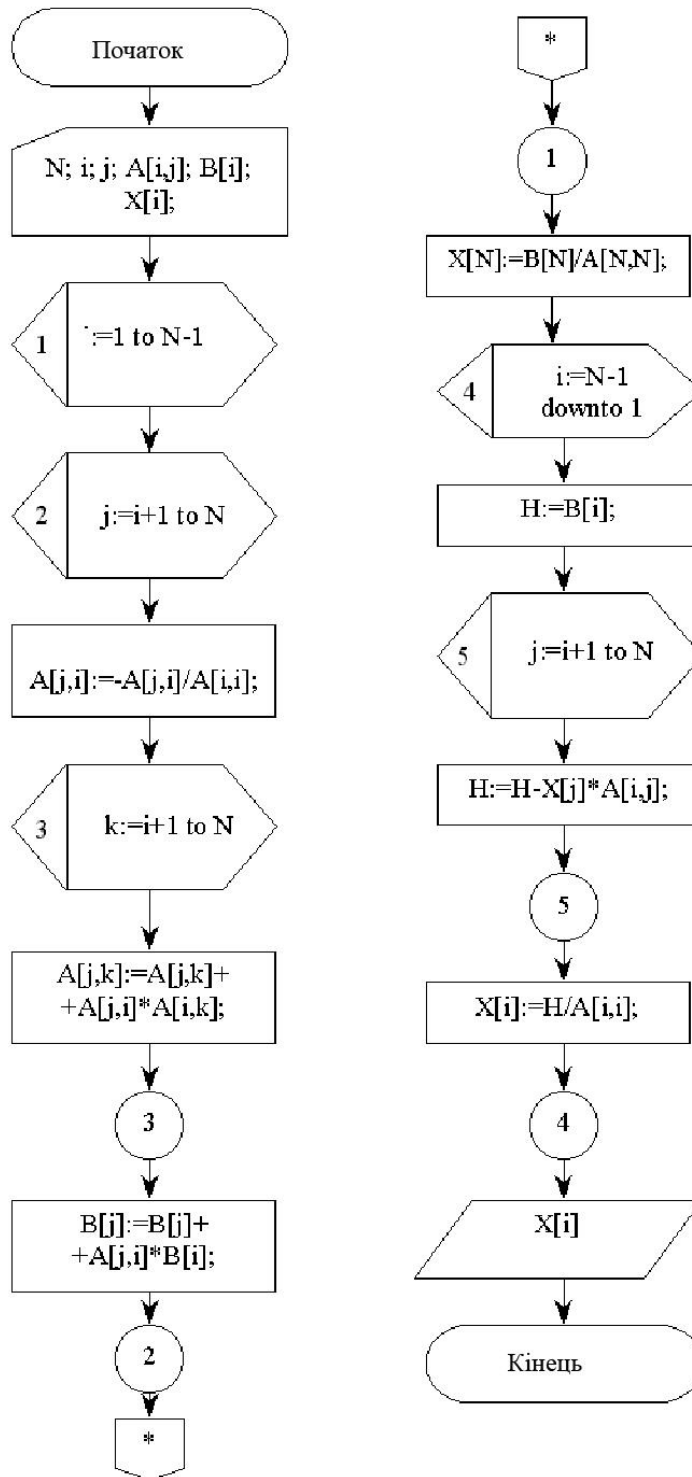
$$x_1 = 0,0000 ; x_2 = -2,6667; x_3 = -3,5000 .$$

Переставимо рівняння системи (4.1.7) місцями. Друге на місце першого, третє на місце другого, а перше на місце третього. Знову визначимо коріння

$$x_1 = 0,0000 ; x_2 = -3,0000; x_3 = -4,0000 .$$

Як видно, останній результат дозволяє отримати тотожності для першого і третього рівнянь системи (4.1.7). Отримане раніше коріння не дозволяє цього. Отже, можна говорити про те, що порядок рівнянь може впливати

на кінцевий результат. Якби можна було здійснити ідеальний обчислювальний процес без помилок округлення (що присутній в реальному обчислювальному процесі), то незалежно від порядку рівнянь в обох випадках отримали б одну і ту ж відповідь.



Блок-схема PROGRAM S_G_PE

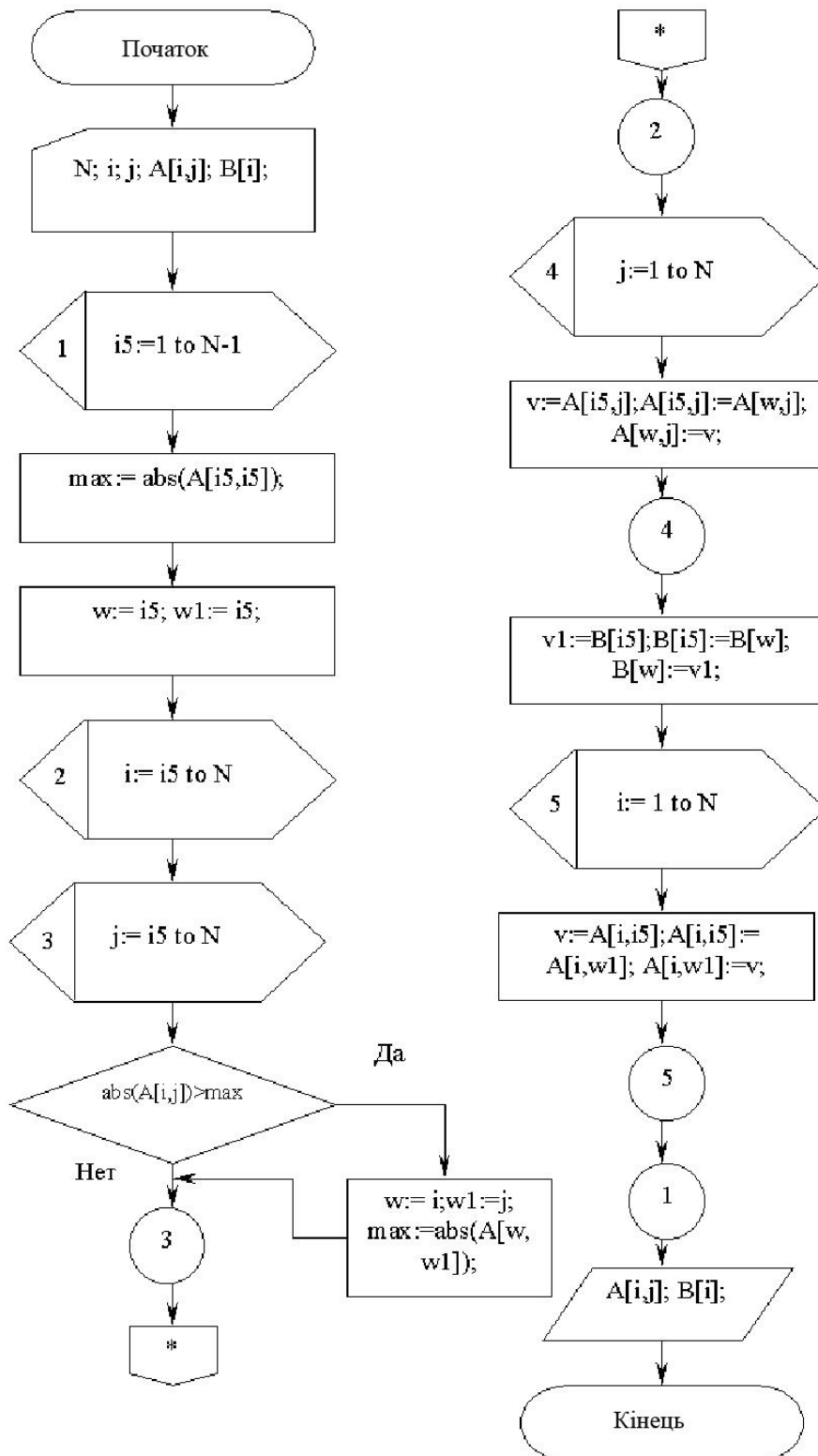
Якщо ми при обчисленнях обмежимося не 4, а 11 значущими цифрами після коми, то тоді останній результат матиме вигляд

$$x_1 = 0,0000000000\mathcal{X} ; x_2 = -2,999999998\mathcal{X} ; x_3 = -3,999999998\mathcal{X} .$$

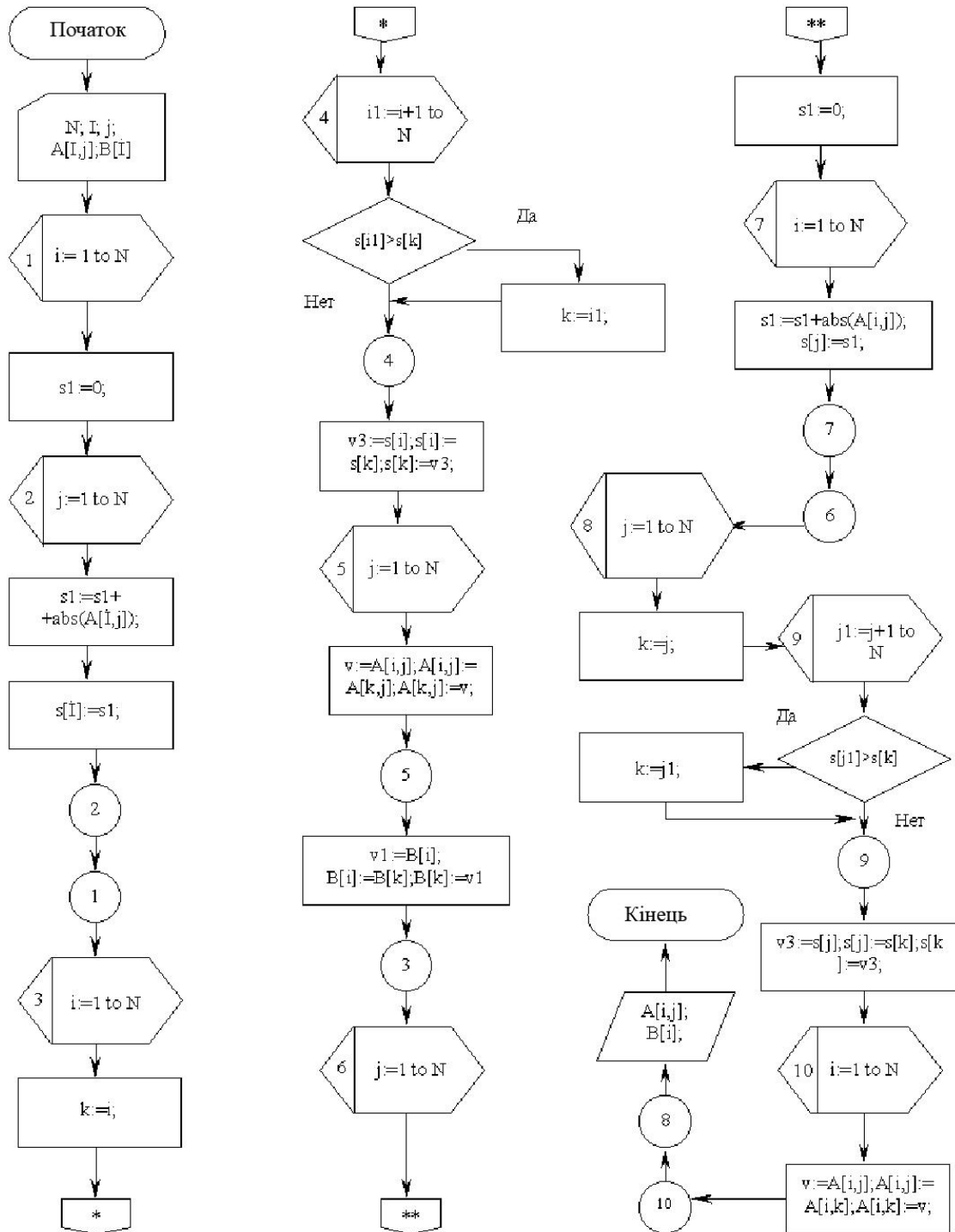
При такому результаті після підстановки в (4.1.7) всі три рівняння звертаються в тотожність.

Така втрата точності пояснюється дуже маленьким значенням коефіцієнта a_{11} , що приводить до різкого зростання коефіцієнтів $a^{(1)}_{22} \dots a^{(1)}_{2n} ; b^{(1)}_2 ; \dots a^{(1)}_{n2} \dots a^{(1)}_{nn} ; b^{(1)}_n$ на першому кроці прямого ходу. Для виключення цього використовуються різні схеми перетворення матриць коефіцієнтів на початковому етапі перед визначенням коріння. На рис.4.1.2 і 4.1.3 представлені блок-схеми програм, що реалізують два методи подібного перетворення. Перший з них полягає в перетворенні матриці коефіцієнтів на основі вибору головного (ведучого) елемента, який по модулю має максимальне значення серед інших. На першому кроці вибирається максимальний по модулю елемент a_{ij} . Даний рядок матриці переставляється на перше місце $a_{1j} = a_{ij}$. Потім стовпець, що містить максимальний по модулю елемент $\max = \text{abs}(a_{ij})$, міняється місцями з першим стовпцем $a_{11} = a_{ij}$. Таким чином, елемент a_{ij} переміщається в крайню ліву позицію на головній діагоналі матриці коефіцієнтів. На другому кроці розглядають ту, що залишилася $n-1$ систему рівнянь. Знову визначається максимальний по модулю елемент і переміщається на другу позицію на головній діагоналі. Процес продовжується до тих пір, поки останні два рівняння системи (4.1.3) не будуть розташовані в порядку, визначуваному абсолютною величиною їх постійних коефіцієнтів. Програма *PROGRAM PRE_MATR_VGE*, що дозволяє реалізувати цей метод, приведена в Додатку. Метод вирішення системи лінійних рівнянь, побудований на такому перетворенні матриці коефіцієнтів, носить назву методу Гауса з вибором головного елемента. Програма *PROGRAM S_G_VGE* визначення коріння системи лінійних

рівнянь з реалізацією методу Гауса (з вибором головного елемента) приведена в Додатку.



Блок-схема PROGRAM PRE_MATR_VGE



Блок-схема PROGRAM PRE_MAT_MAX_SUM

Інший метод полягає в наступному (см.рис.4.1.3): на першому кроці підсумовуються абсолютні значення всіх коефіцієнтів матриці по рядках $\sum_{j=1}^n abs(a_{ij})$. Після цього відбувається їх порівняння і рядки матриці розташовуються в порядку убутання суми $s1$. На другому кроці

відбувається підсумовування модулів всіх елементів матриці коефіцієнтів по стовпцях. Після цього відбувається порівняння отриманих сум, і стовпці розташовуються зліва на право в порядку убутання сум. Програма **PROGRAM PRE_MAT_MAX_SUM** для реалізації даного методу перетворення матриці коефіцієнтів приведена в Додатку.

З великими складнощами доводиться стикатися, коли ряд коефіцієнтів матриці дорівнює нулю. В цьому випадку може виникнути ситуація, коли машина виконає не коректну операцію (ділення на нуль) і виконання програми буде припинено. Для виключення цього можна використовувати метод обертання, який дозволяє приводити систему лінійних рівнянь (4.1.3) до системи лінійних рівнянь з правою трикутною матрицею на прямому ходу по наступних формулах

$$\begin{aligned} M_{ki}(\sum_{j=1}^n a_{ij}x_j - b_i) - L_{ki}(\sum_{j=1}^n a_{kj}x_j - b_k) &= 0, \\ L_{ki}(\sum_{j=1}^n a_{ij}x_j - b_i) - M_{ki}(\sum_{j=1}^n a_{kj}x_j - b_k) &= 0, \\ M_{ki} &= \frac{a_{ii}}{\sqrt{a_{ii}^2 + a_{ki}^2}}, \quad L_{ki} = -\frac{a_{ki}}{\sqrt{a_{ii}^2 + a_{ki}^2}}. \end{aligned} \quad (4.1.8)$$

де $i=1,2,\dots,n-1; k=i+1,\dots,n$.

У випадку якщо $a_{ii} = 0, a_{ki} \neq 0$, то $M_{ki} = 0, L_{ki} = 1$. Коли $a_{ii} = a_{ki} = 0$, то $M_{ki} = 1$ і $L_{ki} = 0$. Програма **PROGRAM S_M_VRAS** для реалізації методу обертання приведена в Додатку.

Ітераційні методи вирішення системи лінійних рівнянь розглянемо на прикладі методу простих ітерацій (програма **PROGRAM S_I_PRO**, що реалізовує цей метод, приведена в Додатку) і методу Зейделя (програма **PROGRAM S_I_ZEID** приведена в Додатку).

Розглянемо метод простих ітерацій, реалізація якого здійснюється з використанням наступної формули ітераційного процесу

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}} \left(\sum_{k=1}^n a_{ik}x_{k(j)} - b_i \right). \quad (4.1.9)$$

Обчислення продовжуються до тих пір, поки $x_i^{(j+1)} - x_i^{(j)} > \varepsilon$, де ε - задана погрішність обчислень коріння системи лінійних рівнянь.

Декілька слів необхідно сказати про достатню умову збіжності методу простих ітерацій. Якщо

$$C = \max_{1 \leq i \leq n} (C_i) < 1, \quad (4.1.10)$$

де $C_i = \left(\sum_{j=1}^n \left| \frac{a_{ij}}{a_{ii}} \right| - 1 \right)$, $i=1,2,\dots,n$, то система лінійних рівнянь (4.1.3) має єдине

рішення і ітераційний процес сходиться до рішення із швидкістю геометричної прогресії. Умову (4.1.10) можна інтерпретувати і так, ітераційний процес сходиться, якщо величина модуля кожного діагонального елемента матриці коефіцієнтів більше суми модулів решти елементів. Така умова вимагає, щоб матрицю коефіцієнтів (перед визначенні коріння системи лінійних рівнянь) перетворити до вигляду, коли на головній діагоналі матриці розташовуватимуться максимальні по модулю елементи.

Метод Зейделя організовує ітераційний процес по наступній формулі

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}} \left(\sum_{k=1}^{i-1} a_{ik} x_{k(j+1)} + \sum_{k=i}^n a_{ik} x_{k(j)} - b_i \right). \quad (4.1.11)$$

Його відмінність від методу простих ітерацій полягає в тому, що уточнені значення $x_i^{(j+1)}$ відразу підставляються в подальші рівняння.

Достатньою умовою збіжності методу Зейделя є наступний вираз

$$\sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{a_{ij}}{a_{ii}} \right| \leq q, \quad i=1\dots n, \quad q < 1. \quad (4.1.12)$$

Зазвичай метод Зейделя сходиться швидше, ніж метод простих ітерацій.

Декілька слів необхідно сказати про обумовленість матриці коефіцієнтів, яка визначається з виразу

$$\mu = \frac{\sup \left[\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} \right]}{\inf \left[\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} \right]} \geq 1. \quad (4.1.13)$$

Якщо обумовленість матриці коефіцієнтів μ близька до 1, то для такої системи відносна помилка при відшуканні коріння рівнянь порівнянна з відносною погрішністю, з якою задається права частина. Система буде погано обумовленою якщо $\mu > 1$.

Розглянемо простій приклад. Хай дана система двох рівнянь

$$\begin{aligned} 2x_1 + 0x_2 &= 2, \\ 5x_1 + 0,05x_2 &= 5. \end{aligned}$$

Вирішення даної системи рівнянь має вид $x_1=1, x_2=0$. Трохи змінимо праву частину другого рівняння, тоді

$$\begin{aligned} 2x_1 + 0x_2 &= 2, \\ 5x_1 + 0,05x_2 &= 5,05. \end{aligned}$$

Вирішення нової системи рівнянь матиме вид $x_1=1, x_2=1$. Незначна зміна правій частині другого рівняння привела до істотної зміни рішення. Визначимо по (4.1.13) обумовленість матриці коефіцієнтів останньої системи

$$\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} = \sqrt{29x_1^2 + 0,5x_1x_2 + 2,5 \cdot 10^{-3} x_2^2}. \quad (4.1.14)$$

Знайдемо найбільше і найменше значення функції (4.1.14) на одиничному колі із заміною $x_1=\cos \alpha, x_2=\sin \alpha$. Тоді

$$\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} = \sqrt{29 \cos^2 \alpha + 0,5 \cos \alpha \sin \alpha + 2,5 \cdot 10^{-3} \sin^2 \alpha}. \quad (4.1.15)$$

Пошук безумовного екстремуму функції (4.1.15) однієї змінної дозволив визначити

$$\begin{aligned} \sup \left[\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} \right] &= 5,385, \\ \inf \left[\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right)^2} \right] &= 0,0186 . \end{aligned} \quad (4.1.16)$$

Підставляємо результати (4.1.16) в (4.1.13) і визначаємо обумовленість матриці коефіцієнтів

$$\mu = \frac{5,385}{0,0186} = 289,54 .$$

Отриманий результат дозволяє зробити вивід про те, що погана обумовленість матриці коефіцієнтів приводить до істотно різних вирішень розглянутої системи двох лінійних рівнянь при незначній зміні правій частині другого рівняння на **0,05**.

4.2. ТРАНСЦЕНДЕНТНІ І АЛГЕБРАЇЧНІ РІВНЯННЯ

Важливу роль в історії математики, в розвитку її ідей і методів зіграли рівняння. Вони знаходять найширше розповсюдження при вирішенні теоретичних і прикладних завдань САПР устаткування і технологічних процесів легкої і текстильної промисловості.

Методи вирішення лінійних рівнянь були відомі ще стародавнім грекам. Коріння квадратного рівняння (рівняння алгебри 2-го ступеня)

$$ax^2 + bx + c = 0, \quad (4.2.1)$$

де **a**, **b**, **c** – коефіцієнти рівняння (4.2.1); **x** – невідоме; визначаються по формулах

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \quad (4.2.2)$$

Залежно від знаку дискримінанта $D = b^2 - 4ac$ маємо: два дійсних і різних кореня ($D > 0$); два комплексні корені ($D < 0$); один кратний корінь ($D = 0$) $-b/2a$. Формули Вієта

$$x_1 + x_2 = -\frac{b}{a}, \quad x_1 x_2 = \frac{c}{a}, \quad (4.2.3)$$

зв'язують між собою коріння і коефіцієнти квадратного рівняння (4.2.1). Теорема Вієта, для позитивного коріння, встановлює справедливість співвідношень між ними і коефіцієнтами многочлена [14]

$$P(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0, \quad (4.4.2)$$

у вигляді

$$\begin{aligned} a_0 &= (-1)^n x_1 x_2 \dots x_n, \\ a_1 &= (-1)^{n-1} (x_1 x_2 \dots x_{n-1} + x_1 \dots x_{n-2} x_n + \dots + x_2 x_3 \dots x_n), \\ &\dots\dots\dots \\ a_{n-2} &= x_1 x_2 + x_1 x_3 + \dots + x_{n-1} x_n, \\ a_{n-1} &= -(x_1 + x_2 + \dots + x_n). \end{aligned} \quad (4.2.5)$$

Перші вирішення квадратних рівнянь (4.2.1) з позитивними коефіцієнтами (для позитивного коріння) виявлені ще в працях Мухаммеда аль Хорезмі і Діофанту. У XVI столітті М.Штіфель запропонував узагальнений метод рішення (4.2.1) включаючи і негативне коріння. Остаточне рішення, з урахуванням випадку уявного коріння, запропоноване Р.Бомбеллі і С.Стевіном [14].

У рівнянні (4.4.2) n називають ступенем рівняння. Вирішення рівняння $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$ полягає в знаходженні всього його коріння, лежачого в даній області значень невідомого. Вирішення рівнянь третього ($n = 3$) і четвертого ($n = 4$) ступенів було отримане італійськими математиками Дж. Кардано, Н.Тарталья і Л.Феррарі в XVI столітті.

Рівняння алгебри 3-го ступеня

$$ax^3 + bx^2 + cx + d = 0, \quad a \neq 0, \quad (4.2.6)$$

приводимо до канонічного вигляду

$$y^3 + py + q = 0, \quad (4.2.7)$$

де $y = x + \frac{b}{3a}$; $p = -\frac{b^2}{3a^2} + \frac{c}{a}$; $q = \frac{2b^3}{27a^3} - \frac{bc}{3a^2} + \frac{d}{a}$.

Коріння рівняння (4.2.7) визначається по формулах Дж. Кардано

$$y_1 = A + B, \quad y_{2,3} = -\frac{A+B}{2} \pm i \frac{A-B}{2} \sqrt{3},$$

$$A = \sqrt[3]{-\frac{q}{2} + \sqrt{Q}}, \quad B = \sqrt[3]{-\frac{q}{2} - \sqrt{Q}},$$

$$Q = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2.$$
(4.2.8)

У якості A і B вибираються будь-які значення кубічного коріння, що задовольняє рівності $AB = -p/3$.

Метод Л. Феррарі вирішення рівнянь 4-го ступеня зводиться до рішення одного кубічного і двох квадратних рівнянь. Рівняння четвертого ступеня

$$ax^4 + bx^3 + cx^2 + dx + e = 0, \tag{4.2.9}$$

шляхом заміни $y = x + \frac{b}{4a}$ приводиться до неповного вигляду

$$y^4 + py^2 + qy + r = 0. \tag{4.2.10}$$

Коріння рівняння (4.2.10) визначається по формулах

$$y_1 = \frac{1}{2}(\sqrt{z_1} + \sqrt{z_2} + \sqrt{z_3}), \quad y_2 = \frac{1}{2}(\sqrt{z_1} - \sqrt{z_2} - \sqrt{z_3}),$$

$$y_3 = \frac{1}{2}(-\sqrt{z_1} + \sqrt{z_2} - \sqrt{z_3}), \quad y_4 = \frac{1}{2}(-\sqrt{z_1} - \sqrt{z_2} + \sqrt{z_3}).$$
(4.2.11)

де z_1, z_2, z_3 - коріння кубічної резольвенти

$$z^3 + 2pz^2 + (p^2 - 4r)z - q^2 = 0.$$

Знаки перед корінням в системі (4.2.11) вибираються так, щоб виконувалася умова $\sqrt{z_1}\sqrt{z_2}\sqrt{z_3} = -q$.

Безрезультатні спроби (що тривали близько 300 років) завершилися в XIX в., направлені на вирішення рівнянь 5-ої і вищих ступенів, дозволили норвезькому математикові Н.Абелю сформулювати відому теорему: ні для якого n , більшого або рівнішого п'яти, не можна вказати формулу, яка виражала б коріння будь-якого рівняння n -й ступеня через його коефіцієнти за допомогою радикалів і арифметичних дій [14].

Ще більші складнощі виникають при вирішенні рівнянь не алгебри (трансцендентних), у яких визначити явні вирази для коріння не представляється можливим.

Вирішення трансцендентних рівнянь $f(x)=0$ рекомендується здійснювати в два етапи. На першому етапі проводиться приблизне визначення коріння. Звичайно це можна здійснити графічним способом, а на другому етапі проводиться уточнення значення коріння. При цьому вважаємо, що трансцендентне рівняння або рівняння на деякому відрізку $[a, b]$ відповідають теоремі про існування кореня безперервної функції: якщо функція $f(x)$ безперервна на відрізку $[a, b]$, її похідна $f'(x)$ в $]a, b[$ зберігає знак, на кінцях відрізка функція приймає значення з протилежними знаками $f(a)f(b) < 0$, то на цьому відрізку існує корінь рівняння $f(x)=0$ (принаймні, хоч би один).

На рис.4.2.1а представлені графічні залежності двох функцій

$$y = ax,$$

$$y = e^{\arccos\left(\frac{1}{x}\right)},$$

трансцендентного рівняння

$$ax = e^{\arccos\left(\frac{1}{x}\right)}. \quad (4.2.12)$$

Таку форму рівнянь отримуємо при вирішенні деяких завдань теорії механізмів і машин, механіки нитки [1,16]. Рівняння прямих побудовані для різних значень коефіцієнта $a(0 < a < 2)$. Точки перетину і відповідатимуть вирішенню рівняння (4.2.12). Причому, на вибраному інтервалі зміни x , для різних значень коефіцієнтів a , може бути від 1 до 2 коріння.

При визначенні величини кута прибою точною нитки на ткацьких верстатах приходимо до системи двох трансцендентних рівнянь

$$bx^2 - x^4 - y^a = 0,$$

$$a_1x - b_1 - y = 0. \quad (4.2.13)$$

На рис.4.2.1б представлено графічне (наближене) визначення коріння системи (4.2.13) для різних значень a, b, a_1, b_1 . Їх величина залежить від ряду технологічних параметрів і конструктивних особливостей ткацького верстака [17].

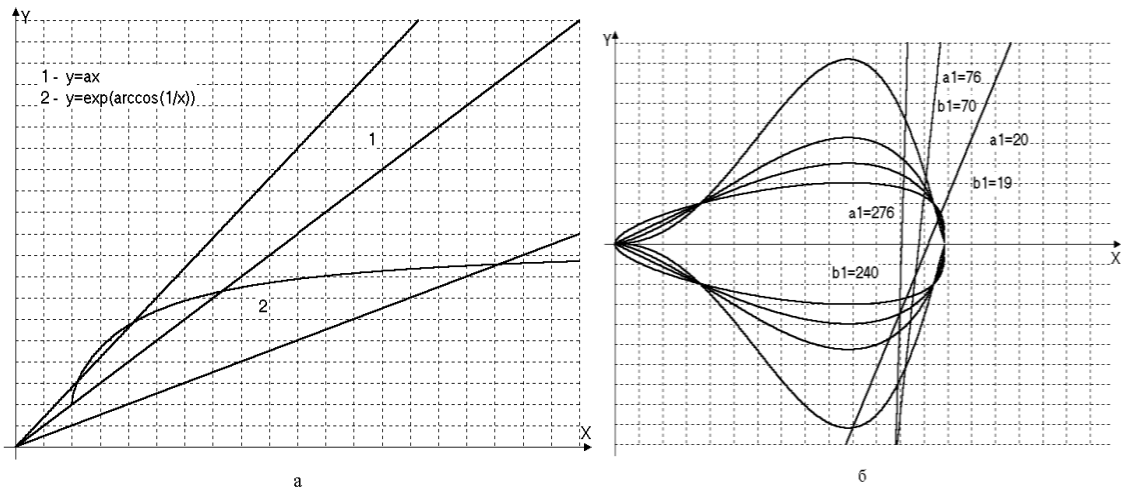


Рис.4.2.1. Графічне визначення коріння трансцендентного рівняння $f(x)=0$

Якщо в процесі досліджень приходимо до систем, що містять більше двох трансцендентних рівнянь, то графічне дослідження стає скрутним, а іноді і неможливим.

Необхідно відзначити, що вимога безперервності функції $f(x)$ на відріжку $[a, b]$ істотно. Непорушною вимогою є відсутність точок розриву.

Сформульована вище теорема існування може доводитися двома способами: конструктивно з використанням методу фактичної побудови шуканого рішення; неконструктивний, ґрунтуючись на міркуваннях від осоружного - ланцюг логічних висновків показує, що вирішення трансцендентного рівняння повинне існувати інакше виникне суперечність.

Перейдемо до опису методів уточнення коріння трансцендентних рівнянь. Першим з них є метод ітерацій (послідовних наближень) [10,14].

Трансцендентне рівняння

$$f(x) = 0, \quad (4.2.14)$$

представимо до вигляду

$$x = f_1(x), \quad (4.2.15)$$

де $f_1(x) = f(x) + x$.

Скориставшись графічним способом, визначимо наближене значення кореня x_0 з області визначення функції $f_1(x)$ і підставимо його в праву частину рівняння (4.2.15). Будуватимемо послідовність чисел $\{x_n\}$, визначених за допомогою ітераційної формули

$$x_{n+1} = f_1(x_n), \quad n = 0, 1, 2, \dots$$

Послідовність чисел $\{x_n\}$ носить назву ітераційної послідовності. Якщо існує $\lim_{n \rightarrow \infty} x_n = \xi$, то переходячи в рівності (4.2.15) до межі і рахуючи функцію $f_1(x)$ безперервною, матимемо

$$\lim_{n \rightarrow \infty} x_{n+1} = f_1\left(\lim_{n \rightarrow \infty} x_n\right),$$

або

$$\xi = f_1(\xi).$$

З останньої рівності виходить, що ξ буде коренем трансцендентного рівняння (4.2.14). Ітераційний процес продовжується до тих пір, поки дотримується умова

$$|x_{n+1} - x_n| \geq \varepsilon, \quad (4.2.16)$$

де ε - задана погрішність обчислення кореня ξ .

Перед доведенням теореми про збіжність ітераційної послідовності зупинимося на умові Лібшиця, яка свідчить, що функція $f(x)$ задовольняє йому, якщо існує така постійна $q < 1$, що для будь-яких x_1, x_2 , що належать відрізка $[a, b]$, виконується нерівність

$$\left| \frac{f(x_1) - f(x_2)}{x_1 - x_2} \right| \leq q. \quad (4.2.17)$$

Якщо функція (4.2.14) задовольняє умові (4.2.17), то вона є безперервною на відрізка $[a, b]$. Повідомимо аргументу x приріст Δx . Для початкового наближення визначимо приріст функції (4.2.14)

$$\Delta f = f(x_0 + \Delta x) - f(x_0),$$

звідки, з використанням умови Ліпшиця, отримаємо підтвердження безперервності функції (4.2.14) на відрізку $[a, b]$

$$|\Delta f| \leq \alpha |\Delta x|,$$

$$\lim_{\Delta x \rightarrow 0} \Delta f = 0.$$

Теорема про збіжність ітераційної послідовності може бути сформульована таким чином. Допустимо, що функція $f_1(x)$ визначена і диференціюєма на відрізку $[a, b]$, причому всі її значення $f_1(x) \in [a, b]$.

Тоді з використанням умови Ліпшиця $\left| \frac{df_1(x)}{dx} \right| \leq q < 1$ (при $a < x < b$) отримаємо, що процес ітерації $x_{n+1} = f_1(x_n)$ сходиться незалежно від початкового значення $x_0 \in [a, b]$ і граничне значення $\xi = \lim_{n \rightarrow \infty} x_n$ є єдиним коренем рівняння $x = f_1(x)$ на відрізку $[a, b]$.

Доведемо це твердження. Візьмемо початкове наближення вирішення трансцендентного рівняння (4.2.14) x_0 на відрізку $[\xi - \delta, \xi + \delta]$, яке розташоване від точки ξ на відстані не більше ніж δ ($|\xi - x_0| \leq \delta$). Виконаємо ітераційний процес з використанням умови Ліпшиця і з обліком (4.2.15)

$$x_1 = f_1(x_0), \quad x_1 - \xi = f_1(x_0) - f_1(\xi),$$

$$|x_1 - \xi| = |f_1(x_0) - f_1(\xi)| \leq q |x_0 - \xi| \leq q \delta,$$

$$|x_2 - \xi| = |f_1(x_2) - f_1(\xi)| \leq q |x_1 - \xi| \leq q^2 |x_0 - \xi| \leq q^2 \delta,$$

.....

$$|x_n - \xi| \leq q^n |x_0 - \xi| \leq q^n \delta.$$

З (4.2.18) витікає, що

$$\lim_{n \rightarrow \infty} (x_n - \xi) = 0 \Rightarrow \lim_{n \rightarrow \infty} x_n = \xi.$$

Теорема залишається правильною, якщо функція $f_1(x)$ визначена і диференціюєма на інтервалі $]-\infty, +\infty[$. Для оцінки наближення скористаємося нерівністю

$$|\xi - x_{n+1}| \leq \frac{q}{1-q} |x_{n+1} - x_n|.$$

Коли $q < \frac{1}{2}$ то приходимо до (4.2.16). Остаточно

$$\xi = x_{n+1} \pm \varepsilon.$$

Програма *TR_U_M_PI*, що реалізує метод простих ітерацій, приведена в Додатку.

Перейдемо до визначення коріння трансцендентного рівняння (4.2.14) з використанням методу Ньютона (або метод дотичних). На рис.4.2.2 показана графічна схема, що реалізує метод Ньютона.

За допомогою графічного методу визначаємо початкове наближення кореня x_0 трансцендентного рівняння $f(x)=0$. Рівняння дотичної до графіка функції $f(x)$ з координатою x_0 має вигляд

$$f_K = f(x_0) + f'(x_0)(x - x_0), \quad (4.2.19)$$

де $f'(x_0)$ - значення похідної функції $f(x)$ в точці x_0 .

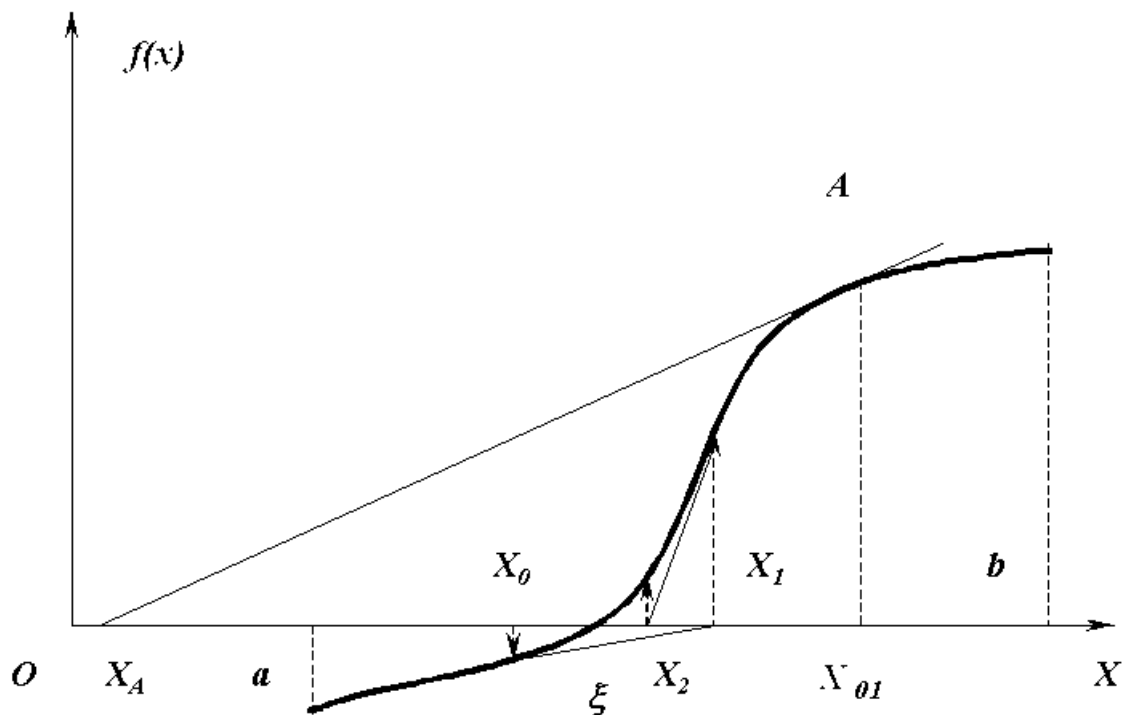


Рис.4.2.2. Вирішення трансцендентного рівняння методом Ньютона (дотичних)

При $x = x_1$, $f_K = 0$. Тоді з (4.2.19) отримаємо

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Продовжуючи процес побудови ітераційної послідовності $\{x_n\}$, отримуємо наступну рекурентну формулу для здійснення ітераційного процесу сходження до кореню трансцендентного рівняння

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots \quad (4.2.20)$$

Метод Ньютона, що реалізовується при допомозі (4.2.20), володіє високою швидкістю збіжності, проте він дуже чутливий до вибору початкового наближення x_0 . На рис.4.2.2 показано, що вибір як початкове наближення x_{01} (точка A на $f(x)=0$, яка розташовується далі від шуканого кореня ξ ніж x_0) приводить вже на першому кроці до випадання точки x_A з меж відрізка $[a, b]$. Це обриває процес побудови ітераційної послідовності $\{x_n\}$. Даний метод забезпечує швидку (квадратичну) збіжність якщо

$$f(x_0)f''(x_0) > 0.$$

У якості x_0 вибирають те значення з $[a, b]$, де знаки $f(x_0)$ і $f''(x_0)$ співпадають.

Коли $|f'(x)| \geq m > 0$, $|f''(x)| \leq M$, $x \in [a, b]$ (m – найменше значення похідної $f'(x)$ в $[a, b]$; M – найбільше значення похідної $f''(x)$ в $[a, b]$), то знайдеться таке $\delta : 0 < \delta \leq \min(\xi - a, b - \xi)$, що при будь-якому виборі початкового наближення на відрізку $[\xi - \delta, \xi + \delta] \in [a, b]$ існує нескінченна ітераційна послідовність (4.2.20) і ця послідовність сходиться до кореню ξ трансцендентного рівняння (4.2.14). Дане твердження формалізує теорему про збіжність методу Ньютона. Її доказ побудований з використанням основного рекурентного співвідношення методу простих ітерацій (4.2.15)

$$x = f_1(x), \quad f_1(x) = x - \frac{f(x)}{f'(x)},$$

$$f_1'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}, \quad \lim_{x \rightarrow \xi} f(x) = f(\xi) = 0, \quad (4.2.21)$$

$$|f_1'(x)| \leq \frac{M}{m^2} |f(x)|, \quad |f(x) - f(\xi)| = |f(x)| \leq \frac{m^2}{2M},$$

$$|f_1'(x)| \leq q = 0,5.$$

Програма *TR_U_M_NJ*, що реалізовує метод Ньютона (дотичних), приведена в Додатку.

Для оцінки погрішності n -го наближення x_n можна використовувати формулу

$$|\xi - x_n| \leq \frac{|f(x_n)|}{m_1}, \quad (4.2.22)$$

де m_1 – найменше значення модуля першої похідної $|f'(x)|$ на відріжку $[a, b]$.

Одним з недоліків методу Ньютона є необхідність обчислення похідної $f'(x)$. Для подолання цього використовується модифікований метод Ньютона, який полягає в тому, що замість обчислення похідної $f'(x)$, на кожному кроці ітерації, знаходиться її наближене значення

$$\frac{df(x)}{dx} \cong \frac{f(x_n + \Delta x) - f(x_n)}{\Delta x} = \frac{\Delta f(x_n)}{\Delta x}, \quad (4.2.23)$$

$$\Delta x = \varepsilon.$$

Тоді з обліком (4.2.23) рівняння (4.2.20) прийме вигляд

$$x_{n+1} = x_n - \frac{\Delta x f(x_n)}{f(x_n + \Delta x) - f(x_n)}. \quad (4.2.24)$$

Рекурентну формулу (4.2.24) використовують для побудови ітераційної послідовності $\{x_n\}$ для реалізації модифікованого методу Ньютона.

Програма *TR_U_M_N_M*, для реалізації модифікованого методу Ньютона для знаходження коріння трансцендентного рівняння, приведена в Додатку.

Перейдемо до розгляду методу ділення відрізка $[a, b]$ навпіл (метод дихотомії). Суть методу полягає в побудові ітераційної послідовності вкладених один в одного відрізків $[a_n, b_n]$, кінці яких представляють монотонні послідовності $\{a_n\}, \{b_n\}$, причому

$$a_n \leq \xi, \quad b_n \geq \xi, \quad n = 1, 2, \dots$$

де ξ - корінь трансцендентного рівняння (4.2.14) на відрізку $[a, b]$.

Збіжність даного методу повільна. Проте при будь-якій ширині відрізання $[a, b]$ збіжність гарантована.

Вважатимемо, що $f(a)f(b) < 0$ и $f(a) < 0, f(b) > 0$. Тоді візьмемо середню точку відрізка $[a, b]$

$$\xi_1 = \frac{a+b}{2}. \quad (4.2.25)$$

Обчислюємо в даній крапці значення функції $f(x)$. Припустимо, що $f(\xi_1) = 0$. В цьому випадку процес визначення кореня трансцендентного рівняння закінчується. Якщо $f(\xi_1) \neq 0$, то розглядаємо два відрізка $[a, \xi_1]$ и $[\xi_1, b]$. Вибираємо той з них, де виконується умова $f(a)f(\xi_1) < 0$ або $f(\xi_1)f(b) < 0$. Вибраний відрізок знову поділяємо навпіл приймаючи

$$a_1 = a, \quad b_1 = \xi_1 \quad \text{або} \quad a_1 = \xi_1, \quad b_1 = b$$

$$\xi_2 = \frac{a_1 + b_1}{2}. \quad (4.2.26)$$

Необмежене продовження процесу ділення дозволяє отримати ітераційну послідовність вкладених один в одного відрізків $[a_n, b_n]$, причому

$$a_n \leq a_{n+1} < b_{n+1} \leq b_n. \quad (4.2.27)$$

Ліві кінці відрізків утворюють ітераційну, монотонну послідовність $\{a_n\}$, яка в межі представлятиме деяку величину z_1

$$\lim_{n \rightarrow \infty} a_n = z_1, \quad (4.2.28)$$

а праві кінці відрізків утворюють ітераційну, монотонну послідовність $\{b_n\}$, яка в межі представлятиме деяку величину z_2

$$\lim_{n \rightarrow \infty} b_n = z_2. \quad (4.2.29)$$

Очевидно, що

$$\begin{aligned} a_n &\leq z_1 \leq z_2 \leq b_n, \\ z_2 - z_1 &\leq b_n - a_n = \frac{b-a}{2^n}. \end{aligned} \quad (4.2.30)$$

З останнього рівняння системи (4.2.30) отримуємо $z_2 - z_1 = \theta$, т.ч. дана різниця менше будь-якого наперед заданого позитивного числа. Звідки матимемо $z_2 = z_1 = \xi$. Отже

$$f(\xi) = \lim_{n \rightarrow \infty} f(a_n) = \lim_{n \rightarrow \infty} f(b_n).$$

З обліком (4.2.30)

$$a_n \leq \xi \leq b_n.$$

Аналіз останньої нерівності показує, що величина a_n визначає шуканий корінь ξ з недоліком, а величина b_n з лишком. Помилка при цьому не перевищує довжину відрізання $b_n - a_n$ і прагне до нуля при збільшенні n за законом геометричної прогресії із знаменником $1/2$. Число ітерацій N , необхідне для досягнення заданої точності ε визначається з виразу

$$N \cong \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2}.$$

Програма **TR_U_M_DI**, для визначення кореня трансцендентного рівняння $f(x) = 0$ на відрізку $[a, b]$ методом дихотомії, приведена в Додатку.

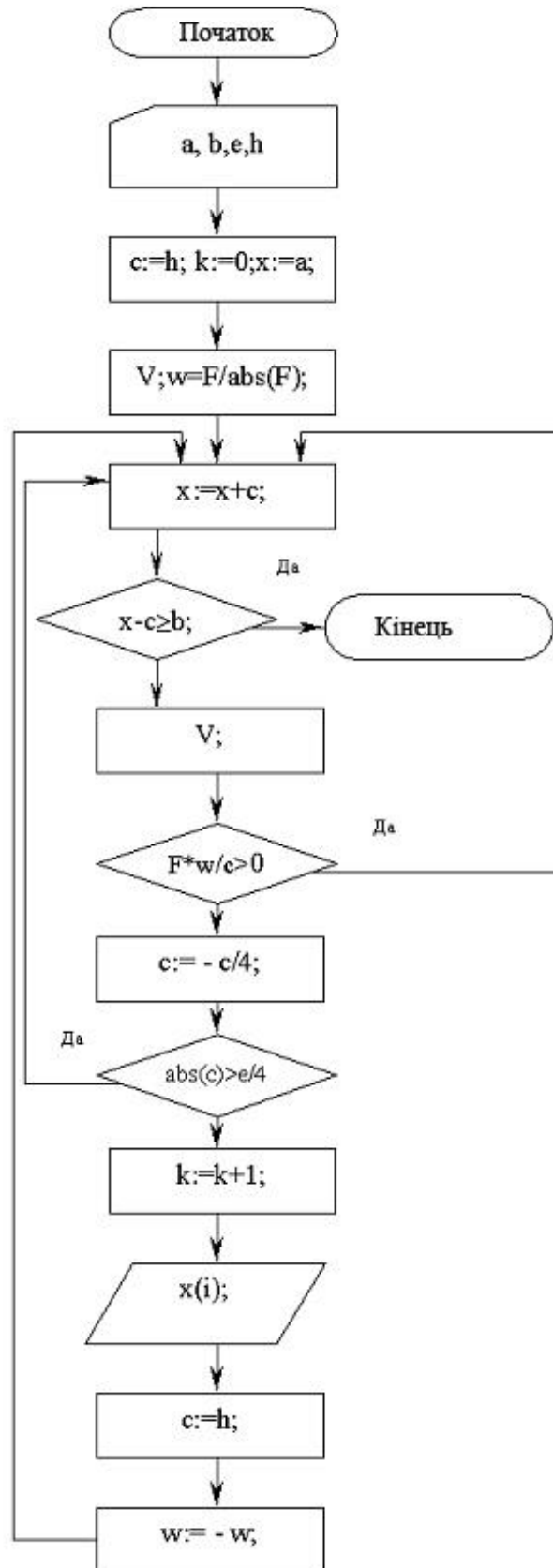


Рис.4.2.3. Блок-схема алгоритму для реалізації методу порозрядного наближення

Представляє інтерес метод порозрядного наближення для обчислення всього існуючого коріння трансцендентного рівняння на відрізку $[a, b]$ із заданою погрішністю ε . На рис.4.2.3 представлена блок-схема алгоритму для реалізації вказаного методу.

Особливість полягає в зміні величини кроку 3 шляхом введення показника розрядності R (у нашому випадку приймали $R = 4$).

На початковому етапі задається крок і початок межі відрізання a . Потім виконуємо операцію привласнення $x = x + c$ і порівнюємо отриманий результат із значенням b , що визначає кінець відрізання. Далі обчислюємо значення функції (4.2.14) і перевіряємо умову

$$\frac{f(x) \operatorname{sign}[f(x)]}{C} \geq 0. \quad (4.2.31)$$

На наступному етапі зменшуємо крок $C = -\frac{C}{R}$ і перевіряємо виконання умови $|C| \geq \frac{\varepsilon}{R}$, де ε - задана погрішність обчислення кореня. Якщо умова не виконується, виводимо значення кореня на друк.

Програма *TR_U_M_PP*, для визначення всього коріння трансцендентного рівняння методом порозрядного наближення на інтервалі $[a, b]$, приведена в Додатку.

Для вирішення системи n трансцендентних рівнянь

$$\begin{aligned} f_i(x_1, \dots, x_i) &= 0, \\ i &= 1, 2, \dots, n, \end{aligned} \quad (4.2.32)$$

скористаємося модифікованим методом Ньютона. Кожне з трансцендентних рівнянь розкладаємо в ряд Тейлор і формуємо матрицю Якобі для розрахунку приростів $f_i(x_i)$ при малій зміні змінних, замінюючи приватні похідні в матриці Якобі їх наближеними кінцево-різницевиими значеннями

$$\frac{\partial f_i}{\partial x_i} \cong \frac{f_i(x_i + \varepsilon |x_i|) - f_i(x_i)}{\varepsilon |x_i|}, \quad (4.2.33)$$

де ε - відносна погрішність обчислення.

Матриця Якобі в розгорненому вигляді, з обліком (4.2.33), може бути представлена так

$$\begin{aligned} & \left\| \begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{array} \right\| \cong \\ & \cong \left\| \begin{array}{ccc} \frac{f_1(x_1 + \varepsilon|x_1|) - f_1(x_1)}{\varepsilon|x_1|} & \dots & \frac{f_1(x_n + \varepsilon|x_n|) - f_1(x_n)}{\varepsilon|x_n|} \\ \frac{f_n(x_1 + \varepsilon|x_1|) - f_n(x_1)}{\varepsilon|x_1|} & \dots & \frac{f_n(x_n + \varepsilon|x_n|) - f_n(x_n)}{\varepsilon|x_n|} \end{array} \right\|. \end{aligned} \quad (4.2.34)$$

Далі складаємо і вирішуємо систему лінійних рівнянь для малих приростів x_i

$$\begin{aligned} & \left\| \begin{array}{ccc} \frac{f_1(x_1 + \varepsilon|x_1|) - f_1(x_1)}{\varepsilon|x_1|} & \dots & \frac{f_1(x_n + \varepsilon|x_n|) - f_1(x_n)}{\varepsilon|x_n|} \\ \frac{f_n(x_1 + \varepsilon|x_1|) - f_n(x_1)}{\varepsilon|x_1|} & \dots & \frac{f_n(x_n + \varepsilon|x_n|) - f_n(x_n)}{\varepsilon|x_n|} \end{array} \right\| \cdot \left\| \begin{array}{c} \Delta x_1 \\ \dots \\ \Delta x_n \end{array} \right\| = \\ & = \left\| \begin{array}{c} -f_1(x_1 \dots x_n) \\ \dots \\ -f_n(x_1 \dots x_n) \end{array} \right\|. \end{aligned} \quad (4.2.35)$$

Визначені з системи (4.2.35) значення приростів використовуємо на наступному кроці $x_{n(i+1)} = x_{n(i)} + \Delta x_n$. Далі здійснюємо перевірку з використанням заданої величини відносної погрішності ε .

Програма `SU_TR_U_M_N_M`, для вирішення системи трансцендентних рівнянь модифікованим методом Ньютона, приведена в Додатку.

Часто при вирішенні прикладних завдань стикаємося з необхідністю визначення всього коріння полінома з дійсними коефіцієнтами

$$f(x) = x^n + a_{n+1} + \sum_{i=2}^n a_i x^{n-i+1}, \quad (4.2.36)$$

де n - ступінь полінома; a_i - дійсні числа.

Для вирішення даного завдання скористаємося схемою Горнера [14]. На початковому етапі виділяємо квадратичний тричлен і приводимо його до вигляду

$$x^2 + px + q, \quad (4.2.37)$$

де p, q – коефіцієнти квадратного трьохчлена.

Якщо обчислення коріння полінома (4.2.36) починається з найбільших по абсолютній величині значень, то беремо трьох перших членів і

$$p = a_2, \quad q = a_3, \quad (4.2.38)$$

а якщо обчислення коріння починаємо з найменших по абсолютній величині значень, то беремо трьох останніх членів і

$$p = \frac{a_n}{a_{n-1}}, \quad q = \frac{a_{n+1}}{a_{n-1}}. \quad (4.2.39)$$

Схема Горнера полягає в тому, що якщо коріння (4.2.37) є корінням полінома (4.2.36), то останній згідно теоремі Безу ділиться на (4.2.37) без залишку. Пропонований метод використовувався ще математиками в середньовічному Китаї. На початку XIX століття він був «відкритий незалежно У.Горнером (1819) і П.Руффіні (1802)» [14].

Після виконання операції ділення (4.2.36) на (4.2.37) отриманий поліном матиме ступінь $n-2$ і з ним виконуються операції описані вище. Таким чином, вираз (4.2.36) буде приведений до вигляду: для випадку коли n - парне

$$f(x) = \prod_{i=1}^{\left[\frac{n}{2}\right]} (x^2 + c_{2i}x + c_{2i+1}), \quad (4.4.20)$$

для випадку коли n - непарне

$$f(x) = \prod_{i=1}^{\left[\frac{n}{2}\right]} (x^2 + c_{2i}x + c_{2i+1})(x + c_{n+1}), \quad (4.4.21)$$

де c_{ij} - дійсні коефіцієнти.

Ітераційний процес визначення коріння на кожному подальшому етапі пониження ступеня полінома (4.2.36) проводиться по наступному алгоритму

$$\begin{aligned}
 f(x) &= V(x)(x^2 + px + q) = 0, \\
 V(x) &= x^{n-2} + b_2x^{n-3} + \dots + b_{n-2}x + b_{n-1}, \\
 b_1 &= 1, \quad b_2 = a_2 - p, \quad b_i = a_i - pb_{i-1} - qb_{i-2}, \quad i = 3 \dots n+1, \\
 c_1 &= 1, \quad c_2 = b_2 - p, \quad c_j = b_j - pc_{j-1} - qc_{j-2}, \quad j = 3 \dots n-1, \\
 c_n &= -pc_{n-1} - qc_{n-2}, \quad d = c_{n-1}^2 - c_n c_{n-2}, \\
 \Delta p &= \frac{b_n c_{n-1} - b_{n+1} c_{n-2}}{d}, \quad \Delta q = \frac{b_{n+1} c_{n-1} - b_n c_n}{d},
 \end{aligned} \tag{4.4.22}$$

$$p = p + \Delta p, \quad q = q + \Delta q, \quad |\Delta p| \leq \varepsilon, \quad |\Delta q| \leq \varepsilon,$$

де ε - погрішність визначення коріння.

Обчислення по (4.4.22) проводяться поки $n > 2$. Програма *KOR_POLIN*, для обчислення всього коріння полінома з дійсними коефіцієнтами, приведена в Додатку.

4.3. ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ

У додатках математики до технічних наук диференціальні рівняння займають особливо важливе місце. Багато прикладних процесів з їх допомогою описуються простішим і повнішим. Вони дають можливість вирішувати багато питань загально технічних і спеціальних прикладних дисциплін: фізики, теоретичної механіки, опору матеріалів, гідравліки, теорії механізмів і машин, хімії, технології виробництва, біології, фінансово-економічних дисциплін і ін.

Відомо, що далеко не завжди вдається в замкнутому вигляді знайти вирішення диференціального рівняння. Тому розроблені методи для їх наближеного вирішення. Питання чисельного вирішення диференціальних рівнянь як звичайних, так і в приватних похідних вже багато років

привертають увагу учених, проте питання про якнайкращий з методів чисельного рішення наукою поки не дозволене.

Залежно від характеру граничних умов диференціальні рівняння можна розбити на три класи: одно точечні; дво точкові і багато точкові. До першого класу відносяться звичайні диференціальні рівняння, всі граничні умови яких повинні виконуватися в одній крапці. До другого класу відносяться диференціальні рівняння, у яких граничні або початкові умови задані в двох крапках. Багато точкові граничні умови мають місце у диференціальних рівнянь порядку вище другого.

Вирішення звичайних диференціальних рівнянь з одно точечними граничними умовами приводить до покрокового завдання чисельної інтеграції.

У загальному випадку, в рамках завдання Коші (знайти таку безперервну функцію $y = y(x)$ у області $[x_0, x]$, яка задовольняє диференціальному рівнянню $\frac{dy}{dx} = f(x, y)$), ми розглядатимемо чисельну інтеграцію системи звичайних диференціальних рівнянь першого порядку

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(x, y_1, y_2, \dots, y_j, \dots, y_n), \\ \frac{dy_2}{dx} &= f_2(x, y_1, y_2, \dots, y_j, \dots, y_n), \\ & \dots \dots \dots \dots \dots \dots \dots \\ \frac{dy_j}{dx} &= f_j(x, y_1, y_2, \dots, y_j, \dots, y_n), \\ & \dots \dots \dots \dots \dots \dots \dots \\ \frac{dy_n}{dx} &= f_n(x, y_1, y_2, \dots, y_j, \dots, y_n), \end{aligned} \tag{4.3.1}$$

де x - незалежна змінна (час, переміщення, температура і ін.); y_j - залежна змінна; $j = 1 \dots n$; n - число диференціальних рівнянь, що входять в систему (4.3.1).

значень функції будуються нові, які потім уточнюються за допомогою самих диференціальних рівнянь.

До перших можна віднести методи Ейлера - Коші, метод трапецій, метод Рунге - Кута. До других відносяться метод Адамса, метод Крилова-Адамса [10].

Перейдемо до розгляду методу Ейлера - Коші. На рис.4.3.1 показана розрахункова схема.

Хай задано диференціальне рівняння

$$\frac{dy}{dx} = f(x, y). \quad (4.3.7)$$

Необхідно знайти наближене рішення (4.3.7) в крапках з координатами

$x_1 = x_0 + h$, $x_2 = x_0 + 2h, \dots, x_n = x_0 + nh$, де h - постійний крок; x_0 - координата початку відрізка.

Початкова умова $x = x_0$, $y_0 = y(x_0)$. Наближене значення першої похідної має вигляд

$$\frac{dy_k}{dx_k} \cong \frac{\Delta y_k}{\Delta x_k} = \frac{y_{k+1} - y_k}{h}, \quad (4.3.8)$$

де $k = 0, 1, \dots, n-1$.

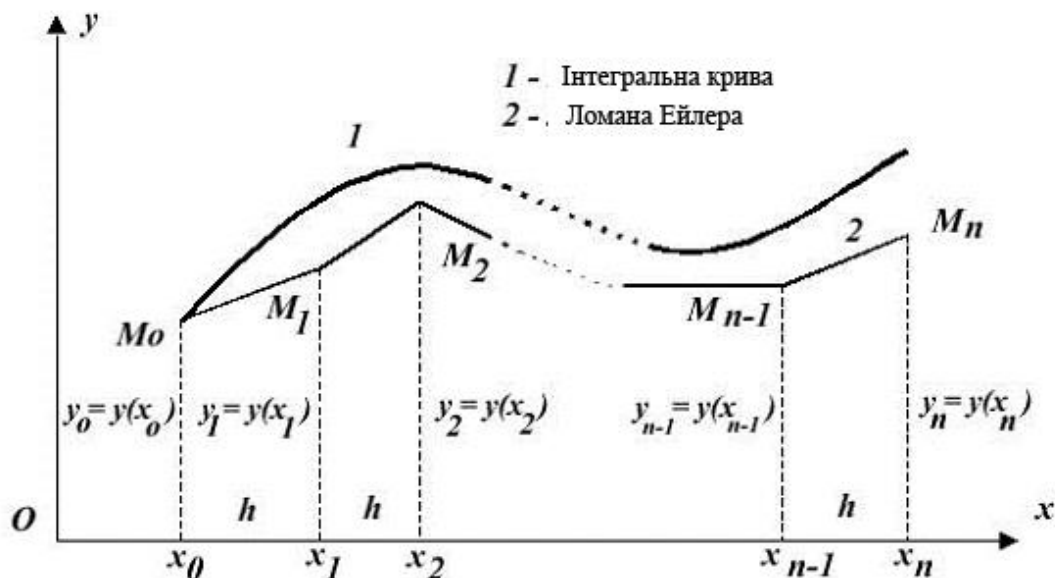


Рис.4.3.1. Розрахункова схема методу Ейлера-Коші
Прирівнюючи (4.3.7) і (4.3.8), отримаємо

$$\frac{y_{k+1} - y_k}{h} = f(x_k, y_k),$$

звідки

$$y_{k+1} = y_k + hf(x_k, y_k). \quad (4.3.9)$$

Використовуючи рекурентну формулу (4.3.9) для точок $k = 0, 1, \dots, n-1$ будуємо ламану Ейлера 2, яка приблизно замінює інтегральну криву 1 (см.рис.4.3.1). Суть методу Ейлера-Коші полягає в тому, що через початок кожного відрізка $[x_k, x_{k+1}]$ проводиться дотична до інтегральної кривої 1.

Точність методу Ейлера-Коші невелика. Погрішність методу пропорційна h^2 .

Різновидом методу Ейлера-Коші є метод трапецій. Він реалізується застосуванням на кожному кроці рекурентної формули

$$y_{k+1} = y_k + \frac{h}{2} \left\{ f(x_k, y_k) + f\left[x_k + h, y_k + hf(x_k, y_k)\right] \right\}. \quad (4.3.10)$$

Погрішність методу трапецій пропорційна h^3 і його також відносять до загальних методів Рунге - Кутта. Програма *PROGRAM DIF_UR_SIS_M_T*, для чисельної інтеграції звичайного диференціального рівняння методом трапецій, приведена в Додатку.

Перейдемо до розгляду методу Рунге-Кутта. Рекурентна формула для його реалізації має вигляд

$$y_{k+1} = y_k + hf\left[x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)\right]. \quad (4.3.11)$$

Графічна інтерпретація даного методу показана на рис.4.3.2. Його суть полягає в тому, що на початковому етапі через точку M_k проводять лінію поля $f(x_k, y_k)$. Далі обчислюють координати точки M_{kc} , яка ділить відрізок $[x_k, x_{k+1}]$ навпіл.

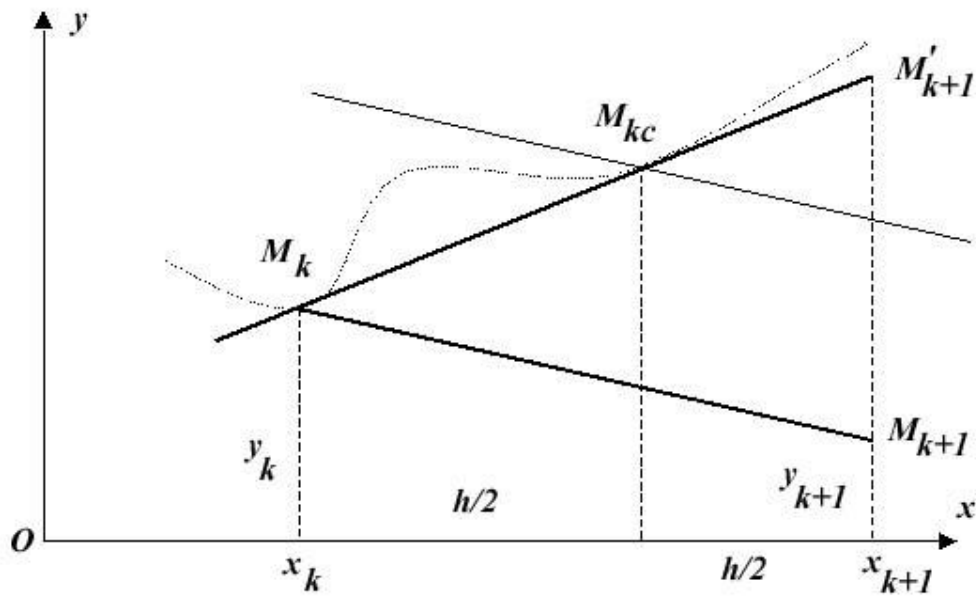


Рис.4.3.2. Графічна інтерпретація методу Рунге-Кутта

Його координати будуть рівні $x_k + \frac{h}{2}$, $y_k + \frac{h}{2} f(x_k, y_k)$. Лінія поля $f\left[x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)\right]$ матиме орієнтацію, відмінну від попередньої за рахунок повороту поля на інтервалі $]x_k, x_{k+1}[$. Потім через точку M_k проводимо пряму, паралельну лінії нового напрямку поля в точці M_{kc} . Ордината точки M_{k+1} і буде шуканою ординатою y_{k+1} .

Різновидом даного методу є метод Рунге-Кутта четвертого порядку, рекурентна формула якого має вигляд

$$a_k = f\left[x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)\right],$$

$$b_k = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2} a_k\right),$$

$$(4.3.12)$$

$$c_k = f(x_k + h, y_k + b_k h),$$

$$y_{k+1} = y_k + \frac{h}{6} [f(x_k, y_k) + 2a_k + 2b_k + c_k].$$

Погрішність даного методу пропорційна h^5 . Він володіє більшою стійкістю до виникнення нестійкості рішення (див. Додаток – **PROGRAM DIF_UR_SIS_M_RK4**).

Можливість забезпечувати наближену оцінку погрішності на кожному кроці інтеграції дозволяє метод Рунге-Кутта-Мерсона з автоматичною зміною кроку. Програма **PROGRAM DIF_UR_SIS_M_RKM_AV**, що дозволяє здійснити реалізацію даного методу, приведена в Додатку.

Рекурентна система рівнянь має вигляд

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{6} [f(x_k, y_k) + 4V_{3k} + V_{4k}], \\ V_{1k} &= hf \left[x_k + \frac{1}{3}h, y_k + \frac{1}{3}hf(x_k, y_k) \right], \\ V_{2k} &= hf \left[x_k + \frac{1}{3}h, y_k + \frac{h}{6}f(x_k, y_k) + \frac{1}{6}V_{1k} \right], \\ V_{3k} &= hf \left[x_k + \frac{1}{2}h, y_k + \frac{h}{8}f(x_k, y_k) + \frac{3}{8}V_{2k} \right], \\ V_{4k} &= hf \left[x_k + h, y_k + \frac{h}{2}f(x_k, y_k) - \frac{3}{2}V_{2k} + 2V_{3k} \right]. \end{aligned} \quad (4.3.13)$$

Погрішність обчислення y_{k+1} , з обліком (4.3.13), визначається по формулі

$$R_{k+1} = \frac{-2hf(x_k, y_k) + 9V_{2k} - 8V_{3k} + V_{4k}}{30}. \quad (4.3.14)$$

Порівнюємо отриману погрішність R_{k+1} із заданою погрішністю E

$$|R_{k+1}| \leq E, \quad |R_{k+1}| \geq \frac{E}{30}. \quad (4.3.15)$$

Якщо перша умова (4.3.15) не виконується, то крок h зменшується удвічі. Якщо друга умова не виконується, то крок збільшується удвічі. Значення y_{k+1} виводиться на друк, тільки при виконанні одночасно двох умов (4.3.15). Погрішність чисельної інтеграції даного методу має порядок h^5 .

Багатокрокові вирішення диференціальних рівнянь (кінцево-різницеві методи) базуються на використанні даних рішення на попередніх кроках. Це дозволяє підвищити швидкість обчислення. Для реалізації кінцево-різницевих методів при чисельному інтегруванні диференціальних рівнянь потрібно знати значення функції і її похідної в декількох точках близьких

до початкової. Тут можна відзначити метод Пікара і метод розкладання. Перший метод заснований на апriorному виборі першого наближення поблизу граничної точки (x_0, y_0)

$$y = d_1(x). \quad (4.3.16)$$

Для (4.3.7), з обліком (4.3.16), отримаємо

$$\begin{aligned} y &= y_0 + \int_{x_0}^x f[x, d_1(x)] dx = d_2(x), \\ y &= y_0 + \int_{x_0}^x f[x, d_2(x)] dx = d_3(x), \\ &\dots\dots\dots \\ y &= y_0 + \int_{x_0}^x f[x, d_{n-1}(x)] dx = d_n(x). \end{aligned} \quad (4.3.17)$$

Метод Пікара поблизу граничної крапки при збіжності дає необхідну точність рішення, проте область його застосування обмежена. Це пояснюється необхідністю інтеграції (4.3.7) в кінцевому вигляді.

Другий метод відшукування наближеного рішення поблизу граничної крапки (x_0, y_0) пов'язаний з розкладанням в ряд Тейлора

$$\begin{aligned} y &= y_0 + \frac{(x-x_0)}{1!} d'(x_0) + \frac{(x-x_0)^2}{2!} d''(x_0) + \dots + \\ &\quad + \frac{(x-x_0)^n}{n!} d^n(x_0) + \dots, \\ & y = d(x). \end{aligned} \quad (4.3.18)$$

Вважаючи, що

$$d'(x) = f(x, y) = \frac{dy}{dx}, \quad (4.3.19)$$

продиференціюємо останню рівність з урахуванням значень в граничній точці

$$\frac{d^2 y}{dx^2} = \frac{\partial f(x, y)}{\partial x} + \frac{\partial f(x, y)}{\partial y} \frac{dy}{dx},$$

або, з обліком (4.3.19), отримаємо

$$d''(x_0) = \left[\frac{\partial f(x, y)}{\partial x} \right]_{\substack{x=x_0 \\ y=y_0}} + \left[\frac{\partial f(x, y)}{\partial y} \right]_{\substack{x=x_0 \\ y=y_0}} \cdot f(x_0, y_0).$$

Метод Адамса з третіми і вищими різницями реалізується за допомогою наступної рекурентної залежності

$$y_{i+1} = y_i + q_i + \frac{1}{2} \Delta q_{i-1} + \frac{5}{12} \Delta^2 q_{i-2} + \frac{3}{8} \Delta^3 q_{i-3}, \quad i = 3, 4, \dots, \quad (4.3.20)$$

$$q_k = hf(x_k, y_k), \quad \Delta^m q_k = h \Delta^m y'_k,$$

$$k = 0, 1, 2, \dots$$

Для реалізації (4.3.20) при рішенні завдання Коші необхідно визначати на початковому етапі (з використанням методу Пікара або методу розкладання) y_i , де $i = 1 \dots 3$. Дана процедура повторюється при зміні кроку інтеграції.

Враховуючи чисельну нестійкість рішення, складність програмної реалізації багатокрокові методи чисельної інтеграції диференціальних рівнянь рідко використовуються [21].

Приклад № 1. Визначення форми балона.

Прядіння і перемотування пряжі супроводжуються переміщенням нитки. Специфіка даних технологічних процесів приводить до виникнення криволінійної ділянки нитки між нитко спрямовувачем (див. рис.4.3.3) A і бігунком B . Дана ділянка має двояку кривизну. При цьому нитка бере участь в двох рухах: переносному навколо осі OZ і відносному уздовж своєї квазістатичної форми рівноваги. На нитку діятимуть сили опору повітря, сила тяжкості, переносна і кориолісові сили інерції. В рамках справжнього завдання вважатимемо відносну лінійну швидкість і переносну кутову швидкість постійною.

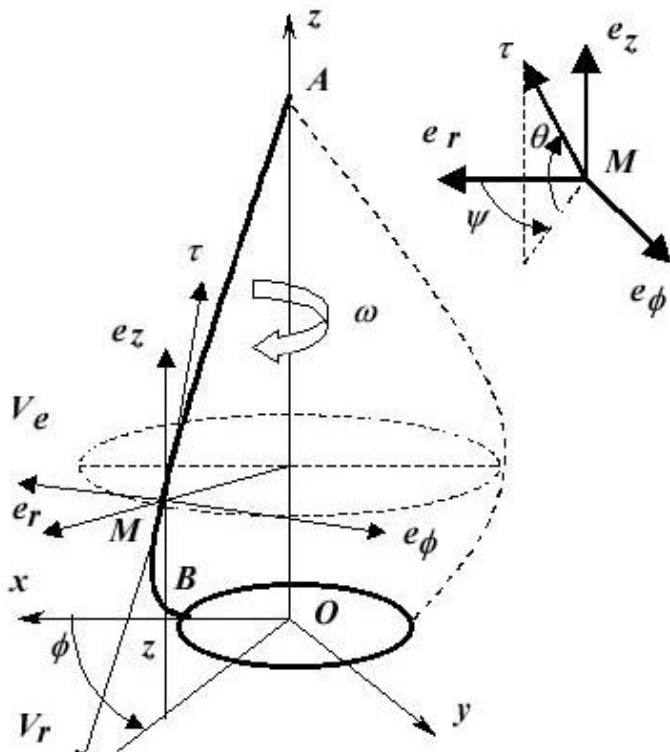


Рис.4.3.3. Розрахункова схема для визначення форми балонуючої нитки

Для визначення форми балона введемо три координатні системи [1,9,20]. Декартову $OXYZ$, циліндрову з одиничними ортами e_r, e_ϕ, e_z і сферичну з одиничними ортами $\cos\theta\cos\psi, \cos\theta\sin\psi, \sin\theta$. Кут θ утворює орт дотичної осі τ в даній точці M з горизонтальною площиною циліндрової координатної системи. Кут ψ утворює проекція орта дотичної на горизонтальну площину

циліндрової координатної системи з одиничним ортом e_r .

Введення сферичної координатної системи дозволяє спростити початкову систему диференціальних рівнянь. Дані координати повинні задовольняти нелінійному неголономному зв'язку

$$\left(\frac{dr}{ds}\right)^2 + r^2\left(\frac{d\phi}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2 = 1,$$

$$\frac{dr}{ds} = \cos\theta\cos\psi,$$

$$r\frac{d\phi}{ds} = \cos\theta\sin\psi,$$

$$\frac{dz}{ds} = \sin\theta,$$
(4.3.21)

де s - дугова координата.

Основна система диференціальних рівнянь рівноваги елементарного відрізка нитки ds у проекціях на осі циліндрової координатної системи має вигляд [1,9,20]

$$\begin{aligned} \frac{d}{ds} \left(P \frac{dr}{ds} \right) - Pr \left(\frac{d\varphi}{ds} \right)^2 + F_r &= 0, \\ \frac{d}{ds} \left(Pr^2 \frac{d\varphi}{ds} \right) + rF_\varphi &= 0, \\ \frac{d}{ds} \left(P \frac{dz}{ds} \right) + F_z &= 0, \end{aligned} \quad (4.3.22)$$

де P - натяг нитки; F_r, F_φ, F_z - проекції активно заданих сил на осі циліндрової координатної системи.

З другого рівняння системи (4.3.21) отримаємо диференціального оператора

$$\frac{d}{ds} = \cos\theta \cos\psi \frac{d}{dr}. \quad (4.3.23)$$

Використовуючи його, перетворимо третє і четверте диференціальні рівняння системи (4.3.21)

$$\begin{aligned} \frac{d\varphi}{dr} &= \frac{\operatorname{tg}\psi}{r}, \quad \frac{dz}{dr} = \frac{\operatorname{tg}\theta}{\cos\psi}, \\ \frac{ds}{dr} &= \frac{1}{\cos\theta \cos\psi}. \end{aligned} \quad (4.3.24)$$

Використовуючи (4.3.23) перетворимо систему диференціальних рівнянь (4.3.21)

$$\begin{aligned} \cos\theta \cos\psi \left[\cos\theta \cos\psi \frac{dP}{dr} + P \frac{d}{dr} (\cos\theta \cos\psi) \right] - \frac{P}{r} \cos^2\theta \sin^2\psi + F_r &= 0, \\ \cos\theta \cos\psi \left[r \cos\theta \sin\psi \frac{dP}{dr} + P \frac{d}{dr} (r \cos\theta \sin\psi) \right] + rF_\varphi &= 0, \\ \cos\theta \cos\psi \left[\sin\theta \frac{dP}{dr} + P \frac{d}{dr} (\sin\theta) \right] + F_z &= 0, \end{aligned} \quad (4.3.25)$$

Помножимо перше рівняння (4.3.25) на $\cos\theta \cos\psi$, друге на $\frac{\cos\theta \sin\psi}{r}$, а третє на $\sin\theta$ і складемо їх [9]

$$\frac{dP}{dr} = - \left(F_r + F_\varphi \operatorname{tg}\psi + F_z \frac{\operatorname{tg}\theta}{\cos\psi} \right). \quad (4.3.26)$$

Перетворимо перші два диференціальні рівняння системи (4.3.25) до вигляду [20]

$$\cos \theta \cos \psi \left[-P \cos \theta \sin \psi \frac{d\psi}{dr} + \cos \psi \frac{d}{dr} (P \cos \theta) \right] - \frac{P}{r} \cos^2 \theta \sin^2 \psi + F_r = 0,$$

$$\cos \theta \cos \psi \left[Pr \cos \theta \cos \psi \frac{d\psi}{dr} + r \sin \psi \frac{d}{dr} (P \cos \theta) + P \cos \theta \sin \psi \right] + r F_\varphi = 0.$$

Помножимо перше з отриманих рівнянь на $-r \sin \psi$, а друге на $\cos \psi$.

Складемо результати, тоді

$$\frac{d\psi}{dr} = \frac{F_r \sin \psi - F_\varphi \cos \psi}{P \cos^2 \theta \cos \psi} - \frac{\text{tg} \psi}{r}. \quad (4.3.27)$$

Перетворимо третє рівняння системи (4.3.25) з обліком (4.3.26) до вигляду

$$\frac{d\theta}{dr} = \frac{\text{tg} \theta}{P} \left(F_r + F_\varphi \text{tg} \psi - \frac{F_z}{\text{tg} \theta \cos \psi} \right). \quad (4.3.28)$$

Рівняння (4.3.24) (4.3.26) (4.3.27).(4.3.28) утворюють систему п'яти диференціальних рівнянь першого порядку. Залишається визначити проєкції F_r, F_φ, F_z активно заданих сил на осі циліндрової координатної системи. Головний вектор зовнішніх сил буде рівний [9,20]

$$\begin{aligned} \vec{F} &= F_r \vec{e}_r + F_\varphi \vec{e}_\varphi + F_z \vec{e}_z, \\ F_r &= \mu \omega^2 r - 2 \mu \omega u \cos \theta \sin \psi - \lambda_2 r^2 \Phi_n(\chi) \text{ctg} \chi \cos \theta \cos \psi, \\ F_\varphi &= 2 \mu \omega u \cos \theta \cos \psi + \lambda_2 r^2 \Phi_n(\chi) \frac{1}{\sin \chi} - \lambda_2 r^2 \Phi_n(\chi) \text{ctg} \chi \cos \theta \sin \psi, \\ F_z &= -q - \lambda_2 r^2 \Phi_n(\chi) \text{ctg} \chi \sin \theta, \\ \Phi_n(\chi) &= a \sin^2 \chi, \\ \lambda_2 &= C \rho \frac{\omega^2}{2} d, \end{aligned} \quad (4.3.29)$$

де μ - лінійна щільність нитки; ω - переносна кутова швидкість; u - лінійна відносна швидкість; χ - кут атаки між ортом дотичної τ і вектором лінійної переносної швидкості елемента нитки; C, a - деякі постійні; q - питома вага нитки; ρ - щільність повітря; d - діаметр нитки.

Підставляючи отримані результати з (4.3.29) в (4.3.24) (4.3.26) (4.3.27).(4.3.28), отримаємо наступну систему диференціальних рівнянь

$$\begin{aligned} \frac{dP}{dr} &= q \frac{dz}{dr} - \mu \omega^2 r, \\ \frac{d\theta}{dr} &= \frac{1}{P} \left[\operatorname{tg} \theta (\mu \omega^2 r + \lambda_2 r^2 a \sin \chi \operatorname{tg} \psi) + \frac{q}{\cos \psi} \right], \\ \frac{d\psi}{dr} &= \frac{\mu \omega^2 r \sin \psi - \lambda_2 r^2 a \sin \chi - 2 \mu \omega u \cos \theta}{P \cos^2 \theta \cos \psi} - \frac{\operatorname{tg} \psi}{r}, \\ \frac{d\varphi}{dr} &= \frac{\operatorname{tg} \psi}{r}, \\ \frac{dz}{dr} &= \frac{\operatorname{tg} \theta}{\cos \psi}. \end{aligned} \quad (4.3.30)$$

Отриману систему диференціальних рівнянь необхідно інтегрувати з використанням чисельних методів. При розрахунках приймаємо $\lambda_2 a \sin \chi = 1$. При інтеграції необхідно використовувати наступні початкові умови: при $r = r_B = 50 \text{ мм}$, $\varphi = \varphi_0 = 0$, $z = z_0 = 0$.

При інтегруванні системи диференціальних рівнянь (4.3.30) використовувався метод Рунге-Кутта четвертого порядку (див. Додаток). Розрахунки проводилися для бавовняної пряжі 25 текс при наступних значеннях:

$$\mu = 25 \cdot 10^{-6} \text{ кг/м}, \quad q = 25 \cdot 10^{-5} \text{ н/м}, \quad \omega = 520 \text{ с}^{-1} (5000 \text{ об/мин}), \quad u = 20 \text{ м/с}.$$

Інтегрування виконували за наступних початкових умов:

$$P = P_0 = 0,2 \text{ Н}, \quad \theta = \theta_0 = 1,2 \text{ рад}, \quad \psi = \psi_0 = 0,7 \text{ рад}.$$

На першому етапі виконували інтеграцію, коли виконується умова $\psi \leq 1,57 \text{ рад}$ (відповідає максимальному значенню r_{\max}). На другому етапі виконуємо інтеграцію, коли виконується умова $r \geq 0$.

4.4. ЧИСЕЛЬНЕ ІНТЕГРУВАННЯ

Чисельну інтеграцію використовують тоді, коли знаходження первісною скрутно або неможливо. Процес чисельної інтеграції полягає в інтерполяції підінтегральної функції $f(x)$ на відрізку $[a, b]$ відповідним

поліномом, для якого певний інтеграл обчислюється по формулах чисельної інтеграції. Вибір виду полінома визначається властивістю класу функцій, для наближення якого призначаються інтерполяційні формули. Для наближення періодичних функцій (з періодом 2π) використовують тригонометричну систему функцій. Для наближення обмежених або зростаючих функцій на піввісь $[0, \infty)$ ефективно використовувати систему раціональних або показових функцій, які враховують поведінку функцій, що наближаються, на нескінченності.

Широкого поширення набула інтерполяція алгебри з використанням інтерполяційних многочленів (Лагранжа, Ньютона, Гауса, Бесселя і ін.). На практиці найчастіше використовуються параболічні або кубічні поліноміальні сплайни, які на кожному з відрізків $[x_{n-1}, x_n]$ представляють многочлени 2-го і 3-го ступеня, що належать класу двічі функцій, що безперервно диференціюються.

Хай задана підінтегральна функція $y = f(x)$ і відомі її значення в точках $x_0 = a, x_1, x_2, \dots, x_n = b$, причому $y_i = f(x_i), (i = 0, 1, 2, \dots, n)$. Необхідно визначити функцію, що наближає $g(x)$ в точках $x_0, x_1, x_2, \dots, x_n$, щоб виконувалася наближена рівність

$$\int_a^b f(x) dx \cong \int_a^b g(x) dx. \quad (4.4.1)$$

Інтерполяційний поліном Лагранжа для функції $f(x)$ буде мати вигляд

$$L_n(x) = \sum_{i=0}^n f(x_i) \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} = \sum_{i=0}^n f(x_i) \frac{\prod_{j=0}^n (x - x_j)}{(x - x_i) \prod_{j \neq i} (x_i - x_j)}. \quad (4.4.2)$$

Інтерполяційний поліном Ньютона з розділеними різницями матиме вигляд

$$L_n(x) = f(x_0) + f(x_0, x_1)(x - x_0) + \dots + f(x_0, x_1, x_2, \dots, x_n) \prod_{j=0}^{n-1} (x - x_j), \quad (4.4.3)$$

де розділені різниці першого, другого, $k-1$ порядку визначатимуться по формулах

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_j - x_i},$$

$$f(x_i, x_j, x_k) = \frac{f(x_j, x_k) - f(x_i, x_k)}{x_k - x_i},$$

$$f(x_0, x_1, \dots, x_k) = \prod_{j=0}^k \left[\frac{f(x_j)}{\prod_{\substack{j \neq i \\ i=0}}^k (x_j - x_i)} \right].$$

Використовуючи вирази для інтерполяційних поліномів (4.4.2)-(4.4.3), отримаємо з (4.4.1)

$$\int_a^b f(x) dx = \int_a^b L_n(x) dx + R_n, \quad (4.4.4)$$

де R_n - погрішність квадратурної формули.

Величина R_n на класі F подинтегральних функцій обчислюється за формулою [14]

$$R_n(F) = \sup_{f \in F} \left| \sum_{k=0}^n C_k f(x_k) - \int_{\Omega} f(x) \omega(x) dx \right| \quad (4.4.5)$$

У (4.4.5) вираз, що стоїть в модульних дужках визначає погрішність квадратури при обчисленні інтеграла від заданої функції $f(x)$.

Наближене значення інтеграла може бути визначене з (4.4.4) по формулі

$$\int_a^b f(x) dx = \int_{x_0}^{x_n} f(x) dx \cong \sum_{i=0}^n A_i f(x_i),$$

$$A_i = (x_n - x_0) H_i,$$

$$H_i = \frac{(-1)^{n-i}}{ni!(n-i)!} \int_0^n \frac{u(u-1)(u-2)\dots(u-n)}{u-i} du, \quad (4.4.6)$$

$$u = \frac{(x-x_0)n}{(x_n-x_0)}, \quad i = 0, 1, 2, \dots, n.$$

Для певного значення n коефіцієнти Лагранжа H_i , приймають певні значення. При $n=2$ (функція, що наближає, є поліномом другого ступеня) набудемо наступних значень коефіцієнтів Лагранжа

$$\begin{aligned} H_0 &= \frac{1}{4} \int_0^2 (u-1)(u-2) du = \frac{1}{6}, \\ H_1 &= -\frac{1}{2} \int_0^2 u(u-2) du = \frac{2}{3}, \\ H_2 &= \frac{1}{4} \int_0^2 u(u-1) du = \frac{1}{6}. \end{aligned} \quad (4.4.7)$$

Підставляємо (4.4.7) в (4.4.6), тоді

$$\int_{x_0}^{x_2} f(x) dx \cong \frac{(x_2 - x_0)}{6} [f(x_0) + 4f(x_1) + f(x_2)] \quad (4.4.8)$$

Розіб'ємо відрізок $[a, b]$ на парне число n ділянок $[x_0, x_2], [x_2, x_4], \dots, [x_{n-2}, x_n]$ довжини $(b-a)/n = h$. Використовуючи (4.4.8) і (4.4.6) отримаємо

$$\begin{aligned} \int_a^b f(x) dx \cong & \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + \frac{h}{3} [f(x_2) + 4f(x_3) + f(x_4)] + \\ & + \dots + \frac{h}{3} [f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \end{aligned}$$

Згрупуємо члени і отримаємо відому формулу Сімпсона [14]

$$\int_a^b f(x) dx \cong \frac{b-a}{3n} \{ f(a) + f(b) + 4[f(x_1) + f(x_3) + \dots + f(x_{n-1})] + 2[f(x_2) + f(x_4) + \dots + f(x_{n-2})] \}. \quad (4.4.9)$$

На рис.4.4.1 представлена графічна схема, що ілюструє інтерполяцію підінтегральної функції $f(x)$ і обчислення певного інтеграла на відрізку $[a, b]$ з використанням полінома другого ступеня по формулі Сімпсона. Програма *PROGRAM IN_M_SIM*, для обчислення певного інтеграла від функції $f(x)$ по формулі Сімпсона, приведена в Додатку.

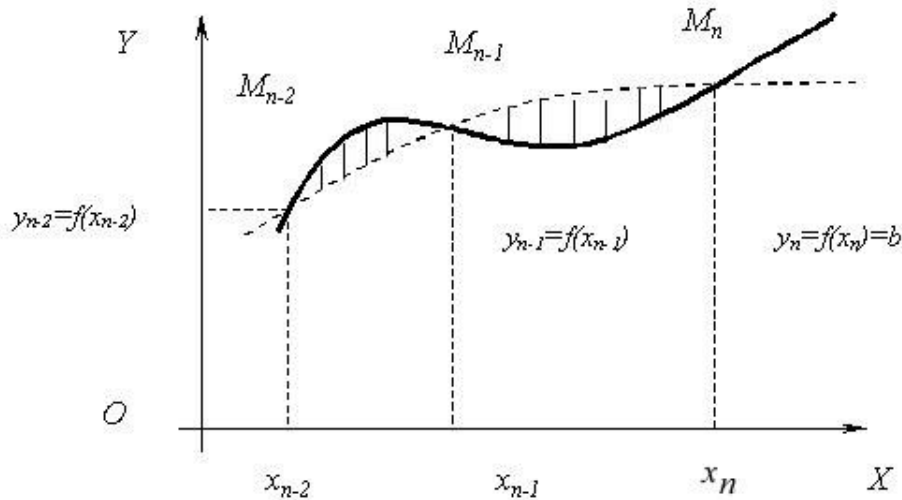


Рис.4.4.1. Розрахункова схема інтерполяції підінтегральної функції по формулі Сімпсона

Погрішність квадратурної формули Сімпсона, за умови безперервності четвертої похідної функції $f(x)$ на відрізку $[a, b]$ визначається з виразу

$$R_n = \frac{(b-a)h^4}{180} |y^{IV}(x_*)|,$$

де x_* - точка, що належить відрізку $[a, b]$.

При $n=1$ (функція, що наближає, є лінійною) з (4.4.6) набудемо наступних значень коефіцієнтів Лагранжа

$$H_0 = -\int_0^1 (u-1)du = \frac{1}{2}$$

$$H_1 = \int_0^1 udu = \frac{1}{2}.$$
(4.4.10)

Підставляємо (4.4.10) в (4.4.6), тоді

$$\int_{x_0}^{x_1} f(x)dx \cong \frac{(x_1 - x_0)}{2} [f(x_0) + f(x_1)]$$
(4.4.11)

Розіб'ємо відрізок $[a, b]$ на n рівних частин $[x_0 = a, x_1], [x_1, x_2], [x_2, x_3], \dots, [x_{n-1}, x_n = b]$ тоді

$$x_1 - x_0 = \frac{b-a}{n} = h.$$
(4.4.12)

Використовуючи (4.4.11),(4.4.12) і (4.4.6) отримаємо з урахуванням суміжних точок відрізків

$$\int_a^b f(x)dx \cong \frac{b-a}{n} \left[\frac{1}{2} f(a) + f(x_1) + f(x_2) + \dots + f(x_{n-1}) + \frac{1}{2} f(b) \right]. \quad (4.4.13)$$

Формула (4.4.13) є формулою трапецій для наближеного обчислення певного інтеграла (4.4.1).

Геометричний сенс (4.4.13) (см.рис.4.4.2) полягає в тому, що подинтегральної функція $f(x)$ замінюється на кожному інтервалі деякою лінійною функцією, що наближає. На кожному інтервалі $[a, x_1], [x_1, x_2], [x_2, x_3], \dots, [x_{n-1}, b]$ визначається площа трапеції, як твір напів суми підстав на висоту. Висота для всіх трапецій, за умови постійності кроку розбиття, дорівнюватиме $h = (b-a)/n$.

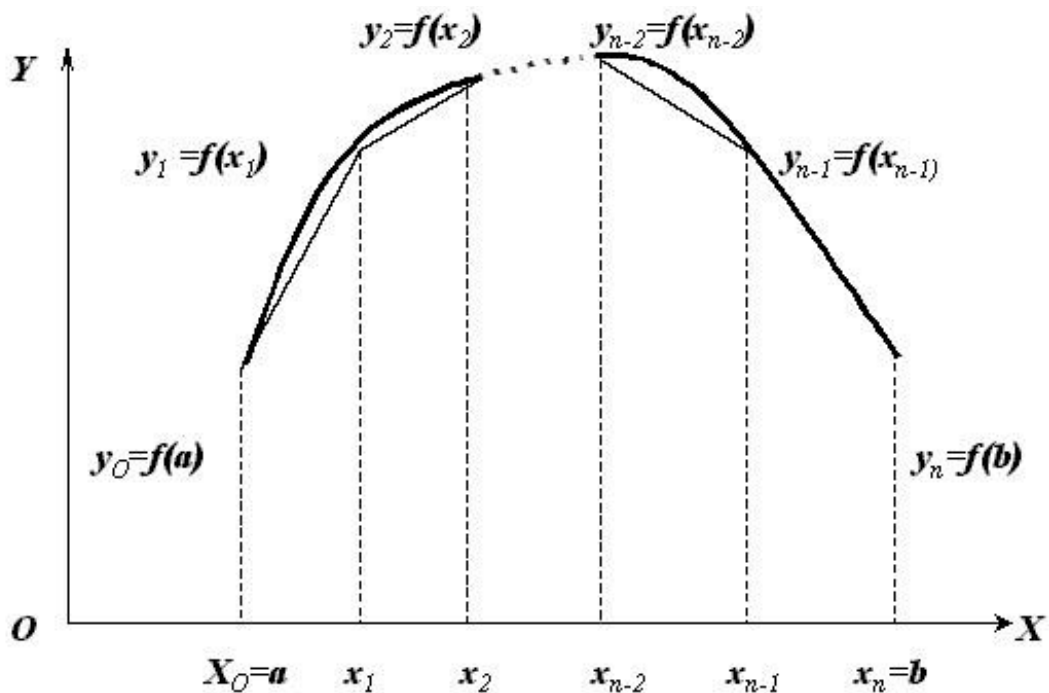


Рис.4.4.2. Геометрична інтерпретація формули трапеції

Програма **PROGRAM IN_M_TR**, для наближеного обчислення певного інтеграла від функції $f(x)$ по формулі трапецій, приведена в Додатку.

Погрішність квадратурної формули трапецій, за умови безперервності другої похідної функції $f(x)$ на відрізку $[a, b]$ визначається з виразу

$$R_n = \frac{(b-a)^3}{12n^2} |f''(x_*)|. \quad (4.4.14)$$

При $n=0$ (функція, що наближає, є лінійній, паралельній осі x) з (4.4.6) набуємо значення коефіцієнта Лагранжа

$$H_0 = 1. \quad (4.4.15)$$

Підставляємо (4.4.15) в (4.4.6), тоді для відрізання $[x_0, x_1]$ отримаємо

$$\int_{x_0}^{x_1} f(x) dx \cong (x_1 - x_0) f(x_0). \quad (4.4.16)$$

Розіб'ємо відрізок $[a, b]$ на n рівних частин $[x_0 = a, x_1], [x_1, x_2], [x_2, x_3], \dots, [x_{n-2}, x_{n-1}], [x_{n-1}, x_n = b]$. Середини даних відрізків визначаються з виразів

$$\xi_1 = a + \frac{h}{2}, \quad \xi_2 = x_1 + \frac{h}{2}, \dots, \xi_{n-1} = x_{n-2} + \frac{h}{2}, \quad \xi_n = x_{n-1} + \frac{h}{2}. \quad \text{Використовуючи}$$

(4.4.12), (4.4.16) і (4.4.6) отримаємо

$$\int_a^b f(x) dx \cong [f(\xi_1) + f(\xi_2) + f(\xi_3) + \dots + f(\xi_{n-1}) + f(\xi_n)] \frac{b-a}{n}. \quad (4.4.17)$$

Вираз (4.4.17) є формулою прямокутників для наближеного обчислення певного інтеграла (4.4.1).

Геометричний сенс (4.4.17) (см. рис. 4.4.3) полягає в тому, що площа криволінійної трапеції замінюється площею фігури, що складається з прямокутників. Всі прямокутники мають однакову підставу $(b-a)/n$ а їх висоти визначаються значеннями функції $f(x)$ у середніх крапках $\xi_1, \xi_2, \dots, \xi_{n-1}, \xi_n$ відрізків $[x_0 = a, x_1], [x_1, x_2], [x_2, x_3], \dots, [x_{n-2}, x_{n-1}], [x_{n-1}, x_n = b]$.

Програма **PROGRAM IN_M_PR**, для наближеного обчислення певного інтеграла за формулою прямокутників, приведена в Додатку.

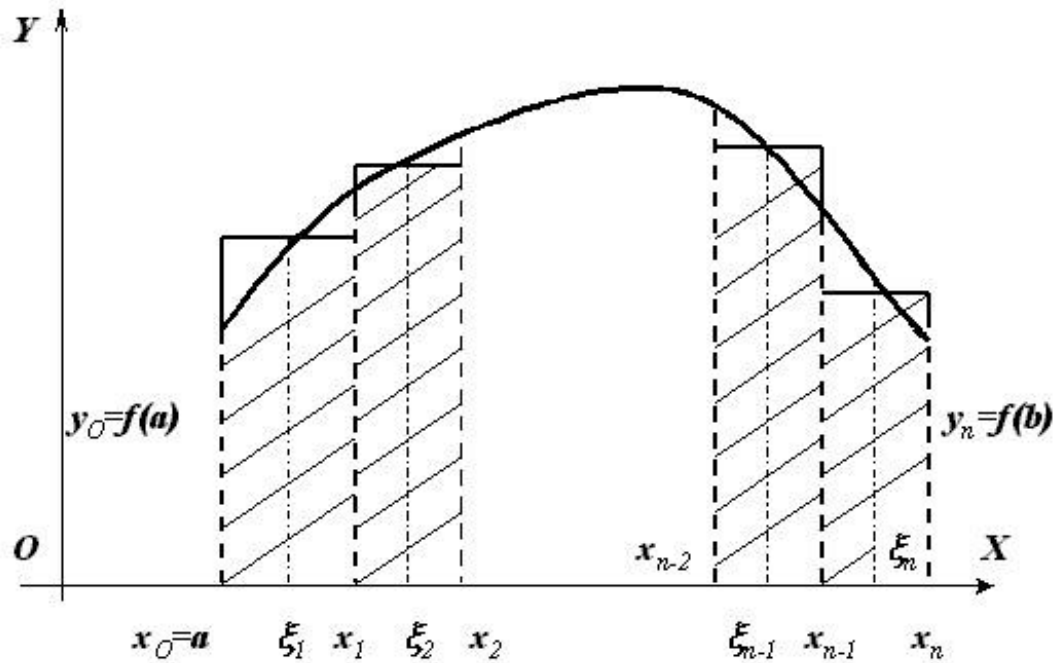


Рис.4.4.3. Геометрична інтерпретація формули прямокутників
 Погрішність квадратурної формули прямокутників, за умови
 безперервності другої похідної функції $f(x)$ на відрізку $[a, b]$ визначається
 з виразу

$$R_n = \frac{(b-a)^3}{24n^2} |f''(x_*)|. \quad (4.4.18)$$

Аналіз залежностей (4.4.18) і (4.4.14) дозволяє стверджувати, що
 формули прямокутників і трапецій володіють приблизно однаковою
 точністю. Вибір тієї або іншої формули залежить від виду подінтегральної
 формули, обчислювальних можливостей ЕОМ.

4.5. ІНТЕРПОЛЯЦІЯ, ЕКСТРАПОЛЯЦІЯ І АПРОКСИМАЦІЯ

Як наголошувалося в попередньому підрозділі, при розгляді чисельної
 інтеграції питання, зв'язані з наближенням функцій, виникають при
 чисельній інтеграції, вирішенні диференціальних рівнянь, обробці

експериментальних даних. Зокрема, наближення довільного контуру швейних лекал, деталей взуття простими функціями.

По характеру наближення методи наближення розділяють на інтерполяцію, екстраполяцію і апроксимацію.

Спосіб наближення функції, коли функція, що наближає $g(x)$ у заданих точках співпадає по значеннях із заданою $f(x)$ називається інтерполяцією. Хай задана деяка сукупність крапок $x_1, x_2, x_3, \dots, x_n$.

Інтерполяція має на увазі виконання наступної рівності

$$f(x_i) = g(x_i), \quad i = 1, 2, \dots, n. \quad (4.5.1)$$

Метод інтерполяції іноді називають точним методом. При екстраполяванні використовується інформація про поведінку функції в деякому кінцевому наборі крапок (у вузлах інтерполяції) і робиться припущення про подальше її поведінці. Даний метод іноді називають змішаним методом.

Під апроксимацією мається на увазі заміна заданій функції $f(x)$ деякою функцією $f_1(x)$ яка добре відображає властивості досліджуваного процесу, і різниця між значеннями даних функцій в деяких крапках з області їх визначення була б мінімальною

$$v_i = |f_1(x_i) - f(x_i)| = \min, \quad i = 1, 2, \dots, n, \quad (4.5.2)$$

де v_i - різниця між значенням заданої функції $f(x_i)$ і функції, що наближає $f_1(x_i)$ у деякій крапці x_i .

Іноді даний метод називають наближеним методом аналітичного опису дискретно заданої функції (по фіксованих значеннях у вузлах інтерполяції).

При обробці експериментальних даних від функції, що наближає, визначають з вимоги мінімізації суми

$$S = \min \left\{ \sum_{k=1}^n \alpha_k [f(x_k) - f_1(x_k)]^2 \right\}, \quad (4.5.3)$$

де α_k - задані числа.

Такий підхід до визначення функції, що наближає $f_1(x)$ називається інтерполяцією по методу найменших квадратів.

Найбільш поширений метод лінійної інтерполяції, коли функція (4.5.1), що наближає, представляється інтерполяційним многочленом

$$g(x) = \sum_{i=1}^n a_i \varphi_i(x), \quad (4.5.4)$$

де a_i - коефіцієнти многочлена (4.5.4), визначувані з умови збіги заданої функції $f(x)$ і функції, що наближає $g(x)$ у крапках $x_1, x_2, x_3, \dots, x_n$; $\varphi_i(x)$ – деякі фіксовані функції.

Обчислення коефіцієнтів многочлена (4.5.4) можна проводити по методу Ейткена шляхом послідовного застосування формул лінійної інтерполяції при довільному розташуванні вузлів без явної побудови інтерполяційного полінома по формулах

$$P_{i,i+1}(x) = \frac{1}{x_{i+1} - x_i} \begin{vmatrix} x - x_i & f(x_i) \\ x - x_{i+1} & f(x_{i+1}) \end{vmatrix}, \quad i = 1, 2, \dots, n-1, \quad (4.5.5)$$

$$P_{j-1,j,j+1}(x) = \frac{1}{x_{j+1} - x_{j-1}} \begin{vmatrix} x - x_{j-1} & P_{j-1,j}(x) \\ x - x_{j+1} & P_{j,j+1}(x) \end{vmatrix}, \quad j = 2, \dots, n-1.$$

Хай у нас задані значення $f(x_1)=10$, $f(x_2)=4$, $f(x_3)=2$, $f(x_4)=8$, при $x_1=0$, $x_2=2$, $x_3=3$, $x_4=6$. Використовуючи (4.5.5) по методу Ейткена за допомогою функціональних визначників другого порядку побудуємо інтерполяційний поліном

$$P_{1,2}(x) = \frac{1}{2} \begin{vmatrix} x & 10 \\ x-2 & 4 \end{vmatrix} = \frac{1}{2} (4x - 10x + 20) = 10 - 3x,$$

$$P_{2,3}(x) = \frac{1}{1} \begin{vmatrix} x-2 & 4 \\ x-3 & 2 \end{vmatrix} = 8 - 2x,$$

$$P_{3,4}(x) = \frac{1}{3} \begin{vmatrix} x-3 & 2 \\ x-6 & 8 \end{vmatrix} = \frac{1}{3} (8x - 24 - 2x + 12) = 2x - 4,$$

$$P_{1,2,3}(x) = \frac{1}{3} \begin{vmatrix} x & 10-3x \\ x-3 & 8-2x \end{vmatrix} = \frac{1}{3} (8x - 2x^2 + 3x^2 - 19x + 30) = \frac{1}{3} (x^2 - 11x + 30),$$

$$P_{2,3,4}(x) = \frac{1}{4} \begin{vmatrix} x-2 & 8-2x \\ x-6 & 2x-4 \end{vmatrix} = \frac{1}{4} (4x^2 - 28x + 56) = x^2 - 7x + 14,$$

$$P_{1,2,3,4}(x) = \frac{1}{6} \begin{vmatrix} x & \frac{1}{3}(x^2 - 11x + 30) \\ x-6 & x^2 - 7x + 14 \end{vmatrix} = \frac{1}{6} \left[x^3 - 7x^2 + 14x - \frac{(x-6)}{3}(x^2 - 11x + 30) \right].$$

Для перевірки правильності обчислення підставимо в останнє значення $x_1=0$ і $x_2=2$, отримаємо $P_{1,2,3,4}(0)=10$, $P_{1,2,3,4}(2)=4$.

Як наголошувалося в підрозділі 4.4, найчастіше при інтерполяції використовують інтерполяційні поліноми Лагранжа і Ньютона. Для періодичних функцій $f(x)$ з періодом $b-a$ ($x_1=a, x_n=b$) функція, що наближає $g(x)$ може бути представлена тригонометричним поліномом [14]

$$LT_n(x) = A + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx), \quad (4.5.6)$$

де A, a_k, b_k - коефіцієнти тригонометричного полінома (4.5.6). Завжди можна підібрати $2n+1$ коефіцієнтів A, a_k, b_k полінома n -го порядку так, щоб значення функції, що наближає, були рівні значенням $f(x_k)$ у $2n+1$ наперед заданих точках x_k проміжку $[a=0, b=2\pi]$ 2π -періодичній функції.

Використовуючи приведений вище вираз інтерполяційного полінома Лагранжа, представимо тригонометричні поліноми на відрізку $[-\pi]$ для парної функції $f(x)$

$$LT_n(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{\cos x - \cos x_j}{\cos x_i - \cos x_j},$$

для непарної функції $f(x)$

$$LT_n(x) = \sum_{i=1}^n f(x_i) \frac{\sin x}{\sin x_i} \prod_{\substack{j=1 \\ j \neq i}}^n \frac{\cos x - \cos x_j}{\cos x_i - \cos x_j}.$$

Враховуючи складність побудови інтерполяційних тригонометричних многочленів відрізок $[a=0, b=2\pi]$ розбивається на вузли, віддалені один від одного на рівні відстані $h = 2\pi / (2n + 1)$.

У звичайної поліноміальної інтерполяції зміна $f(x)$ поблизу якогось вузла робить вплив на зміну $f(x)$ по всій кривій, що якісно невірно описує фізичні явища лежачі в основі не лінійності інтерпольованої функції. Нерідко із збільшенням числа вузлів погрішність такої інтерполяції не тільки не зменшується, але і починає рости. Апаратом наближення функцій, позбавленим цих недоліків, є сплайни – функції, що складаються з різних многочленів, які наближають функцію на фіксованих ділянках відрізання $[a, b]$. Сплайн - інтерполяція відноситься до багатоінтервальної інтерполяції і забезпечує не тільки рівність $f(x_i) = g(x_i)$ у вузлах інтерполяції, але і безперервність заданого числа похідних на межах конкретних інтервалів. Гідністю локально заданих сплайнів є опис властивостей функції $f(x)$ на кожному окремому інтервалі незалежно від її властивостей на інших інтервалах.

На практиці ширше уживаються сплайни невисокого ступеня – параболічні і кубічні. Вони представляють інтерполюючу функцію у вигляді полінома другого або третього ступеня.

Сплайн-функція $S_m(\Delta n; x)$ ($m = 2, 3$) визначена на відрізку $[a, b]$ і співпадає на відрізках $[x_i, x_{i+1}]$ з деякими многочленами алгебри ступеня

не вище m . Сплайн-функція має на відрізку $[a, b]$ безперервну $m-1$ похідну. Сітка $\Delta_n: a = x_0 < x_1 < \dots < x_n = b$ розбиває відрізок $[a, b]$ на часткові відрізки. Якщо сплайн-функція має на відрізку $[a, b]$ безперервну $m-k$ -ю похідну ($k \geq 1$), а $m-k+1$ -я похідна у вузлах сплайна має розрив, то говорять, що сплайн має дефект порядку k .

Загальний вираз для сплайн-функції має вигляд [14]

$$S_m(\Delta_n; x) = P_{m-1}(x) + \sum_{k=0}^{n-1} c_k (x - x_k)_+^m, \quad (4.5.7)$$

де c_k - дійсні числа $P_{m-1}(x)$ - многочлен алгебри ступеня не вищий $m-1$;
 $(x - x_k)_+^m = [\max(0, x - x_k)]^m$.

Для параболічного сплайна потрібно дві додаткові краєві умови

$$\begin{aligned} S'_2(a) &= a_n, \quad S'_2(b) = b_n, \\ S''_2(a) &= A_n, \quad S''_2(b) = B_n. \end{aligned} \quad (4.5.8)$$

Якщо функція $f(x)$ має першу і другу похідні на відрізку $[a, b]$ то значення системи (4.5.8) можуть бути визначені з наступної системи рівнянь

$$\begin{aligned} a_n &= f'(a), \quad b_n = f'(b), \\ A_n &= f''(a), \quad B_n = f''(b). \end{aligned} \quad (4.5.9)$$

Для параболічного сплайна отримаємо

$$\begin{aligned} S_2(x) &= f(x_i) + m_i(x - x_i) + c_i(x - x_i)^2, \\ c_i &= \frac{m_{i+1} - m_i}{2h_i} - \frac{f(x_{i+1}) - f(x_i)}{h_i(\bar{h}_i - h_i)} + \frac{m_{i+1} + m_i}{2(\bar{h}_i - h_i)}, \\ h_i &= x_{i+1} - x_i, \quad \bar{h}_i = x_{i+1} - \bar{x}_{i+1}, \quad m_i = S'_2(x_i), \\ & \quad i = 0, \dots, n, \end{aligned} \quad (4.5.10)$$

де \bar{x}_i - координата вузла сплайна або точка можливого розриву другої похідної; x_i - координата інтерполяції.

На практиці частіше розглядаються випадки, коли координати вузлів сплайна розташовуються посередині, між вузлами інтерполяції

$$\bar{x}_i = x_i + \frac{x_{i+1} - x_i}{2}, \quad i = 0, \dots, n.$$

Для кубічного сплайна на часткових відрізках $[x_i, x_{i+1}]$ загальна формула, з обліком (4.5.7), має вигляд

$$S_3(x) = \frac{1}{6h_i} \left[m_i (x_{i+1} - x)^3 + m_{i+1} (x - x_i)^3 \right] + \frac{1}{h_i} \left\{ \left[f(x_i) - \frac{m_i h_i^2}{6} \right] (x_{i+1} - x) + \left[f(x_{i+1}) - \frac{m_{i+1} h_i^2}{6} \right] (x - x_i) \right\}, \quad (4.5.11)$$

$$h_i = x_{i+1} - x_i, \quad m_i = f''(x_i), \quad i = 1, 2, \dots, n,$$

де n - число вузлів.

При інтерполяції $f(x_i) = S_3(x_i)$ отримаємо систему лінійних рівнянь для визначення m_i

$$h_i m_i + 2(h_i + h_{i+1}) m_{i+1} + h_{i+1} m_{i+2} = 6 \left[\frac{f(x_{i+2}) - f(x_{i+1})}{h_{i+1}} - \frac{f(x_{i+1}) - f(x_i)}{h_i} \right], \quad (4.5.12)$$

$$i = 1, 2, \dots, n.$$

Систему лінійних рівнянь (4.5.12) необхідно доповнити граничними умовами

$$\begin{aligned} S_3'(a) &= a_n, \quad S_3'(b) = b_n, \\ S_3''(a) &= A_n, \quad S_3''(b) = B_n. \end{aligned} \quad (4.5.13)$$

Якщо функція $f(x)$ має відповідні похідні, то система (4.5.13) визначається з (4.5.9). Якщо значення другої похідної рівні $m_1=0$, $m_n=0$, то отримуємо нормальні сплайн-функції. У разі, коли $m_1 = m_n$ маємо періодичні сплайн-функції.

Рішення задачі інтерполяції функції зводиться до побудови інтерполяційного полінома Лагранжа, а у разі рівновіддалених вузлів – до побудови інтерполяційного полінома Ньютона.

При проведенні теоретичних і експериментальних досліджень часто виникає необхідність отримання більш простій емпіричної формули, яка добре відображає фізичні властивості досліджуваного процесу. Цього

можна добитися з використанням апроксимації. Опис зв'язку між деяким числом N пара значень x_i і y_i із забезпеченням найменшої середньоквадратичної погрішності, можна здійснити з використанням методу найменших квадратів, шляхом апроксимації функцій ортогональними поліномами. З метою зменшення випадкових помилок і отримання більш гладкої функції згладжування часто використовують кубічні сплайни.

У справжній роботі ми зупинимося тільки на апроксимації функцій з використанням методу найменших квадратів. Аналіз методу почнемо з лінійної апроксимації. На мал. 4.5.1 показаний розподіл пар значень x_i, y_i і графік лінійної апроксимуючої залежності $y = y(x)$.

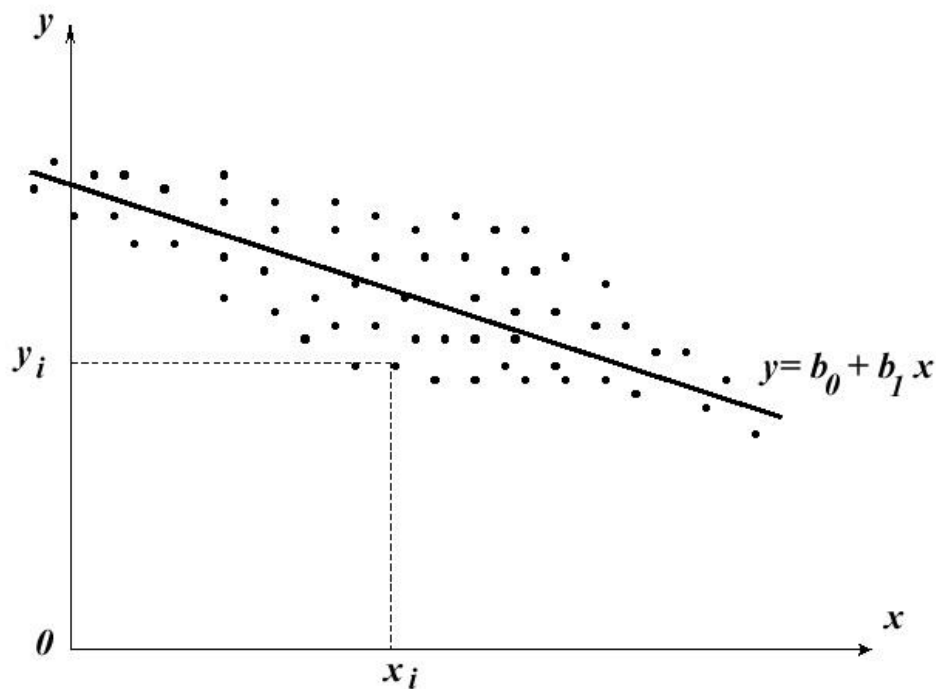


Рис.4.5.1. Лінійна апроксимація

Геометрична суть процесу лінійної апроксимації полягає в тому, щоб провести пряму $y(x) = b_0 + b_1 x$ так, щоб величина всіх відхилень відповідала умові

$$\begin{aligned}
 U &= \sum_{i=1}^N U_i = \min, \\
 U_1 &= [y_1 - b_1 x_1 - b_0]^2, \\
 U_2 &= [y_2 - b_1 x_2 - b_0]^2, \\
 &\dots\dots\dots \\
 U_N &= [y_N - b_1 x_N - b_0]^2.
 \end{aligned}
 \tag{4.5.14}$$

Для виконання першої умови системи (4.5.14) необхідно визначити відповідні приватні похідні

$$\frac{\partial U_i}{\partial b_0} = 0, \quad \frac{\partial U_i}{\partial b_1} = 0, \quad i = 1\dots N.
 \tag{4.5.15}$$

Визначимо значення даних приватних похідних

$$\begin{aligned}
 \frac{\partial U_1}{\partial b_0} &= -2[y_1 - b_1 x_1 - b_0], \quad \frac{\partial U_1}{\partial b_1} = \\
 &= -2[y_1 - b_1 x_1 - b_0]x_1 = -2[y_1 x_1 - b_1 x_1^2 - b_0 x_1], \\
 \frac{\partial U_2}{\partial b_0} &= -2[y_2 - b_1 x_2 - b_0], \quad \frac{\partial U_2}{\partial b_1} = \\
 &= -2[y_2 - b_1 x_2 - b_0]x_2 = -2[y_2 x_2 - b_1 x_2^2 - b_0 x_2], \\
 &\dots\dots\dots \\
 \frac{\partial U_N}{\partial b_0} &= -2[y_N - b_1 x_N - b_0], \quad \frac{\partial U_N}{\partial b_1} = \\
 &= -2[y_N - b_1 x_N - b_0]x_N = -2[y_N x_N - b_1 x_N^2 - b_0 x_N]
 \end{aligned}
 \tag{4.5.16}$$

З обліком (4.5.15), (4.5.16) отримаємо

$$\sum_{i=1}^N \frac{\partial U_i}{\partial b_0} = 0, \quad \sum_{i=1}^N \frac{\partial U_i}{\partial b_1} = 0.
 \tag{4.5.17}$$

Вирішуючи спільно (4.5.16) і (4.5.17), отримаємо два рівняння для визначення постійних коефіцієнтів в лінійному рівнянні апроксимації

$$\begin{aligned}
 \sum_{i=1}^N y_i &= b_1 \sum_{i=1}^N x_i + N b_0, \\
 \sum_{i=1}^N x_i y_i &= b_1 \sum_{i=1}^N x_i^2 + b_0 \sum_{i=1}^N x_i.
 \end{aligned}
 \tag{4.5.18}$$

Вирішуючи систему двох лінійних рівнянь (4.5.18) щодо постійних коефіцієнтів лінійного рівняння апроксимації, отримаємо

$$b_0 = \frac{1}{N} \left(\sum_{i=1}^N y_i - b_1 \sum_{i=1}^N x_i \right),$$

$$b_1 = \frac{\sum_{i=1}^N y_i \sum_{i=1}^N x_i - N \sum_{i=1}^N x_i y_i}{\left(\sum_{i=1}^N x_i \right)^2 - N \sum_{i=1}^N x_i^2}. \quad (4.5.19)$$

Обчислення середньоквадратичної погрішності дозволить якісно оцінювати ступінь наближення крапок x_i, y_i до апроксимуючої прямої. Величина середньоквадратичної погрішності визначиться по формулі

$$\sigma^2 = \frac{1}{N} \left[\sum_{i=1}^N y_i^2 - b_1 \sum_{i=1}^N y_i - b_0 \sum_{i=1}^N x_i y_i \right].$$

Більш загальний випадок апроксимації пар значень x_i, y_i забезпечується за допомогою поліноміальної регресії вигляду

$$y(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m. \quad (4.5.20)$$

При $m = N$ має місце звичайна інтерполяція (значення $y(x)$ при $x = x_i$ точно співпадають з визначеним y_i). При $m < N$ такого збігу в загальному випадку немає.

Суть поліноміальної апроксимації полягає, з обліком (4.5.20), у виконанні наступної умови

$$U = \sum_{i=1}^N U_i = \min,$$

$$U_1 = [y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \dots - a_m x_1^m]^2,$$

$$U_2 = [y_2 - a_0 - a_1 x_2 - a_2 x_2^2 - \dots - a_m x_2^m]^2,$$

.....

$$U_N = [y_N - a_0 - a_1 x_N - a_2 x_N^2 - \dots - a_m x_N^m]^2. \quad (4.5.21)$$

Для мінімізації величини U з обліком (4.5.21), визначимо значення приватних похідних по відповідних коефіцієнтах полінома (4.5.20)

$$\begin{aligned}
 \frac{\partial U_i}{\partial a_0} = \frac{\partial U_i}{\partial a_1} = \frac{\partial U_i}{\partial a_2} = \dots = \frac{\partial U_i}{\partial a_m} &= 0, \quad i = 1 \dots N, \\
 \frac{\partial U_1}{\partial a_0} &= -2[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \dots - a_m x_1^m], \\
 \frac{\partial U_1}{\partial a_1} &= -2[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \dots - a_m x_1^m] x_1, \\
 \frac{\partial U_1}{\partial a_2} &= -2[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \dots - a_m x_1^m] x_1^2, \\
 &\dots \dots \dots \dots \dots \dots \dots \dots \\
 \frac{\partial U_1}{\partial a_m} &= -2[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \dots - a_m x_1^m] x_1^m, \\
 &\dots \dots \dots \dots \dots \dots \dots \dots \\
 \frac{\partial U_N}{\partial a_0} &= -2[y_N - a_0 - a_1 x_N - a_2 x_N^2 - \dots - a_m x_N^m], \\
 \frac{\partial U_N}{\partial a_1} &= -2[y_N - a_0 - a_1 x_N - a_2 x_N^2 - \dots - a_m x_N^m] x_N, \\
 \frac{\partial U_N}{\partial a_2} &= -2[y_N - a_0 - a_1 x_N - a_2 x_N^2 - \dots - a_m x_N^m] x_N^2, \\
 &\dots \dots \dots \dots \dots \dots \dots \dots \\
 \frac{\partial U_N}{\partial a_m} &= -2[y_N - a_0 - a_1 x_N - a_2 x_N^2 - \dots - a_m x_N^m] x_N^m.
 \end{aligned} \tag{4.5.22}$$

З обліком (4.5.21), отримаємо

$$\sum_{i=1}^N \frac{\partial U_i}{\partial a_0} = \sum_{i=1}^N \frac{\partial U_i}{\partial a_1} = \sum_{i=1}^N \frac{\partial U_i}{\partial a_2} = \dots = \sum_{i=1}^N \frac{\partial U_i}{\partial a_m} = 0.$$

Використовуючи останню систему рівнянь можна визначити лінійну систему рівнянь для визначення постійних коефіцієнтів в поліноміальному апроксимуючому рівнянні (4.5.20)

$$\begin{aligned}
 \sum_{i=1}^N y_i &= Na_0 + a_1 \sum_{i=1}^N x_i + a_2 \sum_{i=1}^N x_i^2 + \dots + a_m \sum_{i=1}^N x_i^m, \\
 \sum_{i=1}^N x_i y_i &= a_0 \sum_{i=1}^N x_i + a_1 \sum_{i=1}^N x_i^2 + a_2 \sum_{i=1}^N x_i^3 + \dots + a_m \sum_{i=1}^N x_i^{m+1}, \\
 \sum_{i=1}^N x_i^2 y_i &= a_0 \sum_{i=1}^N x_i^2 + a_1 \sum_{i=1}^N x_i^3 + a_2 \sum_{i=1}^N x_i^4 + \dots + a_m \sum_{i=1}^N x_i^{m+2}, \\
 &\dots \dots \dots \\
 \sum_{i=1}^N x_i^m y_i &= a_0 \sum_{i=1}^N x_i^m + a_1 \sum_{i=1}^N x_i^{m+1} + a_2 \sum_{i=1}^N x_i^{m+2} + \dots + a_m \sum_{i=1}^N x_i^{2m}.
 \end{aligned}
 \tag{4.5.23}$$

Систему лінійних рівнянь (4.5.23) можна вирішити методом Гауса з вибором головного елемента (див. 4.1).

Поліноміальна регресія дозволяє апроксимувати N пара x_i, y_i з мінімальною середньоквадратичною погрішністю

$$E = \sqrt{\sum_{i=1}^N \frac{\varepsilon_i^2}{N+1}}.
 \tag{4.5.24}$$

Поліноміальна апроксимація з автоматичним вибором ступеня полінома дозволяє шляхом підвищення ступеню полінома з $m = 1$ до $m = k$ визначати величини коефіцієнтів $a_0 \dots a_m$.

По (4.5.24) визначається середньоквадратична погрішність E яка може порівнюватися із заданою E_1 . При виконанні нерівності $E > E_1$ ступінь полінома m збільшується на 1, до того моменту, коли виконуватиметься нерівність $E < E_1$.

Для реалізації даного методу був розроблений алгоритм, який був реалізований в програмі **PROGRAM AP_M_NK** (див. Додаток).

4.6. МАТЕМАТИЧНІ МОДЕЛІ, ОТРИМАНІ ШЛЯХОМ ОБРОБКИ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Розглянуті нижче методи планування експериментальних досліджень дозволяють отримувати математичні моделі з необхідною точністю і статистичною надійністю при мінімальних трудомісткості і матеріальних

витратах.

Результатом будь-якого досвіду є подія, яка може мати якісну або кількісну характеристику. Події можуть бути достовірними, неможливими, випадковими. По характеру відповідності подій між собою їх можна класифікувати з одного боку як несумісні і сумісні, з іншого боку як залежні і незалежні.

У класичній схемі вірогідність $P(s)$ виникнення події S визначається відношенням числа елементарних випадків, сприятливих даній події SB до загального числа всіх рівно можливих випадків SO

$$P(S) = \frac{SB}{SO}. \quad (4.6.1)$$

Велике значення при проведенні експериментальних досліджень необхідно приділяти точність вимірювань. Отриманий експериментальним шляхом результат завжди містить деяку помилку. Розрізняють систематичні, випадкові і грубі помилки. Систематичні помилки з'являються унаслідок несправності вимірювальних приладів, недосконалості методики вимірювання. Випадкові помилки при повторних вимірюваннях можуть мати як об'єктивний, так і суб'єктивний характер (наприклад, короткочасна зміна температури навколишнього середовища, напруга живлення приладів і ін.). При проведенні вимірювань можуть виникати грубі помилки, які викликані неухважністю експериментатора. Для визначення випадкових і грубих помилок використовують методи математичної статистики, які засновані на елементах теорії вірогідності. Випадкові величини можуть бути дискретними і безперервними.

Безперервні випадкові величини характеризуються функцією розподілу $F(x)$ щільністю вірогідності $\varphi(x)$ які зв'язані співвідношенням [31]

$$F(x) = \int_{-\infty}^x \varphi(x) dx. \quad (4.6.2)$$

На практиці широко використовуються числові характеристики (статистики) випадкових величин. До них відносяться математичне

очікування $M\{x\} = a$ або середнє значення випадкової величини x (абсциса центру тяжіння площі під кривій щільності вірогідності $\varphi(x)$ - рис.4.6.1 а), мода випадкової величини $M_o\{x\}$ (абсциса відповідає максимальному значенню щільності вірогідності – рис.4.6.1 би) медіана випадкової величини $M_e\{x\}$ (абсциса точок прямої, яка ділить площу, обмежену кривій щільності вірогідності, навпіл – рис.4.6.1 в) [31].

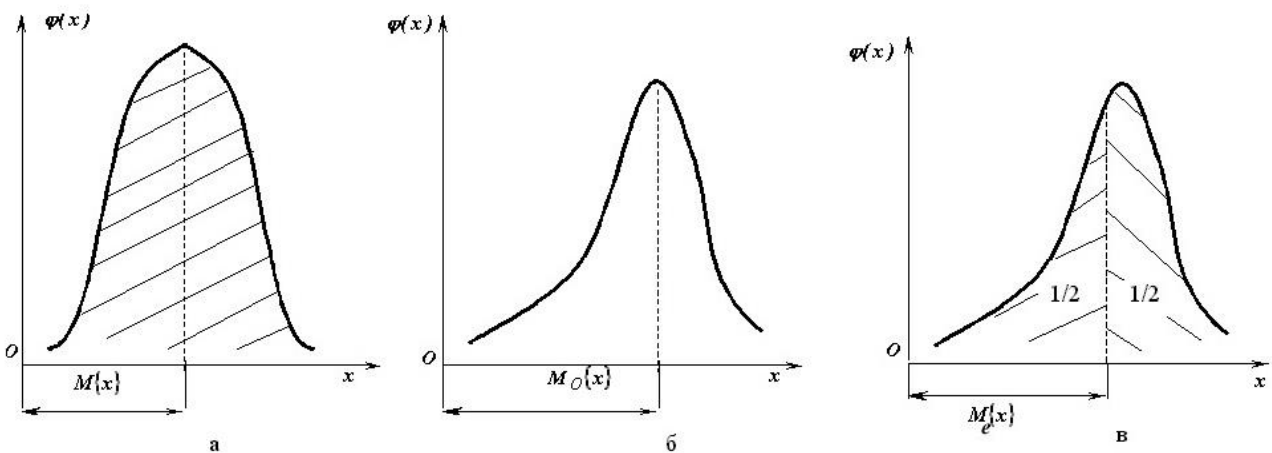


Рис.4.6.1. Графік щільності вірогідності з вказівкою: математичного очікування, моди і медіани

Дисперсія $D\{x\}$ (σ^2) є однієї з основних характеристик розсіяння випадкової величини x біля центру розподілу і визначається по формулі

$$D\{x\} = \sigma^2 = \int_{-\infty}^{\infty} (x - a)^2 \varphi(x) dx. \quad (4.6.3)$$

Середнім квадратичним відхиленням (стандартним відхиленням) називається позитивне значення квадратного кореня з дисперсії

$$\sigma = \sqrt{\sigma^2}. \quad (4.6.4)$$

Відношення середнього квадратичного відхилення до математичного очікування називається коефіцієнтом варіації γ який визначається по формулі

$$\gamma = \frac{\sqrt{\int_{-\infty}^{\infty} \left[x - \int_{-\infty}^{\infty} x\varphi(x)dx \right]^2 \varphi(x)dx}}{\int_{-\infty}^{\infty} x\varphi(x)dx} = \frac{\sigma}{a}. \quad (4.6.5)$$

Вираз (4.6.5) показує наскільки велике розсіяння в порівнянні з середнім значенням випадкової величини.

Показник S_k асиметрії розподілу визначається по формулі

$$S_k = \frac{\mu_3}{\sigma^3}, \quad \mu_3 = \int_{-\infty}^{\infty} (x-a)^3 \varphi(x)dx. \quad (4.6.6)$$

Для оцінки гостро піковості розподілу (крутизни) використовується показник ексцесу E_k який визначається по формулі

$$E_k = \frac{\mu_4}{\sigma^4} - 3, \quad \mu_4 = \int_{-\infty}^{\infty} (x-a)^4 \varphi(x)dx. \quad (4.6.7)$$

Визначені по (4.6.6),(4.6.7) показники асиметрії і ексцесу дорівнюють нулю у разі нормального розподілу.

Нижче приведені виразу для основних диференціальних функцій розподілу (щільності вірогідності) випадкових величин [31]:

нормальний розподіл Гауса

$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-a)^2}{2\sigma^2}\right], \quad -\infty < x < \infty,$$

логарифмічний нормальний розподіл

$$\varphi(x) = \frac{1}{x\sigma_y\sqrt{2\pi}} \exp\left[-\frac{(\ln x - M\{\ln x\})^2}{2\sigma_y^2}\right], \quad \sigma_y^2 = D\{\ln x\}, \quad 0 \leq x < \infty,$$

експоненціальне

$$\varphi(x) = \frac{1}{M\{x\}} \exp\left(-\frac{x}{M\{x\}}\right), \quad 0 \leq x < \infty, \quad \frac{1}{M\{x\}} > 0,$$

показово-статичне

$$\varphi(x) = \frac{x^{M\{x\}-1}}{(M\{x\}-1)!} \exp(-x), \quad 0 \leq x < \infty, \quad M\{x\}-1 > 0,$$

Вейбулла - Гнеденко

$$\varphi(x) = \begin{cases} 1 - \exp\left[-\left(\frac{x - x_H}{c}\right)^b\right], & \text{при } x > x_H, \\ 0, & \text{при } x \leq x_H, \end{cases}$$

$b > 0, c > 0, x_H$ - нижня межа зміни випадкової величини

Бета - розподіл

$$\varphi(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1},$$

$$M\{x\} = \frac{\alpha}{\alpha + \beta}, \quad \sigma^2 = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}, \quad 0 \leq x < \infty.$$

При проведенні експериментальних досліджень неможливо провести дуже багато виміри. З цього виходить, що не можна побудувати функцію розподілу для визначення дійсного значення x . Таким чином, хорошим наближенням до дійсного значення можна рахувати середньоарифметичне значення

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k, \quad (4.6.8)$$

де n - число вимірювань.

Достатньо точною оцінкою помилки можна вважати вибіркву дисперсію S_n^2 витікаючи з відповідного (наприклад, нормального) закону розподілу, але що відноситься до кінцевого числа вимірювань. Така назва пояснюється тим, що зі всієї безлічі можливих значень x_k (генеральній сукупності) вимірюють лише кінцеве число значень n яке називається вибіркою і характеризується вибірквим середнім значенням і вибірквою дисперсією.

Середньою квадратичною погрішністю окремого вимірювання називається величина рівна кореню квадратному з вибірквої дисперсії

$$S_n = \sqrt{\frac{\sum_{k=1}^n (\bar{x} - x_k)^2}{n-1}}. \quad (4.6.9)$$

Середня квадратична погрішність ряду вимірювань визначається по формулі

$$S_{\bar{x}} = \frac{S_n}{\sqrt{n}}. \quad (4.6.10)$$

Аналіз залежності (4.6.10) показує, що із збільшенням числа вимірювань середня квадратична погрішність прагне до нуля. Проте, це досягається лише при вельми значному числі вимірювань.

Після визначення наближеного значення вимірюваної величини необхідно визначити надійність знайденого дійсного значення x . Довірчим інтервалом при цьому буде інтервал $(\bar{x} - \varepsilon; \bar{x} + \varepsilon)$ у якому знаходиться із заданою вірогідністю дійсне значення x ($\varepsilon = t \cdot S_{\bar{x}}$ $t = \frac{|\bar{x} - x|}{S_{\bar{x}}}$)

Таблиця 4.6.1. Значення критерію Стюдента (t -критерію) для різної довірчої вірогідності (α), для різного числа вимірювань (n)

n	α								
	0,50	0,60	0,70	0,80	0,90	0,95	0,98	0,99	0,999
2	2,00	1,38	2,0	3,1	6,31	12,71	31,8	63,7	637
3	0,82	1,06	1,3	1,9	2,92	4,30	6,96	9,92	31,6
4	0,77	0,98	1,25	1,6	2,35	3,18	4,54	5,84	12,9
5	0,74	0,94	1,2	1,5	2,13	2,78	3,75	4,60	8,6
6	0,73	0,92	1,2	1,5	2,02	2,57	3,36	4,03	6,9
7	0,72	0,90	1,1	1,4	1,94	2,45	3,14	3,71	6,0
8	0,71	0,90	1,1	1,4	1,90	2,37	3,00	3,50	5,4
9	0,71	0,89	1,1	1,4	1,86	2,31	2,90	3,36	5,0
10	0,70	0,88	1,1	1,4	1,83	2,26	2,82	3,25	4,8
11	0,70	0,88	1,1	1,4	1,81	2,23	2,76	3,17	4,6
12	0,70	0,87	1,1	1,4	1,80	2,20	2,72	3,10	4,5
13	0,70	0,87	1,1	1,4	1,78	2,18	2,68	3,05	4,3

14	0,69	0,87	1,1	1,4	1,77	2,16	2,65	3,30	4,2
15	0,69	0,87	1,1	1,3	1,76	2,15	2,62	2,98	4,1
16	0,69	0,87	1,1	1,3	1,75	2,13	2,60	2,95	4,0
17	0,69	0,86	1,1	1,3	1,75	2,12	2,58	2,92	4,0
18	0,69	0,86	1,1	1,3	1,74	2,11	2,56	2,90	4,0
19	0,69	0,86	1,1	1,3	1,73	2,10	2,55	2,88	3,9
20	0,69	0,86	1,1	1,3	1,73	2,09	2,54	2,85	3,9
30	0,68	0,85	1,1	1,3	1,7	2,0	2,5	2,8	3,7
40	0,68	0,85	1,1	1,3	1,7	2,0	2,4	2,7	3,6
60	0,68	0,85	1,0	1,3	1,7	2,0	2,4	2,7	3,5
120	0,68	0,85	1,0	1,3	1,7	2,0	2,4	2,6	3,4
∞	0,67	0,84	1,0	1,3	1,6	2,0	2,3	2,6	3,3

Якщо по n результатам розраховується середньоарифметичне значення \bar{x} то квадрат експериментальної оцінки середньо квадратичного відхилення $S_{(x_k)}^2$ є експериментальною оцінкою дисперсії

$$S_{(x_k)}^2 = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2. \quad (4.6.11)$$

Числом мір свободи f називається різниця між числом незалежних результатів в n повторностях і числом рівнянь, в яких ці результати використані для розрахунку невідомих оцінок (в даному випадку вони використовувалися в одному рівнянні (4.6.8) для розрахунку середньоарифметичного значення \bar{x}).

Довірча помилка $\varepsilon_{(x_k)}$ для одиничного результату, з обліком (4.6.11), дорівнює $\varepsilon_{(x_k)} = t \cdot S_{(x_k)}$ а довірча помилка для середнього результату $\varepsilon_{(\bar{x})}$ з обліком (4.6.10), визначається по формулі $\varepsilon_{(\bar{x})} = t \cdot S_{(\bar{x})}$.

Відносну помилку середнього результату визначають по формулі

$$O_{(\bar{x})} = \frac{\varepsilon_{(\bar{x})}}{\bar{x}} 100\%. \quad (4.6.12)$$

Відносне стандартне відхилення визначається як відношення стандартного відхилення до середньоарифметичного значення вимірюваної величини

$$\Delta = \frac{S(\bar{x})}{\bar{x}} 100\%. \quad (4.6.13)$$

Як практичне застосування розглянемо випадок визначення необхідної кількості повторних дослідів для отримання результатів із заданою точністю. На попередньому етапі визначуваний по (4.6.9).(4.6.10) величину середньої квадратичної погрішності $S(\bar{x})$ для невеликого числа дослідів (2-3 вимірювання). Задаємося довірчою вірогідністю α (для інженерних розрахунків ця величина дорівнює 0,95) і значенням довірчої помилки для середнього результату (зазвичай можна приймати $\varepsilon(\bar{x}) = (0,05 \dots 0,1) \cdot \bar{x}$). Тоді, з обліком (4.6.10).(4.6.11), отримаємо

$$t_p = \frac{\varepsilon(\bar{x})}{S(x_k)} \sqrt{n}. \quad (4.6.14)$$

Для заданого значення довірчої вірогідності α і числа мір свободи $f = n - 1$ визначаємо по таблиці 4.6.1 значення t - критерію Стьюдента t_T . По формулі (4.6.14) визначаємо розрахункове значення t_p - критерію Стьюдента. Для забезпечення точності результату із заданою довірчою помилкою $\varepsilon(\bar{x})$ з довірчою вірогідністю α необхідно, щоб виконувалася нерівність $t_p > t_T$. Інакше, необхідно збільшити число повторностей досвіду n і по формулі (4.6.14) визначити нове значення розрахункового t_p - критерію Стьюдента. При визначенні нового табличного значення t_T - критерію Стьюдента необхідно враховувати зміну числа мір свободи $f = n - 1$.

При обробці експериментальних даних, для визначення грубих помилок, можна використовувати t - критерій Стьюдента. Послідовність виконання операцій буде наступна. Для заданого числа повторностей досвіду n визначаємо значення x_{ks} яке викликає сумнів. Відкидаємо його і для тих, що залишилися $n - 1$ даних визначаємо середньоарифметичне значення $\bar{x}_{(n-1)}$ і середньо квадратичне відхилення (стандартне

відхилення) $S_{(x_k)}$ по формулі

$$\bar{x}_{(n-1)} = \frac{\sum_{k=1}^{n-1} x_k}{n-1}, \quad S_{(x_k)} = \sqrt{\frac{\sum_{k=1}^{n-1} [\bar{x}_{(n-1)} - x_k]^2}{n-2}}.$$

Потім визначаємо розрахункове значення t_p - критерію Стюдента по формулі

$$t_p = \frac{|x_{kS} - \bar{x}_{(n-1)}|}{S_{(x_k)}}. \quad (4.6.15)$$

Результат досвіду вважається грубою помилкою, якщо розрахункове значення t_p по модулю більше табличного $t_p > t_T$. В даному випадку результат x_{kS} можна виключити як грубу помилку (табличне значення t_T - критерію Стюдента вибирається для числа мір свободи $f = (n - 1) - 1$).

Сучасна математична теорія оптимального планування експерименту включає два основні напрями: вивчення механізмів складних процесів і поведінка багатокомпонентних систем на основі планування експерименту; оптимізація технологічних процесів і властивостей багатокомпонентних систем на основі планування експерименту. Експериментальні дослідження можуть проводитися безпосередньо на об'єкті або його моделі. Останнім часом разом з фізичними моделями всього більшого поширення набувають абстрактні математичні моделі. В цьому випадку говорять про проведення обчислювального експерименту.

Основними вимогами, що пред'являються до експерименту, є відтворюваність результатів експерименту і керованість об'єктами процесу.

Експериментальні дослідження можуть бути пасивними і активними. Якщо результати виходять при випадкових змінах вхідних параметрів, то говорять про пасивний експеримент. Якщо вхідні параметри змінюються по складеному експериментатором плану, то говорять про активний експеримент [31]. Необхідно відзначити, що абсолютно керованих об'єктів

дослідження не існує. Це пояснюється тим, що на об'єкт діють некеровані чинники, які впливають на відтворюваність результатів експерименту. Якщо всі чинники, що діють, є некерованими, то основним інструментом для дослідження поведінки об'єкту є пасивний експеримент.

Нижче буде розглянута тільки методика проведення і обробки отриманих даних для активних експериментів.

При проведенні експериментальних досліджень чинники можуть бути представлені в натуральних і кодованих величинах. Для виконання кодування використовують формулу

$$x_i = \frac{X_i - X_{i0}}{h_i}, \quad (4.6.16)$$

де x_i - значення чинника в кодованих змінних (може приймати значення $-1, 0, +1$); X_i - значення чинника на одному з рівнів в натуральних величинах; X_{i0} - значення чинника на нульовому (основному) рівні в натуральних величинах; h_i - інтервал варіювання; i - номер чинника.

При плануванні експерименту основна увага повинна приділятися вибору оптимального плану, коли при мінімальному числі дослідів дослідник отримує максимальну інформацію про об'єкт. Так для лінійних моделей, при числі чинників $k \leq 5$ вельми ефективним є повний факторний експеримент (ПФЕ) або дробовий факторний експеримент (ДФЕ). Дані плани володіють ортогональністю (сума почлених творів два будь-яких стовпців матриці рівна нулю), ротатабельністю (рух від центру експерименту на всіх напрямках є рівнозначним) і симетричністю щодо центру експерименту.

Для моделей другого порядку широко використовуються ортогональні, ротатабельні (Боксу) D -оптимальні плани (Кифера). Вельми характерним є число дослідів N у плані. Так для ортогонального плану для $k=2$ $N=9$, для $k=3$ $N=15$, для $k=4$ $N=25$. Для ротатабельного плану (Боксу) при $k=2$ $N=13$, при $k=3$ $N=20$, при $k=4$ $N=31$. Для D -оптимального плану

(Кифера) при $k=2$ $N=9$, при $k=3$ $N=26$, при $k=4$ $N=72$.

Зупинимося на розгляді ПФЕ (у ній реалізуються всі можливі комбінації даних рівнів чинників). Загальне число дослідів визначається з виразу

$$N = 2^k. \quad (4.6.17)$$

Графічна інтерпретація ПФЕ для двох чинників показана на рис.4.6.2 а. Матриця ПФЕ включає: стовпці, відповідні кожному чиннику; стовпці, відповідні взаємодіям чинників. Число рядків визначається по (4.6.17). У таблиці 4.6.2 показана матриця ПФЕ для двох чинників.

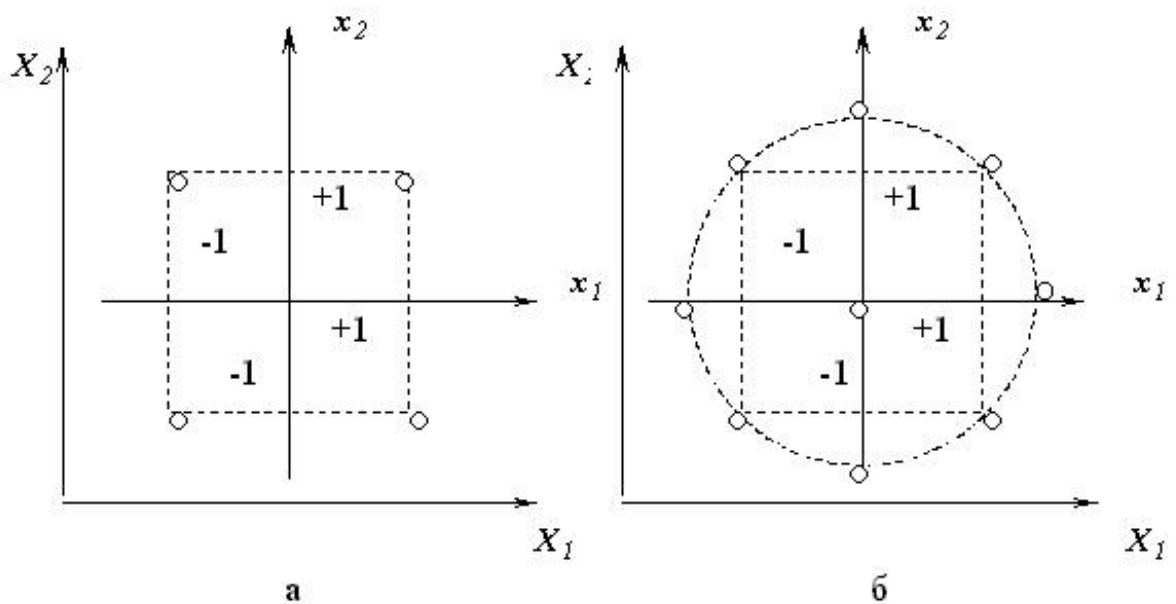


Рис.4.6.2. Графічна інтерпретація ПФЕ і ротатабельного плану для 2 чинників

Таблиця 4.6.2. Матриця ПФЕ для двох чинників

Номер досвіду	Чинники		
	x_1	x_2	x_1x_2
1	+	+	+
2	-	+	-
3	+	-	-
4	-	-	+

Рівняння регресії для двох і трьох чинників мають вигляд [32]

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2, \quad (4.6.18)$$

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3.$$

Коефіцієнти в рівняннях регресії (4.6.18) визначаються по наступних

формулах

$$b_0 = \frac{\sum^N \bar{y}_u}{N}, \quad b_i = \frac{\sum^N x_{iu} \bar{y}_u}{N}, \quad b_{ij} = \frac{\sum^N x_{iu} x_{ju} \bar{y}_u}{N}, \quad b_{ijk} = \frac{\sum^N x_{iu} x_{ju} x_{ku} \bar{y}_u}{N}, \quad (4.6.19)$$

де \bar{y}_u - середньоарифметичне значення критерію оптимізації для кожного досвіду в матриці (залежить від числа повторних вимірювань).

При збільшенні числа чинників $k > 5$ проведення експерименту за планом ПФЕ стає вельми трудомістким. В цьому випадку використовують тільки певну частину ПФЕ. Такий підхід називається проведенням експерименту по матриці ДФЕ. Дробові репліки підрозділяються на регулярних і нерегулярних. Перші виходять з матриці ПФЕ діленням її на число частин, кратне 2 в якому-небудь ступені [32]. При постановці ДФЕ виходять з припущення, що коефіцієнти регресії при взаємодіях вищих порядків (потрійних і вище) незначучо відрізняються від нуля. Тоді стає можливим використовувати в плані ПФЕ стовпці, що відносяться до взаємодій вищих порядків, для оцінки додаткових лінійних ефектів.

Оцінка значучості коефіцієнтів регресії (4.6.19) базується на визначенні довірчого інтервалу (коефіцієнт буде значущим якщо його абсолютна величина більше довірчого інтервалу), який рівний $2\Delta b_i$. Величина Δb_i визначається по формулі

$$\Delta b_i = \frac{t \cdot S_{\{y\}}}{\sqrt{N_1 n}}, \quad S_{\{y\}} = \sqrt{\frac{\sum^N \sum^1 (y_{uj} - \bar{y}_u)^2}{N(n-1)}}, \quad (4.6.20)$$

де n - число спостережень в окремому досвіді; N - число рядків матриці планування експерименту; t - критерій Стьюдента (див. таблицю 4.6.1); N_1 - число дослідів, що враховуються при розрахунку коефіцієнта.

Ротатабельне планування другого порядку використовується у разі, коли поверхня відгуку носить явно нелінійний характер. В зв'язку з цим варіювання чинників необхідно здійснювати на трьох і п'яти рівнях. Ротатабельний план виходить з плану ПФЕ шляхом додавання до нього

певної кількості «зоряних» і нульових крапок. «Зоряні» крапки розташовуються на осях на відстані $2^{\frac{k}{4}}$ від нульової крапки, яке називається плечем. На рис.4.6.2 би показана графічна інтерпретація ротатабельного плану другого порядку для двох чинників. У таблиці 4.6.3 представлені основні співвідношення між числом чинників і загальним числом дослідів для ротатабельного плану другого порядку [32].

Таблиця 4.6.3. Залежність числа дослідів від числа чинників для ротатабельного плану другого порядку

Число чинників k	Число точок «ядра» n_j	Число «зоряних» крапок	Число нульових крапок	Величина плеча для «зоряних» крапок	Загальне число дослідів N
2	4	4	5	1,414	13
3	8	6	6	1,682	20
4	16	8	7	2,000	31
5	32	10	10	2,378	52

У таблиці 4.6.4 представлена матриця планування експерименту для ротатабельного плану другого порядку для 2 чинників

Таблиця 4.6.4. Матриця планування експерименту

Номер досвіду	Чинники		
	x_1	x_2	x_1x_2
1	+	+	+
2	-	+	-
3	+	-	-
4	-	-	+
5	-1,414	0	0
6	+1,414	0	0
7	0	-1,414	0
8	0	+1,414	0

9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0

Коефіцієнти в рівнянні регресії для ротатбельного плану визначаються по наступних формулах [32]

$$b_0 = a_1 \sum_{j=1}^N y_j - a_2 \sum_{i=1}^k \sum_{j=1}^N x_{ij}^2 y_j, \quad b_i = a_3 \sum_{j=1}^N x_{ij} y_j, \quad (4.6.21)$$

$$b_{ij} = a_4 \sum_{j=1}^N x_{ij} x_{jj} y_j, \quad b_{ii} = a_5 \sum_{j=1}^N x_{ij}^2 y_j + a_6 \sum_{i=1}^k \sum_{j=1}^N x_{ij}^2 y_j - a_7 \sum_{j=1}^N y_j,$$

де $a_1 \dots a_7$ - постійні коефіцієнти, визначувані з таблиці 4.6.5.

Таблиця 4.6.5. Значення коефіцієнтів $a_1 \dots a_7$ у виразі (4.6.21)

Число чинників k	Число дослідів N	Коефіцієнти						
		a_1	a_2	a_3	a_4	a_5	a_6	a_7
2	13	0,2	0,1	0,125	0,25	0,125	0,0187	0,1
3	20	0,1663	0,0568	0,0732	0,125	0,0625	0,0069	0,0568
4	31	0,1428	0,0357	0,0417	0,0625	0,0312	0,0037	0,0357
5	52	0,0988	0,0191	0,0231	0,0312	0,0156	0,0015	0,0191

Величина довірчого інтервалу для коефіцієнтів в рівнянні регресії при ротатбельном плануванні другого порядку визначається по формулах

$$\Delta b_0 = \pm 2S_{\{b_0\}}, \quad \Delta b_i = \pm 2S_{\{b_i\}}, \quad \Delta b_{ii} = \pm 2S_{\{b_{ii}\}}, \quad \Delta b_{ij} = \pm 2S_{\{b_{ij}\}},$$

$$S_{\{b_0\}} = \sqrt{a_8} S_{\{\bar{y}\}}, \quad S_{\{b_i\}} = \sqrt{a_9} S_{\{\bar{y}\}}, \quad S_{\{b_{ii}\}} = \sqrt{a_{10}} S_{\{\bar{y}\}}, \quad S_{\{b_{ij}\}} = \sqrt{a_{11}} S_{\{\bar{y}\}}, \quad (4.6.22)$$

$$S_{\{\bar{y}\}} = \frac{S_{\{y\}}}{\sqrt{n}}, \quad S_{\{y\}} = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^n (y_{ij} - \bar{y}_i)^2}{N(n-1)}}.$$

Значення коефіцієнтів $a_8 \dots a_{11}$ приведені в таблиці 4.6.6 [32].

Таблиця 4.6.6. Значення коефіцієнтів $a_8 \dots a_{11}$ у виразі (4.6.22)

Число чинників k	Число дослідів N	Коефіцієнти			
		a_8	a_9	a_{10}	a_{11}
2	13	0,2	0,125	0,1438	0,25
3	20	0,1663	0,0732	0,0694	0,125
4	31	0,1428	0,0417	0,0341	0,0625

При ортогональному плануванні другого порядку матрицю планування ПФЕ добудовують шляхом введення нульової крапки і певного числа «зоряних» крапок. У таблиці 4.6.7 приведені основні характеристики ортогонального плану для різного числа чинників.

Таблиця 4.6.7. Ортогональні плани другого порядку

Число чинників k	Число точок «ядра» n_j	Число «зоряних» крапок	Число нульових крапок	Величина плеча для «зоряних» крапок	Загальне число дослідів N
2	4	4	1	1,000	9
3	8	6	1	1,215	15
4	16	8	1	1,414	25

Коефіцієнти в рівнянні регресії для ортогонального планування другого порядку визначаються по формулах

$$b_i = \frac{\sum_1^{N_1} x_{iu} y_u}{N_1}, \quad b_{ij} = \frac{\sum_1^{N_2} x_{iu} x_{ju} y_u}{N_2}, \quad b_{ii} = \frac{\sum_1^N (x'_{iu})^2 y_u}{\sum_1^N (x'_{iu})^2}, \quad (4.6.23)$$

$$b_0 = \frac{\sum_1^N y_u}{N} - \frac{\sum_1^N x_{iu}^2}{N} \sum_1^k b_{ii}, \quad (x'_{iu})^2 = x_{iu}^2 - \frac{\sum_1^N x_{iu}^2}{N},$$

де N_1 - число дослідів не в нульових крапках; N_2 - число відмінних від нуля парних творів чинників.

При ортогональному плануванні для визначення довірчого інтервалу $2\Delta b_i$ користуються наступною залежністю

$$\Delta b_i = \pm 2S_{\{b_i\}}, \quad S_{\{b_i\}} = \frac{S_{\{\bar{y}\}}}{\sqrt{A}}, \quad S_{\{\bar{y}\}} = \frac{S_{\{y\}}}{\sqrt{n}}, \quad S_{\{y\}} = \sqrt{\frac{\sum_{I=1}^N \sum_{I=1}^n (y_{uj} - \bar{y}_u)^2}{N(n-1)}}. \quad (4.6.24)$$

Значення коефіцієнта A визначаємо з таблиці 4.6.8 [32].

Таблиця 4.6.8. Значення коефіцієнта A у виразі (4.6.24)

Число чинників k	A			
	b_0	b_i	b_{ij}	b_{ii}
2	9	6	4	2
3	15	10,94	8	4,34
4	25	20	16	8

Перевірка отриманих рівнянь регресії здійснюється за допомогою критерію Фішера. Розрахункове значення критерію Фішера визначається по формулі

$$F_p = \frac{S_{ad}^2}{S_{\{y\}}^2}, \quad (4.6.25)$$

де S_{ad}^2 - дисперсія адекватності рівняння регресії; $S_{\{y\}}^2$ - дисперсія відтворюваності.

Для ПФЕ

$$S_{ad}^2 = \frac{\sum_{I=1}^N n(\bar{y}_u - \hat{y}_u)^2}{N - k - 1}, \quad S_{\{y\}}^2 = \frac{\sum_{I=1}^N \sum_{I=1}^n (y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

Для ортогонального планування другого порядку

$$S_{ad}^2 = \frac{\sum_{I=1}^N n(\bar{y}_u - \hat{y}_u)^2}{N - \frac{(k+2)(k+1)}{2}}, \quad S_{\{y\}}^2 = \frac{\sum_{I=1}^N \sum_{I=1}^n (y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

Для ротатбельного планування другого порядку

$$S_{ad}^2 = \frac{\sum_I^N n(\bar{y}_u - \hat{y}_u)^2}{N - \frac{(k+2)(k+1)}{2} - (n_0 - 1)}, \quad S_{\{y\}}^2 = \frac{\sum_I^N \sum_I^n (y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

де n_0 - число повторностей нульового досвіду.

На наступному етапі, після визначення по (4.6.25) розрахункового значення критерію Фішера, визначаємо табличне значення критерію Фішера F_T .

Число мір свободи для більшої дисперсії визначається по формулах: для ПФЕ

$$f_{ad} = N - k - 1,$$

для ортогонального планування другого порядку

$$f_{ad} = N - \frac{(k+2)(k+1)}{2},$$

для ротатабельного планування другого порядку

$$f_{ad} = N - \frac{(k+2)(k+1)}{2} - (n_0 - 1).$$

Число мір свободи для меншої дисперсії визначається по формулі

$$f_e = N(n-1).$$

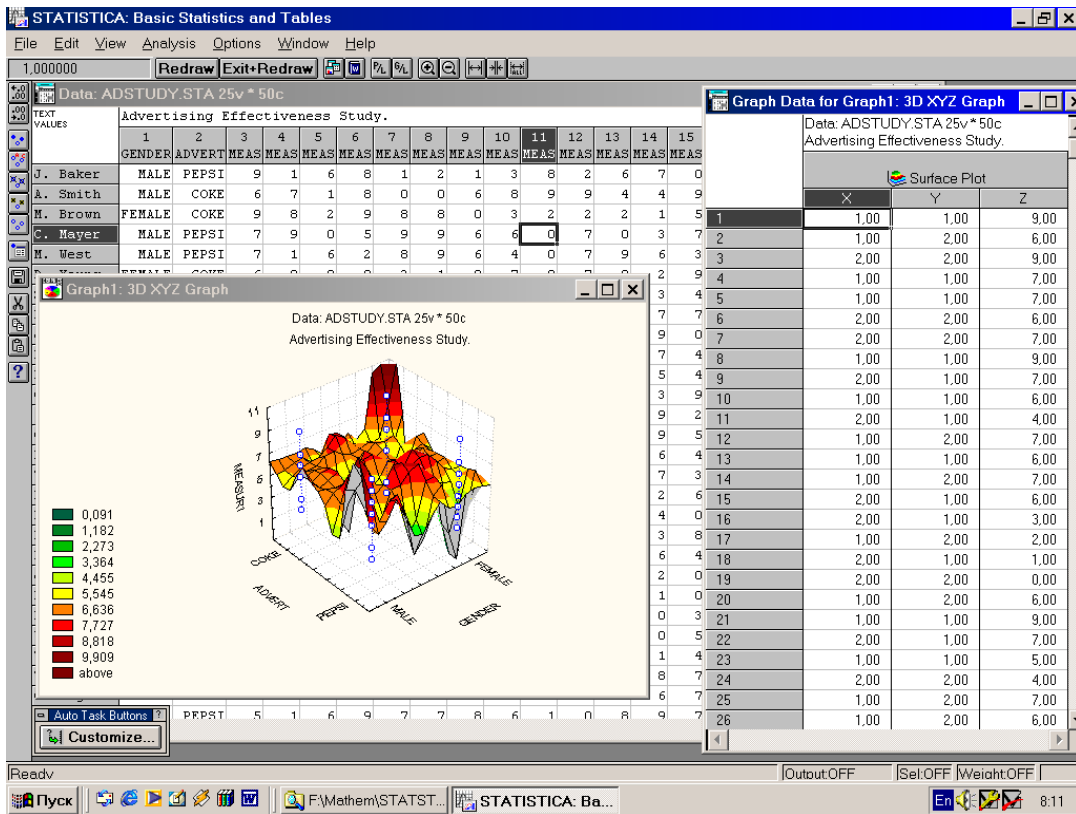
Для того, щоб регресійна модель, при вибраній довірчій вірогідності, адекватно описувала досліджуваний процес необхідно виконання умови

$$F_p < F_T. \quad (4.6.26)$$

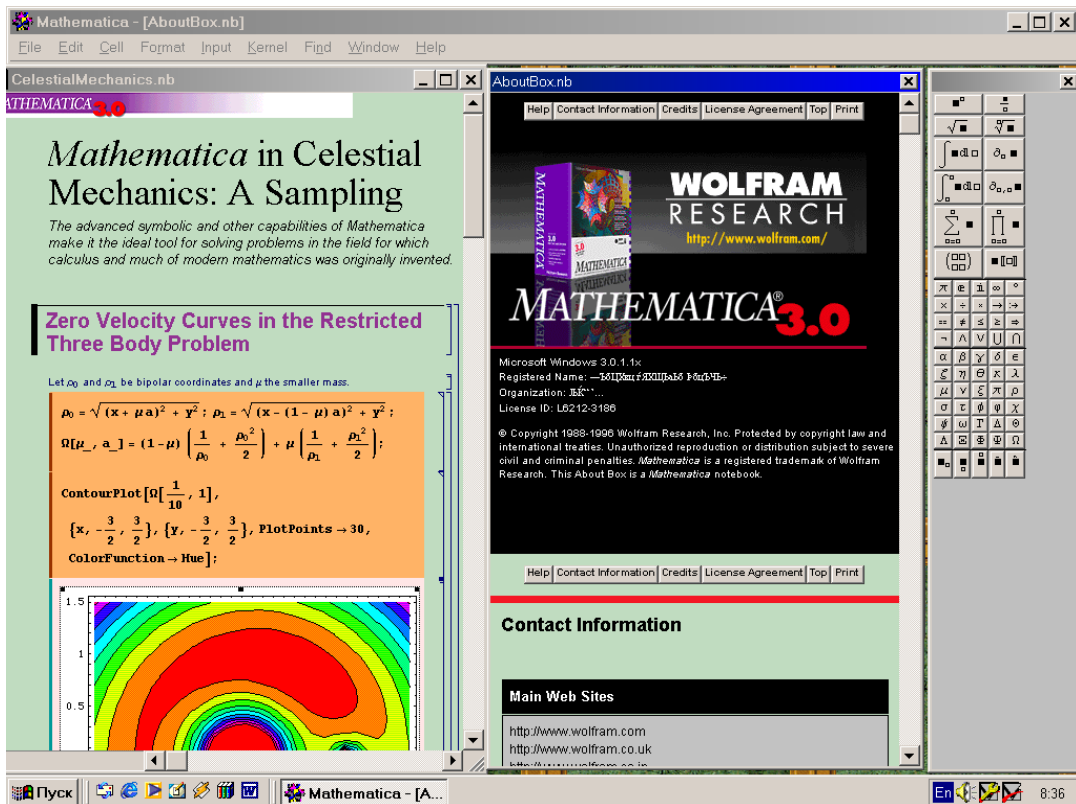
З розвитком інформаційних технологій велике значення придбали різні прикладні програмні пакети для статистичної обробки даних і реалізації активного планування експерименту. На рис.4.6.3а,б показані робочі вікна пакетів *STATISTICA 5.0* і *Mathematica3* [33, 34]. На рис.4.6.4а,б показані робочі вікна пакетів *SPSS* і *STATGRAPHICS* [35].

Статистичний пакет *STATISTICA* [33] (проводиться фірмою Stat Soft Inc. США) включає наступні статистичні модулі: непараметрична статистика, дисперсійний аналіз, множинна регресія, нелінійне оцінювання, аналіз тимчасових рядів, кластерний і факторний аналіз, функціональний аналіз дискримінанта, канонічна кореляція, моделювання структурних рівнянь,

аналіз процесів, планування експерименту і ін.

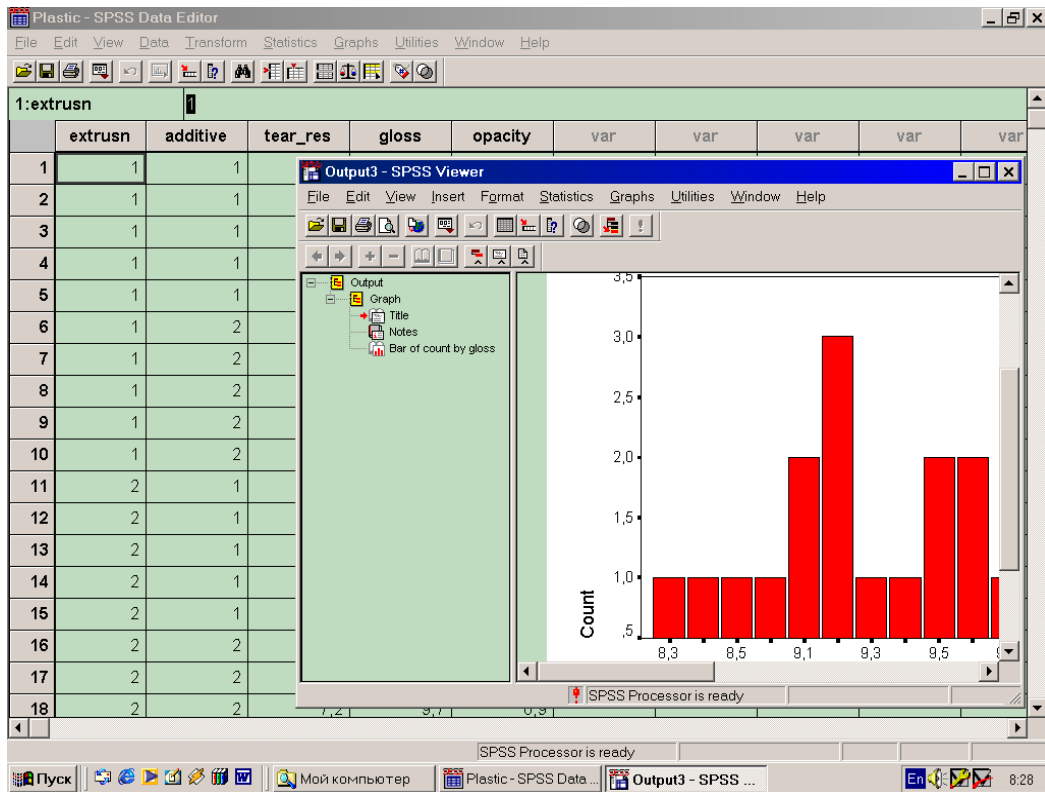


а

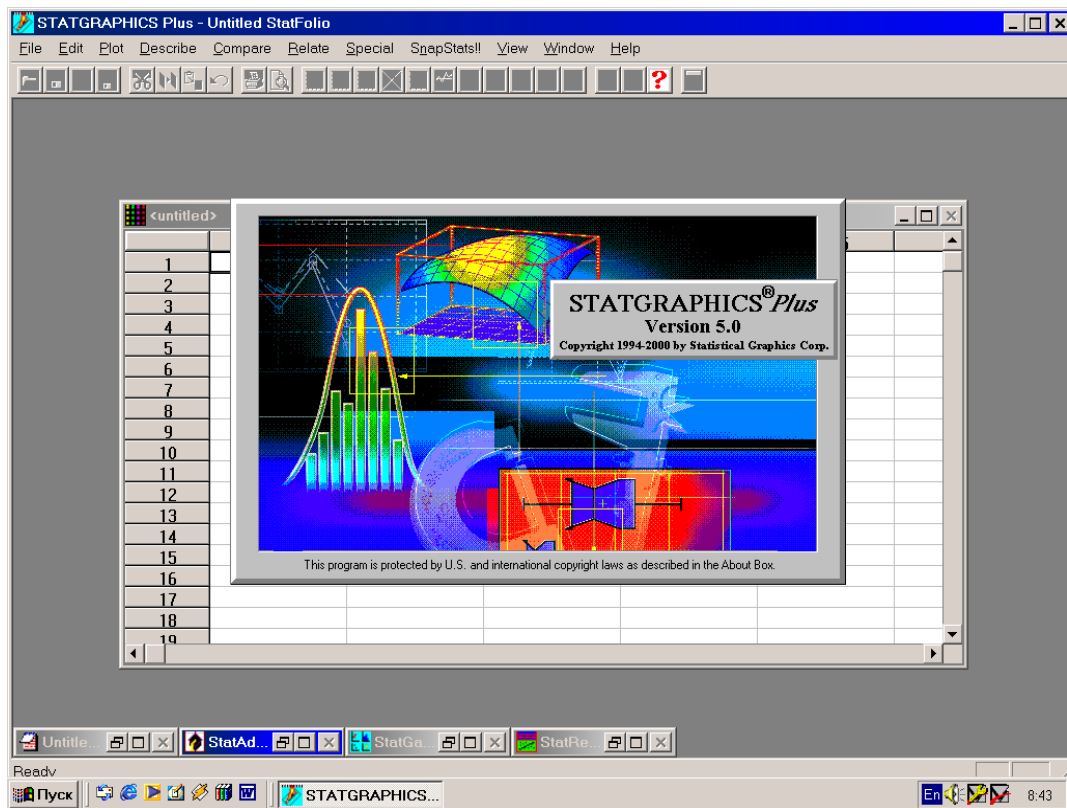


би

Рис.4.6.3. Робочі вікна пакетів STATISTICA 5.0 і Mathematica3



а



би

Рис.4.6.4. Робочі вікна пакетів *SPSS* і *STATGRAPHICS*

У 1995 році вийшла одна з останніх версій даного пакету *STATISTICA 5.0* для Windows, в якій статистичний аналіз експериментальних даних розбитий на наступні етапи: введення даних; використання певної процедури статистичної обробки даних; виведення результатів на дисплей або друк.

Статистична обробка даних в пакеті символічної математики *Mathematica3* міститься в пакеті розширення системи *STATISTI* який орієнтований на статистичну обробку даних [34]. Даний пакет містить наступні функції: *Mean*- повертає середнє значення даних $\{x_1, x_2, \dots, x_n\}$; *Median*- повертає медіану даних $\{x_1, x_2, \dots, x_n\}$; *StandardDeviation* - повертає стандартне середньо квадратичне відхилення для даних $\{x_1, x_2, \dots, x_n\}$; *Variance*- повертає варіацію для даних $\{x_1, x_2, \dots, x_n\}$.

Апроксимація і інтерполяція експериментальних даних здійснюється з використанням наступних функцій: *InterpolatingFunction* - повертає функцію, значення якої у вузлових точках визначаються за допомогою інтерполяції; *InterpolatingPolynomial* - повертає інтерполяційний поліном; *Interpolation* - повертає функцію, що наближає; *InterpolationOrder* - повертає функцію (статечною поліном), що наближає, з вказаним ступенем.

Для вирішення завдань лінійної і нелінійної регресії використовується функція *Fit*. Отримані результати забезпечують мінімальну середньо квадратичну погрішність між значеннями регресійної залежності і експериментальними даними для певних значень варійованого параметра.

Серед прикладних пакетів для обробки експериментальних даних можна виділити *SPSS* (розробник SPSS Inc.) і *STATGRAPHICS* (STATistical Graphics System) – розробник Manugistic Inc. [35]. Останній пакет *STATGRAPHICS Plus for Windows*. Даний пакет має модульну структуру: *Describe*- містить статистичні методи аналізу для одно- і

багатьох змінних; *Compare* - містить процедури одно- і багатofакторного дисперсійного аналізу, методи порівняння декількох вибірок даних; *Re late* - включає операції простого поліноміального і множинного регресійного аналізу; *Special* - включає контроль якості, планування експерименту, аналіз тимчасових рядів, багатовимірні методи, розширений регресійний аналіз. *Design of Experiment* (Планування експерименту) включає повні і дробові плани, плани Плакетта-Бурмена, блокові плани, центральні композиційні плани, трирівневі факторіали, плани Бокса-Бенкена, плани Дрейпера-Лана, прості центроїдні і латинські плани.

Описані вище пакети прикладних програм для статистичної обробки експериментальних даних і активного планування експерименту дозволяють значно скоротити час отримання математичної моделі.

У справжній роботі нами для аналізу геометричної форми поверхні відгуку використовувалися інваріанти поверхонь другого порядку [1]. Їх використання дозволяє досягти поставленої мети без приведення рівнянь до канонічного вигляду.

Суть методу полягає в тому, що для безлічі точок тривимірного простору, координати яких в декартовій системі задовольняють рівнянню алгебри 2-го ступеня

$$a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + 2a_{23}yz + 2a_{14}x + 2a_{24}y + 2a_{34}z + a_{44} = 0, \quad (4.6.42)$$

визначають основні інваріанти S , T , ϕ , U і напів інваріантні (семі інваріанти) U_2 , UU . Потім по таблиці 4.6.15 визначають, який геометричний образ має поверхню відгуку для тієї або іншої регресійної залежності (4.6.27.4.6.41).

Основні інваріанти і напів інваріантні визначаються як визначники і їх суми відповідних матриць коефіцієнтів многочлена алгебри. Формули для розрахунку приведені у вигляді системи (4.6.43). Перетворимо регресійні залежності (4.6.27).(4.6.41) до вигляду (4.6.42), для чого підставимо в

останні значення кута обхвату і швидкості руху нитки, які відповідатимуть своїм мінімальним значенням (див.

$$\begin{aligned}
 S &= a_{11} + a_{22} + a_{33}; \\
 \delta &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix}; \\
 T &= \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} \\ a_{13} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{23} & a_{33} \end{vmatrix}; \\
 \Delta &= \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{vmatrix}; \\
 \Delta' &= \begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{23} & a_{33} & a_{34} \\ a_{24} & a_{34} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} & a_{14} \\ a_{13} & a_{33} & a_{34} \\ a_{14} & a_{34} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{12} & a_{22} & a_{24} \\ a_{14} & a_{24} & a_{44} \end{vmatrix}; \\
 \Delta'' &= \begin{vmatrix} a_{11} & a_{14} \\ a_{14} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{24} \\ a_{24} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{33} & a_{34} \\ a_{34} & a_{44} \end{vmatrix}.
 \end{aligned} \tag{4.6.43}$$

план проведення експерименту табл. 4.6.11).

При підстановці $\mu=2,44$ радий швидкість $x=0,9$ м/с. Такі значення даних величин вибрані нами виходячи з умови найменшої величини натягнення ведучої галузь нитки при варіюванні цих чинників.

Виконуючи перетворення залежностей (4.6.27).(4.6.41) до (4.6.42), вважали, що $x=R$, $y=P0$, $z=P$. Відповідним чином були розподілені і коефіцієнти при конкретних величинах.

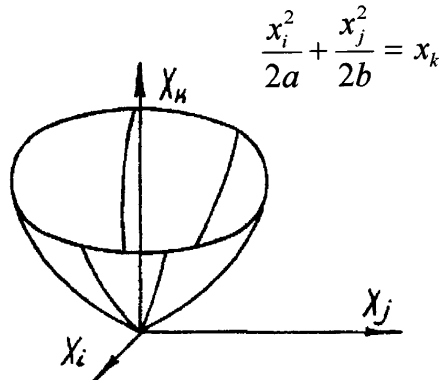
Результати розрахунку по формулах (4.6.43) інваріантів і напів інваріантних також приведені в таблиці 4.6.15.

Аналіз отриманих результатів дозволив встановити, що всі геометричні образи поверхонь відгуку зводяться до еліптичного параболоїда, параболічного циліндра і пари паралельних площин (див. мал. 4.6.9).

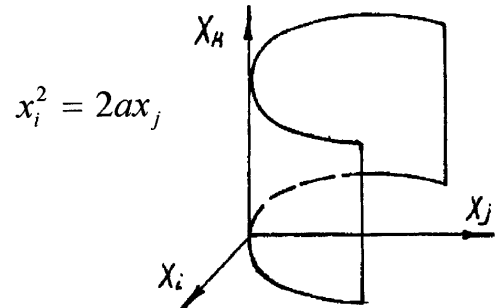
З цього можна зробити вивід - при перетині поверхонь відгуку площинами у нас виходитимуть параболі і прямі лінії. Ця обставина враховувалася нижче при дослідженні впливу радіусу кривизни такою, що направляє,

швидкості руху нитки, натягнення веденої гілки на

Варианти III, IV, VII



Варианти V-VII, IX, X, XII-XV



Варианти I, II, VII, XI

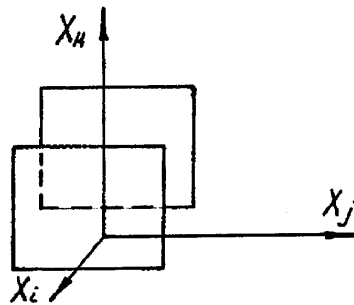


Рис.4.6.9. Геометричні образи поверхонь відгук

натягнення ведучої гілки.

Такий диференційований підхід до вивчення регресійних залежностей (4.6.27) -(4.6.41) необхідний, оскільки велике число чинників утрудняє процес аналізу.

4.7. ПИТАННЯ ОПТИМІЗАЦІЇ

Найважливішою умовою науково-технічного прогресу є пошук якнайкращого варіанту, з погляду наміченої мети, при проектуванні складних механічних систем, економічному плануванні, розподілі обмежених ресурсів, вдосконаленні технологічних процесів. Відшукування найменшого (найбільшого) значення цільової функції є основним

завданням оптимізації. Властивості цільової функції впливають на постановку завдання оптимізації і методи її дослідження. Завдання оптимізації можна трактувати як екстремальні завдання, методи вирішення яких розглядаються в теорії екстремальних завдань.

У багатьох завданнях економічного змісту зустрічаються опуклі і лінійні функції і множини. У 30-і роки радянським математиком Канторовічем Л.В. були розроблені нові напрями в теорії екстремальних завдань – лінійне і опукле програмування.

Теорія оптимального управління, доповнюючи розділ варіаційного числення, дозволяє вирішувати завдання управління технологічними процесами, приладами, системами.

Питаннями, присвяченими теорії і методам знаходження екстремумів функцій багатьох змінних, займається математичне програмування. При цьому, необхідно мати додаткові обмеження на ці змінні, які мають форму рівності і (або) нерівностей [36]. Залежно від властивостей цільової функції і додаткових обмежень, що накладаються на дані змінні, математичне програмування розбивається на лінійне і нелінійне програмування, дискретне, параметричне і стохастичне програмування. У разі, коли пошук оптимального рішення (залежно від виду цільової функції і обмежень) здійснюється у вигляді багатокрокового процесу, то говорять про динамічне програмування.

Зупинимося на загальних питаннях постановки і методів вирішення одновимірних завдань оптимізації. Математична формалізація завдання, в даному випадку, може бути записана таким чином: визначити найменше (найбільше) значення цільової функції $f(x)$ і значення змінної x за умови коли $x \in X$ (X - деяка множина).

Для цього скористаємося теоремами Вейерштраса і Ферма [14]. Теорема Вейерштраса свідчить, що всяка цільова функція

$f(x)$ безперервна на відрізку $[a, b]$ приймає на цьому відрізку своє найменше $f(x_1)$ і найбільше $f(x_2)$ значення ($x_1 \in [a, b], x_2 \in [a, b]$)

$$f(x_1) \leq f(x) \leq f(x_2). \quad (4.7.1)$$

У загальному випадку, цільова функція на заданому відрізку може досягати найменшого (найбільшого) значення в декількох крапках (наприклад, періодичні функції). Хай цільова функція $f(x)$ визначена на відрізку $[a, b]$. Можна стверджувати, що крапка x_1 (або x_2) доставляє локальний мінімум (максимум) якщо можна вказати таке ε що для всіх крапок x з відрізка $[a, b]$ для яких $|x - x_1| < \varepsilon$ (або $|x - x_2| < \varepsilon$), виконується нерівність $f(x) \geq f(x_1)$ (або $f(x) \leq f(x_2)$).

Для випадку, коли цільова функція безперервна не на відрізку, а на всій осі, то можна скористатися наслідком з теореми Вейерштраса: якщо цільова функція $f(x)$ безперервна на всій прямій і якщо $\lim_{x \rightarrow \infty} f(x) = \lim_{x \rightarrow -\infty} f(x) = \infty$ те рішення задачі пошуку мінімуму (максимуму) цільової функції існує. Дане слідство можна використовувати і у разі, коли цільова функція безперервна на промені вигляду $a \leq x < \infty$ або $a < x < \infty$.

Теорема Ферма [14] формулюється таким чином: якщо цільова функція диференційована в точці x_1 (для максимуму x_2) і якщо цільова функція має локальний екстремум в цій крапці, то $f'(x_1) = 0$ (або $f'(x_2) = 0$).

Крапки, для яких $f'(x) = 0$ називаються стаціонарними. Стаціонарні крапки спільно з кінцевими крапками називаються критичними.

Співвідношення $f'(x_1) = 0$ (або $f'(x_2) = 0$) є лише необхідною умовою екстремуму. Спираючись на теореми Вейерштраса і Ферма можна стверджувати, що якщо відрізок $[a, b]$ кінцевий, цільова функція $f(x)$ безперервна на відрізку $[a, b]$ і диференційована у внутрішніх крапках $a < x < b$ те рішення знаходиться серед критичних крапок.

Вирішуючи рівняння

$$f'(x) = 0, \quad (4.7.2)$$

визначаємо дійсне його коріння, а також ті крапки, в яких похідна не існує. Розміщуємо всі критичні крапки в порядку зростання їх абсцис. У кожному з отриманих інтервалів вибирають довільну крапку і встановлюють в цій крапці знак першої похідної. Розглядають знаки $f'(x)$ у двох сусідніх інтервалах, послідовно переходячи зліва направо від першого інтервалу до останнього. Якщо похідна $f'(x)$ міняє знак з «-» на «+», то функція має мінімум в даній критичній крапці. Якщо знаки змінюються з «+» на «-», то функція має в критичній крапці максимум. Якщо ж в двох сусідніх інтервалах похідна $f'(x)$ не міняє знаку, то екстремуму в даній критичній крапці немає.

Приведені вище міркування безпосередньо пов'язані з поняттям опуклої функції. Функція називається опуклою, якщо для будь-якої хорди, що сполучає дві точки графіка цієї функції, її графік в проміжних крапках лежить нижче за цю хорду.

Перейдемо до дослідження на екстремум функції $f(x_1, x_2, \dots, x_n)$ декілька змінних x_1, x_2, \dots, x_n вважаючи, що вони незалежні між собою. Необхідною умовою існування екстремуму функції $f(x_1, x_2, \dots, x_n)$ є наступне: якщо функція $f(x_1, x_2, \dots, x_n)$ досягає екстремуму при $x_1 = x_{10}, x_2 = x_{20}, \dots, x_n = x_{n0}$ то кожна приватна похідна першого порядку функції $f(x_1, x_2, \dots, x_n)$ по відповідній незалежній змінній x_1, x_2, \dots, x_n звертається в нуль при цих значеннях аргументів або не існує [37]

$$\begin{aligned} & \left[\frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} \right]_{\substack{x_1=x_{10} \\ x_2=x_{20} \\ \dots \\ x_n=x_{n0}}} = 0, \\ & \left[\frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} \right]_{\substack{x_1=x_{10} \\ x_2=x_{20} \\ \dots \\ x_n=x_{n0}}} = 0, \\ & \dots \dots \dots \dots \dots \dots \dots \\ & \left[\frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} \right]_{\substack{x_1=x_{10} \\ x_2=x_{20} \\ \dots \\ x_n=x_{n0}}} = 0. \end{aligned} \tag{4.7.3}$$

Значення $x_{10}, x_{20}, \dots, x_{n0}$ називаються критичними точками функції $f(x_1, x_2, \dots, x_n)$. При дослідженні значення функції в критичних крапках можна підійти до визначення достатніх умов існування екстремуму. Для виведення достатніх умов використовують формулу Тейлора для представлення функції у вигляді суми многочлена ступеня n ($n=0, 1, 2, \dots$) і залишкового члена [14]. Хай в деякій області, що містить крапку $M_0(x_{10}, x_{20}, \dots, x_{n0})$ функція $f(x_1, x_2, \dots, x_n)$ має безперервні приватні похідні до третього порядку включно. Хай крапка $M_0(x_{10}, x_{20}, \dots, x_{n0})$ є критичною крапкою для функції $f(x_1, x_2, \dots, x_n)$ тоді

$$\begin{aligned} & \left[\frac{\partial f(x_{10}, x_{20}, \dots, x_{n0})}{\partial x_1} \right] = 0, \\ & \left[\frac{\partial f(x_{10}, x_{20}, \dots, x_{n0})}{\partial x_2} \right] = 0, \\ & \dots \dots \dots \dots \dots \dots \dots \\ & \left[\frac{\partial f(x_{10}, x_{20}, \dots, x_{n0})}{\partial x_n} \right] = 0. \end{aligned} \tag{4.7.4}$$

Достатні умови існування екстремуму, наприклад для функції $f(x_1, x_2)$ два змінних x_1, x_2 мають вигляд [37]:

функція $f(x_1, x_2)$ має максимум, якщо

$$\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1 \partial x_2} \right]^2 > 0, \quad \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} < 0,$$

функція $f(x_1, x_2)$ має мінімум, якщо

$$\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1 \partial x_2} \right]^2 > 0, \quad \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} > 0,$$

функція $f(x_1, x_2)$ не має ні максимуму, ні мінімуму, якщо

$$\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1 \partial x_2} \right]^2 < 0,$$

функція $f(x_1, x_2)$ може мати екстремум в крапці $M_0(x_{10}, x_{20})$ а може і не мати, якщо

$$\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 f(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 f(x_{10}, x_{20})}{\partial x_1 \partial x_2} \right]^2 = 0.$$

У тих випадках, коли змінні x_1, x_2, \dots, x_n є незалежними можна говорити про відшукування абсолютних мінімумів і максимумів. У разі, коли змінні зв'язані один з одним деякими додатковими умовами, говорять про умовні (відносних) екстремуми [14,37]. У загальному випадку змінні x_1, x_2, \dots, x_n зв'язані m ($m < n$) рівняннями

$$\begin{aligned} \varphi_1(x_1, x_2, \dots, x_n) &= 0, \\ \varphi_2(x_1, x_2, \dots, x_n) &= 0, \\ &\dots \dots \dots \dots \dots \dots \\ \varphi_m(x_1, x_2, \dots, x_n) &= 0. \end{aligned} \tag{4.7.5}$$

Необхідні умови існування екстремуму виходять шляхом визначення приватних похідних від функції

$$\begin{aligned} \Phi(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) &= f(x_1, x_2, \dots, x_n) + \lambda_1 \varphi_1(x_1, x_2, \dots, x_n) + \\ &+ \lambda_2 \varphi_2(x_1, x_2, \dots, x_n) + \dots + \lambda_m \varphi_m(x_1, x_2, \dots, x_n) = 0, \end{aligned} \tag{4.7.6}$$

по відповідних змінних x_1, x_2, \dots, x_n . У виразі (4.7.6) $\lambda_1, \lambda_2, \dots, \lambda_m$ називаються множниками Лагранжа. Прирівнюючи нулю виразу для відповідних приватних похідних, отримаємо

$$\frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_1 \partial x_2} \right]^2 = 0.$$

У разі, коли змінні зв'язані додатковими умовами $\sum_{i=1}^m \varphi_i(x_1, x_2, \dots, x_n) = 0$ у вигляді лінійних функцій і нерівностей, що містять увігнуті функції, то можна використовувати умови Куна-Таккера для загального завдання оптимізації у разі нелінійного програмування з обмеженнями у вигляді рівності і нерівностей [38].

Розглянемо загальне завдання нелінійного програмування, коли необхідно мінімізувати функцію $f(x_1, x_2, \dots, x_n)$ при наступних обмеженнях

$$\sum_{i=1}^m \varphi_i(x_1, x_2, \dots, x_n) = 0 \quad \sum_{j=1}^V g_j(x_1, x_2, \dots, x_n) \geq 0$$

Достатність умов Куна-Таккера полягає в наступному. Якщо цільова функція $f(x_1, x_2, \dots, x_n)$ опукла, а всі обмеження у вигляді нерівностей $\sum_{j=1}^V g_j(x_1, x_2, \dots, x_n) \geq 0$ містять увігнуті функції $g_j(x_1, x_2, \dots, x_n)$ а обмеження у вигляді рівності містять лінійні функції $\varphi_i(x_1, x_2, \dots, x_n)$ то якщо існує рішення $(x_{10}, x_{20}, \dots, x_{n0}, u_1, u_2, \dots, u_V, \lambda_1, \lambda_2, \dots, \lambda_m)$ що задовольняє умовам

$$\frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_k} - \frac{\partial}{\partial x_k} \left[\sum_{k_1=1}^V u_{k_1} g_{k_1}(x_1, x_2, \dots, x_n) \right] - \frac{\partial}{\partial x_k} \left[\sum_{k_2=1}^m \lambda_{k_2} \varphi_{k_2}(x_1, x_2, \dots, x_n) \right] = 0,$$

$$k = 1, 2, \dots, n, \quad k_1 = 1, 2, \dots, V, \quad k_2 = 1, 2, \dots, m,$$

$$\sum_{i=1}^m \varphi_i(x_1, x_2, \dots, x_n) = 0, \quad \sum_{j=1}^V g_j(x_1, x_2, \dots, x_n) \geq 0, \quad (4.7.8)$$

$$u_{k_1} g_{k_1}(x_1, x_2, \dots, x_n) = 0, \quad u_{k_1} \geq 0,$$

то сукупність $x_{10}, x_{20}, \dots, x_{n0}$ є оптимальним рішенням задачі нелінійного програмування [38].

У разі, коли цільова функція $f(x_1, x_2, \dots, x_n)$ є лінійною функцією

$$\text{змінних } x_1, x_2, \dots, x_n \text{ а умови } \sum_{i=1}^m \varphi_i(x_1, x_2, \dots, x_n) = 0 \quad \sum_{j=1}^V g_j(x_1, x_2, \dots, x_n) \geq 0$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \quad (4.7.11)$$

Крім того, необхідно, щоб вказана сукупність змінних x_1, x_2, \dots, x_n задовольняла системі рівнянь (4.7.10). Оптимальним рішенням, зі всіх допустимих рішень, буде таке, при якому значення цільової функції (4.7.9) прийме мінімальне (максимальне) значення. У разі коли сукупність змінних x_1, x_2, \dots, x_n задовольняє (4.7.10), але не задовольняє (4.7.11) або рівняння системи (4.7.10) суперечать один одному, то говорять, що завдання лінійного програмування не має допустимих рішень. Коли цільова функція $f(x_1, x_2, \dots, x_n)$ в області допустимих рішень не обмежена знизу, то серед допустимих рішень основної задачі лінійного програмування немає оптимального рішення.

У разі, коли обмеження, що визначають допустимі значення змінних x_1, x_2, \dots, x_n задані у вигляді лінійних нерівностей, то говорять про завдання лінійного програмування з обмеженнями-нерівностями

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - c_1 &\geq 0, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - c_2 &\geq 0, \\ &\dots \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n - c_m &\geq 0. \end{aligned} \quad (4.7.12)$$

Використовуючи (4.7.12) необхідно знайти таку сукупність ненегативних значень змінних x_1, x_2, \dots, x_n щоб набуті мінімального (максимальне) значення цільової функції (4.7.9). Дану задачу вирішують шляхом переходу до основного завдання лінійного програмування шляхом введення нових базисних (додаткових) змінних

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - c_1, \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - c_2, \\ &\dots \dots \dots \dots \dots \dots \dots \dots \dots \\ y_m &= a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n - c_m. \end{aligned} \quad (4.7.13)$$

Базисні змінні y_1, y_2, \dots, y_m і початкові x_1, x_2, \dots, x_n повинні задовольняти нерівностям

$$y_j \geq 0, x_k \geq 0, \quad j = 1, \dots, m, k = 1, \dots, n. \quad (4.7.14)$$

Отже, необхідно знайти ненегативні значення $n+m$ змінних $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m$

Геометрична інтерпретація системи лінійних нерівностей (4.7.12) (залежно від величини коефіцієнтів $a_{11}, \dots, a_{mn}, c_1, \dots, c_m$) може бути представлена трьома варіантами (для наочності обмежимося двома змінними x_1 і x_2), які представлені на рис.4.7.1 а-в.

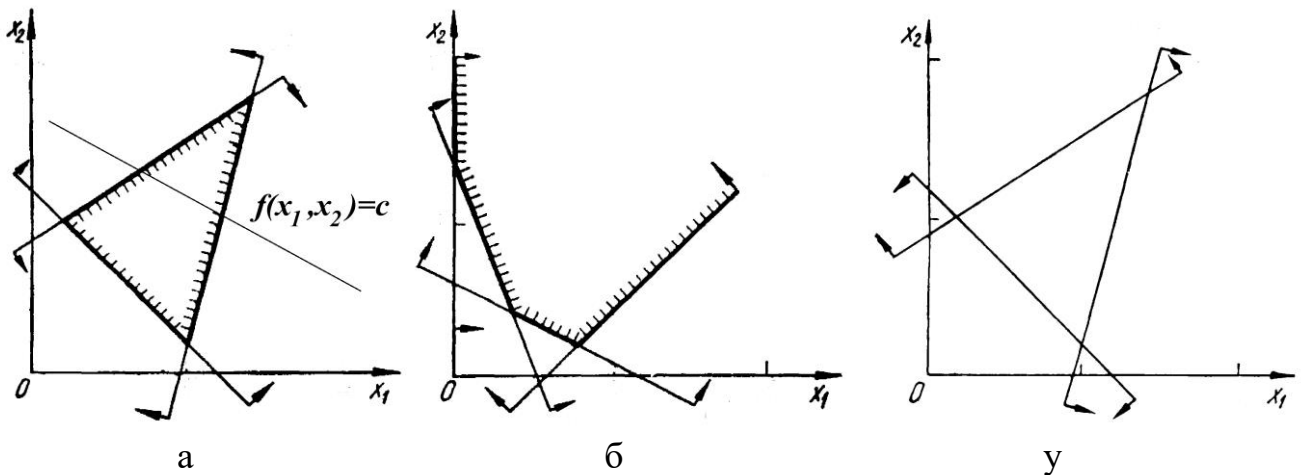


Рис.4.7.1. Области вирішень системи лінійних нерівностей

Рис.4.7.1 а показує, що область вирішення системи нерівностей (4.7.12) є обмежене, опукле тіло (трикутник). Стрілками на малюнку вказані на пів площини, які є областями вирішень відповідних нерівностей системи (4.7.12), яка є визначеною. На рис.4.7.1 би показаний випадок, коли область вирішення системи нерівностей (4.7.12) є необмеженим, опуклим тілом. В даному випадку система нерівностей є невизначеною. На рис.4.7.1 в область вирішень системи нерівностей (4.7.12) порожня – не містить жодної крапки. Дана система нерівностей є несумісною.

Для пошуку оптимального рішення в завданні лінійного програмування розглянемо лінії рівня функції (4.7.9) на площині x_1, x_2 (см.рис.4.7.1а)

$$b_{11}x_1 + b_{12}x_2 = c. \quad (4.7.15)$$

Рівняння (4.7.15) описує сімейство прямих, паралельних прямою

$$b_{11}x_1 + b_{12}x_2 = 0. \quad (4.7.16)$$

Оптимальне значення (максимальне або мінімальне) цільової функції відповідатиме прямій сімейства (4.7.15), яка міститиме крайню крапку з області вирішення системи нерівностей (4.7.12). У разі пошуку максимуму пряма повинна бути найбільш видаленою від початку координат, а у разі пошуку мінімуму пряма повинна бути найменш видалена від початку координат.

Для вирішення завдань лінійного програмування з числом змінних x_1, x_2, \dots, x_n більше двох використовується сімплексний метод (метод послідовного поліпшення плану). Суть його полягає в тому, що знаючи деякий опорний план завдання, шляхом кінцевого числа кроків необхідно отримати оптимальний план виконуючи перехід від однієї крайньої точки опуклої множини до іншої з виконанням необхідної умови – зменшенням (збільшенням) значення цільової функції $f(x_1, x_2, \dots, x_n)$.

Одним з розділів математичного програмування є динамічне програмування – спеціальний обчислювальний метод оптимізації вирішення завдань, в основі якого лежить принцип оптимальності що стверджує, що який би не був шлях досягнення деякого стану системи, подальші рішення повинні належати оптимальній програмі для частини шляху, що залишилася, починається з цього стану. Для застосування принципу оптимальності в конкретних завданнях користуються прийомом, який називають зануренням. Його суть полягає в тому, що замість рішення початкової задачі з даним початковим станом і даним числом кроків, вирішується ціле сімейство завдань з довільним початковим станом і з довільним числом кроків [14].

Припустимо, що деяка система S може змінювати свій стан. Дана система S буде керованою якщо ми зможемо впливати на стан даної системи переводячи її з одного стану в інше за допомогою деяких дій, які

назвемо управлінням u . Критерій стану системи W є функцією управління

$$W = W(u). \quad (4.7.17)$$

Суть завдання оптимального управління полягає у відшуванні такого u^* яке дозволяє переводити систему S з початкового стану в кінцевий так, щоб критерій W приймав максимальне (мінімальне) значення

$$W^* = W(u^*) = \max_u [W(u)]. \quad (4.7.18)$$

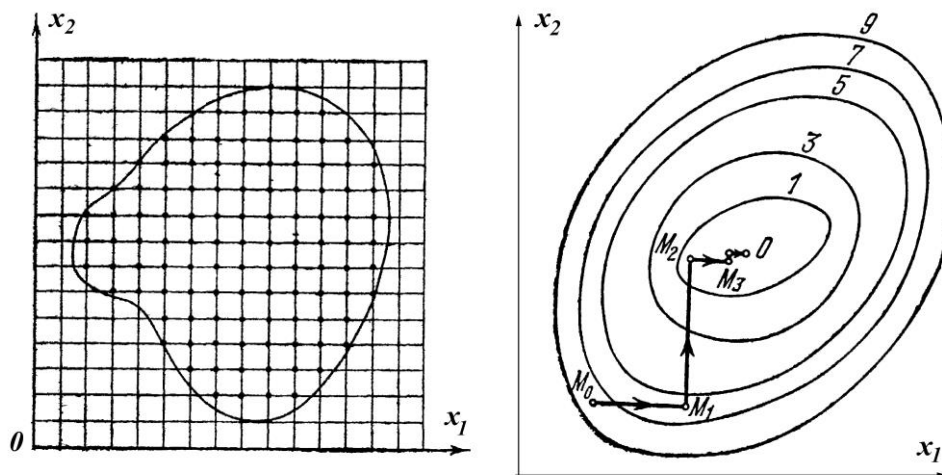
Стан системи S характеризується певною кількістю параметрів x_1, x_2, \dots, x_n (фазовими координатами системи). Процес переходу системи з початкового положення в кінцеве розіб'ємо на певне число кроків. Після цього, на кожному з кроків, починаючи з останнього, визначимо умовне оптимальне управління. Це робиться з урахуванням всіляких припущень про результати попереднього кроку. Дії продовжуються до тих пір, поки процес оптимізації буде доведений до початкового стану системи S . Потім знову проходиться вся послідовність кроків починаючи з першого. На кожному кроці з безлічі умовних оптимальних управлінь вибирається одне. Тут необхідно відзначити, що початковий стан системи S може бути точно задано, або його необхідно вибрати з допустимої області з умовою досягнення максимуму (мінімуму) критерію W у початковому стані.

Рішення задачі оптимізації, з геометричної точки зору, полягає у відшуванні максимальної (мінімальною) точки поверхні відгуку цільової функції $f(x_1, x_2, \dots, x_n)$. У разі, коли вид цільової функції відомий екстремальні крапки визначаються по описаних вище методиках. У іншому випадку може бути, що аналітичної залежності цільової функції немає, а у дослідника є можливість визначити її значення в будь-якій точці даної області (за допомогою експериментальних досліджень, в результаті розрахунку). Очевидно, що це дозволяє визначити значення цільової функції лише в кінцевому числі крапок. Надалі, при описі методів пошуку

екстремуму цільової функції $f(x_1, x_2, \dots, x_n)$ розглядатимемо функції двох параметрів $f(x_1, x_2)$.

На рис.4.7.2 а показана реалізація методу, коли дана область зміни параметрів x_1, x_2 покривається сіткою з вертикальним і горизонтальним кроком h . Після цього, визначається значення цільової функції $f(x_1, x_2)$ у вузлах сітки і вибирається вузол, в якому маємо максимальне значення. Проте, даний метод є вельми наближеним і вимагає значних витрат машинного часу для розрахунків.

На рис.4.7.2 б показана реалізація методу по координатного спуску. Його суть полягає в наступному. На першому етапі в початковій точці M_0 фіксуємо змінну x_2 і починаємо змінювати x_1 . Рухатимемося від початкової точки $x_1 = x_{10}$ у бік убуття (зростання) функції, поки не дійдемо до її мінімуму ($x_1 = x_{11}$ (максимуму)) після якого вона починає зростати (убувати). Дану крапку позначимо M_1 . Фіксуємо тепер змінну $x_1 = x_{11}$ і починаємо змінювати другу змінну $x_2 = x_{20}$ у бік убуття (зростання) цільової функції. Знову дійдемо до мінімуму (максимуму) цільової функції. Позначимо цю крапку через M_2 . Після цього знову повертаємося до змінній x_1 . Даний процес обриваємо на деякому кроці і приблизно приймаємо значення цільової функції за екстремальне.



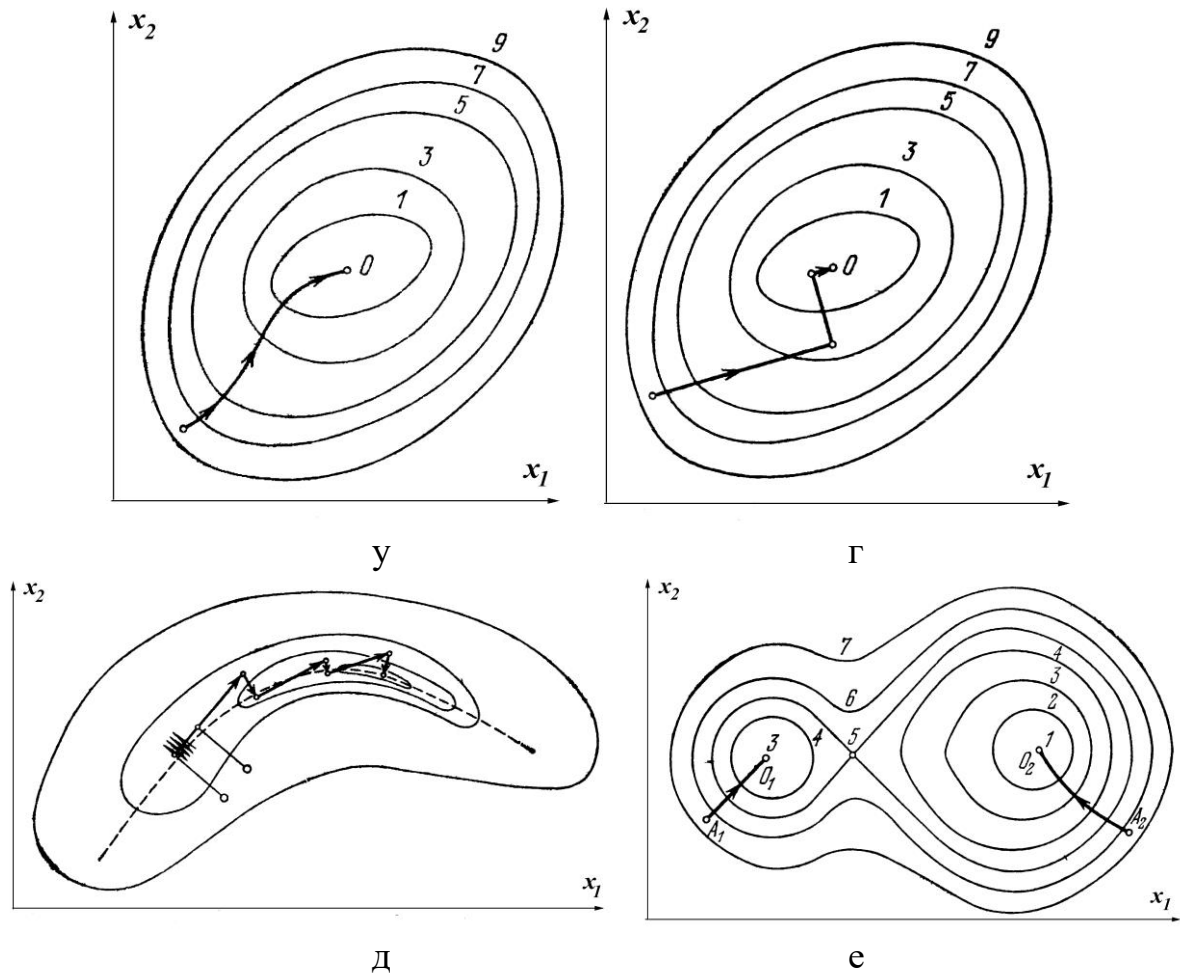


Рис.4.7.2. Методи визначення екстремуму цільової функції $f(x_1, x_2)$

На рис.4.7.2 в показана реалізація методу градієнтного спуску (підйому). Для цього методу необхідно знати вираз цільовій функції. Вектор градієнта показує напрям найбільш швидкого зростання функції. Його модуль рівний

$$|\mathit{grad} f(x_1, x_2)| = \sqrt{\left[\frac{\partial f(x_1, x_2)}{\partial x_1}\right]^2 + \left[\frac{\partial f(x_1, x_2)}{\partial x_2}\right]^2}. \quad (4.7.19)$$

Суть методу полягає в наступному. Вибираємо початкову точку і визначаємо в ній градієнт. Після цього робимо невеликий крок в цьому напрямі. У новій крапці повторюємо процедуру. Складність даного методу полягає у визначенні по (4.7.19) вирази для градієнта на кожному кроці.

Досконалішою є методика модифікованого градієнтного методу – метод найскорішого спуску. На рис.4.7.2 г показана його реалізація. Послідовність операцій буде наступна. У початковій точці визначають градієнт і роблять в його напрямі не маленький крок, а рухаються в його напрямі до тих пір, поки цільова функція зростає. У цій крапці знову обчислюють градієнт і рухаються в новому напрямі.

Вельми часто при пошуку екстремуму цільової функції дослідники стикаються з проблемою «ярів» (лінії рівня сильно витягнуті в одному напрямі і сплюснуті в іншому). На рис.4.7.2 д показані перетини поверхні відгуку з областю «яру». Для вирішення цієї проблеми з двох близьких крапок здійснюють градієнтний спуск на дно «яру». Потім сполучають знайдені точки прямої і роблять уздовж неї великий крок «яру». З отриманої крапки знову здійснюють градієнтний спуск на дно «яру» і роблять наступний крок «яру».

В деяких випадках цільові функції володіють властивістю багато екстремальності (см.рис.4.7.2 е). Для визначення абсолютного екстремуму необхідно проводити пошук з декількох різних крапок. Якщо набудемо різних значень цільової функції, то після їх порівняння вибираємо найбільше (найменше).

Приклад № 1. Завдання пошуку оптимального розміру циліндрової судини. Необхідно визначити радіус r підстави судини постійного об'єму V . Як цільові функції, на попередньому етапі, виберемо дві. Зажадаємо щоб площа поверхні циліндрової судини $S(r)$ була мінімальною (на його виготовлення піде менше матеріалу) і зажадаємо щоб довжина зварних швів $l(r)$ була мінімальною (економиться час на виконання операції зварювання при виготовленні судини). Вирази для цільових функцій мають вигляд

$$\begin{aligned} S(r) &= 2\pi r^2 + 2\pi r h, \\ l(r) &= 4\pi r + h, \\ V &= \pi r^2 h, \end{aligned} \tag{4.7.20}$$

де h - висота судини.

Перетворимо (4.7.20) до вигляду

$$\begin{aligned} S(r) &= 2\pi r^2 + \frac{2V}{r}, \quad 0 < r < \infty, \\ l(r) &= 4\pi r + \frac{V}{\pi r^2}, \quad 0 < r < \infty. \end{aligned} \tag{4.7.21}$$

Набутих значень цільових функцій показую, що вони залежать від однієї змінної r . Скориставшись (4.7.2) визначимо їх похідні і прирівняємо нулю

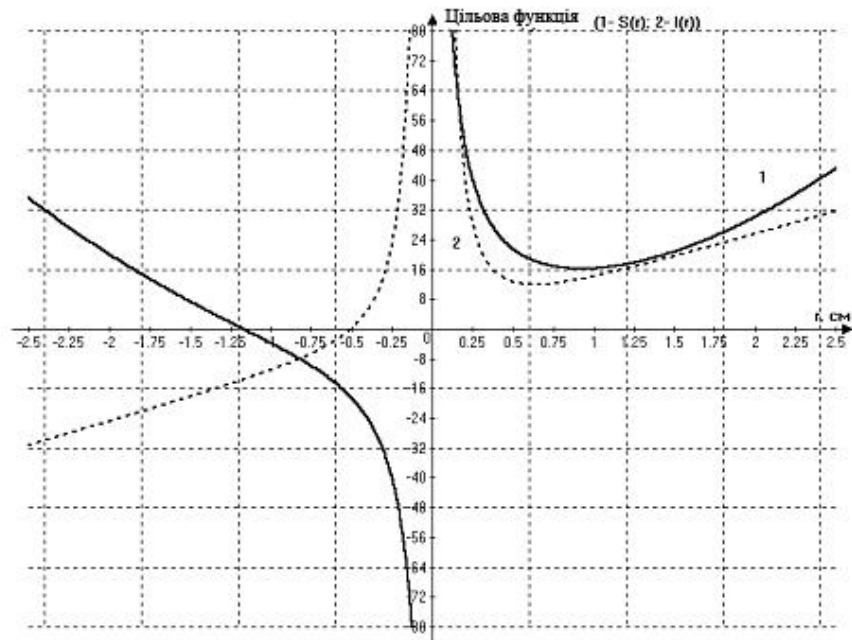
$$\begin{aligned} \frac{\partial S(r)}{\partial r} &= \frac{2}{r^2}(2\pi r^3 - V) = 0, \\ \frac{\partial l(r)}{\partial r} &= \frac{2}{\pi r^3}(2\pi^2 r^3 - V) = 0. \end{aligned}$$

З отриманої системи рівнянь визначаємо значення радіусу циліндрової судини при якому отримаємо мінімальну площу поверхні $S(r_S)$ і мінімальну довжину швів $l(r_l)$

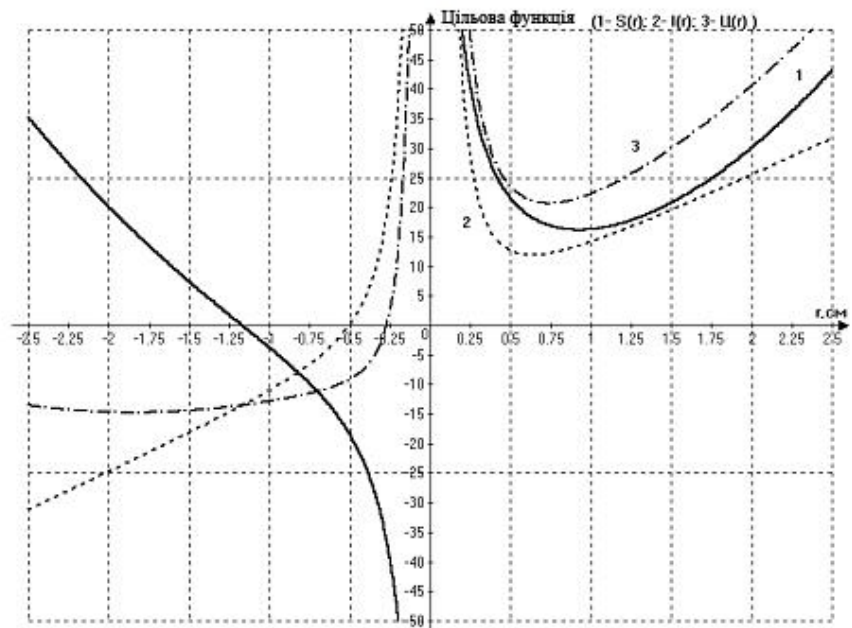
$$\begin{aligned} r_S &= \sqrt[3]{\frac{V}{2\pi}}, \quad h_S = 2r_S, \\ r_l &= \sqrt[3]{\frac{V}{2\pi^2}}, \quad h_l = 2\pi r_l. \end{aligned} \tag{4.7.22}$$

На рис.4.7.3 а показані графічні залежності зміни цільових функцій $S(r)$ і $l(r)$ (при розрахунках приймали $V=5$ см³).

Ускладнимо завдання. Хай нам необхідно визначити радіус r циліндрової судини постійного об'єму V . Нам вимагається щоб на виготовлення судини пішов якомога менше матеріалу, щоб довжина зварних швів була мінімальною. Крім того, відомо що ціна одиниці площі матеріалу судини C_1 а ціна одиниці довжини зварного шва C_2 . Тоді як цільова функція можна узяти вартість циліндрової судини C .



а



б

Рис.4.7.3. Графіки зміни цільових функцій $S(r)$, $I(r)$ и $U(r)$

З обліком (4.7.20), (4.7.21), отримаємо

$$U(r) = 2U_1\pi r^2 + \frac{2U_1V}{r} + 4U_2\pi r + \frac{VU_2}{\pi r^2}, \quad 0 < r < \infty. \quad (4.7.23)$$

Визначимо першу похідну і прирівняємо її нулю

$$\frac{\partial \Pi(r)}{\partial r} = 4\Pi_1\pi r - \frac{2\Pi_1 V}{r^2} + 4\Pi_2\pi r^3 - \frac{2V\Pi_2}{\pi r^3} = 0. \quad (4.7.24)$$

Після перетворень (4.7.24) прийме вигляд

$$4\Pi_1\pi^2 r^4 - 2\Pi_1 V\pi r + 4\Pi_2\pi^2 r^3 - 2V\Pi_2 = 0.$$

Вирішуємо отримане рівняння четвертого ступеня чисельним методом (см.4.7). Для $V=5\text{см}^3$, $\Pi_1=0,5$ ед. ціни $\Pi_2=1$ ед. ціни отримаємо чотири корені: $r_1=0,7364$ см $r_2= - 1,8472$ см $r_{3,4}= - 0,4446 \pm 0,418j$. Комплексне коріння відкидаємо і внаслідок того, що $0 < r < \infty$ отримуємо $r_{\Pi}=0,7364$ див.

На рис.4.7.3 би показані графічні залежності зміни цільових функцій $S(r)$ $l(r)$

$$r_l < r_{\Pi} < r_s.$$

Приклад № 2. Визначення мінімальної відстані r_0 від носика спрямовувача нитки (з координатами $x=a, y=b, z=c$) до відбійної площини плосков'язальної машини, вираженої рівнянням

$$Ax + By + Cz + D = 0, \quad (4.7.25)$$

де A, B, C, D - деякі постійні.

Квадрат відстані від носика спрямовувача нитки до змінної крапки (x, y, z) виражається залежністю

$$r^2 = (x-a)^2 + (y-b)^2 + (z-c)^2. \quad (4.7.26)$$

Вирішуємо дану задачу, з обмеженнями у вигляді рівності, з використанням множників Лагранжа. Необхідно мінімізувати вираз (4.7.26). Допоміжна функція має вигляд (з обліком (4.7.6))

$$\Phi = (x-a)^2 + (y-b)^2 + (z-c)^2 + \lambda_1(Ax + By + Cz + D).$$

Визначаємо її приватні похідні і прирівнюємо їх нулю. Тоді отримаємо

$$\begin{aligned} x &= a - \frac{1}{2}\lambda_1 A, \\ y &= b - \frac{1}{2}\lambda_1 B, \\ z &= c - \frac{1}{2}\lambda_1 C. \end{aligned} \quad (4.7.27)$$

Підставляючи значення x, y, z у (4.7.25) визначаємо значення множника Лагранжа

$$\lambda_1 = \frac{2(Aa + Bb + Cc + D)}{A^2 + B^2 + C^2}. \quad (4.7.28)$$

Підставляючи (4.7.27) в (4.7.26), з обліком (4.7.28), отримаємо вираз для визначення мінімальної відстані від носика спрямовувача нитки до відбійної площини плосков'язальної машини

$$r_0^2 = \frac{1}{4} \lambda_1^2 (A^2 + B^2 + C^2).$$

Приклад № 3. Проектування коробки у формі паралелепіпеда з шматка матеріалу постійної площі $2a$ найбільшого об'єму V . Цільова функція матиме вигляд

$$V = xyz, \quad (4.7.29)$$

де x, y, z - довжини сторін паралелепіпеда.

Рівняння, що зв'язує змінні x, y, z має вигляд

$$xy + xz + yz - a = 0, \quad (4.7.30)$$

$$0 < x < \infty, \quad 0 < y < \infty, \quad 0 < z < \infty.$$

Складаємо вираз для допоміжної функції з використанням множника Лагранжа

$$\Phi = xyz + \lambda(xy + xz + yz - a).$$

Визначаємо приватні похідні від останнього виразу і прирівнюємо їх нулю

$$\begin{aligned} yz + \lambda(y + z) &= 0, \\ xz + \lambda(x + z) &= 0, \\ xy + \lambda(x + y) &= 0. \end{aligned} \quad (4.7.31)$$

Вирішуючи спільно (4.7.30) і (4.7.31), отримаємо

$$\lambda = -\frac{3xyz}{2a}.$$

Підставляючи λ у (4.7.31), отримаємо

$$\frac{3x}{2a}(y + z) = 1, \quad \frac{3y}{2a}(x + z) = 1, \quad \frac{3z}{2a}(y + x) = 1.$$

Вирішуючи останню систему рівнянь визначимо, що $x = y = z = \sqrt{\frac{a}{3}}$. Таким чином, коробка матиме максимальний об'єм при постійній площі поверхні, коли вона формою представлятиме куб.

5. УДОСКОНАЛЕННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ТЕКСТИЛЬНОЇ ПРОМИСЛОВОСТІ З УРАХУВАННЯМ НЕЛІНІЙНИХ ХАРАКТЕРИСТИК ДЕФОРМАЦІЇ ТА ТЕРТЯ В ЗОНІ КОНТАКТУ НИТКИ З НАПРЯМНОЮ

Взаємодія ниток з циліндричними напрямними поверхнями зустрічається в багатьох технологічних процесах текстильної і трикотажної промисловості. При цьому можна розділяти взаємодію ниток з напрямними великої і малої кривини. В даній роботі розглядається останній випадок.

Взаємодія ниток з циліндричними напрямними з постійною кривизною поверхні широко зустрічається в трикотажній, ткацькій промисловості, коли відбувається відносне переміщення напрямних та ниток.

Удосконалення технологічних процесів легкої та текстильної промисловості повинно базуватися на комплексних дослідженнях процесу взаємодії ниток з напрямними поверхнями малої кривизни [1,4]. Отримання теоретичних залежностей дозволить отримати рівняння для визначення натягу нитки в зоні формування тканини та трикотажу, зменшити відсоток обривів ниток, поліпшити якість готової продукції [2]. В багатьох проведених дослідженнях не враховується нелінійна залежність сили та коефіцієнта тертя при отриманні остаточних рівнянь [4]. Все це призводить до значних розбіжностей між теоретичними та експериментальними залежностями, що не дозволяє використовувати перші при конкретних розрахунках.

Теоретичні дослідження процесу взаємодії ниток з циліндричними нитко напрямними пристроями з урахуванням нелінійної залежності сили та коефіцієнту тертя ставлять перед собою мету визначення їх натягу в довільній точці, що дозволить удосконалити процес їх переробки на технологічному устаткуванні [11].

5.1.ВЗАЄМОДІЯ НИТКИ З НАПРАВЛЯЮЧИМИ ПОВЕРХНЯМИ МАЛОЇ КРИВИЗНИ

Отримання диференціальних рівнянь, які описують рівновагу нитки на поверхні малої кривизни

На рис.5.1.1 показана загальна розрахункова схема. Нитка огинає циліндричну напрямну поверхню радіусу R . Вхідний натяг нитки дорівнює P_0 , а вихідний натяг нитки дорівнює P . Кут охоплення ниткою циліндричної напрямної дорівнює φ .

Система диференціальних рівнянь, яка описує рівновагу нескінченно малого елемента нитки $ds = R d\varphi$ (s - дугова координата) має вигляд [1,4]

$$\frac{dP}{ds} = F_{mp}, \quad \frac{P}{R} = N, \quad (5.1.1)$$

де P - натяг нитки;

F_{mp} - сила тертя, яка діє на нескінченно малий елемент нитки;

N - питома нормальна реакція напрямної поверхні.

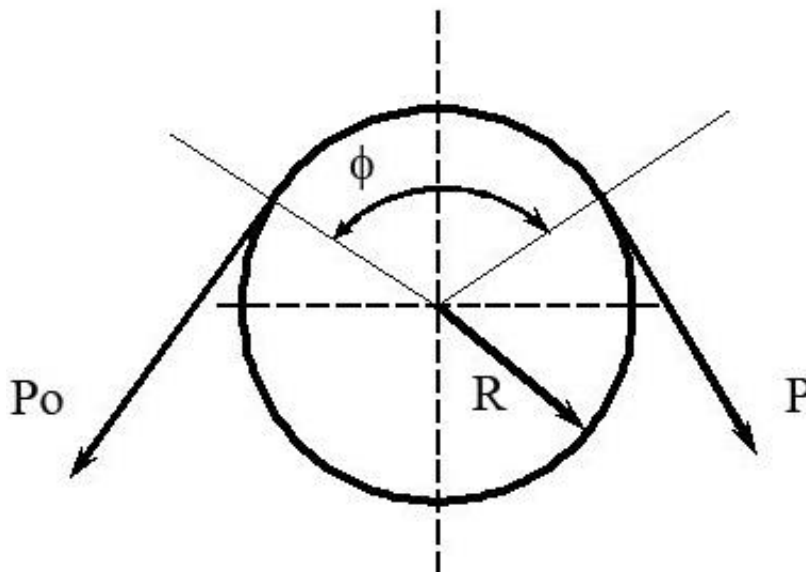


Рис.5.1.1. Загальна розрахункова схема

Дослідження, проведені в цьому напрямку [17, 19, 20], свідчать про те, що сила та коефіцієнт тертя нелінійно залежать від попереднього натягу, кута охоплення ниткою напрямної, діаметру циліндричної напрямної [1,4]. На рис.5.1.2 представлені графічні залежності зміни коефіцієнта тертя f від попереднього натягу нитки P_0 (крива 1), кута охоплення ниткою напрямної ϕ (крива 2), діаметру циліндричної напрямної d (крива 3).

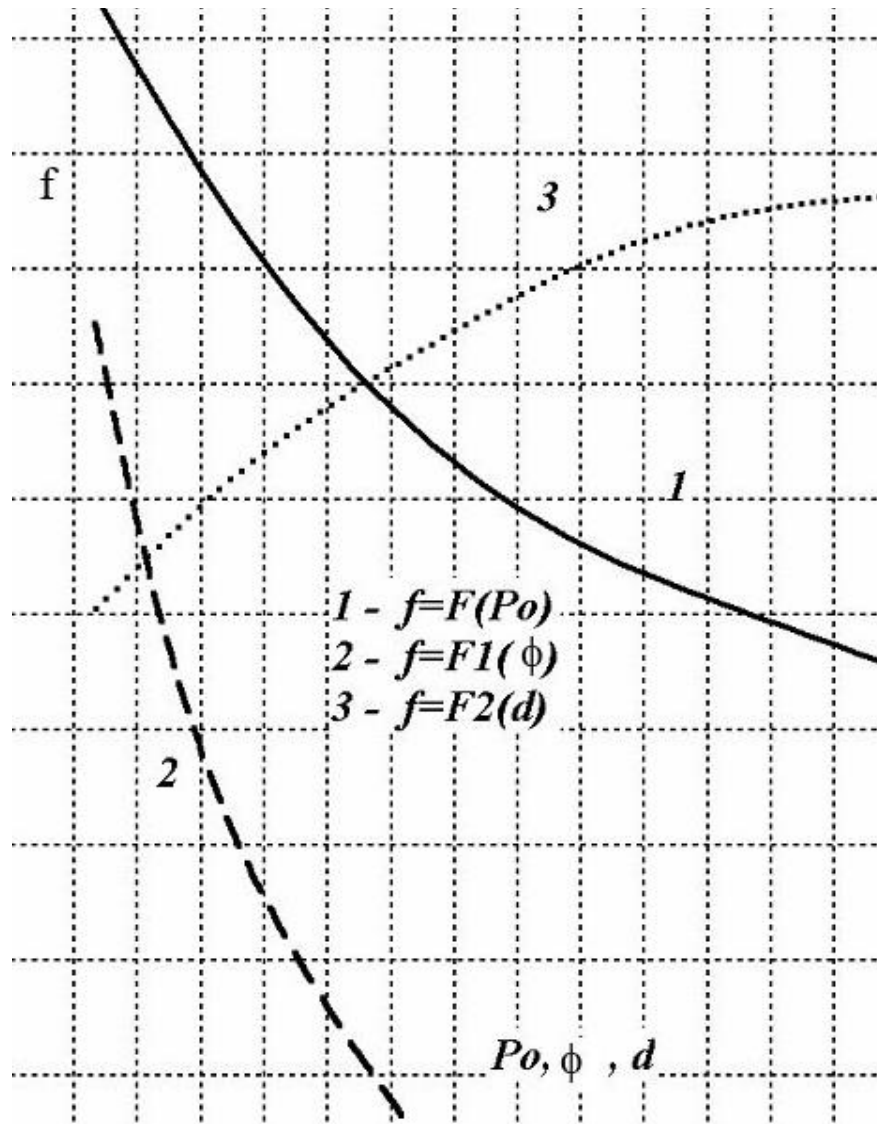


Рис.5.1.2. Графічні залежності коефіцієнта тертя від натягу нитки, кута охоплення та діаметру напрямної

Виходячи з цього силу тертя $F_{тр}$ та коефіцієнт тертя f можна представити наступними залежностями

$$\begin{aligned} F_{mp} &= fN^n, \\ f &= \frac{a}{b\varphi^{n_1}}, \end{aligned} \quad (5.1.2)$$

де n, a, b, n_1 - деякі константи, значення яких залежать від виду матеріалів нитки та напрямної та умов взаємодії між ними. Так значення n знаходиться в межах $\frac{2}{3} \leq n \leq 1$ [19]. Коли $n_1 = 0$, то з другого рівняння системи (5.1.2) коефіцієнт тертя буде дорівнювати $f = \frac{a}{b}$.

З урахуванням системи рівнянь (5.1.2), представимо систему диференціальних рівнянь (5.1.1) у вигляді

$$\frac{dP}{Rd\varphi} = \frac{aP^n}{b\varphi^{n_1}R^n}. \quad (5.1.3)$$

Розділяємо в диференційному рівнянні (5.1.3) змінні і отримаємо наступне диференціальне рівняння

$$\frac{dP}{P^n} = \frac{aRd\varphi}{b\varphi^{n_1}R^n}.$$

Проінтегруємо останнє диференціальне рівняння

$$\int_{P_0}^P P^{-n} dP = \frac{a}{bR^{n-1}} \int_0^\varphi \varphi^{-n_1} d\varphi, \quad (5.1.4)$$

або

$$\frac{1}{1-n} P^{1-n} \Big|_{P_0}^P = \frac{a\varphi^{1-n_1}}{bR^{n-1}(1-n_1)} \Big|_0^\varphi.$$

Підставляємо верхнє та нижнє значення параметрів визначених інтегралів та отримуємо

$$P^{1-n} - P_0^{1-n} = \frac{(1-n)a}{bR^{n-1}(1-n_1)} \varphi^{1-n_1}.$$

В остаточному вигляді отримаємо формулу для визначення натягу ведучої гілки нитки

$$P^{1-n} = P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1}. \quad (5.1.5)$$

Визначення натягу нитки з урахуванням нелінійної залежності сили та коефіцієнта тертя

Перейдемо до аналізу рівняння (5.1.5). Прологарифмуємо його ліву та праву частини

$$(1-n)\ln P = \ln \left[P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1} \right],$$

або

$$\ln P = \frac{\ln \left[P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1} \right]}{(1-n)}.$$

Визначимо значення натягу ведучої гілки нитки для чого перейдемо в останньому рівнянні до границі при $n \rightarrow 1$

$$\lim_{n \rightarrow 1} \ln P = \lim_{n \rightarrow 1} \frac{\ln \left[P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1} \right]}{(1-n)}. \quad (5.1.6)$$

Використовуючи правило Лопіталя будемо мати

$$\lim_{n \rightarrow 1} \ln P = \lim_{n \rightarrow 1} \frac{\ln \left[P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1} \right]}{(1-n)} = \lim_{n \rightarrow 1} \frac{F_1}{F_2},$$

$$F_1 = \frac{d}{dn} \left\{ \ln \left[P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1} \right] \right\}, \quad (5.1.7)$$

$$F_2 = \frac{d}{dn} (1-n) = -1.$$

Знайдемо значення функції F_1 для чого визначимо значення відповідної похідної

$$F_1 = \frac{-P_0^{1-n} \ln P_0 - \frac{a\varphi^{1-n_1} R^{1-n} \ln R}{b(1-n_1)} - \frac{a\varphi^{1-n_1} R^{1-n}}{b(1-n_1)} + \frac{a\varphi^{1-n_1} n R^{1-n} \ln R}{b(1-n_1)}}{P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1}}. \quad (5.1.8)$$

Підставляємо рівняння (5.1.8) та третє рівняння системи (5.1.7) в перше рівняння системи (5.1.7), отримаємо

$$\lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{P_0^{1-n} \ln P_0 + \frac{a\varphi^{1-n_1} R^{1-n} \ln R}{b(1-n_1)} + \frac{a\varphi^{1-n_1} R^{1-n}}{b(1-n_1)} - \frac{a\varphi^{1-n_1} n R^{1-n} \ln R}{b(1-n_1)}}{P_0^{1-n} + \frac{(1-n)aR^{1-n}}{b(1-n_1)} \varphi^{1-n_1}}. \quad (5.1.9)$$

Після підстановки (5.1.9) в перше рівняння системи (5.1.7), перетворимо його у вигляд

$$\ln P = \lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{P_0^{1-n} \left[\ln P_0 + \frac{a\varphi^{1-n_1} \ln R}{b(1-n_1)} \left(\frac{R}{P_0} \right)^{1-n} + \frac{a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0} \right)^{1-n} - \frac{a\varphi^{1-n_1} n \ln R}{b(1-n_1)} \left(\frac{R}{P_0} \right)^{1-n} \right]}{P_0^{1-n} \left[1 + \frac{(1-n)a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0} \right)^{1-n} \right]}.$$

Спростимо останнє рівняння, отримаємо

$$\ln P = \frac{\ln P_0 + \frac{a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n} [\ln R(1-n) + 1]}{1 + \frac{(1-n)a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n}}. \quad (5.1.10)$$

Аналізуючи рівняння (5.1.10), при $n \rightarrow 1$, отримаємо

$$\ln P - \ln P_0 = \frac{\frac{a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n}}{1}.$$

Остаточний вираз для визначення натягу ведучої гілки нитки буде мати вигляд

$$P = P_0 e^{\frac{a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n}}. \quad (5.1.11)$$

Для випадку, коли коефіцієнт тертя є сталою величиною, враховуючи рівняння (5.1.2), при $n_1 = 0$ $f = \frac{a}{b}$, отримаємо з (5.1.11)

$$P = P_0 e^{f\varphi \left(\frac{R}{P_0}\right)^{1-n}}. \quad (5.1.12)$$

Тут необхідно вказати, що формула (5.1.12) відрізняється від формули, яка отримана в роботі [4]. Спрощення формули в роботі [1] призводить до невірних остаточних результатів. Покладаючи в формулі (5.1.12) $n=1$, отримаємо відому формулу Л.Ейлера

$$P = P_0 e^{f\varphi}.$$

На рис.5.1.3 представлені результати визначення натягу для капронової нитки 28 Т для $P_0=5, 10, 15, 20$ сН по формулі (5.1.11). При розрахунках $a=0,4$; $b=2,7$; $n=0,7$; $n_1=0,8$; $\varphi=3,14$. На цьому рисунку наведені також експериментальні дані. Порівняння теоретичних та

експериментальних даних показало, що похибка не перевищує 2-4%. При розрахунку за формулою Л.Ейлера похибка перевищує 8%.

Отримання диференційних рівнянь, які описують рівновагу жорсткої на згин нитки на поверхні малої кривизни

В роботі наведені результати теоретичних та експериментальних досліджень по визначенню натягу жорсткої на згин нитки, яка взаємодіє з циліндричною напрямною поверхнею з урахуванням нелінійної залежності сили та коефіцієнту тертя. При проведенні досліджень використовувалися циліндричні напрямні поверхні, радіус кривизни яких був значно більшим радіуса нитки. Отримані рівняння для визначення натягу нитки використовувалися для удосконалення технологічних процесів текстильної промисловості.

Низька якість сировини, недосконалий ниткотракт технологічного устаткування при зростанні натягу по глибині заправки призводить до обривів. Прості технологічного устаткування, пов'язані з ліквідацією обривів, складають 75-80% від загального часу загальних простоїв [4,15].

Відсутність фундаментальних досліджень в галузі вивчення взаємодії ниток з напрямними великої та малої кривизни з урахуванням змінання, жорсткості на згин ускладнює вирішення такої важливої задачі як удосконалення технологічних процесів ткацтва та трикотажного виробництва.

Значно скоротити кількість обривів можна шляхом використання якісної сировини та оптимізацією умов переробки ниток на технологічному устаткуванні, де має місце їх взаємодія з направляючими та робочими органами великої та малої кривизни. На даний момент відсутні теоретичні та експериментальні дослідження процесу взаємодії ниток з напрямними поверхнями з урахуванням їх реальних фізико-

механічних властивостей [1,4]. По першу чергу це стосується нелінійної залежності сили та коефіцієнта тертя, впливу жорсткості нитки на згин, її змінання в зоні контакту з напрямною поверхнею [15-18].

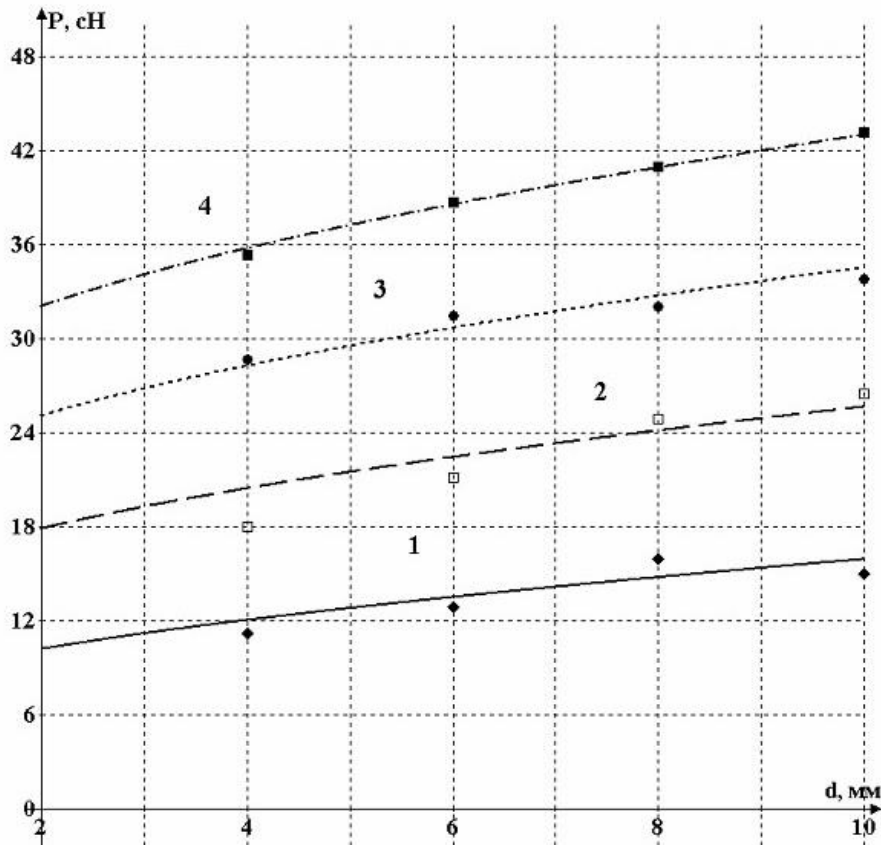


Рис.5.1.3. Залежність натягу капронової нитки 28 Т від радіуса кривизни циліндричної напрямної

Таким чином, метою даного дослідження є дослідження процесу взаємодії ниток з циліндричною напрямною з урахуванням реальних фізико-механічних властивостей та визначення натягу нитки після проходження нитконапрямних пристроїв.

На рис.5.1.4 представлена загальна розрахункова схема взаємодії жорсткої на вигин нитки з циліндричною напрямною поверхнею. Нитка огинає циліндричну напрямну радіуса R . Кут охоплення напрямної, при відсутності жорсткості на вигин дорівнював би $2\alpha = \pi = 180^\circ$ (вектори сил

натягу ведучої P та веденої P_0 гілок нитки мають бути спрямовані по дотичних в точках C і D). Завдяки наявності жорсткості нитки на вигин реальний кут охоплення буде становити $\varphi_p = 2\alpha - \gamma - \gamma_0$, де γ, γ_0 - кути (кути жорсткості) нахилу дотичних в точках O та O_1 до вертикальної прямої (або вісі y). Таким чином, необхідно визначити значення кутів γ, γ_0 . Рівняння рівноваги пружної лінії запишемо у вигляді (див.рис.5.1.4)

$$\frac{y'' \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^j}{(1 + y'^2)^{\frac{3}{2}}} = M, \quad M = P(x_A - x), \quad (5.1.13)$$

де $\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{j(K)}$ - коефіцієнт жорсткості нитки на вигин;

E - модуль пружності нитки на розтягнення;

d_i - діаметри окремих елементарних волокон (філаментів), з яких складається комплексна нитка чи пряжа;

w - кількість філаментів, з яких складається пряжа;

$j(K)$ - коефіцієнт, який залежить від крутки K нитки;

y', y'' - перша та друга похідна координати y по x ;

M - згинаючий момент;

P - натяг нитки;

x_A - координата точки прикладання сили натягу P .

Проведені дослідження [1,4] показали, що коефіцієнт жорсткості на вигин залежить від крутки K .

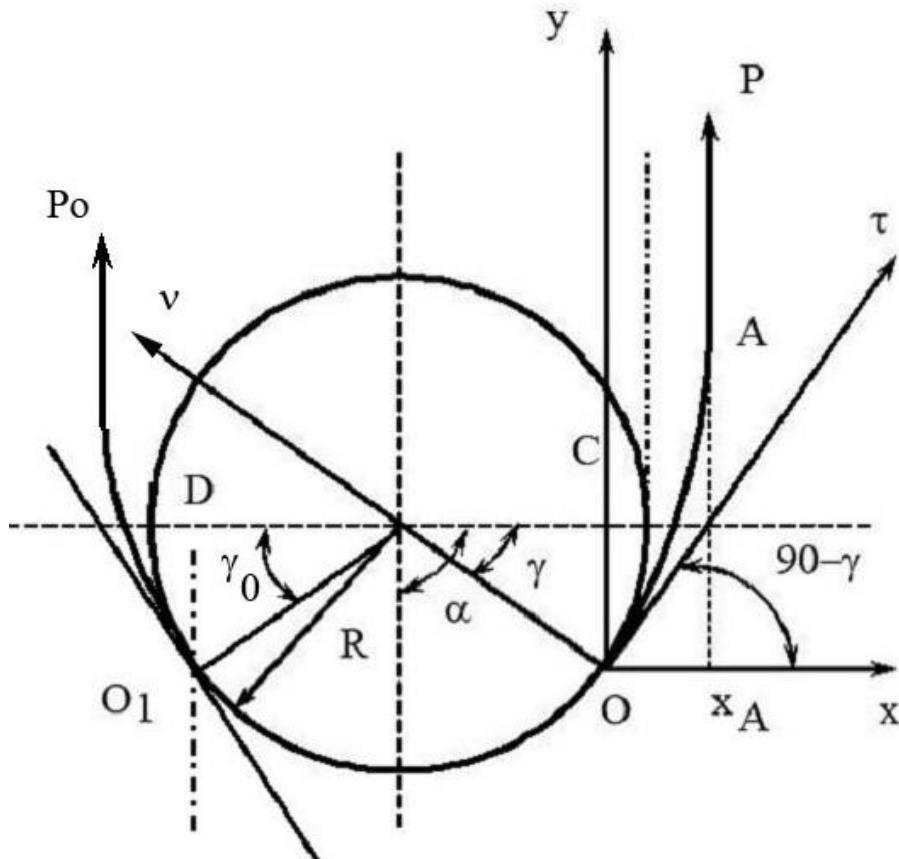


Рис.5.1.4. Загальна розрахункова схема

Вплив цього параметра оцінюється коефіцієнтом

$$j(K) = 1 + u_1 K^{u_2}, \quad (5.1.14)$$

де u_1, u_2 - деякі постійні коефіцієнти.

При $K \rightarrow 0$ $j(K) \rightarrow 1$ при $K \rightarrow K_{кр}$, $j(K) \rightarrow j(K_{кр})$, де $K_{кр}$ - критичне значення крутки. В останньому випадку коефіцієнт жорсткості нитки на вигин буде визначатися як у мононитки

$$B = EI = \frac{E\pi d^4}{64},$$

де $d = 2r$ - розрахунковий діаметр нитки.

З урахуванням (5.1.14) диференційне рівняння (5.1.13) запишемо у вигляді

$$\frac{y'' \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{(1+y'^2)^{\frac{3}{2}}} = P(x_A - x). \quad (5.1.15)$$

Розділяємо в диференційному рівнянні (5.1.15) змінні і отримаємо

$$\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} d(y')}{(1+y'^2)^{\frac{3}{2}}} = P(x_A - x) dx.$$

Інтегруємо останнє диференційне рівняння

$$\int \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} d(y')}{(1+y'^2)^{\frac{3}{2}}} = \int P(x_A - x) dx.$$

Отримаємо

$$\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} y'}{(1+y'^2)^{\frac{1}{2}}} = P(x_A x - \frac{x^2}{2}) + C,$$

де C - будь яка стала інтегрування.

Враховуючи що $y' = tg \varphi$, де φ - кут нахилу дотичної до вісі x , останнє диференційне рівняння можна представити у вигляді

$$\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} \sin \varphi = P(x_A x - \frac{x^2}{2}) + C. \quad (5.1.16)$$

При $x = 0, \varphi = 90 - \gamma$. Тоді стала інтегрування визначається з виразу

$$C = \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} \sin(90 - \gamma), \quad C = \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} \cos \gamma.$$

Тоді рівняння (5.1.16) прийме вигляд

$$\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} (\sin \varphi - \cos \gamma) = P(x_A x - \frac{x^2}{2}). \quad (5.1.17)$$

При $x = x_A$, $\varphi = \frac{\pi}{2}$. З урахуванням цього рівняння (5.1.17) представимо у вигляді

$$\cos \gamma = 1 - \frac{P x_A^2}{2 \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}. \quad (5.1.18)$$

Значення координати x_A знайдемо з умови рівноваги моментів сил відносно точки O

$$\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} / (R+r) = P x_A, \Rightarrow x_A = \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{P(R+r)}.$$

$$\cos \gamma = 1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P(R+r)^2}. \quad (5.1.19)$$

Аналогічний вивід рівняння для визначення кута γ_0 дозволяє отримати наступний результат

$$\cos \gamma_0 = 1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2}. \quad (5.1.20)$$

Використовуючи залежності (5.1.19) та (5.1.20) визначимо натяг нитки в точках O, O_1

$$P_0 = P \cos \gamma = P \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P(R+r)^2} \right], P_{0_1} = P_0 \cos \gamma_0 = P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2} \right]. \quad (5.1.21)$$

Перейдемо до розгляду питання, яке спрямоване на визначення закону зміни натягу нитки на ділянці OO_1 .

Нитка огинає циліндричну напрямну поверхню радіусу R . Вхідний натяг нитки дорівнює P_0 , а вихідний натяг нитки дорівнює P . Кут охоплення ниткою циліндричної напрямної дорівнює $\varphi_p = 2\alpha - \gamma - \gamma_0$.

Система диференціальних рівнянь, що описує рівновагу нескінченно малого елемента нитки $ds = R d\varphi$ (s - дугова координата) має вигляд [1,4]

$$\begin{aligned} \frac{dP}{ds} - \frac{Q}{(R+r)} - F_{mp} &= 0, & \frac{dQ}{ds} + \frac{P}{(R+r)} - N &= 0, \\ \frac{dM}{ds} + Q &= 0, & M &= \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{(R+r)} = const, \end{aligned} \quad (5.1.22)$$

де P - натяг нитки; Q - перерізуюча сила; F_{mp} - сила тертя, яка діє на нескінченно малий елемент нитки; N - питома нормальна реакція напрямної поверхні.

Дослідження, проведені в цьому напрямку [15-18], свідчать про те, що сила та коефіцієнт тертя нелінійно залежать від попереднього натягу, кута охоплення ниткою напрямної, діаметра циліндричної напрямної [15]. Виходячи з цього силу тертя F_{mp} та коефіцієнт тертя f можна представити наступними залежностями

$$F_{mp} = fN^n, f = \frac{a}{b\varphi^{n_1}}, \quad (5.1.23)$$

де n, a, b, n_1 - деякі константи, значення яких залежить від виду матеріалів нитки та напрямної та умов взаємодії між ними. Так значення n знаходиться в межах $\frac{2}{3} \leq n \leq 1$ [120]. Коли $n_1 = 0$, то з другого рівняння

системи (5.1.23) коефіцієнт тертя буде дорівнювати $f = \frac{a}{b}$.

Враховуючи третє та четверте рівняння системи (5.1.22), а також (5.1.23), представимо систему диференціальних рівнянь (5.1.22) у вигляді

$$\frac{dP}{(R+r)d\varphi} = \frac{aP^n}{b\varphi^{n_1}(R+r)^n}. \quad (5.1.24)$$

Розділяємо в диференційному рівнянні (5.1.24) змінні і отримаємо наступне диференційне рівняння

$$\frac{dP}{P^n} = \frac{a(R+r)d\varphi}{b\varphi^{n_1}(R+r)^n}.$$

Проінтегруємо останнє диференційне рівняння

$$\int_{P_{0_1}}^{P_0} P^{-n} dP = \frac{a}{b(R+r)^{n-1}} \int_0^{\varphi_p} \varphi^{-n_1} d\varphi, \quad (5.1.25)$$

або

$$\frac{1}{1-n} P^{1-n} \Big|_{P_{0_1}}^{P_0} = \frac{a\varphi^{1-n_1}}{b(R+r)^{n-1}(1-n_1)} \Big|_0^{\varphi_p}.$$

Підставляємо верхнє та нижнє значення параметрів визначених інтегралів та отримуємо

$$P_0^{1-n} - P_{0_1}^{1-n} = \frac{(1-n)a}{b(R+r)^{n-1}(1-n_1)} \varphi_p^{1-n_1}.$$

В остаточному вигляді отримаємо формулу для визначення натягу ведучої гілки нитки

$$P_0^{1-n} = P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1}. \quad (5.1.26)$$

Визначення натягу нитки жорсткої на згин з урахуванням нелінійної залежності сили та коефіцієнта тертя

Перейдемо до аналізу рівняння (5.1.26). Прологарифмуємо його ліву та праву частини

$$(1-n)\ln P_0 = \ln \left[P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1} \right],$$

або

$$\ln P_0 = \frac{\ln \left[P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1} \right]}{(1-n)}.$$

Визначимо значення натягу ведучої гілки нитки для чого перейдемо в останньому рівнянні до границі при $n \rightarrow 1$

$$\lim_{n \rightarrow 1} \ln P_0 = \lim_{n \rightarrow 1} \frac{\ln \left[P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1} \right]}{(1-n)}. \quad (5.1.27)$$

Використовуючи правило Лопіталя будемо мати

$$\lim_{n \rightarrow 1} \ln P_0 = \lim_{n \rightarrow 1} \frac{\ln \left[P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1} \right]}{(1-n)} = \lim_{n \rightarrow 1} \frac{F_1}{F_2},$$

$$F_1 = \frac{d}{dn} \left\{ \ln \left[P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1} \right] \right\}, \quad (5.1.28)$$

$$F_2 = \frac{d}{dn} (1-n) = -1.$$

Знайдемо значення функції F_1 для чого визначимо значення відповідної похідної

$$F_1 = \frac{-P_{0_1}^{1-n} \ln P_{0_1} - \frac{a\varphi_p^{1-n_1}(R+r)^{1-n} \ln(R+r)}{b(1-n_1)} - \frac{a\varphi_p^{1-n_1}(R+r)^{1-n}}{b(1-n_1)} + \frac{a\varphi_p^{1-n_1}n(R+r)^{1-n} \ln(R+r)}{b(1-n_1)}}{P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1}}. \quad (5.1.29)$$

Підставляємо рівняння (5.1.29) та третє рівняння системи (5.1.28) в перше рівняння системи (5.1.28), отримаємо

$$\lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{P_{0_1}^{1-n} \ln P_{0_1} + \frac{a\varphi_p^{1-n_1}(R+r)^{1-n} \ln(R+r)}{b(1-n_1)} + \frac{a\varphi_p^{1-n_1}(R+r)^{1-n}}{b(1-n_1)} - \frac{a\varphi_p^{1-n_1}n(R+r)^{1-n} \ln(R+r)}{b(1-n_1)}}{P_{0_1}^{1-n} + \frac{(1-n)a(R+r)^{1-n}}{b(1-n_1)} \varphi_p^{1-n_1}}. \quad (5.1.30)$$

Після підстановки (5.1.30) в перше рівняння системи (5.1.28), перетворимо його у вигляд

$$\ln P_0 = \lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{P_{0_1}^{1-n} \left[\ln P_{0_1} + \frac{a\varphi_p^{1-n_1} \ln(R+r)}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n} + \frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n} - \frac{a\varphi_p^{1-n_1} n \ln(R+r)}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n} \right]}{P_{0_1}^{1-n} \left[1 + \frac{(1-n)a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n} \right]}$$

Спростимо останнє рівняння, отримаємо

$$\ln P_0 = \frac{\ln P_{0_1} + \frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n} [\ln(R+r)(1-n) + 1]}{1 + \frac{(1-n)a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n}}. \quad (5.1.31)$$

Аналізуючи рівняння (5.1.31), при $n \rightarrow 1$, отримаємо

$$\ln P_0 - \ln P_{0_1} = \frac{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n}}{1}.$$

Остаточний вираз для визначення натягу ведучої гілки нитки в точці O буде мати вигляд

$$P_0 = P_{0_1} e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n}}. \quad (5.1.32)$$

Для випадку, коли коефіцієнт тертя є сталою величиною, враховуючи рівняння (5.1.23), при $n_1 = 0$ $f = \frac{a}{b}$, отримаємо з (5.1.32)

$$P_0 = P_{0_1} e^{f\varphi_p \left(\frac{(R+r)}{P_{0_1}} \right)^{1-n}}. \quad (5.1.33)$$

Тут необхідно вказати, що формула (5.1.33) відрізняється від формули, що отримана в роботі [4]. Спрощення формули в роботі [4] призводить до невірних остаточних результатів. Покладаючи в формулі (5.1.33) $n = 1$, отримаємо відому формулу Л.Ейлера.

$$P_0 = P_{0_1} e^{f\varphi}.$$

Підставляючи результати системи (5.1.21) в (5.1.32) отримаємо наступний вираз

$$P \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P(R+r)^2} \right] = P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2} \right] e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2} \right]} \right)^{1-n}} .$$

В остаточному вигляді будемо мати

$$P = P_0 e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2} \right]} \right)^{1-n}} + \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2(R+r)^2} (1 - e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left(\frac{(R+r)}{P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0(R+r)^2} \right]} \right)^{1-n}}) .$$

Коли в останньому рівнянні знехтувати поперечними розмірами перетину нитки ($r=0$), а також вважати, що нитка має крутку $K \rightarrow K_{kp}$, $j(K) \rightarrow j(K_{kp})$ та

$$\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})} = EI, \quad I = \frac{\pi d^4}{64},$$

для жорсткої на вигин нитки отримаємо наступне рівняння

$$P = P_0 e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left[\frac{R}{P_0 \left(1 - \frac{EI}{2R^2} \right)} \right]^{1-n}} + \frac{EI}{2R^2} (1 - e^{\frac{a\varphi_p^{1-n_1}}{b(1-n_1)} \left[\frac{R}{P_0 \left(1 - \frac{EI}{2R^2} \right)} \right]^{1-n}}) . \quad (5.1.34)$$

Для випадку, коли можна знехтувати нелінійною залежністю сили та коефіцієнта тертя ($n=1, n_1=0$) отримаємо рівняння для визначення натягу жорсткої на вигин нитки

$$P = P_0 e^{f\varphi_p} + \frac{EI}{2R^2} (1 - e^{f\varphi_p}) . \quad (5.1.35)$$

Цей результат збігається з результатами наведеними в роботах [2,12]. У випадку, коли можна вважати нитку абсолютно гнучкою ($EI = 0$) отримаємо класичну формулу Л.Ейлера

$$P = P_0 e^{f\varphi}.$$

На рис.5.1.5 представлені результати розрахунку натягу ведучої гілки: 1 – капронова мононитка 52 Т(1 крива), $B=53,7\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,12\text{ мм}$, $R=0,5\dots5\text{ мм}$, $\varphi_p=3,14\text{ рад}$, $n=0,85$, $n_1=0,15$; 2 – капронова мононитка 36 Т(2 крива), $B=21,9\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,1\text{ мм}$, $R=0,5\dots5\text{ мм}$, $\varphi_p=3,14\text{ рад}$, $n=0,85$, $n_1=0,15$; 3 – капронова комплексна нитка 28 Т(3 крива), $B=0,22\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,1\text{ мм}$, $R=0,5\dots5\text{ мм}$, $\varphi_p=3,14\text{ рад}$, $n=0,85$, $n_1=0,15$. Аналіз отриманих результатів показав, що для 1 мононитки реальний кут охоплення напрямної найменший з обраних варіантів. При натязі веденої гілки нитки в 10 сН, мононитка 1 в недостатній мірі охоплює напрямну. Для 3 нитки (капронової комплексної), коефіцієнт жорсткості на вигин на порядок менше ніж у монониток, тому реальний кут охоплення буде найбільшим, а відповідно і натяг.

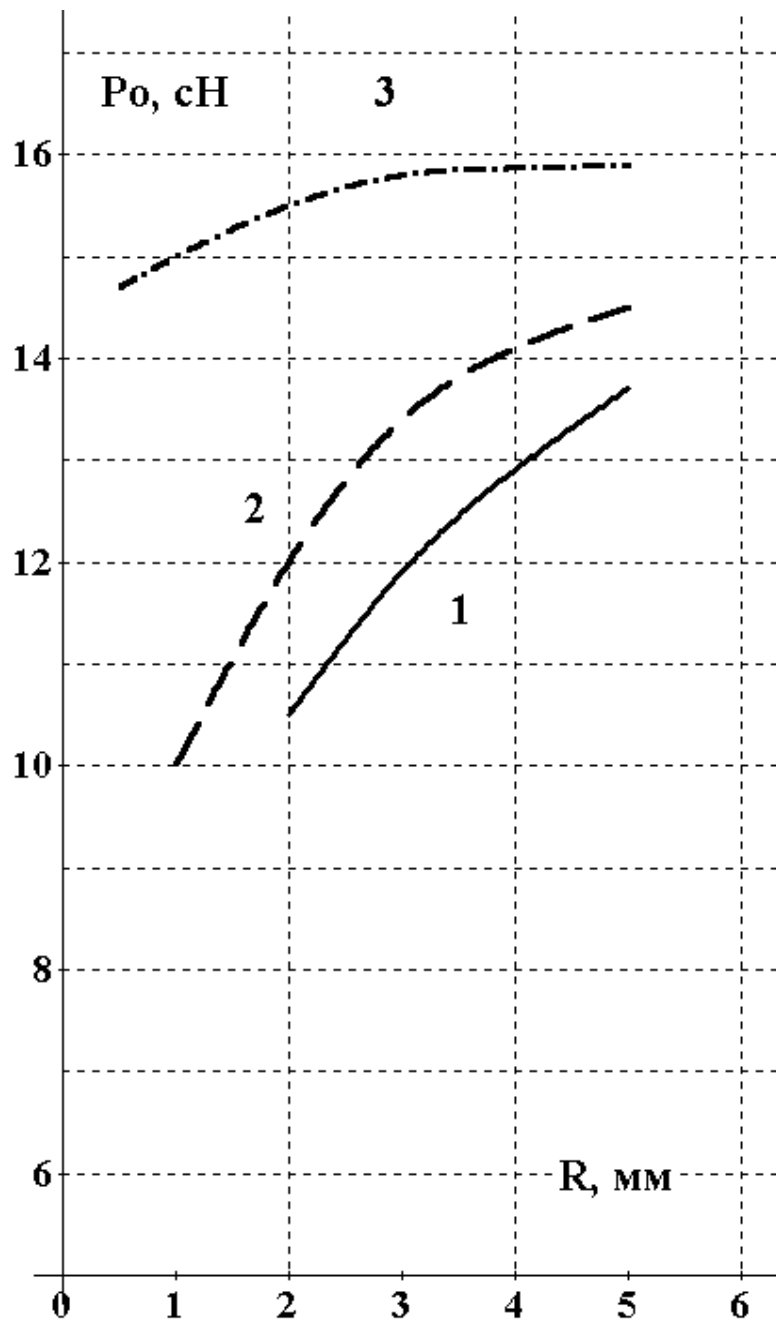


Рис.5.1.5. Розрахунок натягу ведучої гілки нитки

5.2. ДОСЛІДЖЕННЯ ВЗАЄМОДІЇ КОМПЛЕКСНИХ НИТОК І ПРЯЖІ З ЦИЛІНДРИЧНОЮ НАПРЯМНОЮ ПОВЕРХНЕЮ

Взаємодія ниток з циліндричними напрямними поверхнями зустрічається в багатьох технологічних процесах текстильної та трикотажної промисловості. У більшості випадків

має місце взаємодія нитки з циліндричною направляючою великої кривизни. Такий вид взаємодії вимагає, при виведенні основних рівнянь, враховувати мнучкість і жорсткість нитки на вигин. Врахування реальних фізико-механічних показників властивостей нитки дозволить отримати рівняння для визначення натягу і форми осі, які можна використовувати при удосконаленні технологічних процесів.

Удосконалення технологічних процесів легкої і текстильної промисловості має базуватися на комплексних дослідженнях процесу взаємодії ниток з напрямними поверхнями великої кривизни з урахуванням їх реальних фізико-механічних характеристик [1,4]. Отримання теоретичних залежностей дозволить скласти рівняння для визначення натягу нитки в зоні формування тканини і трикотажу, зменшити відсоток обривів ниток, поліпшити якість готової продукції [15-18]. У багатьох дослідженнях не враховуються реальні фізико-механічні властивості і нелінійна залежність сили і коефіцієнта тертя при отриманні остаточних рівнянь [12-14]. Все це призводить до значних розбіжностей між теоретичними та експериментальними залежностями, що не дозволяє використовувати перші при конкретних розрахунках.

Основним завданням є те, щоб шляхом інтегрування системи диференціальних рівнянь одержати залежності натягу нитки, з урахуванням мнучкості в зоні контакту, жорсткості на вигин і нелінійної залежності сили і коефіцієнта тертя, від радіуса

кривизни циліндричної напрямної, початкового натягу і кута охоплення ниткою направляючої поверхні.

Теоретичні дослідження процесу взаємодії ниток з циліндричними нитконапрямними пристроями з урахуванням мнучкості, жорсткості на вигин і нелінійної залежності сили і коефіцієнта тертя ставлять перед собою завдання визначення натягу ниток в довільній точці, рішення якого дозволить удосконалити процес їх переробки на технологічному обладнанні.

Визначення впливу мнучкості і жорсткості на вигин на натяг нитки при її взаємодії з циліндричною направляючою

На рис. 5.2.1 показана загальна розрахункова схема. Нитка огинає циліндричну направляючу поверхню радіуса R . Вхідний натяг нитки дорівнює P_0 , а вихідний натяг нитки дорівнює P . Кут охоплення ниткою циліндричної направляючої дорівнює $\varphi = \varphi_P + \varphi_{см1} + \varphi_{см2} - \varphi_{иж1} - \varphi_{иж2}$, де φ_P - кут охоплення ниткою циліндричної направляючої без урахування мнучкості і жорсткості на вигин (на рис.5.2.1 $\varphi_P = \pi$); $\varphi_{см1}, \varphi_{см2}$ - кути, на які збільшується кут охоплення φ_P за рахунок деформації зминання в зоні контакту нитки з направляючою; $\varphi_{иж1}, \varphi_{иж2}$ - кути, на які зменшується кут охоплення φ_P за рахунок наявності жорсткості нитки на згин[1,4].

Пошук взаємозв'язку між натягом ведучої P и веденої гілки нитки необхідно починати з встановлення залежності між натягом в точці P_A і натягом в точці P_B . На цих ділянках вільні гілки нитки підпорядковуються законам пружних на вигин одномірних об'єктів. Зв'язок між натягом в цих точках має вид [15-18]

$$P_A = P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0 [R + r(1 - \delta_0)]^2} \right],$$

$$P_B = P \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P [R + r(1 - \delta)]^2} \right],$$
(5.2.1)

де $\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{j(K)}$ - коефіцієнт жорсткості нитки на згин;

E - модуль пружності нитки на розтягнення;

d_i - діаметр окремого елементарного волокна (філамент), з яких складається комплексна нитка або пряжа;

w - кількість філаментів, з яких складається пряжа;

$j(K) = 1 + u_1 K^{u_2}$, - коефіцієнт, який залежить від крутки нитки;

δ, δ_0 - відносна деформація поперечного перерізу нитки в точках її сходу и входу на напрямну;

u_1, u_2 - деякі сталі коефіцієнти.

При $K \rightarrow 0$ $j(K) \rightarrow 1$ при, $K \rightarrow K_{KP}$, $j(K) \rightarrow J(K_{KP})$ де K_{KP} - критичне значення крутки. В останньому випадку коефіцієнт жорсткості нитки на вигин буде визначатися як у мононитки

$$B_0 = EI = \frac{E\pi d^4}{64},$$

де $d = 2r$ - розрахунковий діаметр нитки.

Перейдемо до визначення взаємозв'язку між натягом нитки в точках А і В. На цій ділянці нитка буде взаємодіяти з циліндричною напрямною поверхнею.

Система диференціальних рівнянь, яка описує рівновагу безкінечно малого елемента нитки $ds = (R+r)d\varphi$ (s - дугова координата) має вигляд [15-18]

$$\frac{dP}{ds} = F_{mp}, \quad \frac{P}{[R+r(1-\delta)]} = N, \quad N = bE_1\delta, \quad (5.2.2)$$

где P - натяг нитки;

F_{mp} - сила тертя, яка діє на безкінечно малий елемент нитки;

N - питома нормальна реакція напрямної поверхні;

s - дугова координата;

b - ширина сліду контакту нитки з напрямною поверхнею;

E_1 - модуль пружності нитки при зтисканні [4] .

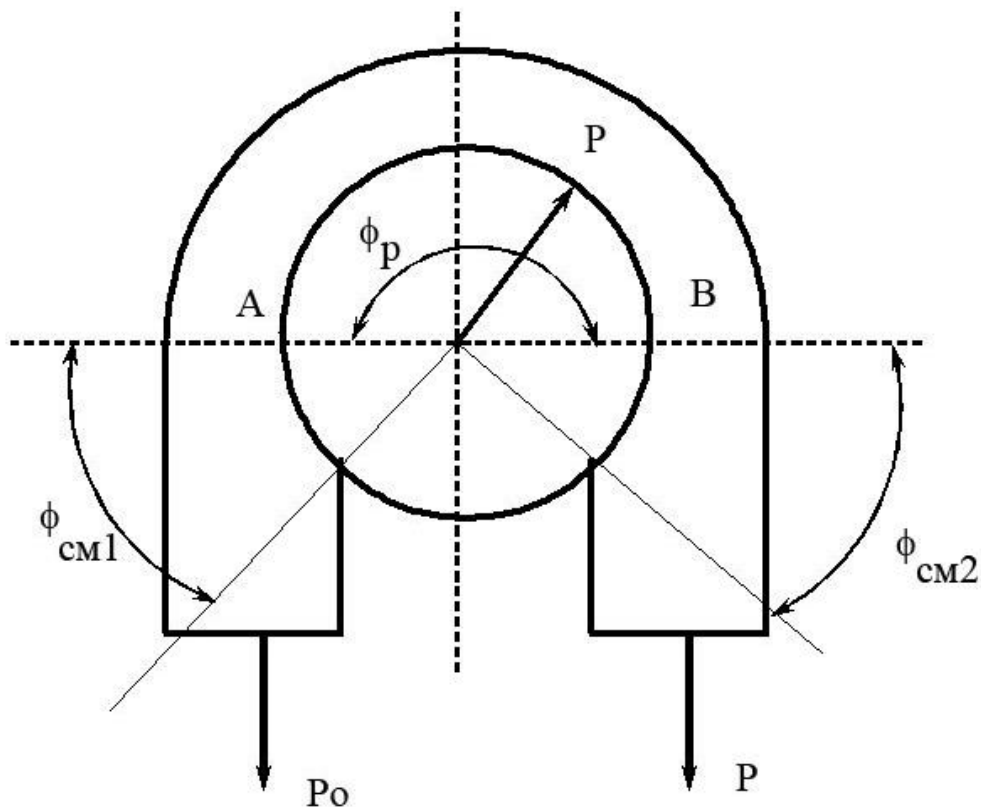


Рис.5.2.1. Загальна розрахункова схема

Дослідження, проведені в цьому напрямку [1,4], свідчать про те, що сила і коефіцієнт тертя нелінійно залежать від попереднього натягу, кута охоплення ниткою направляючої, діаметра циліндричної направляючої [2-

3] Виходячи з цього силу тертя і коефіцієнт тертя f можна представити наступними залежностями

$$\begin{aligned} F_{mp} &= fN^n, \\ f &= \frac{a}{b\varphi^{n_1}}, \end{aligned} \quad (5.2.3)$$

де $-a, b, n, n_1$ деякі константи, значення яких залежить від виду матеріалів нитки і напрямної та умов взаємодії між ними. Так значення n знаходиться в межах $\frac{2}{3} \leq n \leq 1$ [4]. При $n_1 \rightarrow 0$, з другого рівняння системи (5.2.3), визначимо коефіцієнт тертя, який буде дорівнювати $f = \frac{a}{b}$.

З системи (5.2.2) виключимо значення питомої нормальної реакції

$$\frac{P}{[R + r(1 - \delta)]} = bE_1\delta. \quad (5.2.4)$$

Підставляємо (5.2.4) в перше рівняння системи (5.2.2), з урахуванням (5.2.3), отримаємо

$$\frac{d}{ds} \{ [R + r(1 - \delta)] bE_1\delta \} = \frac{a}{b\varphi^{n_1}} (bE_1)^n (\delta)^n.$$

Розкриваємо ліву частину останнього рівняння

$$bE_1 \frac{d\delta}{ds} R + bE_1 \frac{d\delta}{ds} r - 2bE_1\delta \left(\frac{d\delta}{ds} \right) = \frac{a}{b\varphi^{n_1}} (bE_1)^n (\delta)^n.$$

Нехтуючи в останньому виразі добутками і квадратами малих величин, будемо мати

$$(R + r) \frac{d\delta}{ds} = \frac{a}{b\varphi^{n_1}} (bE_1)^{n-1} (\delta)^n.$$

Розділяємо в диференційному рівнянні змінні і отримаємо наступне диференційне рівняння

$$\frac{d\delta}{(\delta)^n} = \frac{a(bE_1)^{n-1} d\varphi}{b\varphi^{n_1}}.$$

Проінтегруємо останнє диференційне рівняння

$$\int_{\delta_0}^{\delta} (\delta)^{-n} d\delta = \frac{a(bE_1)^{n-1}}{b} \int_0^{\varphi} \varphi^{-n_1} d\varphi, \quad (5.2.5)$$

або

$$\frac{1}{1-n} (\delta)^{1-n} \Big|_{\delta_0}^{\delta} = \frac{a(bE_1)^{n-1} \varphi^{1-n_1}}{b(1-n_1)} \Big|_0^{\varphi}.$$

Підставляємо верхнє і нижнє значення параметрів визначених інтегралів і отримуємо

$$(\delta)^{1-n} - (\delta_0)^{1-n} = \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1}.$$

В кінцевому вигляді отримаємо формулу

$$(\delta)^{1-n} = (\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1}. \quad (5.2.6)$$

Перейдемо до аналізу рівняння (5.2.6). Прологарифмуємо його ліву і праву частини

$$(1-n)\ln(\delta) = \ln\left[(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1}\right].$$

або

$$\ln(\delta) = \frac{\ln\left[(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1}\right]}{(1-n)}.$$

Визначимо швидкість зміни відносної деформації, для чого перейдемо в останньому рівнянні до межі

$$\lim_{n \rightarrow 1} \ln(\delta) = \lim_{n \rightarrow 1} \frac{\ln \left[(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1} \right]}{(1-n)}. \quad (5.2.7)$$

Використовуючи правило Лопітала будемо мати

$$\lim_{n \rightarrow 1} \ln(\delta) = \lim_{n \rightarrow 1} \frac{\ln \left[(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1} \right]}{(1-n)} = \lim_{n \rightarrow 1} \frac{F_1}{F_2}, \quad (5.2.8)$$

$$F_1 = \frac{d}{dn} \left\{ \ln \left[(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1} \right] \right\},$$

$$F_2 = \frac{d}{dn} (1-n) = -1.$$

Після перетворень, отримаємо

$$\lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{(\delta_0)^{1-n} \ln(\delta_0) - \frac{a\varphi^{1-n_1}(bE_1)^{n-1} \ln bE_1}{b(1-n_1)} + \frac{a\varphi^{1-n_1}(bE_1)^{n-1}}{b(1-n_1)} + \frac{a\varphi^{1-n_1}n(bE_1)^{n-1} \ln bE_1}{b(1-n_1)}}{(\delta_0)^{1-n} + \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \varphi^{1-n_1}}. \quad (5.2.9)$$

Після підстановки (5.2.9) в перше рівняння системи (5.2.8), перетворимо його до вигляду

$$\ln(\delta) = \lim_{n \rightarrow 1} \frac{F_1}{F_2} = \frac{(\delta_0)^{1-n} \left[\ln(\delta_0) - \frac{a\varphi^{1-n_1} \ln bE_1}{b(1-n_1)[bE_1(\delta_0)]^{1-n}} + \frac{a\varphi^{1-n_1}}{b(1-n_1)[bE_1(\delta_0)]^{1-n}} + \frac{a\varphi^{1-n_1}n \ln bE_1}{b(1-n_1)[bE_1(\delta_0)]^{1-n}} \right]}{(\delta_0)^{1-n} \left[1 + \frac{(1-n)a\varphi^{1-n_1}}{b(1-n_1)[bE_1(\delta_0)]^{1-n}} \right]}.$$

Спростимо останнє рівняння, отримаємо

$$\ln(\delta) = \frac{\ln(\delta_0) + \frac{a\varphi^{1-n_1}}{b(1-n_1)[bE_1(\delta_0)]^{1-n}} [1 - \ln bE_1(1-n)]}{1 + \frac{(1-n)a\varphi^{1-n_1}}{b(1-n_1)[bE_1(\delta_0)]^{1-n}}}. \quad (5.2.10)$$

Аналізуючи рівняння (5.2.10), при $n \rightarrow 1$, отримаємо

$$\ln(\delta) - \ln(\delta_0) = \frac{\frac{a\varphi^{1-n_1}}{b(1-n_1)}}{[bE_1(\delta_0)]^{1-n}}.$$

Остаточний вираз для визначення натягу ведучої гілки нитки буде мати вид

$$\delta = \delta_0 e^{\frac{a\varphi^{1-n_1}}{b(1-n_1)[bE_1(\delta_0)]^{1-n}}}. \quad (5.2.11)$$

Для випадку, коли коефіцієнт тертя є сталою величиною, враховуючи рівняння (5.2.3), при $n_1 = 0$ $f = \frac{a}{b}$, отримаємо з (5.2.11)

$$\delta = \delta_0 e^{\frac{f\varphi}{[bE_1(\delta_0)]^{1-n}}}. \quad (5.2.12)$$

Перетворимо останнє рівняння (5.2.12), враховуючи

$$\delta_0 = \frac{P_A}{bE_1[R+r(1-\delta_0)]}. \quad (5.2.13)$$

Тоді рівняння (5.2.12) приймає вигляд

$$\delta = \delta_0 e^{f\varphi \left(\frac{R+r(1-\delta_0)}{P_A} \right)^{1-n}}. \quad (5.2.14)$$

При $n = 1$, отримаємо

$$\delta = \delta_0 e^{f\varphi}. \quad (5.2.15)$$

Підставляємо (5.2.15) в перше рівняння системи (5.2.2), з урахуванням першого рівняння системи (5.2.3), будемо мати

$$\frac{dP}{ds} = f(bE_1)^n (\delta_0)^n e^{f\varphi n}.$$

Розділяємо в останньому диференційному рівнянні змінні і інтегруємо

$$\int_{P_A}^{P_B} dP = f(bE_1)^n (\delta_0)^n (R+r) \int_0^\varphi e^{f\varphi n} d\varphi.$$

Отримаємо

$$P_B - P_A = \frac{P_A^n (R+r)}{n[R+r(1-\delta_0)]^n} e^{fn\varphi} - \frac{P_A^n (R+r)}{n[R+r(1-\delta_0)]^n}.$$

В кінцевому вигляді, отримаємо

$$P_B = P_A \left[1 + \frac{P_A^{n-1} (R+r)}{n[R+r(1-\delta_0)]^n} (e^{fn\varphi} - 1) \right]. \quad (5.2.16)$$

При

$$P_B = P_A \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right]. \quad (5.2.17)$$

Вирішуємо (5.2.17) разом з системою рівнянь (5.2.1), тоді

$$P \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P[R+r(1-\delta)]^2} \right] = P_0 \left[1 - \frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_0[R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right].$$

В кінцевому вигляді будемо мати

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2[R+r(1-\delta)]^2} \right] - \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2[R+r(1-\delta_0)]^2} \right] \times (5.2.18) \\ \times \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right].$$

Кут $\varphi = \varphi_P + \varphi_{\text{см1}} + \varphi_{\text{см2}} - \varphi_{\text{уж1}} - \varphi_{\text{уж2}}$, можна розраховувати з використанням формул [2-3,18-21]

$$\varphi_{\text{см1}} = \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{\text{см2}} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right], \quad (5.2.19) \\ \varphi_{\text{уж1}} = \arccos \left[1 - \frac{B}{2P_0(R+r)^2} \right], \quad \varphi_{\text{уж2}} = \arccos \left[1 - \frac{B}{2P(R+r)^2} \right].$$

При $K \rightarrow 0$ $j(K) \rightarrow 1$, при $K \rightarrow K_{KP}$, $j(K) \rightarrow J(K_{KP})$ де K_{KP} - критичне значення скрутки. В останньому випадку коефіцієнт жорсткості нитки на вигин буде визначатися як для монопітки $B_0 = EI = E\pi d^4 / 64$. Тоді рівняння (5.2.18) прийме вид

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \frac{B_0}{2[R+r(1-\delta)]^2} - \\ - \frac{B_0}{2[R+r(1-\delta_0)]^2} \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right]. \quad (5.2.20)$$

Для нитки, коефіцієнтом жорсткості на вигин котрої можна знехтувати, отримаємо

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right]. \quad (5.2.21)$$

Для ідеальної гнучкої нитки, коли $\delta_0 = 0$, отримуємо відому формулу Л.Ейлера

$$P = P_0 e^{f\varphi}.$$

Рівняння (5.2.18) необхідно розв'язувати з використанням систем (5.2.2), (5.2.19). Отримана залежність представляє трансцендентне рівняння відносно P . Для його вирішення використовувались чисельні методи з застосуванням метода дихотомії. Для цього було розроблене спеціальне програмне забезпечення.

На рис.5.2.2 представлені графічні залежності натягу ведучої гілки нитки P для: капронової мононитки 36 Т (2 крива), $B=21,9\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,1\text{ мм}$, $R=0,5\dots 10\text{ мм}$, $\varphi_p=3,14\text{ рад}$, $n=0,85$; капронової комплексної нитки 28 Т (1 крива), $B=0,22\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,1\text{ мм}$, $R=0,5\dots 10\text{ мм}$, $\varphi_p=3,14\text{ рад}$, $n=0,85$.

Крива 3, яка є асимптотою для залежностей 1 і 2, побудована по формулі Л.Ейлера без урахування жорсткості нитки на вигин і зминання. Аналіз графічних залежностей показує, що для капронових ниток однакового діаметра ($r=0,1\text{ мм}$) характер зміни натягу суттєво відрізняється. Так для капронової комплексної нитки 28 Т (крива 1) натяг плавно зменшується. Після значення $R=0,4\text{ мм}$ бачимо різке зростання натягу. Це пов'язано із зростанням кута охоплення ниткою напрямної за рахунок мнучкості поверхні нитки в зоні контакту з напрямною.

Для капронової мононитки 36 Т можна спостерігати протилежну картину (крива 2). Натяг нитки буде зростати з ростом величини радіуса кривизни циліндричної направляючої поверхні. Це пов'язано зі зменшенням кута охоплення ниткою направляючої поверхні. У результаті розрахунків було встановлено, що реальний контакт по деякій дузі буде існувати тільки при $R \geq 1\text{ мм}$. На ділянці, коли $R < 1\text{ мм}$ фактичний контакт між ниткою і

циліндричною направляючою буде здійснюватися в точці. Таким чином, при збільшенні радіуса кривизни направляючої лінії 1 і 2 будуть наближатися до кривої 3, яка характеризує класичну залежність Л.Ейлера. З цього випливає, що для більшості напрямних поверхонь технологічного обладнання текстильної та трикотажної промисловості, коли радіус поперечного перерізу нитки є порівняним з радіусом кривизни циліндричної напрямної поверхні, необхідно враховувати як структуру нитки так і її реальні фізико-механічні характеристики.

Розробка програмного забезпечення для визначення натягу нитки, коефіцієнта жорсткості на вигин оптимального радіуса кривизни направляючої

Перейдемо до опису алгоритму для визначення натягу нитки, коефіцієнта жорсткості на вигин оптимального радіуса кривизни напрямної, які приводяться до вигляду $F(x)=0$ [2,3]. Для чисельного визначення коренів трансцендентного рівняння будемо використовувати метод дихотомії (половинного ділення відрізка) [2]. Даний метод хороший тим, що при будь-якій ширині відрізка $[a,b]$ збіжність процесу є гарантована. Число ітерацій при використанні методу дихотомії визначається за формулою

$$N \cong \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2},$$

де N - число ітерацій;

a, b - границі відрізка;

ε - точність визначення кореня трансцендентного рівняння.

Блок-схема алгоритма для визначення кореня трансцендентного рівняння з використанням метода дихотомії приведена на рис.5.2.3.

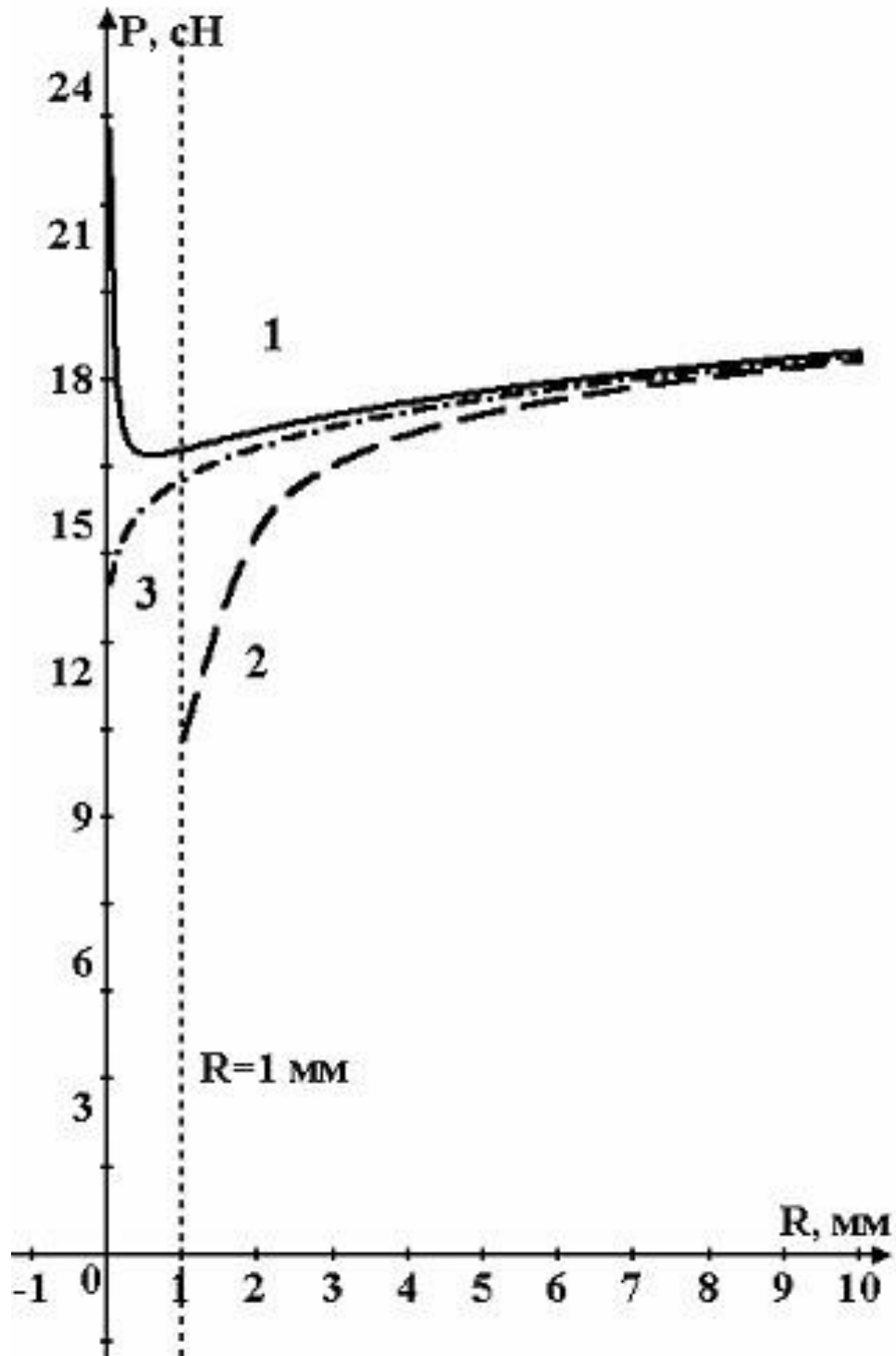


Рис.5.2.2. Залежності натягу ведучої гілки нитки від радіуса циліндричної напрямної

На початковому етапі задаються координати відрізка $[a, b]$ і точність визначення кореня трансцендентного рівняння ε . На наступному етапі визначається значення

$$x = \frac{a + b}{2}.$$

Дане значення x підставляється в вихідне рівняння $F(x)$. Після цього проводимо порівняння

$$F(x) > 0,$$

у випадку якщо умова виконується, то проводимо присвоєння

$$a = x.$$

В протилежному випадку

$$b = x.$$

На наступному етапі проводимо порівняння

$$b - a > \varepsilon.$$

У разі якщо умова виконується, то повертаємося до другого пункту алгоритму. В іншому випадку виводимо результат на друк, вважаючи його вірним із заданою точністю ε .

Для реалізації даного алгоритму була написана програма (див. Додаток) мовою Object Pascal в середовищі Delphi. Дана програма містить чотири модулі TUDI 1, TUDI 2, Uerrors і Synt. Перші три з них реалізуються з використанням трьох форм TfrmTUDI1, TfrmTUDI2, TFErrors. На рис. 5.2.4 представлена форма TfrmTUDI1.

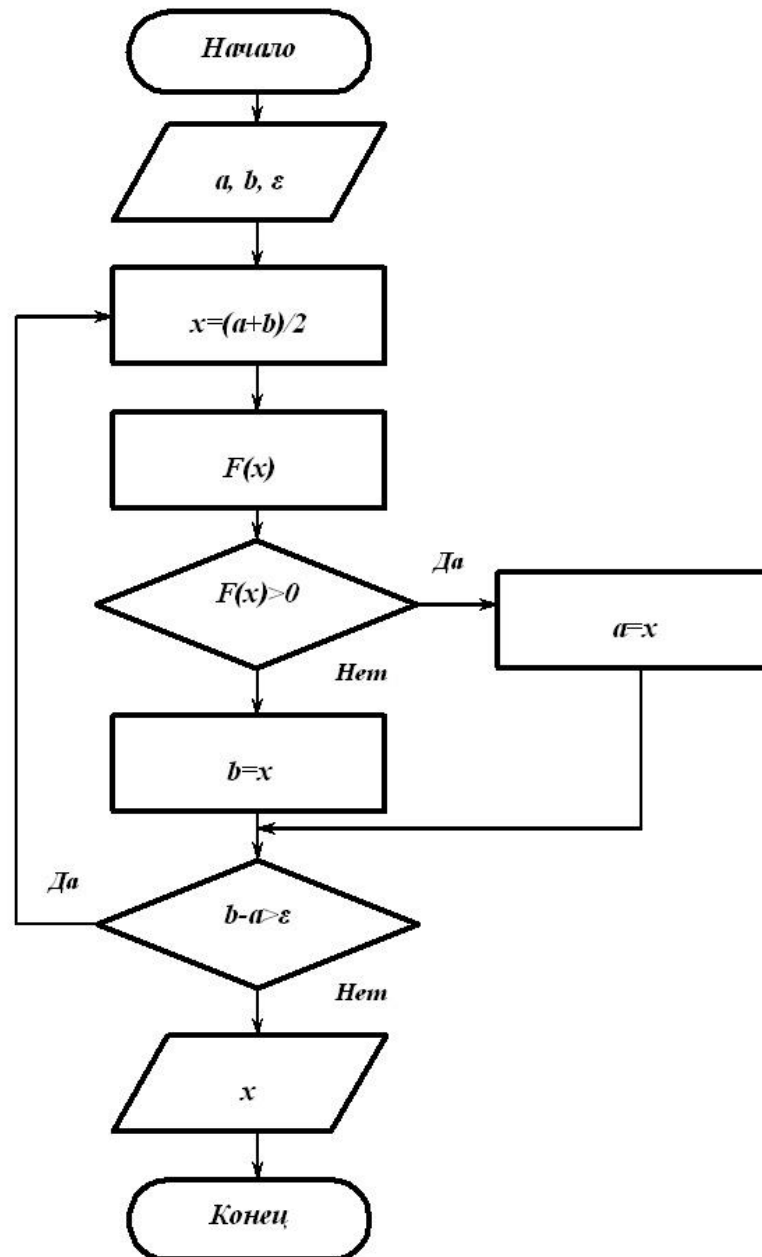


Рис.5.2.3. Блок-схема реалізації метода дихотомії для визначення кореней трансцендентного рівняння

Дана форма містить в собі наступні компоненти

```
TfrmTUDI1 = class(TForm)
```

```
  lb11TUDI: TLabel;
```

```
  lb12TUDI: TLabel;
```

```
  lb13TUDI: TLabel;
```

```
  lb14TUDI: TLabel;
```

```
  btn11TUDI1: TButton;
```

```
img1TUDI1: TImage;  
Image1: TImage;  
Image2: TImage;  
Image3: TImage;  
Image4: TImage;  
Image5: TImage;  
Label1: TLabel.
```

Реалізація процедури

```
procedure TfrmTUDI1.btn1TUDI1Click(Sender: TObject);  
begin  
  frmTUDI1.Hide;  
  frmTUDI2.Show;  
end;
```

призводить до закриття першої форми TfrmTUDI1 і відкриття другої, основної форми TfrmTUDI2, яка приведена на рис.5.2.5.



Рис.5.2.4. Форма TfrmTUDI1

Вона містить п'ять міток, одну кнопку і шість компонентів Timage.

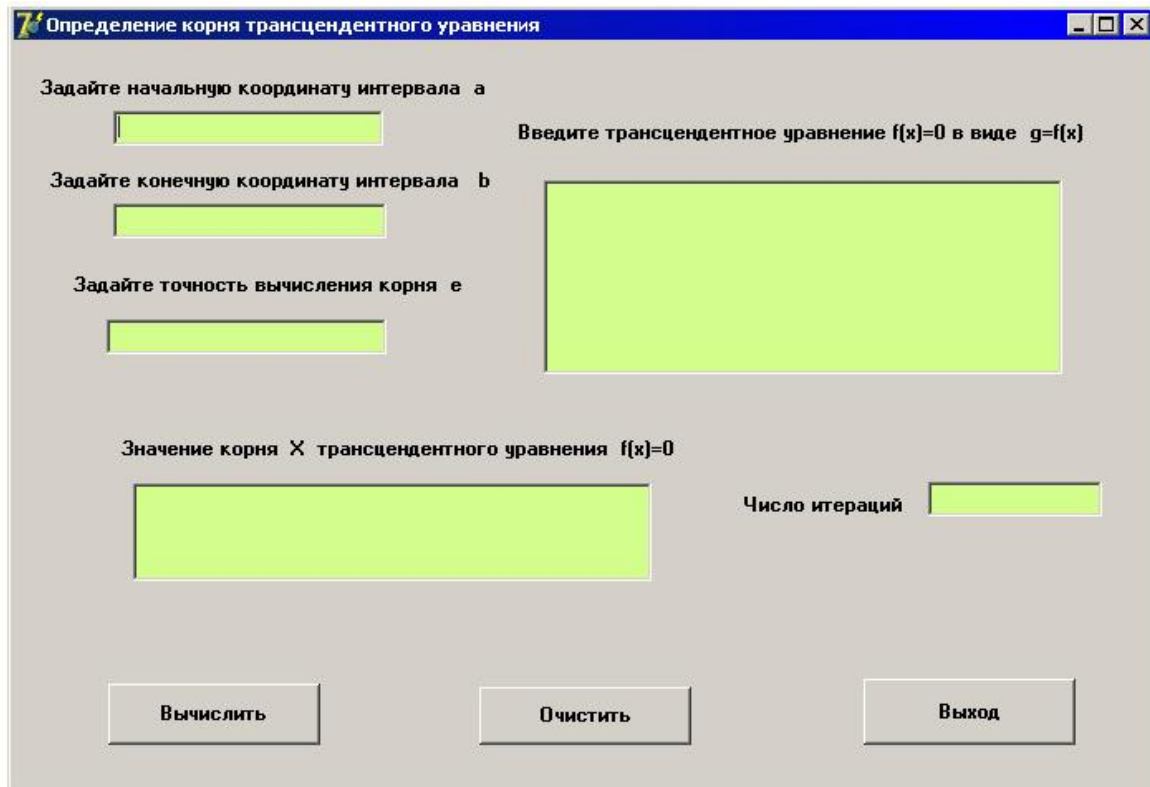


Рис.5.2.5. Основна форма TfrmTUDI2 програми

На даній формі TfrmTUDI2 містяться наступні компоненти

```

TfrmTUDI2 = class(TForm)
  lb121TUDI2: TLabel;
  lb122TUDI2: TLabel;
  lb123TUDI2: TLabel;
  lb124TUDI2: TLabel;
  lb125TUDI2: TLabel;
  lb126TUDI2: TLabel;
  edt21TUDI2: TEdit;
  edt22TUDI2: TEdit;
  edt23TUDI2: TEdit;
  Memo1: TMemo;
  mem21TUDI2: TMemo;
  edt24TUDI2: TEdit;

```

btn21TUDI2: TButton;

btn22TUDI2: TButton;

btn23TUDI2: Tbutton.

Це шість міток lb121TUDI2: TLabel; lb122TUDI2: TLabel; lb123TUDI2: TLabel; lb124TUDI2: TLabel; lb125TUDI2: TLabel; lb126TUDI2: TLabel; чотири однорядкових вікна редагування edt21TUDI2: TEdit; edt22TUDI2: TEdit; edt23TUDI2: TEdit; edt24TUDI2: Tedit; три кнопки btn21TUDI2: TButton;

btn22TUDI2: TButton; btn23TUDI2: Tbutton і два вікна багаторядкового редагування mem21TUDI2: Tmemo, Memo1: TMemo. Активація трьох кнопок btn21TUDI2: TButton; btn22TUDI2: TButton; btn23TUDI2: Tbutton дозволяє реалізувати три наступні процедури

```
procedure TfrmTUDI2.btn21TUDI2Click(Sender: TObject);
```

```
procedure TfrmTUDI2.btn23TUDI2Click(Sender: TObject);
```

```
begin
```

```
frmTUDI1.Close;
```

```
end;
```

```
procedure TfrmTUDI2.btn22TUDI2Click(Sender: TObject);
```

```
begin
```

```
edt21TUDI2.Clear;
```

```
edt22TUDI2.Clear;
```

```
edt23TUDI2.Clear;
```

```
edt24TUDI2.Clear;
```

```
mem21TUDI2.Clear;
```

```
Memo1.Clear;
```

```
End.
```

Перша процедура procedure TfrmTUDI2.btn21TUDI2Click (Sender: TObject) реалізовує основний обчислювальний процес. Друга процедура procedure TfrmTUDI2.btn23TUDI2Click (Sender: TObject) здійснює вихід з програми.

Третя процедура `procedure TfrmTUDI2.btn22TUDI2Click (Sender: TObject)` дозволяє очистити вміст однорядкових вікон редагування `edt 21 TUDI 2: TEdit; edt 22 TUDI 2: TEdit; edt 23 TUDI 2: TEdit; edt 24 TUDI 2: Tedit` і двох багаторядкових вікон редагування `mem 21 TUDI 2: Tmemo, Memo 1: Tmemo`.

Вікно `edt 21 TUDI 2: Tedit` служить для введення початкової межі інтервалу відрізка a . Вікно `edt22TUDI2: Tedit` служить для вводу кінцевої границі інтервала b . Вікно `edt23TUDI2: Tedit` служить для введення значення точності визначення кореня трансцендентного рівняння ε .

В багаторядкове вікно редагування `Мемо1: Tmemo` вводяться вхідні данні і основне рівняння $F(x) = 0$. Форма `TfrmTUDI2`, з введеними даними, показана на рис.5.2.6. Реалізація процедури `procedure TfrmTUDI2.btn21TUDI2Click(Sender: TObject)` запускає процес обрахунку кореня трансцендентного рівняння $F(x) = 0$.

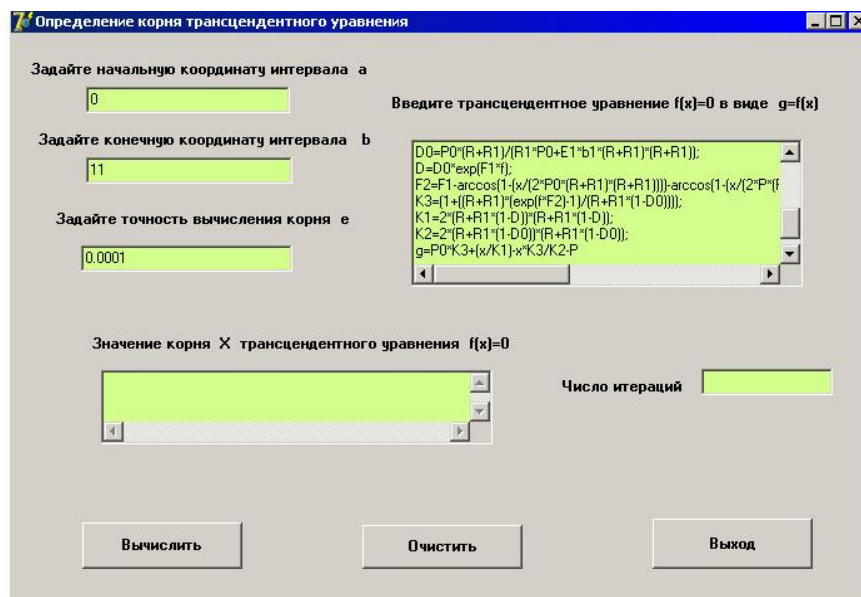


Рис.3.2.6. Форма `TfrmTUDI2` з введеними даними

На рис.5.2.7 показана форма `TfrmTUDI2` з результатами розрахунку кореня трансцендентного рівняння з заданою точністю ε .

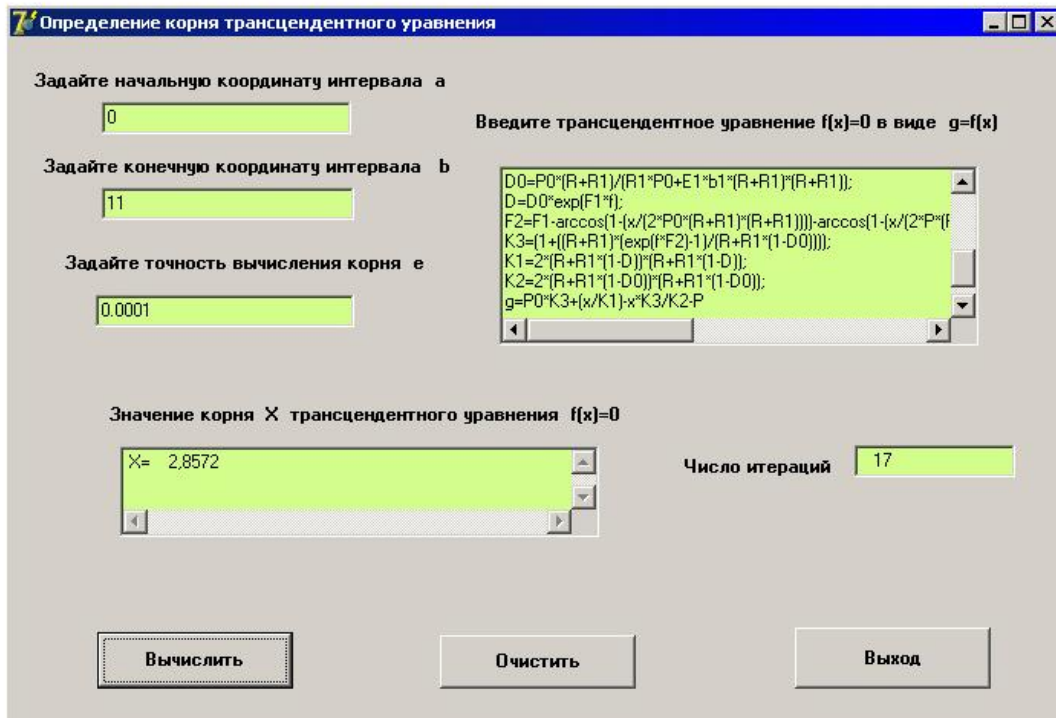


Рис.3.2.7. Результаты розрахунку корня трансцендентного рівняння

В однорядковому вікні редагування edt24TUDI2: Tedit показано число ітерацій N , яке відбувається при визначенні корня. В багаторядковому вікні редагування mem21TUDI2: Tmemo виведено значення корня.

Третя форма TFErrors = class(TForm) служить для виведення повідомлень про помилки при вводі вхідних даних в багаторядкове вікно редагування Memo1: Tmemo.

Окремий модуль Synt, який використовується в програмі, не має своєї форми і служить виключно для трансляції виразів, які введені в багаторядкове вікно редагування Memo 1: Tmemo. У даному трансляторі, відкритому для редагування, використовується дев'ять основних функцій. Відповідний фрагмент коду програми має вигляд

```
begin
  try
    case -PZ[i]-100 of
      1: Stack[High(Stack)]:=Exp(Stack[High(Stack)]);
```

```
2: Stack[High(Stack)]:=Sin(Stack[High(Stack)]);
3: Stack[High(Stack)]:=Cos(Stack[High(Stack)]);
4: Stack[High(Stack)]:=Sqrt(Stack[High(Stack)]);
5: Stack[High(Stack)]:=Abs(Stack[High(Stack)]);
6: Stack[High(Stack)]:=Ln(Stack[High(Stack)]);
7: Stack[High(Stack)]:=Tan(Stack[High(Stack)]);
8: Stack[High(Stack)]:=ArcTan(Stack[High(Stack)]);
9: Stack[High(Stack)]:=ArcCos(Stack[High(Stack)]);
End.
```

**Визначення натягу ведучої гілки нитки залежно від матеріалу,
структури і розмірів напрямних машин легкої і текстильної
промисловості**

Для визначення впливу структури, матеріалу ниток і розмірів циліндричної напрямної в роботі було реалізовано обчислювальний експеримент. Для цього були обрані нитки, основні характеристики яких наведені в таблиці 5.2.1.

Вирази (5.2.18) - (5.2.20) використовувалися для складання трансцендентних рівнянь. Трансцендентне рівняння, яке використовується для визначення натягу ведучої гілки нитки P з використанням метода дихотомії, має вигляд

$$f(P) - P = 0. \quad (5.2.22)$$

Тут кілька слів необхідно зауважити про специфіку обчислень окремих параметрів, які входять в рівняння (5.2.18).

Відносна деформація поперечного перерізу нитки в точці входу δ_0 і точці сходу нитки з напрямної δ є функцією початкового натягу P_0 , модуля

пружності нитки на зтискання E_1 , радіуса напрямної R і розрахункового радіуса поперечного перерізу нитки r , ширини сліда контакту b_1 (див. формули (5.2.11-5.2.14)). На рис.5.2.8 представлені графічні залежності зміни відносної деформації для різноманітних ниток.

Таблиця 5.2.1

Геометричні і механічні характеристики ниток

№	Нитка	Будова	Текс, Т	Розрахунковий діаметр d , мм	Модуль пружності на розтягнення E , сН/мм ²	Коефіцієнт жорсткості на згин B_0 , сН·мм ²	Модуль пружності на сжатие E_1 , сН/мм ²	Ширина сліда контакту b_1 , мм	Коефіцієнт тертя	
									a	b
1	Капронова (80 фл., $d_i=0,037$ мм)	Комплексна нитка	29	0,2	$4 \cdot 10^5$	0,37	814,2	0,012	0,1765	0,1186
2	Бавовна (80 фл., $d_i=0,039$ мм)	Пряжа	30,8	0,22	$3 \cdot 10^5$	0,11	625,5	0,014	0,1656	0,059
3	Вовняна (80 фл., $d_i=0,042$ мм)	Пряжа	29,9	0,23	$1,5 \cdot 10^5$	0,23	480,9	0,015	0,1330	0,091
4	Віскозна (80 фл., $d_i=0,041$ мм)	Пряжа	29	0,22	$4,5 \cdot 10^5$	0,62	1254,8	0,011	0,1580	0,114
5	Капронова	Мононитка	36	0,2	$4 \cdot 10^5$	31,4	$2,6 \cdot 10^4$	-	0,1765	0,1186
6	Капронова	Комплексна нитка	280,5	0,64	$4 \cdot 10^5$	3,5	814,2	0,012	0,1765	0,1186
7	Капронова	Комплексна нитка	174	0,48	$4 \cdot 10^5$	2,2	814,2	0,012	0,1765	0,1186
8	Віскозна	Пряжа	93,5	0,4	$4,5 \cdot 10^5$	1,9	1254,8	0,011	0,1580	0,114

9	Бавовна	Пряжа	100	0,41	$3 \cdot 10^5$	0,36	625,5	0,014	0,1656	0,059
10	Вовняна	Пряжа	89,7	0,4	$1,5 \cdot 10^5$	0,69	480,9	0,015	0,1330	0,091

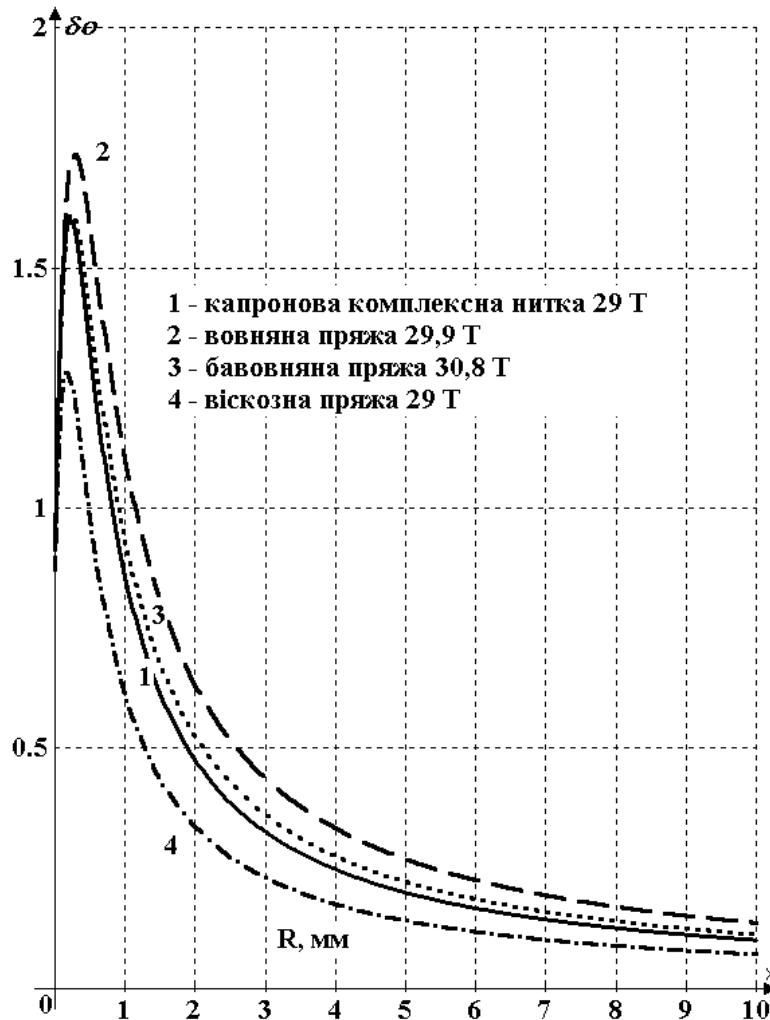


Рис.5.2.8. Графічні залежності зміни відносної деформації поперечного перерізу ниток

Для комплексних ниток і прямих, з невисокою круткою, величина відносної деформації має максимальне значення рівне 1. Це відповідає випадку, коли абсолютна деформація перерізу в точці контакту з напрямною буде дорівнювати розрахунковому радіусу поперечного перерізу нитки. Аналіз графіків показує, що розрахункове значення відносної деформації

поперечного перерізу нитки на інтервалі від 0,1 до 1,2 мм перевищує 1. Це відноситься і до відносної деформації поперечного перерізу в точці сходу з направляючої. В даному випадку, в програмі передбачалась спеціальна процедура порівняння, яка здійснювала коригування. В результаті розв'язання трансцендентного рівняння (5.2.22), з використанням чисельного методу дихотомії, були отримані значення натягу ведучої гілки нитки з урахуванням мнучкості і жорсткості на вигин, які наведені у таблиці 5.2.2.

Таблиця 5.2.2

Значення натягу ведучої гілки нитки і коефіцієнта жорсткості на вигин

Нитка		Радіус R, мм												
		0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	2	3	4	5
Капронова	P	-	16.5	16.4	16.3	16.2	16.2	16.2	16.1	16.0	16.0	16.2	16.4	16.6
	B_0	-	-	-	-	1.96	1.81	1.69	1.58	1.49	0.87	0.59	0.45	0.36
Вовняна	P	-	16.7	16.1	15.8	15.6	15.4	15.2	15.1	15.1	14.7	14.7	14.8	14.9
	B_0	-	-	-	-	2.57	2.33	2.17	2.03	1.91	1.14	0.78	0.58	0.46
Бавовна	P	-	21.2	19.7	18.9	18.4	18.0	17.8	17.6	17.4	16.6	16.5	16.52	16.6
	B_0	-	-	-	-	2.26	2.06	1.92	1.79	1.69	0.99	0.68	0.51	0.4
Віскозна	P	-	15.3	15.2	15.19	15.17	15.15	15.05	15.0	14.9	15.1	15.3	15.6	15.7
	B_0	-	-	-	-	1.38	1.28	1.18	1.1	1.03	0.58	0.39	0.29	0.23
Моно-нитка	P	-	-	-	-	-	-	-	-	10	11.5	12.9	13.9	14.6

На рис.5.2.9 представлені графічні залежності натягу ведучої гілки нитки залежно від радіуса кривизни напрямної поверхні, які були отримані на підставі даних таблиці.

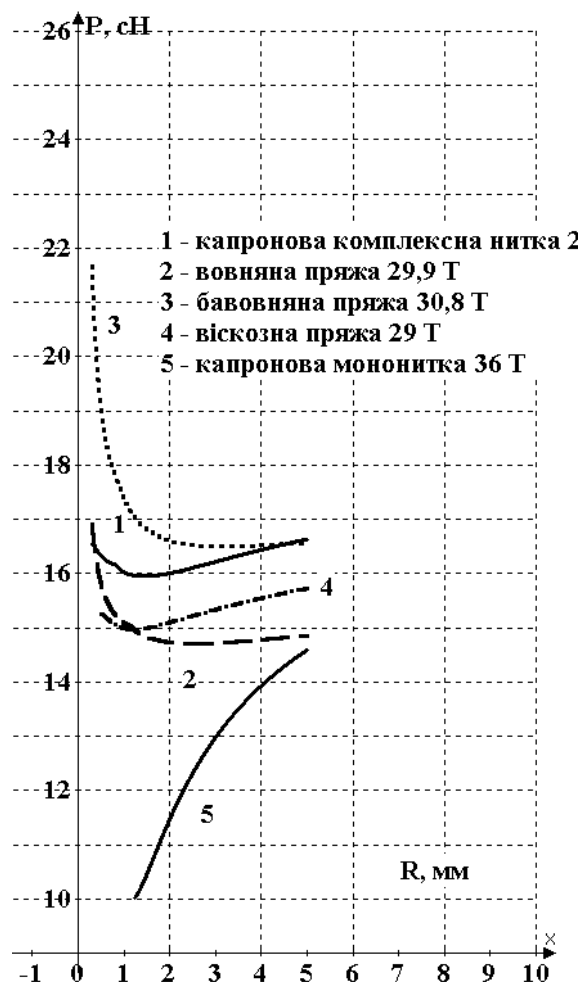


Рис.5.2.9. Залежність натягу ведучої гілки нитки від радіуса кривизни напрямної поверхні

Аналіз даних графічних залежностей показує, що натяг нитки при зменшенні радіуса кривизни спочатку спадає, а потім починає різко зростати. Це пояснюється збільшенням реального кута охоплення ниткою направляючої за рахунок мнучкості в зоні контакту. Тут спостерігається подвійний ефект. З одного боку реальний кут охоплення зменшується за рахунок жорсткості нитки на вигин, коли радіус кривизни направляючої порівняний з розрахунковим радіусом нитки, а з іншого боку відбувається збільшення кута за рахунок мнучкості. Це добре видно на графіку 5

рис.3.3.2. Він отриманий для капронової мононитки, у якій мнучкість відсутня. Натяг мононитки зі зменшенням радіуса кривизни падає.

На рис.5.2.10 представлені графічні залежності зміни натягу капронових ниток різної товщини.



Рис.5.2.10. Залежність зміни натягу капронових комплексних ниток від радіуса кривизни напрямної

Аналіз даних залежностей показує, що для ниток більшого діаметра розмах коливань значення натягу ведучої гілки нитки буде більшим.

Визначення оптимального значення коефіцієнта жорсткості нитки на вигин

Представляє теоретичний і практичний інтерес для визначення коефіцієнта жорсткості нитки на вигин B_0 , при якому натяг ведучої гілки

нитки, зі зміною радіуса кривизни напрямної R , буде змінюватися по закону Л.Эйлера (див. підр.5.2). В цьому випадку перетворимо рівняння (5.2.20) до вигляду

$$P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \frac{B_0}{2[R+r(1-\delta)]^2} - \frac{B_0}{2[R+r(1-\delta_0)]^2} \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] - P_0 e^{f\varphi} = 0. \quad (5.2.23)$$

Рівняння (3.4.1) є трансцендентним рівнянням. Його розв'язання, з урахуванням залежностей (5.2.3), (5.2.19), дозволяє визначити значення коефіцієнта жорсткості нитки на вигин. В таблиці 3.3.2 представлені значення коефіцієнта B_0 для різноманітних видів ниток.

Розв'язок трансцендентного рівняння (5.2.23) здійснювалось у відповідності до методики, приведеної в підрозділі 3.2. На рис.5.2.11 представлені графічні залежності зміни коефіцієнта жорсткості нитки на вигин в залежності від радіуса кривизни напрямної.

Залежність (5.2.23) показує, що для комплексних ниток і пряжі коефіцієнт жорсткості нитки на вигин залежить від величини крутки K

$$B_0 = \left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{j(K)}, \quad j(K) = 1 + u_1 K^{u_2}, \quad (5.2.24)$$

де u_1, u_2 - деякі сталі коефіцієнти.

Використовуючи дані таблиці 5.2.2, а також дані, приведені в роботі [1], для капронової комплексної нитки 29Т отримані наступні дані

$$\begin{aligned} K = 20, & \quad B_0 = 0,139, \\ K = 100, & \quad B_0 = 0,189. \end{aligned} \quad (5.2.25)$$

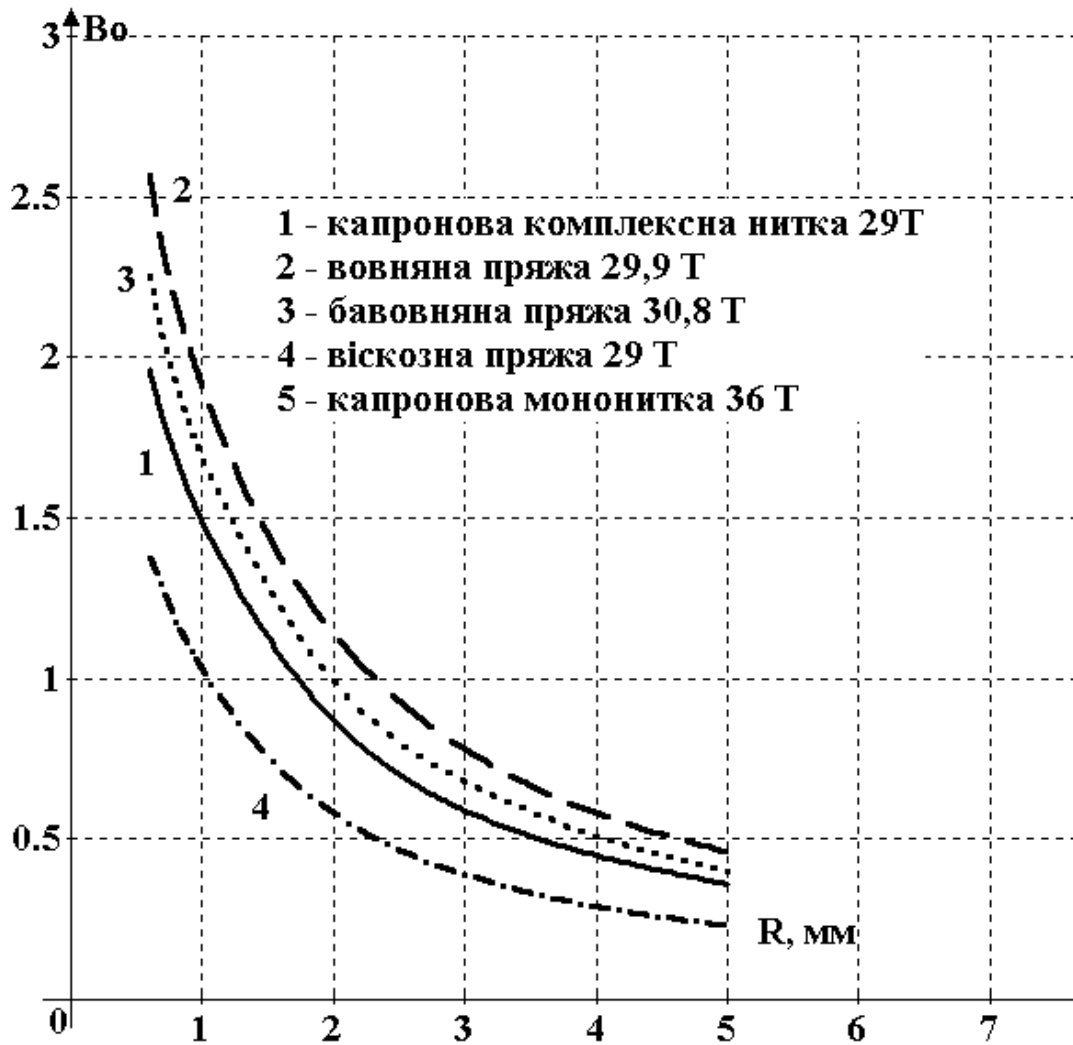


Рис.5.2.11. Зміна коефіцієнта жорсткості нитки на вигин в залежності від радіуса кривизни напрямної

Вирішуючи сумісно (5.2.24)-(5.2.25) отримаємо наступну систему трансцендентних рівнянь для визначення u_1, u_2

$$0,139 = (4,3 \cdot 10^{-5})^{(1+u_1 20^{u_2})},$$

$$0,189 = (4,3 \cdot 10^{-5})^{(1+u_1 100^{u_2})}.$$

Чисельне вирішення данні системи трансцендентних рівнянь здійснювалось з використанням модифікованого метода Ньютона (програма приведена в Додатку). В результаті були отримані наступні значення коефіцієнтів u_1, u_2

$$U_1 = -0,7498 \quad U_2 = 0,023.$$

Дана методика дозволяє отримувати необхідний коефіцієнт жорсткості нитки на вигин задавши нитці зазначену крутку.

Визначення оптимального радіуса кривизни напрямних пристроїв технологічного обладнання

Дуже важливим є визначення оптимальних розмірів нитконапрямних пристроїв машин легкої промисловості. Для розрахунків були обрані капронова комплексна нитка 29Т, бавовняна пряжа 30,8Т, вовняна пряжа 29,9Т, віскозна пряжа 29Т. Дані нитки мають приблизно рівний діаметр поперечного перерізу, а їх геометричні та механічні властивості наведені в таблиці 5.2.2. Натяг веденої гілки дорівнював $P_0=10$ сН. Розрахунковий кут охоплення дорівнював $\varphi_P=3,14$ рад.

Вираз для визначення натягу ведучої гілки нитки, з урахуванням (5.2.20), представимо у вигляді

$$P = z_6(R) + z_7(R) - z_8(R),$$

де

$$\begin{aligned}
 z_0(R) &= f = a \left(\frac{R}{P_0} \right)^b, \\
 z_1(R) &= \delta_0 = \frac{P_0(R+r)}{rP_0 + E1b_1(R+r)^2}, \\
 z_2(R) &= \delta = z_1(R)e^{\varphi_P z_0(R)}, \\
 z_3(R) &= \varphi_P - 2 \arccos \left[1 - \frac{B_0}{2P_0(R+r)^2} \right] - \arccos \left[1 - z_1(R) \left(\frac{2r}{R} \right)^2 \right] - \arccos \left[1 - z_2(R) \left(\frac{2r}{R} \right)^2 \right], \\
 z_5(R) &= e^{z_0(R)z_3(R)} - 1, \\
 z_6(R) &= P_0 \left[1 + \frac{(R+r)z_5(R)}{R+r[1-z_1(R)]} \right], \\
 z_7(R) &= \frac{B_0}{2[R+r(1-z_2(R))]^2}, \\
 z_8(R) &= \frac{B_0}{2[R+r(1-z_1(R))]^2} \left[1 + \frac{(R+r)z_5(R)}{[R+r(1-z_1(R))]} \right].
 \end{aligned}$$

Оптимальне значення радіуса кривини R напрямною, при якому натяг ведучої гілки P буде мінімальним, визначається з наступної рівності

$$\frac{dP}{dR} = \frac{dz_6(R)}{dR} + \frac{dz_7(R)}{dR} - \frac{dz_8(R)}{dR} = 0. \quad (5.2.26)$$

Для розв'язання рівняння (3.5.1) необхідно знайти відповідні похідні функцій $z_0(R), z_1(R), z_2(R), z_3(R), z_5(R)$ по радіусу R , які будуть використовуватись при визначенні похідних. Їх значення визначаються з наступних виразів

$$\begin{aligned} \frac{dz1(R)}{dR} &= \frac{d\delta_0}{dR} = \frac{P_0}{P_0 r + E1b_1(R+r)^2} - \frac{2E1P_0b_1(R+r)^2}{[P_0 r + E1b_1(R+r)^2]^2}, \\ \frac{dz2(R)}{dR} &= \frac{d\delta}{dR} = \frac{P_0 e^{a\varphi_P \left(\frac{R}{P_0}\right)^b}}{P_0 r + E1b_1(R+r)^2} - \frac{2E1P_0b_1 e^{a\varphi_P \left(\frac{R}{P_0}\right)^b} (R+r)^2}{[P_0 r + E1b_1(R+r)^2]^2} + \\ &\quad + \frac{ab\varphi_P e^{a\varphi_P \left(\frac{R}{P_0}\right)^b} \left(\frac{R}{P_0}\right)^{b-1} (R+r)}{P_0 r + E1b_1(R+r)^2}, \\ z4(R) &= \frac{d}{dR} z3(R) = \frac{2B_0}{P_0 (R+r)^3 \sqrt{1 - \left[\frac{B_0}{2P_0 (R+r)^2} - 1\right]^2}} - \\ &\quad - \frac{8r^2 z2(R)}{R^3} - \frac{4r^2 \frac{d}{dR} z2(R)}{R^2} - \frac{8r^2 z1(R)}{R^3} - \frac{4r^2 \frac{d}{dR} z1(R)}{R^2}, \\ \frac{dz5(R)}{dR} &= e^{z0(R)z3(R)} \left[z0(R) \frac{dz3(R)}{dR} + z3(R) \frac{dz0(R)}{dR} \right]. \end{aligned}$$

Визначимо похідні функцій $z6(R), z7(R)$, що входять у вираз (5.2.26)

$$\begin{aligned} \frac{dz6(R)}{dR} &= P_0 \left[\frac{z5(R)}{R-r[z1(R)-1]} + \frac{\frac{dz5(R)}{dR}(R+r)}{R-r[z1(R)-1]} + \frac{z5(R) \left(r \frac{dz1(R)}{dR} - 1 \right) (R+r)}{[R-r[z1(R)-1]]^2} \right], \\ \frac{dz7(R)}{dR} &= \frac{B_0 \left[1 + (R+r) \frac{z5(R)}{[R+r[1-z1(R)]]} \right]}{2[R+r[1-z1(R)]]^2}. \end{aligned}$$

Для функції $z8(R)$ вираз першої похідної буде мати вигляд

$$\frac{dz_8(R)}{dR} = \frac{B_0 \left[\frac{z_5(R)}{R - r[z_1(R) - 1]} + \frac{\frac{dz_5(R)}{dR}(R+r)}{R - r[z_1(R) - 1]} + \frac{z_5(R) \left(r \frac{dz_1(R)}{dR} - 1 \right) (R+r)}{[R - r[z_1(R) - 1]]^2} \right]}{2[R - r[z_1(R) - 1]]^2} + \frac{B_0 \left[r \frac{dz_1(R)}{dR} - 1 \right] \left[\frac{z_5(R)(R+r)}{R - r[z_1(R) - 1]} + 1 \right]}{[R - r[z_1(R) - 1]]^3}.$$

Підставляючи останні вирази похідних в (5.2.26) отримаємо трансцендентне рівняння виду $F(R) = 0$. Для його розв'язання будемо використовувати алгоритм, що реалізує модифікований метод Ньютона для системи трансцендентних рівнянь і програму, яка приведена в Додатку.

В результаті розрахунків були отримані наступні оптимальні значення радіуса кривизни напрямної: 1,5 мм – для капронової комплексної нитки 29Т; 3,4 мм – для бавовняної пряжі 30,8Т; 2,5 мм – для вовняної пряжі 29,9Т; 1 мм – для віскозної пряжі 29Т.

5.3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ УМОВ ВЗАЄМОДІЇ НИТОК З НАПРЯМНИМИ ПОВЕРХНЯМИ

В четвертому розділі наводяться результати досліджень по визначенню впливу різних чинників на зміну натягу ведучої гілки нитки при її взаємодії з направляючою поверхнею. Дослідження процесу взаємодії ниток з направляючими поверхнями вимагає проведення цілої серії дослідів для підтвердження коректності зроблених допущень при виведенні теоретичних залежностей в розділах 2 і 3 і визначення меж використання останніх.

Активне планування експерименту, використання методів математичної статистики дозволяє отримати регресійні залежності, які описують процес з необхідною точністю, при цьому число чинників може

досягати значної величини. Застосування ЕОМ дозволяє легко визначати величину коефіцієнтів в рівняннях регресії, їх значущість. Отримані залежності, після відповідних перетворень, можна використовувати для оптимізації процесу взаємодії нитки з направляючою поверхнею.

Розробка установок для проведення експерименту. Обґрунтування вибору вимірювально-реєструючої апаратури

Для реалізації плану експериментальних досліджень потрібно було розробити спеціальні установки. Велике значення має обґрунтування і вибір вимірювально-реєструючої апаратури, яка є невід'ємною частиною будь-якої експериментальної установки.

Вивчення процесу взаємодії ниток з направляючими поверхнями і нитками, що виступають як напрямні, коли кривизна поверхні співрозмірна з радіусом поперечного перетину нитки, призвело до необхідності імітації даного виду взаємодії в експериментальних установках.

На думку автора, подальший розвиток питання по експериментальному дослідженню процесу взаємодії ниток з напрямними повинен йти шляхом розробки універсальних, багатофункціональних установок, що дозволяють проводити комплексні дослідження процесу взаємодії ниток з напрямними, імітувати реальні умови, що виникають при переробці ниток на технологічному устаткуванні.

Для здійснення експериментальних досліджень нами були розроблені прилади для визначення натягу ведучої гілки нитки; для визначення фрикційних властивостей ниток в умовах, наближених до реальних, таких, що виникають на технологічному устаткуванні. Дві з цих установок захищені авторськими свідоцтвами (патент України №31888 та патент України №45939).

Аналог першого з розроблених пристроїв містить шків, який встановлено з можливістю горизонтального переміщення відносно осі обертання, причому одна його торцева сторона має нахил та упирається в ролик, а інша контактує з пружиною. Гнучка стрічка з жорсткістю в повздовжньому та поперечному напрямках, що з'єднана з вантажем з однієї сторони і робочим рухомим органом динамометра з іншої сторони, має на нижній поверхні зі сторони шківу прикріплений зразок нитки, яка досліджується, та розташовується між трьома парами роликів, що закріплені на двох кронштейнах. Кронштейни роликів з'єднані з робочими рухомими органами динамометрів.

Але розвиток сучасного машинобудування та технологій в легкій та текстильній промисловості призводить до різкого збільшення швидкості руху нитки. Як відомо зі збільшенням швидкості руху нитки змінюється і величина коефіцієнту тертя [1]. Наявність вищої кінематичної пари (кулачкової), яку утворюють ролик та шків з похилою торцевою стороною і пружини для силового замикання, не дозволяє використовувати його при високих швидкостях обертання шківу. Це пояснюється наявністю розривів неперервності в графіках прискорення та похідних від прискорення будь-яких порядків, що мають місце в кулачкових механізмах [4].

Це призводить до того, що в процесі роботи з'являються удари, відривання шківів від ролика. Це спричиняє нестационарний рух шківів в поперечному напрямку відносно осі обертання і не дозволяє визначити коефіцієнт тертя в поперечному напрямку при великих швидкостях руху нитки.

Задачею запропонованої моделі є розширення технологічних можливостей за рахунок збільшення діапазону вимірювання фрикційних властивостей ниток при великих швидкостях руху.

Поставлена задача досягається тим, що прилад для визначення

коефіцієнту тертя ниток, який містить шків, встановлений з можливістю горизонтального переміщення відносно осі обертання ведучого валу, гнучку стрічку з жорсткістю в повздовжньому та поперечному напрямках, яка з'єднана з вантажем з однієї сторони та робочим рухомим органом динамометра з іншої сторони, на нижній поверхні якої зі сторони шківу закріплено зразок нитки, що досліджується, і розташовується між трьома парами роликів, які закріплені на двох кронштейнах, які з'єднані з робочими рухомими органами динамометрів, оснащений двома конічними зубчастими колесами, які жорстко закріплені на валах, які перетинаються та розташовані в опорах, один з яких є ведучим, а другий веденим, на веденому валові розташовано кривошип, який кінематично зв'язаний з шатуном, а шатун зв'язаний з повзуном, який розташовується в горизонтальних направляючих, вилкоподібним кронштейном, на кінцях якого розміщені двосторонні опорні підшипники ковзання, що зв'язані з шипами, закріпленими з торців шківу, який жорстко встановлено на повзуні.

Порівняльний аналіз з прототипом показує, що заявлений прилад для визначення коефіцієнту тертя ниток відрізняється тим, що на валах, які перетинаються та розташовані в опорах, один з яких є ведучим, а другий веденим, розташовано два конічних зубчастих колеса, на веденому валу розташовано кривошип, який кінематично зв'язаний з шатуном, а шатун зв'язаний з повзунком, що розміщений в горизонтальних направляючих, вилкоподібним кронштейном, на кінцях якого розташовані двосторонні упорні підшипники ковзання, що зв'язані з шипами, які закріплено з торців шківу, який жорстко встановлений на повзуні. Ознаки, що були перераховані вище, дозволяють виміряти коефіцієнт тертя ниток в повздовжньому і поперечному напрямках при великих швидкостях руху, що дає можливість широко досліджувати фрикційні властивості матеріалу.

На рис.5.3.1 зображена схема приладу. На схемі умовно не показані гнучка стрічка, вантаж і рухомі робочі органи динамометрів для визначення коефіцієнтів тертя у повздовжньому і поперечному напрямках.

Прилад для визначення коефіцієнта тертя ниток містить шків 1, встановлений з можливістю горизонтального переміщення відносно осі обертання ведучого вала 2, гнучку стрічку із жорсткістю в повздовжньому й поперечному напрямку, з'єднаний з вантажем з однієї сторони й робочим рухомим органом динамометра з іншої сторони, на нижній поверхні якого з боку шківа закріплений зразок нитки, що досліджується, й розташовується між трьома парами роликів, укріпленими на двох кронштейнах, які з'єднані з робочими рухомими органами динамометрів, який має два конічні зубчасті колеса 3, що жорстко закріплені на перетнутих валах, що розташовуються в опорах, один із яких є ведучим 2, а другий веденим. На веденому валу розташованій кривошип 5, кінематично зв'язаний з шатуном 5, а шатун зв'язаний з повзуном 7, розташованим у горизонтальних направляючих.

Вилкоподібний кронштейн 8, на кінцях якого розміщені двосторонні опорні підшипники ковзання 9, які зв'язані із шипами 10, що закріплені з торців шківа 1, жорстко закріпленій на повзуні 7, шипи 10 розташовуються з боків шківа 1.

Прилад працює в такий спосіб. Під дією приладу рух від ведучого вала 2 передається шківу 1 і через конічні зубчасті колеса 3 і 4 кривошипу 5. Через кінематично зв'язаний з ним шатун 6 і повзун 7 вилкоподібний кронштейн 8 отримує зворотно-поступальний рух, який через двосторонні опорні підшипники 9 і шипи 10 передається шківу 1. При цьому, навіть при більших швидкостях обертання ведучого вала 2 шків 1 рухається за заданим законом.

Використання запропонованого приладу дозволяє значно розширити діапазон вимірюваних величин шляхом визначення коефіцієнта тертя

ниток при більших швидкостях руху.

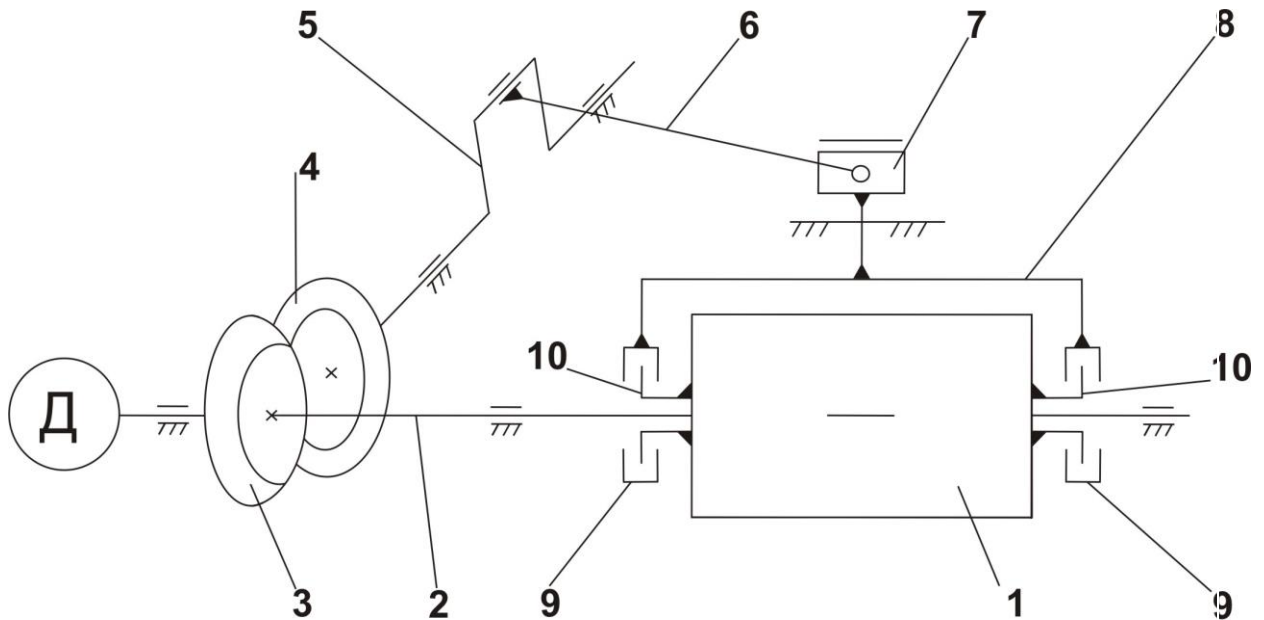


Рис.5.3.1. Схема вимірювального вузла (патент України №31888)

Аналог другого з розроблених пристроїв містить змонтовані в корпусі бобіноутримувач, нитконапрячник, нитконатяжник, вимірювальний вузол у вигляді каретки з трьома нерухомо закріпленими на ній ниткопровідниками, встановлений в напрямній корпусу з можливістю переміщення під дією нитки, пристрій для переміщення нитки та ролики для нитки.

Але використання даного пристрою не дозволяє визначати фрикційні властивості ниток для різних кутів охоплення нитконапрячника. В сучасних трикотажних та текстильних машинах кут охоплення ниткою різних нитконапрямувачів вимірюється в широких межах (від 10° до 180°). Як відомо, при зміні кута охоплення фрикційні властивості суттєво змінюються [4].

Метою запропонованої моделі є розширення технологічних можливостей

вимірювального пристрою.

Поставлена мета досягається тим, що пристрій для визначення фрикційних властивостей нитки, який містить змонтовані в корпусі бобіноутримач, нитконапрячник, каретку з закріпленим кронштейном та вказівником, направляючу каретки з приводом, металеву нескінченну стрічку, що обгинає шків з упором, що несе підпружинений контакт, пристрій для протягування нитки, підтримуючі ролики, вимірювальну шкалу згідно з корисною моделлю містить два рухомих шківів, вісі яких з пристроями фіксації та вказівниками розташовані в наскрізному горизонтальному пазу каретки, нитконапрячник з пристроєм фіксації та вказівником, які розташовані в наскрізному вертикальному пазі каретки, горизонтальну та вертикальну шкалу.

Порівняльний аналіз з прототипом показує, що заявлений пристрій для визначення фрикційних властивостей нитки відрізняється тим, що містить два рухомих шківів, вісі яких з пристроями фіксації та вказівниками розташовані в наскрізному горизонтальному пазу каретки, нитконапрячник з пристроєм фіксації та вказівником, які розташовані в наскрізному вертикальному пазу каретки, горизонтальну і вертикальну шкалу. Ознаки, які були перераховані вище, дозволяють вимірювати фрикційні властивості ниток при зміні кута охоплення нитконапрячників, що дає змогу широко досліджувати фрикційні властивості матеріалу.

На Рис.5.3.2 зображена схема пристрою (центральна каретка вимірювального вузла показана в загальному вигляді). На рис.5.3.3 показана центральна каретка вимірювального вузла.

Пристрій для визначення фрикційних властивостей нитки містить в корпусі 1 приладу бобіноутримач 2, нитконапрячник 3, нитконатягувач 4, вимірювальний вузол в вигляді каретки 5. Каретка 5 встановлена в направляючій 7 корпусу 1 та містить індикаторну стрілку 8 та фігурний кронштейн 9. Нитка отримує рух від пристрою 11. Індикаторна лінійка 10

промаркована в одиницях, які характеризують фрикційні властивості нитки. На корпусі приладу змонтовані рухомі рамки 12-14 для переміщення нитки. Направляюча 7 встановлена з можливістю повздовжнього переміщення за допомогою механізму 15 для переміщення відносно нерухомо змонтованих підшипником ковзання 16. Нижче направляючої 7 на корпусі 1 змонтовані два шківів 17 та 18 з нескінченною гнучкою металевою стрічкою 19, на якій закріплена металева пластинка 20 з контактом 21 та пружиною 22. Два рухомих шківів 23 розташовуються на осях з пристроями фіксації та вказівниками 25, які розміщені в наскрізному горизонтальному пазу 26 каретки 5.

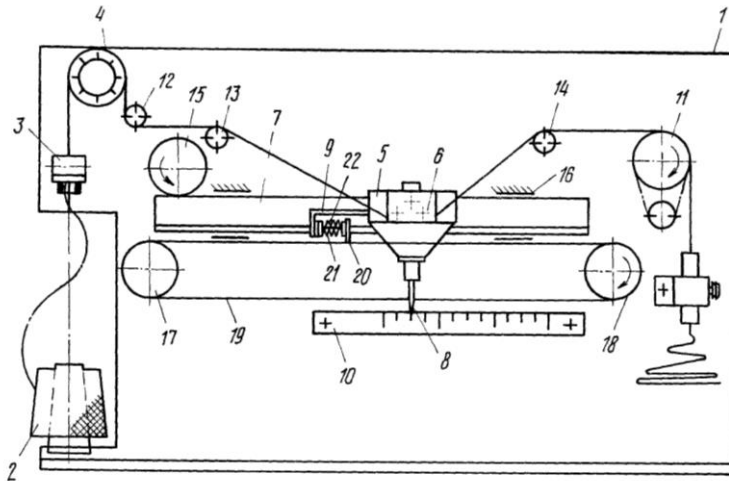


Рис.5.3.2. Схема вимірювального пристрою (патент України №45939)

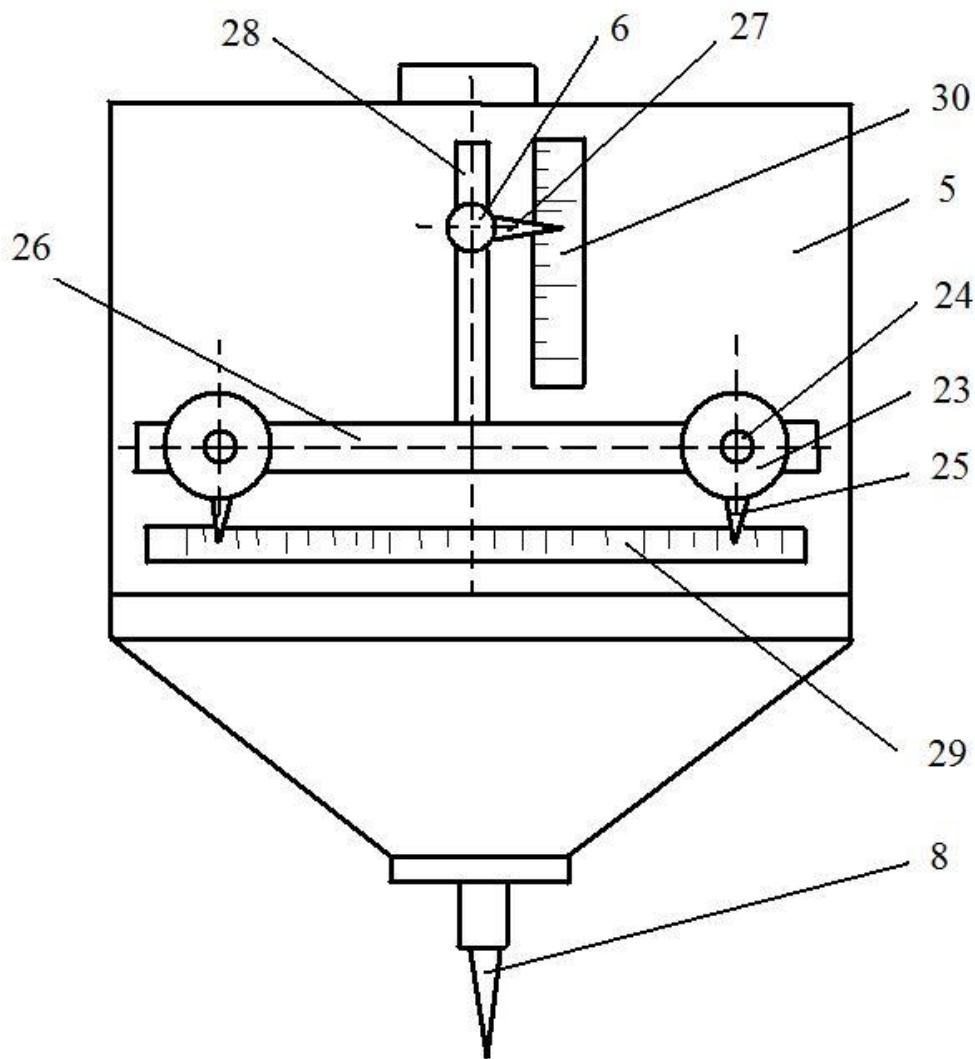


Рис.5.3.3. Центральна каретка вимірювального пристрою(патент України №45939)

Нитконапрямок 6 з пристроєм фіксації та вказівником 27 розташовані в наскрізному вертикальному пазу 28 каретки 5. Горизонтальна 29 і вертикальна 30 шкали мають метричну шкалу.

Пристрій працює наступним чином. Послаблюючи пристрої фіксації розташовують рухомі шківні 23 на осях 24 та нитконапрямок 6 відносно горизонтальної 29 та вертикальної 30 шкал таким чином, щоб отримати необхідний кут охоплення ниткою нитконапрямка 6. Контроль здійснюється за показниками вказівників 25 та 27. Переміщенням каретки 5 встановлюють індикаторну стрілку 8 на нульове положення індикаторної

лінійки 10. Прокручуючи шків 18 проти годинникової стрілки встановлюють пластину 20 з силовим контактом 21 з правого боку від робочої грані фігурного кронштейна 9. Включають пристрій 11 для переміщення нитки. При цьому каретка 5 починає рухатись, а кронштейн 9 деформує пружину 22 та замикає контакт 21 електродвигуна пристрою 15 для переміщення направляючої 7.

Використання запропонованого пристрою дозволяє значно розширити діапазон вимірювальних величин шляхом вимірювання фрикційних властивостей ниток при зміні кута охоплення нитконапрямників, що дозволяє вдосконалити технологічні процеси та обладнання легкої та текстильної промисловості.

Для експериментального визначення сумісного впливу радіусу кривизни, швидкості, кута охоплення на натяг ведучої гілки нитки нами була розроблена установка, представлена на рис.5.3.4. Установка включає третій вимірювальний вузол 3. Це було зроблено для того, щоб можна було одночасно з експериментальним дослідженням умов взаємодії нитки з напрямною 6, встановленою нерухомо в корпусі 5, проводити експеримент за визначенням умов роботи різних нитконатягувачів, які встановлювалися між 2 і 3 вимірювальними вузлами.

Крім того, для варіювання швидкості руху нитки був встановлений ступінчастий привод 8, який забезпечує необхідну швидкість обертання приймальному барабану 9.

Зміна кута обхвату ниткою направляючої 6 здійснювалося за допомогою двох рухомих роликів 4, встановлених в корпусі 5 з можливістю горизонтального переміщення. Нитка на вимірювальний вузол 2 подавалася з циліндричної бобіни з певним попереднім натягом, який створювався шляхом пригальмовування бобіни.

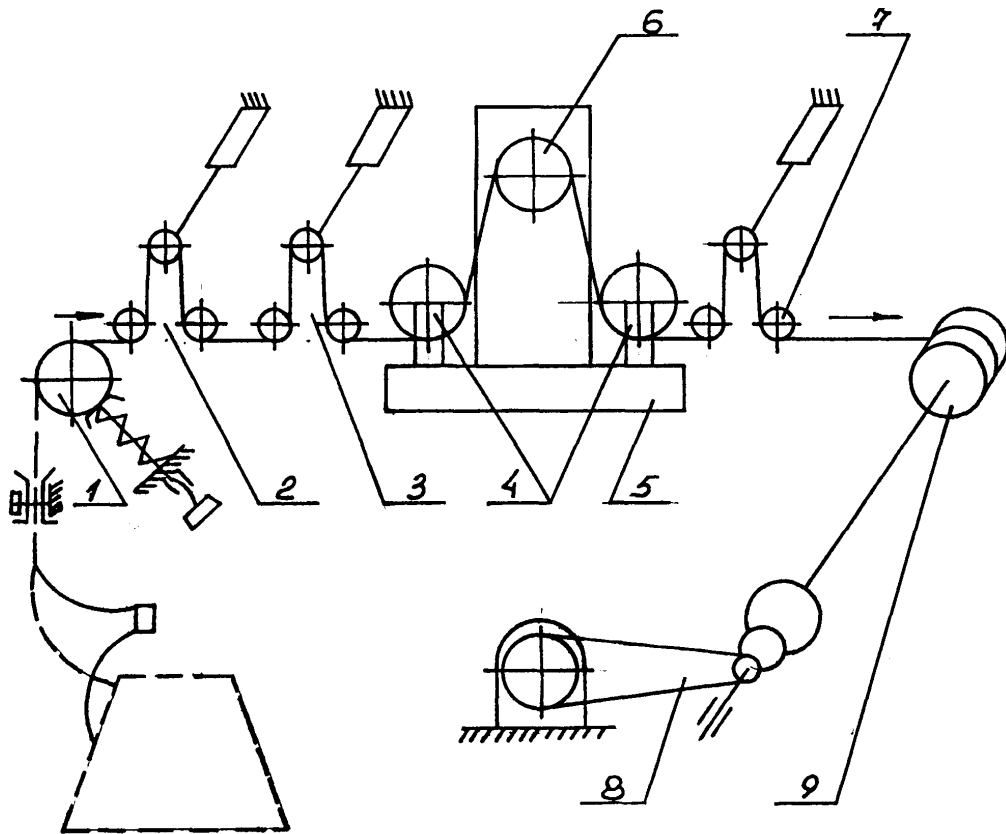


Рис.5.3.4. Електротензометрична установка

У приладі передбачалася подача нитки безпосередньо з конічної бобіни, для чого в його конструкцію були введені ниткоспрямовувачі і шайбовий нитконатягувач. Натяг веденої гілки нитки перед напрямною 6 реєструвався вимірювальним вузлом 3. Натяг ведучої гілки нитки - вимірювальним вузлом 7. У приладі була передбачена можливість заміни направляючої 6 з метою визначення впливу радіусу її кривизни на умови взаємодії з ниткою.

Описані вище установки дозволяли варіювати: швидкість руху нитки від 0,1 до 1,5 м/с; натягнення веденої гілки нитки від 5 до 15 сН; кут обхвату що направляє від 1,57 до 3,14 радіан; радіус кривизни направляючої поверхні від 1 до 7 мм.

Складовою частиною приладів була електротензометрична апаратура. Її вибір не був випадковим. Висока точність у вимірюванні величин,

можливість реєструвати швидкоплинні процеси, хороша наочність - ось далеко не повний перелік цих переваг. При проведенні експериментальних досліджень використовувався дванадцяти каналний осцилограф К12-22, підсилювач 8АНЧ-7М.

При виборі гальванометрів осцилографа визначальними чинниками були: частотна характеристика процесу, що вивчається, і вихідний струм з відповідного каналу підсилювача на осцилограф. Зразки запису осцилограм натягу ведучої гілки нитки приведені на рис.5.3.5. Для кожного варіанту з плану експерименту знімалися три повторні осцилограми.

Необхідно відзначити, що осцилограф К12-22, 8-ми каналний підсилювач 8АНЧ-7М і відповідні блоки живлення до них проходили спеціальну тестову перевірку на предмет своєї функціональної придатності і надійності в роботі.

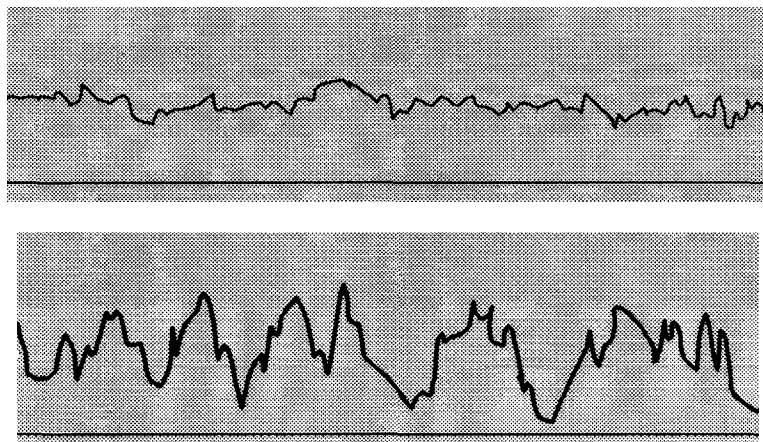


Рис.5.3.5. Зразки осцилограм

Обґрунтування вибору ниток і пряжі

Вивчення впливу матеріалу і структури ниток на величину натягу ведучої гілки призвело до вибору широкого асортименту ниток, як по структурі, так і по хімічному складу. По структурі текстильні нитки можна

класифікувати на комплексні нитки і пряжу. За походженням всі волокна діляться на натуральні і хімічні.

При виборі ниток нами враховувалися і геометричні розміри поперечного перетину (розрахунковий діаметр) . Найширше, при виробленні тканин, використовуються нитки, діаметр яких коливається в межах від 0,2 до 0,7 мм. У трикотажній промисловості, при виробництві полотен, різного асортименту верхнього і нижнього трикотажу, використовують нитки з розрахунковим діаметром від 0,1 до 0,25 мм.

Для експерименту були вибрані наступні нитки (таблиця 5.3.1):

Натуральна сировина:

- бавовняна пряжа натуральна, органічна, рослинна, целюлозна 30,8 Т;
- вовняна пряжа натуральна, органічна, тваринна, білкова 31 Т.

Хімічні волокна (таблиця 5.3.1):

- лавсанова комплексна нитка хімічна, органічна, синтетична, гетероланцюгова, поліефірна 33,2 Т текс;
- капронова комплексна нитка хімічна, органічна, синтетична, гетероланцюгова, поліамідна лінійної щільності 31,2 Т;
- віскозна комплексна нитка хімічна, органічна, штучна, гідратцелюлозна 29 Т.

На підставі набутих значень фактичного тексту визначався розрахунковий діаметр нитки за відомою методикою [4].

Отримані експериментальні дані оброблялися методами математичної статистики в наступній послідовності: визначалося середньоарифметичне значення величини натягу ведучої гілки, знаходили відхилення від середнього значення для кожного результату; визначали величину дисперсії; стандартне відхилення окремого вимірювання і стандартного відхилення середнього результату; перевірку надійності отриманого результату проводили по критерію Ст'юдента для проведеного числа дослідів $n=3$ при вибраній довірчій вірогідності $\alpha=0,95$; визначали

помилку отриманого середнього результату і визначали інтервал, в якому з довірчою вірогідністю α знаходиться середній результат; обчислювали відносну помилку.

Таблиця 5.3.1.

Фізико-механічні властивості ниток

№ п/п	Нитка	Характеристика	Структура	Фактичний текс, Т	Розрахунковий діаметр, мм	Коефіцієнт жорсткості на згин, Во,сН·мм ²
1	Бавовняна 15,4x2	натуральна, органічна, рослинна, целюлозна	Пряжа	30,8	0,22	0,7
2	Вовняна	натуральна, органічна, тваринна, білкова	Пряжа	31	0,24	0,1
3	Лавсанова	хімічна, органічна, синтетична, гетероланцюгова, поліефірна	Комплексна нитка	33,2	0,24	0,25
4	Капронова 15,6x2	хімічна, органічна, синтетична, гетероланцюгова поліамідна	Комплексна нитка	31,2	0,22	0,22
5	Віскозна	хімічна, органічна, штучна, гідратцелюлозна	Комплексна нитка	29	0,22	0,4

Плани проведення експерименту

Визначення основних чинників, які обумовлюють зміну натягу нитки, що рухається по направляючій поверхні, має значення при виборі стратегії проведення експерименту.

Наукова інформація, яка накопичилася до теперішнього часу, з даної проблеми дозволяє побудувати середню апріорну діаграму рангів. По одній осі відкладаються чинники, а по іншій суми рангів. До даних чинників були віднесені: радіус кривизни направляючої поверхні, натяг веденої гілки нитки, кут охоплення ниткою направляючої, швидкість руху нитки, температура навколишнього середовища, вологість повітря, коефіцієнт тертя між ниткою і напрямною, характер обробки напрямної поверхні, матеріал і структура нитки.

Аналіз середньої апріорної діаграми рангів дозволив встановити, що для радіусу кривизни направляючої, натягу веденої гілки, кута охоплення розподіл відповідних сум рангів рівномірний.

Таким чином, при проведенні експериментальних досліджень, нами було поставлено завдання по визначенню впливу радіуса кривизни напрямної поверхні R , натягу веденої гілки P_0 , кута охоплення φ на натяг ведучої гілки P нитки.

Після визначення чинників, що впливають на збільшення натягу ведучої гілки нитки, обґрунтування і вибору сировини для випробувань, необхідно скласти план проведення експерименту. Це дозволить виробити певну стратегію в пошуку оптимального поєднання чинників, що впливають на даний процес.

Для кожної з ниток даним планом (таблиця 5.3.2 і таблиця 5.3.3) передбачається 15 дослідів. По кожному з дослідів передбачалося 3 повторних.

Вибір розмірів направляючої поверхні при проведенні експерименту, в пропонованому діапазоні від 1 мм до 7 мм, пояснюється тим, що даний

інтервал охоплює до 80% всього асортименту голково-платинових виробів ниткоспрямовувачів, робочих органів текстильних і трикотажних машин. Одна серія експериментів проводилася для діапазону зміни радіусу кривизни направляючої від 1 до 3 мм. Друга серія експериментів проводилася для діапазону зміни радіусу кривизни направляючої від 3 до 7 мм. Це пояснюється тим, що при взаємодії ниток з направляючими, радіус кривизни яких є порівняним з радіусом поперечного перетину нитки починає позначатися мнучкість і жорсткість ниток на вигин[1,4].

Таблиця 5.3.2

План проведення експерименту ($R=1\dots 3$ мм)

№ варіанта	Досліджувані чинники						Характерні зони матриці
	x1	R	x2	Po	x3	φ	
i-1- 1	-1	R1i+h1	+1	Poi+ h2	+1	φ i+ h3	Точки ядра
i-1- 2	+1	R1i-h1	-1	Poi+ h2	+1	φ i+ h3	
i-1- 3	-1	R1i+h1	+1	Poi- h2	-1	φ i+ h3	
i-1- 4	+1	R1i-h1	+1	Poi- h2	+1	φ i+ h3	
i-1- 5	+1	R1i+h1	+1	Poi+ h2	-1	φ i- h3	
i-1- 6	+1	R1i-h1	-1	Poi+ h2	-1	φ i- h3	
i-1- 7	-1	R1i+h1	-1	Poi- h2	-1	φ i- h3	
i-1- 8	-1	R1i-h1	-1	Poi- h2	+1	φ i- h3	
i-1- 9	-1,215	R1i-*	0	Poi	0	φ i	Зоряні точки
i-1- 10	+1,215	R1i+*	0	Poi	0	φ i	
i-1- 11	0	R1i	-1,215	Poi-*	0	φ i	
i-1- 12	0	R1i	+1,215	Poi+*	0	φ i	
i-1- 13	0	R1i	0	Poi	-1,215	φ i-*	
i-1- 14	0	R1i	0	Poi	+1,215	φ i+*	
i-1- 15	0	R1i	0	Poi	0	φ i	Нульова точка

Інтервали зміни величини натягу веденої гілки нитки (від 5 до 15 сН) були вибрані виходячи з умов переробки ниток на ткацьких верстатах і в'язальних машинах, підготовчому устаткуванні.

Кут охоплення направляючої поверхні, для кожного конкретного випадку, є величиною змінною. Зважаючи на специфіку переробки ниток на трикотажному устаткуванні, де мають місце великі перегини ниток при взаємодії з робочими органами, нами був вибраний інтервал зміни від 90° до 180° .

У таблиці 5.3.3 приведений план проведення експерименту по визначенню впливу радіусу кривизни направляючої поверхні R , натягу веденої гілки P_o , кута охоплення φ на натяг ведучої гілки P нитки при діапазоні зміни радіусу кривизни напрямної від 3 до 7 мм.

Таблиця 5.3.3

План проведення експерименту ($R=3\dots7$ мм)

№ варіанта	Досліджувані чинники						Характерні зони матриці
	x1	R	x2	Po	x3	φ	
i-2- 1	-1	R2i+h1	+1	Poi+ h2	+1	φ i+ h3	Точки ядра
i-2- 2	+1	R2i-h1	-1	Poi+ h2	+1	φ i+ h3	
i-2- 3	-1	R2i+h1	+1	Poi- h2	-1	φ i+ h3	
i-2- 4	+1	R2i-h1	+1	Poi- h2	+1	φ i+ h3	
i-2- 5	+1	R2i+h1	+1	Poi+ h2	-1	φ i- h3	
i-2- 6	+1	R2i-h1	-1	Poi+ h2	-1	φ i- h3	
i-2- 7	-1	R2i+h1	-1	Poi- h2	-1	φ i- h3	
i-2- 8	-1	R2i-h1	-1	Poi- h2	+1	φ i- h3	
i-2- 9	-1,215	R2i-*	0	Poi	0	φ i	Зоряні точки
i-2- 10	+1,215	R2i+*	0	Poi	0	φ i	
i-2- 11	0	R2i	-1,215	Poi-*	0	φ i	
i-2- 12	0	R2i	+1,215	Poi+*	0	φ i	

i-2- 13	0	R2i	0	Poi	-1,215	ϕ_i^*	
i-2- 14	0	R2i	0	Poi	+1,215	ϕ_i^{+*}	
i-2- 15	0	R2i	0	Poi	0	ϕ_i	Нульова точка

При плануванні експерименту, при побудові моделей другого порядку нами було використано ортогональне планування другого порядку. Хоча існують інші оптимальні плани, ортогональне планування використовувалося через свою достатню простоту і можливості застосування вже розроблених пакетів програм для ЕОМ з обробки отриманих експериментальних даних.

Спільний вплив радіусу кривизни направляючої, натягу веденої гілки, кута охоплення на натяг ведучої гілки для направляючої великої кривизни

Реалізуючи план експериментальних досліджень (табл.5.3.2) за визначенням впливу радіусу кривизни направляючої поверхні, натягу веденої гілки нитки, кута охоплення на натяг ведучої гілки, для різних ниток (табл.5.3.1), нами було отримано ряд даних, представлених в таблиці 5.3.5.3.

Вище ми зупинялися на тих міркуваннях, які привели до вибору ортогонального плану. Останні вимоги застосування того або іншого плану, що пред'являються до оцінки ефективності, привели до того, що основними критеріями необхідно вважати: максимальну величину дисперсії передбачених значень критерію оптимізації в області експерименту; об'єм еліпсоїда розсіяння оцінок параметрів та ін.

При проведенні експерименту іменовані величини необхідно замінити на кодовані. В нашому випадку радіус кривизни напрямної R позначимо через $x1$. В нульовій точці радіус кривизни дорівнював 2 мм. Довжина плеча складала 1 мм. Натяг веденої гілки нитки Po позначимо через $x2$. В

нульовій точці натяг веденої гілки нитки дорівнює 7,5 сН. Довжина плеча складала 2,5 сН. Кут охоплення ниткою напрямної φ позначимо через x_3 . В нульовій точці кут охоплення ниткою напрямної дорівнює $3\pi/4$ рад. Довжина плеча складала $\pi/4$ рад.

Тоді система рівнянь для переходу від іменованих до кодованих значень чинників буде мати вигляд

$$x_1 = R - 2, \quad x_2 = \frac{P_0 - 7,5}{2,5}, \quad x_3 = \frac{\varphi - \frac{3\pi}{4}}{\frac{\pi}{4}}. \quad (5.3.1)$$

Таблиця 5.3.4

Спільний вплив радіусу кривизни направляючої, натягу веденої гілки, кута охоплення на натяг ведучої гілки для направляючої великої кривизни

Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН
1-1- 1	16,00	2-1- 1	13,90	3-1- 1	14,16	4-1- 1	15,74	5-1- 1	15,00
1-1- 2	7,90	2-1- 2	6,86	3-1- 2	6,99	4-1- 2	7,77	5-1- 2	7,20
1-1- 3	14,30	2-1- 3	11,64	3-1- 3	12,00	4-1- 3	14,01	5-1- 3	12,17
1-1- 4	16,80	2-1- 4	14,59	3-1- 4	14,87	4-1- 4	16,52	5-1- 4	15,70
1-1- 5	15,00	2-1- 5	13,55	3-1- 5	12,85	4-1- 5	14,61	5-1- 5	13,06
1-1- 6	8,00	2-1- 6	6,95	3-1- 6	7,08	4-1- 6	7,87	5-1- 6	7,22
1-1- 7	11,00	2-1- 7	9,56	3-1- 7	11,00	4-1- 7	10,82	5-1- 7	9,92
1-1- 8	9,50	2-1- 8	8,25	3-1- 8	8,41	4-1- 8	9,34	5-1- 8	8,57
1-1- 9	14,00	2-1- 9	12,16	3-1- 9	12,39	4-1- 9	13,77	5-1- 9	12,62
1-1- 10	13,00	2-1- 10	11,29	3-1- 10	11,51	4-1- 10	12,79	5-1- 10	11,72

1-1-11	8,00	2-1-11	6,95	3-1-11	7,08	4-1-11	7,87	5-1-11	7,21
1-1-12	16,00	2-1-12	13,90	3-1-12	14,16	4-1-12	15,74	5-1-12	14,23
1-1-13	10,00	2-1-13	8,69	3-1-13	8,85	4-1-13	9,84	5-1-13	9,02
1-1-14	12,20	2-1-14	10,60	3-1-14	10,70	4-1-14	12,10	5-1-14	11,1
1-1-15	12,20	2-1-15	10,60	3-1-15	10,80	4-1-15	12,00	5-1-15	11,00

На рис.5.3.6 представлена поверхня відгуку для бавовняної пряжі.

Рівняння регресії для бавовняної пряжі буде мати вигляд

$$P = 12,17 - 0,36x_1 + 3,16x_2 + 0,45x_3 + 0,82x_1^2 + 0,8x_1x_2 + 0,15x_1x_3 - 0,09x_2^2 + 0,68x_2x_3 - 0,63x_3^2. \quad (5.3.2)$$

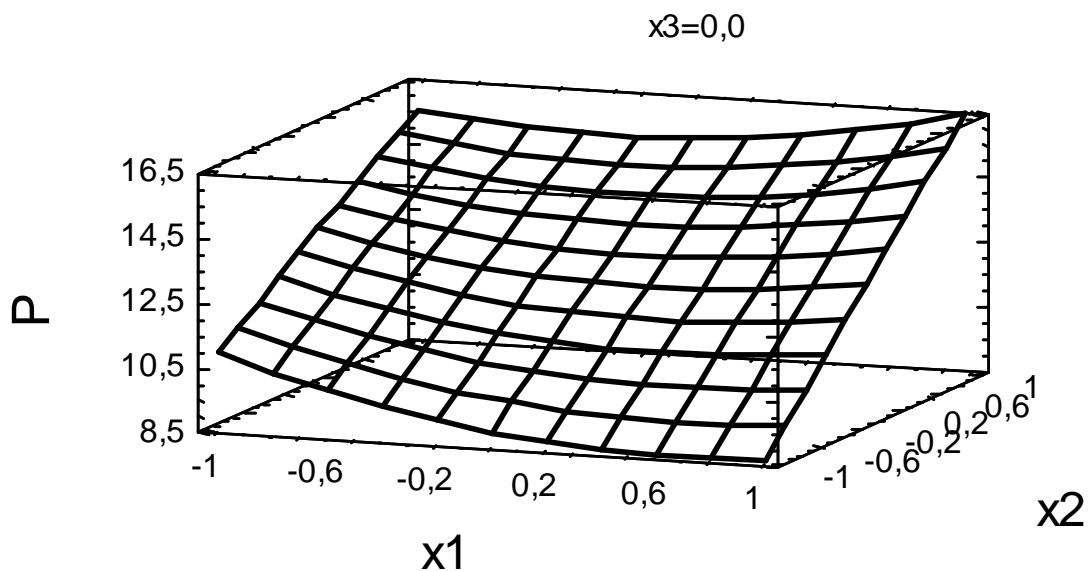


Рис.5.3.6. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.2) отримаємо відповідну діаграму, яка представлена на рис.5.3.7. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2^2 .

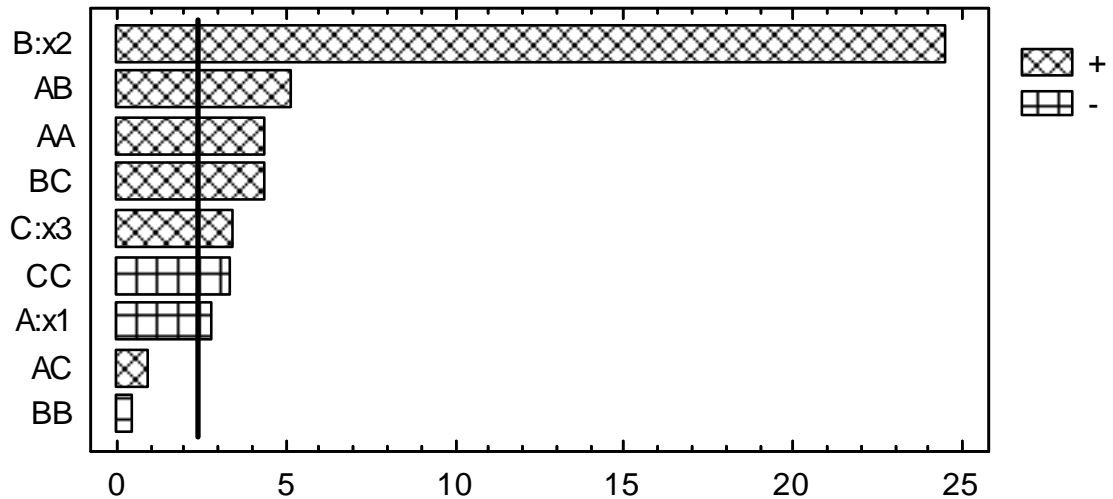


Рис.5.3.7. Діаграма значущості коефіцієнтів в рівнянні регресії
З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.2), а також з урахуванням системи (5.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P = 8,94 - 0,376P_0 - 12,04R + 5,60\varphi + 1,63R^2 - 2,044\varphi^2 + 0,64P_0R + 0,687P_0\varphi. \quad (5.3.3)$$

На рис.5.3.8 представлена поверхня відгуку для вовняної пряжі. Рівняння регресії для вовняної пряжі буде мати вигляд

$$P = 10,50 - 0,35x_1 + 2,61x_2 + 0,26x_3 + 0,78x_1^2 + 1,01x_1x_2 + 0,18x_1x_3 - 0,0008x_2^2 + 0,77x_2x_3 - 0,47x_3^2. \quad (5.3.4)$$

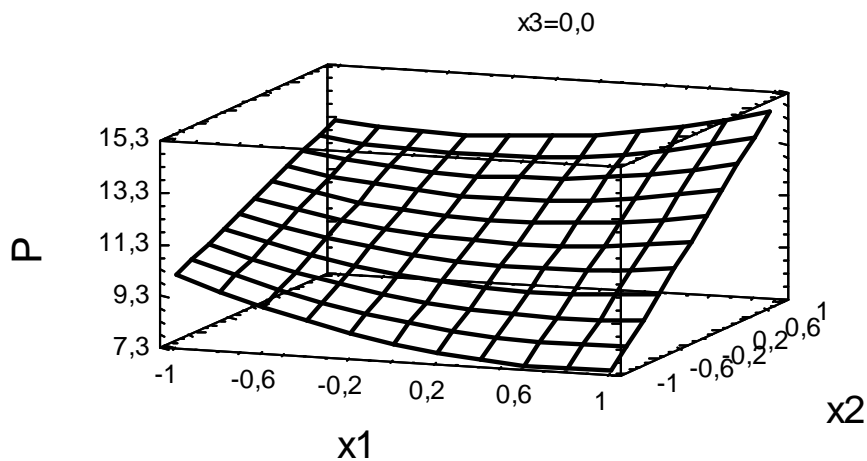


Рис.5.3.8. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.4) отримаємо відповідну діаграму, яка представлена на рис.5.3.9. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1 , x_3 , x_1x_3 , x_2^2 , x_3^2 .

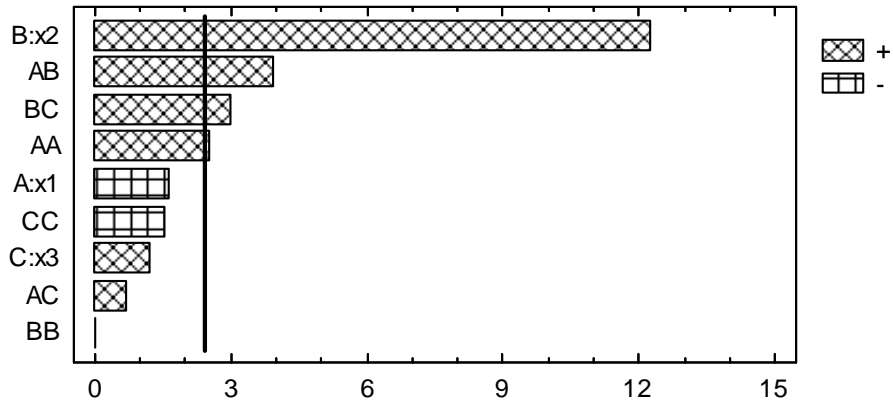


Рис.5.3.9. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.4), а також з урахуванням системи (5.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P = 27,01 - 1,364P_0 - 12,31R - 5,88\varphi + 0,804P_0R + 0,784P_0\varphi + 1,57R^2. \quad (5.3.5)$$

На рис.5.3.10 представлена поверхня відгуку для лавсанової комплексної нитки. Рівняння регресії для лавсанової комплексної нитки буде мати вигляд

$$P = 10,74 - 0,43x_1 + 2,61x_2 + 0,34x_3 + 0,76x_1^2 + 0,86x_1x_2 + 0,29x_1x_3 - 0,043x_2^2 + 0,86x_2x_3 - 0,55x_3^2. \quad (5.3.6)$$

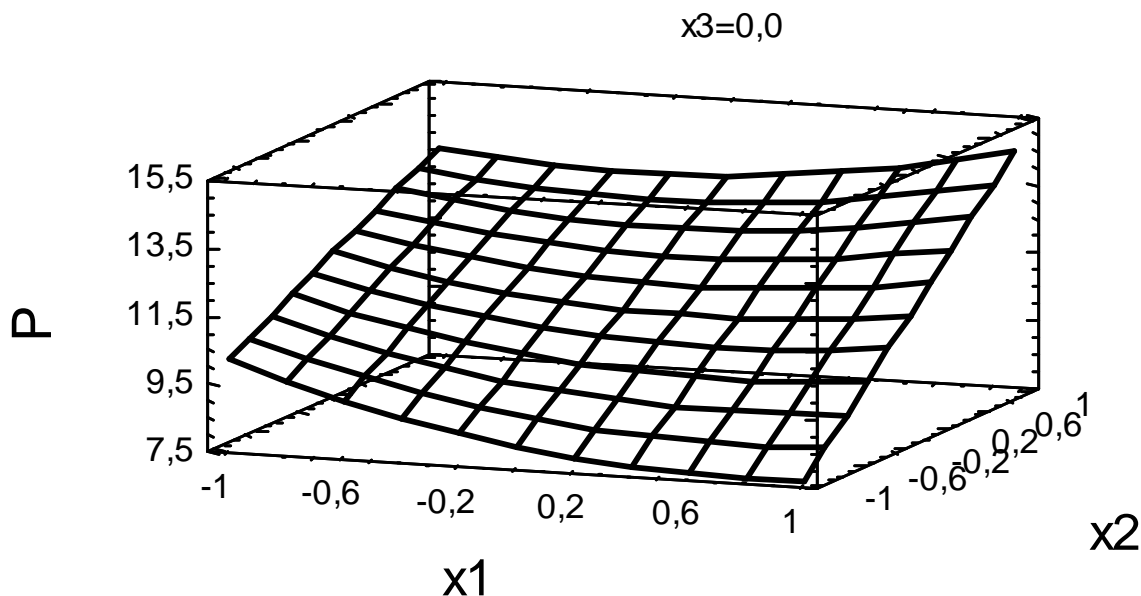


Рис.5.3.10. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.6) отримуємо відповідну діаграму, яка представлена на рис.5.3.11. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_3 , x_1x_3 , x_2^2 .

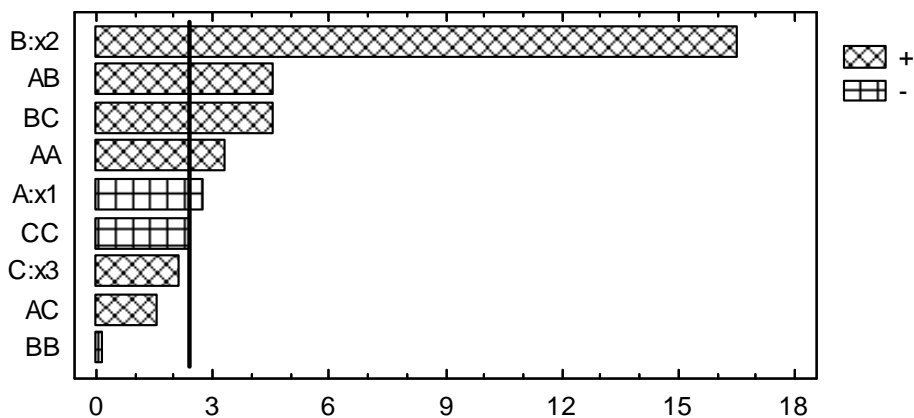


Рис.5.3.11. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.6), а також з урахуванням системи (5.3.1), перейдемо до іменованих значень, остаточно отримуємо

$$P = 18,77 - 1,36P_0 - 12,14R + 1,91\varphi + 1,52R^2 - 1,801\varphi^2 + 0,692P_0R + 0,876P_0\varphi. \quad (5.3.7)$$

На рис.5.3.12 представлена поверхня відгуку для капронової комплексної

нитки. Рівняння регресії для капронової комплексної нитки буде мати вигляд

$$P = 11,98 - 0,39x_1 + 3,12x_2 + 0,45x_3 + 0,79x_1^2 + 0,75x_1x_2 + 0,21x_1x_3 - 0,102x_2^2 + 0,64x_2x_3 - 0,61x_3^2. \quad (5.3.8)$$

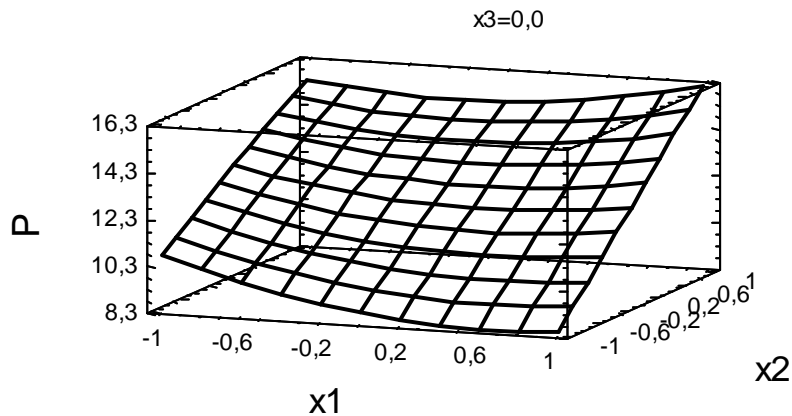


Рис.5.3.12. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.8) отримаємо відповідну діаграму, яка представлена на рис.5.3.13. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2^2 .

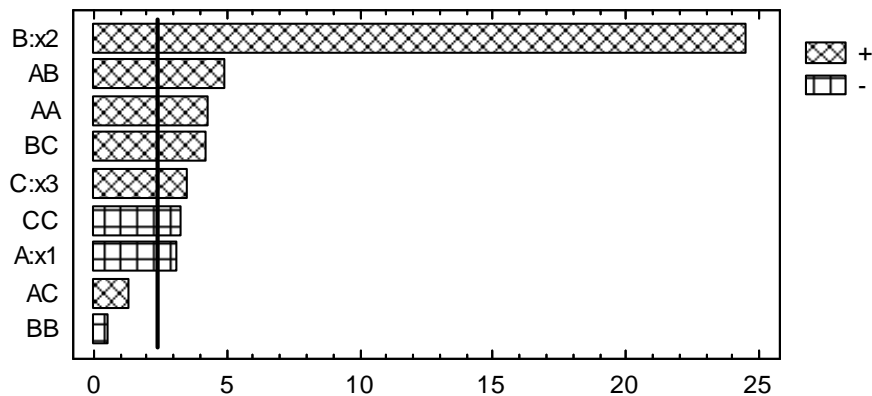


Рис.5.3.13. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.8), а також з урахуванням системи (5.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P = 8,13 - 0,24P_0 - 11,61R + 5,49\varphi + 1,58R^2 - 1,963\varphi^2 + 0,6P_0R + 0,652P_0\varphi. \quad (5.3.9)$$

На рис.5.3.14 представлена поверхня відгуку для віскозної комплексної нитки. Рівняння регресії для віскозної комплексної нитки буде мати вигляд

$$P = 10,98 - 0,38x_1 + 2,78x_2 + 0,54x_3 + 0,73x_1^2 + 0,63x_1x_2 + 0,064x_1x_3 - 0,145x_2^2 + 0,78x_2x_3 - 0,54x_3^2. \quad (5.3.10)$$

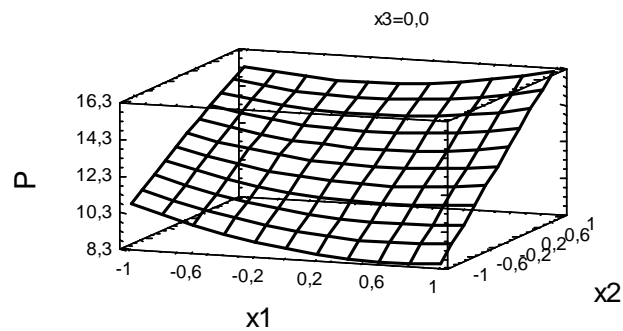


Рис.5.3.15.3. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.10) отримаємо відповідну діаграму, яка представлена на рис.5.3.15. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2^2 .

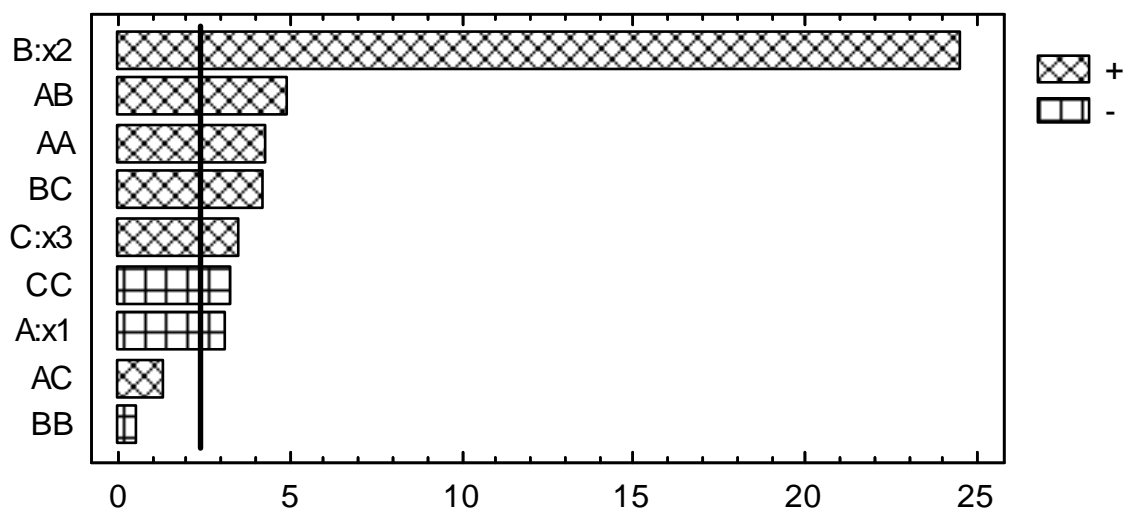


Рис.5.3.15. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.10), а також з урахуванням системи (5.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P = 10,09 - 0,644P_0 - 10,38R + 3,79\varphi + 1,46R^2 - 1,768\varphi^2 + 0,504P_0R + 0,789P_0\varphi. \quad (5.3.11)$$

Розрахунки по формулам (5.3.3), (5.3.5), (5.3.7), (5.3.9), (5.3.11) дозволили отримати масив даних (дивись додаток А), використовуючи який були побудовані графічні залежності що представлені на рис.5.3.16-5.3.18.

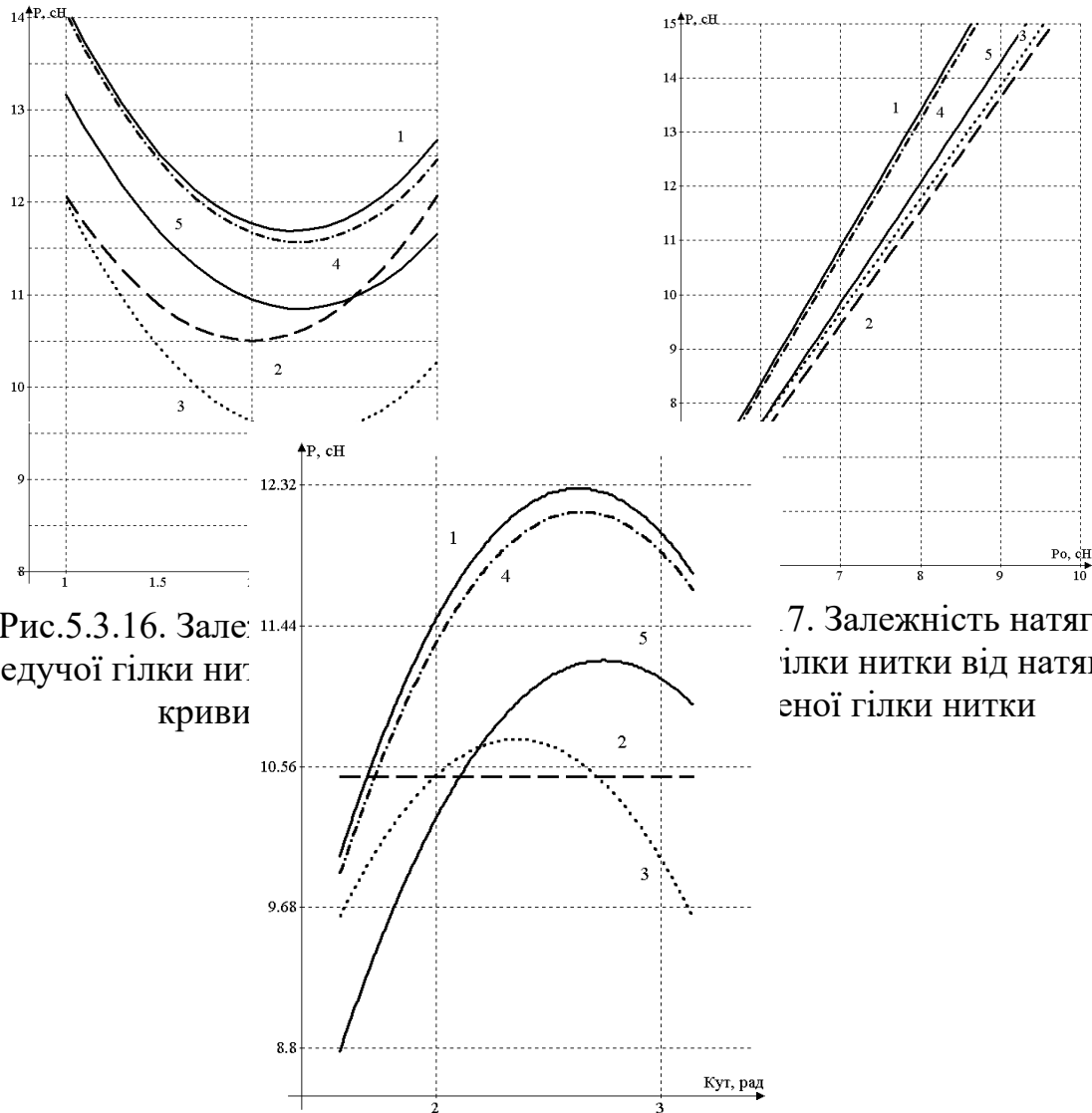


Рис.5.3.16. Залежність натягу ведучої гілки нитки від натягу еної гілки нитки

7. Залежність натягу ілки нитки від натягу еної гілки нитки

Рис.5.3.18. Залежність натягу ведучої гілки нитки від кута охоплення

Аналіз графічних залежностей на рис.5.3.16-5.3.18 показав, що при збільшенні натягу веденої гілки нитки натяг ведучої гілки зростає пропорційно. При збільшенні радіуса кривизни напрямної (рис.5.3.16) натяг спочатку зменшується, а далі починає зростати. Це пояснюється впливом зминання нитки в зоні контакту та її жорсткістю на вигин. При збільшенні кута охоплення (рис.5.3.18) простежується зворотна картина. Натяг спочатку збільшується, а далі починає спадати. Це пояснюється зміною величини коефіцієнта тертя [12, 12]. Ці результати необхідно враховувати при проектуванні технологічних процесів легкої та текстильної промисловості.

Спільний вплив радіусу кривизни направляючої, натягу веденої гілки, кута охоплення на натяг ведучої гілки для направляючих малої кривизни

Реалізуючи план експериментальних досліджень (табл.5.3.3) за визначенням впливу радіусу кривизни направляючої поверхні, натягу веденої гілки нитки, кута охоплення на натяг ведучої гілки, для різних ниток (табл.5.3.1), нами було отримано ряд даних, представлених в таблиці 5.3.5.

При проведенні експерименту іменовані величини необхідно замінити на кодовані. В нашому випадку радіус кривизни напрямної R позначимо через $x1$. В нульовій точці радіус кривизни дорівнював 5 мм. Довжина плеча складала 2 мм. Натяг веденої гілки нитки Po позначимо через $x2$. В нульовій точці натяг веденої гілки нитки дорівнює 10 сН. Довжина плеча складала 5 сН. Кут охоплення ниткою напрямної φ позначимо через $x3$. В нульовій точці кут охоплення ниткою напрямної дорівнює $3\pi/4$ рад. Довжина плеча складала $\pi/4$ рад.

Тоді система рівнянь для переходу від іменованих до кодованих значень

чинників буде мати вигляд

$$x_1 = \frac{R-5}{2}, \quad x_2 = \frac{P_0-10}{5}, \quad x_3 = \frac{\varphi - \frac{3\pi}{4}}{\frac{\pi}{4}}. \quad (5.3.12)$$

Таблиця 5.3.5

Спільний вплив радіусу кривизни направляючої, натягу веденої гілки, кута охоплення на натяг ведучої гілки для направляючих малої кривизни

Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН	Варіант	Натяг, сН
1-2- 1	18,34	2-2- 1	18,34	3-2- 1	17,89	4-2- 1	18,45	5-2- 1	18,11
1-2- 2	16,11	2-2- 2	9,52	3-2- 2	10,11	4-2- 2	9,35	5-2- 2	9,99
1-2- 3	18,32	2-2- 3	16,11	3-2- 3	15,91	4-2- 3	15,92	5-2- 3	16,14
1-2- 4	24,50	2-2- 4	22,20	3-2- 4	22,40	4-2- 4	24,70	5-2- 4	23,00
1-2- 5	20,11	2-2- 5	16,53	3-2- 5	18,44	4-2- 5	19,11	5-2- 5	19,22
1-2- 6	12,08	2-2- 6	6,47	3-2- 6	6,58	4-2- 6	7,43	5-2- 6	7,77
1-2- 7	7,12	2-2- 7	5,89	3-2- 7	5,37	4-2- 7	5,99	5-2- 7	5,83
1-2- 8	9,14	2-2- 8	8,12	3-2- 8	7,24	4-2- 8	7,08	5-2- 8	6,95
1-2- 9	12,61	2-2- 9	12,49	3-2- 9	11,67	4-2- 9	10,44	5-2- 9	12,87
1-2- 10	14,55	2-2- 10	16,07	3-2- 10	14,82	4-2- 10	13,88	5-2- 10	17,69
1-2- 11	10,77	2-2- 11	9,33	3-2- 11	9,53	4-2- 11	9,19	5-2- 11	8,12
1-2- 12	17,38	2-2- 12	19,10	3-2- 12	18,12	4-2- 12	19,41	5-2- 12	18,41
1-2- 13	10,60	2-2- 13	12,25	3-2- 13	12,03	4-2- 13	11,12	5-2- 13	13,57
1-2- 14	21,10	2-2- 14	16,22	3-2- 14	17,55	4-2- 14	15,40	5-2- 14	17,88
1-2- 15	16,80	2-2- 15	15,00	3-2- 15	15,20	4-2- 15	16,40	5-2- 15	15,40

На рис.5.3.19 представлена поверхня відгуку для бавовняної пряжі.

Рівняння регресії для бавовняної пряжі буде мати вигляд

$$P = 15,55 + 1,98x_1 + 4,01x_2 + 2,12x_3 - 0,63x_1^2 - 0,49x_1x_2 + 0,79x_1x_3 - 0,33x_2^2 - 0,205x_2x_3 + 0,74x_3^2. \quad (5.3.13)$$

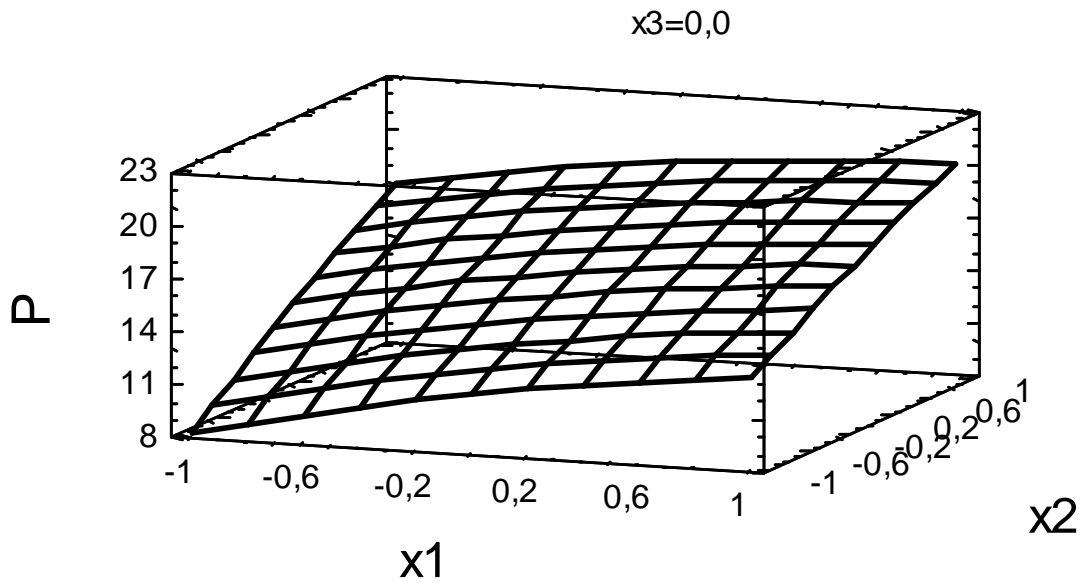


Рис.5.3.19. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.10) отримаємо відповідну діаграму, яка представлена на рис.5.3.20. Аналіз даної діаграми показує, що значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1 , x_2 , x_3 .

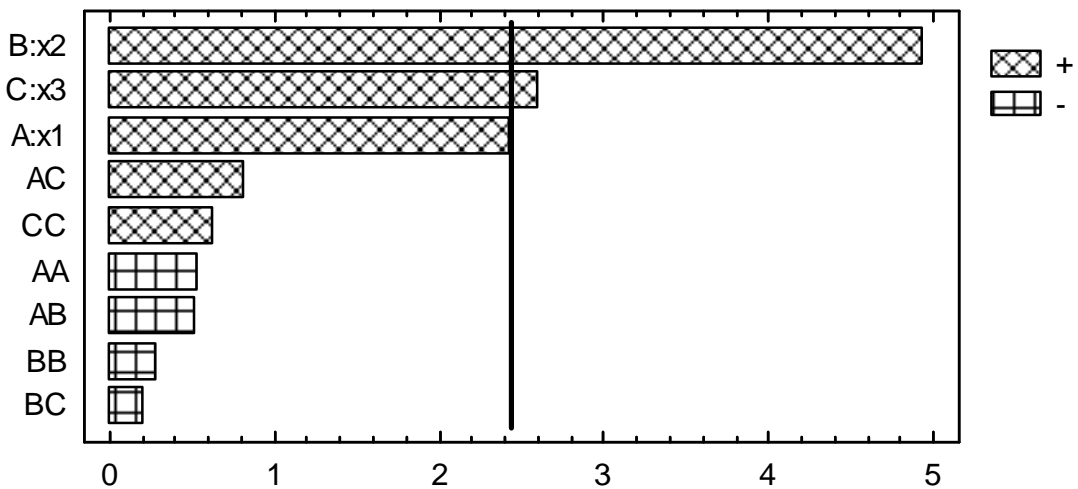


Рис.5.3.20. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.13), а також з урахуванням системи (5.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P = 1,602P_0 + 1,98R + 5,4\varphi - 23,09. \quad (5.3.14)$$

На рис.5.3.21 представлена поверхня відгуку для вовняної пряжі. Рівняння регресії для вовняної пряжі буде мати вигляд

$$P = 15,29 + 0,96x_1 + 4,93x_2 + 1,62x_3 - 0,75x_1^2 + 0,29x_1x_2 + 0,53x_1x_3 - 0,79x_2^2 + 0,33x_2x_3 - 0,77x_3^2. \quad (5.3.15)$$

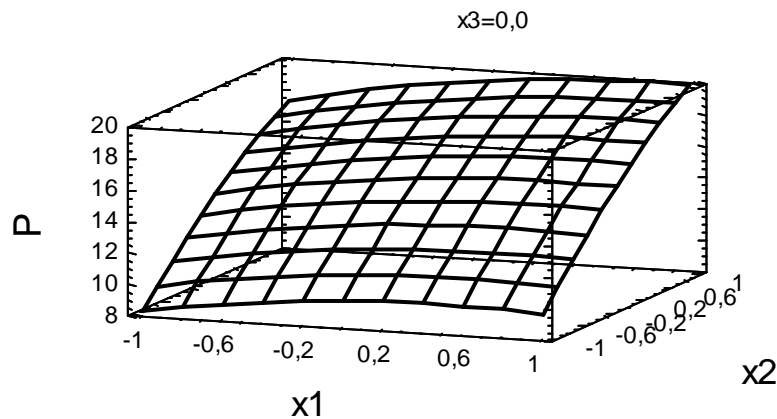


Рис.5.3.21. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.15) отримаємо відповідну діаграму, яка представлена на рис.5.3.22. Аналіз даної діаграми показує, що значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1 , x_2 , x_3 .

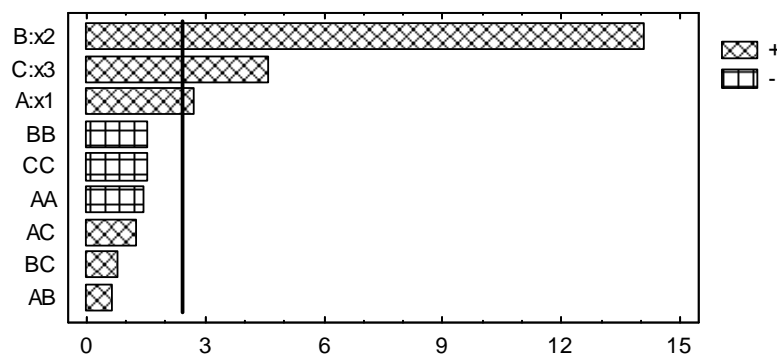


Рис.5.3.22. Діаграма значущості коефіцієнтів в рівнянні регресії з урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.15), а також з урахуванням системи (5.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P = 1,97P_0 + 0,96R + 4,12\varphi - 18,92. \quad (5.3.16)$$

На рис.5.3.23 представлена поверхня відгуку для лавсанової комплексної нитки. Рівняння регресії для лавсанової комплексної нитки буде мати вигляд

$$P = 15,18 + 1,34x_1 + 4,99x_2 + 1,63x_3 - 1,16x_1^2 + 0,37x_1x_2 + 0,46x_1x_3 - 0,81x_2^2 + 0,07x_2x_3 - 0,23x_3^2. \quad (5.3.17)$$

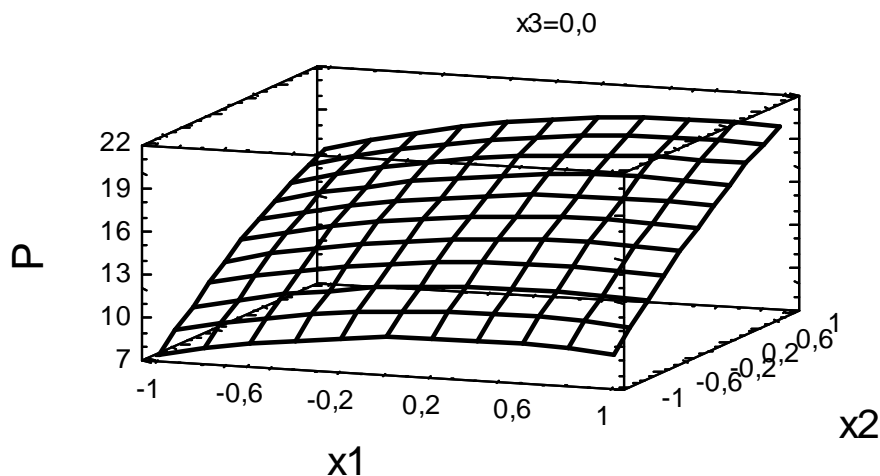


Рис.5.3.23. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.17) отримаємо відповідну діаграму, яка представлена на рис.5.3.25.3. Аналіз даної діаграми показує, що значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1, x_2, x_3 .

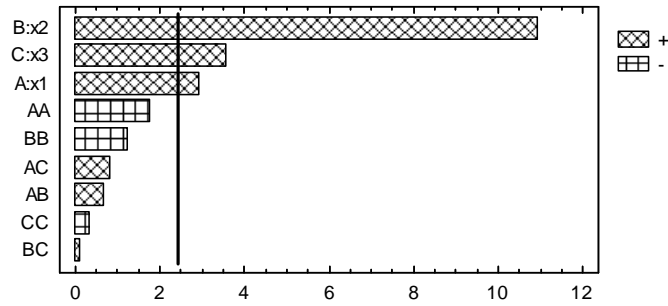


Рис.5.3.25.3. Діаграма значущості коефіцієнтів в рівнянні регресії з урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.17), а також з урахуванням системи (5.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P = 1,99P_0 + 1,34R + 4,15\varphi - 21,24. \quad (5.3.18)$$

На рис.5.3.25 представлена поверхня відгуку для капронової комплексної нитки. Рівняння регресії для капронової комплексної нитки буде мати вигляд

$$P = 15,26 + 1,55x_1 + 5,44x_2 + 1,47x_3 - 1,35x_1^2 + 0,72x_1x_2 + 0,49x_1x_3 - 0,06x_2^2 + 0,64x_2x_3 - 0,69x_3^2. \quad (5.3.19)$$

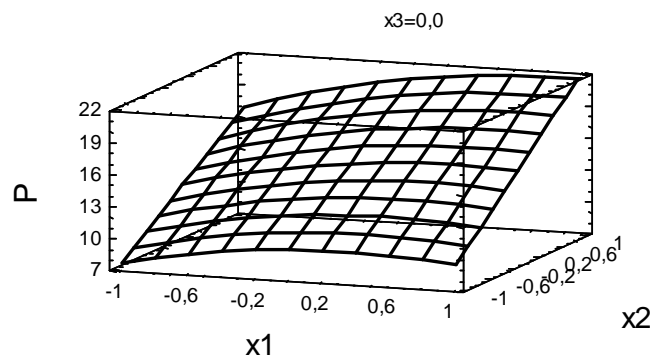


Рис.5.3.25. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.19) отримаємо відповідну діаграму, яка представлена на рис.5.3.26. Аналіз даної діаграми показує, що значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1, x_2, x_3 .

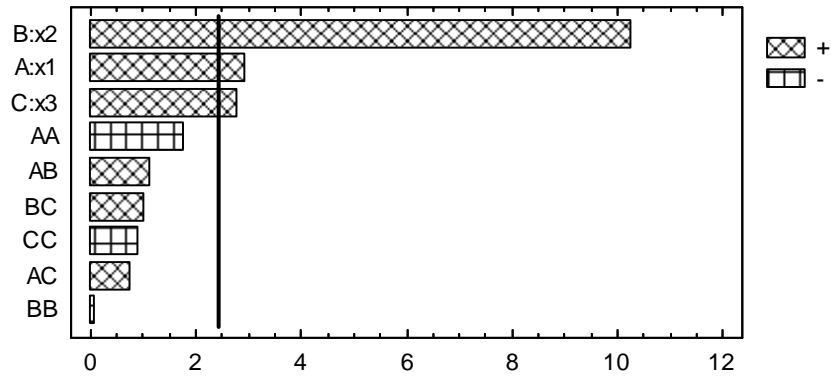


Рис.5.3.26. Діаграма значущості коефіцієнтів в рівнянні регресії
 З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.19), а також з урахуванням системи (5.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P = 2,17P_0 + 1,56R + 3,75\varphi - 23,08. \quad (5.3.20)$$

На рис.5.3.27 представлена поверхня відгуку для віскозної комплексної нитки. Рівняння регресії для віскозної комплексної нитки буде мати вигляд

$$P = 15,75 + 1,69x_1 + 5,23x_2 + 1,29x_3 - 0,44x_1^2 + 0,37x_1x_2 + 0,36x_1x_3 - 1,66x_2^2 + 0,30x_2x_3 - 0,17x_3^2. \quad (5.3.21)$$

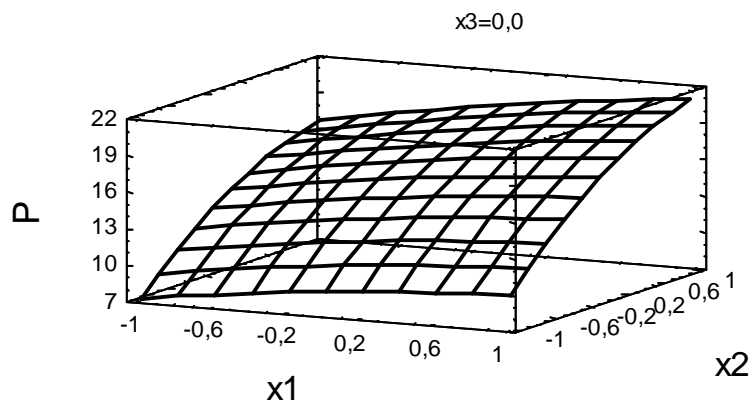


Рис.5.3.27. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (5.3.21) отримаємо відповідну діаграму, яка представлена на рис.5.3.28. Аналіз даної діаграми показує, що значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1, x_2, x_3, x_2^2 .

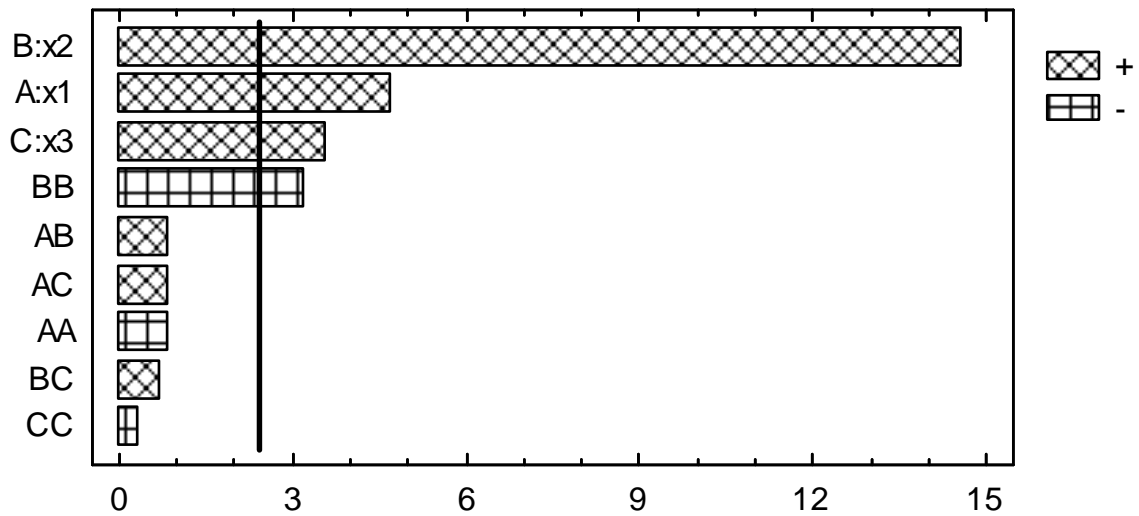


Рис.5.3.28. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (5.3.21), а також з урахуванням системи (5.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P = 4,74P_0 + 1,69R + 3,29\varphi - 0,132P_0^2 - 34,66. \quad (5.3.22)$$

Розрахунки по формулам (5.3.14), (5.3.16), (5.3.18), (5.3.20), (5.3.22) дозволили отримати масив даних (дивись додаток А), використовуючи який були побудовані графічні залежності представлені на рис.5.3.29-5.3.31.

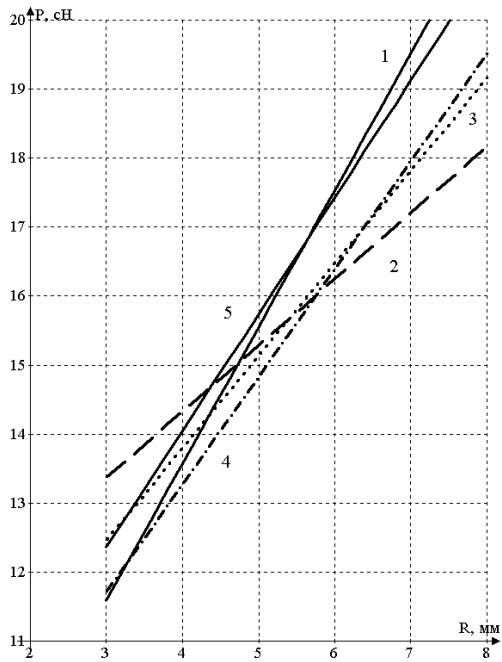


Рис.5.3.29. Залежність натягу ведучої гілки нитки від радіусу кривизни

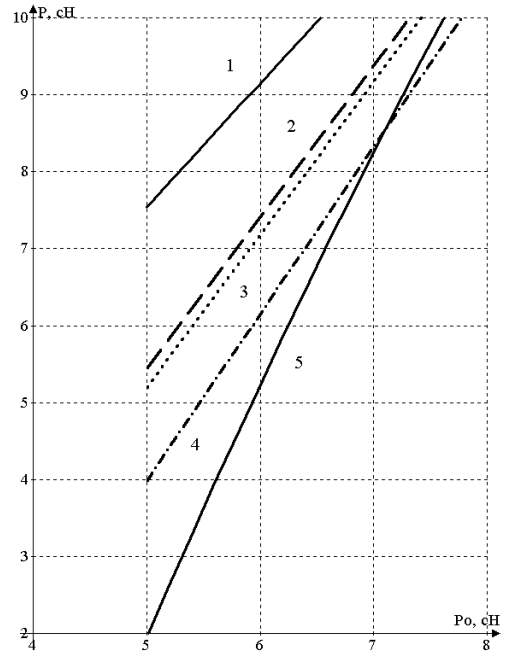


Рис.5.3.30. Залежність натягу ведучої гілки нитки від натягу веденої гілки нитки

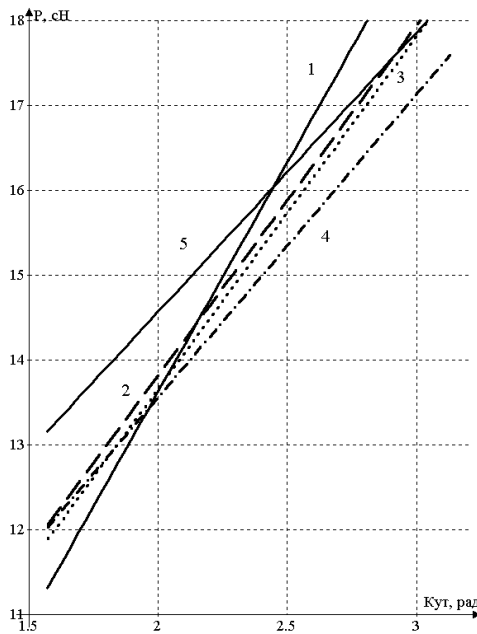


Рис.5.3.31. Залежність натягу ведучої гілки нитки від кута охоплення

Аналіз графічних залежностей на рис.5.3.29-5.3.31 показав, що при збільшенні радіусу кривизни, натягу веденої гілки нитки, кута охоплення

натяг ведучої гілки зростає пропорційно.

5.4. УДОСКОНАЛЕННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ТЕКСТИЛЬНОГО ВИРОБНИЦТВА НА ОСНОВІ МІНІМІЗАЦІЇ НАТЯГУ ПРУЖНОЇ СИСТЕМИ ЗАПРАВКИ

Збільшення випуску продукції підприємствами легкої і текстильної промисловості України повинне проводитися на основі їх технічного переозброєння і реконструкції, модернізації устаткування, впровадження нових високопродуктивних технологічних процесів і засобів комплексної механізації.

Підвищення продуктивності технологічного устаткування в легкій і текстильній промисловості нерозривно пов'язане із зменшенням часу простоїв на ліквідацію обриву ниток. Обриви ниток обумовлені низькою якістю початкової сировини, недосконалістю окремих вузлів і механізмів, системи ниткоподачі і форми ниткотракту.

Обриви ниток (так звані технологічні відмови) складають до 30% від сумарного часу відновлення всіх відмов технологічного устаткування. Число технологічних відмов унаслідок обривності складає до 80% від їх загального числа.

Таким чином, мінімізація натягу пружної системи заправки технологічного обладнання є актуальною темою, яка має велике значення для удосконалення як технологічних процесів так і обладнання.

Удосконалення системи ниткоподачі ткацьких верстатів

Схема пружної заправки ткацького верстата наведена на рис.5.4.1. Нитка, сходячи з навою, охоплює скало D, проходить ламелі C та, проходячи крізь очка A галев ремізних рамок, надходить в зону B формування тканини.

Після формування тканини вона, проходячи крізь вальєн, намотується на товарний валик. В процесі ткацтва, за рахунок вертикального руху очок А галев ремізних рамок, створюється зів, в який прокладається уточна нитка.

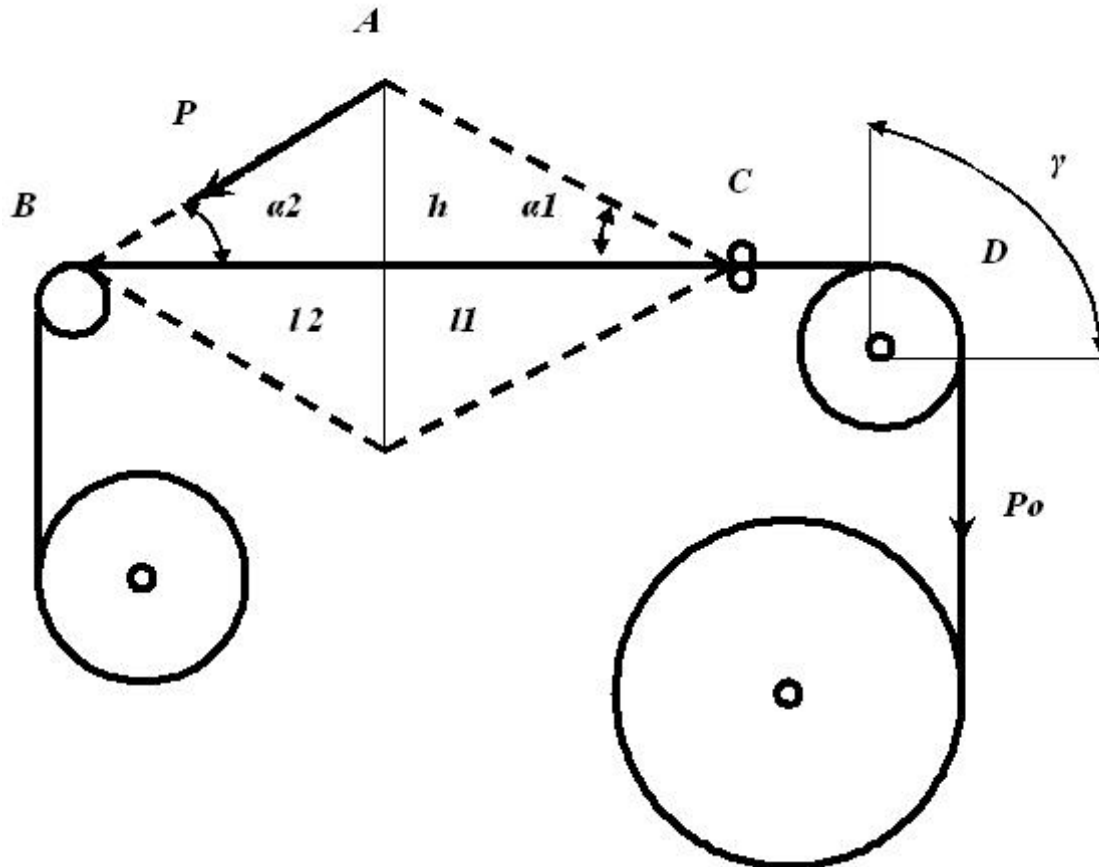


Рис.5.4.1. Схема пружної заправки ткацького верстата

Нитка має попередній натяг P_0 . Огинаючи скало D під кутом γ вона поступає в зевоутворюючий пристрій, де величина натягу зростає до P . Знайдемо закон зміни натягу при цьому. На ділянці DC натяг будемо визначати за формулою [4]

$$P_1 = P_0 e^{\mu_1 \gamma}, \quad (5.4.1)$$

де P_1 - натяг нитки після скала D;

μ_1 - коефіцієнт тертя нитки по поверхні скала.

В подальшому зростання натягу буде здійснюватися за рахунок розтягання нитки при зевоутворюванні та відносному ковзанні в очках галев ремізних рамок.

Визначимо поточну величину натягу при зевотворюванні. Очевидно, що це зростання можна визначити за формулою [1,4]

$$P_Z = C_O \Delta L_O, \quad (5.4.2)$$

де P_Z - натяг нитки при зевотворюванні;

C_O - коефіцієнт жорсткості основних ниток;

ΔL_O - деформація основних ниток при зевотворюванні.

Остання величина може бути визначена з наступної системи рівнянь (див.рис.5.4.1)

$$\begin{aligned} l_{AB} &= \sqrt{l_2^2 + h(t)^2}, & l_{AC} &= \sqrt{l_1^2 + h(t)^2}, \\ L_{BAC} &= \sqrt{l_2^2 + h(t)^2} + \sqrt{l_1^2 + h(t)^2}, & (5.4.3) \\ \Delta L_O &= L_{BAC} - (l_1 + l_2) = \sqrt{l_2^2 + h(t)^2} + \sqrt{l_1^2 + h(t)^2} - (l_1 + l_2), \end{aligned}$$

де $h(t)$ - поточне значення вертикального зміщення очка галева ремісної рамки;

t - час;

l_1, l_2 - винос та глибина зева.

Тоді значення поточної величини натягу при зевотворюванні, з урахуванням (5.4.2),(5.4.3), буде мати вигляд

$$P_Z = C_O [\sqrt{l_2^2 + h(t)^2} + \sqrt{l_1^2 + h(t)^2} - (l_1 + l_2)]. \quad (5.4.4)$$

Зростання натягу нитки за рахунок руху відносно поверхні очка галева ремісної рамки можна визначити за формулою [4]

$$P = P_\Sigma \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\mu_2 \alpha} - 1) \right] + \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2[R+r(1-\delta)]^2} \right] - \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2[R+r(1-\delta_0)]^2} \right] \times \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\mu_2 \alpha} - 1) \right].$$

де $\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{j(K)}$ - коефіцієнт жорсткості нитки на згин;

E - модуль пружності нитки на розтягнення;

d_i - діаметри окремих елементарних волокон (філаментів), з яких складається комплексна нитка чи пряжа;

w - кількість філаментів, з яких складається пряжа;

$j(K) = 1 + u_1 K^{u_2}$, - коефіцієнт, який залежить від крутки K нитки;

δ, δ_0 - відносна деформація перетину нитки в точках її сходу та входу на напрямну;

u_1, u_2 - деякі сталі коефіцієнти;

R - радіус поверхні очка галева ремісної рамки;

r - розрахунковий радіус нитки;

μ_2 - коефіцієнт тертя між ниткою та поверхнею очка галева ремісної рамки;

$\alpha = \alpha_1 + \alpha_2$ - кут охоплення поверхні очка галева ремісної рамки;

$P_\Sigma = P_1 + P_Z$ - попередній натяг.

Для текстильних ниток, коефіцієнтом жорсткості на вигин яких можна знехтувати $B=0$, отримаємо

$$P = P_\Sigma \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\mu_2 \alpha} - 1) \right]. \quad (5.4.5)$$

Величину кута α , який входить до формули (5.4.5), визначимо по формулі

$$\alpha_1 = \arctg \left[\frac{h(t)}{l_1} \right], \quad \alpha_2 = \arctg \left[\frac{h(t)}{l_2} \right]. \quad (5.4.6)$$

Тоді значення натягу нитки P , з урахуванням (5.4.1), (5.4.4), (5.4.5)-(5.4.6), прийме вигляд

$$P = \left\{ P_O e^{\mu_1 \gamma} + C_O \left[\sqrt{l_2^2 + h(t)^2} + \sqrt{l_1^2 + h(t)^2} - (l_1 + l_2) \right] \right\} \times \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\mu_2 \left\{ \arctg \left[\frac{h(t)}{l_1} \right] + \arctg \left[\frac{h(t)}{l_2} \right] \right\}} - 1 \right) \right]. \quad (5.4.7)$$

На рис.5.4.2 та в Додатках представлені результати визначення натягу ниток основи. Вихідні дані бралися для ткацького верстата АТТ-120-5М і дорівнювали (розрахунки проводилися для однієї нитки в заправці): $P_O = 50$ сН; $\gamma = 1,57$ рад.; $\mu_1 = 0,2$; $\mu_2 = 0,4$; $t = 0,0,314$ с (для кутової швидкості

головного вала верстата $\omega=20 \text{ с}^{-1}$); $l_1=300 \text{ мм}$; $l_2=400 \text{ мм}$; $c_0=2 \text{ сН/мм}$;
 $h = h_{max}=100 \text{ мм}$.

На рис.5.4.2 крива 1 показує зміну натягу основних ниток в процесі зевотворювання. Пряма 2 показує значення попереднього натягу. Аналіз графічної залежності 1 дозволяє стверджувати, що максимальне зростання натягу під час зевотворювання майже в 3 рази більше ніж попередній натяг нитки основи.

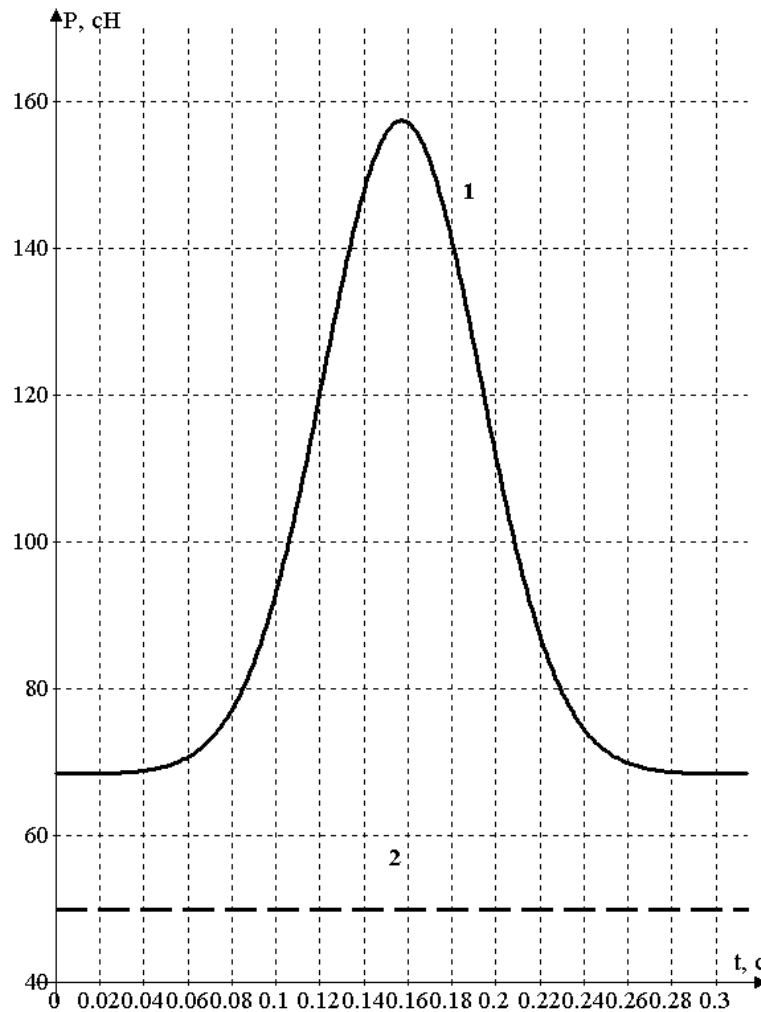


Рис.5.4.2. Графічні залежності натягу основної нитки

Це може призводити до обриву основи, що негативно впливає на продуктивність технологічного обладнання та якість продукції.

Для мінімізації натягу основи в процесі ткацтва в роботі була запропонована конструкція нитконапрямого пристрою ткацького верстата

[16]. На рис.5.4.3 представлена його кінематична схема.

Нитконапрямник ткацького верстата містить кривошип OA , шарнірно закріплений зі станиною, два шатуна BC , CD , шатун BC шарнірно з'єднаний з кривошипом OA , повзунком B , встановленим в горизонтальних направляючих, та шатуном CD . Шатун CD шарнірно з'єднаний з шатуном BC та повзунком D (скало ткацького верстата), встановленим в горизонтальних направляючих.

Нитконапрямник ткацького верстата працює наступним чином. Кривошип OA отримує обертальний рух від головного валу станка через трансмісію. Він передає рух повзунку D , який виконує зворотно-поступальний рух. Вибір розмірів шатунів BC , CD та кривошипу OA в певній пропорції дозволяє забезпечити висотою повзунка D в крайньому правому положенні в момент прокладання уточної нитки (точка шарнірного з'єднання кривошипа з першим шатуном ділить його у відношенні $3/2,5$, а довжина другого шатуна більше довжини кривошипа в $3,5$ разів).

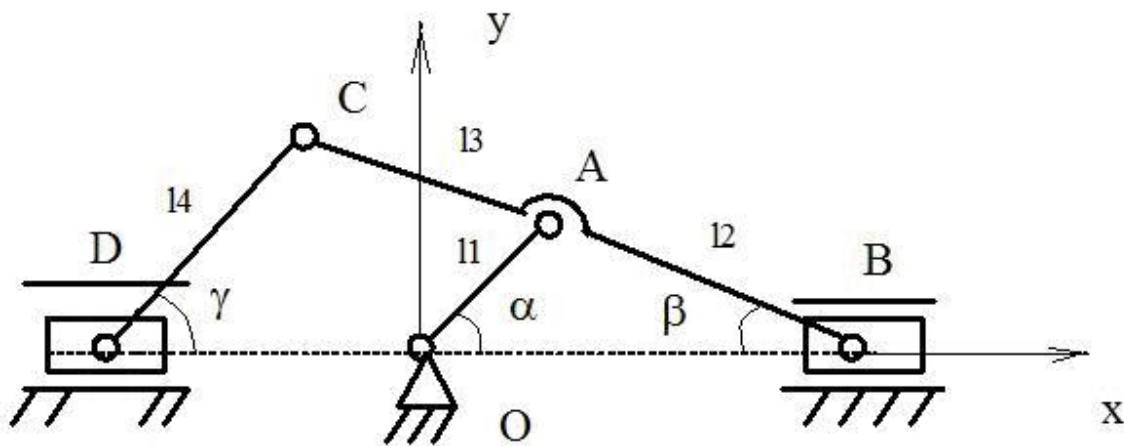


Рис.5.4.3. Кінематична схема нитконапрямника ткацького верстата

Використання даного нитконапрямника, дозволяє стабілізувати натяг ниток основи при зевотворенні за рахунок забезпечення необхідної функції ниткоподачі, що дає змогу виключити можливість обриву ниток. Використання запропонованого нитконапрямника ткацького верстата дозволяє стабілізувати натяг ниток основи при зевотворенні за рахунок

забезпечення необхідної функції ниткоподачі, що дозволяє вдосконалити технологічні процеси та обладнання текстильної промисловості.

Проведемо кінематичний аналіз роботи даного механізму. Для зручності подальших розрахунків визначимо координати і траєкторії руху точок механізму нитконапрямника ткацького верстата. Для цього розташуємо початок координатної системи OXY у точці O – шарніра кріплення кривошипа OA .

Координати точки A визначаються по формулах

$$x_A(\alpha) = l_1 \cos(\alpha), \quad y_A(\alpha) = l_1 \sin(\alpha), \quad (5.4.8)$$

де $x_A(\alpha)$, $y_A(\alpha)$ - координати шарніра A ;

l_1 - довжина кривошипа OA ;

α - кут повороту кривошипа OA .

Для визначення координат точки B прирівняємо вертикальні проекції кривошипа OA і ділянки шатуна AB

$$l_1 \sin(\alpha) = l_2 \sin(\beta),$$

де β - кут нахилу шатуна CB до горизонтальної осі.

Зведемо праву і ліву частини останнього рівняння в квадрат і визначимо вираз для визначення кута β

$$\cos(\beta) = \sqrt{1 - \frac{l_1^2}{l_2^2} \sin^2(\alpha)}. \quad (5.4.9)$$

З урахуванням останньої рівності вираз для визначення координати точки B повзуна визначиться з виразу

$$x_B(\alpha) = l_1 \cos(\alpha) + l_2 \sqrt{1 - \frac{l_1^2}{l_2^2} \sin^2(\alpha)}. \quad (5.4.10)$$

Координати точки C , з урахуванням (4.2), можна визначити по формулах

$$x_C(\alpha) = - \left[(l_2 + l_3) \sqrt{1 - \frac{l_1^2}{l_2^2} (\sin^2(\alpha) - x_B(\alpha))} \right], \quad (5.4.11)$$

$$y_C(\alpha) = (l_2 + l_3) \frac{l_1 \sin(\alpha)}{l_2},$$

де $x_C(\alpha), y_C(\alpha)$ - координати точки С.

Для визначення координат точки D прирівнюємо вертикальні проекції шатуна СВ і шатуна CD

$$l_4 \sin(\gamma) = (l_2 + l_3) \sin(\beta), \quad (5.4.12)$$

де γ - кут нахилу шатуна CD до горизонтальної осі.

Тоді, формула для визначення кута γ , після зведення в квадрат лівої і правої частин рівності (5.4.12), матиме вигляд

$$\cos(\gamma) = \sqrt{1 - \frac{(l_2 + l_3)^2 l_1^2}{l_4^2 l_2^2} (\sin^2(\alpha))}. \quad (5.4.13)$$

З урахуванням виразів (5.4.12) -(5.4.13) переміщення точки D визначається по формулі

$$x_D(\alpha) = - \left[l_4 \sqrt{1 - \frac{(l_2 + l_3)^2 l_1^2}{l_4^2 l_2^2} (\sin^2(\alpha))} + (l_2 + l_3) \sqrt{1 - \frac{l_1^2}{l_2^2} (\sin^2 \alpha - x_B(\alpha))} \right], \quad (5.4.14)$$

де $x_D(\alpha)$ - координата переміщення точки D.

Використовуючи формули (5.4.8) -(5.4.14) були побудовані графіки зміни координат точок В, С и D, які представлені на рис.5.4.4.

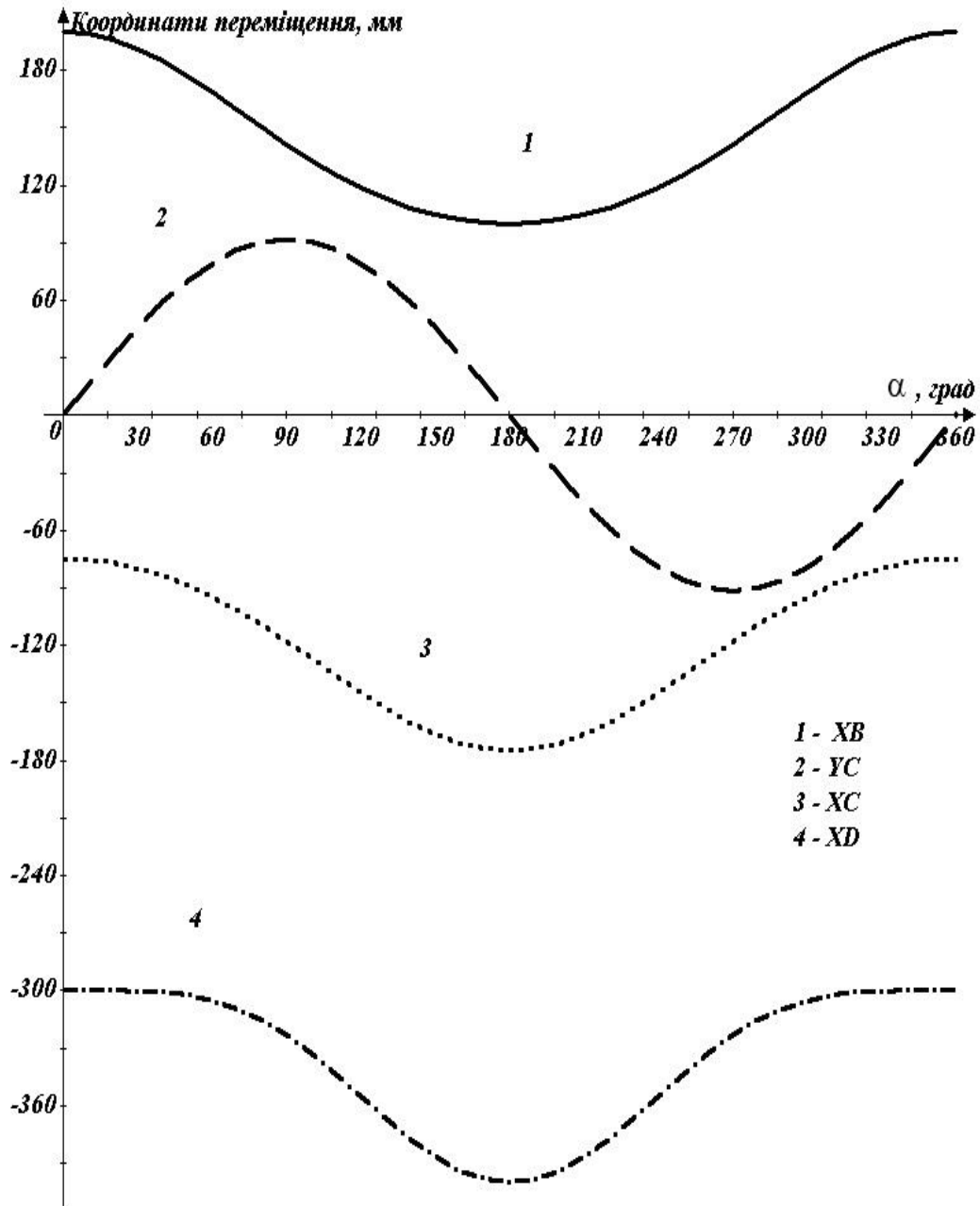


Рис.5.4.4.Графіки зміни координат точок В, С і D

Для визначення швидкості точки В продиференціюємо за часом вираз (5.4.10). В результаті отримаємо

$$V_B(t) = -\frac{\omega l_1^2 \cos(\omega t) \sin(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega l_1 \sin(\omega t), \quad (5.4.15)$$

де $V_B(t)$ - швидкість точки В залежно від часу повороту кривошипа.

Для визначення проекцій вектора швидкості точки С на координатні осі X, Y продиференціюємо вираз (5.4.11) за часом. Після цього отримаємо систему рівнянь

$$V_{CX}(t) = \frac{\omega l_1^2 \cos(\omega t) \sin(\omega t) (l_2 + l_3)}{l_2^2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega l_1^2 \cos(\omega t) \sin(\omega t)}{l_2^2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega l_1 \sin(\omega t), \quad (5.4.16)$$

$$V_{CY}(t) = \frac{\omega l_1 \cos(\omega t) (l_2 + l_3)}{l_2},$$

де $V_{CX}(t), V_{CY}(t)$ - відповідно проекції вектора швидкості точки С на координатні осі X, Y .

Величина швидкості точки С визначиться по формулі

$$V_C = \sqrt{V_{CX}^2(t) + \left[\frac{\omega l_1 \cos(\omega t) (l_2 + l_3)}{l_2} \right]^2}, \quad (5.4.17)$$

де V_C - швидкість точки С.

Для визначення швидкості точки D продиференціюємо за часом вираз (5.4.14). В результаті отримаємо

$$V_D(t) = \frac{\omega l_1^2 \cos(\omega t) \sin(\omega t) (l_2 + l_3)}{l_2^2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega l_1^2 \cos(\omega t) \sin(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega l_1 \sin(\omega t) + \frac{\omega l_1^2 \cos(\omega t) \sin(\omega t) (l_2 + l_3)^2}{l_2^2 l_4 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t) (l_2 + l_3)^2}{l_2^2 l_4^2}}}, \quad (5.4.18)$$

де $V_D(t)$ - швидкість точки D залежно від часу повороту кривошипа ОА.

Використовуючи залежності (5.4.15) -(5.4.18) були побудовані графічні залежності, які представлені на рис.5.4.5.4.

Крива 1 відповідає швидкості точки С. Крива 2 відповідає швидкості точки В. Крива 3 відповідає швидкості точки D. Аналіз отриманих кривих показує, що для точки D на ділянці траєкторії точки С, коли вона рухається по колу радіусу CD швидкість дорівнює нулю.

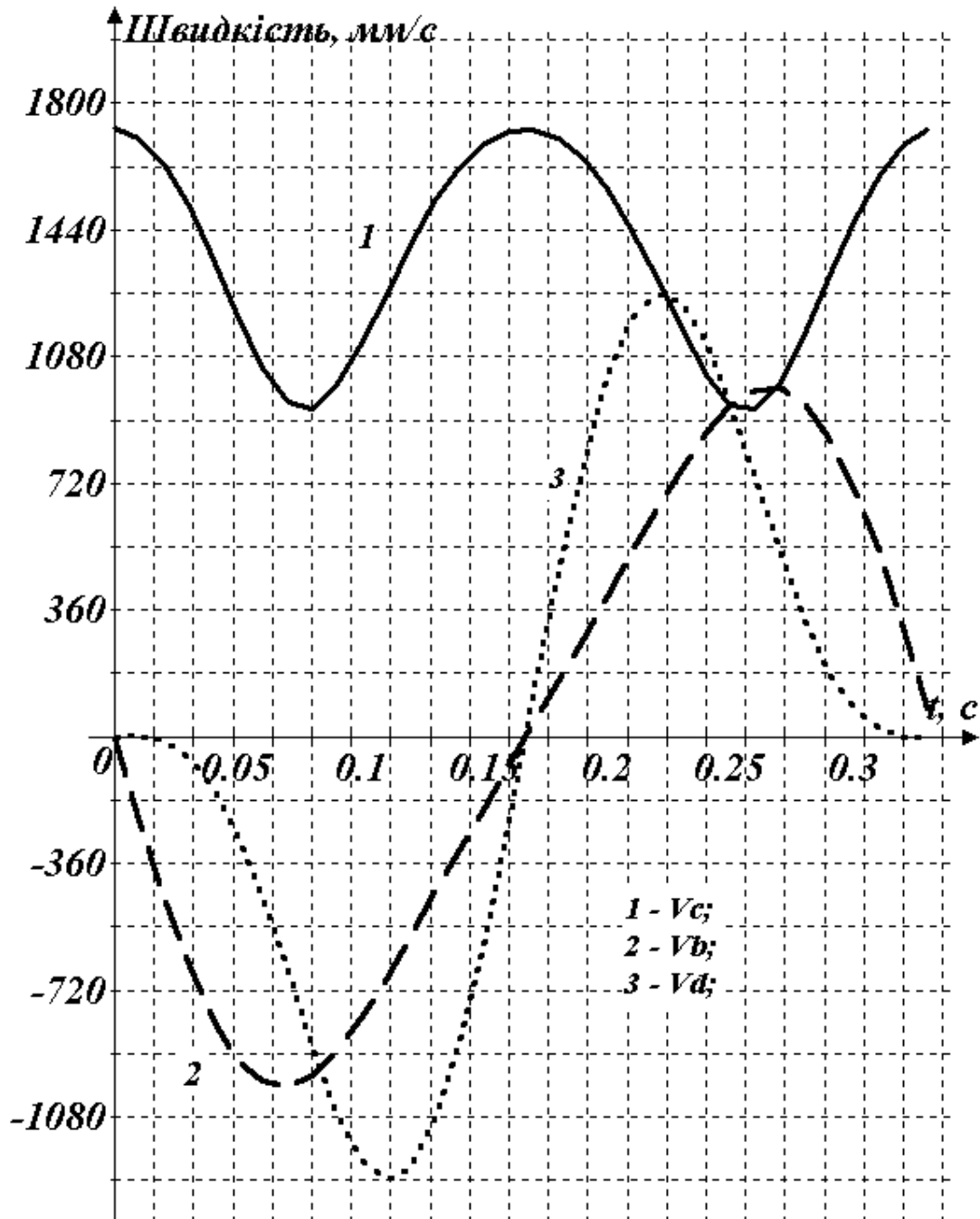


Рис.5.4.5.4. Графічні залежності зміни швидкостей точок механізму нитконапрявника

Для визначення прискорення точки В продиференціюємо за часом вираз (5.4.15). В результаті отримаємо

$$W_B(t) = \frac{\omega^2 l_1^2 \sin^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^2 \cos^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega^2 l_1 \cos(\omega t) - \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)}{l_2^3 \left(1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}\right)^{\frac{3}{2}}}, \quad (5.4.19)$$

де $W_B(t)$ - прискорення точки В залежно від часу повороту кривошипа ОА. Для визначення проєкцій вектора прискорення точки С на координатні осі X, Y продиференціюємо вираз (5.4.16) за часом. Після цього отримаємо систему рівнянь

$$W_{CX}(t) = \frac{\omega^2 l_1^2 \sin^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^2 \cos^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega^2 l_1 \cos(\omega t) + \frac{\omega^2 l_1^2 \cos^2(\omega t)(l_2 + l_3)}{l_2^2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^2 \sin^2(\omega t)(l_2 + l_3)}{l_2^2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)}{l_2^3 \left(1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}\right)^{\frac{3}{2}}} + \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)(l_2 + l_3)}{l_2^4 \left(1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}\right)^{\frac{3}{2}}}, \quad (5.4.20)$$

$$W_{CY}(t) = -\frac{\omega^2 l_1 \sin(\omega t)(l_2 + l_3)}{l_2}.$$

де $W_{CX}(t), W_{CY}(t)$ - відповідно проєкції прискорення точки С на осі X, Y в залежності від часу повороту кривошипа ОА.

Величина прискорення точки С визначиться по формулі

$$W_C(t) = \sqrt{W_{CX}^2 + \left[\frac{\omega^2 l_1 \sin(\omega t)(l_2 + l_3)}{l_2} \right]^2}. \quad (5.4.21)$$

Для визначення прискорення точки D продиференціюємо за часом вираз (5.4.18). В результаті отримаємо

$$\begin{aligned}
 W_D(t) = & -\frac{\omega^2 l_1^2 \sin^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^2 \cos^2(\omega t)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \omega^2 l_1 \cos(\omega t) + \\
 & + \frac{\omega^2 l_1^2 \cos^2(\omega t)(l_2 + l_3)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \frac{\omega^2 l_1^2 \sin^2(\omega t)(l_2 + l_3)}{l_2 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}}} - \\
 & - \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)}{l_2^3 \left(1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}\right)^{\frac{3}{2}}} + \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)(l_2 + l_3)}{l_2^4 \left(1 - \frac{l_1^2 \sin^2(\omega t)}{l_2^2}\right)^{\frac{3}{2}}} + \\
 & + \frac{\omega^2 l_1^2 \cos^2(\omega t)(l_2 + l_3)^2}{l_2^2 l_4 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)(l_2 + l_3)^2}{l_2^2 l_4^2}}} - \frac{\omega^2 l_1^2 \sin^2(\omega t)(l_2 + l_3)^2}{l_2^2 l_4 \sqrt{1 - \frac{l_1^2 \sin^2(\omega t)(l_2 + l_3)^2}{l_2^2 l_4^2}}} + \\
 & + \frac{\omega^2 l_1^4 \cos^2(\omega t) \sin^2(\omega t)(l_2 + l_3)^4}{l_2^4 l_4^3 \left(1 - \frac{l_1^2 \sin^2(\omega t)(l_2 + l_3)^2}{l_2^2 l_4^2}\right)^{\frac{3}{2}}}.
 \end{aligned} \tag{5.4.22}$$

де $W_D(t)$ - прискорення точки D залежно від часу повороту кривошипа OA.

Використовуючи залежності (5.4.19) -(5.4.22) були побудовані графічні залежності, які представлені на рис.5.4.6.

Крива 1 відповідає прискоренню точки В. Крива 2 відповідає прискоренню точки С. Крива 3 відповідає прискоренню точки D. Аналіз отриманих кривих показує, що для точки D, на ділянці траєкторії точки С, коли вона рухається по колу радіусу CD прискорення орієнтовано дорівнює нулю.

Результати кінематичних досліджень руху механізму нитконапрячника можуть бути використані при кінетостатичному дослідженні роботи вказаного механізму.

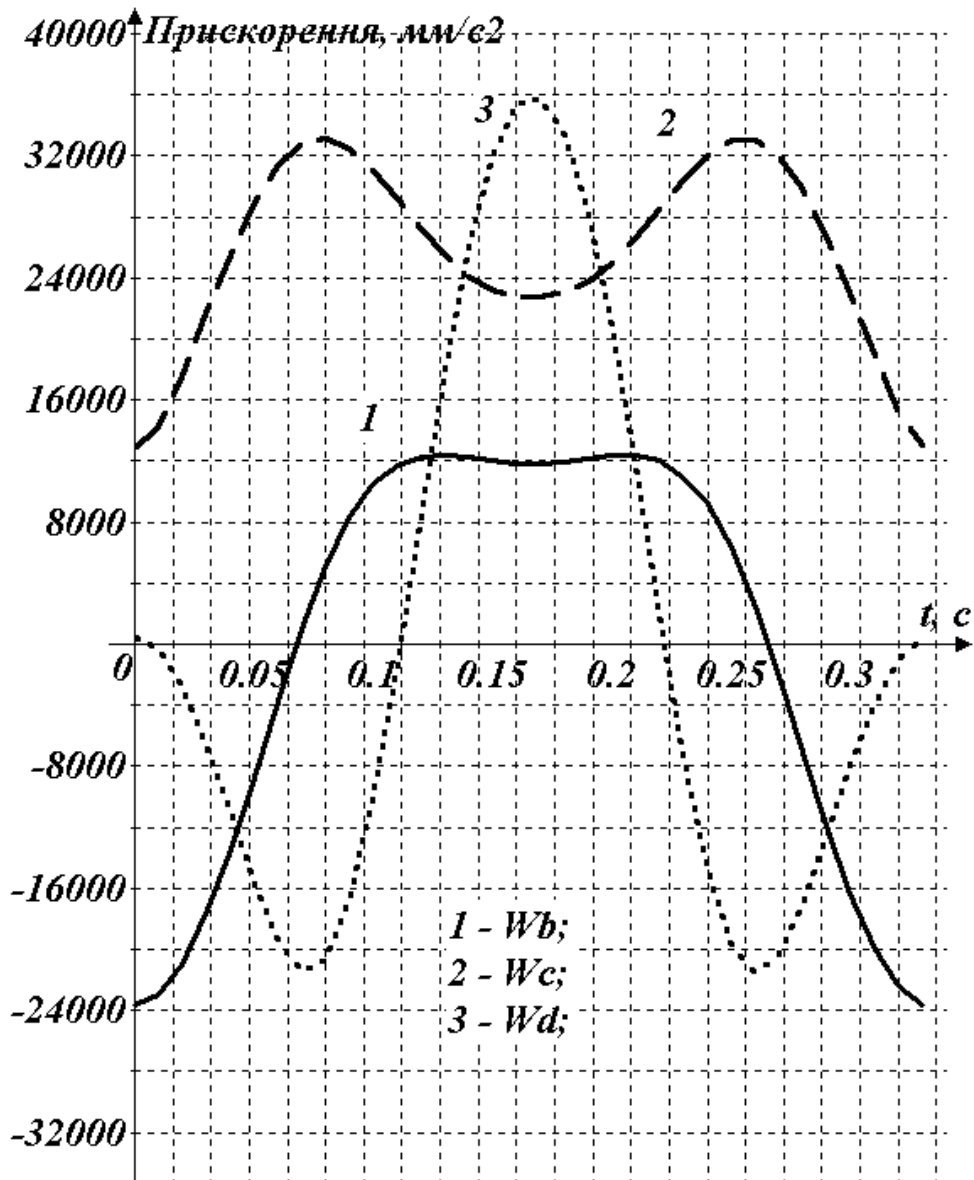


Рис.5.4.6. Графіки зміни прискорень точок механізму нитконапрявника

Використовуючи результати кінематичних досліджень, а також досліджень по вибору оптимальних розмірів циліндричних нитконапрявників [18], представимо формулу (5.4.7) для визначення натягу ниток основи у вигляді

$$P = P_0 e^{\mu_3 \gamma} \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\mu_2 \left\{ \arctg \left[\frac{h(t)}{l1} \right] + \arctg \left[\frac{h(t)}{l2} \right] \right\}} - 1 \right) \right], \quad (5.4.23)$$

де μ_3 - коефіцієнт тертя нитки по поверхні скала у випадку його оптимальних геометричних розмірів. На рис.5.4.7 та в Додатках

представлені залежності натягу ниток основи від кута обертання головного вала ткацького верстата.

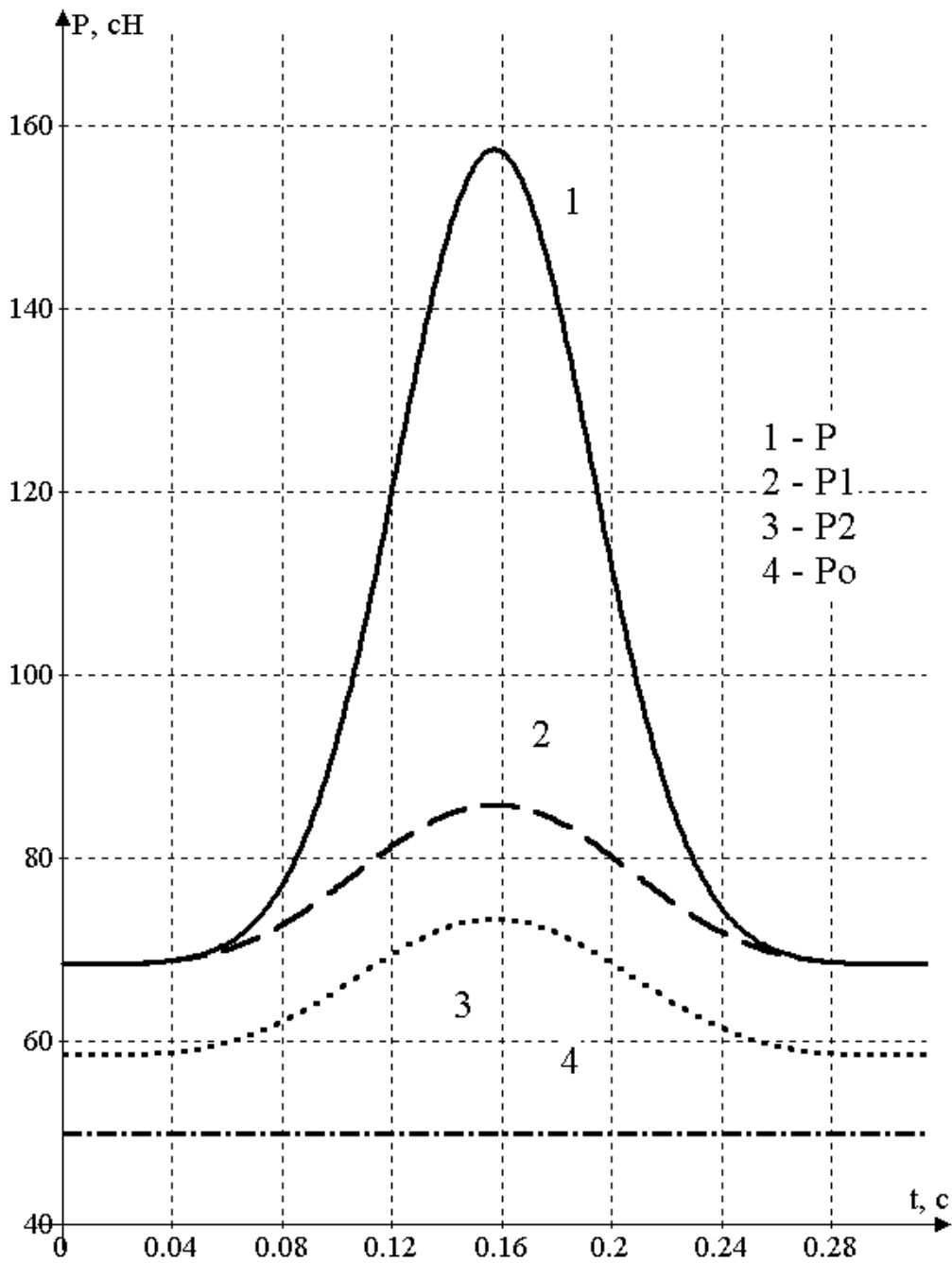


Рис.5.4.7. Графічні залежності натягу ниток основи

Аналіз графічних залежностей показав, що при використанні нитконапрявника натяг ниток основи зменшується на 60-70 сН. При використанні скала з оптимальними геометричними розмірами натяг основи зменшується на 10-15 сН. Таким чином, загальне зменшення натягу

основи буде складати 70-80 сН. Використання запропонованого нитконапрямника ткацького верстата дозволяє мінімізувати натяг ниток основи при зевотворенні за рахунок забезпечення необхідної функції ниткоподачі, що дозволяє вдосконалити технологічні процеси та обладнання текстильної промисловості.

Оптимізація процесу взаємодії нитки з напрямними з урахуванням анізотропії фрикційних властивостей

Подовжнє і поперечне ковзання ниток по напрямних з постійною кривизною поверхні або по площині має місце при формуванні елементу тканини і трикотажу, при взаємодії ниток з направляючими органами швейних і текстильних машин [1,4]. Визначення натягу і форми осі, з урахуванням анізотропії фрикційних властивостей як ниток [15] так і напрямних, дозволяє на початковому етапі проектування технологічних процесів вибрати їх оптимальні параметри. Таким чином, тема цього підрозділу є актуальною, що має важливе значення для удосконалення технологічних процесів текстильної і трикотажної промисловості.

Удосконалення технологічних процесів легкої і текстильної промисловості, як об'єктів дослідження, повинно базуватися на комплексних дослідженнях процесу взаємодії ниток з направляючими поверхнями великої і малої кривизни [17], площинами з урахуванням анізотропії фрикційних властивостей як ниток так і напрямних. Отримані результати дозволять оптимізувати процес взаємодії ниток з напрямними з урахуванням анізотропії фрикційних властивостей як ниток так і напрямних в зоні формування тканини і трикотажу, зменшити відсоток обривів ниток, поліпшити якість готової продукції.

Будова поверхонь робочих органів трикотажних і текстильних машин

(рис.5.4.8), а також самих ниток (рис.5.4.9) свідчать про те, що величина сили тертя в ортогональних напрямках (наприклад дотичній і нормалі до точки на осі нитки) відрізнятимуться. Це дозволяє говорити про необхідність врахування анізотропії фрикційних властивостей. У загальному випадку величина сили тертя визначається з виразу [22]

$$F = N\sqrt{k_1^2 + k_2^2}, \quad (5.4.24)$$

де F - сила тертя; N - сила нормального тиску; $k = \sqrt{k_1^2 + k_2^2}$ - ефективний коефіцієнт тертя [123].

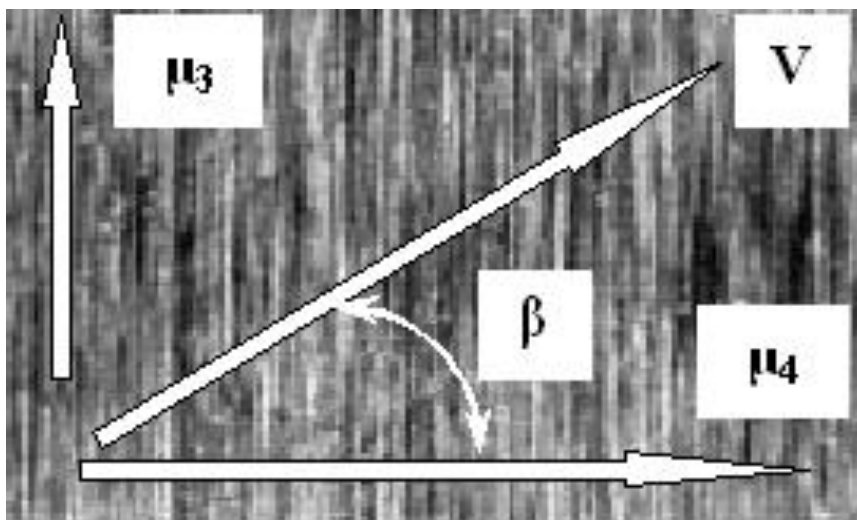


Рис.5.4.8. Анізотропія фрикційних властивостей напрямної

Для двох ортогональних напрямів (дотична і нормаль) вираз ефективного коефіцієнта тертя матиме вигляд

$$k = \sqrt{k_1^2 + k_2^2} = \sqrt{\mu_2^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta}. \quad (5.4.25)$$

де μ_1, μ_2 - коефіцієнти тертя нитки в двох ортогональних напрямках;
 β - кут, який утворює вектор швидкості з дотичною до осі нитки.

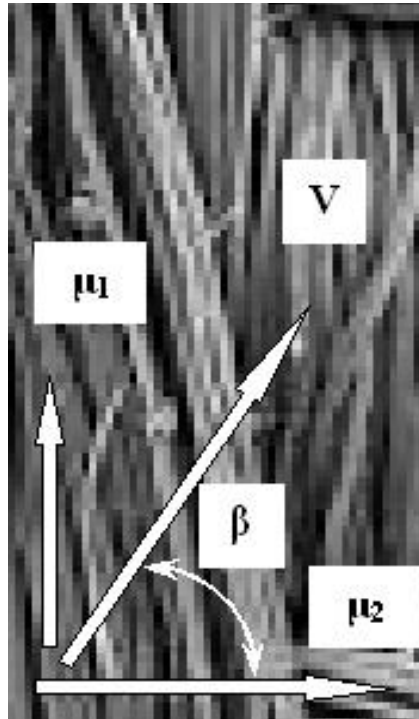


Рис.5.4.9. Анізотропія фрикційних властивостей нитки

Для теоретичного дослідження впливу матеріалу нитки і анізотропії тертя на натяг і форму її осі були вибрані комплексні нитки та пряжа (рис.5.4.9), які мають значну анізотропію фрикційних властивостей в двох ортогональних напрямках – вздовж окремих філаментів та перпендікулярно до них.

Знайдемо похідну від ефективного коефіцієнта тертя по куту β та прирівняємо її до нуля

$$\begin{aligned} \frac{\partial}{\partial \beta} \left(\frac{F}{N} \right) &= \frac{\partial}{\partial \beta} \left(\sqrt{\mu_2^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta} \right) = \\ &= \frac{-\mu_2^2 \cos \beta \sin \beta + \mu_1^2 \cos \beta \sin \beta}{\sqrt{\mu_2^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta}} = 0. \end{aligned} \quad (5.4.26)$$

Це рівняння буде можливим у двох випадках

$$\sin \beta = 0, \quad \beta = 0, \quad \cos \beta = 0, \quad \beta = \pi / 2.$$

Знайдемо другу похідну, для чого рівняння (5.4.26) ще раз продифференціюємо по β

$$\frac{\partial^2}{\partial \beta^2} \left(\frac{F}{N} \right) = \frac{-(\mu_2^2 \cos \beta \sin \beta - \mu_1^2 \cos \beta \sin \beta)^2}{(\mu_2^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta)^{\frac{3}{2}}} - \frac{\mu_2^2 \cos^2 \beta - \mu_2^2 \sin^2 \beta - \mu_1^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta}{\sqrt{\mu_2^2 \cos^2 \beta + \mu_1^2 \sin^2 \beta}}. \quad (5.4.27)$$

В характерних точках $\beta = 0$, $\beta = \pi/2$ значення других похідних наведено в системі рівнянь (5.4.28). У випадку $\beta = 0$ ефективний коефіцієнт тертя буде мати своє максимальне значення, а у випадку $\beta = \pi/2$ - мінімальне значення.

$$\begin{aligned} \frac{\partial^2}{\partial \beta^2} \left(\frac{F}{N} \right) &= \frac{\mu_2^2 - \mu_1^2}{\mu_1}, \quad \beta = \frac{\pi}{2}, \quad \frac{\partial^2}{\partial \beta^2} \left(\frac{F}{N} \right) = \frac{-\mu_2^2 + \mu_1^2}{\mu_2}, \quad \beta = 0, \quad \mu_2 > \mu_1, \\ \beta = 0 &\rightarrow \frac{\partial^2}{\partial \beta^2} \left(\frac{F}{N} \right) < 0 \rightarrow \max, \quad \beta = \frac{\pi}{2} \rightarrow \frac{\partial^2}{\partial \beta^2} \left(\frac{F}{N} \right) > 0 \rightarrow \min. \end{aligned} \quad (5.4.28)$$

В загальному випадку, враховуючи структуру направляючої (рис.5.4.8) та самої нитки (рис.5.4.9), анізотропія фрикційних властивостей може бути представлена системою нерівностей $\mu_1 > \mu_2$, $\mu_3 > \mu_4$. У разі, коли $\mu_1 = \mu_2$, $\mu_3 = \mu_4$ фрикційні властивості направляючої поверхні та нитки обраховуються як для ізотропних об'єктів. На рис.5.4.10 представлена загальна розрахункова схема взаємодії двох поверхонь з анізотропними фрикційними властивостями.

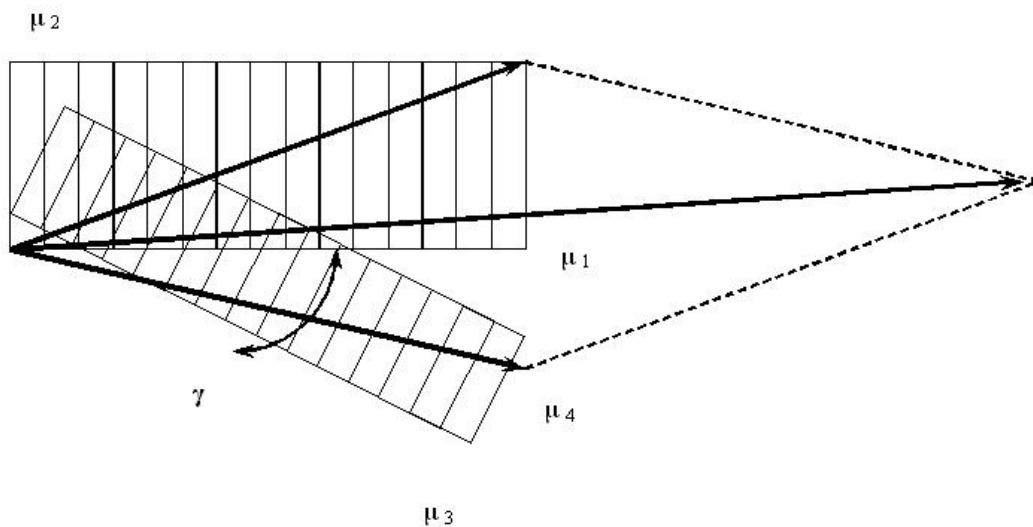


Рис.5.4.10. Розрахункова схема

Два прямокутники, діагоналі яких представляють собою ефективні

коефіцієнти тертя (з кутами відхилення діагоналей від горизонталі $\beta_{12} = \arctg(\mu_2 / \mu_1)$, $\beta_{34} = \arctg(\mu_4 / \mu_3)$, в загальному випадку будемо вважати повернутими один відносно другого на кут γ . Тоді значення ефективного коефіцієнта тертя для цієї пари можна визначити по формулі

$$k = \sqrt{(\mu_3 \cos \gamma + \mu_4 \sin \gamma + \mu_1)^2 \cos^2 \beta + \{\mu_2 - [\mu_3 \sin(\gamma + \pi) - \mu_4 \cos \gamma]\}^2 \sin \beta}. \quad (5.4.29)$$

Для направляючої поверхні коефіцієнт тертя в напрямку найменшого супротиву руху позначимо u , який характеризуватиме ефект від супротиву зсуву у цьому напрямку. Враховуючи, що $\mu_3 > \mu_4$ отримаємо

$$\mu_4 = u, \quad \mu_3 = au, \quad a > 1, \quad \frac{\mu_3}{\mu_4} = a, \quad \mu_3 = a\mu_4.$$

Тоді рівняння (6) прийме вигляд

$$k = \sqrt{(au \cos \gamma + u \sin \gamma + \mu_1)^2 \cos^2 \beta + \{\mu_2 - [au \sin(\gamma + \pi) - u \cos \gamma]\}^2 \sin \beta}. \quad (5.4.30)$$

Коли прямокутники не повернуті один відносно другого, при $\gamma = 0$, рівняння (5.4.30) прийме вигляд

$$k = \sqrt{(au + \mu_1)^2 \cos^2 \beta + (\mu_2 + u)^2 \sin \beta}. \quad (5.4.31)$$

У випадку, коли $\gamma = \frac{\pi}{2}$, рівняння (5.4.31) прийме вигляд

$$k = \sqrt{(u + \mu_1)^2 \cos^2 \beta + (\mu_2 + au)^2 \sin \beta}. \quad (5.4.32)$$

Для ізотропної направляючої поверхні, при $a = 1$, рівняння (5.4.32) прийме вигляд

$$k = \sqrt{(u + \mu_1)^2 \cos^2 \beta + (\mu_2 + u)^2 \sin \beta}. \quad (5.4.33)$$

Аналіз рівнянь (5.4.29-5.4.33) дозволяє зробити висновок, що враховуючи фрикційні властивості напрямної і самої нитки, а також напрямок їх відносного переміщення, який характеризується кутом β , можна отримати найбільш оптимальну схему, при якій супротив відносного зсуву нитки відносно напрямної буде мінімальним. На рис.5.4.11 представлені відповідні діаграми ефективного коефіцієнту тертя

k . Результати були отримані для наступних варіантів: 1 - $\beta=0, \gamma=0, k=0,6$; 2 - $\beta=0, \gamma=\pi/2, k=0,5$; 3 - $\beta=\pi/2, \gamma=0, k=0,2$; 4 - $\beta=\pi/2, \gamma=\pi/2, k=0,3$. Аналіз отриманих результатів показав, що значення ефективного коефіцієнту тертя (варіант 1) в три рази вище ніж у випадку, коли $\beta=\pi/2, \gamma=0, k=0,2$ (варіант 3).

Виходячи з цього необхідно при виконанні розрахунків натягу нитки, та визначення форми її осі враховувати значення ефективного коефіцієнту тертя та підбирати відповідні параметри таким чином, щоб отримати мінімальне значення сили тертя .

У випадку руху нитки по поверхні циліндра $\beta=0, \gamma=0$ залежність для визначення натягу буде мати вираз [17]

$$\ln P = \ln P_0 + \frac{a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n} [\ln R(1-n)+1] / [1 + \frac{(1-n)a\varphi^{1-n_1}}{b(1-n_1)} \left(\frac{R}{P_0}\right)^{1-n}] , P = P_0 e^{k\varphi \left(\frac{R}{P_0}\right)^{1-n}}$$

де P - вихідний натяг нитки; P_0 - вхідний натяг нитки; R - радіус циліндричної напрямної; φ - кут охоплення ниткою циліндричної напрямної; n, a, b, n_1 - деякі константи, значення яких залежить від виду матеріалів нитки та напрямної та умов взаємодії між ними.

k - ефективний коефіцієнт тертя

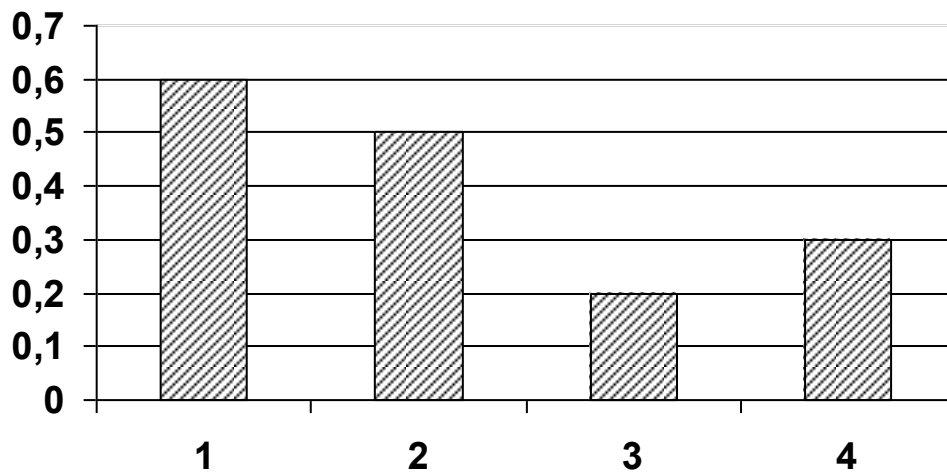


Рис.5.4.11. Значення ефективного коефіцієнту тертя

У випадку руху нитки по поверхні деякої площини $\beta \neq 0$, $\gamma \neq 0$ було отримано диференціальне рівняння, яке представляє диференціальне рівняння другого порядку і визначає форму осі нитки при поперечному ковзанні з урахуванням анізотропії тертя [12]

$$\left[1 + \left(\frac{dy}{dx} \right)^2 \right]^{\frac{\mu_1}{2\mu_2} - 1} \frac{d^2 y}{dx^2} = \frac{\mu_2 \mu_0^2 g}{P_0}.$$

Отримано основні залежності для оптимізації процесу взаємодії нитки з напрямними з погляду мінімізації величини сили тертя. Це дозволило визначити форму осі нитки і її натяг в довільній точці, удосконалити процес її переробки на технологічному устаткуванні текстильної і трикотажної промисловості.

Найбільший натяг матимуть нитки з найбільшим значенням ефективного коефіцієнта тертя. Реальний натяг, в середньому, у анізотропних ниток в порівнянні з ізотропними буде більше на 20-30%.

Удосконалення нитконатягувального пристрою в'язальних машин

Корисна модель відноситься до області вдосконалення нитконатяжних пристроїв машин легкої та текстильної промисловості, зокрема для вдосконалення конструкції тарілчастих нитконатягувальних пристроїв.

Відомий тарілчастий нитконатягувальний пристрій [4], який використовується в трикотажних та швейних машинах. Нитконатягувальний пристрій складається зі стрижня, на який надягнуто дві тарілочка. Тиск між тарілочками створюється пружиною. Силу тиску можна змінювати за допомогою гайки.

Тарілчастий нитконатягувальний пристрій дозволяє забезпечити необхідний натяг для реалізації технологічного процесу. Але даний тарілчастий нитконатягувальний пристрій має суттєвий недолік. При проходженні вузлів, шишок та локальних потовщень відбувається різке збільшення натягу за рахунок ударної дії, що призводить до обриву нитки. Крім того, різке збільшення натягу призводить до порушення нормального проходження технологічного процесу переробки нитки.

Найбільш близьким по суті та бажаному ефекту до технічного рішення є тарілчастий нитконатягувальний пристрій [32-36].

Нитконатягувальний пристрій містить стрижень з надягнутими на нього двома тарілочками. Тиск між тарілочками створюється пружиною. Силу тиску можна змінювати за допомогою гайки. Для стабілізації натягу пристрій оснащено двоплечовим важелем. Один кінець важеля взаємодіє з пружиною, а з іншим взаємодіє рухома нитка. Вісь двоплечового важеля і стрижень тарілчастого нитконатягувального пристрою перехрещуються під кутом 90° . При зміні натягу відбувається обертання двоплечового важеля, що призводить до зміни ступеня стискання пружини, а це в свою чергу, призводить до зміни тиску тарілочок на нитку в зоні їх контакту.

Але при наявності на нитці вузла, шишки або локального потовщення при взаємодії з двоплечовим важелем відбувається ударний вплив, що призводить до різкого збільшення натягу та обриву нитки. Це спричинено тим, що радіус кривизни направляючої поверхні двоплечового важеля співрозмірний з діаметром вузла, шишки або потовщення. Крім того, наявність двоплечового важеля не виключає ударну взаємодію вузла, шишки або локального потовщення з поверхнею тарілочок нитконатягувального пристрою. Ця обставина негативно впливає на нормальне проходження технологічного процесу.

Метою даної корисної моделі є виключення стрибкоподібного збільшення натягу нитки, що з'являється за рахунок ударного впливу при проходженні вузлів, шишок або локальних потовщень через тарілчасті нитконатягувальні пристрої і, як наслідок, виключення обриву нитки.

Поставлена мета досягається тим, що нитконатягувальний пристрій містить стрижень з надягнутими на нього двома тарілочками, пружину, яка надягнута на стрижень, яка забезпечує необхідне стискання тарілочок за допомогою гайки [32-36]. Пристрій містить фігурний вилковий одноплечий важіль, вісь якого паралельна осі стрижня нитконатягувального пристрою, робоча поверхня важеля має ввігнуту форму, що утворена двома півколами, а в точці контакту з тарілочками з неробочої сторони має два клиноподібні штовхачі, обмежувач переміщень верхньої тарілки у вигляді гвинтової пари та пружину стискання. Форма фігурного вилкового одноплечого важеля утворюється з відрізка прямої та двох дуг кола, радіус якого дорівнює зовнішньому радіусу тарілочок.

Порівняльний аналіз з прототипом показує, що заявлений нитконатягувальний пристрій від прототипу тим, що має фігурний одновилковий важіль, вісь якого паралельна вісі стрижня нитконатягувального пристрою. Крім того, робоча поверхня важеля має ввігнуту форму, яка створена двома півколами, а в точці контакту з тарілочками з

неробочої сторони має два клиноподібних штовхачі. Нитконатягувальний пристрій також містить обмежувач переміщення верхньої тарілки у вигляді гвинтової пари й пружини стискання, а форма фігурного вилкового одноплечого важеля утворюється з відрізка прямої і двох дуг кіл радіуса, який дорівнює зовнішньому радіусу тарілочок. Перераховані вище ознаки дозволяють виключити стрибкоподібне збільшення натягу нитки, що виникає за рахунок ударного впливу при проходженні вузлів, шишок і локальних потовщень через тарілчасті нитконатягувальні пристрої і, як наслідок, виключити обриви нитки [4].

На рис.5.4.12 зображено нитконатягувальний пристрій, вид збоку; на рис.5.4.13 зображено нитконатягувальний пристрій, вид зверху; на рис.5.4.14 - переріз А-А на рис.5.4.13.

Нитконатягувальний пристрій містить стрижень 1, з двома тарілочками 2 і 3, що надягнуті на нього, пружину 4 і гайку 5, що забезпечують необхідний ступінь стискання пружини 4. На осі 6 розташовується фігурний вилковий одноплечий важіль 7, що прикріплений до пружини стискання 8 і має два клиноподібних штовхачі 9. Обмежувач 10 переміщення верхньої тарілки виконаний у вигляді гвинтової пари.

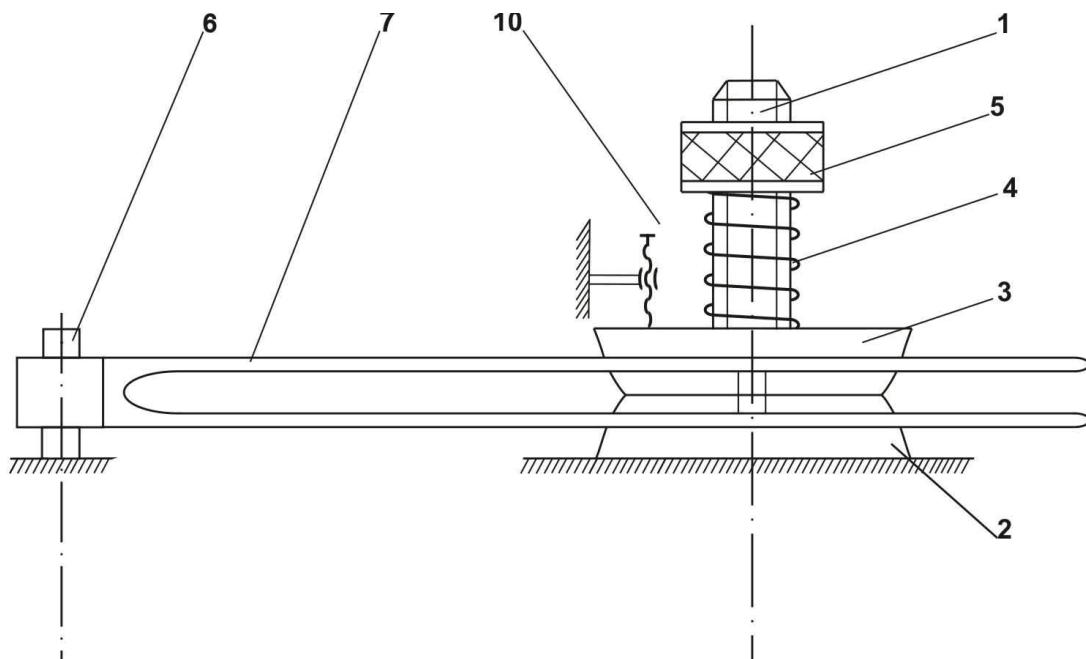


Рис.5.4.12. Нитконатягувальний пристрій (вид збоку)

Нитконатягувальний пристрій працює в такий спосіб. Нитка 11, виходячи із точки, заправляється між зубцями фігурного вилкового одноплечого важеля 7, далі проходить під тарілочками 2, 3, огинає стрижень 1 і надходить у точку С. При відсутності вузлів і потовщень нитка має форму, що показана пунктирною лінією на рис.5.4.13. При наявності вузла або шишки, останні, при переміщенні нитки, починають давити на поверхню одноплечого важеля 7. Відстань між зубцями вилки менша діаметра вузла. Момент, що виникає від сили впливу вузла на робочу поверхню одноплечого важеля 7, викликає стискання пружини 8, при цьому два клиноподібних штовхачі розсовують тарілочку 2 і 3 і вивільняють нитку 11. Величина переміщення верхньої тарілочку регулюється обмежувачем 10. При цьому нитка 11 має поточну форму, показану штрих-пунктирною лінією на рис.5.4.13. Вузол або шишка рухаються по робочій поверхні одноплечого важеля 7 не потрапляючи під тарілочку. Після сходу вузла або нитки з робочої поверхні одноплечого важеля 7 під дією пружини стискання у вихідний стан.

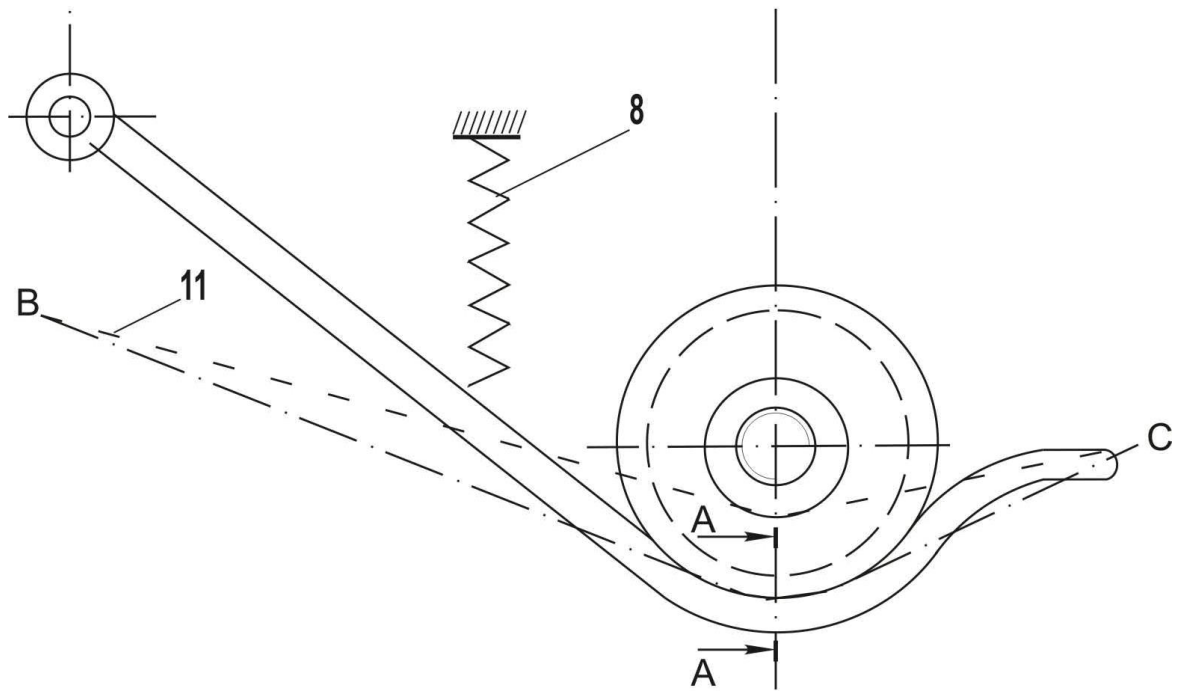


Рис.5.4.13. Нитконатягувальний пристрій (вид зверху)

Використання запропонованого нитконатягувального пристрою дозволяє виключити стрибкоподібне збільшення натягу нитки, що виникає за рахунок ударного впливу при проходженні вузлів, шишок і локальних потовщень і виключити обрив ниток. Це дозволить значно підвищити продуктивність технологічного устаткування за рахунок зниження часу простоїв для ліквідації обриву й підвищити якість продукції, що випускається.

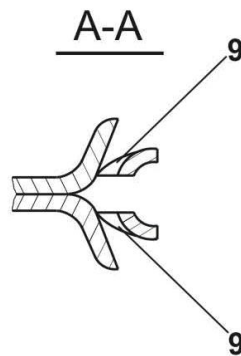


Рис.5.4.14. Перетин А-А

Розрахунок нитконатягувального пристрою за відомою методикою [1,4]

призводить до неточних результатів. При розрахунках необхідно враховувати змінання нитки в зоні контакту з напрямною, її жорсткість на вигин. Виходячи з цього для розрахунку вихідного натягу будемо враховувати результати отримані в 3 розділі. Тоді, з використанням формул (3.1.18)-(3.1.21), отримаємо

$$P = P_o \left\{ \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_o [R+r(1-\delta)]^2} \right] - \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_o [R+r(1-\delta_0)]^2} \right] \right\} \times$$

$$\times \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \frac{\mu_1 N}{1 + \sin(\beta - \frac{\varphi}{2})} (1 + \left\{ \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] + \right.$$

$$\left. + \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_o [R+r(1-\delta)]^2} \right] - \left[\frac{\left(\frac{E\pi}{64} \sum_{i=1}^w d_i^4 \right)^{(1+u_1 K^{u_2})}}{2P_o [R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] \right\}.$$

Для більшості текстильних ниток жорсткістю на вигин можна знехтувати, тоді останнє рівняння прийме вигляд

$$P = P_o \left\{ \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] \right\} +$$

$$+ \frac{\mu_1 N}{1 + \sin(\beta - \frac{\varphi}{2})} (1 + \left\{ \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{f\varphi} - 1) \right] \right\}). \quad (5.4.34)$$

На рис.5.4.15 представлені графічні залежності зміни вихідного натягу в нитконатягувальному пристрої за старою методикою (крива 1) та за формулою (5.4.34) (крива 2). Вертикальна лінія 3 показує той момент, коли кут охоплення φ стрижня ниткою буде співпадати с кутом $\beta = \arcsin(r1/R1)$, де $r1$ - радіус стрижня; $R1$ - контактний радіус тарілок. Порівняльний аналіз показує, що розрахунки вихідного натягу за старою методикою відрізняються від даних отриманих при розрахунках з урахуванням

зминання та жорсткості нитки на згин на 3-22 сН (в середньому різниця складає 20%). Це підтвердили і результати експериментальних досліджень [1,4]. Таке занижене значення натягу може призвести до суттєвих помилок при проектуванні системи ниткоподачі на трикотажних машинах.

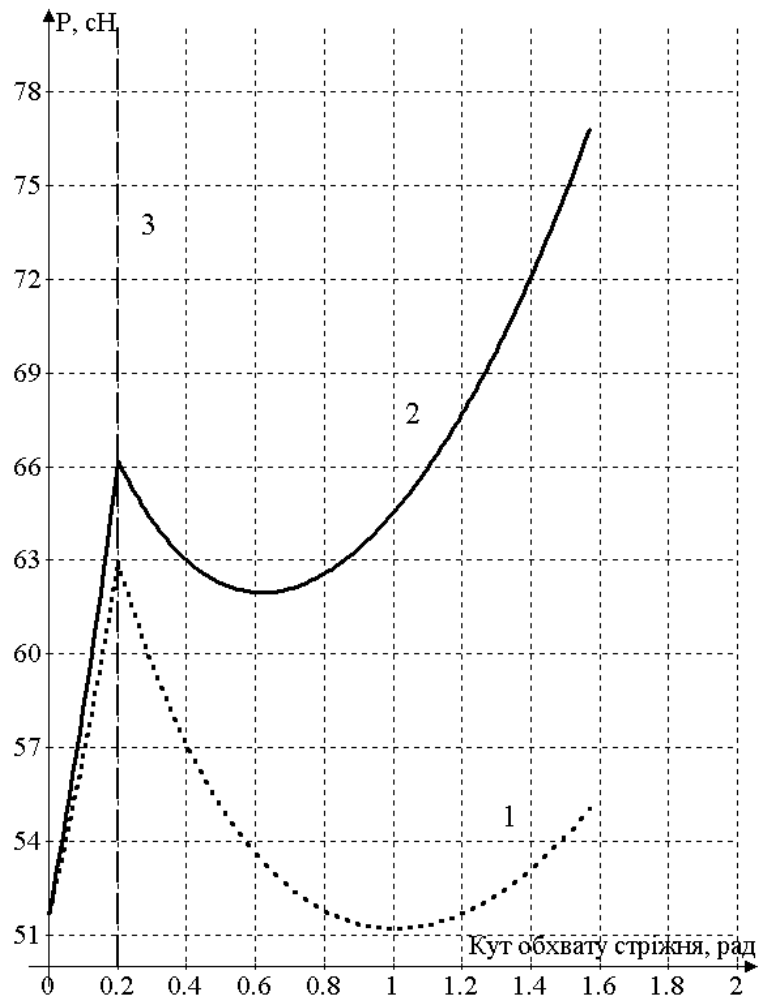


Рис.5.4.15.4. Графік зміни натягу нитки в нитконатягувальному пристрої

Удосконалення системи ниткоподачі на мотальній машині М-150-2

На рис.5.4.16 представлена схема пружної заправки мотальної машини М-150-2. На цій схемі можна виділити чотири основні зони: зона А – від бобіни до балоногасника; зона Б – від балоногасника до двозонного

нитконатягувача; зона В – включає нитконатягувач і до контрольно-очисного пристрою; зона Г – від контрольно-очисного пристрою до зони намотування.

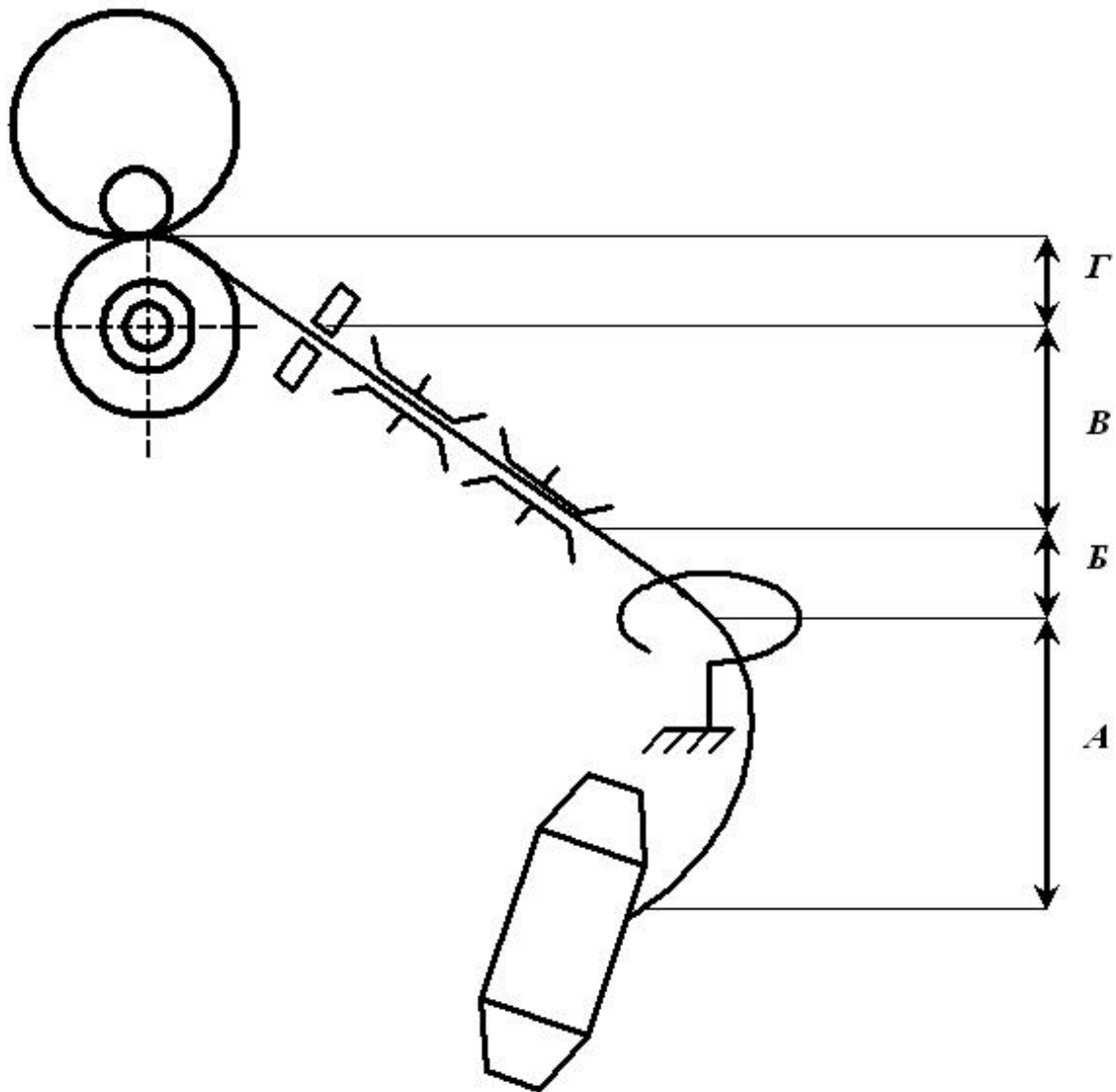


Рис.5.4.16.Схема пружної заправки мотальної машини М-150-2

На рис.5.4.17 представлена діаграма зміни відносного натягу по інтегрованим зонам. Зона А-Б - відношення натягу нитки наприкінці зони А до натягу наприкінці зони Б. Зона Б-В - відношення натягу нитки наприкінці зони Б до натягу наприкінці зони В. Зона В-Г - відношення натягу нитки наприкінці зони В до натягу наприкінці зони Г. Аналіз даних

діаграми показує, що максимальний відносний натяг буде у зоні В-Г – 3,9. Відносний натяг в зонах А-Б та В-Г близький до 1. Таким чином постає задача по модернізації умов переробки нитки в зоні В.

Ця зона представляє собою двозонний нитконатягувач. На рис.5.4.18 представлений один з нитконатягувачів, а на рис.5.4.19 представлена розрахункова схема.

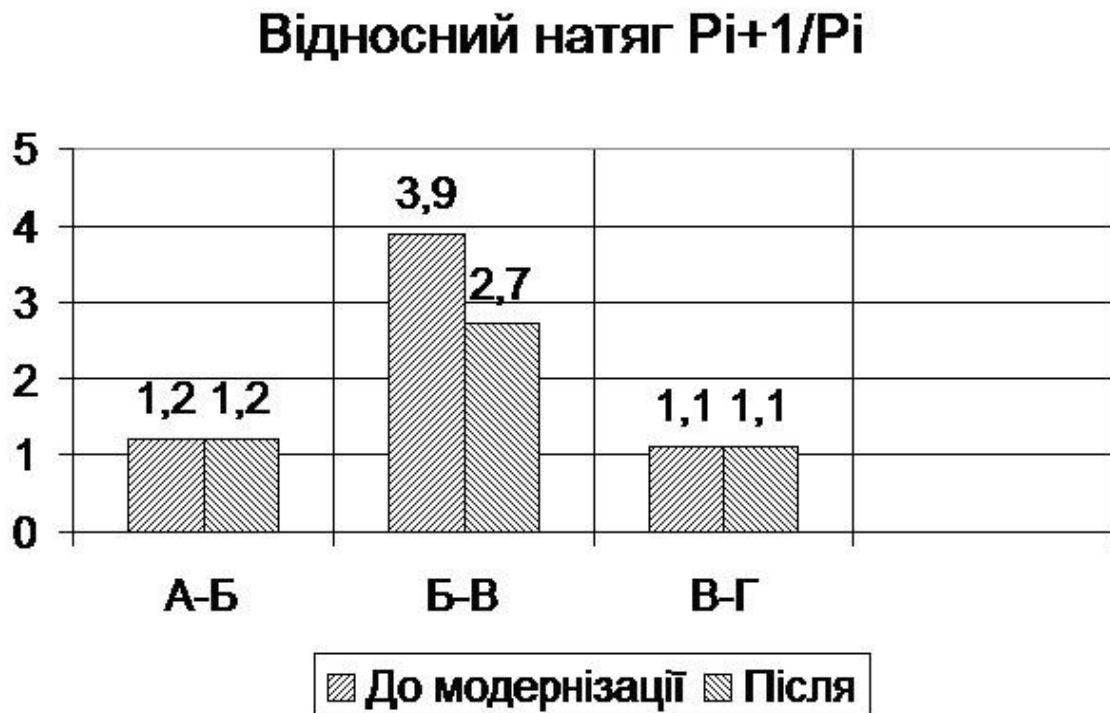


Рис.5.4.17. Діаграма зміни відносного натягу по зонам

З урахуванням (5.4.34) система рівнянь для визначення натягу нитки після зони В матиме вигляд

$$\begin{aligned}
 P_0 = P_B, \quad P_1 = P_B & \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{f \frac{\pi}{2}} - 1 \right) \right] + \\
 + \frac{\mu_1 N}{1 + \sin(\beta - \frac{\pi}{4})} & \left(1 + \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{f \frac{\pi}{2}} - 1 \right) \right] \right). \\
 P = P_B, \quad P_B = P_1 & \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{f \frac{\pi}{2}} - 1 \right) \right] + \\
 + \frac{\mu_1 N}{1 + \sin(\beta - \frac{\pi}{4})} & \left(1 + \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{f \frac{\pi}{2}} - 1 \right) \right] \right).
 \end{aligned}
 \tag{5.4.35}$$



Рис.5.4.18. Загальний вигляд нитконтягувального приладу

Після розрахунків при $N=10$ сН, $P_B=12$ сН, $\mu_1=0,2$, $f=0,24$, $r=0,1$ мм, $R=2$ мм отримаємо значення відносного натягу 3,9.

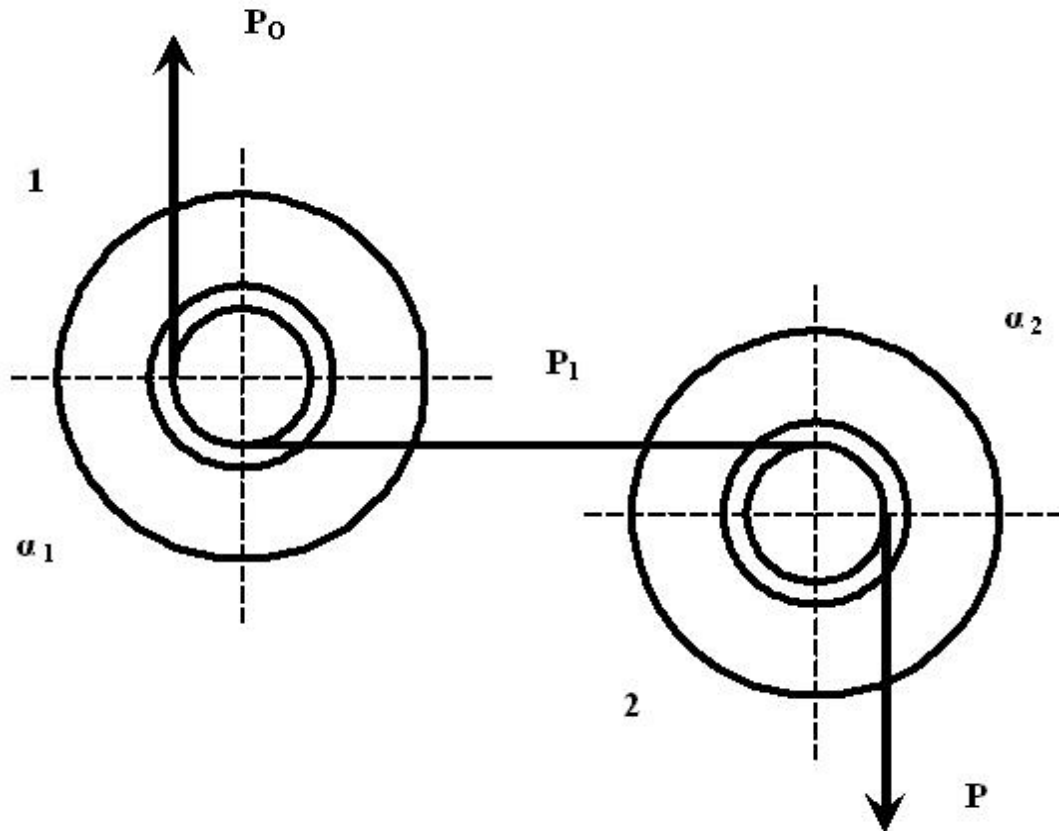


Рис.5.4.19.Розрахункова схема

Використовуючи результати досліджень третього розділу здійснюємо оптимізацію циліндричних нитконатягувачів двозонного нитконатягувача. Для цього визначаємо перші похідні рівнянь системи (5.4.35) та прирівнюємо їх до нуля $\frac{\partial P_1}{\partial R} = 0$, $\frac{\partial P}{\partial R} = 0$. Розв'язуючи отриману систему трансцендентних рівнянь підбираємо відповідні значення радіусів циліндричних напрямних. В результаті цього отримаємо значення відносного натягу 2,7 (див. рис.5.4.17). На підставі даної модернізації удосконалено систему ниткоподачі на мотальній машині М-150-2, що дозволило зменшити натяг нитки в пружній системі заправки на 15-30%.

6. СУЧАСНІ САПР ЛЕГКОЇ ПРОМИСЛОВОСТІ, ЇХ МОЖЛИВОСТІ І СКЛАДОВІ ПІДСИСТЕМИ

Науково-технічний прогрес, пов'язаний з автоматизацією різноманітних етапів проектування, знаходить широке застосування в багатьох галузях промисловості, у тому числі і легкої. Тому при підготовці виробництва необхідно максимально використовувати досягнення сучасної науки і техніки.

Зростання об'єму проектних робіт в умовах частотої зміни моделей особливо гостро ставить задачу скорочення часу та підвищення якості процесу проектування. Також метою автоматизації проектування є, зниження матеріальних затрат, скорочення термінів проектування та підвищення продуктивності праці робітників, що зайняті проектуванням. Для автоматизації проектування характерне систематичне використання ЕОМ при раціональному розподілі функцій між людиною та ЕОМ. На ЕОМ вирішуються задачі, які піддаються формалізації, за умови, що їх машинне рішення більш ефективно ніж ручне.

Створення САПР характеризує нове науково-технічне направлення в проектуванні та визначає більш високий рівень застосування обчислювальної техніки в творчій інтелектуальній діяльності проектувальника. Незважаючи на специфіку проектування об'єктів, для багатьох галузей промисловості основні положення теорії та практики САПР є загальними, наприклад принцип побудови, формування типових задач та програм, організація взаємодії інженера та ЕОМ, підходи до одержання математичних моделей та ін.

Кожна САПР являє собою універсальну систему, яка має своє призначення (проектування і підготовка до розкрою одягу, шкіргалантерейних виробів, головних уборів, взуття, іграшок, тощо). Деякі з них призначені для автоматичної побудови розкрійних схем, щоб повністю замінити ручну побудову комп'ютерною. Для отримання

оптимальної розкрійної схеми достатньо вказати необхідні комплекти лекал і параметри тканини. Основними перевагами таких систем є підвищення ефективності використання тканини (в результаті автоматизованої побудови розкрійних схем процент використання матеріалу збільшується в середньому на 3-5%); багаторазове збільшення швидкості отримання схем розкрою; перевірка підбору оптимальних поєднань комплектів. Системами забезпечується ручна, півавтоматична і автоматична побудова схем розкрою на штучній шкірі або для виводу на папір чи картон, забезпечується висока точність, урахування параметрів тканини, швидкий алгоритм перебору при автоматичній розкладці, зручний і наочний інтерфейс користувача та інше.

САПРи для виробів легкої промисловості дають можливість автоматизувати наступні процеси: конструювання лекал; автоматичне або напівавтоматичне введення геометрії лекал в систему і збереження всієї інформації про лекала в пам'яті комп'ютера; створення архіву інформації про лекала; вибірка по запитам окремих лекал або інформації про них; вивід лекал в графічному вигляді; підготовка до розкрою на полотні за заданими параметрами; створення розкладки в інтерактивному вигляді на екрані монітора; визначення площі лекал і щільності укладки; вивід бажаної розкрійної схеми на друк за заданим масштабом; зберігання отриманих розкрійних схем в базі даних.

Результатом застосування САПР є скорочення витрат на виконання технологічної підготовки виробництва; скорочення часу на підготовку випуску нових моделей або відновлення випуску моделей що випускались раніше; можливість гнучкої заміни асортименту виробів; автоматизація монотонної праці; скорочення часу на ведення лекального господарства; надання нових можливостей для конструктора по створенню нових моделей, економія матеріалів; економія виробничої площі.

Основні підсистеми програмного забезпечення САПР: підсистема

„конструювання” лекал, що здійснює введення лекал в систему або через дигітайзер, або за допомогою графічного редактора; підсистема „розкладки” лекал яка за вихідними даними (параметри матеріалу, модель, розмір партії і т.п.) автоматизує розміщення лекал на матеріалові; підсистема „зображення”, що призначена для виводу отриманої інформації на екран або на графопобудовник (плотер) чи принтер; підсистема „база даних” де зберігається інформація про деталі, отримані схеми, та інша допоміжна інформація що може знадобитися в виробництві.

На сьогодні накопичено значний досвід щодо автоматизації підготовки виробництва взуття та одягу і в області САПР вже існує багато зразків подібних систем.

З 2004 року на ринку країн СНГ успішно просувається компанія Gerber Technology Solutions. Ця компанія пропонує готові рішення часткової та комплексної автоматизації виробництва на підприємствах легкої, меблевої, автомобільної промисловості. Така її продукція як системи AccuMark™, AccuMark Explorer™, AccuPlot™, GERBERdesigner™, GERBERmarker™, Infinity™, LaunchPRO™, Marking System™, MicroJet™, MicroMark™, and Pattern Design™ включає в себе програмне забезпечення для планування виробництва, створення дизайну, САПР для конструювання, моделювання і градації, тривимірного проектування і планування розкрою, системи маркування, фабричного настилу тканин, розкрійні комплекси. Nester server і AccuNest – це ефективні високо інтегровані, прості в використанні програми автоматичної розкладки для систем САПР AccuMark і MicroMark. Вихідна інформація для них може надходити з програм UltraQue, AccuMark Batch Processing, AccuMark Explorer або MicroMark Explorer™.

САПР лекал і розкладок КОМТЕНС призначена для конструкторсько-технологічної підготовки виробництва в різних галузях. Вона ефективно використовується в швейному виробництві,

при виробленні автомобільних сидінь, м'яких меблів, іграшок, шкіргалантереї, виробів з хутра. Передбачено розв'язок спеціальних задач підготовчого розкрійного виробництва: складання технологічної послідовності виробництва, розрахунок норм використання полотен, розрахунок трас розкрою для автоматичних розкрійних установок. Базові конструкції створюються за допомогою тривимірного віртуального манекена, є вбудовані в програмний комплекс типові модулі побудови лекал основних швейних виробів. Модуль розкладки працює в автоматичному і напівавтоматичному режимах та має функцію ущільнення розкладок.

Компанія САПРЛЕГПРОМ займається розробкою та впровадженням САПР та АПУВ для проектування моделей одягу і управління швейним і трикотажним виробництвом. До складу комплексу "Julivi" входять такі програмні модулі: Побудова конструкції, Конструктор, Дизайн, Розкладка, Оптимізатор файлів, Конвертор, Манекен. В цій потужній системі є можливість здійснювати самостійно побудову конструкцій довільної складності. Крім того в базі даних закладено набір базових конструкцій за загальновідомими методиками, розмірні бази даних, нормативні дані. Програмним комплексом передбачено можливість конвертації даних отриманих з інших САПР. Системою передбачено також різні системи укладання лекал в розкладки для різних асортиментів виробів. Отримана інформація може здійснювати вивід інформації на плоттер або автоматизований розкрійний комплекс.

Потужна САПР для легкої промисловості АССОЛЬ має підсистему для шкіргалантерейного виробництва що включає: засоби моделювання виробів в просторі, засоби перегляду і візуальної оцінки моделі в просторі, засоби вимірювання розмірів при моделюванні, засоби автоматичної розгортки поверхонь для створення лекал що дозволяє проектувати вироби без попереднього макетування.

Автоматично складається технічна документація на модель. Підсистема забезпечує ручну, автоматичну і напівавтоматичну розкладку лекал, вивід їх на папір чи картон.

Близько 2 років тому компанії Lectra і Investronica-Sistemas об'єдналися і якщо раніше Lectra розробляла програми для виготовлення лекал та оптимізацію їх розкладок в швейному виробництві то зараз САПРи Lectra дозволяють конструкторам створювати тривимірні зразки майбутніх виробів і складати каталоги моделей. Таким чином шлях від ідеї до кінцевого продукту значно скорочується. Так, САПР DesignConcept3D Advanced Pattern Modeling V2R1 - 3D Design включає потужні тривимірні інструменти для моделювання поверхонь. Для виробників взуття крім розробки конструкції і автоматичного градирування (Romans Cad 2D) запропоновано DIGILAST-3D – електронна, механічна і оптична системи цифрового введення форми взуття, що доповнена CCD камерами. В свою чергу Romans Cad Software 3D Sole – це програма створення і моделювання підошви і каблука в 3- вимірному просторі.

Для оптимізації роботи зі шкірами існує спеціальне програмне забезпечення яке обраховує повне використання кусків. MFC Class - система розкрою шкіри для маленьких і середніх серійних виробництв, дозволяє обробляти шкіру і тканини на одній платформі. Система забезпечує достатню швидкість і точність при високій якості. Розкрій відбувається ножами.

Фірмою накопичено досвід автоматичної побудови розкладок, коли участь оператора потрібна лише для перевірки та внесення коректив в отримані схеми. Модуль розкладки підвищує ефективність розкрийного цеха і зменшити витрати матеріалів на 3-5 %. Програма Optiplan, дозволяє попередньо обрахувати розділ обсягу партії на настил, формування пачок, долю витрат і часу на кожен етап. Через це керування процесом стає більш гнучким що гарантує рівномірне завантаження цехів. Також фірмою

розроблено автоматичні розкрійні машини Vector і програмне забезпечення для керування ними. Інтегрована система керування підприємством «Lectra» поділяється на блоки «Е-Дизайн», «Е-Виробництво» і «Е-Продаж», які в свою чергу можна об'єднати через LectraOnLine. На практиці це означає, що дані можна безпосередньо передавати з одної підсистеми в іншу, тобто створити завершений цикл.

В КНУТД в 1986 році на кафедрі технологій виробництва виробів зі шкіри було створено власний програмний комплекс для автоматизованого проектування моделей взуття „ПРИС-В” що розроблявся як інтегрована підсистема САПР конструкторської підготовки взуттєвого виробництва. Комплекс являє собою модульну структуру з вирішення окремих завдань конструкторської і технологічної підготовки взуттєвого виробництва.

Перший модуль передбачає кілька режимів для введення значень координат точок контурів деталей чи моделей. Функціональні можливості наступного модуля дозволяють модельєру виконувати різні маніпуляції з контурами деталей. Вихідною інформацією для нього є дискретні контури однієї деталі чи ґрундмоделі для середнього розміру. Це відрізняє комплекс від його аналогів, де робота ведеться з окремими контурами чи відрізками ліній, з яких потім формуються деталі. Оскільки програма розроблялась спеціально для проектування деталей взуття то цей модуль використовує для виконання графічних перетворень цілу низку спеціальних функцій, що базуються на математичному описі дій проектувальника при виконанні перетворень контурів деталей моделі. Третій програмний модуль дозволяє проводити градирування контурів деталей, підібрати розмірний асортимент, сформувати паспорт моделі та ін. Також до складу програмного комплексу входять модулі з напівавтоматичним і автоматичним виконанням розрахунку укладності контурів деталей з виведенням їх схем на графічний пристрій або з допомогою принтера при одержанні твердої копії екрану. Контури серії

деталей формуються в файли моделей і можуть виводитись не тільки на графопобудувач, але й на периферійний вирізаючий пристрій.

Завершальним етапом є напівавтоматична розробка технологічного процесу виготовлення взуття, яка використовує бази даних, значно спрощує і скорочує термін підготовки документації. При цьому база при розробці нових видів взуття постійно поповнюється і формується таким чином, що надалі з'являється можливість розробляти схему технологічного процесу в автоматичному режимі

До комплексу ІРИС входить також програмний модуль для перекодування файлів формату DGT (ІРИС) у DXF (Autocad) і назад, що надає можливість прямого й зворотного переходу між середовищами проектування ІРИС і AutoCAD. Така позиція дає змогу помітно розширити можливості проектувальника. При цьому вимоги до апаратних засобів програмного комплексу ІРИС і підготовки користувача залишаються мінімальними. Організована за принципом автономного автоматизованого робочого місця система може бути легко сконфігурована відповідно до спеціальних запитів замовників.

Встановлення та використання описаних вище та подібних до них систем є зручним для великих багато профільованих підприємств легкої промисловості. Але для підприємств малого бізнесу шкіргалантерейної галузі легкої промисловості закупівля таких програм є недоцільною, оскільки одним з недоліків таких систем є їх велика вартість та необхідність закупівлі додаткового дорогого програмного забезпечення та периферійного обладнання для їх якісної роботи. Тому на сьогодні проблема створення САПР для підготовки розкрою з урахуванням потреб та вимог малих підприємств та фабрик шкіргалантерейної промисловості все ще залишається не вирішеною.

Можливості сучасного розкрійного обладнання

Розкрійні схеми отримані як в автоматичному так і в інтерактивному

режимах надалі практично реалізуються на розкрійному обладнанні. Розглянемо лише те розкрійне обладнання яке можливо інтегрувати в системи автоматизованого розкрою.

Преси-автомати що дозволяють кроїти матеріал в автоматизованому режимі. Мають високу продуктивність, достатню точність позиціонування. Фірма «Лекко» пропонує преси для розкрою деталей шкіргалантереї різних модифікацій потужності та розмірів. Наприклад для мілкої шкіргалантереї використовують преси серії Super SF 35/45 потужністю 16, 18 і 20 тон, з робочою площею 800 x 400 або 900 x 450 мм. Їх використання дозволить виконати економну розкладку матеріалу. Мають два показчика вирубки: дотичний – для мілких деталей і точковий – для великих деталей. Преси цієї серії можуть комплектуватись електронними блоками керування. Для розкрою сумок, папок, портфелів користуються більш потужними пресами цієї серії, або ж гідравлічними пресами з рухомим візком, які можуть бути укомплектованими пристроями для проведення допоміжних операцій: автоматичний пошук штамп, переміщення рулонів, автоматичне повертання каретки, автоматичне повертання вирубочної панелі. Крім того гідравлічні преси успішно використовуються при розкрої галантерейних виробів зі штучної шкіри, картону, прокладок і підкладкових матеріалів. Всі преси мають системи безпеки, сертифіковані і є досить простими в експлуатації. Преси-автомати з програмним забезпеченням що дозволяють розкладати шаблони автоматично або в інтерактивному режимі. Мобільні балочні преси фірми АТОМ легко кроять як великі деталі так і мілкі різноманітні форми. Можуть мати додаткові засоби автоматизації.

Також при розкрої рулонних матеріалів широко використовують автоматизовані настільно-розкрійні комплекси ExactTechnologies і Manufactures Supplies Company які складаються з системи автоматизованого проектування розкладок інтегрованої в розкрійну

машину де розкрій здійснюється за допомогою стрічкових або дискових ножів. Розкрій ножем – забезпечує необхідну точність і дозволяє вирізання деталей довільної конфігурації. Траєкторія задається оператором інтерактивно або автоматично через програмне забезпечення. Особливо економія часу відбувається коли під час викроювання на одній частині матеріалу оператор може знімати вже готові деталі або готувати наступу розкладку. Розроблено ряд пакетів програм побудови розкладок деталей для різних секторів ринку товарів (сумки, гаманці, валізи зі шкіри, компоненти взуття, шкіргалантереї і меблів з синтетичних матеріалів і шкіри, паски, шкіряні підшви, правильні геометричні фігури). Модуль маркування Intelliview призначений для визначення дефектних зон і взагалі контурів матеріалу. Він легко інтегрується в модуль розкладок. Цифрова камера сканує контур матеріалу а за допомогою спеціальних пристроїв (оптична миша) і програм виділяють неякісні ділянки. Дані передаються в систему крою, а новий модуль CUTWEB дозволяє використовувати мережеві технології і керувати процесом з віддаленого комп'ютера. Система розкладок передбачає наступні можливості: ручну інтерактивну розкладку; повністю автоматичну розкладку; „оф-лайн”-розкладку, коли розкрійна схема створюється ще до того як матеріал подається на розкрійний стіл; мережеву розкладку коли розкрій ведеться за схемою що обрахована системою розташованою на іншій машині або навіть на іншому підприємстві. Інтерактивна розкладка відбувається безпосередньо на робочому столі. Головні етапи: розташування матеріалу, визначення контуру і маркування дефектів, розміщення деталей (автоматичне, ручне коли оператор самостійно розміщує деталі, напівавтоматичне коли оператор вручну корегує автоматичну розкладку), початок крою. Оператор сам задає програмі в якому режимі проводити розкрій.

Автоматичні розкрійні машини розроблено фірмою «Lectra» починаючи від автоматів для конструктора - Versalis и ProSpin, и

закінчуючи серією промислових розкрійних машин Vector що передбачають найширше використання: Vector Fashion - для базових матеріалів виробництва одягу; Vector Denim - для джинсових матеріалів і робочого одягу; Vector Furniture - для декоративних тканин виробництва меблів; Vector Auto - для тканин, які використовують при оздобленні салонів машин; Vector Technical textiles - для технічного текстилю. Машини цієї серії характерні своїми широкими можливостями (висота настилу 2,5-7см, ширина розкладок 180-220см) вони гарантують високу якість незалежно від матеріалів. Зараз особлива увага приділяється керуючим програмам, тому в стандартний комплект сучасного розкрійного обладнання входить підпрограма Time Management, що надає точну інформацію по процесу розкрою.

Крім того на даний час достатньо широко представлене обладнання країн СНД () таке як: стрічкові ножі: розкрійно-стрічкова машина РЛ-1000-1 Росія, РЛ-630 Росія, Р 12 АО Орша; вертикальні ножі – ЭЗМ-4 Росія, МР-120; МР-150; МР-170 АО Орша; дискові ножі - ЭЗДМ-3 Росія. Це обладнання дещо дешевше.

Технологія розкрою за допомогою лазерного променя має свої особливості через те що різання відбувається за рахунок термічної руйнації. Ефективність використання залежить від товщини настилу, що в свою чергу визначається властивостями матеріалу. Швидкість різання залежить з одного боку від можливостей машини так і з другого боку при прискоренні руху менше змінюється колір кромки матеріалу що кроїться. Коли використовують синтетичні матеріали, в зону різання подається охолоджене повітря щоб виключити спікання чи склеювання шарів настилу.

Система Taurus TLCS – високошвидкісна модульна розкрійна система за допомогою лазера з програмою автоматичного сканування контурів матеріалу, розпізнавання бракованих ділянок, розкладки лекал і

потужних засобів що допомагають розподілити та зняти отримані лекала. Система має потужну систему розпізнавання периметра матеріалу, дефектів на ньому щоб потім автоматично розкласти деталі. Є можливість здійснювати розкладку в інтерактивному режимі. В систему інтегровано комп'ютерний модуль що забезпечує обрахунок потрібної технологічної інформації (час різання, час витрачений на вибраковку та ін.), зберігання її в базі даних та надання в разі потреби користувачам.

Розкрій за допомогою струменя рідини дає можливість швидше і якісніше кроїти (з якістю як у віброножа або маленького дискового ножа), не створює температурних і механічних деформацій, робить тонкі розтини, не забруднює виробництво пилом і може бути застосоване теоретично для будь-якого матеріалу. Такі системи можуть використовуватись при розкрою шкіри, резини, тканин, штучних матеріалів. Можна розрізати настили в кілька шарів. Швидкість різання 7-24 м.хв. В процесі розкрою матеріал не нагрівається, що є надзвичайно вдалим рішенням. Розкрій за допомогою рідини базується на принципі подачі струменя під дуже високим тиском до 4000 бар крізь виключно малі отвори. Системи струменевого різання передбачають різання або струменем чистої води, або ж гідроабразив (коли до води додають полімери). Кількість різальних насадок залежить від моделі машини.

Такі системи як правило є керованими ЕОМ. Розкрій проводиться у відповідності з визначеною програмою коли жиклер рухається за контурами деталі що вирізується. Так працюють системи KMT Waterjet Systems, Alton Box Board Co фірми McCartney Manufacturing's.

Система MDA фірми Digital Control Specialmaschinen пропонує системи розкрою чистою водою. Програмно забезпечені: сканування параметрів розкрою, зсування деталей, мінімізація відстані між деталями, групування деталей, визначення необхідної кількості деталей, обрахунок залишків матеріалу.

На поточний момент Manufactures Supplies Company (ATOM) запропоновано наступні моделі систем розкрою струменем води: ATOMFLASHJET2400 , FLASHJET2000DA , ATOMFLASHJET1600SA (для невеликого виробництва), МОДЕЛЬ FLASHJET 300DA. Всі водоструменеві системи розкрою фірми АТОМ обладнано складним сучасним програмним забезпеченням розкладки деталей, і вони є одними з швидкісних розкрійних систем на сучасному ринку. Інтегровані в систему Cutvision базова CutVision 15/30, CutVision NesStore 34/28). Програми комп'ютеризують: визначення заборонених зон, розкладку і крій. Тут, як правило, проводиться автономна розкладка що має дві стадії: маркування дефектів за допомогою камери і програмного модуля "Intelliview" і розкрій матеріалу з урахуванням отриманих даних. Машини серії FlashCut (АТОМ) поділяють на дві групи: плоскі системи які дозволяють обробку відразу на двох столах (в той час коли машина здійснює крій на одному столі, оператор збирає вже отримані деталі й готує наступний настил матеріалу) і розкрійні системи під нахилом для кращого доступу до матеріалу.

Необхідно зазначити, що зараз є спроби використовувати різальним інструментом для розкрою плазму, розпечений дрiт, принцип електроерозії. Такі новітні методи знаходять своє застосування через певні переваги: гарантують високу якість, зразу обробляють край деталі, виробництво позбавлене пилу, забруднень і токсичних викидів, роблять тонкі розтини і не спричиняють деформацій, не потребують часу на виготовлення й заміну штампів, придатні для отримання деталей довільної форми. Всі ці розкрійні системи повинні мати програмний модуль що забезпечуватиме рух різального інструменту по заданій траєкторії. Але доцільність використання кожного виду обладнання залежить від конкретних умов виробництва.

Перераховане вище розкрійне обладнання потребує вихідної

інформації про розміщення деталей в схемах розкрою. Така інформація отримується під час технологічної підготовки виробництва, тому необхідно вдосконалювати методику технологічної підготовки.

6.1. Апроксимація контурів деталей

Для автоматизованого розкрою за допомогою сучасного розкрійного обладнання треба мати інформацію про зовнішні контури деталей. Але більшість деталей виробів легкої промисловості і зокрема шкіргалантерейного виробництва мають таку форму, що їх зовнішній контур неможливо точно описати аналітично. Тому зовнішні контури деталей апроксимують.

Розглянемо способи апроксимації, при яких отримана інформація про деталі придатна для введення в ЕОМ. Такі як, наприклад, способи аналітичного опису (за допомогою математичних рівнянь) і дискретного опису (за допомогою таблиць) зовнішнього контуру деталі. Проаналізувавши літературні джерела з даного питання в різних галузях промисловості, можемо розділити способи апроксимації на наступні основні групи:

1. Аналітичний опис зовнішнього контуру деталі:
 - а) через многочлени;
 - б) через інші елементарні математичні функції;
2. Дискретний опис зовнішнього контуру.
 - а) Покриття деталі простими геометричними фігурами;
 - б) Апроксимація зовнішнього контуру сукупністю елементарних ліній.

Апроксимацію за допомогою многочленів можна розділити на два види: апроксимацію через сплайни і через поліноми. Для апроксимації через сплайни автори пропонують застосовувати кубічні сплайни і сплайни інших порядків.

При апроксимації через сплайни точність відтворення контурів

деталей може виявитися недостатньою поблизу точок, де $\frac{dx}{dy} = 0$ і функція $y = f(x)$ повільно змінюється. Тому на практиці при виборі точок полінома треба вибирати вузли сітки поблизу екстремумів або поблизу точок перегину. Більше поширення отримали параметричні кубічні сплайни в яких $y = \varphi(t)$ та $x = f(t)$ де $\varphi(t)$ і $f(t)$ є поліномами третьої степені від параметра $t \in [0,1]$

Методи апроксимації зовнішнього контуру через поліноми можна розбити на дві групи:

1) апроксимація через алгебраїчні поліноми, тобто

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

2) апроксимація через тригонометричні поліноми, тобто

$$y = b_0 + \sum_{k=1}^n (b_k \cos kx + c_k \sin kx) - \text{для декартової системи координат}$$

$$\rho = d_0 + \sum_{k=1}^n (d_k \cos kx + l_k \sin kx) - \text{для полярної системи координат.}$$

Апроксимація контурів деталей алгебраїчним поліномами () зводиться до визначення коефіцієнтів $a_0, a_k, k = \overline{1, n}$, застосовуючи методи лінійного програмування, зокрема симплекс-метод. При цьому забезпечується мінімальне відхилення апроксимуючого полінома від заданого координатами контуру.

Апроксимація контурів деталей тригонометричними поліномами () зводиться також до визначення коефіцієнтів $b_0, d_0, b_k, c_k, d_k, l_k, k = \overline{1, n}$ за умови збігу розрахункових і фактичних значень у вузлах інтерполяції. У проміжних точках можливі деякі відхилення розрахункових значень від фактичних. Збільшення числа вузлів інтерполяції забезпечує відтворення реального контуру деталі розрахункової кривої з наперед заданою точністю.

Заслужують на увагу способи апроксимації через інші функції.

Серед них можна виділити спосіб апроксимації за допомогою R – функцій. З його використанням зроблено аналітичний опис деталей різних конфігурацій. Було показано, що окремі ділянки взуттєвих деталей зручно апроксимувати відрізками прямих і дугами кіл. При такій апроксимації спрощується аналітичний опис деталей, що дуже важливо з погляду практичного застосування.

У загальному вигляді алгоритм аналітичного опису деталей через R - функції полягає в наступному:

1. Складний контур, що обмежує область деталі, розбивається на набір послідовних кривих
2. Складається аналітичний опис найпростіших кривих, які порівняно легко описуються аналітично і є R - функціями.
3. Отримані функції замінюються предикатами, складається логічна модель, що забезпечує складного контуру. Отримана у вигляді булевої функції, логічна формула приводиться до диз'юнктивної нормальної форми.
4. Кон'юнкція, диз'юнкція й заперечення замінюються відповідними R - операціями: R - кон'юнкцією, R - диз'юнкцією, R - запереченням.
5. За допомогою рівнянь переходу складається аналітична функція, що описує заданий контур.

Проаналізувавши способи аналітичного опису зовнішнього контуру можна відзначити, що алгоритми опису контурів деталей для розглянутих способів дуже громіздкі, вимагають значних затрат ручної праці, складних математичних обчислень для якісної апроксимації деталей, недостатньо швидкодіючі. Крім того при такій апроксимації ускладнюється рішення задач автоматизації технологічної підготовки виробництва, яке приводить до значних витрат часу на підготовку вихідної інформації до розв'язку поставлених задач. При аналітичному описі контурів деталей складної конфігурації одержують вираз значної складності, однак при їх

розв'язанні виділяють однакові блоки, що значно полегшує обчислення, особливо з застосуванням ЕОМ.

Розглянемо способи дискретного опису деталей взуття, тобто способи, при яких інформація про деталі подається у вигляді масивів чисел.

Деталь можливо апроксимувати шляхом розміщення на її площі простих геометричних фігур. Залежно від конфігурації деталі її можна апроксимувати колами, трикутниками, трапеціями й іншими геометричними фігурами. Контур деталі в цьому випадку визначається як крива що огинає покриваючі фігури. Точність досягається за умови максимального наближення огинаючої до контуру вихідної деталі.

Спосіб апроксимації деталі колами полягає в наступному: деталі розміщуються в нерухомій системі координат таким чином, щоб її поздовжня вісь збігалася з віссю ординат, а поперечна – з віссю абсцис. Потім контур деталі таким чином заповнюється системою кіл що перетинаються, щоб кожне коло дотикалося контуру деталі як мінімум в одній точці. Кількість кіл визначається з умови, що найкоротша відстань від точки перетину двох сусідніх кіл до контуру не повинна перевищувати заданої величини ε . Масив чисел, складений з радіусів покриваючих кіл R_i і координат їхніх центрів $X_i^o, Y_i^o, i = \overline{1, n}$ визначає форму й розміри апроксимованої деталі. (Рис.6.1). Розміщення кіл на площі деталей особливо ефективно, якщо контури деталей мають плавні криволінійні лінії

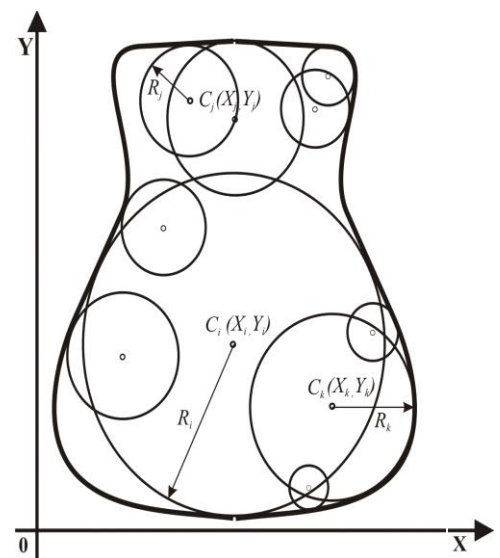


Рис.6.1

При даному способі апроксимації спрощується розв'язок деяких

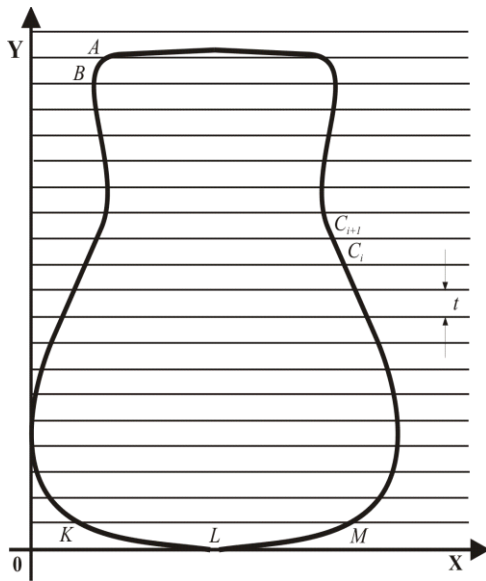


Рис.6.2

технологічних задач підготовки до розкрою, однак істотним недоліком способу покриття колами є трудомісткість підготовки вихідної інформації яка не піддається автоматизації й вимагає більших витрат ручної праці.

Координатно-трапеційний спосіб апроксимації деталей полягає в розміщенні на їхній площі системи трапецій що не перетинаються. Сутність цього способу полягає в тому, що на

площі деталі, вписаної в систему прямокутних координат, розміщують множину паралельних ліній з постійною відстанню між ними. (Рис.6.2). Ці лінії повинні бути паралельні одній з осей координат (наприклад ОХ). При послідовному з'єднанні точок перетину ліній з контуром деталі утворюється множина трапецій. Абсциси точок перетину прямих з контуром і відстань між лініями дають інформацію про зовнішній контур деталі. Зауважимо, що для уникання невизначеності кожна лінія повинна перетинати контур не більше як два рази. Точність апроксимації залежить від відстані між прямими. Найбільша похибка виникає на крайовій ділянці де трапеція перетворюється на трикутник.

Координатно-трапеційний спосіб апроксимації деталей має компактність інформації, а алгоритм розв'язання задач розміщення деталей - достатню швидкодію, але істотним недоліком даного способу є надмірність вихідної інформації при наявності на контурі деталі прямолінійних ділянок і його неуніверсальність, тобто непридатність даного способу до всіх видів деталей.

Одним з методів покриття деталей геометричними фігурами є спосіб опису геометрії деталі за допомогою рецепторних матриць. Область деталі на площині представляють у вигляді прямокутної кістякової матриці (Рис.6.3), елементи якої є булевими змінними і мають значення 0 і 1. Такі матриці називаються рецепторними, при чому:

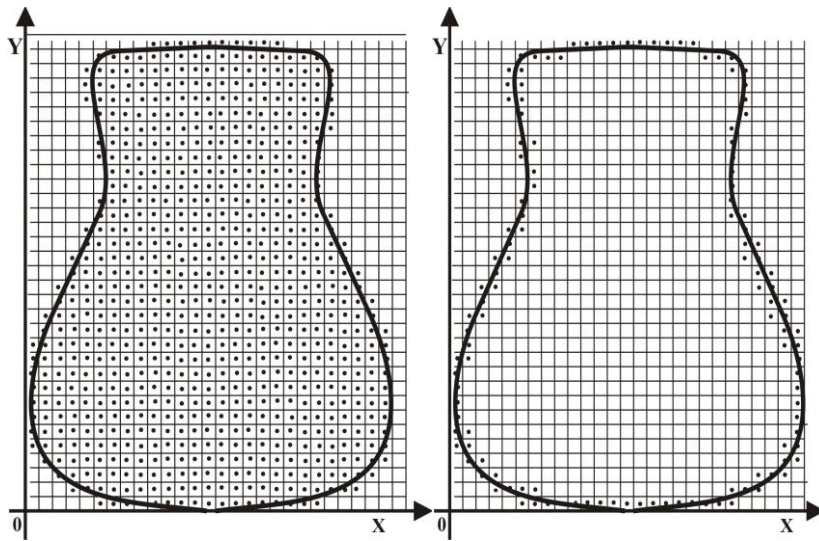


Рис.6.3

$$a_{ij} = \begin{cases} 1 - \text{коли рецептор знаходиться на деталі} \\ 0 - \text{коли рецептор знаходиться зовні деталі} \end{cases}$$

де $i = \overline{1, n}$ $j = \overline{1, m}$

При цьому способі апроксимації спрощується математичне забезпечення для рішення технологічних задач підготовчо-розкрійного виробництва.

До недоліків рецепторного способу апроксимації можна віднести великий обсяг інформації та трудомісткість ручної підготовки інформації.

Одним з найбільш точних ручних кускових способів апроксимації є радіусографічний, основи якого розроблені В.А. Фукінім, Ю.П. Зибінім і О.О. Камриковою. Сутність цього способу полягає в тому, що контур деталі розбивають на елементарні ділянки, які апроксимують відрізками прямих або дугами кіл, що виконується циркульним підбором радіусів дуг. (Рис. 6.4) Підбір радіусів кривизни багато в чому визначається складністю контуру й необхідною точністю. Необхідною умовою знаходження точок сполучення дуг і їхніх центрів є відсутність в даних точках зламу ділянок, що сполучаються (якщо, звичайно, у цій точці контур не має зламу). Це досягається тим, що центри двох суміжних ділянок повинні лежати на одній прямій, яка проходить через точку їхнього сполучення. З одного боку, кількість ділянок має бути максимальною, для досягнення високої точності апроксимації контуру. З іншого боку, число їх бажано мінімізувати для зменшення кількості вихідної інформації, що значно заощаджує машинний час розв'язку задачі. Інформація про контур деталі при радіусографічному способі задається у вигляді масивів координат точок апроксимації X_i, Y_i і масиву радіусів кривизни дуг елементарних ділянок $R_i, i = \overline{1, n}$ (Рис.6.4).

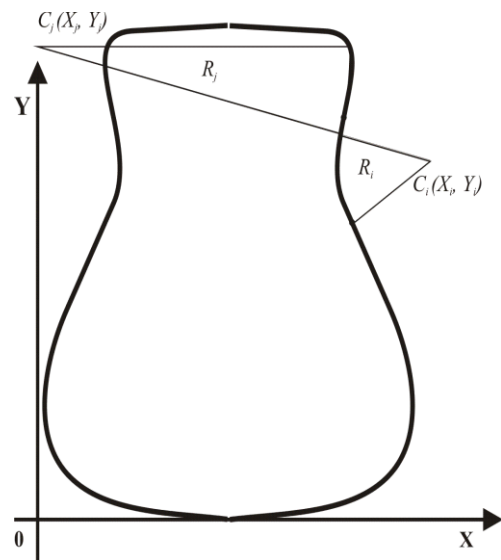


Рис.6.4

До переваг радіусографічного метода можна віднести точність апроксимації, невелику кількість вихідної інформації, але при цьому істотно ускладнюються деякі обчислення пов'язані з перетином контурів. Крім того, неавтоматизована підготовка вихідної інформації при цьому способі апроксимації дуже трудомістка й вимагає значних витрат часу.

Найбільше поширення одержав кусково-лінійний спосіб апроксимації. При цьому способі апроксимації зовнішній контур деталі апроксимується багатокутником. Довжина сторін багатокутника залежить від кривизни контуру та прийнятої погрішності апроксимації. Сторони апроксимуючого багатокутника можуть бути дотичними до контуру деталі, січними, хордами. Проведені дослідження показали доцільність використання такої кусково-лінійної апроксимації при якій сторонами апроксимуючого багатокутника є хорди до зовнішнього контуру деталі (Рис.6.5).

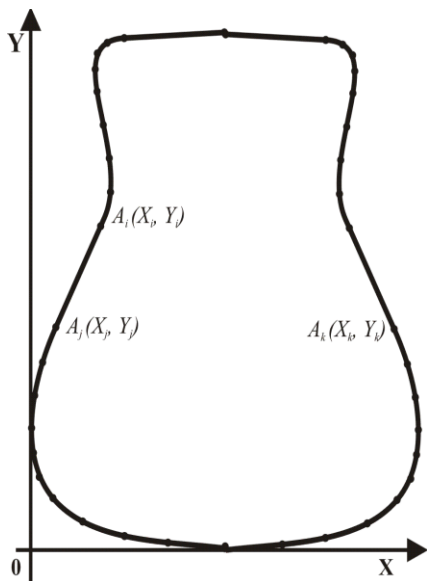


Рис.6.5

Кусково-лінійний спосіб апроксимації є універсальним, тобто придатний до будь-якої форми плоских геометричних об'єктів, не вимагає значних витрат часу при ручному способі апроксимації, легко автоматизується. Крім того інформація при кусково-лінійному способі апроксимації піддається ущільненню, тобто дозволяє виключення зайвих точок без втрати точності апроксимації.

Кожний з розглянутих способів дискретної апроксимації має властиві йому переваги й недоліки. Вибір способу апроксимації для розв'язання поставлених задач будемо робити за наступними критеріями: за точністю відтворення контуру, за простотою введення інформації, за швидкістю підготовки інформації, за можливостями автоматизації вводу-виводу.

Точність відтворення контуру будемо оцінювати при однаковій кількості інформації для кожного способу апроксимації. Очевидно що, за точністю відтворення контуру деталі дані способи апроксимації можна розташувати в наступному порядку: радіусографічний, кусково-лінійний, координатно-трапеційний, покриття колами, рецепторний. За простотою

введення інформації: кусково-лінійний, координатно-трапеційний, рецепторний, радіусографічний, покриття колами. За швидкістю підготовки інформації: кусково-лінійний, координатно-трапеційний, рецепторний, покриття колами, радіусографічний. За можливістю автоматизації вводу-виводу: кусково-лінійний, рецепторний, координатно-трапеційний, радіусографічний, покриття колами.

Отже, кусково-лінійний спосіб апроксимації зовнішнього контуру деталі не накладає обмежень на геометрію деталі і є самим зручним і простим при автоматизованій і неавтоматизованій підготовці інформації. Інформація про контур деталі при кусково-лінійному способі апроксимації легко піддається ущільненню. Цей спосіб апроксимації є одним з найбільш інформаційномістких і універсальних способів апроксимації. Це робить його найпоширенішим при практичному застосуванні, тобто і для розв'язання задач підготовки розкрійних схем.

Огляд робіт з питань раціонального розкрою

Існуючі методи розкрою можна умовно поділити на 3 групи: нормативні, технологічні і оптимізаційні.

Нормативні методи основані на використанні нормативів на відходи що існують на даному підприємстві чи загалом по галузі. Спеціаліст на підставі свого досвіду та вмінь обирає або обчислює розкрійну схему і, якщо вона вкладається в діючий норматив, передає у виробництво. Цей метод при наявності належного досвіду у виконавця може давати задовільні результати, але тут є залежність від „людського фактору”. Крім того цей метод має невисоку продуктивність.

Технологічні методи засновані на використанні чітко заданих технологій. Таким чином отримані раціональні системи розкрою в ситуаціях, що є відмінними від стандартних, можуть не бути оптимальними. Використання обчислювальної техніки в реалізації цих методів лише прискорює роботу, але не покращує значно оптимальність

отриманого рішення.

Оптимізаційні методи основані на використанні математичних методів реалізованих на ЕОМ. Ці методи поділяються на дві групи – чисто оптимізаційні та евристичні.

Задачі розкрою, що визначаються конфігурацією отриманих заготовок поділяють на: задачі фігурного та нефігурного розкрою(). При фігурному розкрою матеріал кроїться на деталі різноманітних конфігурацій. До класу задач нефігурного розкрою відносять задачі лінійного та прямокутного розкрою. В першому випадку матеріал розкроюють на заготовки різної довжини для яких задається лише один лінійний розмір, в другому – отримують заготовки прямокутної форми для яких задано два розміри.

Задачі розкрою що визначаються обсягами продукції також поділяють на два класи: задачі розкрою для умов масового (крупносерійного) виробництва та задачі розкрою в умовах одиничного виробництва. До обох класів можуть належати як задачі фігурного так і нефігурного розкрою.

Більшість оптимізаційних методів використовує лінійні моделі і метод лінійного програмування для їх розв'язку. Задачі розкрою в умовах масового виробництва описують неперервними моделями лінійного програмування, а в умовах одиничного – цілочисельними моделями. Класичні методи лінійного програмування для розв'язку задач розкрою які успішно використовують в умовах масового виробництва важко примінити для одиничного виробництва Але всередині кожного з названих класів, задачі що розглядаються відповідають конкретним технологічним умовам пов'язаним зі специфікою технологічних обмежень, розкрійного обладнання тощо.

Сучасний період розвитку виробничих технологічних процесів характеризується оптимізацією етапів життєвого циклу продукції, що

обумовлено динамічною зміною асортименту та номенклатури при досить жорстких вимогах до собівартості виробів. В цих умовах актуальним є розв'язання оптимізаційних задач розкрою які в умовах одиничного (мілкосерійного) виробництва виникають при виготовленні виробів, як правило, з дорогих матеріалів. При цьому створення раціональних схем розкрою являє собою далеко не тривіальну задачу.

Однак реальні задачі розкрою часто мають нелінійні елементи через що розв'язок не завжди є оптимальним. Евристичні методи є методами побудованими на використанні правил, прийомів, спрощень, узагальнень минулого досвіду. Задачі розкрою, орієнтовані на одиничне виробництво відносять до класу *NP*-складних задач комбінаторної оптимізації, тобто для їх розв'язку немає методів та алгоритмів що знаходять точний розв'язок за поліноміальний час. Точні методи розв'язку цих задач засновані на схемі повного перебору і тому вони є малопридатними для практичного використання. Через це одним з актуальних напрямків досліджень зараз є розробка ефективних наближених і евристичних методів. Серед них виділяють конструктивні методи, в яких розв'язок задачі отримується покомпонентно, додаванням нового компонента до частково отриманого розв'язку доки його не побудують повністю і методи локального пошуку оптимуму. Пошук локально оптимальних розв'язків ведеться в певному наближенні до припустимого розв'язку.

Евристичні методи дають непогані результати якщо їх контролювати через нормативи відходів, але не завжди відомо наскільки оптимальним є той чи інший розв'язок. Таким чином, при евристичному програмуванні розкрою матеріалів на рівні процесів обробки інформації треба виділити самі закони обробки інформації. Розв'язок задач розкрою матеріалів з використанням евристичних методів за допомогою ЕОМ та аналіз отриманих результатів показують, що в прикладному плані ці методи дають задовільні результати.

На практиці найчастіше зустрічаються задачі з ортогональної упаковки і розкрою де в якості малих об'єктів виступають заготовки прямокутної форми або ящики різних розмірів, а в якості крупних об'єктів – матеріал у вигляді рулонів, смуг, прямокутних листів, контейнери різної ємності. На даний час розроблено багато евристичних методів розв'язку даних задач що ускладнює їх вибір та аналіз якості отриманих розв'язків при практичній реалізації в умовах одиничного виробництва.

Оскільки задачі розкрою штучних матеріалів на деталі шкіргалантерейних виробів належать до задач фігурного розкрою, докладніше зупинимось саме на ньому.

Теоретичні дослідження присвячені раціональному розміщенню фігур, зокрема задача найточнішого покриття кривої поверхні плоскими викройками з тканини, належать П.Л.Чебишеву. Також питаннями раціонального використання матеріалів займались в своїх дослідженнях Ю.П. Зыбина, Г.П. Испирян.

В роботах (К. Роджерса и Л.Ф. Тода) отримано важливі результати з точки зору оптимального покриття поверхні набором геометричних фігур. В них надано оцінки щільності укладок і покриття для дво- і тривимірного простору простими фігурами і тілами (кола сфери), приведені класифікація, математична постановка та алгоритми рішення задач нефігурного розкрою.

Методам оптимального розміщення геометричних фігур також були присвячені роботи (Л.В. Канторовича и В.А. Залгаллера) де авторами надавались рекомендації стосовно простих геометричних фігур таких як коло, прямокутник. В роботах, в більшості випадків розглядаються конгруентні прості фігури.

Робота присвячена задачі розміщення плоских конгруентних фігур на площині. Конгруентні правильні шестикутники можна використовувати для заповнення площини таким чином, що в кожній з вершин

сходитимуться три шестикутники. Існують ще способи покриття площини правильними многокутниками: з використанням правильних трикутників – коли в одній вершині сходяться шість трикутників; та з використанням конгруентних квадратів коли в одній вершині сходяться по чотири квадрати. Покриття за допомогою шестикутників є найкращим для укладки вписаних одиничних кіл всередині таких багатокутників. При цьому укладуваність дорівнює $E = \frac{\sqrt{3}}{6} \pi \approx 0,9069$.

Для двовимірного та тривимірного простору авторами отримана оцінка щільності укладок та покриття різними простими фігурами та тілами (кола, сфери). Встановлено, що найщільнішого заповнення площини колами можна досягти „стільниковим” розташуванням.

Математична постановка задач розміщення фігур складної конфігурації значно складніша. Робота Залгаллера, присвячена розгляду багатоекстремальних задач розміщення фігур складної форми.

Розміщення фігур довільної форми на матеріалі заданих розмірів розглядалось в роботах (Г.Я. Машбиц). Задача розміщення на матеріалові фігур довільної форми з урахуванням технологічних обмежень розв'язана.

В роботах припустимі розміщення деталей будуються залежно від вихідних даних про розміри деталей на кожному кроці, а не вважаються заданими. З них обирається оптимальне розміщення.

Спроби математичного моделювання складання схем розкрою натуральних шкір у взуттєвій промисловості не надали бажаних результатів через неоднорідність (анізотропність) властивостей шкіри, наявності вад, а також складності і різноманітності конфігурації як самої шкіри так і деталей. Деталі взуття шкіргалантерейних виробів можуть мати досить складну геометричну форму і тому точно оцінити їх укладаність на площині не вдається. Це обумовило розробку наближених та алгоритмічних методів визначення укладуваності.

В роботі розроблено математичну модель визначення впливу плану розкрою на показник використанні шкіри та отримано певні результати при комбінуванні розміщень деталей різної площі по площині шкіри. Однак при цьому конфігурація самих деталей та матеріалу не враховувалась.

В.Д. Рожок застосував лінійне програмування в оптимізації розкрою шкіри на деталі взуття. Автором була розроблена комплексна економіко-математична модель для основних типів задач що враховували умови розкрою шкіри.

В роботах досліджено наближений метод розрахунку коефіцієнту укладаності деталей взуття за модельною шкалою. Сутність методу полягає в можливості визначення площі паралелограма за параметрами деталей. Але запропонований метод не забезпечує необхідну точність.

Розкрій штучних матеріалів, тканин має свої переваги: однорідність властивостей, правильна геометрична форма, відносна сталість розмірів. Це значно спрощує проведення досліджень.

Аналіз існуючих та розробка раціональних схем розташування деталей було проведено (Б.Л. Елен, Р.М. Вольвич). Авторами було запропоновано спосіб обрахунку показника використання багат шарових настилів з урахуванням всіх відходів, але без урахування розмірного асортименту.

Проектуванню оптимальних решітчастих укладок на ЕОМ присвячено роботи. Ускладнення виникали в формулюванні умов неперетину контурів, але роботи по вектор-функціям щільного розміщення, дозволили створювати математичні моделі і алгоритми розв'язання задач з оптимізації розміщення геометричних об'єктів.

Математичні методи запропоновані в роботі що базувались на координатно-трапеційному способі кодування не дозволяли обрахувати деталі взуття всіх можливих конфігурацій.

Продовженням досліджень з заповнення прямокутної області є робота, в якій запропоновано алгоритм раціонального заповнення прямокутної області двома видами фігур з урахуванням анізотропних властивостей матеріалу, і робота, де проводиться пошук максимального використання прямокутного матеріалу при заповненні його двома видами фігур з можливістю їх повороту відносно один одного на 180^0 .

Роботи з оптимізації обрахунку параметрів розкрійних схем для деталей взуття на штучних матеріалах базувались на математичних моделях в яких деталі апроксимувались колами. Не занадто великі обсяги проміжних обрахунків та зручний алгоритм урахування неперетину деталей дозволяли використовувати для обрахунків ЕОМ того часу.

В роботах розглянуто питання оптимального розміщення деталей взуття за вузлами решітки. Для цього розроблено алгоритм за яким можна реалізувати кілька задач пов'язаних з розміщенням деталей на площині: - визначення параметрів паралелограма максимальної площини для обрахунку укладаності деталей; - визначення параметрів оптимального розміщення одного виду деталей з поворотом на 180^0 на матеріалові прямокутної форми; - вибір з наявного асортименту такого матеріалу, ширина якого забезпечить найбільший коефіцієнт заповнення.

В роботах розроблено алгоритми пошуку оптимальних схем суміщення деталей та побудови раціональних розкрійних схем. Наближене розв'язання отримують перебором області припустимих рішень з постійним кутовим кроком.

В роботі автором розроблено математичну модель і запропоновано алгоритми розв'язання задач з решітчастого розміщення двох видів деталей, орієнтованого розміщення двох видів деталей в заданій області і обрахунку укладаності. Запропоновано класифікацію задач решітчастого розміщення деталей в заданій області залежно від видів деталей (3 класи) і від їх орієнтації відносно матеріалу (5 підкласів).

Впровадження в виробництво ускладнювалось підготовкою вихідної інформації про деталі.

Широке розповсюдження отримав метод побудови найщільніших схем розміщення на площині і напівнескінченній смузі при кусково-лінійній апроксимації зовнішнього контуру деталей.

В наступних роботах описана математична модель задачі з пошуку оптимальних суміщень плоских деталей складної геометричної форми, розглянуто математичні особливості основних структурних компонентів таких оптимізаційних задач, розроблено алгоритми пошуку рішення для контурів апроксимованих кусочно-лінійним способом. А в роботах запропоновано метод розв'язку задачі пошуку мінімального паралелограма, побудованого на векторах \vec{a} , $\vec{q} - \vec{p}$ шляхом перебору локальних мінімумів функції цілі.

Математичні результати досліджень створили теоретичну базу для вирішення прикладних задач пошуку оптимальних схем суміщення деталей взуття та схем розкрою матеріалів.

Графоаналітичний метод визначення раціональної ширини матеріалу для груп деталей, методи підбору раціональних модельних шкал і схем розкрою рулонних матеріалів розроблені в роботах. Також в них розв'язана задача вибору раціонального набору комбінацій деталей взуття з числа можливих з урахуванням ресурсів матеріалу і плану випуску. Але оцінка економічності варіантів комбінування конкретних моделей проводилась без врахування їх особливостей а на базі узагальнених нормативних показників.

Автори розв'язували задачу оптимального розкрою методами лінійного програмування. Вдосконалена модель (см Испирян) передбачала комбінування регулярного і нерегулярного розміщень при побудові варіантів розкрійних схем. При цьому значно збільшувалось число можливих для розгляду розкладок. Також спрощувалось складання

матриці функціональних обмежень в задачі лінійного програмування.

В роботах при побудові розкрійних планів розглядаються прямокутники які являють собою ряди суміщених вручну деталей.

В роботах розроблені економіко-математична модель і алгоритм планування розкрою штучних рулонних матеріалів на деталі шкіргалантереї відносно простої геометричної форми. Планування розкрою базується на обрахуванні обмеженої кількості економічних варіантів розкладок.

Виконано ряд робіт з розробки алгоритмів раціонального суміщення деталей швейних та шкіргалантерейних виробів на рулонних матеріалах. В основі більшості алгоритмів цих робіт лежать алгоритми побудови локальних прямокутних розкладок і потім складання загальної розкладки з отриманих прямокутних розкладок. Автори пропонують різноманітні алгоритми побудови локальних розкладок залежно від конфігурації деталей, технологічних вимог та властивостей матеріалу.

В роботах при проектуванні схем розкрою варіанти схем суміщення деталей будують вручну і вводять в ЕОМ як лінійні параметри. При цьому контури деталей що розміщуються не подаються в явному вигляді. Це є суттєвим недоліком бо не дає можливості візуалізувати отримані результати. Крім того обраховані результати не можуть бути використаними безпосередньо для автоматизованого розкрою. Розглянуті в цих роботах задачі належать до задач нерегулярного розміщення геометричних фігур. В цьому випадкові точне розв'язання методом повного перебору неможливе через значну кількість варіантів схем розкрою. Тому при розв'язку таких задач доцільно говорити лише про знаходження раціональних рішень які тою чи іншою мірою задовольняють поставленим вимогам.

Задачу розкрою рулонних матеріалів в шкіргалантерейній промисловості було запропоновано вирішувати за допомогою економіко-

математичної моделі з булевими змінними. Але запропонована методика може бути успішно використана лише для деталей прямокутної чи близької до неї форми, вихідні схеми суміщення будуються вручну, візуалізація результатів через відсутність інформації про контури деталей практично неможлива.

Математична модель розв'язку задачі побудови схем суміщення деталей взуття в модельних шкалах з аналітичним описом структурних компонент розроблена автором. Розроблено метод пошуку оптимального варіанту схеми суміщення деталей взуття, отримано формули для обрахунку кількості деталей взуття в схемі розкрою при різних вхідних умовах, отримані алгоритми обрахунку параметрів раціональних схем розкрою для системного розташування деталей. Але і математична модель і алгоритм не враховують комплектності (враховують лише один вид деталі і один розмір).

В роботах (Чупринка) запропонована математична модель і комплексний розв'язок задачі автоматизованого проектування множини раціональних схем розкрою листових матеріалів на деталі взуття з урахуванням потреби в деталях.

В роботах досліджено технологічні основи, розроблена математична модель, створено алгоритми рішення задач комп'ютерного проектування комбінованих схем розкрою рулонних матеріалів на деталі взуття з урахуванням комплектності деталей і розмірного асортименту. Запропоновано систему класифікації задач раціонального розміщення деталей взуття в розкрійних схемах.

При тому що можна отримати велику кількість різних варіантів суміщення шаблонів на матеріалі, в економіко-математичну модель лінійного програмування при обрахунках вводиться лише обмежена кількість варіантів які не завжди є найбільш раціональними. Тобто розв'язок задачі отримуємо в умовах неповної інформації. Якщо ж ввести в

модель всі можливі варіанти суміщень (надати повну інформацію) можна отримати оптимальних варіант розкрою.

Аналіз інформаційних джерел показав, що спочатку в дослідженнях з раціонального розкрою обрахункові методи базувались на вхідній інформації про схеми суміщення деталей отриманий вручну, що не давало високої ефективності та якості розробок. Розвиток комп'ютерних технологій і комп'ютеризація виробничої діяльності дозволяють орієнтуватись на розробку більш швидких та якісних алгоритмів пошуку оптимальних або субоптимальних рішень. Саме до таких задач відносять задачі оптимізації проектування комбінованих розкрійних схем рулонних матеріалів з урахуванням технологічних аспектів.

Зараз багато уваги приділяють так званим „еволюційним” алгоритмам. Історія еволюційних обчислень почалася з розробки ряду незалежних моделей. Основними з них були генетичні алгоритми і класифікаційні системи, що були опубліковані на початку 60-х років. У 70-х роках в рамках теорії випадкового пошуку було запропоновано ряд алгоритмів, використовуючих ідеї біонічної поведінки особин. Розвиток цих ідей знайшов віддзеркалення в циклі робіт з еволюційного моделювання. Великий внесок до розвитку еволюційного програмування внесли Фогел (Fogel) і Уолш (Walsh). Не дивлячись на різницю в підходах, кожна з цих "шкіл" взяла за основу ряд принципів, що існують в природі, і спростила їх до такого ступеня, щоб їх можна було реалізувати на комп'ютері. Прикладами таких алгоритмів є метод мурашиних колоній, генетичний алгоритм і нейронні мережі.

Алгоритм мурашки або оптимізація за принципом мурашиної колонії (назва вигадана винахідником алгоритму) може використовуватись для розв'язку як статичних так і динамічних задач. Алгоритм оптимізації мурашиної колонії може бути успішно використаний для розв'язання складних комплексних задач оптимізації мета яких – пошук і визначення

найбільш придатного рішення для оптимізації (знаходження мінімуму чи максимуму) цільової функції (вартості, часу, точності та ін..) з дискретної множини можливих розв'язків.

Наступним напрямком „еволюційного” програмування є генетичні алгоритми. Генетичні алгоритми – методи пошуку що можуть бути використаними для розв'язання задач функціональної оптимізації. Вони засновані на аналогіях з генетичними процесами біологічних організмів: популяції що розвиваються протягом кількох поколінь, підкоряються законам природного відбору за принципом «виживає найбільш пристосований». У відповідності до цього процесу генетичні алгоритми працюють з реальними задачами відповідним чином закодованими. Роботи в цій сфері можна поділити на кілька категорій:

- програми, що реалізують класичний генетичний алгоритм з можливістю настройки параметрів керування основними операторами;
- спеціалізовані програми, що розроблені і оптимізовані для розв'язку вузької чітко визначеної проблеми, наприклад задача оптимізації розкрою;
- чисто наукові дослідження, що полягають в визначенні властивостей і характеристик різновидів самих генетичних алгоритмів.

Відомо два основні шляхи розв'язку задач оптимізації: переборний і локально-градієнтний. У цих методів свої переваги і недоліки, і в кожному конкретному випадку слід подумати, який з них вибрати. Переборний метод простий за своєю суттю і тривіальний в програмуванні. Для пошуку оптимального розв'язку (точки максимуму цільової функції) потрібно послідовно обчислити значення цільової функції у всіх можливих точках, запам'ятовуючи максимальне з них. Недоліком цього методу є велика обчислювальна вартість. Проте, якщо перебір всіх варіантів за розумний час можливий, то знайдене рішення дійсно оптимальне.

Другий популярний спосіб заснований на методі градієнтного спуску. При цьому спочатку вибираються деякі випадкові значення

параметрів, а потім ці значення поступово змінюють, добиваючись найбільшої швидкості росту цільової функції. Досягнувши локального максимуму, такий алгоритм зупиняється, тому для пошуку глобального оптимуму буде потрібно додаткові зусилля. Градієнтні методи працюють дуже швидко, але не гарантують оптимальності знайденого рішення. Вони ідеальні для застосування в так званих унімодальних завданнях, де цільова функція має єдиний локальний максимум (він же - глобальний).

Типове практичне завдання, як правило, мультимодальне і багатовимірне, тобто містить багато параметрів. Для таких завдань не існує жодного універсального методу, який дозволяв би достатньо швидко знайти абсолютно точне рішення. Комбінуючи переборний і градієнтний методи, можна сподіватися отримати хоч би наближене рішення, точність якого зростатиме при збільшенні часу розрахунку. Генетичний алгоритм є саме таким комбінованим методом. Механізми схрещування і мутації в якомусь сенсі реалізують переборну частину методу, а відбір кращих розв'язків - градієнтний спуск. Така комбінація дозволяє забезпечити стійко хорошу ефективність генетичного пошуку для будь-яких типів завдань. Робота алгоритму нагадує випадковий перебір, методом Монте-Карло(). Але, на відміну від методу Монте-Карло, перебір робиться не зовсім випадково, а цілеспрямовано. Завдання алгоритму отримати деяку кількість екземплярів (популяцію) з якнайкращими значеннями критерію оптимізації. Роботу генетичного алгоритму можна розглядати як «виращування» якнайкращої популяції, найбільш пристосованих згідно заданому нами критерію. Для безлічі проблем, генетичні алгоритми дуже часто можуть знаходити хороші рішення (майже оптимальні) приблизно в 100 поколіннях. Це у багато разів швидше чим повний перебір. Наприклад, хромосома, що містить 32 подвійних гена мала б 4,294,967,296 можливих комбінацій при використанні повного перебору. За допомогою генетичного алгоритму така ж проблема буде вирішена перебором 5000

можливих комбінацій.

Генетичні алгоритми можуть з успіхом застосовуватися для широкого класу прикладних задач, включаючи такі, які важко, а іноді і зовсім неможливо, розв'язати іншими методами. В питаннях розкоро матеріалів відомі роботи з застосування генетичного алгоритму при розкоро листових матеріалів, та роботи Мухачевої. З цього можна зробити висновок, що при належному кодуванні особин з урахуванням технологічних особливостей процесу можна застосувати генетичний алгоритм і при побудові розкрійних схем рулонних матеріалів в шкіргалантерейному виробництві. Проте, генетичні алгоритми, як і інші методи еволюційних обчислень, не гарантують виявлення глобального розв'язку за поліноміальний час. Генетичні алгоритми не гарантують того, що глобальний розв'язок буде знайдено, але вони зручні для пошуку "достатньо хорошого" - раціонального розв'язку задачі "достатньо швидко". Головною ж перевагою генетичних алгоритмів є те, що вони можуть застосовуватися навіть на складних задачах, там, де не існує ніяких спеціальних методів. В разі коли задача може бути розв'язана спеціальними методами, то такі методи можуть бути ефективніші за ГА і в швидкодії і в точності знайдених рішень, але навіть там, де добре працюють існуючі методики, можна досягти поліпшення поєднанням їх з генетичними алгоритмами.

Штучні нейронні мережі – це математичні моделі та їх програмні чи апаратні реалізації побудовані за аналогіями організації функціонування мережі нервових клітин організмів. Це спроба моделювання процесів мислення що може „накопичувати досвід”, „навчатись”. Розробка таких моделей спрямована першим чином на розв'язок таких практичних задач як розпізнавання та класифікація текстів, образів, сигналів; прогнозування, адаптивне керування.

Незважаючи на велику зацікавленість до еволюційного моделювання

кількість відомих реально працюючих програм в цій області маловідома.

З аналізу наведених робіт по раціональному розкрою зазначимо що:

- задачі по розробці алгоритмів і програмного забезпечення з побудови раціональних схем розкрою є актуальними оскільки дозволяють раціонально використовувати сировину та заощаджувати робочий час;
- для отримання раціональних розкрійних схем деталей на штучних матеріалах з урахуванням комплектного виходу деталей необхідно при розробці алгоритмів розв'язку враховувати, що розкрійна схема складається з комбінації двох видів розміщення: регулярного і нерегулярного;
- алгоритми і методи, що використовуються для побудови раціональних схем розкрою мають забезпечувати достатньо високу швидкість при отриманні придатних для використання розкрійних схем;
- в результаті обчислення та побудови раціональних схем розкрою має бути отримана множина розкрійних схем що задовольняють технологічним умовам та обмеженням і забезпечують комплектний вихід деталей;
- необхідно забезпечити зберігання параметрів отриманих схем розкрою для можливості виведення їх для перегляду та, в разі потреби, подальшого використання;
- найбільш ефективною є автоматична побудова раціональних розкрійних схем з можливістю подальшого корегування їх в інтерактивному режимі.

6.2. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ З ПІДГОТОВКИ ВХІДНОЇ ІНФОРМАЦІЇ ДЛЯ АВТОМАТИЧНОГО РОЗКРОЮ ДЕТАЛЕЙ ШКІРГАЛАНТЕРЕЙНИХ ВИРОБІВ

Побудова деталей простої конфігурації

Для спрощення проектування деталей шкіргалантерейних виробів пропонується розглядати їх як похідні від фігур простої геометричної форми що отримані шляхом певних конструктивних перетворень. Тому під час проектування моделей для шкіргалантереї спочатку виникає потреба у створенні деталей простої геометричної форми у вигляді: кола, прямокутника, трапеції, трикутника, паралелограма, або ж їх комбінацій. Щоб максимально спростити процес створення такої деталі, координати повинні розраховуватись за мінімальною кількістю параметрів які легко отримати практично, за якими однозначно можна відобразити її контур, Користувач таким чином звільняється від тривалих геометричних розрахунків.

Побудова трикутника

Користувачеві найлегше задавати трикутник за його трьома довжинами сторін. При побудові трикутника передбачається, що він не є виродженим, тобто сума довжин будь-яких двох сторін більше за довжину третьої.

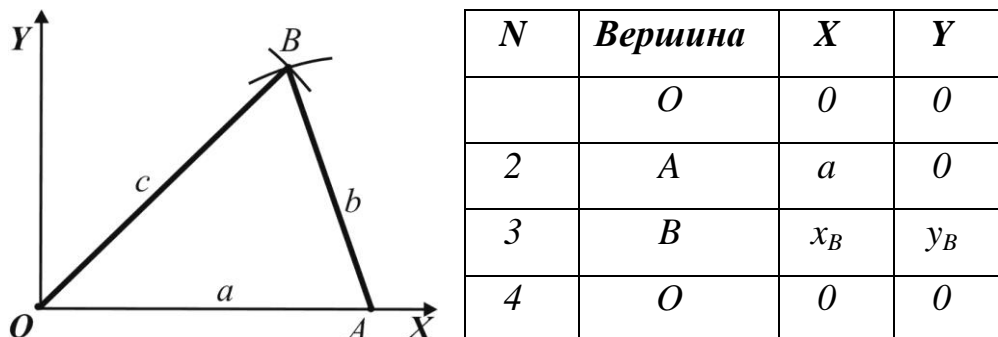


Рис.6.2.1

Позначимо довжини сторін трикутника: a , b , c , і будемо обходити контур по вершинам що пронумеруємо: $1, 2, 3, 4$. ($1=4$). Якщо прийняти, що вершина 1 буде знаходитись на початку координат, можна визначити координати всіх вершин трикутника табл. (рис. 6.2.1).

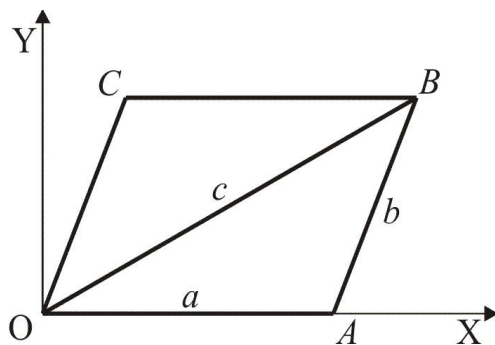
Положення точки $B(x_B, y_B)$ визначимо як точку перетину двох кіл радіусів c і b проведених відповідно з початку координат $O(0,0)$. та точки $A(a,0)$.

$$\begin{cases} x^2 + y^2 = c^2 \\ (x-a)^2 + y^2 = b^2 \end{cases} \Rightarrow \begin{cases} y^2 = c^2 - x^2 \\ x^2 - 2ax + a^2 + c^2 - x^2 = b^2 \end{cases}$$

$$\begin{cases} y^2 = c^2 - x^2 \\ 2ax = a^2 + c^2 - b^2 \end{cases} \Rightarrow \begin{cases} y_B = \pm \sqrt{c^2 - x^2} \\ x_B = \frac{a^2 + c^2 - b^2}{2a} \end{cases}$$

Побудова паралелограма

Побудова паралелограма передбачає, що він не є виродженим та визначається довжинами двох сторін a і b , які виходять з однієї вершини та довжиною діагоналі c , що з'єднує ці сторони. Пронумеруємо вершини : $1, 2, 3, 4, 5$ де вершина 5 співпадає з вершиною 1 .



N	Вершина	X	Y
1	O	0	0
2	A	0	a
3	B	x_B	y_B
4	C	x_C	y_C
5	O	0	0

Рис.6.2.2

Якщо прийняти, що вершина 1 знаходиться в початку координат, то координати всіх вершин паралелограму при обході контура наведемо в таблиці (рис.6.2.2).

Положення точки $B(x_B, y_B)$ визначаємо аналогічно зі знаходженням вершини трикутника як точку перетину двох кіл радіусів c і b проведених відповідно з початку координат $O(0,0)$ та точки $A(a,0)$. Далі згідно з рисунком отримаємо координати точки C :

$$y_C = y_B$$

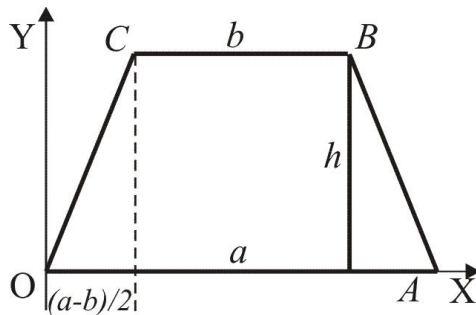
$$x_C = x_B - a$$

Побудова трапеції

Побудова трапеції передбачає, що вона не є виродженою. Рівнобічна трапеція визначається довжинами двох основ та висотою, а в загальному вигляді визначається за довжинами її двох основ, довжиною діагоналі та висотою.

Розглянемо рівнобічну трапецію. Позначимо основи: a, b , висоту - h , а вершини пронумеруємо: 1, 2, 3, 4, 5. Вершина 5 співпадає з вершиною 1.

Вважаючи, що вершина 1 буде знаходитись на початку координат, можна визначити координати всіх вершин трапеції



<i>N</i>	<i>Вершина</i>	<i>X</i>	<i>Y</i>
1	<i>O</i>	0	0
2	<i>A</i>	<i>a</i>	
3	<i>B</i>	x_B	<i>h</i>
4	<i>C</i>	x_C	<i>h</i>
5	<i>O</i>	0	0

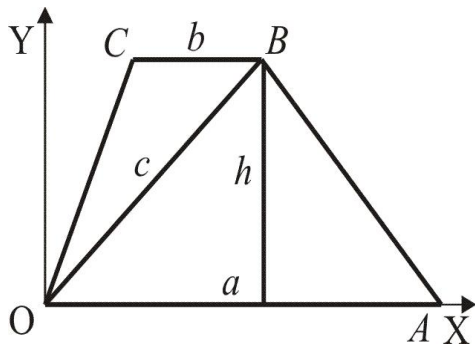
Рис.6.2.3

Координати точок B і C визначаємо як:

$$x_C = \frac{a-b}{2};$$

$$x_B = a - x_C = x_C + b$$

Для трапеції в загальному вигляді позначимо основи: a , b , діагональ – c , висоту – h . Вершини трапеції нумеруємо і наводимо координати в таблиці (рис.6.2.3-4).



N	Вершина	X	Y
1	O	0	0
2	A	a	0
3	B	x_B	h
4	C	x_C	h
5	O	0	0

Рис.6.2.4

Координати вершин B і C визначаємо за формулами:

$$x_B = \sqrt{c^2 - h^2}, \quad y_B = h$$

$$x_C = x_B - b, \quad y_C = h.$$

Побудова N -кутника

Для побудови правильного багатокутника вихідними даними є кількість сторін - N та довжина сторони - a . N – кутник розглядаємо як такий, що вписаний в коло радіуса R (рис.6.2.5).

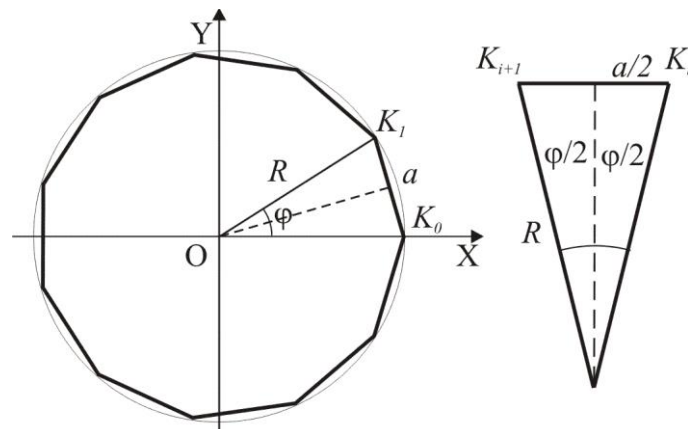


Рис.6.2.5

Координати вершин багатокутника визначаємо як

$$x_{K_i} = R \cdot \cos \varphi_i$$

$$y_{K_i} = R \cdot \sin \varphi_i$$

де $i = 0, 1, \dots, N$ - кількість вершин замкнутого контура. Значення радіуса R та поточного кута φ_i визначаємо за формулами:

$$R = \frac{a}{2 \sin \frac{\varphi}{2}}; \quad \varphi_i = i \cdot \varphi \quad \text{де} \quad \varphi = \frac{2\pi}{N}$$

Побудова кола

Для відображення кола користувачем задається лише його радіус R . Коло розглядатимемо як правильний багатокутник тому побудова відбувається аналогічно до багатокутника.

$$x_i = R \cos \varphi_i$$

$$y_i = R \sin \varphi_i$$

$$\varphi_i = 2\pi i / n, \quad i = 0, 1, \dots, n$$

В цьому випадку кількість вершин залежить від точності апроксимації.

В запропонованому представленні коло являє собою N точок, послідовно з'єднаних між собою (рис.6.2.6).

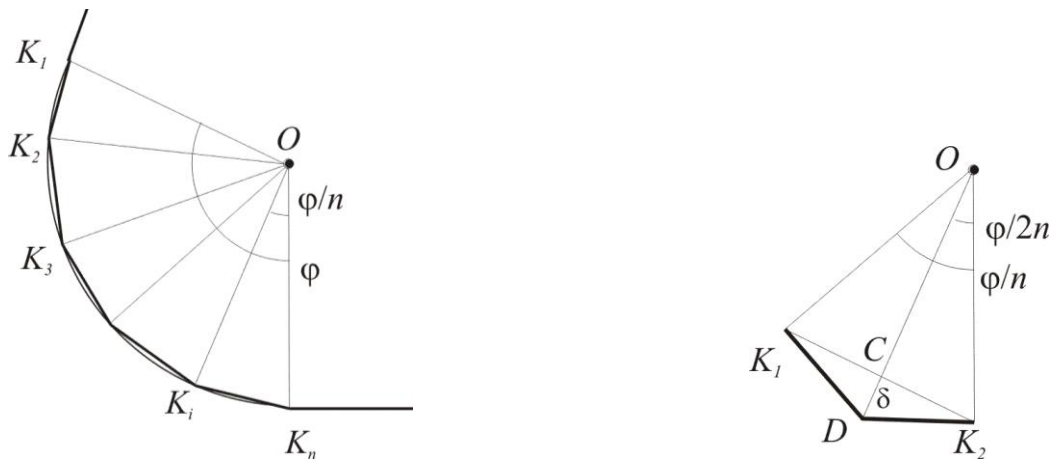


Рис.6.2.6

Знайдемо допустиму похибку при апроксимації δ (рис.....). Допустима похибка при апроксимації δ має бути меншою або дорівнювати заданій користувачем точності:

$$\delta \leq eps$$

$$\left\{ \begin{array}{l} OC = OK_1 * \cos \frac{\varphi}{2N} \\ OD = OK_1 = OK_2 = R \\ \delta = OD - OC = R - OK_1 * \cos \frac{\varphi}{2N} = R - R * \cos \frac{\varphi}{2N} \end{array} \right.$$

З виразів () та () випливає що,

$$eps \geq R - R * \cos \left(\frac{\varphi}{2N} \right)$$

Перетворивши вираз () отримаємо формулу для знаходження кількості точок у дузі заокруглення:

$$N \geq \frac{\varphi}{2 \arccos \left(\frac{R - eps}{R} \right)}$$

Побудова еліпса

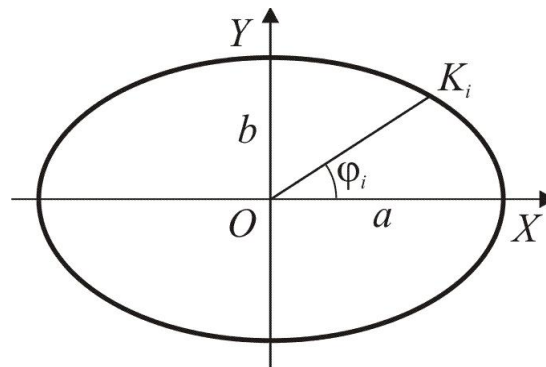


Рис.6.2.7

Еліпс будемо апроксимувати як багатокутник. Для побудови еліпса користувачем задаються його напівосі a та b . (рис.6.2.7). Побудова відбувається за формулами ()

$$\begin{aligned} x_i &= a \cos \varphi_i \\ y_i &= b \sin \varphi_i \\ \varphi_i &= 2\pi i / n, \quad i = 0, 1, \dots, n \end{aligned}$$

Побудова фігури прямокутник-півколо

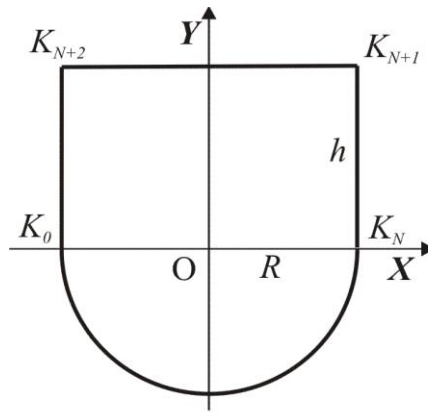


Рис.6.2.8

Для побудови фігури задають радіус півкола R та висоту прямокутної частини h .

Обхід контуру здійснюється проти годинникової стрілки (рис.6.2.8).

Спочатку будується півколо за формулами:

$$\begin{aligned} x_i &= R \cos \varphi_i \\ y_i &= R \sin \varphi_i \end{aligned}$$

де $\varphi_i = (i \cdot \varphi + \pi)$, $i = 0, 1, \dots, N$, $\varphi = \frac{\pi}{N}$.

Далі прямокутна частина має такі координати:

$$\begin{aligned} x_{K_{N+1}} &= R; & y_{K_{N+1}} &= h \\ x_{K_{N+2}} &= -R; & y_{K_{N+2}} &= h \\ x_{K_{N+3}} &= x_{K_0}; & y_{K_{N+3}} &= y_{K_0} = 0 \end{aligned}$$

Побудова фігури прямокутник-півеліпс

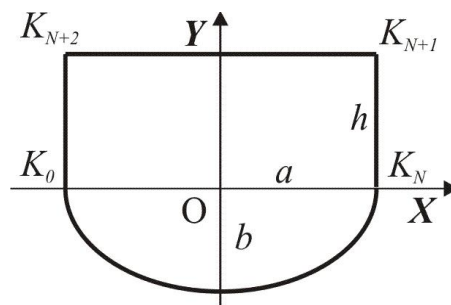


Рис.6.2.9

Для побудови фігури задають величини напівосей еліпса a, b та висоту прямокутної частини – h . Далі за аналогією з попередньою фігурою обходимо контур (рис.6.2.9).

$$x_i = a \cos \varphi_i$$

$$y_i = b \sin \varphi_i$$

де $\varphi_i = (i \cdot \varphi + \pi)$, $i = 0, 1, \dots, N$, $\varphi = \frac{\pi}{N}$.

Координати наступних точок :

$$x_{K_{N+1}} = a; \quad y_{K_{N+1}} = h$$

$$x_{K_{N+2}} = -a; \quad y_{K_{N+2}} = h$$

$$x_{K_{N+3}} = x_{K_0}; \quad y_{K_{N+3}} = y_{K_0} = 0$$

Всі наведені вище алгоритми побудови деталей реалізовано в програмний продукт наведений в Додатку А.

Оскільки не всі деталі шкіргалантерейних виробів можуть задаватись через геометричні примітиви, то виникає потреба в моделюванні отриманих фігур. Під час моделювання будь-якої деталі виникає потреба в додаванні, вилученні чи зміні координат вершин контуру, заокругленні кутів контурів і таке інше. Розглянемо деякі з алгоритмів.

Математична модель та алгоритм вилучення та додавання точки на відрізок контура

В разі вилучення точки з контура задача складається з таких етапів:

- 1) вибір вершини на контурі, яку планується видалити;
- 2) знаходження суміжних до даної точки вершин;
- 3) видалення обраної вершини та відрізків, що виходять з неї;
- 4) з'єднання між собою вершин, що залишились;
- 5) занесення змін в файл з інформацією про деталь.

При зміні координат обраної точки позиції 1) і 3) не зміняться, а далі:

- 4) визначити нові координати обраної точки;
- 5) провести відрізки до неї з суміжних вершин;
- 6) внести зміни в файл.

Вибір вершини для подальших перетворень

Контур деталі представляє собою опукло-ввігнутий багатокутник з вершинами $A_i(x_i, y_i)$, $i=1..n$. Потрібну нам вершину позначаємо курсором. Курсор знаходиться в точці $O(x_0, y_0)$. Нам необхідно обрати з існуючих вершин лише ту вершину, яка найменше віддалена від точки $O(x_0, y_0)$ (рис.6.2.10).

Для ідентифікації потрібної вершини виконаємо наступні дії:

- 1) знайдемо відстані D_i від точки $O(x_0, y_0)$ до вершин $A_i(x_i, y_i)$, $i=1..n$:

$$D_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} ;$$

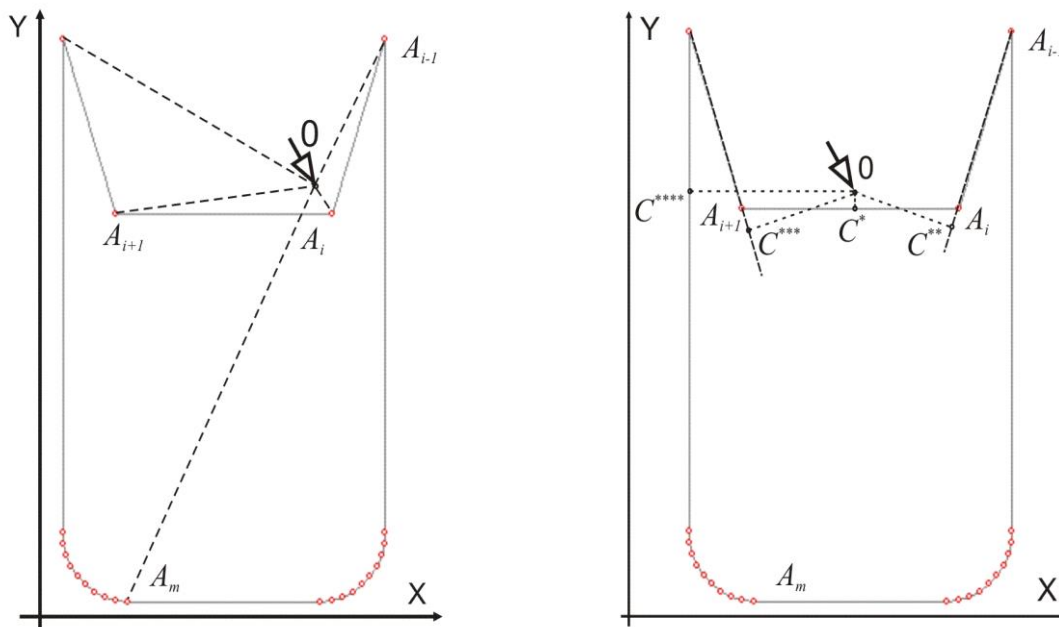


Рис.6.2.10

- 2) визначимо вершину, для якої $D_i = \min\{D_i\}, i=1..n$;

Додавання нової точки в контур

Додавання нової точки відбувається наступними етапами:

- 1) обираємо відрізок контура на якому треба додати вершину;
- 2) визначаємо точки - вершини відрізка;
- 3) розташовуємо точку на відрізкові;
- 4) вносимо зміни в файл.

Нехай нам необхідно розмістити нову точку $C^*(x_c^*, y_c^*)$ між вершинами $A_i(x_i, y_i)$ та $A_{i+1}(x_{i+1}, y_{i+1})$. (рис.). Розташування точки обираємо позицією курсора. Розглядатимемо курсор як точку $O(x_o, y_o)$. Ідентифікація відрізка відбувається коли проекція точки $O(x_o, y_o)$ на пряму що проходить через точки A_i та A_{i+1} , буде належати цьому відрізкові $A_i A_{i+1}$ і відстань OC^* буде мінімальною.

Можливі такі випадки розташування:

а) Відрізок паралельний до осі Y, тобто $x_i = x_{i+1}$. Тоді $x_{c^*} = x_i$, $y_{c^*} = y_o$. В тому разі коли $y_c = y_o \in [y_i, y_{i+1}]$ знаходимо відстань $CO = |x_o - x_i|$;

б) Відрізок паралельний осі X, тобто $y_i = y_{i+1}$. В цьому разі $y_{c^*} = y_i$, $x_{c^*} = x_o$.

Якщо $x_c = x_o \in [x_i, x_{i+1}]$ то відстань визначимо як $CO = |y_o - y_i|$;

в) В загальному випадкові коли $x_i \neq x_{i+1}$, $y_i \neq y_{i+1}$ проекцію точки на відрізок визначаємо за формулами () що використовуються при спряженні кутів. В тому разі коли виконується умова $x_c \in [x_i, x_{i+1}]$ та $y_c \in [y_i, y_{i+1}]$ і проекція належить відрізку, то відстань визначаємо за формулою $CO = \left| \sqrt{(x_o - x_c)^2 + (y_o - y_c)^2} \right|$.

Обрахунки відбуваються в екранних координатах і лише при занесенні інформації в файл відбувається перехід до реальних координат.

Математична модель та алгоритм спряження кута дугою заданого радіуса

Серед завдань моделювання найбільшу складність представляє спряження кута деталі, яке передбачає визначення кількості та координат нових вершин деталі.

Спряження кута відбувається наступними етапами:

- 1) задаємо потрібний радіус дуги,
- 2) визначаємо координати точки - центру дуги заокруглення,
- 3) обчислюємо кутову величину дуги,
- 4) знаходимо точки дотику дуги до сторін кута,
- 5) визначаємо необхідну кількість та координати точок, що утворюють дугу.

Нехай маємо кут, що задається трьома точками $A(x_a, y_a)$, $B(x_b, y_b)$, $C(x_c, y_c)$ (рис.6.2.11) тоді сторони AB та AC задані відповідно рівняннями прямих $A_1x + B_1y + C = 0$ та $A_2x + B_2y + C = 0$ і бажаний радіус дуги спряження R , де коефіцієнти рівнянь прямих A_1, A_2, B_1, B_2, C_1 та C_2 отримуємо з рівняння прямої що проходить через дві точки. Точками є вершини контуру які утворюють даний кут.

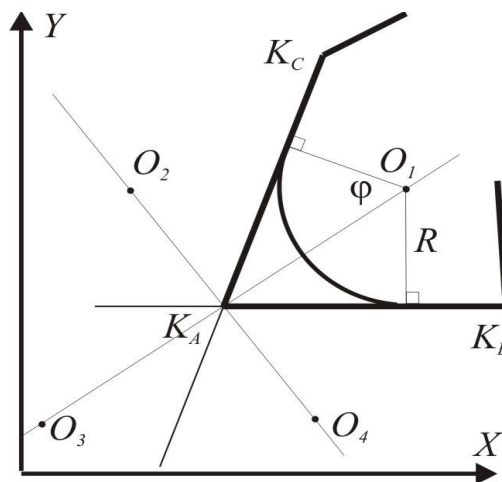


Рис.6.2.11

$$x(y_c - y_a) + y(x_a - y_c) + x_c y_a - x_a y_c = 0 \Rightarrow$$

$$A_1 = y_C - y_B, \quad B_1 = x_A - x_C, \quad C_1 = x_C y_A - x_A y_C$$

$$x(y_B - y_A) + y(x_A - x_B) + x_B y_A - x_A y_B = 0 \Rightarrow$$

$$A_2 = y_B - y_A, \quad B_2 = x_A - x_B, \quad C_2 = x_B y_A - x_A y_B$$

Координати точок O_{1-4} рівновіддалених від даних прямих на заданий радіус можна визначити розв'язанням системи рівнянь:

$$\begin{cases} R = \frac{A_1 x_O + B_1 y_O + C_1}{\sqrt{A_1^2 + B_1^2}} \\ R = \frac{A_2 x_O + B_2 y_O + C_2}{\sqrt{A_2^2 + B_2^2}} \end{cases}$$

Провівши необхідні перетворення отримаємо:

$$\begin{cases} x_{O_1} = \frac{B_1(R\sqrt{A_2^2 + B_2^2} - C_2) - B_2(R\sqrt{A_1^2 + B_1^2} - C_1)}{A_2 B_1 - A_1 B_2} \\ y_{O_1} = \frac{R\sqrt{A_1^2 + B_1^2} - C_1 - A_1 x_{O_1}}{B_1} \end{cases}$$

Залежно від розташування кута що заокруглюємо обираємо додатні чи від'ємні значення коренів в виразах.

Для точки O_1 :

$$\begin{cases} x_{O_1} = \frac{B_1(R\sqrt{A_2^2 + B_2^2} - C_2) - B_2(R\sqrt{A_1^2 + B_1^2} - C_1)}{A_2 B_1 - A_1 B_2} \\ y_{O_1} = \frac{R\sqrt{A_1^2 + B_1^2} - C_1 - A_1 x_{O_1}}{B_1} \end{cases}$$

Для точки O_2 :

$$\begin{cases} x_{O_2} = \frac{B_1(-R\sqrt{A_2^2 + B_2^2} - C_2) - B_2(R\sqrt{A_1^2 + B_1^2} - C_1)}{A_2 B_1 - A_1 B_2} \\ y_{O_2} = \frac{R\sqrt{A_1^2 + B_1^2} - C_1 - A_1 x_{O_2}}{B_1} \end{cases}$$

Для точки O_3 :

$$\begin{cases} x_{O_3} = \frac{B_1(R\sqrt{A_2^2 + B_2^2} - C_2) - B_2(-R\sqrt{A_1^2 + B_1^2} - C_1)}{A_2 B_1 - A_1 B_2} \\ y_{O_3} = \frac{-R\sqrt{A_1^2 + B_1^2} - C_1 - A_1 x_{O_3}}{B_1} \end{cases}$$

Для точки O_4 :

$$\begin{cases} x_{O_4} = \frac{B_1(-R\sqrt{A_2^2 + B_2^2} - C_2) - B_2(-R\sqrt{A_1^2 + B_1^2} - C_1)}{A_2B_1 - A_1B_2} \\ y_{O_4} = \frac{-R\sqrt{A_1^2 + B_1^2} - C_1 - A_1x_{O_4}}{B_1} \end{cases}$$

Критерієм вибору потрібного центра спряження є сума відстаней від точки O_j до точок контура що передують та наслідують вершину кута:

$$\Delta_j = |O_jK_C| + |O_jK_B|$$

де $j=1,2,3,4$. Обираємо ту точку для якої Δ_j буде мінімальною.

Величину кута φ знаходимо як $\varphi = 2\varphi'$, а $\cos\varphi' = \frac{R}{K_AO}$ де K_AO -

відстань від вершини кута до центра дуги заокруглення

$$K_AO = \sqrt{(x_{K_A} - x_O)^2 + (y_{K_A} - y_O)^2}$$

$$\cos\varphi' = \frac{R}{\sqrt{(x_{K_A} - x_O)^2 + (y_{K_A} - y_O)^2}}$$

Точки дотику радіуса дуги до сторін кута визначаємо наступним чином. Нехай рівняння сторони кута має вигляд $\frac{x-x_i}{x_{i+1}-x_i} = \frac{y-y_i}{y_{i+1}-y_i}$ де x_i, y_i

та x_{i+1}, y_{i+1} - координати точок відповідно A_i та A_{i+1} (На рис. це точки K_C і K_A). Цю ж пряму можна представити рівнянням вигляду $Ax + By + C = 0$ де $A = y_{i+1} - y_i$, $B = x_i - x_{i+1}$, $C = y_i x_{i+1} - x_i y_{i+1}$ або $y = kx + b$ х чи інакше

$y = -\frac{A}{B}x - \frac{C}{B}$. Таким чином можемо визначити значення коефіцієнтів

$$k = -\frac{A}{B} = \frac{y_i - y_{i+1}}{x_i - x_{i+1}} \quad b = \frac{x_i y_{i+1} - y_i x_{i+1}}{x_i - x_{i+1}}$$

Пряма OC^* перпендикулярна до сторони A_iA_{i+1} має рівняння $y - y_O = k_1(x - x_O)$ де $k \cdot k_1 = -1 \Rightarrow k_1 = -\frac{1}{k}$ тому приймаємо $y - y_O = -\frac{x - x_O}{k}$.

Для знаходження координат точки перетину двох прямих

розв'яжемо систему рівнянь:
$$\begin{cases} y - y_O = \frac{x - x_O}{k} \\ y = kx + b \end{cases}$$

Провівши необхідні перетворення отримаємо

$$\begin{cases} x_{C^*} = \frac{x_O + k \cdot y_O - k \cdot b}{k^2 + 1} \\ y_{C^*} = k \cdot x_C + b \end{cases}$$

Окремо розглянемо випадки коли сторона до якої проводиться перпендикуляр паралельна одній з осей координат.

1) Відрізок $A_i A_{i+1}$ паралельний до осі OY , тобто $x_{A_i} = x_{A_{i+1}}$

тоді точка дотику радіуса до даної сторони матиме координати:

$$\begin{cases} x_{C^*} = x_{A_i} = x_{A_{i+1}} \\ y_{C^*} = y_O \end{cases}$$

2) Відрізок $A_i A_{i+1}$ паралельний до осі OX , тобто $y_{A_i} = y_{A_{i+1}}$

тоді точка дотику радіуса до даної сторони матиме координати:

$$\begin{cases} x_{C^*} = x_O \\ y_{C^*} = y_{A_i} = y_{A_{i+1}} \end{cases}$$

Кількість точок в дузі визначаємо за аналогією з побудовою кола або еліпса залежно від потрібної точності апроксимації. Для того, щоб знайти координати точок в дузі заокруглення, використаємо лінійне перетворення точки з вихідними координатами (в нашому випадку перетворення точки K_1 для отримання координат точок K_2, K_3, \dots, K_N).

Щоб отримати координати точок K_2, K_3, \dots, K_N , необхідно повернути точку K_1 навколо точки O на кут $\frac{i\varphi}{N}$.

Необхідні для повороту формули отримаємо в результаті композиції матриць паралельного переносу, повороту навколо початку координат і оберненої до матриці паралельного переносу:

$$R_{(x,y)}^\alpha = T_{(x,y)} \circ R_{(0,0)}^\alpha \circ T_{(x,y)}^{-1}$$

Тоді з отримаємо формулу для знаходження координат i -ї вершини деталі (n – кількість точок в дузі заокруглення, φ – кут між векторами OK_1 і OK_N):

$$\begin{cases} x_i = \cos\left(\frac{i\varphi}{n}\right) - \sin\left(\frac{i\varphi}{n}\right) - x_o * \cos\left(\frac{i\varphi}{n}\right) + y_o * \cos\left(\frac{i\varphi}{n}\right) + x_o \\ y_i = \sin\left(\frac{i\varphi}{n}\right) + \cos\left(\frac{i\varphi}{n}\right) - x_o * \sin\left(\frac{i\varphi}{n}\right) - y_o * \cos\left(\frac{i\varphi}{n}\right) + y_o \end{cases}$$

Розміщення гофрів та виточок

При розміщенні виточок або гофрів на деталі задача зводиться до знаходження точок перетину кола з прямою, що проходить через його центр.

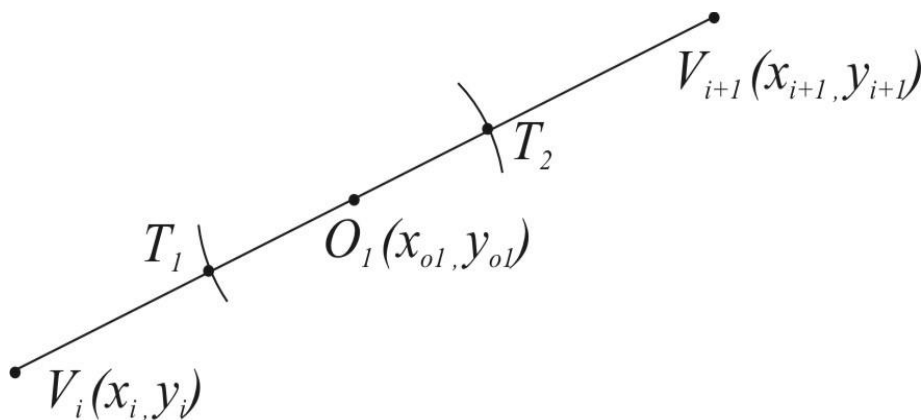


Рис.6.2.12

Розглянемо задачу знаходження точок перетину кола з центром в точці \$O_1(x_{o1}, y_{o1})\$ радіусом \$R_1\$ з прямою що проходить через точки \$V_i(x_i, y_i)\$ та \$V_{i+1}(x_{i+1}, y_{i+1})\$. Знаходимо координати точок \$T_1\$ та \$T_2\$ - коренів системи рівнянь прямої та кола

$$\begin{cases} (x - x_{o1})^2 + (y - y_{o1})^2 = R_1^2 \\ \frac{x - x_i}{x_{i+1} - x_i} = \frac{y - y_i}{y_{i+1} - y_i} \Rightarrow Ax + By + C = 0 \end{cases}$$

де \$A = y_{i+1} - y_i\$, \$B = x_i - x_{i+1}\$, \$C = y_i x_{i+1} - x_i y_{i+1}\$

Можливі 3 випадки:

а) коли \$A=0\$ тоді рівняння прямої набуде вигляду \$y = -\frac{C}{B}\$

$$(x - x_{o1})^2 + \left(-\frac{C}{B} - y_{o1}\right)^2 = R_1^2$$

після перетворень отримаємо вираз: $x^2 + B_1x + C_1 = 0$ де

$$B_1 = -2x_{o1}; C_1 = x_{o1}^2 + \left(\frac{C}{B} + y_{o1}\right)^2 - R_1^2;$$

корені рівняння :

$$x_{1,2} = -\frac{B_1}{2} \pm \sqrt{\frac{B_1^2}{4} - C_1} \quad \text{та} \quad y_{1,2} = -\frac{C}{B} \quad \text{будуть координатами точок } T_1(x_1, y_1) \text{ та } T_2(x_2, y_2)$$

б) коли $B=0$ маємо вираз $Ax + C = 0$ звідки $x = -\frac{C}{A}$, тоді

$$\left(-\frac{C}{A} - x_{o1}\right)^2 + (y - y_{o1})^2 = R_1^2$$

після перетворень отримаємо вираз: $y^2 + B_2y + C_2 = 0$ де

$$B_2 = -2y_{o1}; C_2 = y_{o1}^2 + \left(\frac{C}{A} + x_{o1}\right)^2 - R_1^2;$$

маємо корені рівняння :

$$y_{1,2} = -\frac{B_2}{2} \pm \sqrt{\frac{B_2^2}{4} - C_2} \quad x_{1,2} = -\frac{C}{A} \quad \text{що є координати точок } T_1(x_1, y_1) \text{ та } T_2(x_2, y_2)$$

в) у загальному випадку коли $A \neq 0$ та $B \neq 0$ розглядаємо систему

$$\begin{cases} Ax + By + C = 0 \\ (x - x_{o1})^2 + (y - y_{o1})^2 = R_1^2 \end{cases}$$

виразимо $x = -\frac{B}{A}y - \frac{C}{A}$ тоді

$$\left(-\frac{B}{A}y - \frac{C}{A} - x_{o1}\right)^2 + (y - y_{o1})^2 = R_1^2$$

позначимо через $z = -\frac{B}{A}$, та $q = -\frac{C}{A} - x_{o1}$ тоді

$$(zy + q)^2 + y^2 - 2yy_{o1} + y_{o1}^2 = R_1^2$$

$$(z^2 + 1)y^2 + (2zq - 2y_{o1})y + q^2 + y_{o1}^2 - R_1^2 = 0$$

приведемо рівняння до вигляду: $A_3y^2 + B_3y + C_3 = 0$ де

$$A_3 = z^2 + 1, B_3 = 2zq - 2y_{o1}, C_3 = q^2 + y_{o1}^2 - R_1^2$$

приймемо $D = B_3^2 - 4A_3 \cdot C_3$ тоді

$$y_{1,2} = \frac{-B_3 \pm \sqrt{D}}{2A_3} \quad x_{1,2} = -\frac{B}{A} y_{1,2} - \frac{C}{A}$$

Радіусом кола R_l при розрахунках виточок є величина $\Delta/2$, де Δ - ширина виточки.

При побудові вершини виточки (точка D на рис.6.2.13) розглядається задача побудови перпендикуляра до відрізка, а вже потім на ньому відсікається необхідна глибина виточки H як перетин кола й прямої за вже відомим алгоритмом.

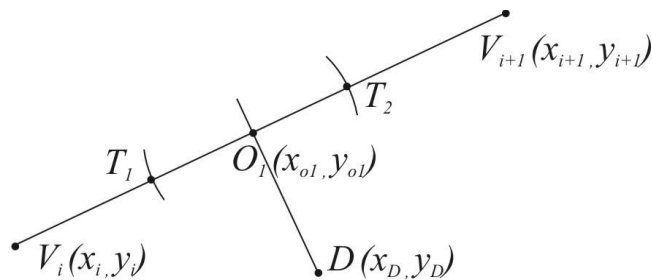


Рис.6.2.13

При побудові перпендикуляра до прямої $V_i V_{i+1}$ яка задана рівнянням виду $Ax + By + C = 0$ в точці $O_l(x_{ol}, y_{ol})$ що належить даній прямій можливі 3 випадки:

а) $A=0$;

в цьому разі рівняння прямої матиме вигляд $By + C = 0 \Rightarrow y = -\frac{C}{B}$

тоді координати точки $D(x_{ol}, y_{ol} \pm H)$

б) $B=0$;

тоді $Ax + C = 0 \Rightarrow x = -\frac{C}{A}$ і тоді координати точки $D(x_{ol} \pm H, y_{ol})$

в) В загальному випадку коли $A \neq 0$ та $B \neq 0$, рівняння довільної прямої що проходить через точку O_l має вигляд $x - x_{ol} = k(y - y_{ol})$. Ми маємо рівняння прямої $V_i V_{i+1}$ вигляду $Ax + By + C = 0$ або $y = -\frac{A}{B}x - \frac{C}{B}$, тобто

$k = -\frac{A}{B}$. Як відомо для взаємно перпендикулярних прямих $k \cdot k_1 = -1$.

Отримали $k_1 = \frac{B}{A}$ звідки рівняння перпендикуляра до $V_i V_{i+1}$ в т. O_1 має вигляд $x - x_{o1} = \frac{B}{A}(y - y_{o1})$. Точку перетину цього перпендикуляра з колом радіуса H з центром в т. O_1 визначимо за відомим алгоритмом розв'язавши систему рівнянь

$$\begin{cases} (x - x_{o1})^2 + (y - y_{o1})^2 = H^2 \\ Ax - By - Ax_{o1} + By_{o1} = 0 \end{cases}$$

З двох отриманих точок обираємо ту що задовольняє нашим вимогам.

В тому разі коли необхідно отримати виточку прямокутної чи трапеційної форми розрахунки дещо змінюються:

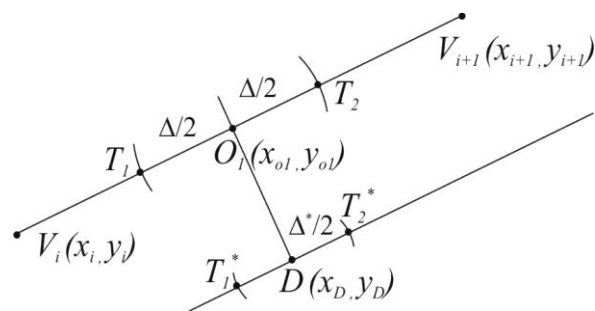


Рис.6.2.14

На заданій глибині виточки проводиться пряма що є паралельною до сторони деталі, за формулою $Ax + By + C^* = 0$ де $C^* = -Ax_D - By_D$. Знаючи ще один необхідний розмір Δ^* координати точки T_1^* та T_2^* знаходимо з системи рівнянь:

$$\begin{cases} (x - x_D)^2 + (y - y_D)^2 = \left(\frac{\Delta^*}{2}\right)^2 \\ Ax + By + C^* = 0 \end{cases}$$

Згладжування контурів деталей за допомогою параметричних сплайнів

В автоматизованому проектуванні часто необхідно побудувати гладку криву або гладку поверхню за набором заданих точок. Є декілька методів побудови гладких кривих. Для згладжування контурів

використаємо апроксимацію параметричними кубічними сплайнами за поширеними алгоритмами.

Для того, щоб провести згладжування відповідної ділянки зовнішнього контуру деталі потрібно:

- 1) вказати крайні точки ділянки зовнішнього контуру деталі, яку нам необхідно згладити;
- 2) ідентифікувати цю ділянку зовнішнього контуру деталі;
- 3) вибрати методи згладжування:
 - за допомогою параметричного В-сплайну;
 - за допомогою параметричного інтерполяційного сплайну;
- 4) перенумерувати вершини апроксимуючого багатокутника таким чином, щоб ділянка контуру, яку необхідно згладити, знаходилась наприкінці;
- 5) провести згладжування;
- 6) замінити ділянку контуру, якій необхідно згладити.

Параметричний інтерполяційний кубічний сплайн

При параболічній інтерполяції параметричним інтерполяційним сплайном інтерполяційна крива має вигляд:

$$C(t) = (1-t)p(r) + tq(s)$$

де r, s, t – параметри,

$p(r)$ – парабола, що проходить через точки K_1, K_2, K_3 ,

$q(s)$ – парабола, що проходить через точки K_2, K_3, K_4 .

$$p(r) = \begin{bmatrix} r^2 & r & 1 \end{bmatrix} [B]$$

$$q(s) = \begin{bmatrix} s^2 & s & 1 \end{bmatrix} [D]$$

$$c(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} [A] [G] = [T] [A] [G]$$

$[B]$, $[D]$ - матриці, що представляють положення вектор точок $K_1, K_2,$

K_3 , та K_2, K_3, K_4 відповідно.

$[T]$ $[A]$ - матриця інтерполяційних функцій, а $[G]$ - геометрична матриця вектор точок K_1, K_2, K_3, K_4 .

$$0 \leq r \leq 1 \text{ на відрізку } [K_1, K_3],$$

$$0 \leq s \leq 1 \text{ на відрізку } [K_2, K_4],$$

$$0 \leq t \leq 1 \text{ на відрізку } [K_2, K_3].$$

Доцільно припустити, що $r = k_1 t + k_2$, $s = k_3 t + k_4$.

Припустимо, що дані розподілені рівномірно

$$p(0) = K_1, \quad q(0) = K_2, \quad C(0) = K_2, \quad p(\frac{1}{2}) = K_2, \quad q(\frac{1}{2}) = K_3,$$

$$p(1) = K_3, \quad p(1) = K_4, \quad C(1) = K_3,$$

$$K_2: \quad r = \frac{1}{2}, \quad t = 0 \Rightarrow k_2 = \frac{1}{2}; \quad s = 0 \quad t = 0 \Rightarrow k_4 = 0;$$

$$K_3: \quad r = 1, \quad t = 1 \Rightarrow k_1 + k_2 = 1 \Rightarrow k_1 = \frac{1}{2}; \quad s = \frac{1}{2} \quad t = 1 \Rightarrow k_3 = \frac{1}{2};$$

$$r(t) = \frac{1}{2}(1+t), \quad s(t) = \frac{1}{2}t.$$

Виразимо $[B]$ через K_1, K_2, K_3

$$K_1 = p(0) = [0 \quad 0 \quad 1] [B]$$

$$K_2 = p(\frac{1}{2}) = [\frac{1}{4} \quad \frac{1}{2} \quad 1] [B], \quad \rightarrow M = \begin{bmatrix} 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{2} & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$K_3 = p(1) = [1 \quad 1 \quad 1] [B]$$

$$[B] = [M]^{-1} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix}, \quad \rightarrow M^{-1} = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

Аналогічно виразимо $[D]$ через K_2, K_3, K_4 ,

$$K_2 = q(0) = [0 \quad 0 \quad 1] [D],$$

$$K_3 = q(\frac{1}{2}) = [\frac{1}{4} \quad \frac{1}{2} \quad 1] [D],$$

$$K_4 = q(1) = [1 \quad 1 \quad 1] [D]$$

$$[D] = [M]^{-1} \begin{bmatrix} K_2 \\ K_3 \\ K_4 \end{bmatrix}$$

Оскільки $C(t) = (1-t) \begin{bmatrix} r^2 & r & 1 \end{bmatrix} [B] + t \begin{bmatrix} s^2 & s & 1 \end{bmatrix} [D]$,

то після підстановки $r(t) = \frac{1}{2}(1+t)$, $s(t) = \frac{1}{2}t$ та виразів для B і D :

$$\begin{aligned}
 C(t) &= \begin{bmatrix} -\frac{t^3}{2} \cdot t^2 - \frac{t}{2} t^3 - t^2 - t + 1 - \frac{t^3}{2} + \frac{t}{2} \\ \frac{t^3}{2} - \frac{3}{2} t^2 + t - t^3 + 2t^2 \frac{t^3}{2} - \frac{t^2}{2} \\ \frac{t^3}{2} + t^2 - \frac{t}{2} \frac{3}{2} t^3 - \frac{5}{2} t^2 + 1 - \frac{3}{2} t^3 + 2t^2 + \frac{t}{2} \frac{t^3}{2} - \frac{t^2}{2} \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} + \\
 &+ \begin{bmatrix} \frac{t^3}{2} - \frac{3}{2} t^2 + t - t^3 + 2t^2 \frac{t^3}{2} - \frac{t^2}{2} \\ \frac{t^3}{2} + t^2 - \frac{t}{2} \frac{3}{2} t^3 - \frac{5}{2} t^2 + 1 - \frac{3}{2} t^3 + 2t^2 + \frac{t}{2} \frac{t^3}{2} - \frac{t^2}{2} \end{bmatrix} \begin{bmatrix} K_2 \\ K_3 \\ K_4 \end{bmatrix} = \\
 &= \begin{bmatrix} -\frac{t^3}{2} + t^2 - \frac{t}{2} \frac{3}{2} t^3 - \frac{5}{2} t^2 + 1 - \frac{3}{2} t^3 + 2t^2 + \frac{t}{2} \frac{t^3}{2} - \frac{t^2}{2} \\ \frac{t^3}{2} + t^2 - \frac{t}{2} \frac{3}{2} t^3 - \frac{5}{2} t^2 + 1 - \frac{3}{2} t^3 + 2t^2 + \frac{t}{2} \frac{t^3}{2} - \frac{t^2}{2} \\ \frac{t^3}{2} + t^2 - \frac{t}{2} \frac{3}{2} t^3 - \frac{5}{2} t^2 + 1 - \frac{3}{2} t^3 + 2t^2 + \frac{t}{2} \frac{t^3}{2} - \frac{t^2}{2} \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix}
 \end{aligned}$$

За поданим масивом точок K_0, K_1, \dots, K_n сплайнова крива на кожній ділянці (K_i, K_{i+1}) визначається за допомогою рівняння:

$$R(t) = \frac{1}{2} (-t(1-t)^2 K_0 + (2-5t^2+3t^3) K_1 + t(1+4t-3t^2) K_2 - t^2(1-t) K_3)$$

Властивості отриманої кривої

- проходить точно через опорні точки;
- має неперервну похідну;
- змінюючи координати окремих точок можна здійснювати локальне регулювання форми кривої.

При побудові сплайнової кривої кожний сегмент на ділянці між парою внутрішніх точок розраховується з набору базових точок чергової четвірки. Тому для побудови інтерполяційної кривої, яка починається в першій базовій точці і закінчується в останній, достатньо доповнити поданий набір копіями першої і останньої. Копія початкової точки при цьому додається у початок набору, а копія останньої - у кінець набору.

Параметричний кубічний В - сплайн

В даному випадку з заданої послідовності точок вибираються дві сусідні точки та між ними будується крива кубічного полінома на основі

позицій чотирьох точок - двох попередніх та двох сусідніх з ними. В-сплайн забезпечує отримання більш гладких кривих, ніж інші способи згладжування тому, що отримані криві не проходять через задані точки. Математично гладкість кривих виражається в термінах неперервності параметричних представлень $x(t)$ та $y(t)$ та їх похідних. Криві типу В-сплайну мають властивість неперервності навіть других похідних $x''(t)$ та $y''(t)$ в точці стиковки двох сусідніх сегментів кривої. Розглядаючи цей метод, будемо використовувати параметричне представлення кривих. Будь-яка точка частини кривої між двома заданими послідовними точками K_i та K_{i+1} буде мати координати $x(t)$ та $y(t)$, де t збільшується від 0 до 1, при переміщенні по кривій від точки K_i до точки K_{i+1} . Координати обчислюються за формулою:

$$x(t) = \{(a_3 t + a_2)t + a_1\} + a_0$$

$$y(t) = \{(b_3 t + b_2)t + b_1\} + b_0$$

Ці рівняння містять наступні коефіцієнти:

$$a_3 = (-x_{i-1} + 3x_i - 3x_{i+1} + x_{i+2})/6$$

$$a_2 = (x_{i-1} - 2x_i + 2x_{i+1})/2$$

$$a_1 = (-x_{i-1} + x_{i+1})/2$$

$$a_0 = (x_{i-1} + 4x_i + x_{i+1})/6$$

а коефіцієнти b_3, b_2, b_1, b_0 розраховуються за значеннями $y_{i-1}, y_i, y_{i+1}, y_{i+2}$ аналогічним чином. Розрахунок значень $x(t)$ швидше проводити за правилом Горнера. Коефіцієнти a_3, a_2, a_1, a_0 розраховуються тільки один раз для одної частини сегмента кривої, що дуже важливо оскільки на кожному сегменті кривої можна обчислити велику кількість проміжних точок $x(t)$ та $y(t)$.

Для отримання деякого представлення про властивості кривої в точках стиковки двох сегментів розглянемо функцію $x(t)$ та її першу та другу похідну для значень $t=0$ та $t=1$ (функція $y(t)$ матиме аналогічні властивості).

$$x(0) = a_0 = (x_{i-1} + 4x_i + x_{i+1}) / 6$$

$$x(1) = a_3 + a_2 + a_1 + a_0$$

Після спрощення отримуємо:

$$x(1) = (x_i + 4x_{i+1} + x_{i+2}) / 6$$

Можна побачити, що значення $x(0)$ точно не дорівнює x -координаті x_i точки K_i : воно залежить від значень координат точки K_{i-1} та K_{i+1} . Для першого сегменту маємо:

$$A = K_i \qquad B = K_{i+1} \qquad C = K_{i+2}$$

$$x_B^* = x(0) = (x_A + 4x_B + x_C) / 6$$

де через x_B^* позначаємо розраховане значення координати x для точки B . Розглядаючи цю ж точку як ту, що належить сегменту BC , маємо:
ФОРМУЛА!!!!

$$A = K_{i+1} \qquad B = K_i \qquad C = K_{i+1}$$

$$x_B^* = x(0) = (x_A + 4x_B + x_C) / 6$$

Бачимо, що обидва способи розрахунку значення x дають однаковий результат, що означає неперервність $x(t)$ в точці B . Продиференціювавши $x(t)$ двічі, знайдемо похідні $x'(t)$ та $x''(t)$. Підставляючи в них значення $t=0$ та $t=1$. можна переконатися, що похідні неперервні в точці B . Оскільки функція $y(t)$ та її перші дві похідні також неперервні, то стає зрозумілим, що крива B -сплайна гладка.

Для розрахунку будь-якого сегменту кривої між точками K_i та K_{i+1} використовуються також точки K_{i-1} та K_{i+2} . З цього отримуємо, що перший сегмент кривої буде розташований між K_1 та K_2 , а останній – між K_{n-1} та K_n . Таким чином початкова та кінцева точки всієї кривої будуть розташовані поблизу K_1 та K_{n-1} , а не поблизу K_0 та K_n .

Збереження інформації в форматі що заданий

Інформація про створені деталі записується в файл формату *.dgt, структура якого є наступною:

- Найменування (або числовий номер) моделі виробу.
- Цифрова константа.
- Кількість деталей у моделі.
- Стовець найменувань (або номерів) деталей по порядку.
- Стовець, кожний рядок якого містить кількість вершин контуру деталі по порядку та кількість одиниць даної деталі у комплекті моделі. Для утворення замкненого контуру, коли остання точка співпадає з першою, кількість вершин збільшуємо на одиницю.
- Стовпці, кількість яких відповідає кількості деталей у моделі. Кількість рядків кожного стовпця відповідає кількості вершин контуру поточної деталі, збільшеній на одиницю. Кожний рядок стовпця містить інформацію про координати вершини контуру по порядку (відповідно X та Y), записані у міліметрах через інтервал. В останньому рядку кожного стовпця записуються координати першої вершини контуру.

6.3. ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ ДЕТАЛЕЙ

ШКІРГАЛАНТЕРЕЙНИХ ВИРОБІВ ПРИ ПОБУДОВІ РОЗКРІЙНИХ СХЕМ

Для використання нових методів автоматизованого розкрою за допомогою водяного струменя, гнучкого ножа, променя лазера та пресів-автоматів з револьверною голівкою необхідні додаткові відомості про геометрію деталей та їх місцеположення на матеріалі. Оскільки розкрійна схема для рулонних матеріалів складається з секцій в яких застосовується прямолінійно-поступальна система розміщення деталей в ряду, спочатку визначаємо варіанти щільної укладки деталей в ряду та щільної укладки рядів деталей і визначаємо їх параметри.

Щільні укладки і визначення їх параметрів залежно від орієнтації деталей в ряду і в рядах

Деталі шкіргалантерейних виробів являють собою плоскі геометричні об'єкти, зовнішній контур яких в більшості випадків не може бути описаний аналітично. При застосуванні кусково-лінійної апроксимації контури деталей представляють собою багатокутники. Для відображення положення деталі на матеріалі достатньо знати інформацію про зовнішній контур деталі та три параметри X_p , Y_p та θ , що визначають положення деталі на матеріалі:

X_p, Y_p - координати заданої точки деталі (так званого полюса, відносно якого обраховуються координати всіх точок контуру) в системі координат, пов'язаній з матеріалом;

θ - кут повороту деталі відносно її вихідного положення (в нашому випадку може приймати значення 0 та 180^0).

Для задання полюса виберемо всередині кожного плоского геометричного об'єкта S^k точку P^k , в яку помістимо початок прямокутної системи координат. Тоді зовнішній контур цього об'єкту може бути представлений координатами вершин $S^k \{X_i^k, Y_i^k\}$, $i=1..n_k$ (n_k -кількість вершин контуру S^k). В нашому випадку при створенні геометричних примітивів полюс деталі задається ще на етапі занесення інформації про контур і розташований в середній точці по висоті та ширині деталі, але при моделюванні деталі його відносне розташування може змінитись. Обхід контуру відбувається за годинниковою стрілкою і кінцева точка співпадає з початковою.

Площу $|S^k|$ кожної із деталей S^k , зовнішній контур якої представлений координатами вершин $S^k \{X_i^k, Y_i^k\}$, $i=1..n_k$, можна визначити за наступною формулою:

$$|S^k| = \frac{1}{2} \left| \sum_{i=1}^{n_k-1} (X_i^k \cdot Y_{i+1}^k - X_{i+1}^k \cdot Y_i^k) \right| \quad 6.3.1$$

Знайдемо для кожної з деталей S^k , $k=1,2,\dots,p$ наступні величини:

$$\begin{aligned} X_{min}^k &= \min_{i=1,2,\dots,n_k} \{ X_i^k \} \\ X_{max}^k &= \max_{i=1,2,\dots,n_k} \{ X_i^k \} \\ Y_{min}^k &= \min_{i=1,2,\dots,n_k} \{ Y_i^k \} \\ Y_{max}^k &= \max_{i=1,2,\dots,n_k} \{ Y_i^k \} \end{aligned} \quad 6.3.2$$

та перерахуємо координати вершин апроксимуючого багатокутника для кожної із деталей наступним чином:

$$\begin{aligned} X_i^k &= X_i^k + \frac{1}{2}(X_{min}^k + X_{max}^k) \\ Y_i^k &= Y_i^k + \frac{1}{2}(Y_{min}^k + Y_{max}^k) \end{aligned}, i = 1,2,\dots,n_k; k = 1,2,\dots,p. \quad 6.3.3$$

Знайдемо довжину та ширину прямокутника, що описаний навколо кожної із деталей S^k , $k=1,2,\dots,p$:

$$\begin{aligned} Dl^k &= X_{max}^k - X_{min}^k \\ Sh^k &= Y_{max}^k - Y_{min}^k \end{aligned} \quad 6.3.4$$

Далі будемо вважати $Dl^k(Sh^k)$ - відповідно довжина деталі та ширина деталі.

Оскільки кожна секція складається із смуг заповнених однаковими деталями, в яких застосовується прямолінійно-поступальна система (система паралелограма або решітчаста система) розміщення однакових деталей в ряду, то будуємо щільні укладки та визначаємо їх параметри при розташуванні :

- а) без повороту деталі (Рис 6.3.1.а);
- б) з поворотом деталі в ряду на 180 градусів по ширині схеми (Рис. 6.3.1.б);
- в) з поворотом деталі в ряду на 180 градусів по довжині схеми (Рис.6.3.1.в);

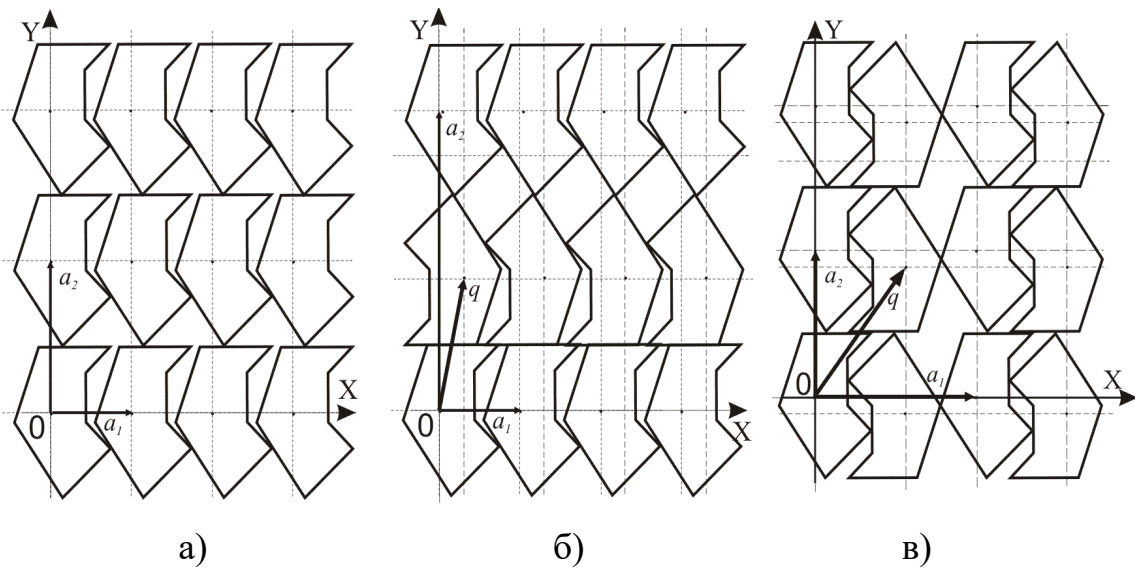


Рис. 6.3.1. Щільні укладки

Визначимо параметри подвійних решітчастих щільних укладок, а саме: вектори решітки a_1, a_2 та вектор зсуву решітки q ;

Множина векторів виду: $r=na_1+ma_2$, де $n,m=0, \pm 1, \pm 2, \pm 3, \dots \pm k \dots$ та a_1, a_2 - лінійно-незалежні вектори, називається решіткою з базисом a_1, a_2 (Рис 6.3.1.а). Позначимо її через $L=L(a_1, a_2)$. Множина векторів виду: $r=na_1+ma_2+ tq$, де $n,m=0, \pm 1, \pm 2, \pm 3, \dots \pm k \dots, t=0,1$, та a_1, a_2 - лінійно-незалежні вектори, q - вектор зсуву, називається подвійною решіткою з базисом a_1, a_2 та вектором зсуву q (Рис 6.3.1.б,в) Позначимо її через $W=W(a_1, a_2, q)$. Очевидно, що для відтворення решітчастої схеми розкрою однаково орієнтованих деталей необхідно знати вектори a_1, a_2 , а для відтворення решітчастої схеми розкрою із поворотом деталей в ряду на 180 градусів необхідно знати вектори a_1, a_2, q

Алгоритм визначення лінійних ефектів при суміщенні деталей

Щоб забезпечити щільне укладання деталей необхідно врахувати лінійні ефекти від суміщення деталей по довжині матеріалу (в ряду) та один лінійний ефект по ширині (при суміщенні рядів). Для деталей без повороту в ряду матимемо один лінійний ефект, а з поворотом – два. Позначимо через L_1^k та L_2^k - лінійні ефекти від щільного суміщення деталей

в ряду, а через L_3^k – лінійний ефект від щільного суміщення рядів. Знаходимо лінійні ефекти в ряду.

Для цього щільно сумістимо прямокутники, які описані навколо першої та другої деталей в ряду (Рис.6.3.2.а). Потім визначимо праву границю першої та ліву границю другої деталей (ліва (права) границя деталі – це контур деталі, який знаходиться зліва (справа) від точок на деталі, що мають максимальне та мінімальне значення координати Y). З вершин правої границі першої деталі проведемо прямі паралельні осі OX до перетину з контуром лівої границі другої деталі. Аналогічно, з вершин

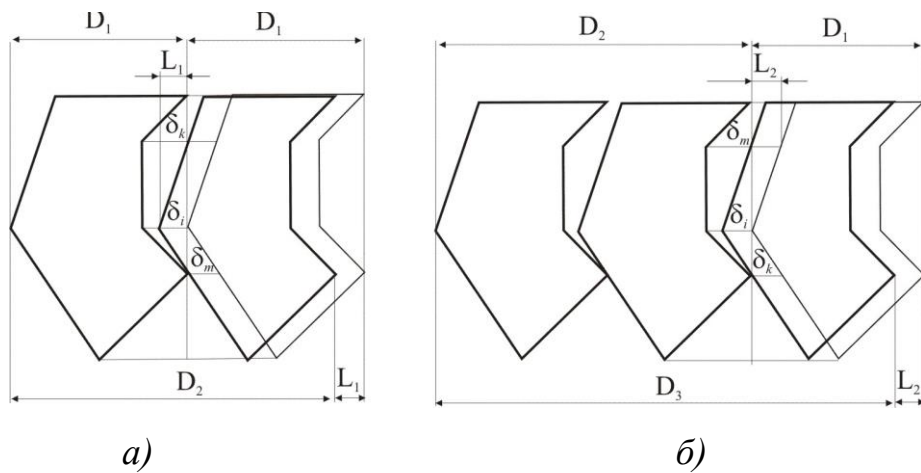


Рис 6.3.2.Лінійні ефекти при суміщенні деталей в ряду

лівої границі другої деталі проведемо прямі паралельні осі OX до перетину з контуром правої границі першої деталі. Одержимо відрізки δ_i , де $i=1..m$. Знайдемо $\delta = \min(\delta_i)$. Це і буде перший лінійний ефект L_1 від щільного суміщення деталей. Аналогічно визначається другий лінійний ефект L_2 від щільного суміщення деталей (Рис. 6.3.4.б) Для цього щільно сумістимо прямокутники, які описані навколо другої та третьої деталей в ряду та визначимо $\delta = \min(\delta_i)$ для даного випадку.

Розглянемо алгоритм визначення довжини відрізка δ_i . Нехай маємо точку з координатами (X_0, Y_0) та відрізок з координатами кінцевих точок (X_1, Y_1) та (X_2, Y_2) . Введемо позначення:

$$\begin{aligned} T_1 &= \min(X_1, X_2); & T_2 &= \max(X_1, X_2); \\ V_1 &= \min(Y_1, Y_2); & V_2 &= \max(Y_1, Y_2). \end{aligned} \quad 6.3.5$$

Тоді, якщо $T_1 \leq X_0 \leq T_2$ та $V_1 \leq Y_0 \leq V_2$, то пряма $X=X_0$, проведена з точки (X_0, Y_0) перетне відрізок з координатами кінцевих точок (X_1, Y_1) та (X_2, Y_2) та

$$\begin{aligned} \delta_i &= |Y_i - Y_0|, \\ \text{де } Y_i &= (X_0 - X_1)(Y_2 - Y_1) / (X_2 - X_1) + Y_1 \end{aligned} \quad 6.3.6$$

Нехай координати полюса першої деталі в заданій смузі будуть (X^p, Y^p) та кут повороту деталі відносно вихідної орієнтації деталі - α . Тоді координати полюсів деталей першого ряду можна виразити наступними співвідношеннями:

$$\begin{aligned} X_{p_{1,2i-1}} &= X^p + (2D_1 - L_1 - L_2)(i-1); & X_{p_{1,2i}} &= X^p + (D_1 - L_1)i + (D_1 - L_2)(i-1); \\ Y_{p_{1,2i-1}} &= Y^p; & Y_{p_{1,2i}} &= H_1 - Y^p; \\ \theta_{2i-1} &= \alpha; & \theta_{2i} &= 180^\circ + \alpha; \\ i &= 1..m, \end{aligned} \quad 6.3.7$$

де D_1 (H_1) – довжина (ширина) описаного навколо деталі прямокутника, L_1 (L_2)-перший (другий) лінійний ефект від суміщення.

Лінійний ефект від суміщення рядів

Для визначення координат полюсів деталей другого та наступних рядів в смузі необхідно знайти лінійний ефект L_3 від суміщення рядів в смузі (секції). Для цього

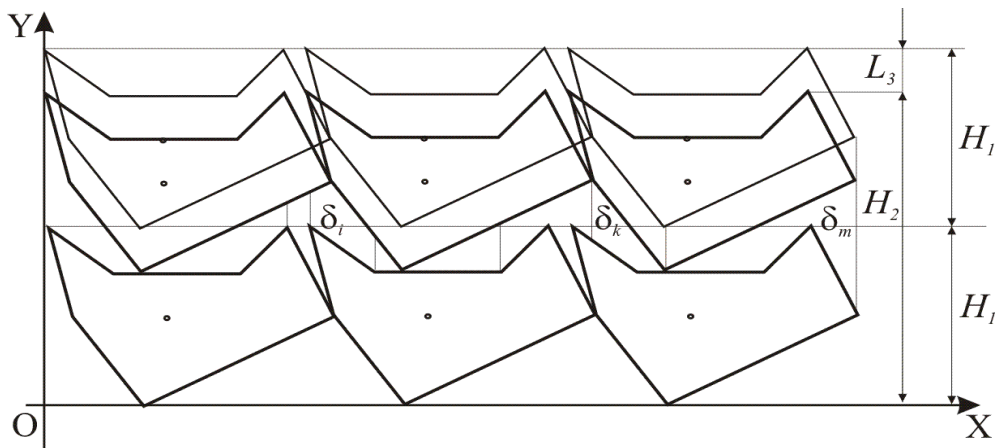


Рис. 6.3.6.3. Лінійні ефекти при суміщенні рядів

проведемо пряму в першому ряду, паралельну осі OX , через точку з найбільшим значенням координати Y , вище цієї лінії розмістимо другий ряд (рис. 6.3.3). Знайдемо верхню границю нижнього ряду та нижню границю верхнього ряду і за аналогією з визначенням лінійного ефекту δ від суміщення деталей в ряду, визначимо $\delta = \min(\delta_i)$. Тоді лінійний ефект від суміщення рядів $L_3 = \delta$.

Координати полюсів деталей будь-якого ряду можна виразити наступними співвідношеннями:

$$Xp_{j,2i-1} = X^p + (2D_1 - L_1 - L_2)(i-1); \quad Xp_{j,2i} = X^p + (D_1 - L_1)i + (D_1 - L_2)(i-1); \quad 6.3.8$$

$$Yp_{j,2i-1} = Y^p + (H_1 - L_3)(j-1); \quad Yp_{j,2i} = H_1 - Y^p + (H_1 - L_3)(j-1);$$

$$\theta_{j,2i-1} = \alpha; \quad \theta_{j,2i} = 180^\circ + \alpha;$$

де $i = 1..m$ та $j = 1..l$.

Не завжди рядна схема розкрою, в якій ряди розміщено подовж матеріалу є раціональною, тому необхідно розглянути випадок коли ряди розташовують по ширині матеріалу. В цьому разі матимемо два лінійних ефекти L_1, L_2 від щільного суміщення деталей по ширині матеріалу та один лінійний ефект L_3 від щільного суміщення деталей подовж матеріалу. Тому для розрахунку координат полюсів деталей необхідно у формулах (6.3.7-6.3.8) поміняти місцями відповідні X з відповідними Y та H_1 з D_1 .

Визначимо координати векторів a_1 , a_2 , q у випадку, коли ряди розміщуються подовж матеріалу. З Рис.6.3.1. очевидно, що $a_1=(Xa_1; Ya_1) = (Xp_{1,3} - Xp_{1,1}; 0)$, $a_2=(Xa_2; Ya_2) = (0; Yp_{2,1} - Yp_{1,1})$, $q=(Xq_1; Yq_1) = (Xp_{2,2} - Xp_{1,1}; Yp_{2,2} - Yp_{1,1})$. У випадку, коли ряди розміщуються по ширині матеріалу, матимемо: $a_1=(Xa_1; Ya_1) = (0; Yp_{1,3} - Yp_{1,1})$, $a_2=(Xa_2; Ya_2) = (Xp_{2,1} - Xp_{1,1}; 0)$, $q=(Xq_1; Yq_1) = (Xp_{2,2} - Xp_{1,1}; Yp_{2,2} - Yp_{1,1})$.

З отриманих схем виберемо ту, для якої площа паралелограма S , що побудований на векторах a_1 , a_2 , буде найменшою. При цьому необхідно розглянути варіанти для однаково орієнтованих деталей та для деталей із поворотами в ряду або з поворотами рядів на 180 градусів. Площу паралелограма визначимо як модуль векторного добутку векторів a_1 , a_2 , тобто $S=|[a_1 \times a_2]|$, або

$$S = \left| \begin{vmatrix} Xa_1 & Ya_1 \\ Xa_2 & Ya_2 \end{vmatrix} \right| = |Xa_1Ya_2 - Xa_2Ya_1| \quad 6.3.9$$

Визначаємо щільність кожної із укладок наступним чином:

$$\delta_{S^k} = \frac{2|S^k|}{\det(\vec{a}_1, \vec{a}_2)} = \frac{2|S^k|}{|\vec{a}_1 \times \vec{a}_2|} = \frac{2|S^k|}{|a_{1x}a_{2y} - a_{1y}a_{2x}|} \quad 6.3.10$$

де в нашому випадку: $\vec{a}_1 = \{a_{1x}, a_{1y}\}$, $\vec{a}_2 = \{a_{2x}, a_{2y}\}$, $\vec{q} = \{q_x, q_y\}$.

Щільне суміщення секцій

Оскільки розкрійна схема складається окремих секцій або розкладок. Ці секції при побудові розкрійної схеми суміщаються як прямокутники, описані навколо них. А це не завжди є раціональним, так як в цьому випадку секції суміщені не завжди щільно (Рис.6.3.7.а) Запропонований алгоритм щільного суміщення секцій при побудові розкрійних схем. Нехай довжина j -ої розкладки дорівнює L^j та координати полюсів будуть X_k^j, Y_k^j , де $k=1..n(j)$. Для щільного суміщення j -ої та i -ої розкладки необхідно знайти нові координати X_k^i , де $k=1..n(i)$. Почакове їх значення можна визначити як $X_k^i = X_k^i + L^j$, тобто для суміщення без врахування можливості

щільного суміщення розкладок(Рис.6.3.7.а). Для щільного суміщення розкладок необхідно знайти праву границю j -ої розкладки та ліву границю i -ої розкладки після попереднього суміщення(Рис.6.3.7.б). (*Изменить рисунок!!!!*)

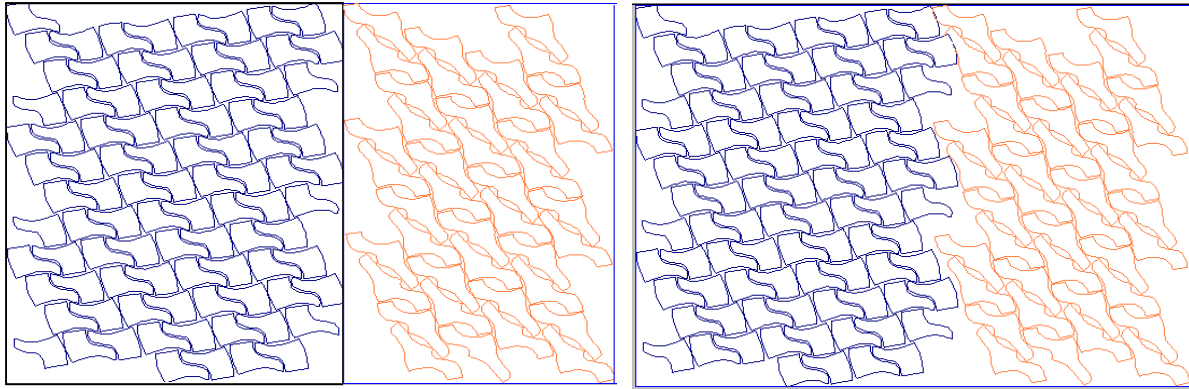


Рис. 6.3.7.

а) щільне суміщення описаних прямокутників б) щільне суміщення

Під правою границею деталі будемо мати на увазі контур деталей, який знаходиться праворуч від прямої, що проведена через вершини із максимальним та мінімальним значеннями $У$. Під лівою границею деталі будемо підрозумівати контур деталі, який знаходиться ліворуч від прямої, що проведена через вершини із максимальним та мінімальним значеннями $У$. Права границя j -ої розкладки складається із правих границь деталей, для яких виконується нерівність $X_k^j > L^j - MaxX^j$, де $MaxX^j = 2 * \max\{MX_q\}$ та q -порядкові номери деталей, що містяться в j -ій розкладці ($q \in 1..n$). Ліва границя i -ої розкладки складається із лівих границь деталей, для яких виконується нерівність $X_k^i < MaxX^i$, де $MaxX^i = 2 * \max\{MX_q\}$ та q -порядкові номери деталей, що містяться в i -ій розкладці. Для щільного суміщення j -ої та i -ої розкладок необхідно щільно сумістити праву границю j -ої розкладки та ліву границю i -ої розкладки (Рис.6.3.7.б) Для цього використаємо алгоритм, який запропонований вище. (див щільне суміщення деталей та статті!!!!)

Висновки по розділу:

В даному розділі розглянуто алгоритми підготовчого етапу побудови розкрійних схем такі як:

- визначення параметрів щільних укладок деталей з урахуванням можливості різної орієнтації деталей в ряду і в рядах;
- визначення лінійних ефектів від щільного суміщення деталей в ряду;
- визначення лінійного ефекту при щільному суміщенні рядів;
- алгоритм щільного суміщення секцій деталей.

6.4. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ГЕНЕРАЦІЇ РОЗКРІЙНИХ СХЕМ ДРІБНОЇ ШКІРГАЛАНТЕРЕЇ НА РУЛОННИХ МАТЕРІАЛАХ

Сучасне виробництво характеризується необхідністю ретельного аналізу і економії матеріалів ще на етапі проектування виробів а також необхідністю швидкої зміни асортименту виробів для задоволення потреб споживачів. Особливо гостро ця проблема стоїть у взуттєвій та шкіргалантерейній промисловості, оскільки взуття та шкіргалантерейні вироби є матеріалоемкими виробами, і вартість матеріалів в собівартості цих виробів досягає 87%. Тому в умовах інтенсифікації взуттєвого та шкіргалантарейного виробництв назріла необхідність створення науково-обґрунтованих методів проектування схем розкрою матеріалів з урахуванням вимог автоматичного розкрою з метою отримання бажаних раціональних результатів в стислі часові строки.

Технологічна постановка задачі. Дано область прямокутної форми із шириною Sh і довжиною Dl та комплект деталей S^k , $k=1,2,..,p$, де Q_k – кількість k – тих деталей в комплекті. Необхідно на прямокутній області $Dl \times Sh$ розмістити R комплектів деталей S^k із урахуванням технологічних вимог таким чином, щоб довжина Dl_R заповнена розкрійною схемою була

найменшою. Під технологічними вимогами матимемо на увазі: допустимий поворот деталі на матеріалі відносно її основного положення (0 або 180 градусів); краєвий зазор σ ; міжмодельний місток Δ .

Математична постановка задачі. В області прямокутної форми, яка визначається наступною системою нерівностей:

$$\begin{cases} \Delta/2 \leq x \leq Dl - \Delta/2 \\ \Delta/2 + \sigma \leq y \leq Sh - \Delta/2 - \sigma \end{cases}$$

з множини допустимих схем розкрою для R комплектів деталей S^k , знайти таку розкрійну схему довжина Dl_R якої буде мінімальною, тобто довжина якої $Dl_R = \min_{i=1,2..t} (Dl_R^i)$, де Dl_R^i - довжина i -тої розкрійної схеми із множини допустимих розкрійних схем.

В алгоритмі можна виділити такі основні етапи:

- підготовчий етап;
- генерування множини допустимих секцій;
- вибір такої комбінації секцій, що забезпечуватиме комплектний вихід деталей, мінімальну довжину схеми та процент використання матеріалу для побудованої розкрійної схеми буде максимальним.

Зупинимося більш детально на кожному з етапів алгоритму.

Підготовчий етап

Зовнішні контури деталей представлено координатами вершин $S^k \{X^k, Y^k\}$, $i=1..n_k$ (n_k -кількість вершин контуру S^k). На даному етапі:

- обчислюємо площу кожної з деталей комплекту за формулою (п.6.4.)
 - знаходимо координати полюсів деталей за формулою
 - за формулами перераховуємо координати вершин контурів деталей
- знаходимо довжину та ширину прямокутника, що описаний навколо кожної із деталей S^k , $k=1,2,..,p$. Далі будемо вважати $Dl^k(Sh^k)$ - відповідно довжина деталі та ширина деталі.

- відсортуємо деталі за спаданням ширини, тобто таким чином, що буде справедлива нерівність $Sh^i > Sh^{i+1}, i = 1, 2, \dots, p-1$.

- будемо щільні укладки однакових деталей та визначаємо їх параметри відповідно до наведених в п.п.....алгоритмів.

- з отриманих схем виберемо ту, для якої площа паралелограма S , що побудований на векторах a_1, a_2 , (див п.п.....) буде найменшою. При цьому необхідно розглянути варіанти для однаково орієнтованих деталей та для деталей із поворотами в ряду або з поворотами рядів на 180 градусів. Площу паралелограма визначимо як модуль векторного добутку векторів a_1, a_2 , тобто $S = |[a_1 \times a_2]|$, або

$$S = \begin{vmatrix} Xa_1 & Ya_1 \\ Xa_2 & Ya_2 \end{vmatrix} = |Xa_1Ya_2 - Xa_2Ya_1|$$

- визначаємо щільність кожної із укладок наступним чином:

$$\delta_{S^k} = \frac{2|S^k|}{\det(\vec{a}_1, \vec{a}_2)} = \frac{2|S^k|}{|[\vec{a}_1 \times \vec{a}_2]|} = \frac{2|S^k|}{|a_{1x}a_{2y} - a_{1y}a_{2x}|}$$

де в нашому випадку: $\vec{a}_1 = \{a_{1x}, a_{1y}\}$, $\vec{a}_2 = \{a_{2x}, a_{2y}\}$, $\vec{q} = \{q_x, q_y\}$.

- обираємо ту укладку для деталі S^k та її параметри, щільність якої буде більшою.

Математична модель секції розкрийної схеми

Генерування секцій полягає в побудові розрахунковим способом множини окремих варіантів розкладок (під окремим варіантом розкладки мається на увазі певний варіант комбінування деталей по ширині і довжині настилу з урахуванням потреби в них). Сукупність окремих варіантів розкладок утворюватиме план розкрою настилу.

Маємо комплект деталей виробу, що складається з p різних деталей. Треба створити множину різних секцій з даних деталей, де для кожної окремої секції виконуються наступні умови: кількість деталей в ній не перевищує потребу в данному типіві деталей, процент використання

матеріалу задовольняє нормативам галузі. В результаті генерування множини допустимих секцій ми отримуємо матрицю $B(t, p)$, де t - кількість згенерованих секцій, p - кількість деталей у комплекті. Тоді елемент матриці $B_{ij}, i=1,2..t; j=1,2..p$ - це кількість j -тих деталей у i -ій секції.

Процент використання матеріалу в секції визначаємо як

$$Vids_i = Snet_i / Sbrut_i \cdot 100$$

де $Sbrut_i$ - площа «брутто» секції, а

$$Snet_i = \sum_{j=1}^p B_{ij} \cdot |S_{det}^j| \text{ корисна площа всіх деталей секції.}$$

Потребу в деталях визначають наступним чином. Нам відома довжина настилу і мінімально допустимий процент використання матеріалу. Обчисливши площі окремих деталей та з урахуванням заданої комплектності можемо визначити допустиму кількість комплектів, що можна викроїти з данного настилу.

$$N = \frac{S_{наст} \cdot Pv}{\sum_{i=1}^p Q_i S_i}$$

де N - допустима кількість комплектів, що можна отримати

$S_{наст}$ - площа настилу

p - загальна кількість різних деталей в комплекті

Q_i - кількість i -тих деталей в комплекті

S_i - площа окремої i -тої деталі

Pv - процент використання матеріалу

Модифікація генетичного алгоритму для побудови секцій деталей шкіргалантерейних виробів

Для генерування множини секцій запропоновано обрати модифікований щодо конкретних умов генетичний алгоритм. На даному етапі задача полягає в отриманні певної кількості різноманітних

комбінацій деталей, так званих секцій деталей, які задовольняють вимозі мінімізації залишків.

Секції розкрійної схеми представляють собою прямокутні області постійної ширини Sh та змінної довжини Dl_i щільно заповнені деталями і є варіантами можливих розв'язків задачі оптимізації корисної площі секції $Vids_R = \max_{i=1,2..t} (Vids_R^i)$, де $Vids_R^i$ - процент використання площі i -тої розкрійної секції з множини допустимих секцій.

При генерації секцій важливо отримати максимальну варіативність деталей, тобто пошук зводиться не до визначення глобального оптимуму, а до знаходження більшості з можливих локальних оптимумів в просторі пошуку. Це обумовлює особливості обраних модифікацій основних операторів генетичного алгоритму. Оскільки ГА є різновидом методу пошуку з елементами випадковості, і мають за мету отримання кращого, а не оптимального рішення, а також на кожному кроці роботи алгоритму використовується кілька точок пошуку одночасно, то генетичний алгоритм як комбінація переборного і градієнтного методів, дозволяє отримувати різноманітні розв'язки з дотриманням вимог до „якості” особин. В нашому випадку це процент використання матеріалу. (лит-ра Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритми 2-изд, испр.и доп. – М. ФИЗМАТЛИТ, 2006, 320с).

Пропонується розглядати секцію як особину що кодується двома хромосомами. Перша хромосома задає набір деталей з яких складається секція, друга визначає метод укладки (кількість деталей кожного виду) в секцію. Хромосоми являють собою вектори, довжина яких дорівнює кількості деталей що передбачається розкласти в секції, наприклад числом різновидів деталей в комплекті. Кількість рядів та кількість деталей в рядах визначаємо за формулами наведеними нижче.

В табл. 6.4.1 наведено приклади кодування особин популяції. Перша особина являє собою секцію що складається з одної першої в комплекті деталі укладеної в 5 рядів. Друга – теж з першої деталі, але укладеної по 3 в ряду. Третя – з 17-ти рядів 3-ої деталі. В четвертій особині маємо 2 ряди першої деталі укладеної по 3 в ряд та 8 рядів третьої деталі укладеної по 4 в ряд.

Номер секції (i)	1-а хромосома (визначає кількість рядів j -тої деталі в i -тій секції) R_{ji}	2-а хромосома (визначає кількість j -тої деталі в ряду i -тої секції) D_{ji}
1	{5,0,0,0,0}	{1,0,0,0,0}
2	{5,0,0,0,0}	{3,0,0,0,0}
3	{0,0,17,0,0}	{0,0,1,0,0}
4	{2,0,8,0,0}	{3,0,4,0,0}

Таблиця 6.4.1. Кодування особин

Класична структура генетичного алгоритму наведена на рис.6.4.1.



Рис.6.4.1

Модифікація генетичного алгоритму для генерації секцій розкрійних схем полягає в наступному:

- алгоритм призначений не для знаходження кращої секції, а для генерування множини нових секцій з раціональними показниками;
- початкова популяція генерується з деталей одного типу, розташованих по одній, дві, три в рядах з різними розташуваннями враховуючі щільність укладки;

- в процедурі розмноження приймає участь та хромосома що відповідає за кількість рядів деталей і тому забезпечує перебірну частину алгоритму;

- процедурою розмноження передбачено ділення не всієї хромосоми на дві частини, а кожного гену хромосоми навпіл (різновид лінійної дискретної рекомбінації (ссілка на книгу Панченко...));

- процедура мутації здійснюється над обома хромосомами і є спрямованою на збільшення проценту використання матеріалу, це забезпечує градієнтну частину алгоритму;

- вихід з алгоритму відбувається тоді коли популяція досягає заданого розміру, або ж коли в циклі не відбувається створення особин що задовольняли б вимогам і не повторювали вже існуючі.

Розробка програмного забезпечення з реалізації генетичного алгоритму для генерації секцій

Програмний модуль що реалізує модифікований генетичний алгоритм має наступну структуру:

1) Створюємо початкову популяцію секцій утворених з однакових деталей. Деталі попередньо відсортовані по довжині. Спочатку генеруються секції з одної деталі в ряду. Потім по дві, три враховуючи розташування деталей залежно від обрахованої на підготовчому етапі за формулами (6.4. -6.4.) щільності. Перевіряємо щоб кількість деталей секції не перевищувала потребу в них, обраховану відповідно до формул.

При розміщенні деталей можливі наступні два випадки:

- деталі в секції розміщаються із поворотом на 180 по ширині секції(рис.6.4.1.а);
- деталі в секції розміщаються із поворотом на 180 по довжині секції(рис.6.4.1.б).

В першому випадку кількість рядів N_r^k визначається наступним чином:

$$N_r^k = \max(2n_1; 2n_2 + 1) \quad (6.4.12)$$

де

$$n_1 = \left[\frac{Sh - L_2 - 2\sigma - \frac{3}{2}\Delta}{2Sh^k - L_1 - L_2 + \Delta} \right] \quad (6.4.13)$$

$$n_2 = \left[\frac{Sh - Sh^k - 2\sigma - 2\Delta}{2Sh^k - L_1 - L_2 + \Delta} \right]$$

та $[]$ - ціла частина виразу, що знаходиться в дужках.

В другому випадку кількість рядів N_r^k визначається наступним чином:

$$N_r^k = \left[\frac{Sh + L_3 - 2\sigma - \frac{3}{2}\Delta}{Sh^r - L_3 + \frac{\Delta}{2}} \right] \quad (6.4.14)$$

Кількість деталей в ряду N_d^k для секції однієї деталі S^k знайдемо наступним чином:

$$N_d^k = \left[\frac{R \cdot Q_k}{N_r^k} \right] \quad (6.4.15)$$

залежно від розмірів деталі та матеріалу за формулою $R_{ji} = SH \cdot \text{div} \cdot Sh_j$ де SH – ширина матеріалу а Sh_j ширина j -ї деталі.

При побудові кожного варіанту комбінування деталей за шириною враховуються лінійні ефекти деталей, припустима ширина некротної смуги і величина міжрізачних втрат.

Ці умови відображаються такою залежністю (СОСЛАТЬСЯ НА Рожка??)

$$SH - \sum_{j=1}^p (Sh_j \cdot R_{ij} + L_j + d_j \varepsilon) \leq \sigma \quad (6.4.16)$$

де Sh – ширина матеріалу

Sh_j - розрахункова ширина j деталі

R_{ij} - кількість деталей j -го виду в i -ому варіанті розміщення деталей ;

L_j - лінійний ефект від суміщення за шириною деталей j -го виду;

σ - величина крайових втрат, що задається;

d_j - кількість міжрізачних містків.

$$d_j = \frac{R_{ij}}{k_j} \quad (6.4.17)$$

якщо d_j - ціле, при дробовому значенні відношення $\frac{R_{ij}}{k_j}$, значення d_j

приймається рівним величині

$$d_j = \frac{R_{ij}}{k_j} + 1 \quad (6.4.18)$$

де k_j - максимально допустима кількість деталей j -го виду в груповому різаку;

ε - ширина міжрізачного містка.

Розглянемо більш детально методику за якою визначається кількість деталей, що можна розташувати по висоті у заданій ширині настилу при створенні всіх можливих секцій з переліку деталей, що входять до комплекту моделі виробу.

1. Деталі комплекту сортуються за довжиною (по спадаючій). Якщо декілька деталей мають однакову довжину, першою ставиться та деталь, що має більшу площу.

2. Перша деталь у відсортованому переліку вкладається у нижній лівий кут настилу.

3. Обирається оптимальне розташування даної деталі (основне або з поворотом на 180°).

4. Над першою деталлю розташовуємо другу (в початковій популяції вкладається та ж сама деталь).

5. Обирається оптимальне розташування другої деталі (основне або з поворотом на 180°) за мінімальним першим (другим) лінійним ефектом від суміщення двох деталей за висотою.

б. Друга деталь зміщується вниз на величину першого (другого) лінійного ефекту за висотою. Кількість рядів для кожної деталі обраховується за формулами ()

Наступна секція – по довжині вкладається ряд з двох однакових деталей першого типу (враховуючи лінійні ефекти суміщення деталей по довжині), а потім вкладається наступний ряд і так далі. Секції, в яких кількість деталей перевищує необхідну (максимально можливу) – не враховуються.

Розмір популяції визначаємо залежно від конкретних умов – кількості різних деталей в комплекті. Кількість особин початкової популяції для кожного окремого випадку залежить від комплектності деталей моделі що розглядається. Таким чином кодуємо всю популяцію і визначаємо відсоток використання площі для кожної особини. Відсортуємо отримані особини за спаданням проценту використання . Знаходимо координати полюсів деталей.

2) Процедура відбору *SELECT* «батьківської пари» секцій (можливих розв'язків) для розмноження з яких отримуватимемо секцію-«нащадка» може відбуватися за різними принципами. Найбільш поширеними операторами є

- панміксія – найпростіший вибір, коли випадковим чином обирається порядковий номер особини що прийматиме участь в подальшому розмноженні;

- селекція - полягає в тому, що предками можуть стати лише ті особини (можливі розв'язки) у яких цільова функція є більшою за задану «порогову» величину що може визначатись також кількома способами.

- інбридінг – метод коли перший «предок» обирається випадково, а другим береться найближча (за мінімальною відстанню Хеммінга, або мінімальною евклідовою відстанню) особина, концентрує пошук в локальних проміжках підозрілих на екстремум;

- аутбридінг подібний до попереднього метода, але друга особина відбирається вже за максимальною відстанню, цей метод є направленим на перегляд нових недосліджених областей простору пошуку.

Інбридінг дозволяє отримувати особини з високими показниками функції цілі, але швидко веде до так званого виродження, коли отримуємо локальний екстремум критерію якості, але губимо різноманітність комбінацій. Аутбридінг навпаки дозволяє поширити коло пошуку.

Для генерації секцій розкрійних схем пропонується застосовувати чередування методів інбридінгу та аутбридінгу.

3) Процедура *RECOMBINATION*. Обрані при селекції пари підлягають операції схрещення. У випадку коли хромосоми закодовано дійсними числами застосовують так звану дискретну рекомбінацію. В нашому разі обираємо лінійну дискретну рекомбінацію в якій кожна „батьківська” хромосома R_i , що визначає кількість рядів, підлягає діленню навпіл (кожен елемент вектору множиться на постійний коефіцієнт 0,5). В разі непарності чисел розглядатимемо тільки цілочисельну частку від ділення на 2 - ($r_{ij} \text{ div } 2$). Отримуємо комбінацію для одного «нащадка» від кожної пари «батьків». За алгоритмом змінюється лише хромосома R_i , одиничні ряди не діляться взагалі, а покращення „якості” особини відбувається в процесі мутації.

4) Процедура *MUTATION* – частина алгоритму що імітує градієнтний спуск, тому що мутація іде в напрямку покращення особини. Мутація отриманих нащадків іде за двома напрямками:

- мутація за висотою перевіряє можливість вкладання ряду деталей по ширині матеріалу, тобто змінює хромосому R_i
- визначаємо висоту незаповненої деталями секції смуги матеріалу як різницю між шириною матеріалу та сумарною висотою всіх рядів деталей з урахуванням лінійних ефектів, міжшаблонних містків та ін.

- шукаємо деталь що за своєю висотою може бути розміщена в смузї. Переглядаємо деталі за спаданням висоти. В тому разі коли в комплекті є деталь що може бути розташована в додатковий ряд, вносимо необхідні зміни в хромосому R_i і обраховуємо необхідні лінійні ефекти;
- якщо немає деталей потрібного розміру, переходимо до наступного кроку.
- мутація по ширині шляхом додавання деталей з метою заповнення секції по довжині матеріалу, тобто змінює хромосому D_i :
- спочатку обраховуємо кількість деталей меншої ширини що можна розташувати в рядах над рядами більших за шириною деталей. Оскільки ширина секції визначається шириною першого ряду деталей то під час вкладання деталей враховується, що ширина деталі (сумарна ширина суміщених деталей) раніше розташованого ряду має бути більша (або дорівнювати) за цей показник для деталі (сумарної ширини суміщених деталей) наступного ряду.
- враховуємо лінійні ефекти від щільного суміщення деталей в ряду за алгоритмом, тобто відбуваються зміни в хромосомі D_i .
- враховуємо лінійний ефект від суміщення новоутворених рядів, за алгоритмом.
- обчислюємо процент використання площі для отриманих особин.

5) Розширення популяції відбувається додаванням нових особин в попередню популяцію, тобто нове покоління створюється з сукупності нащадків та предків.(ссілка на книги, название метода).

6) Процедура *REDUCTION* – створення нової популяції. Тут бажано застосувати принцип елітизму коли в наступне покоління включають кращі особини з попереднього покоління. Використання такої стратегії розширення популяції не дає можливості загубити кращі рішення. Сортуємо отриману популяцію за спаданням коефіцієнту використання і з

відсортованої популяції відкидаємо особини за значенням коефіцієнту використання площі (генеруємо раціональні секції), тобто розмір популяції збільшується. В разі пошуку оптимальної кращої особини, критерієм є збереження розміру популяції. Також необхідно провести перевірку на нетотожність особин щоб уникнути „клонування”.

Умовою закінчення процесу і виходу з програми можуть бути як процент використання площі, що не повинен бути меншим за нормативний для усіх особин популяції (у випадку незмінного розміру популяції), так і розмір згенерованої популяції, що залежить від вихідних даних кожної конкретної задачі і попередньо задається користувачем.

Розглянемо приклад генерації раціональних секцій комплекту з 5-ти різних деталей що розкладається на матеріалові шириною 1500 мм.

В Таблиці 6.4.2 показано початкову популяцію що складається з 12 секцій:

№ секц.	Хромосома 1	Хромосома 2	% використ	Ширина (мм)
1	9 0 0 0 0	1 0 0 0 0	91,6667	1440
2	9 0 0 0 0	2 0 0 0 0	91,6667	1440
3	9 0 0 0 0	3 0 0 0 0	91,667	1440
4	0 4 0 0 0	0 1 0 0 0	74,667	1280
5	0 4 0 0 0	0 2 0 0 0	74,667	1280
6	0 4 0 0 0	0 3 0 0 0	74,667	1280
7	0 0 6 0 0	0 0 1 0 0	87,500	1380
8	0 0 6 0 0	0 0 2 0 0	87,500	1380
9	0 0 0 12 0	0 0 0 1 0	90,000	1440
10	0 0 0 12 0	0 0 0 2 0	90,000	1440
11	0 0 0 0 21	0 0 0 0 1	87,500	1470
12	0 0 0 0 21	0 0 0 0 2	87,500	1470

Таблиця 6.4.2. Вихідна популяція

Після першого циклу розмноження отримаємо популяцію з 19 особин з процентом використання площі не меншим за 75%, решта особин виключена з подальшого розгляду. Після другого циклу популяція складатиме вже 36 варіантів секцій з процентом використання 75% і

більше. Приклади особин популяції після другого циклу наведена в таблиці 6.4.3.

№ секц.	Хромосома 1	Хромосома 2	% використ	Ширина (мм)
1	5 0 5 0 0	1 0 2 0 0	89,8148	1490
2	5 1 0 3 0	2 3 0 9 0	89,1556	1480
3	4 0 0 3 6	3 0 0 13 18	87,4047	1420
4	2 0 2 3 5	1 0 2 4 6	87,1296	1490
5	3 3 0 0 0	3 4 0 0 0	82,4074	1440
6	0 2 0 6 0	0 3 0 9 0	80,5533	1360

Таблиця 6.4.4. Деякі особини популяції після першого циклу генерації

Хромосома1	Хромосома2	Схематичне зображення секцій
Вихідна популяція		
[5,0,0,0] [5,0,0,0] [5,0,0,0] [5,0,0,0] [5,0,0,0] [15,0,0,0] [9,0,0,0] [9,0,0,0]	[1,0,0,0] [-2,0,0,0] [0,1,0,0] [0,-2,0,0] [0,-3,0,0] [0,0,1,0] [0,0,0,1] [0,0,0,1]	
Створення нової генерації		
1-ше покоління (окремі представники)		
[2,3,0,0] [3,2,0,0] [2,0,0,5] [2,0,8,0] [3,2,0,0] [2,3,0,0]	[1,1,0,0] [1,1,0,0] [1,0,0,9] [1,0,2,0] [-2,3,0,0] [-2,-3,0,0]	
Створення нової генерації		
2-ге покоління		
[2,0,4,0,3] [2,1,6,0,0] [2,1,0,0,4]	[1,0,2,0,9] [-2,3,4,0,0] [-2,1,0,0,10]	

Розробка генетичного алгоритму побудови розкрійних схем зі згенерованих секцій

Раціональна розкрійна схема має складатись з такої комбінації секцій, що забезпечить комплектний вихід деталей і процент використання матеріалу в ній буде максимальним

Визначення раціонального набору варіантів розкладок в плані розкрою настилу отримуємо на основі рішення економіко-математичної моделі за допомогою цілочисельного методу лінійного програмування.

Математичну модель цієї задачі можна представити наступним чином:

$$S = \sum_{i=1}^t S_i \cdot k_{ilk_i} \rightarrow \max \quad 6.4.19$$

при наступних обмеженнях:

$$R \cdot Q_j - \Lambda \leq \sum_{i=1}^t B_{ij} \cdot k_{ilk_i} \leq R \cdot Q_j, \quad j = 1, 2, \dots, p \quad 6.4.20$$

де $L_i = \sum_{j=1}^p B_{ij} \cdot |S^j|$ - корисна площа матеріалу в i -ій секції;

Λ - допустиме відхилення кількості деталей від плану задачі;

$$k_{ilk_i} \geq 0; \quad B_{ij} \geq 0; \quad R > 0; \quad Q_j > 0;$$

k_{ilk_i}, Q_j, B_{ij} - цілі.

При використанні традиційних алгоритмів багатопараметричного пошуку для розв'язку подібних задач виникають деякі труднощі, такі як різке зростання обчислювальних витрат та часу при збільшенні числа параметрів що змінюються, або ж можливості «зависання» алгоритму пошуку в наближеності до одного з локальних екстремумів та ін. Незважаючи на те що евристичний характер генетичного алгоритму не гарантує оптимальності отриманого рішення, аналіз результатів використання ГА дозволяє назвати ознаки згідно з якими задачі ефективніше розв'язувати з застосуванням ГА: широке поле пошуку на якому кілька екстремумів; складна в формалізації функція оцінки якості

рішення; базатопараметричність задачі; пошук за даними критеріями прийняттого, а не єдино оптимального розв'язку; пошук розв'язку в режимі близькому до реального часу. Генетичний алгоритм пропонується до застосування тому що: - задача розкрою є NP-складності (см.лит.флешка Резніков); - дозволяє задовольняти досить високі вимоги по часу; - при проведенні досліджень прямокутного розкрою (Моделі и методи рішення задач ортогонального раскроя и упаковки: аналитический обзор и новая технология блочних структур. Є.А.Мухачева и др.// Приложение к журналу Информационніе технологии- 2004 №5 32с) було доведено, що в середньому ГА в різних модифікаціях дають кращі результати порівняно з іншими методиками.

Для нашої задачі необхідно визначити які саме згенеровані секції і в якому наборі доцільно використати щоб задовольнити комплектний випуск продукції і забезпечити макимальне використання рулонних матеріалів за умови що з одного рулону треба отримати цілу кількість комплектів деталей.

В залежності від технологічних можливостей розкрою обрахунки можна проводити як стосовно цілого рулону, або ж обраховуючи розкрій в багат шарових настилах. Але принцип єдиний:

Нетто деталей комплекту S_k розраховується за формулою

$$S_{net} = \sum_{j=1}^p S_j Q_j,$$

де S_j - площа j -тої деталі, $j = 1, 2, \dots, p$,

Q_j - кількість j -тої деталі в комплекті.

Брутто комплекту знаходимо з припущення, що рекомендований відсоток використання матеріалу P_0 , тоді

$$S_{br} = 100 S_{net} / P_0.$$

Кількість комплектів у розкладці

$$m = [Dl * Sh / S_{br}].$$

- Знаючи кількість комплектів далі розраховуємо потребу в кожному окремому виді деталі:

$$T_j = Q_j \times m.$$

- Розкрійну схему $Shem_i$, будемо розглядати як набір з n секцій, отриманих на попередньому етапі генерації. Схему кодуємо хромосомою, яка представляє собою послідовність з нулів та одиниць. Наприклад, при $n=10$ схема може задаватись як $Shem_i=(1,1,0,0,0,1,0,1,0,0)$ де 1 – означає що дана секція входить до розкрійної схеми, а 0 – що не входить (подробнее о кодировке и общий вид формул)
- Для кожної схеми обраховуються сумарні кількості окремих деталей (формули B_{ij})

Метою даної модифікації генетичного алгоритму є знаходження такої комбінації з 0 та 1, коли максимально задовольнятиметься потреба в деталях з урахуванням комплектності.

- Процедури побудови початкової популяції (визначення довжини послідовності-хромосоми) і функції цілі (задання комплекту) залежать від конкретних технологічних умов.
- Процедури селекції, кросоверу, мутації і оновлення популяції є досить типовими для алгоритмів такого призначення:
 - для бінарної послідовності селекція може відбуватись способами панміксії, турнірним чи рулетковим відбором. Обрано селекційний метод як такий що забезпечує більш швидку сходимість алгоритму
 - (Можливість потрапляння в локальній екстремум1) Селекція полягає в тому, що батьками стають лише ті особини, пристосованість яких (критерій якості) не менша за порогову величину, наприклад за середнє значення пристосованості по популяції.

- кросовер можна застосовувати як одно- так і двоточковий (см Книга подробнее). Пропонується простіший одноточковий кросовер в якому випадковим чином обирається одна точка в хромосомі, точка розриву, в якій обидві батьківські хромосоми діляться на дві частини і обмінюються ними (в разі двоточкового – точок розриву дві).
- мутацію обрано найпростішу – коли випадковим чином інвертується довільний ген (0 замінюємо на 1 або навпаки);
- для відбору особин в нову популяцію застосовано елітарний відбір, коли створюється проміжна популяція з предків та нащадків і серед них обирають найкращих з яких створюють нову популяцію того ж самого розміру що й попередня. Така стратегія не дозволить загубити кращі розв'язки.
- критерій виходу – коли отримані нащадки перестають бути кращими за предків (при сортуванні не відбуватиметься перестановок)

В результаті отримаємо популяцію з кращих розкрійних схем, одна з яких буде максимально наближена до оптимальної.

При побудові розкрійної схеми для розкрою в багатошарових настилах пропонується застосовувати комбінацію генетичного алгоритму з традиційним методом динамічного програмування:

- спочатку визначити раціональну схему розкрою конкретного настилу з урахуванням кількості шарів матеріалу і обмежень по кількості деталей;
- обрахувати кількість деталей що залишається;
- розкрій наступного настилу проводити з урахуванням обмежень отриманих на попередньому етапі.

Залишок деталей що не вийдуть в розкрійну схему додається до неї в інтерактивному режимі.

6.5. ІНТЕРАКТИВНЕ КОРИГУВАННЯ РОЗКРІЙНИХ СХЕМ, ЩО ПОБУДОВАНІ В АВТОМАТИЧНОМУ РЕЖИМІ

У багатьох випадках не вдається в автоматичному режимі побудувати схеми розкрою, які б задовольняли технологічні вимоги. Тому доводиться одержані схеми коригувати або будувати нові в інтерактивному режимі. Звідси очевидно актуальність задачі побудови та коригування схем розкрою інтерактивному режимі. При розробці алгоритму інтерактивного коригування розкрійних схем необхідно вирішити такі задачі:

1. розміщення деталей на матеріалі заданих розмірів та неперетин деталями границь матеріалу;
2. вилучення будь-якої раніш розміщеної деталі із розкрійної схеми (визначення взаємного розташування деталі та точки: точка розміщена поза деталлю чи на деталі);
3. визначення взаємного розташування двох деталей (деталі перетинаються чи ні).

Зупинимося більш детально на кожній із перерахованих вище задач. Зовнішні контури деталей виробів легкої промисловості мають складну конфігурацію і в більшості випадків вони не можуть бути описані аналітично. Тому кожна із деталей заданої моделі буде апроксимуватися опукло-ввігнутих багатокутником. В цьому випадку інформацію про зовнішній контур будь-якої деталі моделі можна однозначно задати координатами вершин апроксимуючого багатокутника. Нехай кожна деталь нашої моделі матиме свій порядковий номер $i=1..n$. Тоді координати вершин для апроксимуючого багатокутника i -ї деталі можна представити наступним чином: $\{X_{ij}, Y_{ij}\}$, де $j=1..k_i$ та k_i - кількість вершин апроксимуючого багатокутника для i -ї деталі. Визначимо максимальні значення координат вершин апроксимуючого багатокутника для i -ї деталі:

$$\begin{aligned} \text{Max}X_i &= \max\{X_{ij}\} \\ \text{Max}Y_i &= \max\{Y_{ij}\} \\ \text{Min}X_i &= \min\{X_{ij}\}, j = 1..k_i. \\ \text{Min}Y_i &= \min\{Y_{ij}\} \end{aligned} \quad (6.5.18)$$

Перерахуємо координати апроксимуючих багатокутників таким чином:

$$\begin{aligned} X_{ij} &= X_{ij} - (\text{Max}X_i + \text{Min}X_i)/2 \\ X_{ij} &= X_{ij} - (\text{Max}X_i + \text{Min}X_i)/2, j = 1..k_i \end{aligned} \quad (6.5.19)$$

В цьому випадку :

$$\begin{aligned} \text{Max}X_i &= -\text{Min}X_i = MX_i \\ \text{Max}Y_i &= -\text{Min}Y_i = MY_i \end{aligned} \quad (6.5.20)$$

Зв'яжемо систему координат з матеріалом, який має прямокутну форму (рулони або листи). Нехай початок координат знаходиться у лівому нижньому куті матеріалу. Тоді допустима область, де можуть бути розміщені наші деталі може бути представлена у вигляді системи нерівностей(рис.6.5.8):

$$\begin{cases} 0 \leq X \leq Dl \\ 0 \leq Y \leq Sh \end{cases} \quad (6.5.21)$$

де Dl -довжина матеріалу, Sh -ширина матеріалу.

Для однозначного відображення деталі на матеріалі необхідно знати таку інформацію:

Nd – код деталі, що розміщується (в нашому випадку це $i = 1..n$);

$Xp_m, Yp_m, m=1..q$ – координати полюса деталі (будь-якої фіксованої точки на деталі) в системі координат, яка пов'язана з матеріалом;

Pr – признак положення деталі (в нашому випадку: 0 – основне положення; 1 – деталь, яка повернута на 180 градусів відносно основного положення; 2 – деталь, яка повернута на 90 градусів відносно основного положення; 3 – деталь, яка повернута на 270 градусів відносно основного

положення). Тоді очевидно, що якщо для $Pr=0$ або $Pr=1$ виконується нерівність

$$\begin{cases} MX_i \leq Xp_m \leq Dl - MX_i \\ MY_i \leq Yp_m \leq Sh - MY_i \end{cases}, \quad (6.5.22)$$

для $Pr=2$ або $Pr=2$ виконується нерівність

$$\begin{cases} MY_i \leq Xp_m \leq Dl - MY_i \\ MX_i \leq Yp_m \leq Sh - MX_i \end{cases}, \quad (6.5.23)$$

то деталь, яка буде розміщена на матеріалі(рис.6.5.8), ніколи не вийде за межі матеріалу.

Для вилучення будь-якої раніш розміщеної деталі із розкрійної схеми потрібно ідентифікувати деталь, яку необхідно вилучити. Для цього треба вирішити, чи знаходиться точка всередині опукло-ввігнутого багатокутника. Найефективніший для розв'язку цієї задачі є метод променя. Нехай нам необхідно визначити чи належать точки A_1 та A_2 заданій деталі. Для цього проведемо з точок A_1 та A_2 в точки, розташовані поза деталлю, кілька променів. Промені перетнуть межу деталі кілька разів

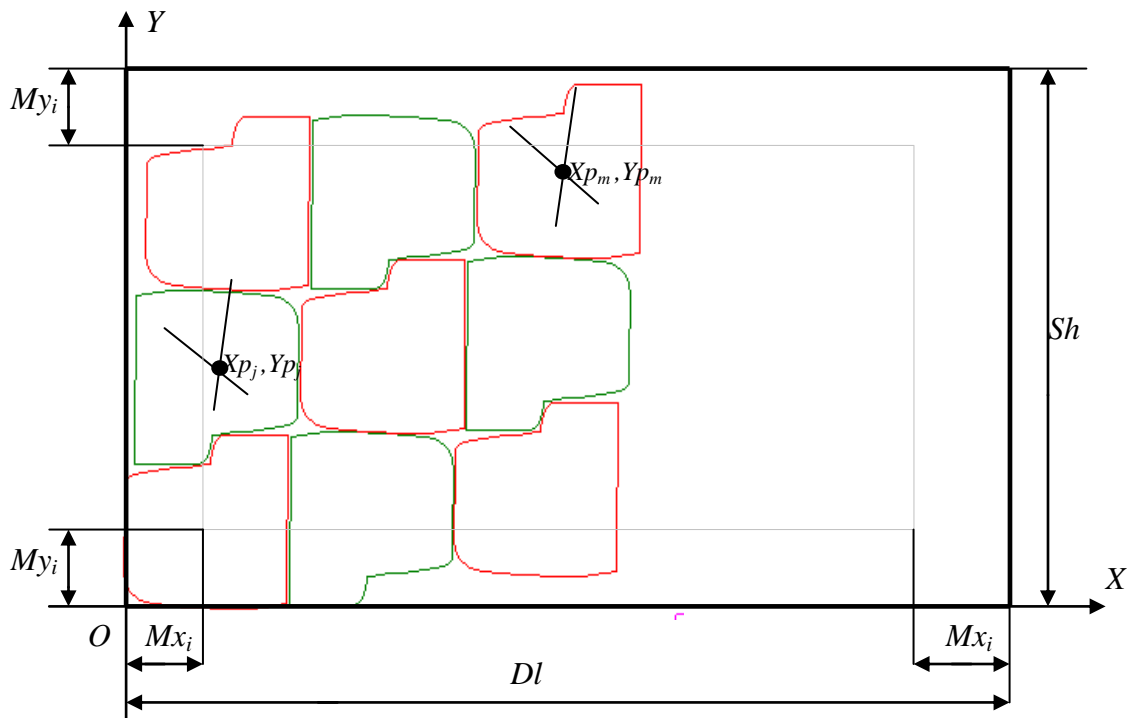


Рис. 6.5.8. Умови неперетину зовнішнього контура деталей із границею матеріалу

або ніразу.

Очевидно, що якщо промінь перетинає деталь непарну кількість разів, то точка лежить на деталі (точка A_1), в іншому випадку точка лежить поза деталлю (точка A_2) (рис.6.5.9).

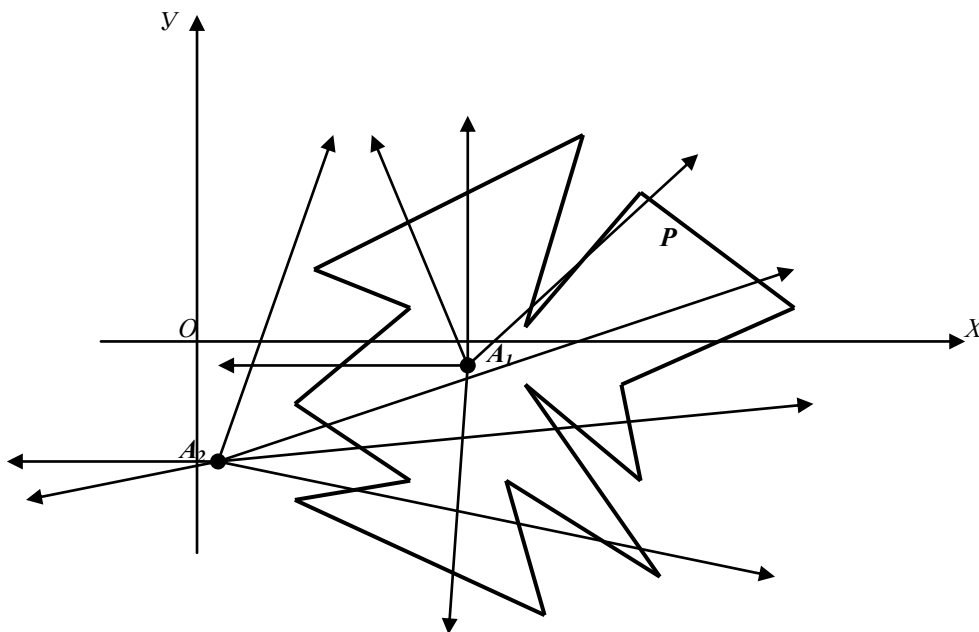


Рис.6.5.9. Перетин деталі променями, які проходять у різних напрямках

Побудова алгоритму на основі цієї ідеї базується на двох особливостях:

- для розв'язку задачі підходить будь-який промінь, який виходить із заданої точки;
- порядок перетину променем зовнішнього контура не має ніякого значення, суттєвою є кількість перетинів променем контура деталі.

Тому виберемо лівий горизонтальний промінь, який починається в заданій точці і спрямований паралельно віссі Ox вліво від точки.

Нехай точка $A_1 (X^1, Y^1)$. Тоді точки перетинів горизонтального променя, який починається в точці A_1 і спрямована паралельно осі OX вліво від заданої точки зі сторонами апроксимуючого многокутника у випадку, коли $Y_{ij+1} \neq Y_{ij}$ визначимо із системи та обмежень (6.5.24):

$$\left\{ \begin{array}{l} y = Y^1 \\ x = X_{ij} + \frac{(Y^1 - Y_{ij})(X_{ij+1} - X_{ij})}{(Y_{ij+1} - Y_{ij})} \text{ де} \end{array} \right. \quad \begin{array}{l} x \leq X^1; T_1 \leq x \leq T_2; Q_1 \leq y \leq Q_2; Y_{ij+1} \neq Y_{ij} \\ T_1 = \min\{X_{ij}, X_{ij+1}\}; Q_1 = \min\{Y_{ij}, Y_{ij+1}\}; \\ T_2 = \max\{X_{ij}, X_{ij+1}\}; Q_2 = \max\{Y_{ij}, Y_{ij+1}\}; \\ i = 0..n; j = 1..k_i \end{array} \quad (6.5.24)$$

а у випадку, коли $Y_{ij+1} = Y_{ij}$ сторона апроксимуючого многокутника буде паралельна даному променю або матиме спільний відрізок із променем. Випадок, коли сторона має спільний відрізок із променем буде розглянутий у особливих випадках, які представлені на рис.6.5. Розглянемо особливі випадки, що можливі при перетину горизонтального променя зі сторонами деталі. У випадках а) та з) будемо вважати, що промінь перетинає дані сторони один раз, у випадку б) та в) – жодного разу, у випадку д) та е) – два рази.

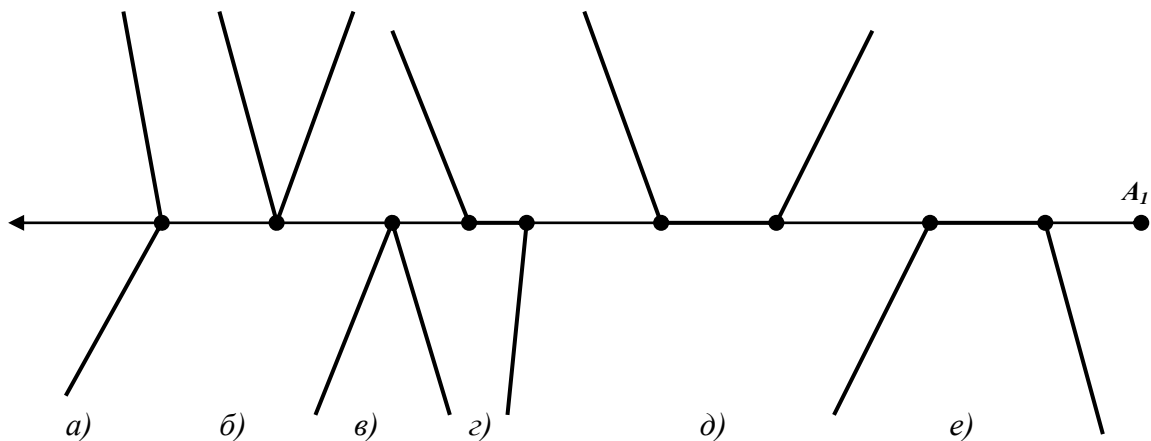


Рис.6.5.10. Особливі випадки перетину сторін деталі горизонтальним променем

Для того, щоб впевнитись, що два апроксимуючі многокутники P_1 та P_2 не перетинаються, потрібно впевнитися, що жодна вершина многокутника P_1 не розташована всередині многокутника P_2 та жодна вершина многокутника P_2 не знаходиться всередині многокутника P_1 . Із

рис. 6.5.11. видно, що необхідною умовою перетину многокутників P_1 та P_2 є перетин описаних навколо них прямокутників, а достатньою умовою неперетину многокутників P_1 та P_2 є неперетин описаних навколо них прямокутників.

Нехай прямокутник $A_1A_2A_3A_4$ визначається системою нерівностей:

$$\begin{cases} X_{\min}^a \leq x \leq X_{\max}^a \\ Y_{\min}^a \leq y \leq Y_{\max}^a \end{cases} \quad (6.5.25)$$

Тоді точка $A_0(X^0, Y^0)$ розташована поза многокутником P_1 , якщо вона не задовольняє системі нерівностей (6.5.25), інакше точка A_0 може належати або не належати многокутнику P_1 . Для в'яснення цього факту потрібно скористатись методом променя і визначити кількість перетинів променем зовнішню границю апроксимуючого многокутника за допомогою системи (6.5.24).

Для прискорення роботи алгоритму визначення взаємного розміщення многокутників P_1 та P_2 розглянемо задачу взаємного розміщення прямокутників $A_1A_2A_3A_4$ та $B_1B_2B_3B_4$, які відповідно описані навколо многокутників P_1 та P_2 . Нехай прямокутник $B_1B_2B_3B_4$ за аналогією з прямокутником $A_1A_2A_3A_4$ визначається системою нерівностей:

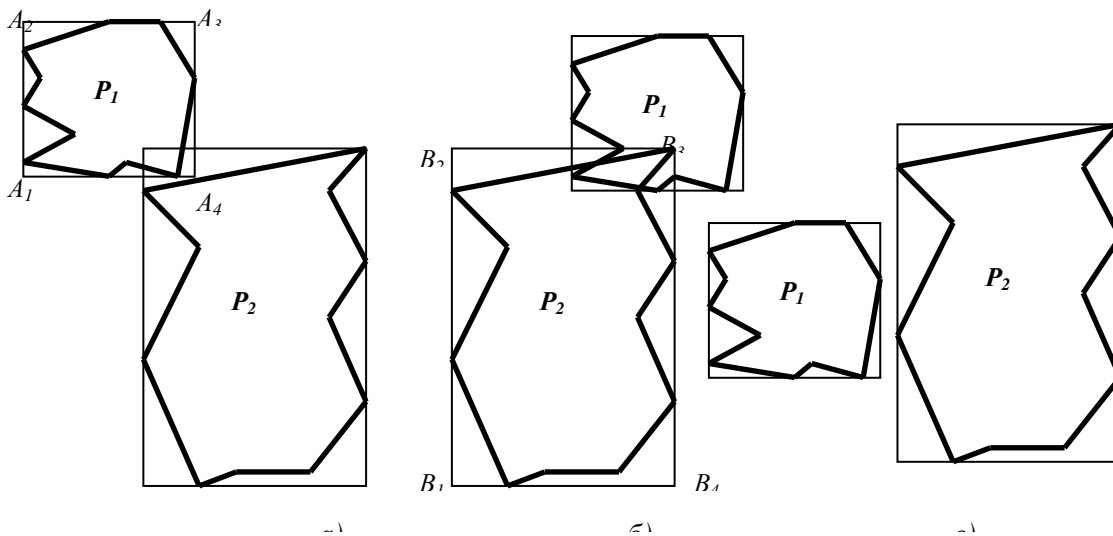


Рис.6.5.11. Взаємне положення деталей P_1 та P_2

- а)прямокутники перетинаються, деталі не перетинаються; б)прямокутники і деталі перетинаються; в)прямокутники і деталі не перетинаються

$$\begin{cases} X_{\min}^b \leq x \leq X_{\max}^b \\ Y_{\min}^b \leq y \leq Y_{\max}^b \end{cases} \quad (6.5.26)$$

Тоді нехай $X_{\min} = \max(X_{\min}^a, X_{\min}^b)$; $Y_{\min} = \max(Y_{\min}^a, Y_{\min}^b)$;
 $X_{\max} = \min(X_{\max}^a, X_{\max}^b)$; $Y_{\max} = \min(Y_{\max}^a, Y_{\max}^b)$. (6.5.27)

Якщо $X_{\min} > X_{\max}$ та $Y_{\min} > Y_{\max}$, то прямокутники перетинаються,

тобто можливо перетинаються багатокутники P_1 та P_2 . Для виявлення цього факту потрібно скористатись методом променя і визначити кількість перетинів променем зовнішню границю апроксимуючого багатокутника за допомогою системи (6.5.24) за описаною раніше методикою. Якщо жодна вершина багатокутника P_1 не знаходиться всередині багатокутника P_2 та жодна вершина багатокутника P_2 не знаходиться всередині багатокутника P_1 , то багатокутники P_1 та P_2 не перетинаються. Інакше багатокутники P_1 та P_2 перетинаються.

Задачу розкрою матеріалів з урахуванням постійного міжшаблонного містка Δ найпростіше розв'язувати в два етапи. На першому етапі побудувати еквідистанти до зовнішніх контурів деталей на відстані $\Delta/2$, а потім будувати схеми розкрою для нових деталей, зовнішнім контуром яких є еквідистанта. Для побудови еквідистанти скористуємося алгоритмом, що запропонований в роботі. При відображенні схем розкрою заміняємо деталь з еквідистантою відповідною вихідною деталлю.

Всі запропоновані вище алгоритми та алгоритми, які розглянуті в цьому розділі, реалізовані в програмний продукт для автоматичної побудови схем розкрою рулонних матеріалів із інтерактивним коригуваннями та інтерактивної побудови схем розкрою натуральних матеріалів на деталі виробів дрібної шкіргалантереї в інтегрованому середовищі програмування Delphi для операційної системи Windows. Програмний продукт має зручний інтерфейс і не потребує

спеціальних знань із компютерних наук для роботи з ним. Він може бути використаний при побудові розкрійних схем для рулонних та листових матеріалів. Структурна схема програмного продукту для інтерактивної побудови схем розкрою натуральних матеріалів на деталі виробів дрібної шкіргалантереї представлена на рис.6.5.14.

Висновки

1. Розроблені алгоритми та програмне забезпечення, дозволяє автоматизувати побудову розкрійних схем рулонних матеріалів на деталі шкіргалантерейних виробів.

2. Розроблені алгоритми, які дозволяють автоматизувати підготовку раціанольних схем розкрою на деталі шкіргалантерейних виробів, а саме алгоритми:

- автоматичного контролю неперетину деталі із деталлю;
- автоматичного контролю неперетину деталі із зовнішньою границею матеріалу будь-якої складної форми та із зонами пороків;
- вилучення будь-якої деталі із розкрійної схеми;
- обчислення площі для матеріала будь-якої складної форми із пороками.

6.5.Розроблений програмний продукт дозволить звільнити технолога від рутинної роботи, збільшити продуктивність праці та автоматизувати розкрійне виробництво.

4.Із використанням алгоритмів, представлених у цьому розділу зявляється можливість автоматизувати процес розкрою із застосуванням розкрійних пристроїв із програмним управлінням.

5.Запропоновані алгоритми можуть бути використані для розробки програмним продуктів для автоматичної та інтерактивної побудови розкрійних схем та виводу креслень плоских деталей в інших галузях промисловості.

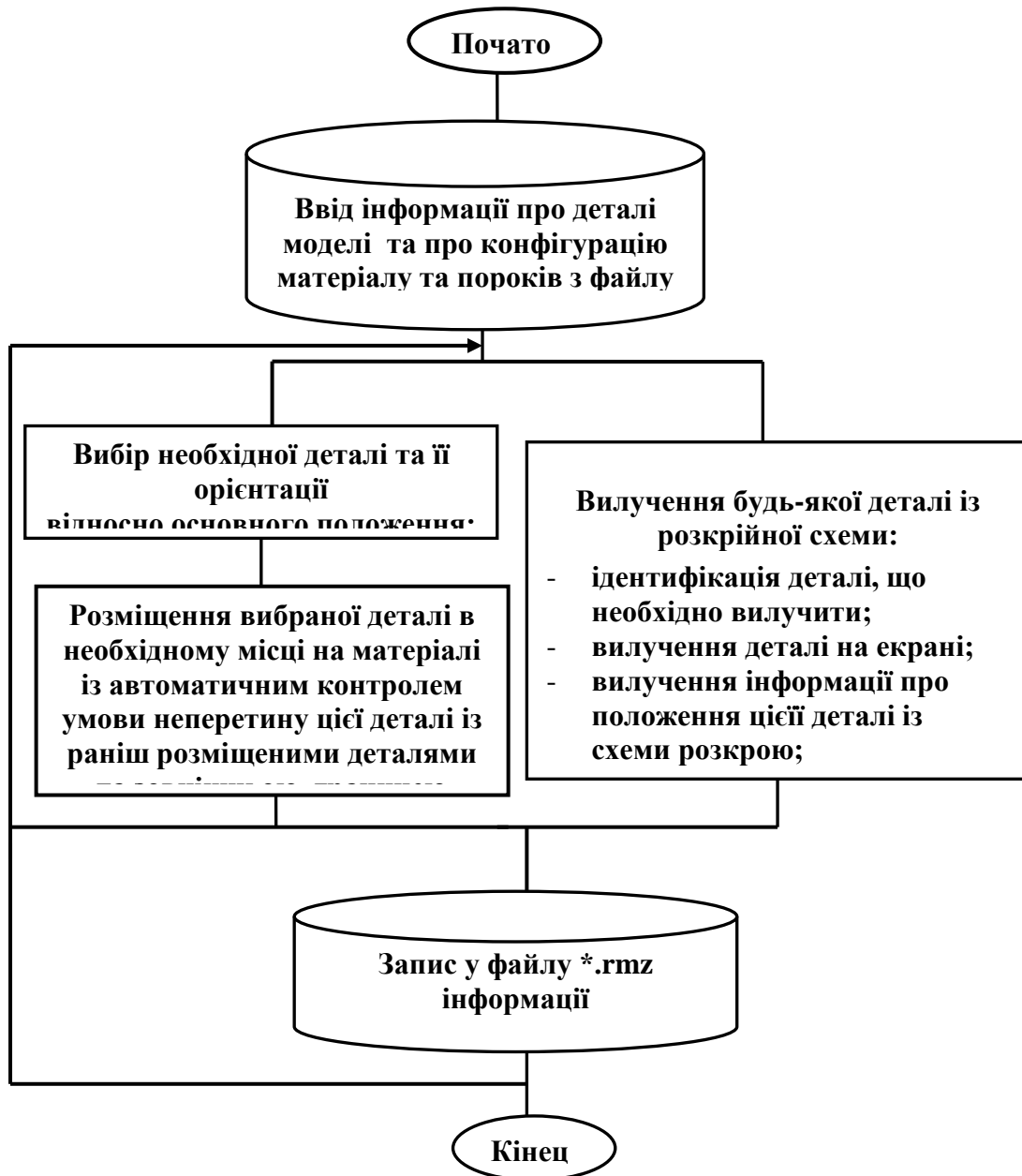


Рис. 6.5.14.
Структурна схема програмного продукту інтерактивної побудови розкрійних схем

7. УДОСКОНАЛЕННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ В 'ЯЗАНЬ ТА ТКАЦТВА З УРАХУВАННЯМ АНІЗОТРОПІЇ ФРИКЦІЙНИХ ВЛАСТИВОСТЕЙ НИТКИ

Поперечне ковзання ниток зустрічається в багатьох технологічних процесах текстильної та трикотажної промисловості. При цьому можна поділяти поперечне ковзання по направляючим великої і малої кривизни. Перший випадок можна розглядати як ковзання нитки по поверхні, у якої радіус кривизни дорівнює нескінченності. У подальших дослідженнях, у розрахунках, будемо вважати, що направляюча поверхня має постійний радіус кривизни.

Поперечне ковзання нитки по плоскій поверхні зустрічається, наприклад, при прибої утокової нитки, коли вона ковзає по поверхні основних ниток. Під дією сил тертя відбувається її втягування між зубами берда. Це спостерігається при використанні берд низьких номерів, при великій кількості ниток основи ввібраних в один зуб берда і при роботі з заступом. Цей процес призводить до збільшення довжини утокової нитки по ширині заправки ткацького верстата, що не дозволяє отримувати тканини заданої поверхневої щільності.

7.1 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ПОПЕРЕЧНОГО РУХУ НИТОК З УРАХУВАННЯМ АНІЗОТРОПІЇ ТЕРТЯ

У разі коли відстань між зубами берда сумірна з розрахунковим радіусом поперечного перерізу нитки, а також при використанні монониток, необхідно в розрахунках враховувати фізико-механічні характеристики нитки: жорсткість нитки на вигин, м'яття в зоні контакту з направляючою.

Поперечне ковзання ниток по направляючим з постійною кривизною поверхні широко зустрічається в трикотажній промисловості, коли відбувається відносне переміщення голок і ниток один щодо

одного. При цьому має місце і поздовжній рух нитки щодо своєї осі, що пояснюється перетяжками нитки зі старих петель в нові.

При взаємодії ниток з направляючими трикотажних машин, коли радіус їх кривизни сумарний з розрахунковим радіусом нитки також необхідно враховувати фізико-механічні властивості ниток.

Теоретичні дослідження процесу взаємодії ниток з направляючими при поперечному ковзанні ставлять перед собою мету визначення форми осі нитки та її натяг в довільній точці, що дозволить удосконалити процес її переробки на технологічному обладнанні.

**Дослідження поперечного руху нитки по площині з
урахуванням анізотропії тертя у випадку, коли розрахунковий радіус
поперечного перерізу нитки незрівнянно малий в порівнянні з
відстанню між опорами**

На рис.7.1.1 показана загальна розрахункова схема. Нитка закріплюється в точках А і В. Тут необхідно зазначити, що ці точки дозволяють нитці рухатися уздовж своєї осі. У довільній точці М на нитку буде діяти сила тертя, сила тяжіння, нормальна реакція опори.

На рис.7.1.2 показаний нескінченно малий елемент нитки довжиною ds з діючими на нього силами. У центрі цього елемента (точка М) розміщуємо натуральний тригранник, який складають дотична, нормальна і бінормальна осі. Їх напрямок визначається відповідними одиничними ортами $\vec{t}, \vec{v}, \vec{\beta}$. Використання натурального тригранника вигідно відрізняється від декартової координатної системи, оскільки дозволяє отримати системи диференціальних рівнянь в найбільш простій формі. Дотична вісь в точці М становить поточний кут φ з позитивним напрямом осі x . Проектуємо сили, що діють на елементарний елемент

нитки, на осі натурального тригранника з урахуванням позитивного напрямку одиничних ортів, отримаємо

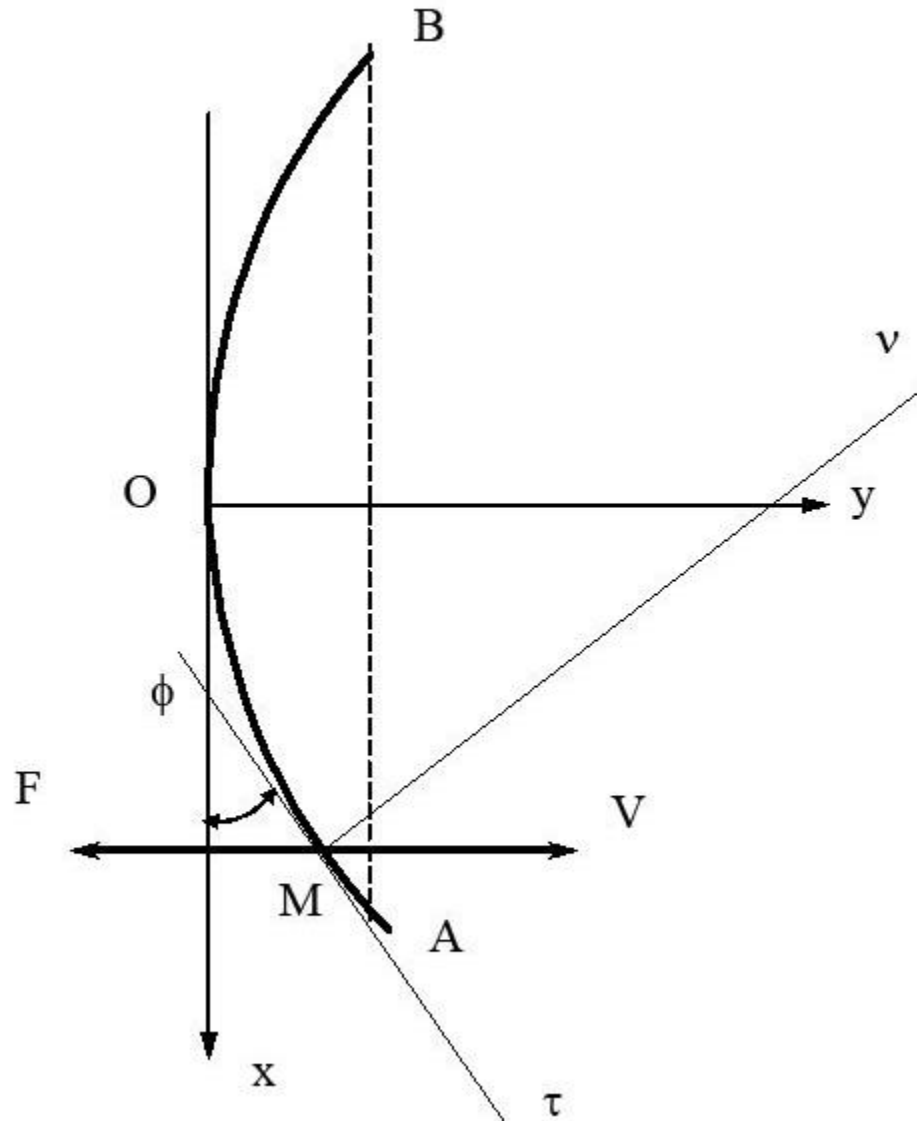


Рис.7.1.1. Загальна розрахункова схема

$$\begin{aligned} \sum_{i=1}^n R_{\alpha} &= -P \cos \frac{d\alpha}{2} + P \cos \frac{d\alpha}{2} + dP \cos \frac{d\alpha}{2} - F_{\tau} ds = 0, \\ \sum_{i=1}^n R_{\beta} &= P \sin \frac{d\alpha}{2} + P \sin \frac{d\alpha}{2} + dP \sin \frac{d\alpha}{2} - F_{\nu} ds = 0, \\ \sum_{i=1}^n R_{\beta_i} &= N ds - \mu_0 g ds = 0, \end{aligned} \quad (7.1.1)$$

де $R_{\alpha}, R_{\nu}, R_{\beta}$ - алгебраїчні суми проєкцій сил, що діють на елементарний відрізок нитки, відповідно на дотичну, нормаль і бінормаль;

P - натяг нитки, Н;

$d\alpha$ - значення центрального кута для нескінченно малого елемента ds , рад;

F_{τ}, F_{ν} - проєкції сили тертя F на дотичну і нормаль;

s - дугова координата, м;

N - питома нормальна реакція, Н;

μ_0 - лінійна щільність нитки до розтягування, кг / м;

g - прискорення вільного падіння, м/с².

Зважаючи на порядок невеликого розміру елементарного відрізка нитки ds замінюємо $\sin \frac{d\alpha}{2} \cong \frac{d\alpha}{2}$, $\cos \frac{d\alpha}{2} \cong 1$, тоді система (7.1.1) прийме

вигляд

$$\begin{aligned} dP - F_{\tau} ds &= 0, \\ Pd\alpha - F_{\nu} ds &= 0, \\ N &= \mu_0 g. \end{aligned} \tag{7.1.2}$$

З урахуванням розтягування нитки, за законом збереження маси можна записати наступну пропорцію

$$\frac{ds}{ds_0} = \frac{\mu_0}{\mu} = f, \tag{7.1.3}$$

де ds_0 - довжина елемента нитки до розтягування, м;

ds - довжина елемента нитки після розтягнення, м;

μ - лінійна щільність елемента нитки після розтягнення, кг/м;

f - коефіцієнт, який характеризує розтяжність нитки.

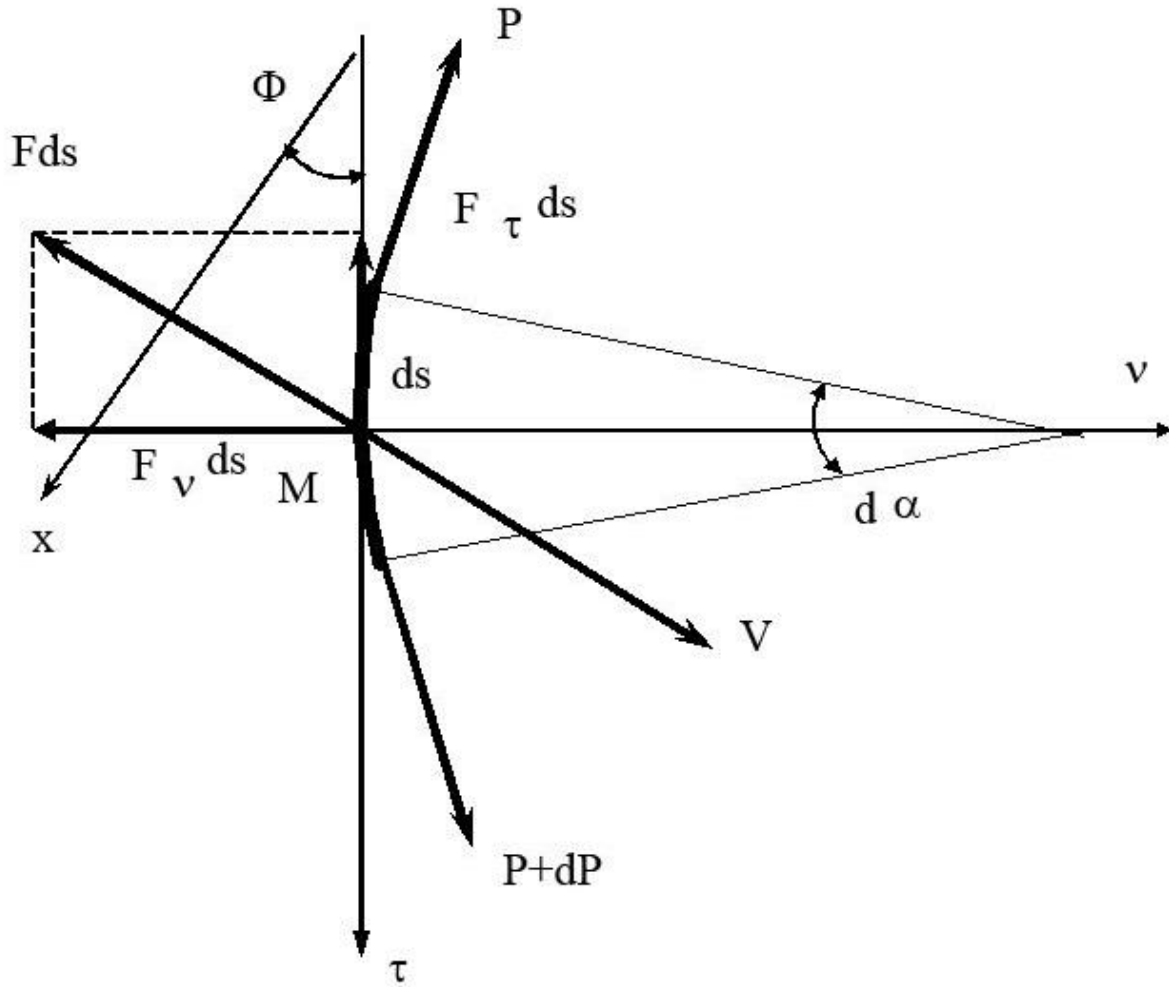


Рис.7.1.2. Схема сил, що діють на елементарний відрізок нитки

Перетворимо систему диференціальних рівнянь (7.1.2), розділивши ліві і праві частини на ds

$$\begin{aligned} \frac{dP}{ds} &= F_{\tau}, \\ \frac{Pd\alpha}{ds} &= F_v, \\ N &= \mu_0 g. \end{aligned} \quad (7.1.4)$$

Величина

$$\rho = \frac{d\alpha}{ds}, \quad (7.1.5)$$

являє собою радіус кривизни. Тоді з урахуванням (7.1.3) система диференціальних рівнянь (7.1.4) матиме вигляд

$$\begin{aligned}\frac{f}{\mu_0} \frac{dP}{ds} &= F_\tau, \\ \frac{f}{\mu_0} \frac{P}{\rho} &= F_v, \\ N &= \mu_0 g.\end{aligned}\tag{7.1.6}$$

Складові сили тертя визначаються за наступними формулами

$$\begin{aligned}F_\tau &= F_1 \sin \varphi, \\ F_v &= F_2 \cos \varphi,\end{aligned}\tag{7.1.7}$$

де φ - кут нахилу дотичної в точці М до осі x , рад.

Величини, що складають силу тертя F_1, F_2 визначаються за формулами

$$F_1 = k_1 N, \quad F_2 = k_2 N,\tag{7.1.8}$$

де k_1, k_2 - відповідно коефіцієнти тертя нитки об направляючу в напрямку дотичної та нормалі.

Тоді, з урахуванням виразів (7.1.7) - (7.1.8), система диференціальних рівнянь (7.1.6) прийме вигляд

$$\begin{aligned}\frac{f}{\mu_0} \frac{dP}{ds} &= k_1 N \sin \varphi, \\ \frac{f}{\mu_0} \frac{P}{\rho} &= k_2 N \cos \varphi, \\ N &= \mu_0 g.\end{aligned}$$

Використовуючи третє рівняння даної системи, отримаємо

$$\begin{aligned}\frac{f}{\mu_0} \frac{dP}{ds} &= k_1 \mu_0 g \sin \varphi, \\ \frac{f}{\mu_0} \frac{P}{\rho} &= k_2 \mu_0 g \cos \varphi.\end{aligned}\tag{7.1.9}$$

У випадку, коли можна знехтувати розтягуванням нитки ($f = 1$), одержимо

$$\begin{aligned} \frac{dP}{ds} &= k_1 \mu_0 g \sin \varphi, \\ \frac{P}{\rho} &= k_2 \mu_0 g \cos \varphi. \end{aligned} \quad (7.1.10)$$

Розділимо перше рівняння системи (7.1.10) на друге

$$\frac{dP\rho}{Pds} = \frac{k_1 \sin \varphi}{k_2 \cos \varphi}, \quad (7.1.11)$$

При розв'язуванні диференціального рівняння (7.1.11) будемо враховувати, що

$$\begin{aligned} \rho &= \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}}, \quad \sin \varphi = \frac{\frac{dy}{dx}}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}}, \\ ds &= \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx, \quad \cos \varphi = \frac{1}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}}, \end{aligned}$$

де x, y - координати точки M , m .

Підставляємо останні залежності в рівняння (7.1.11), отримаємо

$$\frac{dP \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \frac{d^2y}{dx^2} P} = \frac{k_1 dy \sqrt{1 + \left(\frac{dy}{dx}\right)^2}}{k_2 dx \sqrt{1 + \left(\frac{dy}{dx}\right)^2}}.$$

Після перетворень отримаємо наступне диференціальне рівняння

$$\frac{dP \left[1 + \left(\frac{dy}{dx}\right)^2\right]}{P} = \frac{k_1 dy}{k_2 dx} \frac{d^2y}{dx^2} dx.$$

Або

$$\frac{dP}{P} = \frac{k_1}{k_2} \frac{\frac{dy}{dx} d\left(\frac{dy}{dx}\right)}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]} \quad (7.1.12)$$

Інтегруємо останнє диференціальне рівняння, з урахуванням $\frac{dy_0}{dx_0} = 0$
(дотична в точці О паралельна осі x)

$$\int_{P_0}^P \frac{dP}{P} = \frac{k_1}{k_2} \int_0^{\frac{dy}{dx}} \frac{\frac{dy}{dx} d\left(\frac{dy}{dx}\right)}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]}$$

Тоді

$$\ln \frac{P}{P_0} = \frac{k_1}{2k_2} \ln \left[1 + \left(\frac{dy}{dx}\right)^2\right]$$

В остаточному вигляді

$$P = P_0 \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{k_1}{2k_2}} \quad (7.1.13)$$

Підставляємо вираз (7.1.13) у друге рівняння системи (7.1.10)

$$\frac{P_0 \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{k_1}{2k_2}} \frac{d^2y}{dx^2}}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}} = \frac{k_2 \mu_0 g}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}}$$

Після перетворень отримаємо

$$\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{k_1}{2k_2} - 1} \frac{d^2 y}{dx^2} = \frac{k_2 \mu_0 g}{P_0}. \quad (7.1.14)$$

Перетворимо диференціальне рівняння (7.1.14) до виду

$$\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{k_1}{2k_2} - 1} d\left(\frac{dy}{dx}\right) = \frac{k_2 \mu_0 g}{P_0} dx. \quad (7.1.15)$$

У випадку, коли $k_1 = k_2$, отримуємо диференціальне рівняння, яке зводиться до еліптичних інтегралів 1-го роду (форма осі буде ланцюгова лінія).

$$\int_0^{\frac{dy}{dx}} \frac{d\left(\frac{dy}{dx}\right)}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{1}{2}}} = \frac{k_2 \mu_0 g}{P_0} x.$$

У нашому випадку, для вирішення диференціального рівняння (7.1.14) необхідно використовувати чисельне інтегрування. Опис алгоритму, структури програми наведено в третьому розділі.

Чисельне інтегрування дозволить отримати масив даних, які апроксимувались з використанням програмного забезпечення написаного на мові Object Pascal в середовищі Delphi.

Для чисельного інтегрування використовувався метод Рунге-Кутта-Мерсона з автоматичним вибором кроку інтегрування (програма наведена в Додатку до дисертації). Даний метод має більшу стійкість до виникнення нестійкості рішення.

Для апроксимації використовувалася поліноміальна апроксимація з автоматичним вибором ступеня полінома (програма наведена в Додатку до дисертації).

Визначення натягу і форми пружної на вигин нитки при поперечному ковзанні

Цей вид взаємодії відбувається, коли розрахунковий радіус поперечного перерізу нитки сумарний з відстанню між направляючими дугами А і В (див.рис.7.1.1). У цьому випадку необхідно враховувати згинальну жорсткість нитки. При виведенні залежностей будемо вважати, що згинальний момент пропорційний кривизні осі нитки, а швидкість зміни величини перерізуючої сили по дуговій координаті дуже мала, що пояснюється незначною величиною відхилення осі нитки від лінії з'єднує точки А і В. Крім того, будемо вважати, що маса нитки рівномірно розподілена по її довжині.

На рис.7.1.2.1 представлена відповідна розрахункова схема.

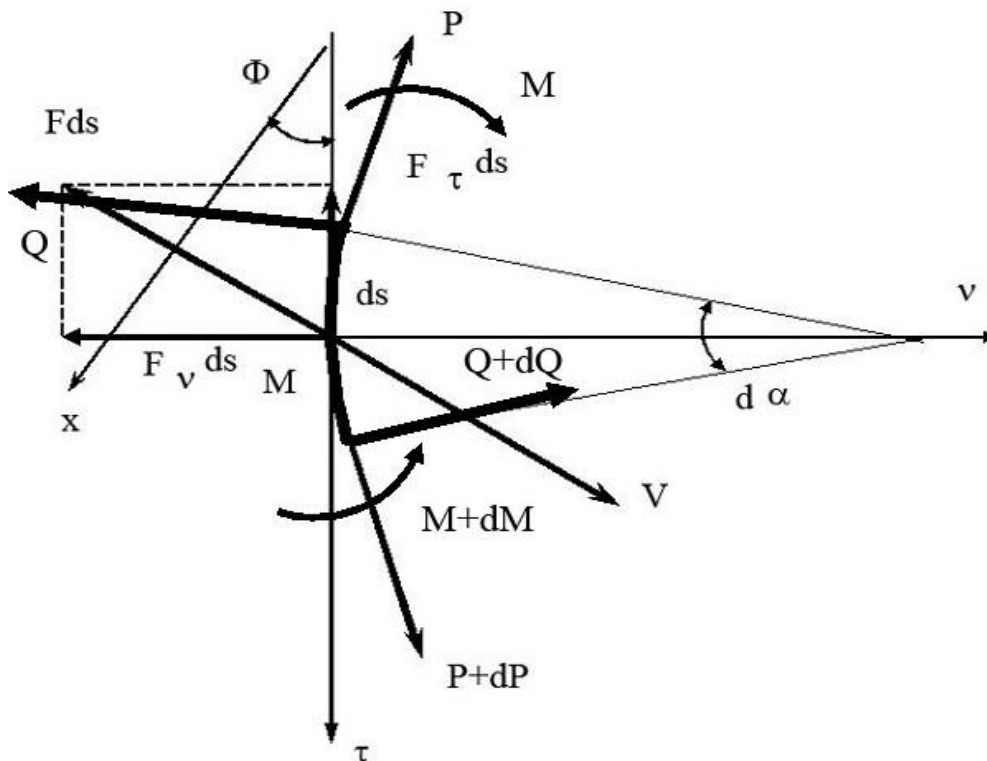


Рис.2.7.1. Розрахункова схема для визначення натягу і форми жорсткої на вигин нитки з урахуванням анізотропії фрикційних властивостей

З урахуванням анізотропії фрикційних властивостей нитки, спроектуємо всі діючі на елементарний відрізок нитки сили на осі натурального тригранника, а також складемо рівняння рівноваги моментів усіх сил відносно точки центру елементарного відрізка.

У результаті одержимо наступну систему диференціальних рівнянь

$$\begin{aligned} \frac{dP}{ds} - qQ - F_{\tau} &= 0, \\ \frac{dQ}{ds} + Pq - F_{\nu} &= 0, \\ \frac{dM}{ds} + Q &= 0, \quad M = Bq, \\ N = \mu_0 g, \quad q &= \frac{d\varphi}{ds}, \end{aligned} \tag{7.1.2.1}$$

де P - натяг нитки, Н;

s - дугова координата, м;

q - кривизна осі нитки, 1/м;

F_{τ}, F_{ν} - проекції сили тертя на осі натурального тригранника;

Q - перерізуюча сила, Н;

M - згинальний момент, Нм;

B - коефіцієнт згинальної жорсткості, Нм²;

φ - кут нахилу дотичної в точці M до осі x , рад.;

N - питома нормальна реакція поверхні, Н;

μ_0 - лінійна щільність нитки, кг/м;

g - прискорення вільного падіння, м/с².

При вирішенні завдання будемо вважати, що нитка нерозтяжна ($f = 1$).

Складові сили тертя визначаються за наступними формулами

$$\begin{aligned} F_{\tau} &= F_1 \sin \varphi, \\ F_{\nu} &= F_2 \cos \varphi. \end{aligned} \tag{7.1.2.2}$$

Величини складових сили тертя визначаються за формулами

$$F_1 = k_1 N, \quad F_2 = k_2 N, \quad (7.1.2.3)$$

де k_1, k_2 - коефіцієнти тертя відповідно при русі в напрямку дотичної і нормальної осей.

Тоді, з урахуванням (7.1.2.2) - (7.1.2.3), система диференціальних рівнянь (7.1.2.1) прийме вигляд

$$\begin{aligned} \frac{dP}{ds} - qQ - k_1 \mu_0 g \sin \varphi &= 0, \\ \frac{dQ}{ds} + Pq - k_2 \mu_0 g \cos \varphi &= 0, \\ \frac{dM}{ds} + Q &= 0, \quad M = Bq, \\ N = \mu_0 g, \quad q &= \frac{d\varphi}{ds}, \end{aligned} \quad (7.1.2.4)$$

Коефіцієнт згинальної жорсткості визначається за формулою

$$B = EI,$$

де E - модуль пружності нитки при розтягуванні, Н/м²;

I - момент інерції перерізу нитки, м⁴;

З урахуванням останнього виразу система диференціальних рівнянь (7.1.2.4) прийме вигляд

$$\begin{aligned} \frac{dP}{ds} - qQ - k_1 \mu_0 g \sin \varphi &= 0, \\ \frac{dQ}{ds} + Pq - k_2 \mu_0 g \cos \varphi &= 0, \\ \frac{dM}{ds} + Q &= 0, \quad M = EIq, \\ N = \mu_0 g, \quad q &= \frac{d\varphi}{ds}, \end{aligned} \quad (7.1.2.5)$$

З третього рівняння системи (7.1.2.5) отримуємо

$$Q = -\frac{dM}{ds}. \quad (7.1.2.6)$$

Підставляємо (7.1.2.6) в четверте рівняння системи диференціальних рівнянь (7.1.2.5) (тут вважаємо, що коефіцієнт згинальної жорсткості нитки на вигин є постійною величиною)

$$Q = -\frac{Bdq}{ds}. \quad (7.1.2.7)$$

Отриманий результат підставляємо в перше рівняння системи (7.1.2.5), отримуємо

$$\frac{dP}{ds} + \frac{Bqdq}{ds} - k_1\mu_0g \sin \varphi = 0.$$

Помножимо ліву і праву частини останнього диференціального рівняння на ds

$$dP + Bqdq - k_1\mu_0g \sin \varphi ds = 0. \quad (7.1.2.8)$$

У рівнянні (7.1.2.8) рівняння

$$\sin \varphi ds = dy.$$

З урахуванням цього матимемо

$$dP + Bqdq - k_1\mu_0g dy = 0.$$

Зінтегруємо останнє диференціальне рівняння

$$P = C - B\frac{q^2}{2} + k_1\mu_0gy, \quad (7.1.2.9)$$

де C - довільна стала інтегрування, H .

При наступних початкових умовах (в точці O -

рис.7.1.1) $P = P_0$, $y = y_0$, $q = q_0$, отримаємо

$$C = P_0 + B\frac{q_0^2}{2}.$$

Підставляючи останній вираз в (7.1.2.9), отримаємо

$$P = P_0 + \frac{B}{2}(q_0^2 - q^2) + k_1\mu_0gy. \quad (7.1.2.10)$$

У другому рівнянні системи (7.1.2.5) можна знехтувати першим членом в лівій частині (швидкість зміни перерізуючої сили по дуговій координаті незначна). Отже

$$P = \frac{k_2 \mu_0 g \cos \varphi}{q}. \quad (7.1.2.11)$$

Підставляємо отриманий результат в перше рівняння системи (7.1.2.5)

$$\frac{k_2 \mu_0 g d(\cos \varphi)}{q ds} - \frac{k_2 \mu_0 g \cos \varphi dq}{q^2 ds} + \frac{Bq^2 dq}{ds} - k_1 \mu_0 g \sin \varphi = 0.$$

Перетворимо останній вираз до наступного вигляду

$$-k_2 \mu_0 g \sin \varphi - \frac{k_2 \mu_0 g \cos \varphi dq}{q d\varphi} + \frac{Bq^2 dq}{d\varphi} - k_1 \mu_0 g \sin \varphi = 0.$$

Згрупуємо в останньому рівнянні подібні члени

$$-(k_1 + k_2) \mu_0 g \sin \varphi - \frac{k_2 \mu_0 g \cos \varphi dq}{q d\varphi} + \frac{Bq^2 dq}{d\varphi} = 0.$$

Множачи праву і ліву частини останнього рівняння на $d\varphi$, розділимо на $\cos \varphi$ і $\mu_0 g$, отримаємо

$$-(k_1 + k_2) \sin \varphi - \frac{k_2 q}{q \cos \varphi} + \frac{Bq^2 dq}{\mu_0 g \cos \varphi} = 0.$$

Після інтегрування одержимо

$$\ln(\cos \varphi)^{(k_1+k_2)} - \ln q^{k_2} + \frac{Bq^3}{3\mu_0 g \cos \varphi} = \ln C_1, \quad (7.1.2.12)$$

де C_1 довільна постійна інтегрування. Тут необхідно зазначити, що при інтегруванні останнього члена використовувалася перша теорема про середню. Значення $\cos \varphi_*$ досліджується на інтервалі від 0 до кута нахилу дотичної в точці закріплення А (отримуємо $\varphi_* = 0$).

Визначаємо довільну постійну інтегрування при наступних граничних умовах при $\varphi = 0 \quad q = q_0$.

Тоді з (7.1.2.12) отримаємо

$$\ln C_1 q_0^{k_2} = \frac{Bq_0^3}{3\mu_0 g},$$

або

$$C_1 q_0^{k_2} = e^{\frac{Bq_0^3}{3\mu_0 g}}.$$

Остаточно

$$C_1 = \frac{e^{\frac{Bq_0^3}{3\mu_0 g}}}{q_0^{k_2}}. \quad (7.1.2.13)$$

Підставляємо (7.1.2.13) в рівняння (7.1.2.12)

$$\ln \frac{\cos \varphi^{(k_1+k_2)} q_0^{k_2}}{q^{k_2} e^{\frac{Bq_0^3}{3\mu_0 g}}} = -\frac{Bq_0^3}{3\mu_0 g}. \quad (7.1.2.14)$$

Перетворимо рівняння (7.1.2.14) у вигляд

$$\frac{\cos \varphi^{(k_1+k_2)} q_0^{k_2}}{q^{k_2}} = \frac{e^{\frac{Bq_0^3}{3\mu_0 g}}}{e^{\frac{Bq_0^3}{3\mu_0 g}}}.$$

Остаточно

$$\cos \varphi = \sqrt{(k_1+k_2)} \frac{q^{k_2}}{q_0^{k_2}} e^{\frac{B}{3\mu_0 g} (q_0^3 - q^3)}. \quad (7.1.2.15)$$

Якщо в рівнянні (7.1.2.15) знехтувати анізотропією тертя і жорсткістю на вигин, то отримаємо класичне рівняння ланцюгової лінії.

На рис.7.1.2.2 представлені розрахунки залежності кута нахилу дотичної і натягу нитки від величини кривизни осі нитки. Розрахунки виконувалися для наступних значень: $q_0=1 \text{ м}^{-1}$, $B=17 \cdot 10^{-8} \text{ Нм}^2$, $\mu_0 g=3 \cdot 10^{-3} \text{ Н/м}$, $k_1=0,1$, $k_2=0,35$.

Аналіз представлених графічних залежностей показує, що зі зменшенням кривизни (збільшенням радіуса кривизни) кут нахилу дотичної до осі x зростає. Така ж тенденція спостерігається і з натягом.

Дослідження поперечного ковзання нитки по направляючій поверхні малої кривизни з урахуванням анізотропії тертя

Цей вид взаємодії нитки з направляючою поверхнею зустрічається в багатьох технологічних процесах текстильної та трикотажної промисловості. Тут мається на увазі, що радіус кривизни направляючої поверхні значно перевищує значення розрахункового радіуса поперечного перерізу нитки.

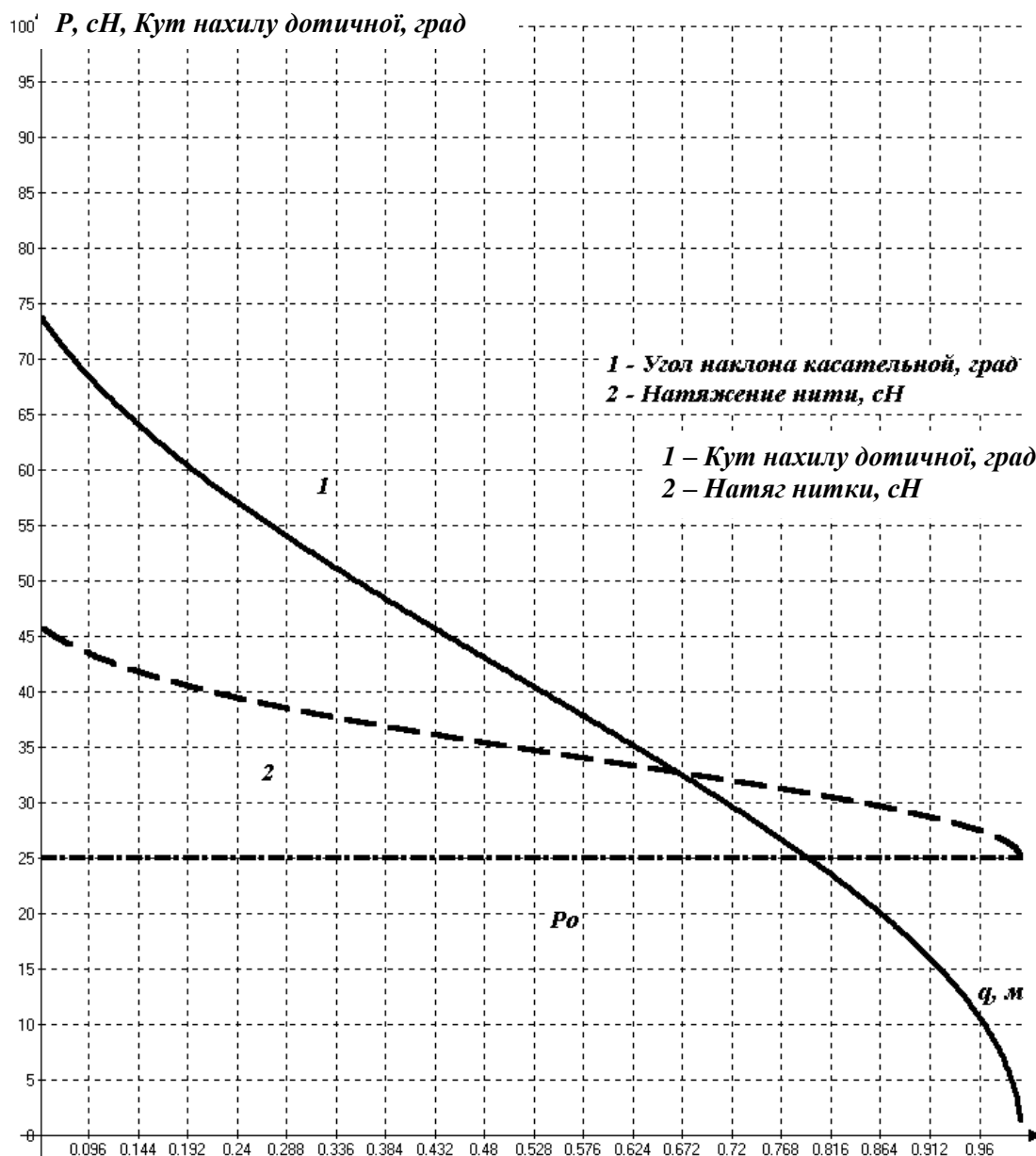


Рис.7.1.2.2. Графічні залежності зміни кута нахилу дотичної та натягу від кривизни

В якості допущення будемо вважати, що направляюча поверхня являє собою циліндричну поверхню.

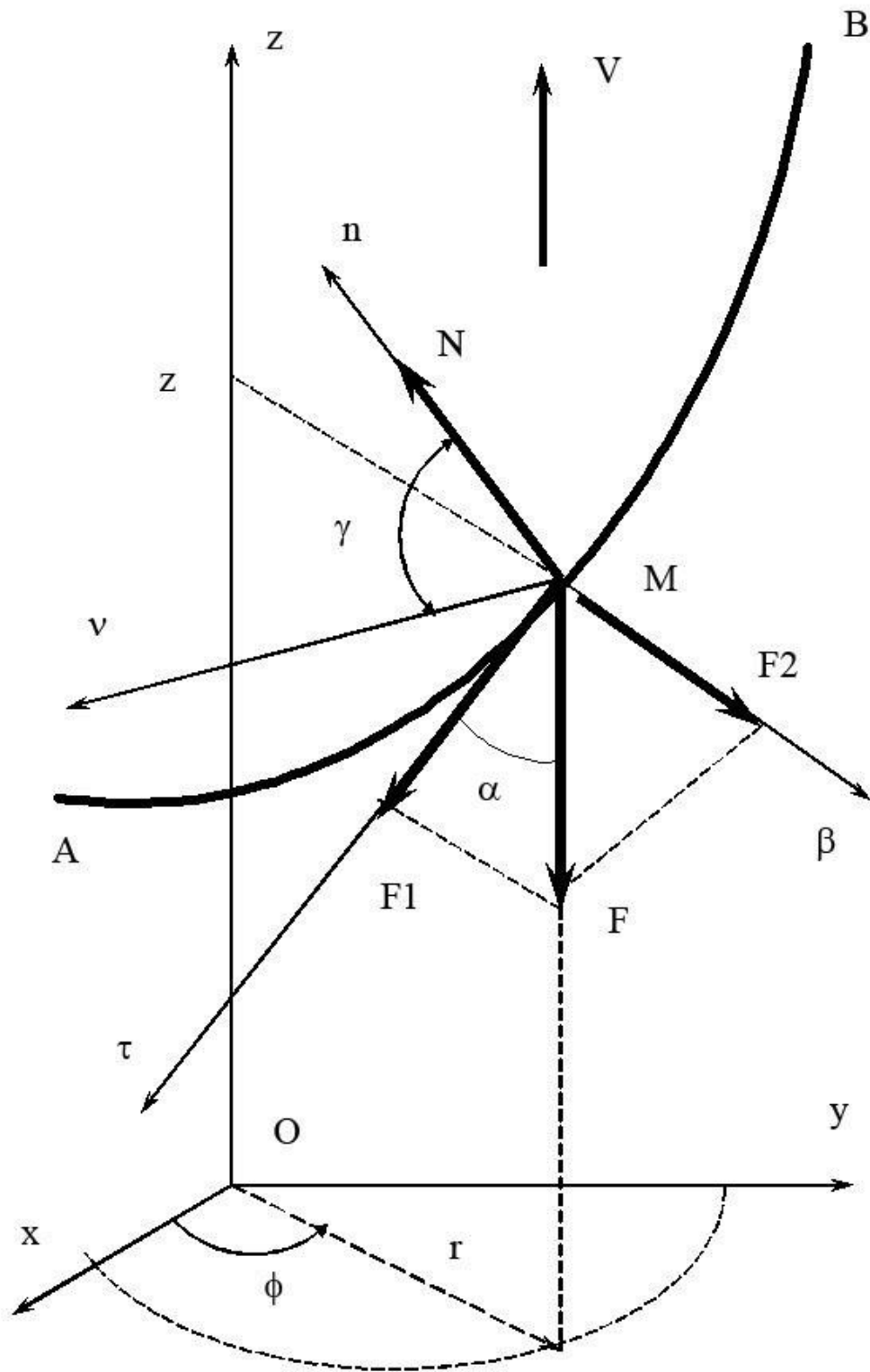


Рис.7.1.3.1. Розрахункова схема

Також будемо вважати, що силами ваги елемента нитки можна знехтувати (натяг нитки, сили тертя перевищують силу ваги елемента нитки на один два порядки). На рис.7.1.3.1 представлена розрахункова схема.

Мета даного дослідження - визначити натяг і форму осі нитки як функцій дугової координати s . Виділяємо на ділянці нитки нескінченно малий елемент ds з точкою M у центрі його ваги.

При поперечному ковзанні по циліндричній поверхні на нього діятиме повна реакція, віднесена до одиниці довжини (надалі для простоти просто сила або реакція) Rds , яка розкладається на дві складові: нормальну реакцію поверхні Nds і силу тертя Fds . Тут необхідно зазначити, що вектор сили тертя завжди буде спрямований протилежно вектору швидкості.

Сила тертя і одиничний орт дотичної τ розташовуються в дотичній площині. Натуральний тригранник розташовується таким чином, щоб і одиничний орт бінормалі також розташовувався в дотичній площині. Кут між нормаллю натурального тригранника і нормаллю n до поверхні в точці M (кут геодезичного відхилення) позначимо γ .

Найбільш зручною системою координат, в даному випадку, буде циліндрична координатна система. В якості координат використовуються r, φ, z . Зв'язок між декартовими координатами і циліндричними показаний на рис.7.1.3.1.

Сила тертя, розташована в дотичній площині, визначається з наступної системи рівнянь

$$\vec{F} = \vec{F}_1 + \vec{F}_2, \quad F = \sqrt{F_1^2 + F_2^2}, \quad (7.1.3.1)$$

де $F_1 = k_1 N$ - проекція сили тертя F на дотичну вісь натурального тригранника;

$F_2 = k_2 N$ - проекція сили тертя F на бінормальну вісь натурального тригранника;

k_1, k_2 - коефіцієнти тертя нитки в подовжньому і поперечному напрямку, котрі характеризують анізотропію фрикційних властивостей.

Тоді величину сили тертя, з урахуванням (7.1.3.1), можна визначити за наступною формулою

$$F = N\sqrt{k_1^2 + k_2^2}. \quad (7.1.3.2)$$

Основна система диференціальних рівнянь в циліндричних координатах, що описує поперечне ковзання нитки по направляючій з урахуванням анізотропії фрикційних властивостей, буде мати вигляд

$$\begin{aligned} \frac{d}{ds} \left(P \frac{dr}{ds} \right) - Pr \left(\frac{d\varphi}{ds} \right)^2 + N &= 0, \\ \frac{d}{ds} \left(Pr \frac{d\varphi}{ds} \right) &= 0, \\ \frac{d}{ds} \left(P \frac{dz}{ds} \right) - \sqrt{k_1^2 + k_2^2} N &= 0, \\ \left(\frac{dr}{ds} \right)^2 + r^2 \left(\frac{d\varphi}{ds} \right)^2 + \left(\frac{dz}{ds} \right)^2 &= 1, \end{aligned} \quad (7.1.3.3)$$

де s - дугова координата, м;

P - натяг нитки, Н;

r - радіус поперечного перерізу направляючої поверхні, м;

φ - циліндрична координата, яка визначає кут повороту радіуса r по відношенню до позитивного напрямку осі x (див.рис.7.1.3.1), рад;

N - нормальна реакція поверхні, Н;

z - циліндрична координата, вісь якої паралельна вектору швидкості, м.

Враховуючи, що в якості направляючої поверхні був обраний циліндр ($r = const$), система диференціальних рівнянь (7.1.3.3) прийме вигляд

$$\begin{aligned}
 -Pr\left(\frac{d\varphi}{ds}\right)^2 + N &= 0, \\
 \frac{d}{ds}\left(Pr\frac{d\varphi}{ds}\right) &= 0, \\
 \frac{d}{ds}\left(P\frac{dz}{ds}\right) - \sqrt{k_1^2 + k_2^2}N &= 0, \\
 r^2\left(\frac{d\varphi}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2 &= 1.
 \end{aligned}
 \tag{7.1.3.4}$$

З другого рівняння системи (7.1.3.4) отримаємо

$$r\frac{d}{ds}\left(P\frac{d\varphi}{ds}\right) = 0.$$

Або

$$\frac{d}{ds}\left(P\frac{d\varphi}{ds}\right) = 0.$$

В остаточному вигляді

$$P\frac{d\varphi}{ds} = C_1 = \text{const.} \tag{7.1.3.5}$$

З рівняння (7.1.3.5) визначимо значення похідної

$$\frac{d\varphi}{ds} = \frac{C_1}{P}. \tag{7.1.3.6}$$

З першого рівняння системи (7.1.3.4) визначаємо значення нормальної реакції

$$N = Pr\left(\frac{d\varphi}{ds}\right)^2. \tag{7.1.3.7}$$

Підставляємо вираз (7.1.3.7) у третє диференціальне рівняння системи (7.1.3.4)

$$\frac{d}{ds}\left(P\frac{dz}{ds}\right) - \sqrt{k_1^2 + k_2^2}Pr\left(\frac{d\varphi}{ds}\right)^2 = 0, \tag{7.1.3.8}$$

Перетворимо рівняння (7.1.3.8) у вигляд

$$\frac{dP}{ds}\frac{dz}{ds} + P\frac{d^2z}{ds^2} - \sqrt{k_1^2 + k_2^2}Pr\left(\frac{d\varphi}{ds}\right)^2 = 0. \tag{7.1.3.9}$$

З четвертого диференціального рівняння системи (7.1.3.4) визначимо

$$\left(\frac{dz}{ds}\right)^2 = 1 - r^2\left(\frac{d\varphi}{ds}\right)^2. \tag{7.1.3.10}$$

З виразу (7.1.3.10) визначаємо значення першої похідної циліндричної координати по дуговій координаті

$$\frac{dz}{ds} = \sqrt{1 - r^2 \left(\frac{d\varphi}{ds}\right)^2}. \quad (7.1.3.11)$$

Диференціюємо рівняння (7.1.3.11) по дуговій координаті s і визначимо значення другої похідної

$$\frac{d}{ds} \left(\frac{dz}{ds} \right) = \frac{d^2 z}{ds^2} = - \frac{r^2 \left(\frac{d^2 \varphi}{ds^2} \frac{d\varphi}{ds} \right)}{\sqrt{1 - r^2 \left(\frac{d\varphi}{ds} \right)^2}}. \quad (7.1.3.12)$$

Для визначення значення другої похідної циліндричної координати φ по дуговій координаті s диференціюємо вираз (7.1.3.6) і отримаємо

$$\frac{d^2 \varphi}{ds^2} = - \frac{C_1 \frac{dP}{ds}}{P^2}. \quad (7.1.3.13)$$

Підставляємо вираз (7.1.3.13) в диференціальне рівняння (7.1.3.12), отримаємо

$$\frac{d^2 z}{ds^2} = - \frac{r^2 \left(- \frac{C_1 \frac{dP}{ds} \frac{d\varphi}{ds}}{P^2} \right)}{\sqrt{1 - r^2 \left(\frac{d\varphi}{ds} \right)^2}}. \quad (7.1.3.14)$$

В остаточному вигляді матимемо

$$\frac{d^2 z}{ds^2} = \frac{r^2 C_1^2}{P^3 \sqrt{1 - r^2 \left(\frac{d\varphi}{ds} \right)^2}} \frac{dP}{ds}. \quad (7.1.3.15)$$

Підставляємо вираз (7.1.3.15) в диференціальне рівняння (7.1.3.9) і з урахуванням (7.1.3.11) отримаємо

$$\frac{dP}{ds} \sqrt{1 - r^2 \left(\frac{d\varphi}{ds} \right)^2} + \frac{r^2 C_1^2}{P^2 \sqrt{1 - r^2 \left(\frac{d\varphi}{ds} \right)^2}} \frac{dP}{ds} - \sqrt{k_1^2 + k_2^2} Pr \left(\frac{d\varphi}{ds} \right)^2 = 0. \quad (7.1.3.16)$$

З урахуванням виразу (7.1.3.6) перетворимо диференціальне рівняння (7.1.3.16) до вигляду

$$\frac{dP}{ds} \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2} + \frac{r^2 C_1^2}{P^2 \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2}} \frac{dP}{ds} - \sqrt{k_1^2 + k_2^2} \frac{C_1^2 r}{P} = 0.$$

Перетворимо останній вираз

$$\frac{dP}{ds} \left[\sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2} + \frac{r^2 C_1^2}{P^2 \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2}} \right] - \sqrt{k_1^2 + k_2^2} \frac{C_1^2 r}{P} = 0.$$

З останнього виразу визначаємо значення першої похідної натягу нитки як функції дугової координати

$$\frac{dP}{ds} = \frac{\sqrt{k_1^2 + k_2^2} \frac{C_1^2 r}{P}}{\sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2} + \frac{r^2 C_1^2}{P^2 \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2}}}. \quad (7.1.3.17)$$

Перетворимо диференціальне рівняння (7.1.3.17) і остаточно отримаємо

$$\frac{dP}{ds} = \frac{\sqrt{k_1^2 + k_2^2} C_1^2 r P \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2}}{\{P^2 [1 - r^2 \left(\frac{C_1}{P}\right)^2] + r^2 C_1^2\}}. \quad (7.1.3.18)$$

Отримане диференціальне рівняння першого порядку дозволяє одержати залежність натягу нитки від дугової координати з урахуванням анізотропії тертя при поперечному ковзанні. Його важко проінтегрувати. Тому для його вирішення будемо використовувати чисельні методи, зокрема метод Рунге-Кутта-Мерсона. Програма для реалізації цього методу наведена в Додатку.

Чисельно вирішуючи диференціальне рівняння (7.1.3.18), отримаємо масив даних, який дозволяє визначати значення натягу нитки залежно від дугової координати.

Цілком очевидно, що для зручності користування отриманими даними їх необхідно представити у вигляді деякої функції. Для цього в роботі було здійснено апроксимацію даних з використанням спеціального програмного забезпечення, описаного в третьому розділі. Апроксимація здійснювалася з використанням степеневого полінома. Ця програма

дозволяє здійснювати апроксимацію або із заданою точністю (ступінь полінома визначається автоматично) або користувач сам задає ступінь полінома.

У результаті отримаємо деяку нову функцію

$$P = V(s) = a_0 + a_1s + a_2s^2 + \dots + a_ns^n, \quad (7.1.3.19)$$

де $a_0, a_1, a_2, \dots, a_n$ - коефіцієнти степеневого полінома.

Перейдемо до визначення залежностей $\varphi(s)$ та $z(s)$. З (7.1.3.11), з урахуванням (7.1.3.6) отримаємо

$$\frac{dz}{ds} = \sqrt{1 - r^2 \left(\frac{C_1}{P} \right)^2}.$$

Підставляємо в останній вираз (7.1.3.19)

$$\frac{dz}{ds} = \sqrt{1 - r^2 \left(\frac{C_1}{a_0 + a_1s + a_2s^2 + \dots + a_ns^n} \right)^2}. \quad (7.1.3.20)$$

Залежність циліндричної координати від дугової координати визначимо з урахуванням (7.1.3.6) і (7.1.3.19)

$$\frac{d\varphi}{ds} = \frac{C_1}{P} = \frac{C_1}{a_0 + a_1s + a_2s^2 + \dots + a_ns^n}. \quad (7.1.3.21)$$

Таким чином, чисельне інтегрування системи трьох диференціальних рівнянь першого порядку

$$\frac{dP}{ds} = \frac{\sqrt{k_1^2 + k_2^2} C_1^2 r P \sqrt{1 - r^2 \left(\frac{C_1}{P} \right)^2}}{\{ P^2 [1 - r^2 \left(\frac{C_1}{P} \right)^2] + r^2 C_1^2 \}},$$

$$\frac{dz}{ds} = \sqrt{1 - r^2 \left(\frac{C_1}{a_0 + a_1s + a_2s^2 + \dots + a_ns^n} \right)^2},$$

$$\frac{d\varphi}{ds} = \frac{C_1}{P} = \frac{C_1}{a_0 + a_1s + a_2s^2 + \dots + a_ns^n},$$

дозволяє одержати залежності натягу і форми осі нитки при поперечному ковзанні по направляючій малої кривизни з урахуванням анізотропії тертя.

Взаємодія жорсткої на вигин нитки з циліндричною направляючою великої кривизни

Певний інтерес представляє дослідження процесу взаємодії жорстких на вигин ниток з циліндричними направляючими великої кривизни з урахуванням анізотропії тертя. До жорстких на вигин ниток відносяться моонитки, комплексні нитки і пряжа з високим скручуванням.

При виведенні основних рівнянь необхідно зробити кілька допущень. По-перше, будемо вважати, що нитка на поверхні буде розташовуватися в дотичній площині (див.рис.7.1.4.1а), тобто кут нахилу площини α до горизонтальної осі z визначається кутом нахилу ведучих голок нитки. Обґрунтованість прийняття такого припущення визначається тим, що під дією згинальних моментів в нормальній і направляючій площинах, за рахунок жорсткості на вигин нитка буде прагнути зайняти положення, яке забезпечить їй найменше значення потенційної енергії, необхідної для забезпечення стаціонарного положення рівноваги в процесі переміщення.

Друге припущення полягає в тому, що масою елемента нитки, який взаємодіє з направляючою поверхнею можна знехтувати. Поєднаємо з віссю циліндра циліндричну координатну систему. Вісь z буде збігатися з віссю циліндра. Її позитивний напрямок буде протилежний напрямку вектора швидкості (тут ми будемо використовувати метод зверненого руху).

Тоді, форма осі жорсткої нитки (див.рис.7.1.4.1б) представлятиме собою еліпс. Велика і мала півосі визначаються з виразів

$$a = \frac{r}{\sin \alpha}, \quad b = r, \quad (7.1.4.1)$$

де r - - радіус циліндра, м;

a, b - відповідно велика і мала півосі еліпса.

Використовуючи основні положення диференціальної геометрії можна отримати дві очевидні диференціальні рівності

$$\begin{aligned} r \frac{d\varphi}{ds} &= \sin \alpha, \\ \frac{dz}{ds} &= \cos \alpha, \end{aligned} \tag{7.1.4.2}$$

де r, φ, z - відповідні циліндричні координати;

s - дугова координата, м;

α - кут нахилу дотичної площини до осі z , рад.

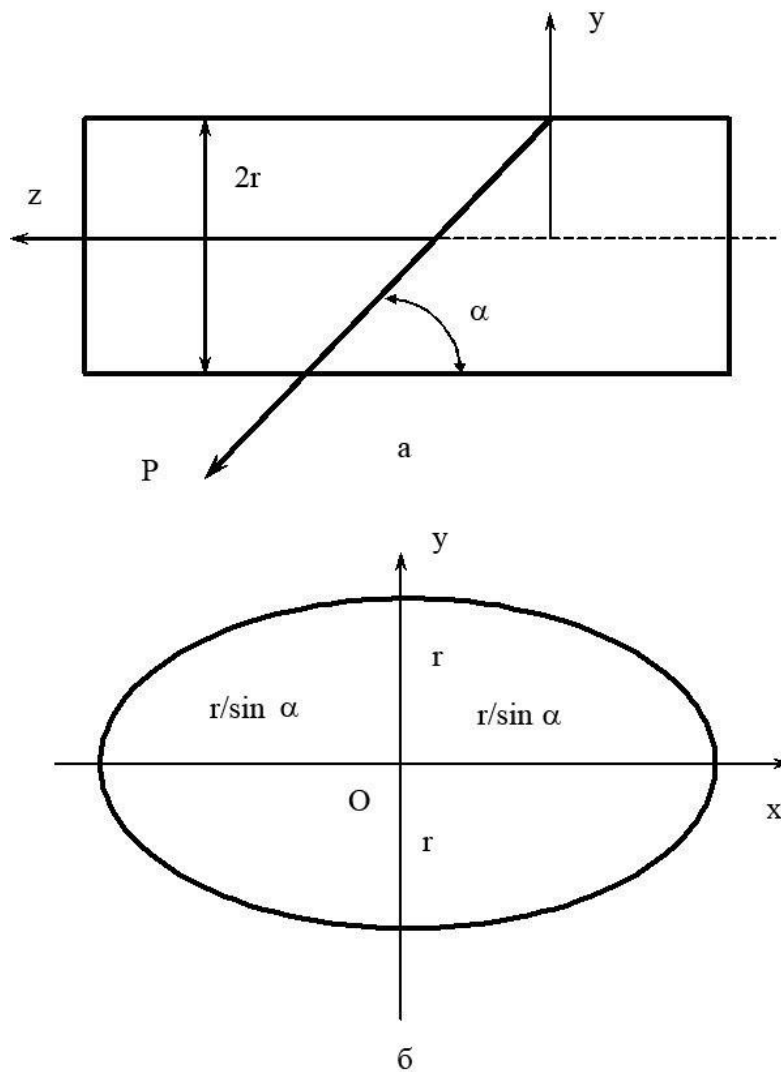


Рис.7.1.4.1. Розрахункова схема для визначення натягу жорсткої на вигин нитки

Сила тертя, розташована в дотичній площині, визначається з наступної системи рівнянь

$$\vec{F} = \vec{F}_1 + \vec{F}_2 \quad , \quad F = \sqrt{F_1^2 + F_2^2}, \quad (7.1.4.3)$$

де $F_1 = k_1 N$ - проекція сили тертя F на дотичну вісь натурального тригранника; $F_2 = k_2 N$ - проекція сили тертя F на бінормальну вісь натурального тригранника;

k_1, k_2 - коефіцієнти тертя нитки в подовжньому і поперечному напрямку, які характеризують анізотропію фрикційних властивостей.

Тоді величину сили тертя, з урахуванням (2. 4. 3), можна визначити за наступною формулою

$$F = N \sqrt{k_1^2 + k_2^2}. \quad (7.1.4.4)$$

Основна система диференціальних рівнянь в циліндричних координатах, що описує поперечне ковзання жорсткої на вигин нитки по направляючій з урахуванням анізотропії фрикційних властивостей, буде мати вигляд

$$\begin{aligned} \frac{d}{ds} \left(P \frac{dr}{ds} \right) - Pr \left(\frac{d\varphi}{ds} \right)^2 + N &= 0, \\ \frac{d}{ds} \left(Pr \frac{d\varphi}{ds} \right) &= 0, \\ \frac{d}{ds} \left(P \frac{dz}{ds} \right) - \sqrt{k_1^2 + k_2^2} N &= 0, \\ \left(\frac{dr}{ds} \right)^2 + r^2 \left(\frac{d\varphi}{ds} \right)^2 + \left(\frac{dz}{ds} \right)^2 &= 1, \end{aligned} \quad (7.1.4.5)$$

де P - натяг нитки, Н;

N - нормальна реакція поверхні, Н.

Враховуючи, що в якості направляючої поверхні був обраний циліндр ($r = const$), система диференціальних рівнянь (2. 4. 5) набуде вигляду

$$\begin{aligned}
 -Pr\left(\frac{d\varphi}{ds}\right)^2 + N &= 0, \\
 \frac{d}{ds}\left(Pr\frac{d\varphi}{ds}\right) &= 0, \\
 \frac{d}{ds}\left(P\frac{dz}{ds}\right) - \sqrt{k_1^2 + k_2^2}N &= 0, \\
 r^2\left(\frac{d\varphi}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2 &= 1.
 \end{aligned}
 \tag{7.1.4.6}$$

Підставимо в останню систему диференціальних рівнянь (7.1.4.6) вирази (7.1.4.2), отримаємо

$$\begin{aligned}
 -Pr\left(\frac{\sin\alpha}{r}\right)^2 + N &= 0, \\
 \frac{d}{ds}(P\sin\alpha) &= 0, \\
 \frac{d}{ds}(P\cos\alpha) - \sqrt{k_1^2 + k_2^2}N &= 0.
 \end{aligned}
 \tag{7.1.4.7}$$

Перетворимо систему диференціальних рівнянь (7.1.4.7) до виду

$$\begin{aligned}
 N &= Pr\left(\frac{\sin\alpha}{r}\right)^2, \\
 \frac{d}{ds}(P\sin\alpha) &= 0, \\
 \frac{d}{ds}(P\cos\alpha) &= \sqrt{k_1^2 + k_2^2}N.
 \end{aligned}
 \tag{7.1.4.8}$$

Підставляємо з першого рівняння системи диференціальних рівнянь (7.1.4.8) значення нормальної реакції поверхні в третє рівняння, отримаємо

$$\cos\alpha\frac{dP}{ds} = P\frac{\sin^2\alpha}{r}\sqrt{k_1^2 + k_2^2}.$$

Перетворимо останнє диференціальне рівняння до вигляду

$$\frac{dP}{P} = \frac{\sin^2\alpha}{r\cos\alpha}\sqrt{k_1^2 + k_2^2}ds.$$

Останнє диференціальне рівняння є диференціальним рівнянням із перемінними. Проігноруємо його

$$\frac{dP}{P} = \frac{\sin^2\alpha}{r\cos\alpha}\sqrt{k_1^2 + k_2^2}ds.$$

$$\int_{P_0}^{P_1} \frac{dP}{P} = \frac{\sin^2\alpha}{r\cos\alpha}\sqrt{k_1^2 + k_2^2} \int_0^s ds.$$

Або

$$\ln \frac{P_1}{P_0} = \frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{k_1^2 + k_2^2} s,$$

де P_1 - натяг нитки в точці сходу з циліндра, Н;

P_0 - натяг нитки в крайній правій точці на траєкторії (див.рис.7.1.4.1), Н;

s - довжина дуги нитки від точки сходу з направляючої до крайньої правої точки, м.

В остаточному вигляді отримаємо

$$P_1 = P_0 e^{\frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{k_1^2 + k_2^2} s}. \quad (7.1.4.9)$$

Для визначення довжини дуги нитки s скористаємося наступною системою диференціальних рівнянь

$$\begin{aligned} ds &= \sqrt{1 + \left(\frac{dx}{dy}\right)^2} dy, \\ \frac{x^2}{a^2} + \frac{y^2}{b^2} &= 1, \\ a &= \frac{r}{\sin \alpha}, \quad b = r. \end{aligned} \quad (7.1.4.10)$$

З другого рівняння системи (7.1.4.10) отримаємо

$$x = a \sqrt{1 - \frac{y^2}{b^2}}. \quad (7.1.4.11)$$

Диференціюємо рівняння (7.1.4.11) по y , отримаємо

$$\frac{dx}{dy} = \frac{ay}{b^2 \sqrt{1 - \frac{y^2}{b^2}}}. \quad (7.1.4.12)$$

Підставляємо (7.1.4.12) в перше рівняння системи (7.1.4.10), отримаємо

$$ds = \sqrt{1 + \frac{a^2 y^2}{b^4 \left(1 - \frac{y^2}{b^2}\right)}} dy. \quad (7.1.4.13)$$

В остаточному вигляді отримаємо

$$ds = \sqrt{\frac{b^4 + y^2(a^2 - b^2)}{b^4 - b^2 y^2}} dy. \quad (7.1.4.14)$$

Проінтегрувати рівняння (7.1.4.14) в елементарних функціях не представляється можливим. Для цієї мети необхідно використовувати відповідні чисельні методи.

У цій роботі ми будемо використовувати метод Сімпсона, суть якого полягає в інтерполяції функції $f(x)$ в n проміжках поліномом Лагранжа. Блок-схема реалізації алгоритму методу Сімпсона наведена на рис.7.1.4.2. Після розбиття відрізка $[a, b]$ на m рівних частин формула Сімпсона буде мати наступний вигляд

$$I = \int_a^b f(x)dx \cong \frac{H}{3}[f(a) + 4f(a+H) + 2f(a+2H) + \dots + 4f(b-H) + f(b)] - \frac{mH^5}{90} f^{IV}(\xi), \quad (7.1.4.15)$$

де $f^{IV}(\xi)$ - значення четвертої похідної функції $f(x)$ в точці $x = \xi$, де вона максимальна.

Для реалізації методу Сімпсона в роботі була розроблена спеціальна прикладна програма на мові Object Pascal в середовищі Delphi (дивися Додаток). Ця програма включає в себе три форми ІМТ 1, ІМТ 2, UErrors. Перші дві форми наведені на рис.7.1.4.3. Для введення підінтегральної функції $f(x)$ на другій формі розміщений стандартний елемент Memo1: Tmemo.

Для коректного введення підінтегральної функції у вікно Memo1: Tmemo в програмі був реалізований спеціально розроблений транслятор Synt.

Основні компоненти форми ІМТ 2 мають вигляд

$$\begin{aligned} TfrmIMT2 &= \text{class}(TForm) \\ \text{lbl1IMT2: TLabel; } &\text{lbl2IMT2: TLabel; } & (7.1.4.16) \\ \text{lbl3IMT2: TLabel; } &\text{edt1IMT2: TEdit; } \end{aligned}$$

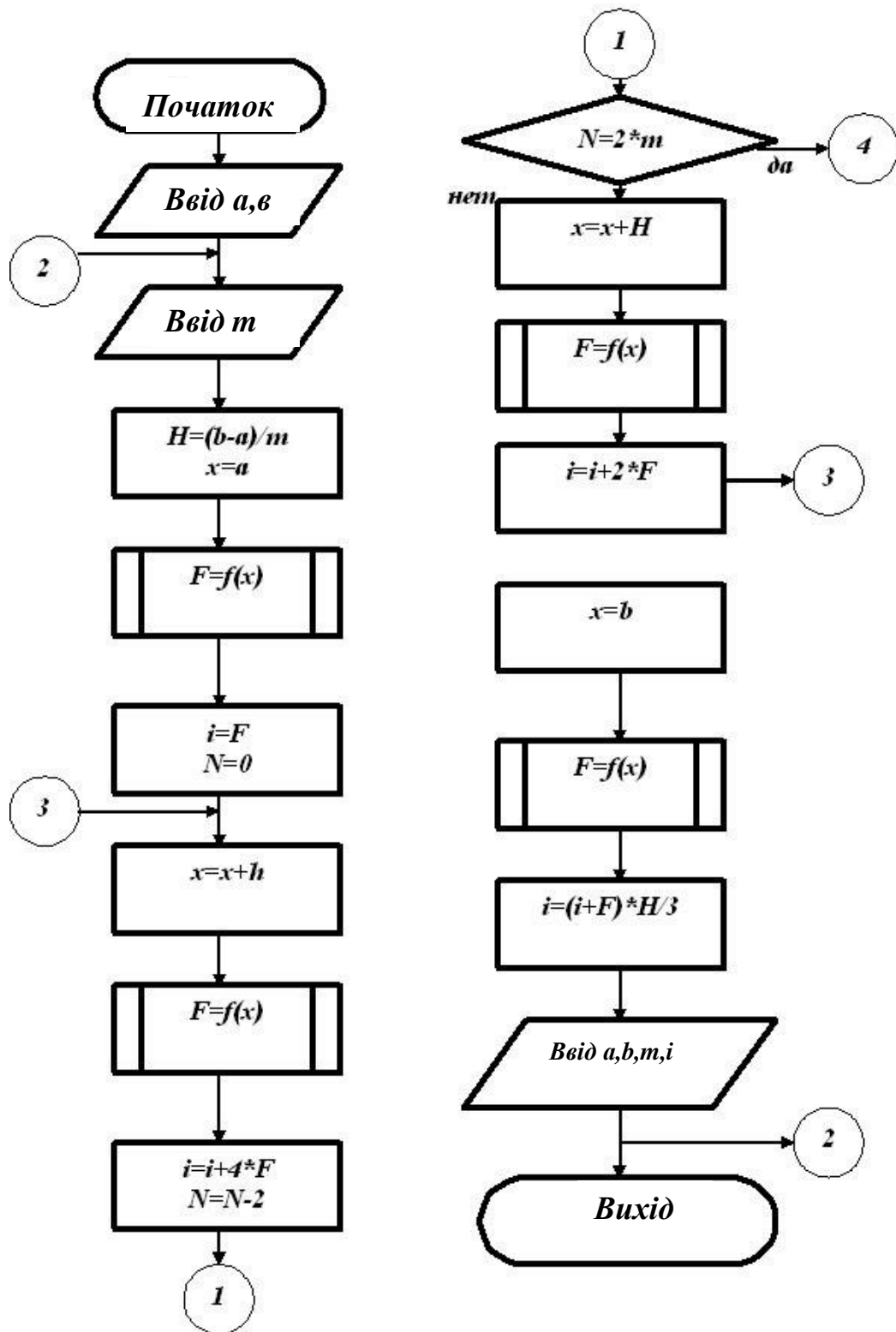


Рис.7.1.4.2. Блок-схема алгоритму методу Сімпсона

Форма `Uerrors` призначена для повідомлень про помилки при введенні підінтегральною функції. Основні компоненти форми `IMT1` мають вигляд

```
lbl1IMT1: TLabel;  
lbl2IMT1: TLabel;  
  lbl3IMT1: TLabel;  
lbl4IMT1: TLabel;  
  lbl5IMT1: TLabel;  
btn1IMT1: TButton; (7.1.4.17)  
img1IMT1: TImage;  
  Label1: TLabel;  
  Label2: TLabel;  
  Label3: TLabel
```



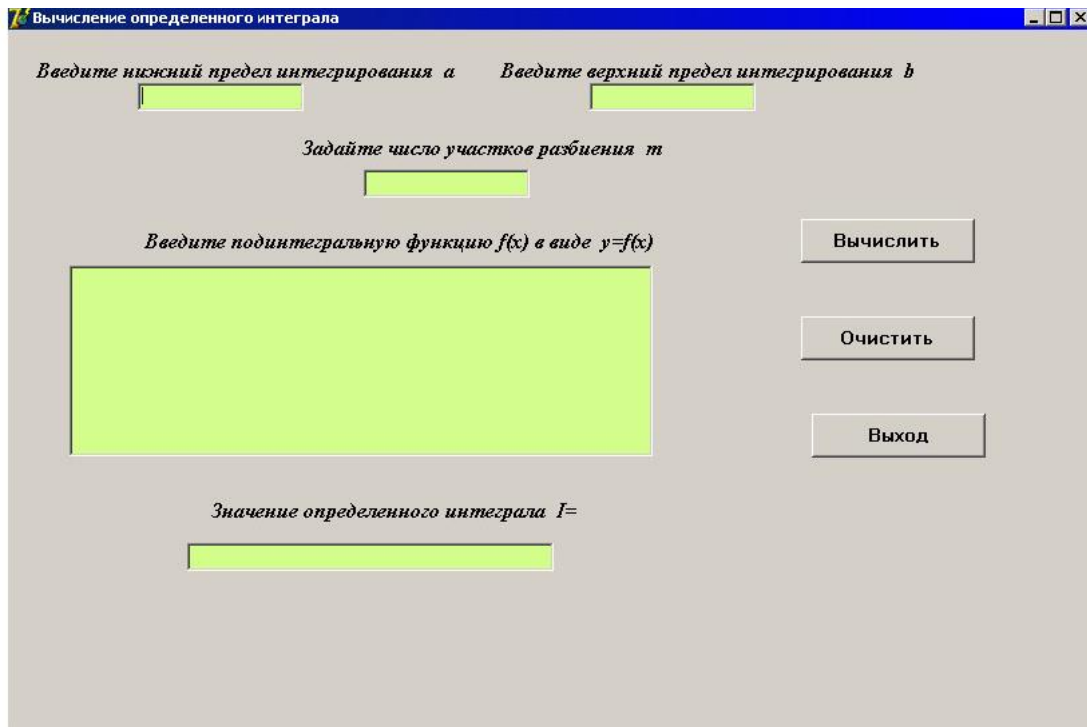


Рис.7.1.4.3. Основні форми програми для обчислення певного інтегралу

Використовуючи розроблену програму обчислюємо значення інтеграла (7.1.4.14)

$$\int_0^s ds = \int_0^y \sqrt{\frac{b^4 + y^2(a^2 - b^2)}{b^4 - b^2 y^2}} dy.$$

Отриманий масив даних апроксимували з використанням спеціальної програми (опис алгоритму і самої програми наведено в розділі 3). В результаті отримуємо степеневий поліном виду

$$s = b_0 + b_1 y + b_2 y^2 + \dots + b_n^n, \quad (7.1.4.18)$$

де $b_0, b_1, b_2 \dots b_n$ - коефіцієнти апроксимаційного полінома.

Тоді вираз (7.1.4.9) для визначення натягу нитки жорсткої на вигин з урахуванням анізотропії тертя прийме вигляд

$$P_1 = P_0 e^{\frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{k_1^2 + k_2^2} (b_0 + b_1 y + b_2 y^2 + \dots + b_n^n)}. \quad (7.1.4.19)$$

7.2. ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ВПЛИВУ МАТЕРІАЛУ НИТКИ І АНІЗОТРОПІЇ ТЕРТЯ НА ЇЇ НАТЯГ І ФОРМУ ОСІ

Для теоретичного дослідження впливу матеріалу нитки і анізотропії тертя на натяг і форму її осі були обрані віскозна комплексна нитка 16,7 Текс, капронова комплексна нитка 31,2 Текс, бавовняна гребінна кручена пряжа 23,6 Текс, вовняна гребінна кручена пряжа 36 Текс.

Вибір даного асортименту, в першу чергу, пояснюється приблизно однаковим тексом ниток. Крім того, цей вид сировини широко використовується для виготовлення текстильних і трикотажних виробів. За результатами експериментальних досліджень (див. розділ 4) коефіцієнти тертя в поздовжньому і поперечному напрямках для цього виду сировини відповідно дорівнювали: віскозна комплексна нитка $k_1 = 0,16$, $k_2 = 0,21$; капронова комплексна нитка $k_1 = 0,18$, $k_2 = 0,24$; бавовняна гребінна кручена пряжа $k_1 = 0,17$, $k_2 = 0,22$; вовняна гребінна кручена пряжа $k_1 = 0,14$, $k_2 = 0,18$ (при виборі значень визначальним була величина в нульовій точці плану експерименту при можливому варіюванні до 10%).

Будова поверхонь робочих органів трикотажних і текстильних машин, а також самих ниток свідчать про те, що величина сили тертя в ортогональних напрямках (наприклад дотичній і нормалі до точки на осі нитки) відрізнятимуться. Це дозволяє говорити про необхідність врахування анізотропії фрикційних властивостей. У загальному випадку величина сили тертя визначається з виразу [11]

$$F = N \sqrt{k_1^2 + k_2^2},$$

де F - сила тертя;

N - сила нормального тиску;

$\sqrt{k_1^2 + k_2^2}$ - ефективний коефіцієнт тертя [10-12].

Для розкриття його суті представимо силу тертя як функцію швидкості

$$\vec{F} = -\frac{NA\vec{v}}{v},$$

де A - матриця, яка задає лінійне переміщення;

v - швидкість точки на осі нитки.

Для двох ортогональних напрямів (дотична і нормаль) вираз для даної матриці матиме вигляд

$$A = \begin{pmatrix} \mu_1 \cos \varphi & 0 \\ 0 & \mu_2 \sin \varphi \end{pmatrix},$$

де μ_1, μ_2 - коефіцієнти тертя нитки в двох ортогональних напрямках;

φ - кут, який утворює вектор швидкості з дотичної до осі нитки.

Тоді ефективний коефіцієнт тертя визначимо по формулі

$$\sqrt{k_1^2 + k_2^2} = \sqrt{\mu_1^2 \cos^2 \varphi + \mu_2^2 \sin^2 \varphi}.$$

Програмне забезпечення е для про визначення форми осі та натягу нитки при поперечному ковзанні з урахуванням фізико-механічних характеристик у випадку, коли відстань між опорами значно перевищує діаметр нитки

У розділі 7.1 було отримано диференціальне рівняння (7.1.15), яке представляє диференціальне рівняння другого порядку і визначає собою форму осі нитки при поперечному ковзанні з урахуванням анізотропії тертя. Виконуючи стандартну процедуру зниження ступеня диференціального рівняння прийдемо до системи диференціальних рівнянь першого порядку. Для вирішення цієї системи диференціальних рівнянь будемо використовувати метод Рунге-Кутта-Мерсона з автоматичною

зміною кроку інтегрування. Блок-схема алгоритму представлена на рис.7.2.1.

Даний метод забезпечує наближену оцінку похибки на кожному кроці інтегрування. Похибка інтегрування диференціальних рівнянь визначається порядком h^5 [5,6]. При високої точності і рішенні у вигляді кривих з різко відрізненою крутизною автоматична зміна кроку забезпечує зменшення загального числа кроків в кілька разів, різко зменшує ймовірність числової нестійкості, дає більш рівномірне розташування точок графіка кривих.

Для реалізації методу Рунге-Кутта-Мерсона з автоматичною зміною кроку інтегрування реалізується наступний алгоритм [5]. На початковому етапі необхідно задати число диференціальних рівнянь першого порядку N , похибка обчислень $\varepsilon = E$, початковий крок інтегрування диференціальних рівнянь $h = H$ і початкове значення аргументу x_0 , Початкове значення функцій системи диференціальних рівнянь $y_{10}, y_{20}, \dots, y_{N0}$. Обчислюємо коефіцієнти $K_{0j}, K_{1j}, K_{2j}, K_{3j}, K_{4j}$ шляхом організації п'яти циклів з керуючою змінною $j = 1, 2, \dots, N$. Система рівнянь буде мати наступний вигляд

$$\begin{aligned}
 K_{0j} &= hF_j(x_i; Y_{ji}), \\
 K_{1j} &= hF_j(x_i + \frac{1}{3}h; Y_{ji} + \frac{1}{3}K_{0j}), \\
 K_{2j} &= hF_j(x_i + \frac{1}{3}h; Y_{ji} + \frac{1}{6}K_{0j} + \frac{1}{6}K_{1j}), \\
 K_{3j} &= hF_j(x_i + \frac{1}{2}h; Y_{ji} + \frac{1}{8}K_{0j} + \frac{3}{8}K_{2j}), \\
 K_{4j} &= hF_j(x_i + h; Y_{ji} + \frac{1}{2}K_{0j} - \frac{3}{2}K_{2j} + 2K_{3j})
 \end{aligned}
 \tag{7.2.1}$$

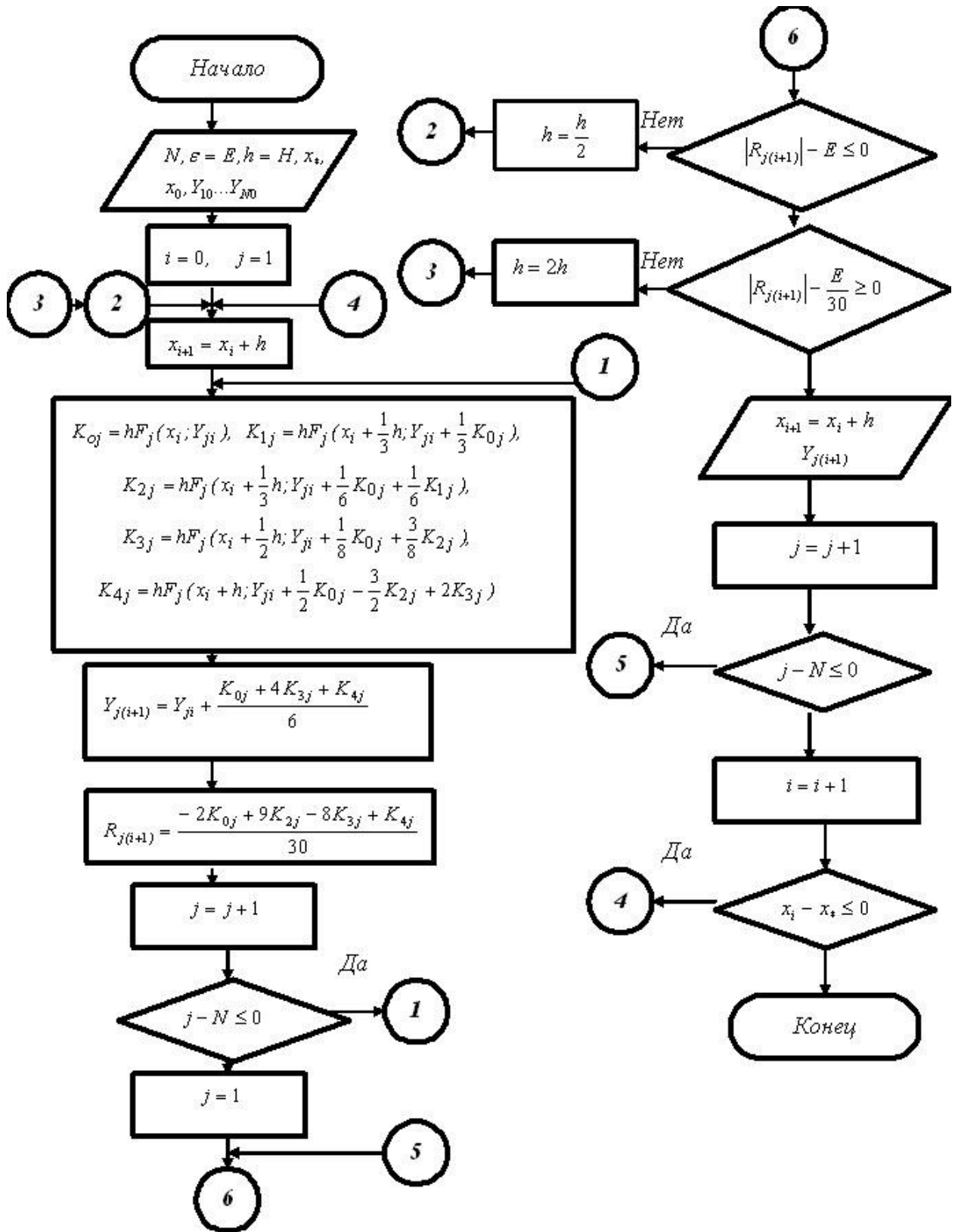


Рис.7.2.1.Блок-схема алгоритму Рунге-Кутта-Мерсона з автоматичною зміною кроку інтегрування

В останньому циклі знаходимо значення

$$Y_{j(i+1)} = Y_{ji} + \frac{K_{0j} + 4K_{3j} + K_{4j}}{6}. \quad (7.2.2)$$

Похибку обчислень визначаємо за формулою

$$R_{j(i+1)} = \frac{-2K_{0j} + 9K_{2j} - 8K_{3j} + K_{4j}}{30}. \quad (7.2.3)$$

На наступному етапі необхідно здійснити перевірку умов

$$|R_{j(i+1)}| - E \leq 0, \quad (7.2.4)$$

і

$$|R_{j(i+1)}| - \frac{E}{30} \geq 0. \quad (7.2.5)$$

Якщо умова (7.2.4) не виконується, то необхідно крок інтегрування диференціальних рівнянь h розділити на 2 і повторити обчислення коефіцієнтів $K_{0j}, K_{1j}, K_{2j}, K_{3j}, K_{4j}$ за формулами (7.2.1) відновивши початкові значення $y_{10}, y_{20} \dots y_{N0}$. Якщо умова (7.2.4) виконується і виконується умова (7.2.5), то значення $x_{i+1} = x_i + h$ і $y_{j(i+1)}$ виводяться на друк.

Якщо умова (7.2.5) не виконується, то крок інтегрування h збільшують удвічі і повторюють обчислення коефіцієнтів $K_{0j}, K_{1j}, K_{2j}, K_{3j}, K_{4j}$ за формулами (7.2.1).

Для реалізації даного алгоритму була написана програма на мові Object Pascal в середовищі Delphi. Програма наведена в додатку А.

На рис.7.2.2 представлена перша форма TfrmSDU1 = class (TForm) програми для інтегрування системи диференціальних рівнянь з використанням алгоритму Рунге-Кутта-Мерсона з автоматичною зміною кроку інтегрування. На даній формі розташовані 11 міток: lbl1SDU1: TLabel; lbl2SDU1: TLabel; lbl4SDU1: TLabel; lbl 5 SDU 1: TLabel; lbl 6 SDU 1: TLabel; lbl 7 SDU 1: TLabel; Label1: TLabel; Label 2: TLabel; Label 3: TLabel; Label 4: TLabel; Label5: TLabel. Кнопка btn1SDU1: Tbutton

призначена для переходу на другу, основну форму програми, яка представлена на рис.7.2.3. Також на цій формі розташований компонент Image1: Timage для розміщення малюнку.



Рис.7.2.2. Перша форма TfrmSDU1 = class (TForm) програми

На другій основній формі TfrmSDU2 = class (TForm) вісім міток lbl1SDU2: TLabel; lbl 2 SDU 2: TLabel; lbl 3 SDU 2: TLabel; lbl 4 SDU 2: TLabel; lbl 5 SDU 2: TLabel; lbl 6 SDU 2: TLabel; lbl 7 SDU 2: TLabel; lbl8SDU2: TLabel і п'ять однорядкових вікон редагування edt1SDU2: TEdit; edt 2 SDU 2: TEdit; edt 3 SDU 2: TEdit; edt 4 SDU 2: TEdit; edt5SDU2: Tedit.

Вікно edt1SDU2: Tedit призначене для завдання числа диференціальних рівнянь першого порядку. Вікно edt2SDU2: Tedit призначено для введення початкового значення аргументу x_0 . Вікно edt3SDU2: Tedit призначене для введення кінцевого значення аргументу E . Вікно edt 4 SDU 2: Tedit призначене для завдання похибки

обчислення. Вікно edt5SDU2: Tedit призначене для введення початкового значення кроку інтегрування.

Багаторядкове вікно редагування Memo1: Tmemo призначене для введення, з використанням транслятора, диференціальних рівнянь першого порядку.

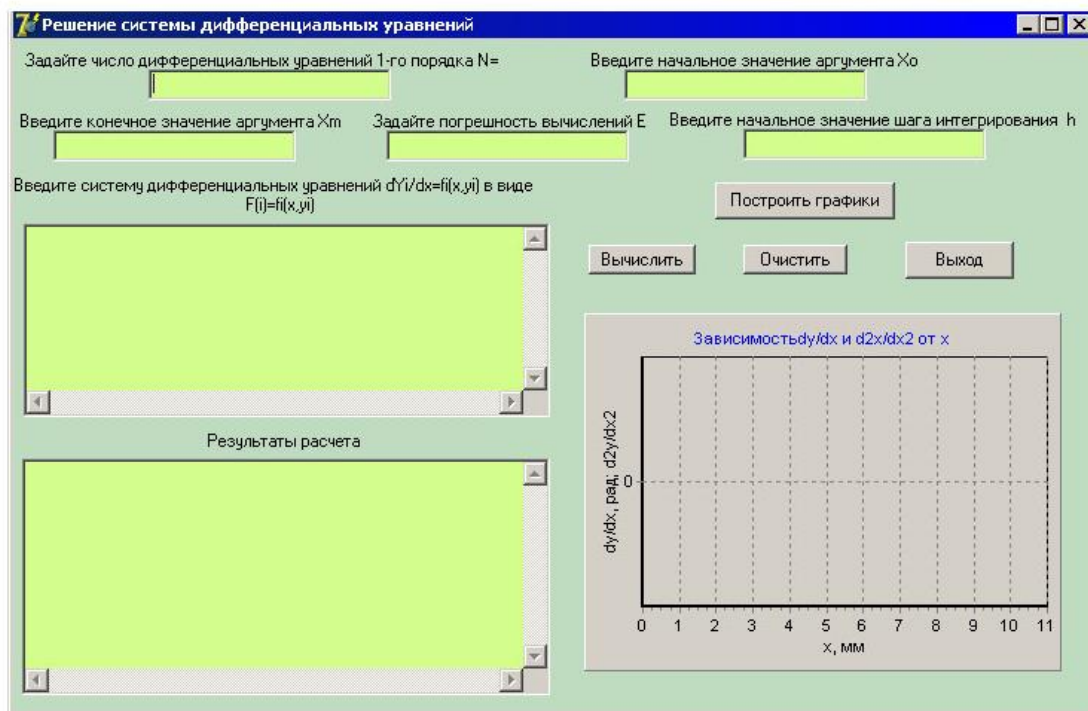


Рис.7.2.3. Основна форма TfrmSDU2 = class (TForm) програми

Багаторядкове вікно редагування mem1SDU2: Tmemo призначене для виведення результатів розрахунку. На рис.7.2.4 представлена форма TfrmSDU2 = class (TForm) з введеними даними.

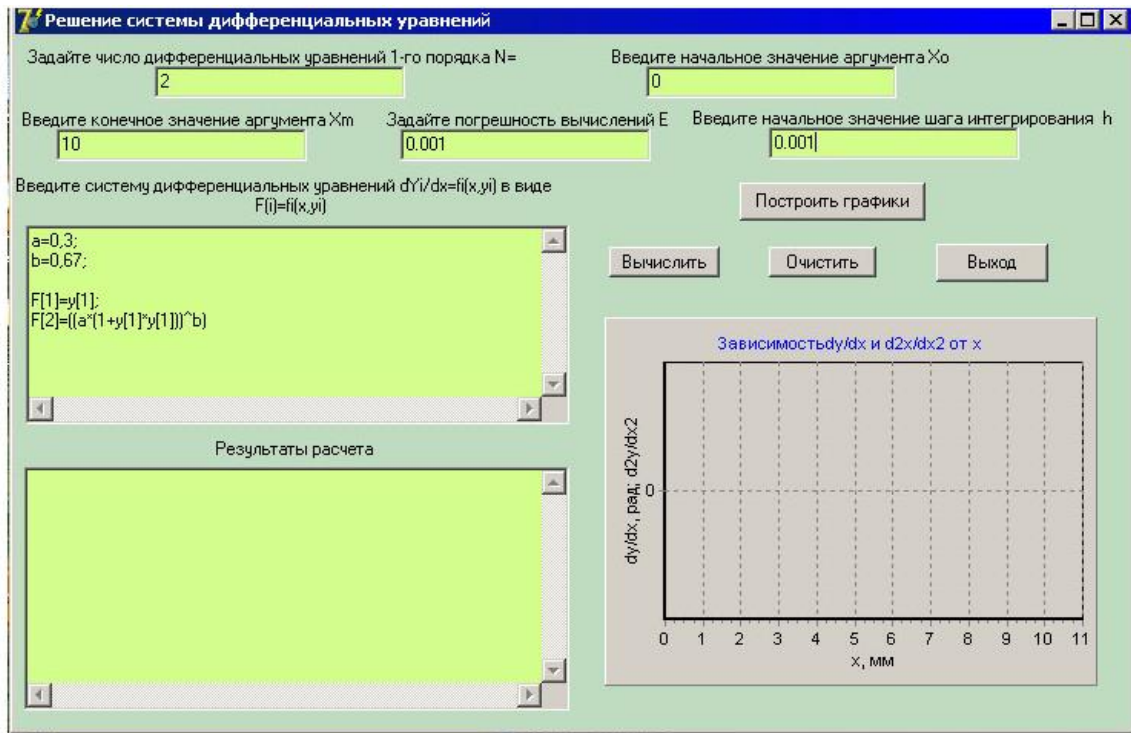


Рис.7.2.4. Ввід даних

Компонент Chart1: Tchart призначений для виведення графічних залежностей обчислюваних величин.

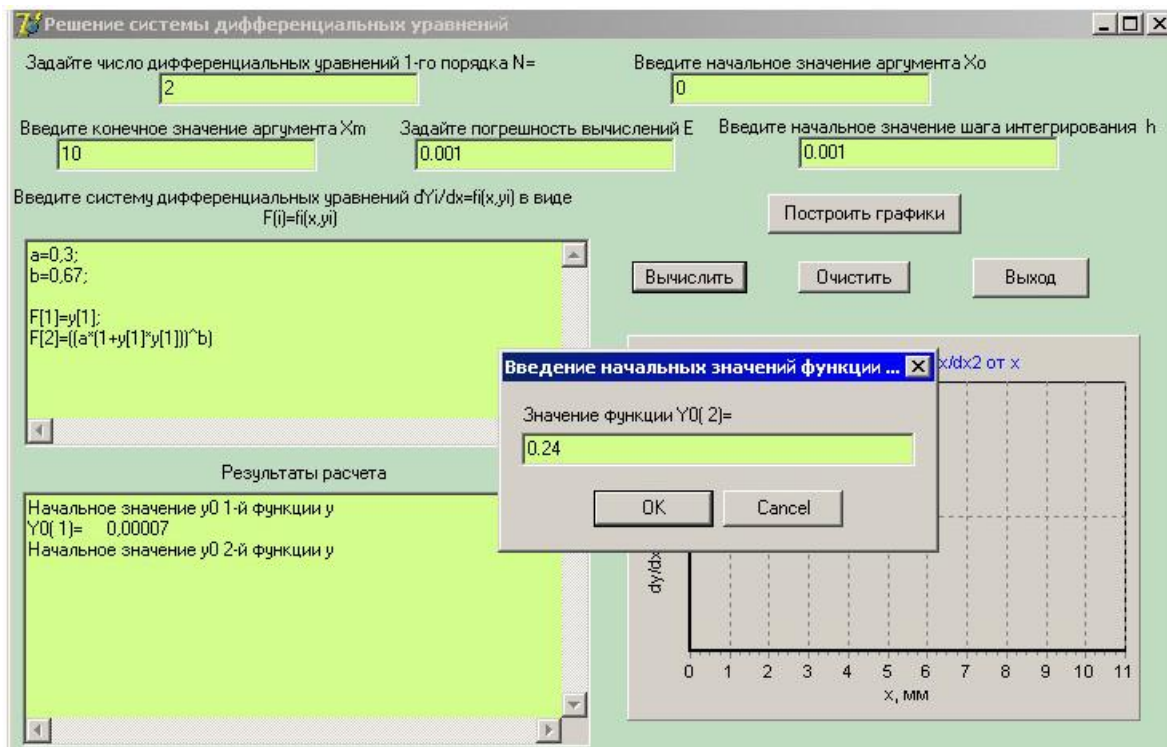


Рис.7.2.5. Задання початкових значень для інтегрування систем

На цій формі розміщені чотири кнопки btn1SDU2: Tbutton; btn2SDU2: TButton; btn3SDU2: Tbutton; Button1: Tbutton. Кнопка btn1SDU2: Tbutton призначена для початку обчислень. Кнопка btn2SDU2: Tbutton призначена для очищення всіх вікон від даних. Кнопка btn3SDU2: Tbutton призначена для виходу з програми. Кнопка Button1: Tbutton призначена для побудови графіків обчислюваних величин.

Для введення початкових значень аргументу і функцій реалізується цикл

```
for i:=1 to N do

begin

ii:= format('%2.0d',[i]);

mem1SDU2.Lines.Add ('Початкове значення y0'+ii+'-й

функції y');

Y0[i]:=InputBox ('Введення початкових значень функції

Y0(i)', 'Значення функціїY0('+ii+')=','');

Val(Y0[i],w[i],code);

y[i]:=w[i];

xx[i]:=format('%12.5f',[w[i]]);

mem1SDU2.Lines.Add('Y0('+ii+')='+xx[i]);

end.
```

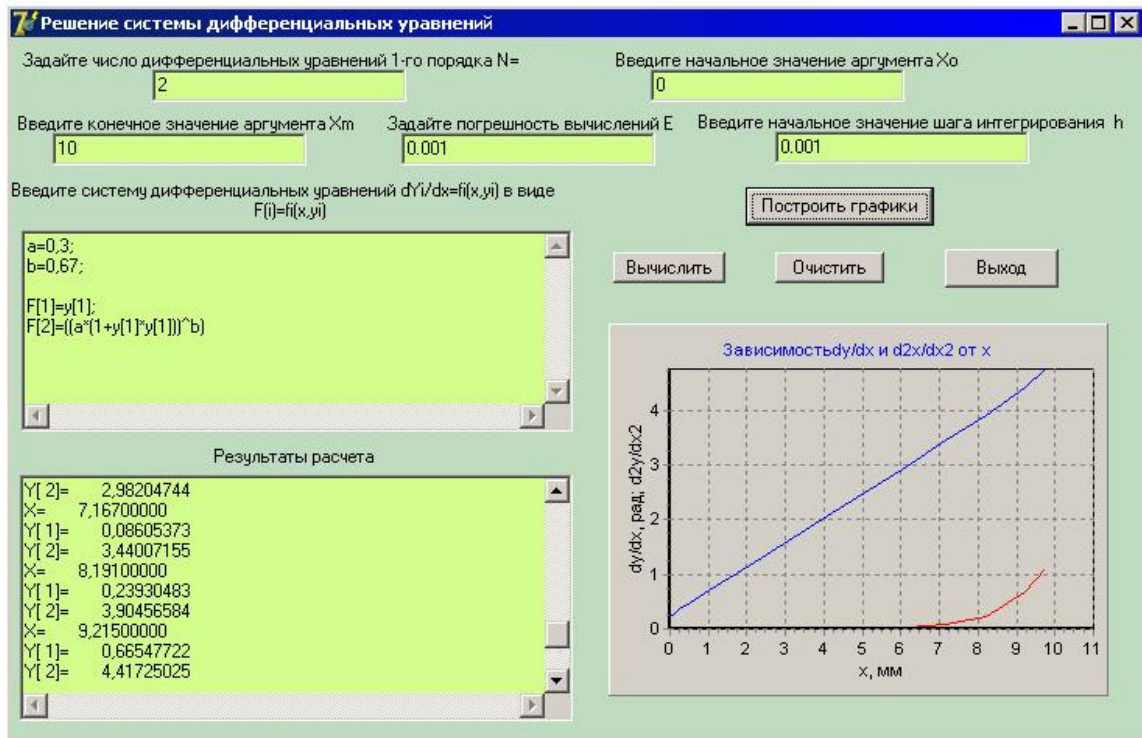



Рис.7.2.6. Висновок результатів інтегрування системи диференціальних рівнянь

Для проведення обчислень використовувалися дані: для віскозної комплексної нитки 16,7 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,16$, $k_2 = 0,21$); для капронової комплексної нитки 31,2 Текс (коефіцієнти тертя дорівнювали 0,18, 0,24); для бавовняної пряжі (гребінного прядіння, кручена) 23,6 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,17$, $k_2 = 0,22$); для вовняної пряжі (гребінного прядіння, кручена) 36 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,14$, $k_2 = 0,18$).

При виборі сировини керувалися тим, що ці нитки і пряжа широко використовуються при виготовленні полотен і окремих виробів на трикотажних машинах і в ткацтві. Крім того, ці чотири групи мають різні фрикційні характеристики.

У таблицях 7.2.1-7.2.4 представлені результати визначення першої та другої похідної $y_1 = \frac{dy}{dx}$, $y_2 = \frac{dy_1}{dx} = \frac{d^2y}{dx^2}$ від абсциси x в результаті

чисельного інтегрування диференціального рівняння (2.1.15) другого порядку.

Таблиця 7.2.1

Визначення першої та другої похідної $y_1 = \frac{dy}{dx}$, $y_2 = \frac{dy_1}{dx} = \frac{d^2y}{dx^2}$ від абсциси x для віскозної комплексної нитки

Віскозна комплексна нитка 16,7 Текс					
Ізотропна			Анізотропна		
x	y_1	y_2	x	y_1	y_2
0	0	0,21	0	0	0,21
0,001	0,00004004	0,2104	0,001	0,0000600	0,2103
0,003	0,0000401	0,211	0,003	0,0000601	0,211
0,007	0,0000402	0,213	0,007	0,0000604	0,212
0,01	0,000040	0,21	0,01	0,000060	0,21
0,03	0,000041	0,22	0,03	0,000061	0,22
0,06	0,000042	0,23	0,06	0,000063	0,23
0,12	0,00004	0,26	0,12	0,00006	0,25
0,25	0,00005	0,32	0,25	0,00007	0,30
0,51	0,00006	0,44	0,51	0,00010	0,40
1,02	0,0001	0,67	1,02	0,00016	0,59
2,04	0,0003	1,14	2,04	0,0004	0,98
4,09	0,002	2,08	4,09	0,003	1,76
6,14	0,017	3,02	6,14	0,026	2,54
8,19	0,133	3,96	7,16	0,073	2,93
9,21	0,371	4,44	8,19	0,205	3,32
10,23	1,033	5,01	10,23	1,588	4,35

Таблиця 7.2.2

Визначення першої та другої похідної $y_1 = \frac{dy}{dx}$, $y_2 = \frac{dy_1}{dx} = \frac{d^2y}{dx^2}$ від абсиси x для капронової комплексної нитки

Капронова комплексна нитка 31,2 Текс					
Ізотропна			Анізотропна		
x	y_1	y_2	x	y_1	y_2
0	0	0,24	0	0	0,24
0,001	0,00006006	0,2404	0,001	0,00008008	0,240
0,003	0,000060	0,241	0,003	0,0000802	0,241
0,007	0,000060	0,243	0,007	0,0000805	0,242
0,01	0,000060	0,24	0,015	0,000081	0,24
0,03	0,000061	0,25	0,03	0,000082	0,25
0,06	0,000063	0,27	0,06	0,000085	0,26
0,12	0,000068	0,30	0,12	0,000090	0,29
0,25	0,000077	0,36	0,25	0,000103	0,34
0,51	0,000100	0,49	0,51	0,0001	0,44
1,02	0,0001	0,74	1,02	0,0002	0,65
2,04	0,0004	1,24	2,04	0,0006	1,07
4,09	0,0035	2,24	4,09	0,004	1,90
6,14	0,0265	3,24	6,14	0,035	2,74
7,16	0,0737	3,75	7,16	0,098	3,15
8,19	0,205	4,25	8,19	0,273	3,58
9,21	0,570	4,79	9,21	0,760	4,06
10,23	1,586	5,50	10,23	2,117	4,86

Таблиця 7.2.3

Визначення першої та другої похідної $y_1 = \frac{dy}{dx}$, $y_2 = \frac{dy_1}{dx} = \frac{d^2y}{dx^2}$ від абсциси x для бавовняної крученої пряжі

Бавовняна кручена пряжа (гребінного прядіння) 23,6 Текс					
Ізотропна			Анізотропна		
x	y_1	y_2	x	y_1	y_2
0	0	0,22	0	0	0,22
0,001	0,00005	0,220	0,001	0,00007007	0,2204
0,003	0,0000501	0,221	0,003	0,0000702	0,221
0,007	0,0000503	0,223	0,007	0,0000704	0,22
0,01	0,0000507	0,22	0,015	0,000071	0,22
0,03	0,000051	0,23	0,031	0,000072	0,23
0,06	0,000053	0,24	0,063	0,000074	0,24
0,12	0,00005	0,27	0,127	0,00007	0,27
0,25	0,00006	0,33	0,255	0,00009	0,32
0,511	0,00008	0,45	0,511	0,0001	0,42
1,02	0,0001	0,69	1,023	0,0002	0,63
2,04	0,0003	1,18	2,047	0,0005	1,04
4,09	0,002	2,14	4,095	0,004	1,87
6,14	0,022	3,10	6,143	0,030	2,69
7,16	0,061	3,58	7,167	0,086	3,11
8,19	0,170	4,06	8,191	0,239	3,52
9,21	0,475	4,56	9,215	0,665	3,98
10,23	1,321	5,19	10,239	1,852	4,68

Таблиця 7.2.4

Визначення першої та другої похідної $y_1 = \frac{dy}{dx}$, $y_2 = \frac{dy_1}{dx} = \frac{d^2y}{dx^2}$ від абсциси x для вовняний крученої пряжі

Вовняна кручена пряжа (гребінного прядіння) 36 Текс					
Ізотропна			Анізотропна		
x	y_1	y_2	x	y_1	y_2
0	0	0,18	0	0	0,18
0,001	0,00002002	0,1804	0,001	0,0000400	0,1803
0,003	0,0000200	0,181	0,003	0,0000401	0,181
0,007	0,0000201	0,182	0,007	0,0000402	0,182
0,01	0,0000203	0,186	0,015	0,0000406	0,185
0,03	0,0000206	0,19	0,03	0,000041	0,19
0,06	0,0000213	0,20	0,06	0,000042	0,20
0,12	0,0000227	0,23	0,12	0,00004	0,22
0,25	0,00002	0,28	0,25	0,00005	0,26
0,511	0,00003	0,39	0,51	0,00006	0,35
1,02	0,00005	0,61	1,02	0,00011	0,53
2,04	0,00015	1,04	2,04	0,00039	0,89
4,09	0,0011	1,91	4,09	0,002	1,61
6,14	0,008	2,78	6,14	0,017	2,33
8,19	0,066	3,65	8,19	0,133	3,05
9,21	0,185	4,09	9,21	0,371	3,43
10,23	0,516	4,55	10,23	1,033	3,88

За формулою (2.1.13) визначаємо значення натягу нитки з урахуванням значень першої похідної $\frac{dy}{dx}$, яку беремо з таблиць 7.2.1-7.2.4. На рис.7.2.7 представлені результати розрахунку.

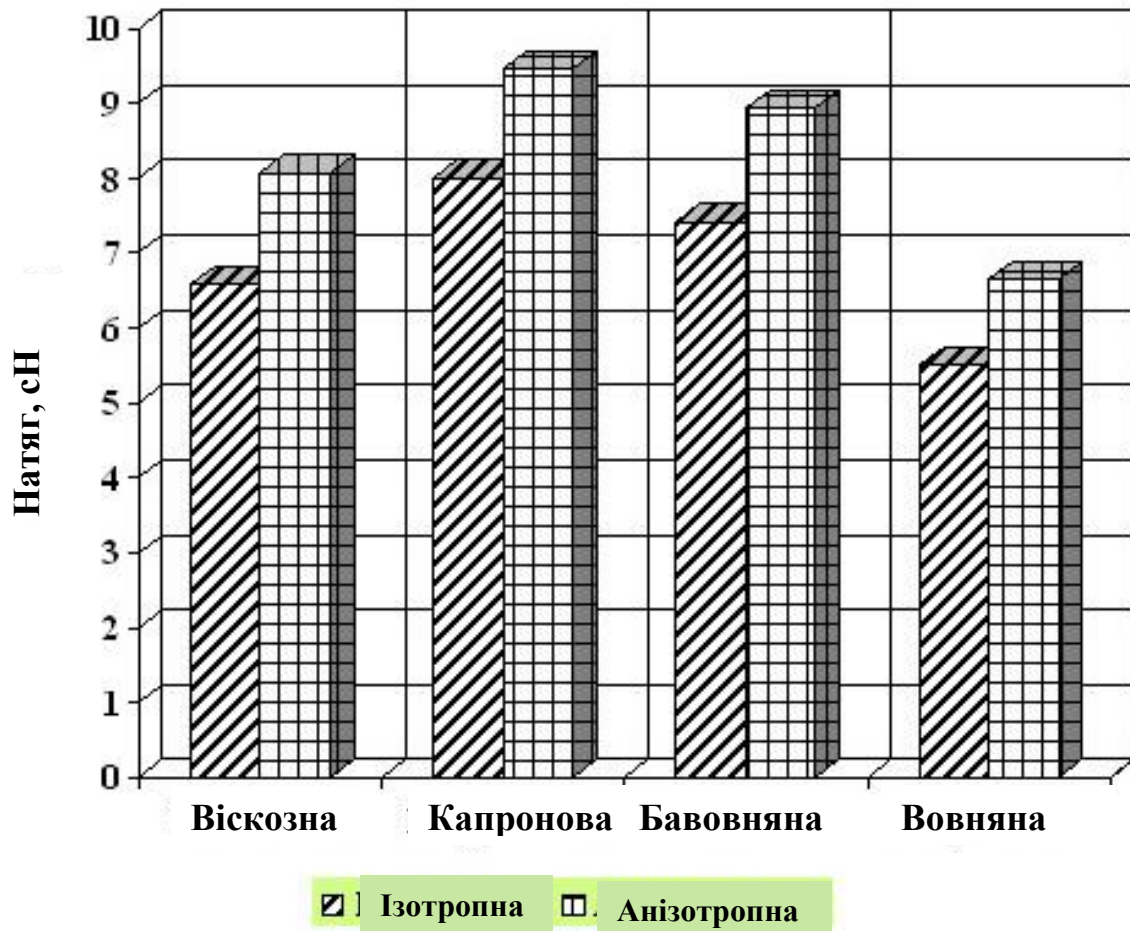


Рис.7.2.7. Визначення натягу ниток з урахуванням анізотропії фрикційних властивостей

Аналіз отриманих графічних залежностей показав, що для всіх ниток найбільший натяг матимуть нитки з анізотропними фрикційними властивостями. Реальний натяг, в середньому, у анізотропних ниток в порівнянні з ізотропним буде більше на 20-30%.

Апроксимаційні залежності для визначення форми осі нитки при поперечному ковзанні по площині

Апроксимація розрахункових (експериментальних) даних передбачає перебування коефіцієнтів полінома

$$y(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m, \quad (7.2.2.1)$$

де $y(x)$ - значення апроксимуючої функції;

$a_0, a_1 \dots a_m$ - значення постійних коефіцієнтів шуканого полінома.

На рис.7.2.2.1 представлена блок-схема алгоритму апроксимації та інтерполяції за методом найменших квадратів з можливістю автоматичного вибору ступеня полінома.

Поліномінальна регресія з автоматичним вибором ступеня полінома виконується за наступним алгоритмом. Спочатку задається число пар N і задається середньоквадратична похибка E_1 . Після цього задається ступінь полінома $m = 1$ (на початковому етапі приймаємо лінійну регресію).

На наступному етапі вводимо значення x_i, y_i для кожної пари. Після цього визначаємо корені системи лінійних рівнянь методом Гауса з вибором головного елемента

$$\begin{aligned}
 c_0a_0 + c_1a_1 + c_2a_2 + \dots + c_ma_m &= d_0, \\
 c_1a_0 + c_2a_1 + c_3a_2 + \dots + c_{m+1}a_m &= d_1, \\
 &\dots \\
 c_ma_0 + c_{m+1}a_1 + c_{m+2}a_2 + \dots + c_{2m}a_m &= d_m,
 \end{aligned} \quad (7.2.2.2)$$

$$c_j = \sum_{i=1}^N x_i^j, \quad j = 0, 1, 2, \dots, 2m,$$

$$d_k = \sum_{i=1}^N x_i^k y_i, \quad k = 0, 1, 2, \dots, m.$$

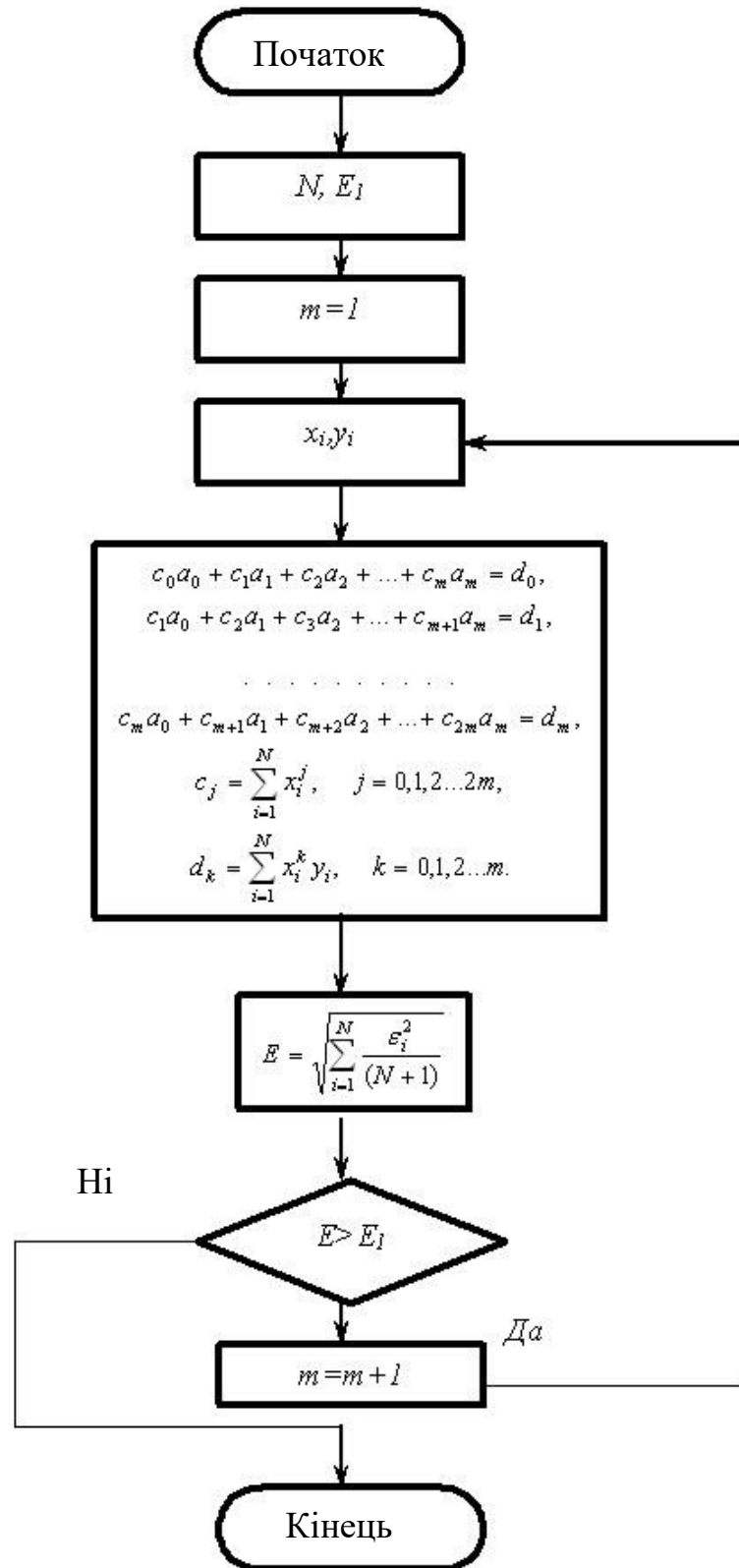


Рис.7.2.2.1. Блок-схема алгоритму апроксимації та інтерполяції

На наступному етапі визначаємо мінімальну середньоквадратичну похибку за формулою

$$E = \sqrt{\sum_{i=1}^N \frac{\varepsilon_i^2}{(N+1)}} \quad (7.2.2.3)$$

Отримане значення (7.2.2.3) порівнюємо із заданою середньоквадратичною похибкою E_1 . У разі якщо $E > E_1$ ступінь полінома збільшуємо на 1 і повторюємо обчислення коренів системи лінійних рівнянь методом Гаусса. В разі $E < E_1$ обчислення закінчуються і результати видаються на друк.

Якщо $m = N$, то має місце звичайна інтерполяція. У цьому випадку значення $y(x)$ при $x = x_i$ точно збігаються із заданими y_i . При $m < N$ такого збігу немає.

Для реалізації даного алгоритму була написана програма на мові Object Pascal в середовищі Delphi. Програма наведена в додатку Б. На рис.7.2.2.2 представлена перша форма `TfrmAPIN1 = class (TForm)`.



Рис.7.2.2.2. Перша форма `TfrmAPIN1 = class (TForm)` програми

На цій формі розташовано вісім міток: `lbl1APIN1: TLabel;`
`lbl2APIN1: TLabel;` `lbl3APIN1: TLabel;` `lbl4APIN1: TLabel;` `lbl5APIN1:`

TLabel; lbl6APIN1: TLabel;lbl7APIN1: TLabel; Label1: TLabel Компонент
img1APIN1: Timage призначений для розміщення малюнка. Кнопка
btn1APIN1: Tbutton призначена для переходу на другу основну форму.

На рис.7.2.2.3 представлена друга основна форма TfrmAPIN2 = class
(Tform). На цій формі розташовано п'ять міток: lbl1APIN2:
TLabel; lbl2APIN2: TLabel; lbl3APIN2: TLabel; lbl4APIN2:
TLabel; lbl5APIN2: TLabel. Для введення даних використовуються три
однорядкових вікна редагування: edt1APIN2: TEdit; edt2APIN2: TEdit;
edt3APIN2: Tedit.

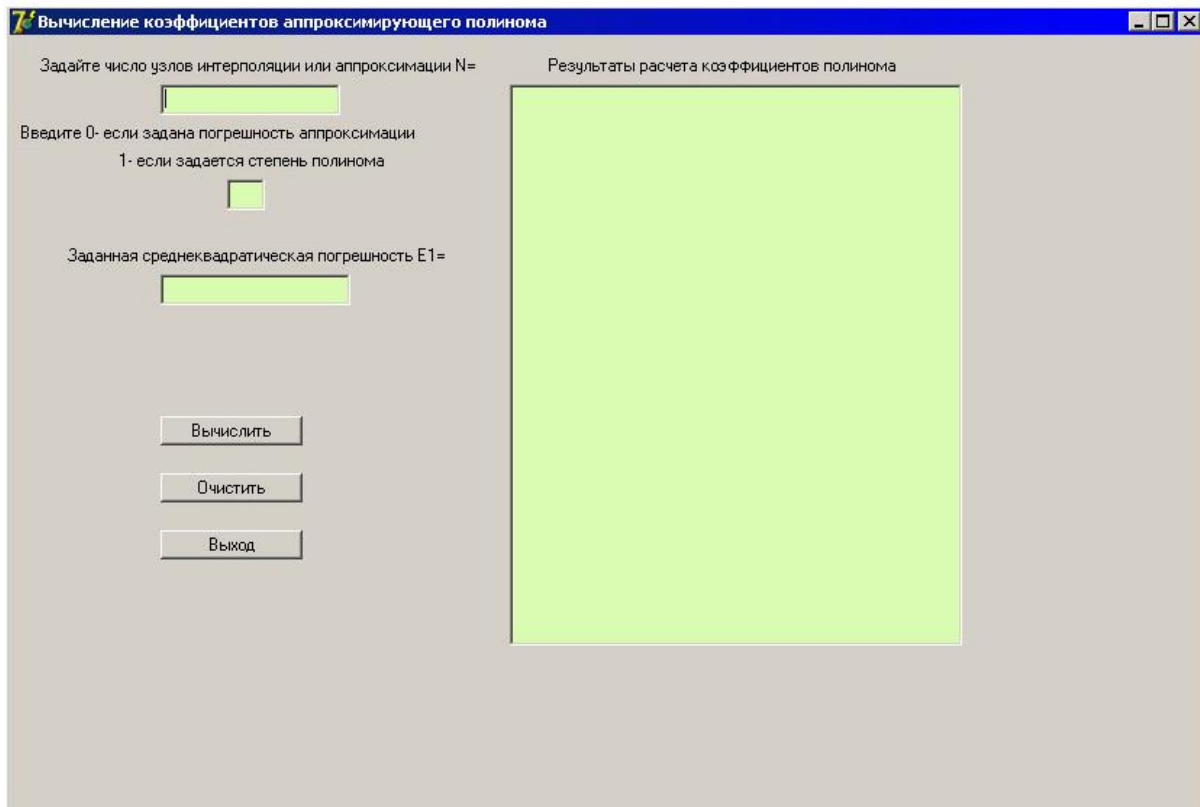


Рис.7.2.2.3. Друга, основна форма TfrmAPIN2 = class (Tform)
програми

Вікно edt1APIN2: Tedit призначене для введення числа вузлів
апроксимації або інтерполяції. Вікно edt2APIN2: Tedit призначене для
введення значень 0 або 1, коли задається похибка апроксимації або ступінь
апроксимаційного полінома.

Третє вікно edt3APIN2: Tedit призначене для введення середньоквадратичної похибки. Багаторядкове вікно редагування mem1APIN2: Tmemo призначене для виведення результатів розрахунку.

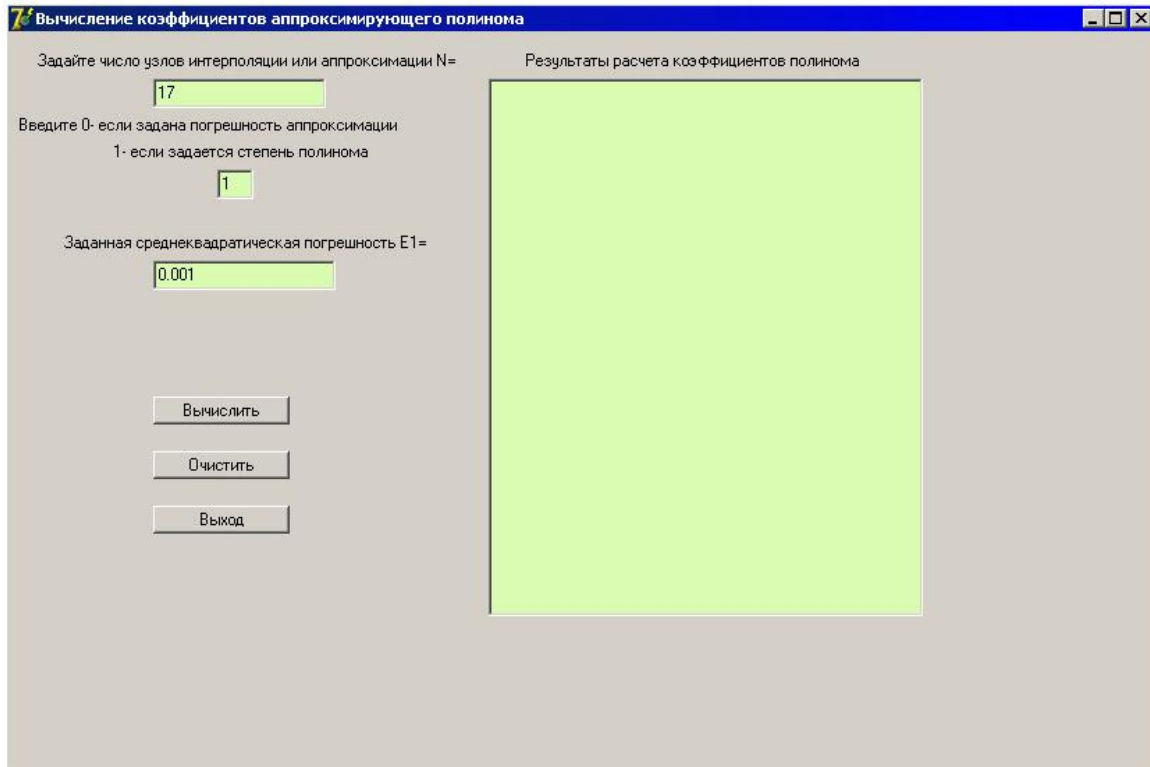


Рис .7.2.2.4. Форма TfrmAPIN 2 = class (Tform) з введеними даними

На формі розташовані три кнопки btn1APIN2: TButton; btn2APIN2: TButton; btn3APIN2: Tbutton. Перша кнопка btn1APIN2: Tbutton призначена для запуску процесу обчислення. Друга кнопка btn2APIN2: Tbutton призначена для очищення всіх вікон. Третя кнопка btn3APIN2: Tbutton призначена для виходу з програми.

Опис всіх типів даних, що застосовуються в програмі, має вигляд

```

var
    frmAPIN2: TfrmAPIN2;
    i,H,VV,N,j,k,N1,k1,j1,M,L,M2,code:Integer;
    E1,R,F,U,S,E,p,Q1:Real;
    X:array[1..50]of Real;
    Y:array[1..50]of Real;
    Z:array[1..50]of Real;
    B:array[1..50]of Real;
    
```

```

G:array[1..50]of Real;
A:array[1..50,1..50]of Real;
C:array[1..50,1..50]of Real;
D:array[1..100]of Real;
vv7,EE1,EE2,NN1,EEE,ikpl,kpl:String;
aa:array[1..50]of String;
aa1:array[1..50]of String

```

На рис.7.2.2.5 представлена основна форма для введення пар даних

x_i , y_i Застосований для цього код має вигляд

```

begin
    vv7:=format('%3.0d',[i]);
    aa[i]:=InputBox('Введення даних,' X['+vv7+']='');
    Val(aa[i],X[i],code);
    mem1APIN2.Lines.Add('X('+vv7+')='+aa[i]);
    aa1[i]:=InputBox('Введення значень,' Y['+vv7+']='');
    Val(aa1[i],Y[i],code);
    mem1APIN2.Lines.Add('Y('+vv7+')='+aa1[i]);
end;

```

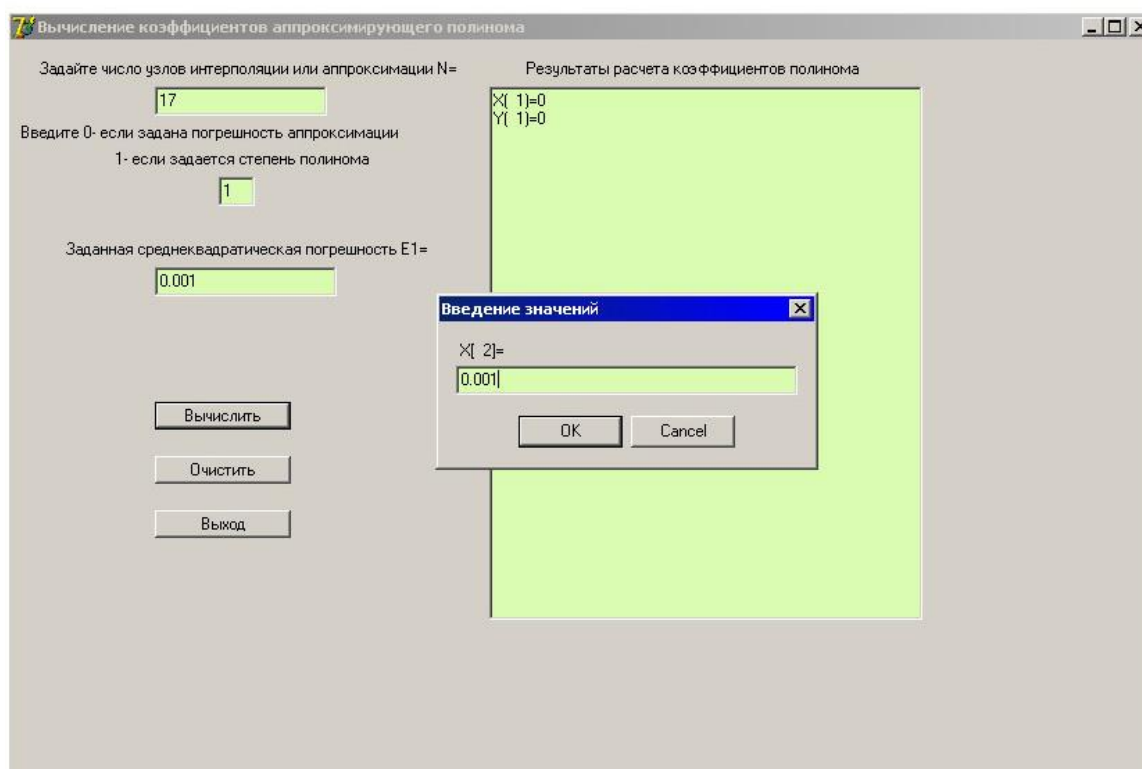


Рис.7.2.2.5. Введення пар даних

На рис.7.2.2.6 показана основна форма з результатами розрахунків коефіцієнтів в рівнянні регресії.

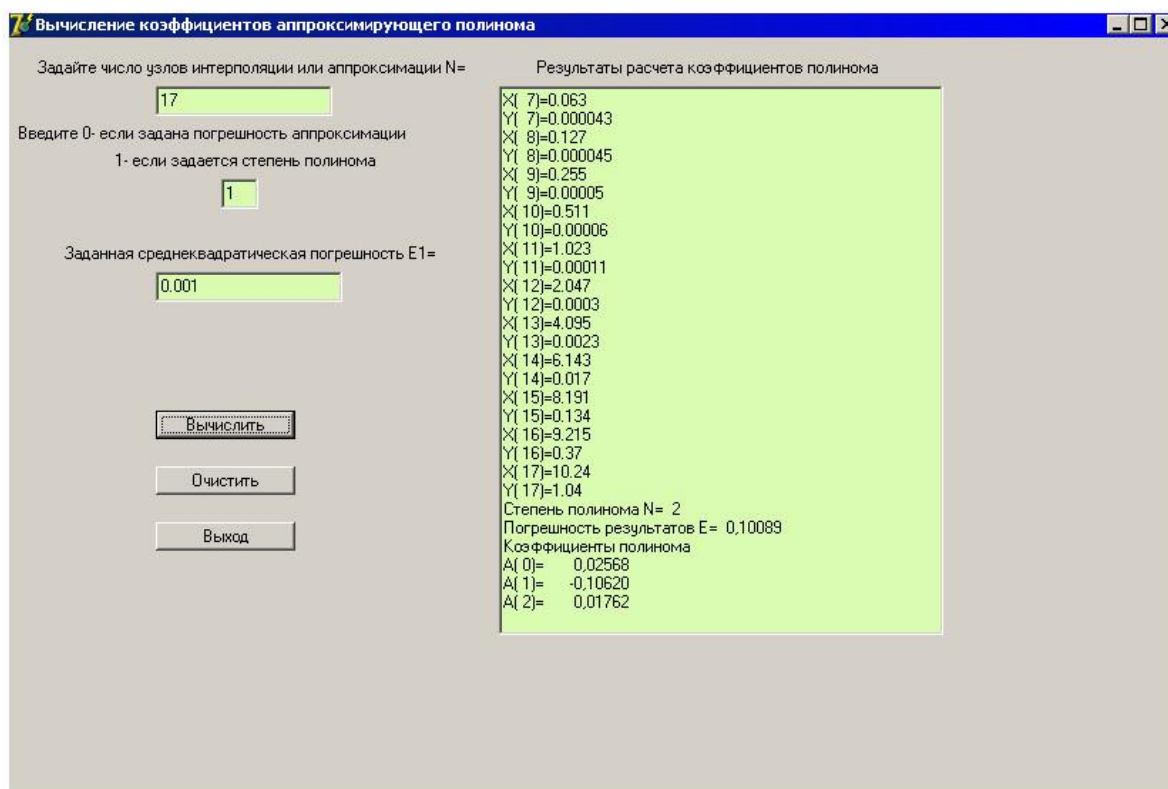


Рис.7.2.2.6. Результаты розрахунку коефіцієнтів в рівняннях регресії

Використовуючи розроблене програмне забезпечення і дані таблиць 7.2.1-7.2.4 були отримані регресійні залежності (7.2.2.4)-(7.2.2.11) зміни Y' та Y'' в залежності від зміни абсциси x .

Для проведення обчислень використовувалися дані: для віскозної комплексної нитки 16,7 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,16$, $k_2 = 0,21$); для капронової комплексної нитки 31,2 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,18$, $k_2 = 0,24$); для бавовняної пряжі (гребінного прядіння, кручена) 23,6 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,17$, $k_2 = 0,22$); для вовняної пряжі (гребінного прядіння, кручена) 36 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,14$, $k_2 = 0,18$).

При проведенні обчислень для значень першої похідної використовувався поліном другого ступеня, а для значень другої похідної - лінійна залежність. Такий вибір обумовлювався характером зміни поведінки даних функцій на виділеному інтервалі.

Для віскозних комплексних нитки 16,7 Текс були отримані наступні регресійні залежності:

для ізотропних фрикційних властивостей

$$\begin{aligned} y' &= 0,018x^2 - 0,11x + 0,03, \\ y'' &= 0,46x + 0,21. \end{aligned} \tag{7.2.2.4}$$

для анізотропних фрикційних властивостей

$$\begin{aligned} y' &= 0,03x^2 - 0,18x + 0,04, \\ y'' &= 0,39x + 0,2. \end{aligned} \tag{7.2.2.5}$$

Для капронової комплексної нитки 31,2 Текс були отримані наступні регресійні залежності:

для ізотропних фрикційні властивостей

$$\begin{aligned} y' &= 0,03x^2 - 0,18x + 0,04, \\ y'' &= 0,49x + 0,23. \end{aligned} \tag{7.2.2.6}$$

для анізотропних фрикційних властивостей

$$\begin{aligned}y' &= 0,038x^2 - 0,24x + 0,06, \\y'' &= 0,43x + 0,22.\end{aligned}\tag{7.2.2.7}$$

Для бавовняної пряжі 23,6 Текс були отримані наступні регресійні залежності:

для ізотропних фрикційних властивостей

$$\begin{aligned}y' &= 0,024x^2 - 0,15x + 0,04, \\y'' &= 0,48x + 0,21.\end{aligned}\tag{7.2.2.8}$$

для анізотропних фрикційних властивостей

$$\begin{aligned}y' &= 0,033x^2 - 0,21x + 0,05, \\y'' &= 0,42x + 0,21.\end{aligned}\tag{7.2.2.9}$$

Для вовняної пряжі 36 Текс були отримані наступні регресійні залежності:

для ізотропних фрикційні властивостей

$$\begin{aligned}y' &= 0,009x^2 - 0,05x + 0,01, \\y'' &= 0,43x + 0,18.\end{aligned}\tag{7.2.2.10}$$

для анізотропних фрикційних властивостей

$$\begin{aligned}y' &= 0,018x^2 - 0,1x + 0,03, \\y'' &= 0,36x + 0,18.\end{aligned}\tag{7.2.2.11}$$

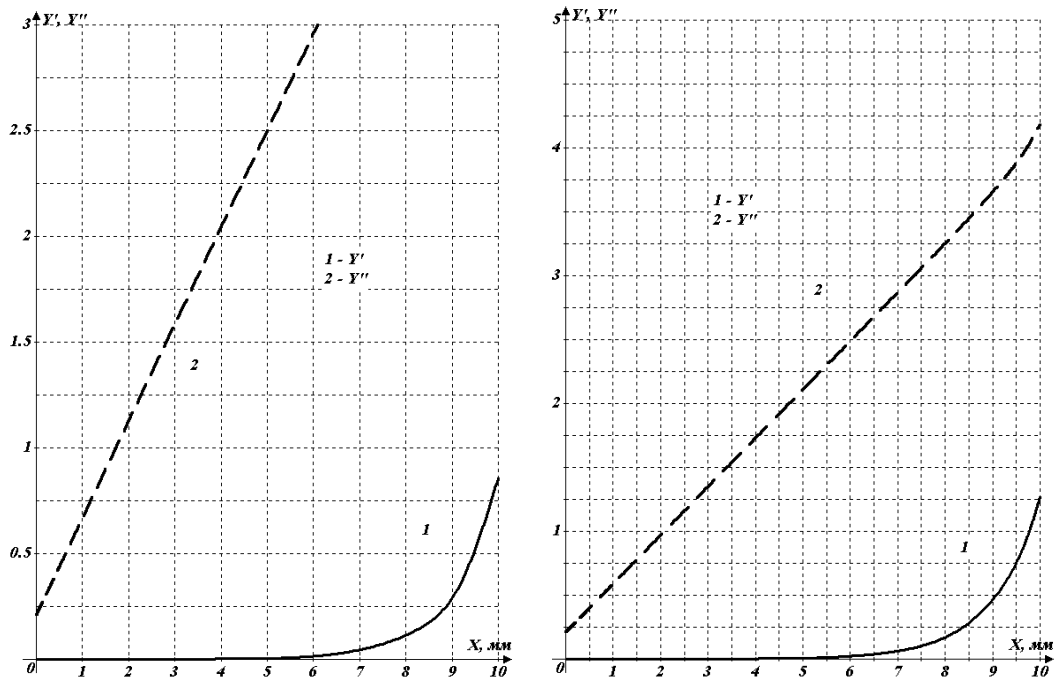


Рис.7.2.2.7.Залежність зміни Y', Y'' для віскозної комплексної нитки

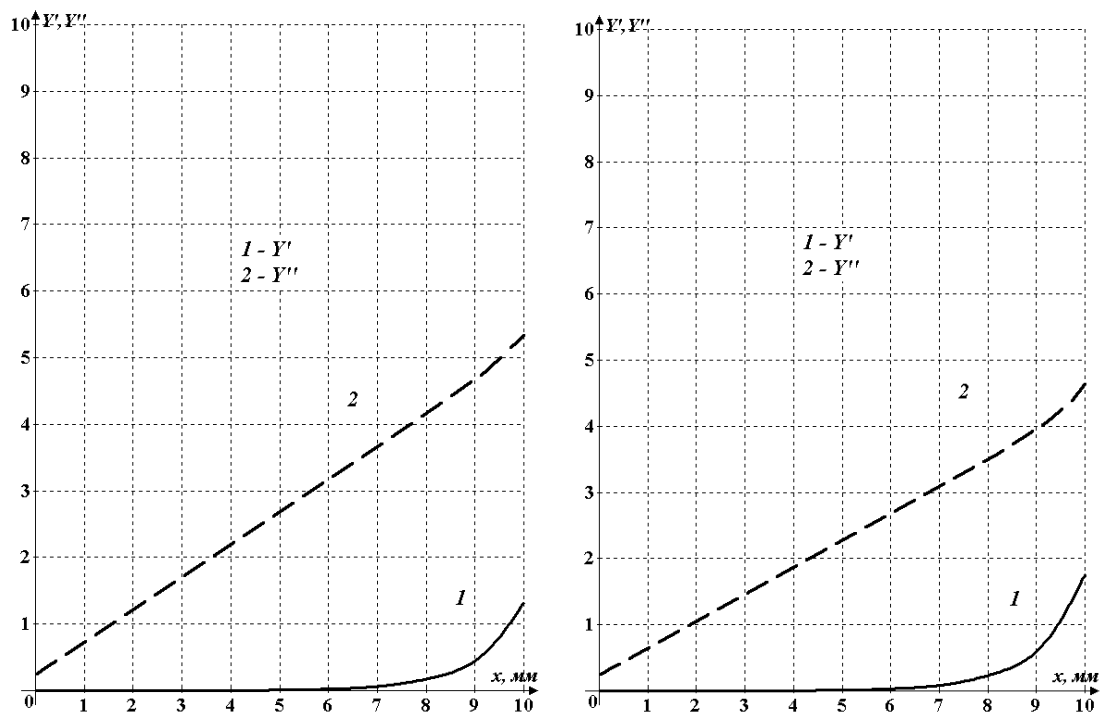


Рис.7.2.2.8.Залежність зміни Y', Y'' для капронової комплексної нитки

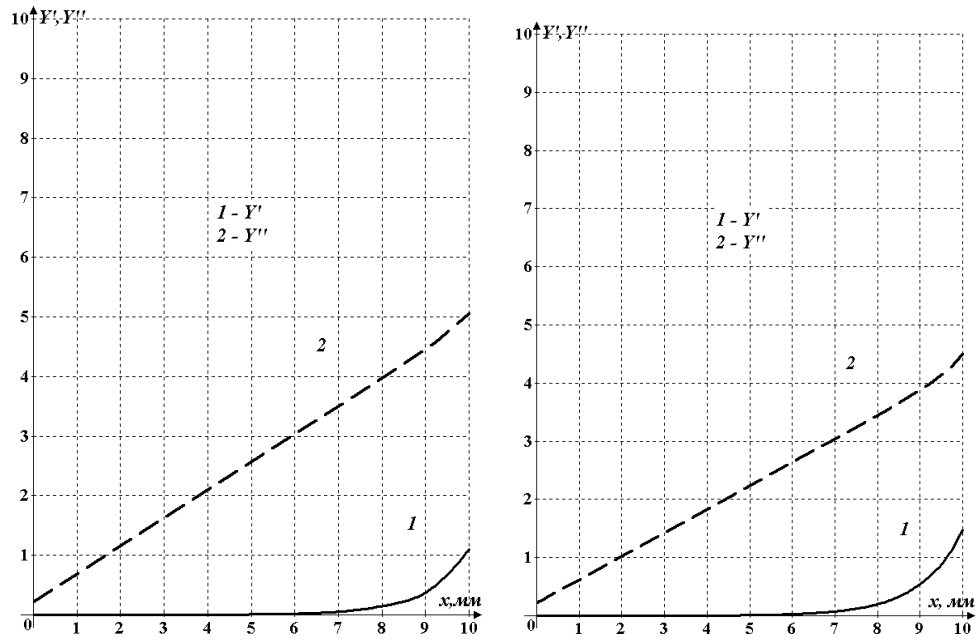


Рис.7.2.2.9. Залежність зміни Y', Y'' для бавовняної комплексної нитки

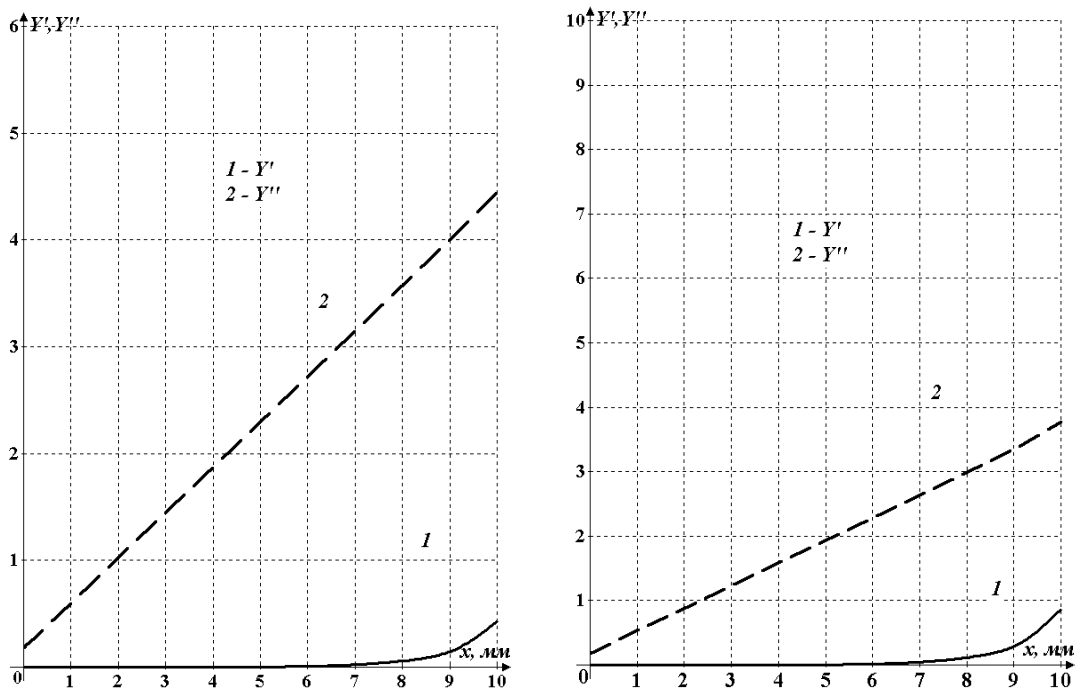


Рис.7.2.2.10. Залежність зміни Y', Y'' для вовняної пряжі

На рис.7.2.2.7-7.2.2.10 представлені графічні залежності зміни значень Y', Y'' . Їх аналіз показує, що нитки мають великий діаметр

поперечного перерізу (більший Текс) мають більш полого форму. Цю обставину необхідно враховувати при проектуванні технологічних процесів легкої і текстильної промисловості.

Дослідження поперечного ковзання жорсткої на вигин нитки по площині

Цей вид взаємодії має місце, коли розрахунковий радіус поперечного перерізу нитки сумарний з відстанню між направляючими кривими А і В (див .рис.2.1.1). У цьому випадку необхідно враховувати згинальну жорсткість нитки. При виведенні залежностей будемо вважати, що згинальний момент пропорційний кривизні осі нитки, а швидкість зміни величини перерізаючої сили по дуговій координаті дуже мала, що пояснюється незначною величиною відхилення осі нитки від лінії з'єднує точки А і В. Крім того, будемо вважати, що маса нитки рівномірно розподілена по її довжині.

Для проведення розрахунків будемо використовувати наступні нитки: віскозна комплексна нитка 16,7 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,16$, $k_2 = 0,21$); капронова комплексна нитка 31,2 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,18$, $k_2 = 0,24$); бавовняна пряжа (гребінного прядіння, кручена) 23,6 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,17$, $k_2 = 0,22$); вовняна пряжа (гребінного прядіння, кручена) 36 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,14$, $k_2 = 0,18$).

Величина кута нахилу дотичної визначалася за формулою (2.2.15)

$$\cos \varphi = {}^{(k_1+k_2)} \sqrt{\frac{q^{k_2}}{q_0^{k_2}} e^{\frac{B}{3\mu_0 g} (q_0^3 - q^3)}}, \quad (7.2.3.1)$$

де φ - кут нахилу дотичної до осі абсцис;

q - кривизна осі нитки, 1/м;

q_0 - кривизна осі нитки в точці 0, 1/м;

B - коефіцієнт згинальної жорсткості, Нм²;

μ_0 - лінійна щільність нитки, кг/м;

g - прискорення вільного падіння, м/с².

Натяг нитки визначається за формулою (2.2.11)

$$P = \frac{k_2 \mu_0 g \cos \varphi}{q} \quad (7.2.3.2)$$

де P - натяг нитки.

З урахуванням (7.2.3.1), формула (7.2.3.2) прийме вигляд

$$P = \frac{k_2 \mu_0 g}{q} (k_1 + k_2) \sqrt{\frac{q^{k_2}}{q_0^{k_2}} e^{\frac{B}{3\mu_0 g} (q_0^3 - q^3)}} \quad (7.2.3.3)$$

Для виконання розрахунків прийняті наступні дані, наведені в таблиці 3.7.2.

Таблиця 7.2.3.1

Фізико-механічні показники ниток

Показники	Віскозна комплексна нитка, 16,7Т	Капронова комплексна нитка, 31,2Т	Бавовняна кручена пряжа, 23,6Т	Вовняна кручена пряжа, 36Т
Жорсткість нитки на вигин, Н м ²	0,2·10 ⁻⁸	0,22·10 ⁻⁸	0,7·10 ⁻⁸	0,25·10 ⁻⁸
$\mu_0 g$, Н/м	1,64·10 ⁻³	3,06·10 ⁻³	2,32·10 ⁻³	3,53·10 ⁻³
k_1	0,16	0,18	0,17	0,14
k_2	0,21	0,24	0,22	0,18

Результати розрахунку за формулами (7.2.3.1) і (7.2.3.3) представлені в таблицях 7.2.3.2 і 7.2.3.3.

Таблиця 7.2.3.2

Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P

Віскозна комплексна нитка, 16,7Т			Капронова комплексна нитка, 31,2Т		
q	φ	P	q	φ	P
1	0	25	1	0	25
0.97	0.158	25	0.97	0.373	25
0.94	0.411	25	0.94	0.866	25.001
0.91	0.723	25	0.91	1.426	25.002
0.88	1.088	25.001	0.88	2.041	25.003
0.85	1.5	25.001	0.85	2.707	25.004
0.82	1.958	25.001	0.82	3.422	25.005
0.79	2.461	25.002	0.79	4.185	25.007
0.76	3.012	25.002	0.76	4.997	25.008
0.73	3.61	25.003	0.73	5.86	25.01
0.7	4.259	25.004	0.7	6.774	25.012
0.67	4.96	25.004	0.67	7.743	25.015
0.64	5.716	25.005	0.64	8.77	25.018
0.61	6.533	25.006	0.61	9.859	25.021
0.58	7.414	25.008	0.58	11.014	25.024
0.55	8.364	25.009	0.55	12.242	25.029
0.52	9.391	25.011	0.52	13.548	25.033
0.49	10.503	25.013	0.49	14.94	25.039
0.46	11.709	25.015	0.46	16.428	25.046
0.43	13.019	25.018	0.43	18.023	25.054
0.4	14.449	25.022	0.4	19.738	25.063
0.37	16.015	25.026	0.37	21.59	25.075
0.34	17.738	25.031	0.34	23.6	25.089
0.31	19.646	25.038	0.31	25.794	25.107
0.28	21.774	25.047	0.28	28.205	25.129
0.25	24.17	25.058	0.25	30.879	25.158
0.22	26.897	25.073	0.22	33.876	25.197
0.19	30.049	25.095	0.19	37.282	25.252
0.16	33.763	25.127	0.16	41.226	25.33
0.13	38.26	25.177	0.13	45.91	25.453
0.1	43.929	25.264	0.1	51.687	25.663
0.07	51.562	25.443	0.07	59.263	26.085
0.04	63.234	25.95	0.04	70.456	27.258
10e-3	89.782	30.397	10e-3	94.395	37.099

Таблиця 7.2.3.3

Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P

Бавовняна кручена пряжа, 23,6 Т			Вовняна кручена пряжа, 36 Т		
q	φ	P	q	φ	P
1	0	25	1	0	25
0.97	0.227	25	0.97	0.049	25
0.94	0.561	25.001	0.94	0.149	25
0.91	0.96	25.001	0.91	0.288	25
0.88	1.414	25.001	0.88	0.461	25.001
0.85	1.917	25.002	0.85	0.669	25.001
0.82	2.469	25.003	0.82	0.911	25.001
0.79	3.067	25.003	0.79	1.187	25.002
0.76	3.714	25.004	0.76	1.5	25.002
0.73	4.41	25.005	0.73	1.851	25.003
0.7	5.157	25.007	0.7	2.242	25.004
0.67	5.958	25.008	0.67	2.676	25.004
0.64	6.816	25.009	0.64	3.156	25.005
0.61	7.734	25.011	0.61	3.686	25.007
0.58	8.719	25.013	0.58	4.269	25.008
0.55	9.774	25.016	0.55	4.913	25.01
0.52	10.907	25.019	0.52	5.622	25.012
0.49	12.125	25.022	0.49	6.404	25.014
0.46	13.437	25.026	0.46	7.269	25.018
0.43	14.856	25.031	0.43	8.226	25.021
0.4	16.394	25.037	0.4	9.29	25.026
0.37	18.069	25.044	0.37	10.477	25.031
0.34	19.902	25.052	0.34	11.807	25.039
0.31	21.919	25.063	0.31	13.307	25.048
0.28	24.155	25.077	0.28	15.012	25.059
0.25	26.656	25.095	0.25	16.968	25.075
0.22	29.485	25.119	0.22	19.241	25.097
0.19	32.732	25.153	0.19	21.922	25.128
0.16	36.53	25.203	0.16	25.153	25.174
0.13	41.094	25.282	0.13	29.163	25.249
0.1	46.796	25.417	0.1	34.359	25.381
0.07	54.392	25.692	0.07	41.588	25.659
0.04	65.851	26.467	0.04	53.129	26.473
10e-3	91.292	33.132	10e-3	81.434	34.031

За результатами, представленими у таблицях (7.2.3.2) - (7.2.3.3), побудовані графічні залежності (рис.7.2.3.1-7.2.3.4).

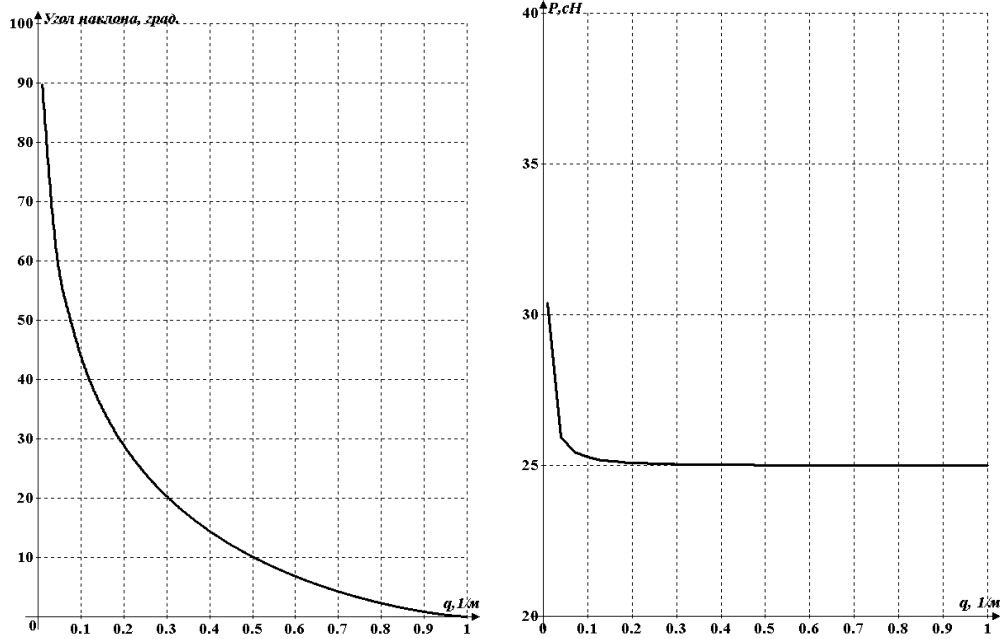


Рис.3.7.2.Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P для віскозної комплексної нитки, 16,7Т

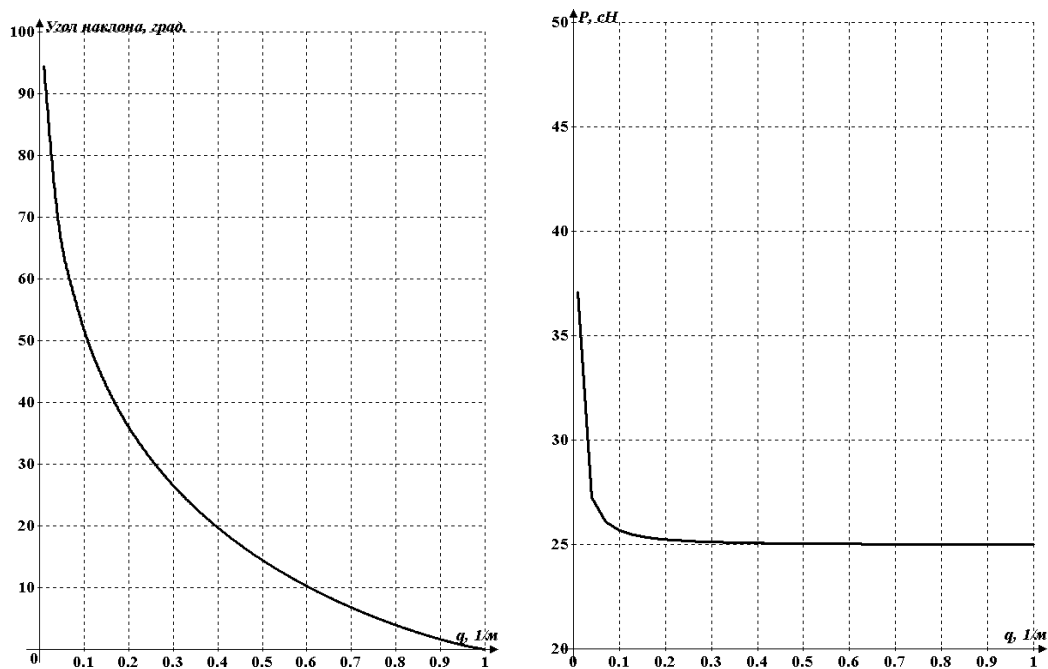


Рис.3.7.2.2. Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P для капронової комплексної нитки, 31,2Т

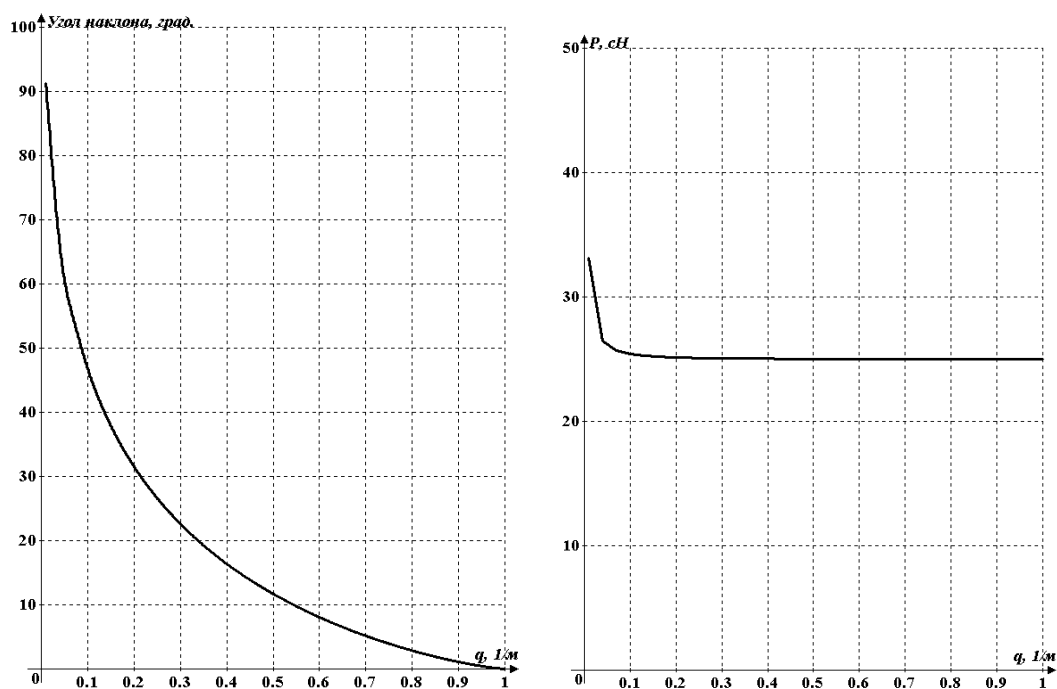


Рис.7.2.3.3. Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P для бавовняної крученої пряжі, 23,6Т

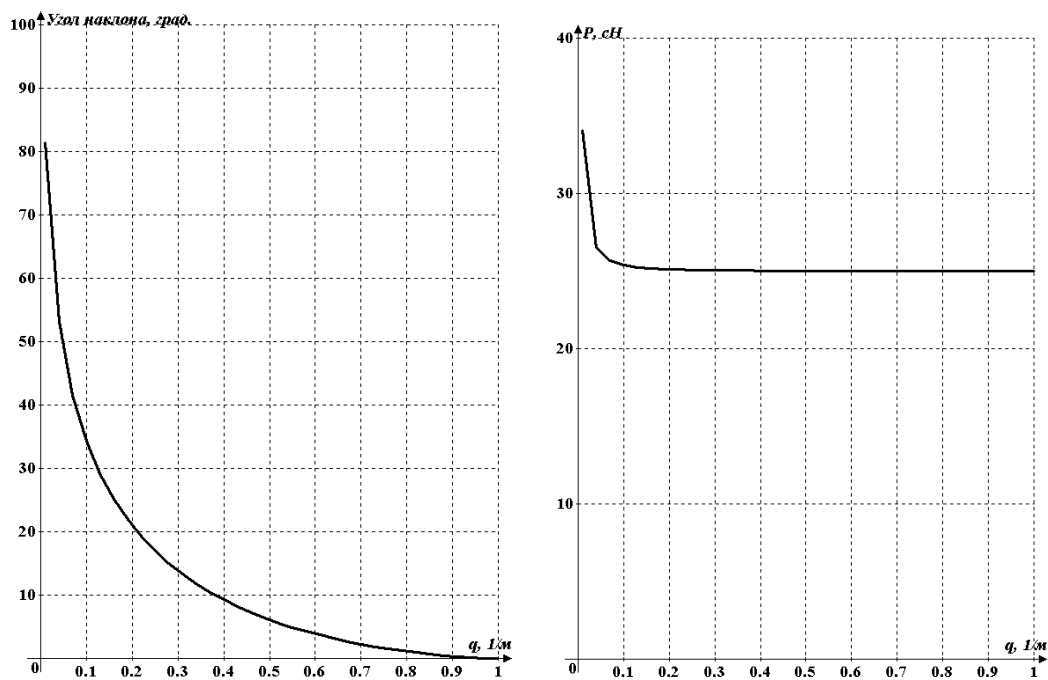


Рис.7.2.7.2.4. Значення кута нахилу φ дотичної до осі абсцис і натягу нитки P для шерстяної крученої пряжі, 36Т

Аналіз отриманих графічних залежностей показує, що нитки більшого текса і більшої жорсткості на вигин мають більш полого форму, що необхідно враховувати при проектуванні технологічних процесів.

Визначення форми осі і натягу нитки при поперечному ковзанні по направляючій малої кривизни з урахуванням анізотропії тертя

Для визначення форми осі і натягу нитки при поперечному ковзанні по циліндричній направляючій з урахуванням анізотропії тертя. Для проведення розрахунків будемо використовувати наступні нитки: віскозна комплексна нитка 16,7 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,16$, $k_2 = 0,21$); капронова комплексна нитка 31,2 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,18$, $k_2 = 0,24$); бавовняна пряжа (гребінного прядіння, кручена) 23,6 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,17$, $k_2 = 0,22$); вовняна пряжа (гребінного прядіння, кручена) 36 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,14$, $k_2 = 0,18$) (див. підрозділ 3.2-3.3).

Для розрахунків натягу і форми осі нитки, при використанні циліндричної системи координат, використовуємо системи диференціальних рівнянь першого порядку (2.3.6), (2.3.11), (2.3.18)

$$\frac{dP}{ds} = \frac{\sqrt{k_1^2 + k_2^2} C_1^2 r P \sqrt{1 - r^2 \left(\frac{C_1}{P}\right)^2}}{\{P^2 [1 - r^2 \left(\frac{C_1}{P}\right)^2] + r^2 C_1^2\}}, \quad (7.2.4.1)$$

$$\frac{dz}{ds} = \sqrt{1 - r^2 \left(\frac{d\varphi}{ds}\right)^2},$$

$$\frac{d\varphi}{ds} = \frac{C_1}{P}.$$

Для інтегрування системи диференціальних рівнянь (7.2.4.1) будемо використовувати метод Рунге-Кутта-Мерсона з автоматичним вибором кроку та програмне забезпечення наведене в підрозділі 7.2. Результати розрахунку наведені в таблицях 7.2.4.1-7.2.4.4.

Таблиця 7.2.4.1

Визначення натягу і форми нитки для віскозної комплексної нитки

16,7 Текс

$s, мм$	$P, сН$	$z, мм$	$\varphi, рад.$
0	20,0	0	0
0,2	20,8	0.28	0.12
0,6	21,7	1.14	0.24
0,9	22,4	2.56	0.36
1,4	22,9	4.56	0.48
2,5	23,5	7.12	0.6
3,5	23,9	10.26	0.72
4,8	24,4	13.97	0.84
6,3	24,9	18.24	0.96
14,6	25,6	23.09	1.08
25,7	26,2	28.51	1.2
37,1	26,9	34.5	1.32
48,9	27,7	41.05	1.44
60,4	28,4	48.18	1.57

Таблиця 7.2.4.2

Визначення натягу і форми нитки для капронової комплексної

нитки 31,2 Текс

$s, мм$	$P, сН$	$z, мм$	$\varphi, рад.$
0	20,0	0	0
0,4	22,3	0,24	0.12
1,5	26,7	0,98	0.24
3,8	28,9	2,21	0.36
5,4	31,1	3,94	0.48
9,3	31,4	6.15	0.6
14,3	32,0	8.86	0.72
20,5	32,3	12.06	0.84
25,1	32,6	15.75	0.96
34,2	32,8	19.94	1.08
45,0	33,3	24.62	1.2
56,8	33,7	29.79	1.32
58,3	34,1	35.45	1.44

60,4	34,4	41,62	1.57
------	------	-------	------

Таблиця 7.2.4.3

Визначення натягу і форми нитки для бавовняної пряжі (гребінного прядіння, кручена) 23,6 Текс

<i>s, мм</i>	<i>P, сН</i>	<i>z, мм</i>	<i>φ, рад.</i>
0	20,0	0	0
0,3	20,8	0.27	0.12
1,2	21,7	1.08	0.24
3,6	22,8	2.44	0.36
5,1	23,4	4.35	0.48
9,0	24,5	6.80	0.6
13,8	25,8	9.79	0.72
19,5	26,7	13.33	0.84
23,1	27,4	17.41	0.96
32,2	28,0	22.04	1.08
41,0	28,8	27.21	1.2
53,8	29,5	32.93	1.32
55,3	30,2	39.19	1.44
60,4	30,6	45.99	1.57

Таблиця 7.2.4.4

Визначення натягу і форми нитки для вовняної пряжі (гребінного прядіння, кручена) 36 Текс

<i>s, мм</i>	<i>P, сН</i>	<i>z, мм</i>	<i>φ, рад.</i>
--------------	--------------	--------------	----------------

0	20,0	0	0
0,1	20,5	0.31	0.12
0,5	20,9	1.24	0.24
0,8	21,5	2.79	0.36
1,1	22,1	4.97	0.48
2,3	22,6	7.77	0.6
3,3	23,1	11.19	0.72
4,5	23,8	15.24	0.84
6,1	24,4	19.90	0.96
14,2	24,9	25.19	1.08
25,0	25,3	31.10	1.2
36,8	25,8	37.63	1.32
48,3	26,0	44.79	1.44
60,4	26,3	52.56	1.57

За даними таблиць 7.2.4.1-7.2.4.4 побудовані апроксимовані графічні залежності, представлені на рис.7.2.4.1-7.2.4.4.

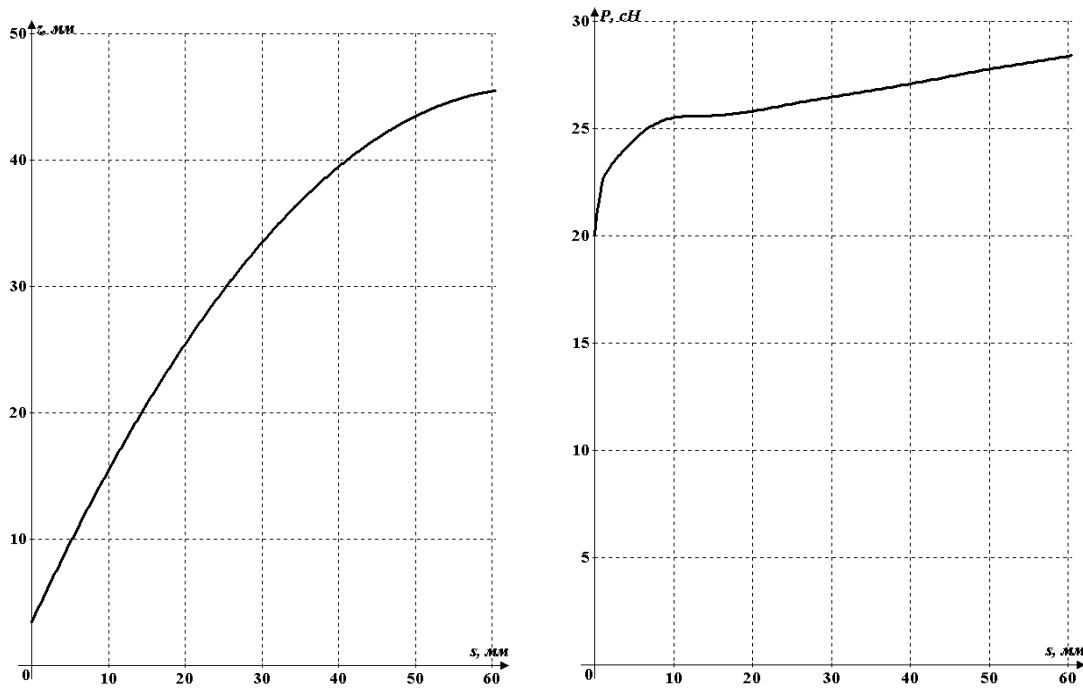


Рис.7.2.4.1. Значення координати z і натягу нитки P для віскозної комплексної нитки, 16,7Т

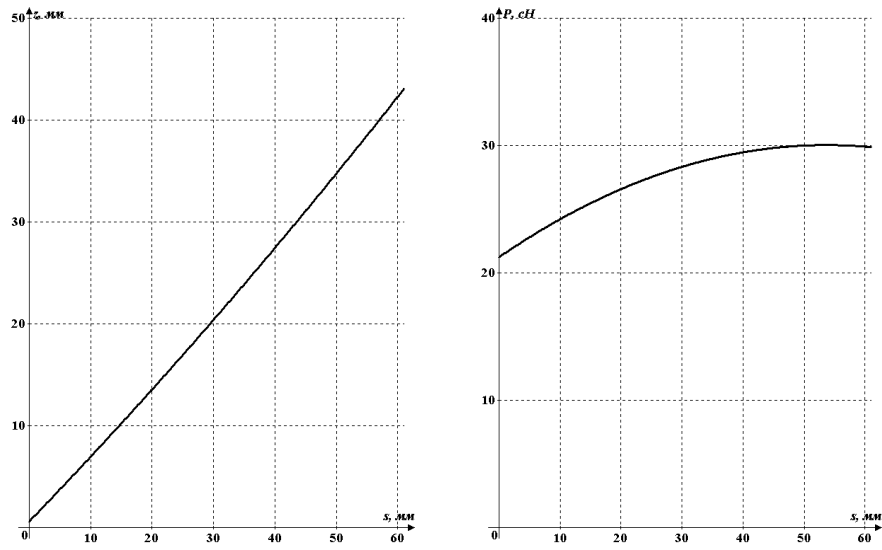


Рис.7.2.4.2. Значення координати z і натягу нитки P для капронової комплексної нитки, 31,2Т

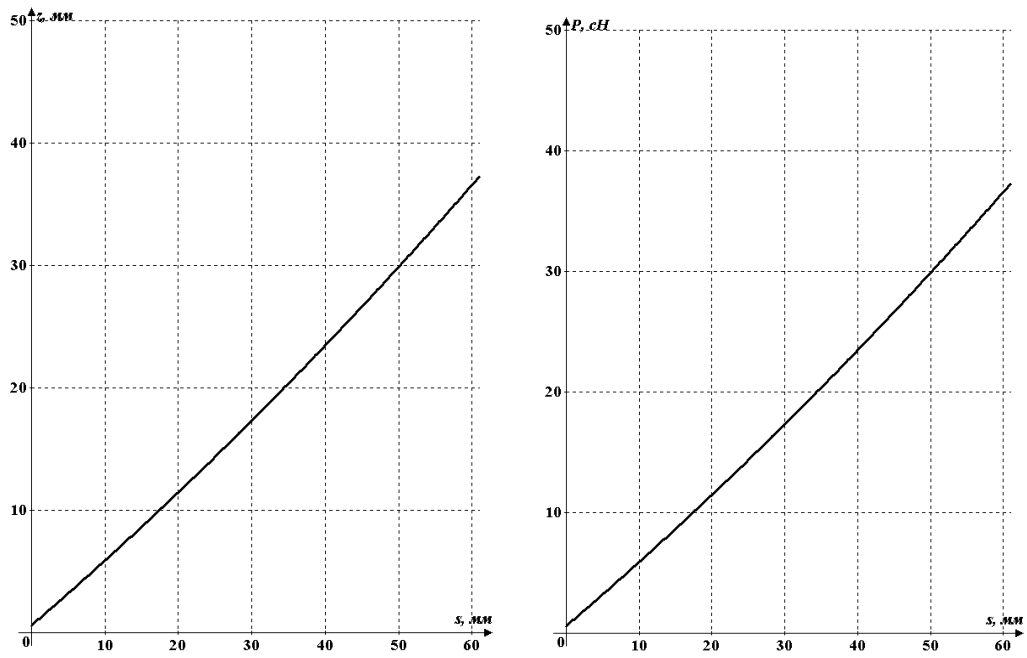


Рис.7.2.4.3. Значення координати z і натягу нитки P для бавовняної крученої пряжі, 23,6Т

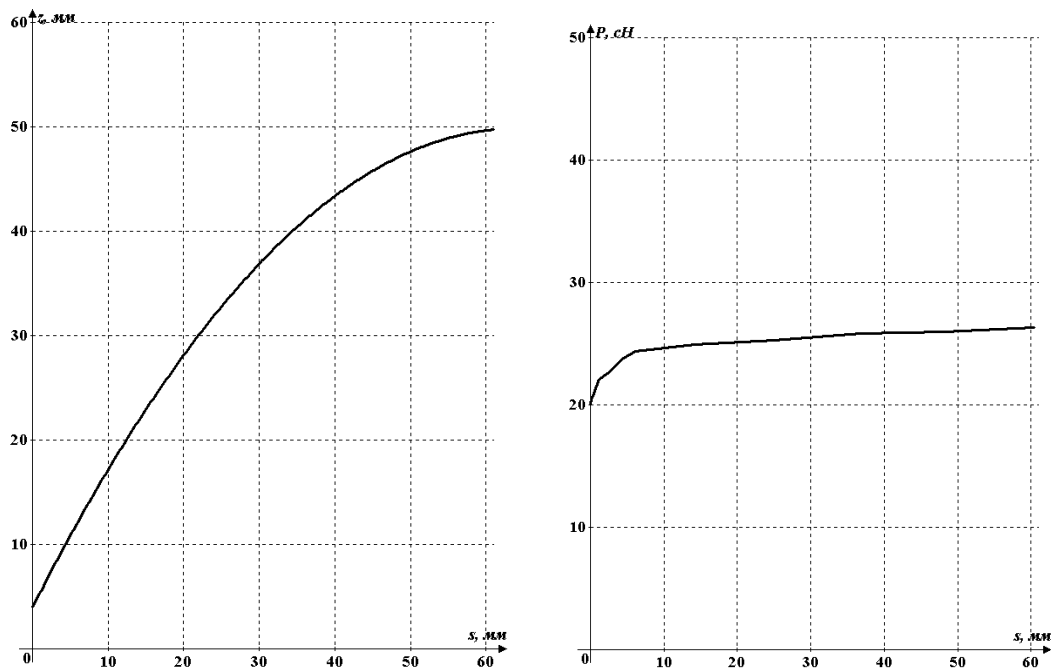


Рис.7.2.4.4. Значення координати z і натягу нитки P для вовняний крученої пряжі, 36т

Аналіз отриманих графічних залежностей (рис.7.2.4.1-7.2.4.4) показав, що нитки, які мають менше значення коефіцієнта тертя, мають більш полого форму. Їх натяг зростає монотонно. У ниток з великими значеннями коефіцієнта тертя натяг на початку зростає більш інтенсивно, що пояснюється її більш опуклою формою.

Визначення форми і натягу жорсткої на вигин нитки з циліндричною направляючою великої кривизни

Для визначення форми осі і натягу жорсткої на вигин нитки при поперечному ковзанні по циліндричній направляючій великої кривизни з урахуванням анізотропії тертя. Для проведення розрахунків (див. підрозділи 3.1-3.4) будемо використовувати наступні нитки: віскозна

комплексна нитка 16,7 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,16$, $k_2 = 0,21$); капронова комплексна нитка 31,2 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,18$, $k_2 = 0,24$); бавовняна пряжа (гребінного прядіння, кручена) 23,6 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,17$, $k_2 = 0,22$); вовняна пряжа (гребінного прядіння, кручена) 36 Текс (коефіцієнти тертя дорівнювали $k_1 = 0,14$, $k_2 = 0,18$) (див. підрозділи 3.2-3.3).

Для визначення натягу нитки скористаємося наступною системою рівнянь (з урахуванням (2.4.9), (2.4.14), (2.4.18) - (2.4.19)), отримаємо

$$P_1 = P_0 e^{\frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{k_1^2 + k_2^2} s},$$

$$\int_0^s ds = \int_0^y \sqrt{\frac{b^4 + y^2(a^2 - b^2)}{b^4 - b^2 y^2}} dy, \quad (7.2.5.1)$$

$$s = b_0 + b_1 y + b_2 y^2 + \dots + b_n^n,$$

$$P_1 = P_0 e^{\frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{k_1^2 + k_2^2} (b_0 + b_1 y + b_2 y^2 + \dots + b_n^n)}.$$

Результати розрахунків зміни координати s і натягу P представлені в таблицях 7.2.5.1-7.2.5.2.

Таблиця 7.2.5.1

Значення координати s і натягу нитки P

у, мм	Віскозна комплексна нитка 16,7 Текс ($\alpha = 12^\circ$)		Капронова комплексна нитка 31,2 Текс ($\alpha = 14^\circ$)	
	s, мм	P, сР	s, мм	P, сР

0	0	20.0	0	20.0
0.1	0.1	20.023	0.1	20.036
0.2	0.23	20.023	0.22	20.04
0.3	0.39	20.05	0.37	20.08
0.4	0.59	20.102	0.55	20.153
0.5	0.86	20.181	0.78	20.262
0.6	1.19	20.287	1.08	20.407
0.7	1.62	20.42	1.44	20.587
0.8	2.17	20.58	1.93	20.805
0.9	2.97	20.769	2.61	21.062
1.0	4.58	20.987	3.99	21.358

Таблиця 7.2.5.2

Значення координати s і натягу нитки P

y , мм	Бавовняна пряжа 23,6 Текс ($\alpha = 13^0$)		Вовняна пряжа 36 Текс ($\alpha = 10^0$)	
	s , мм	P , сР	s , мм	P , сР
0	0	20.0	0	20.0
0.1	0.1	20.029	0.1	20.019
0.2	0.22	20.031	0.24	20.016
0.3	0.38	20.065	0.42	20.039
0.4	0.57	20.131	0.66	20.087
0.5	0.82	20.23	0.96	20.16
0.6	1.13	20.362	1.36	20.259
0.7	1.52	20.528	1.86	20.383
0.8	2.04	20.728	2.53	20.534
0.9	2.78	20.964	3.48	20.712
1.0	4.27	21.236	5.40	20.917

Використовуючи алгоритм для апроксимації розрахункових даних отримані наступні регресійні залежності:

віскозна комплексна нитка 16,7 Текс ($\alpha = 12^0$)

$$s = 0,21 - 1,67y + 5,59y^2, \quad (7.2.5.2)$$

капронова комплексна нитка 31,2 Текс ($\alpha = 14^0$)

$$s = 0,18 - 1,29y + 4,74y^2, \quad (7.2.5.3)$$

бавовняна пряжа 23,6 Текс ($\alpha = 13^0$)

$$s = 0,19 - 1,48y + 5,14y^2, \quad (7.2.5.4)$$

вовняна пряжа 36 Текс ($\alpha = 10^0$)

$$s = 0,25 - 2,16y + 6,79y^2. \quad (7.2.5.5)$$

Отримані вирази (7.2.5.2) - (7.2.5.5) підставлялися в останнє рівняння системи (7.2.5.1).

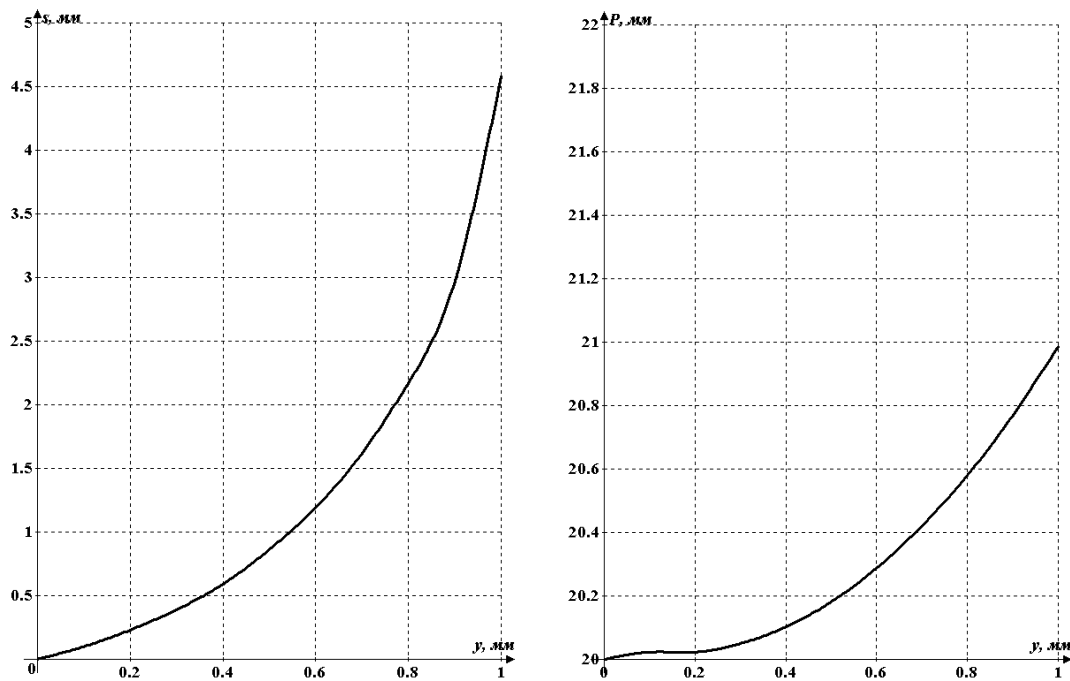


Рис.7.2.5.1. Значення координати s і натягу нитки P для віскозної комплексної нитки, 16,7Т

Результати розрахунків значень координати і натягу нитки представлені на рис.7.2.5.1-7.2.5.4.

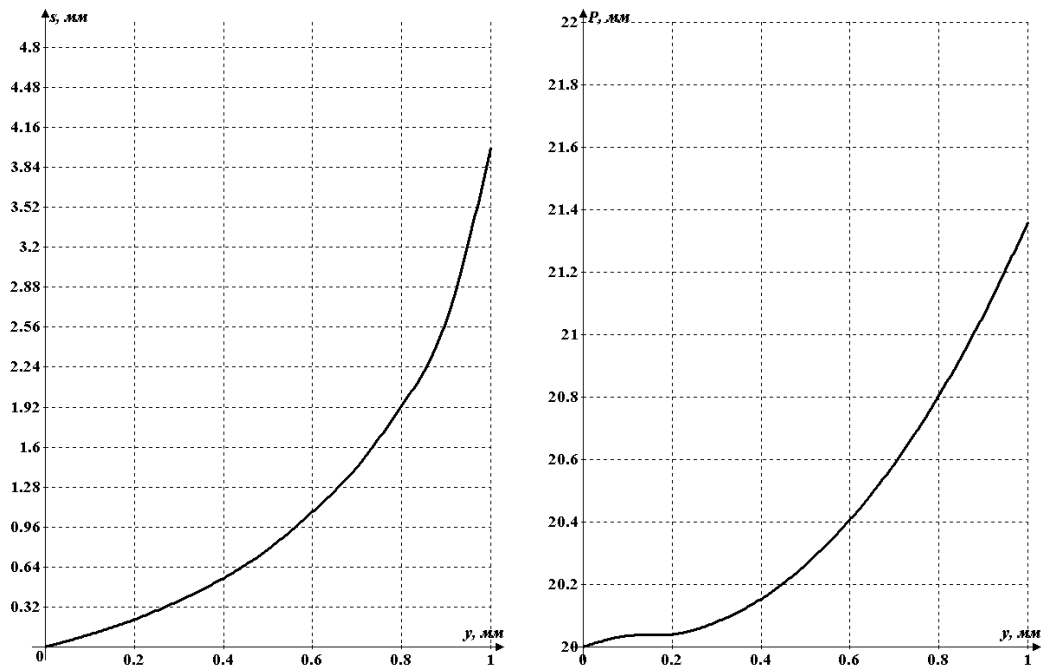


Рис.7.2.5.2. Значення координати s і натягу нитки P для капронової комплексної нитки, 31,2Т

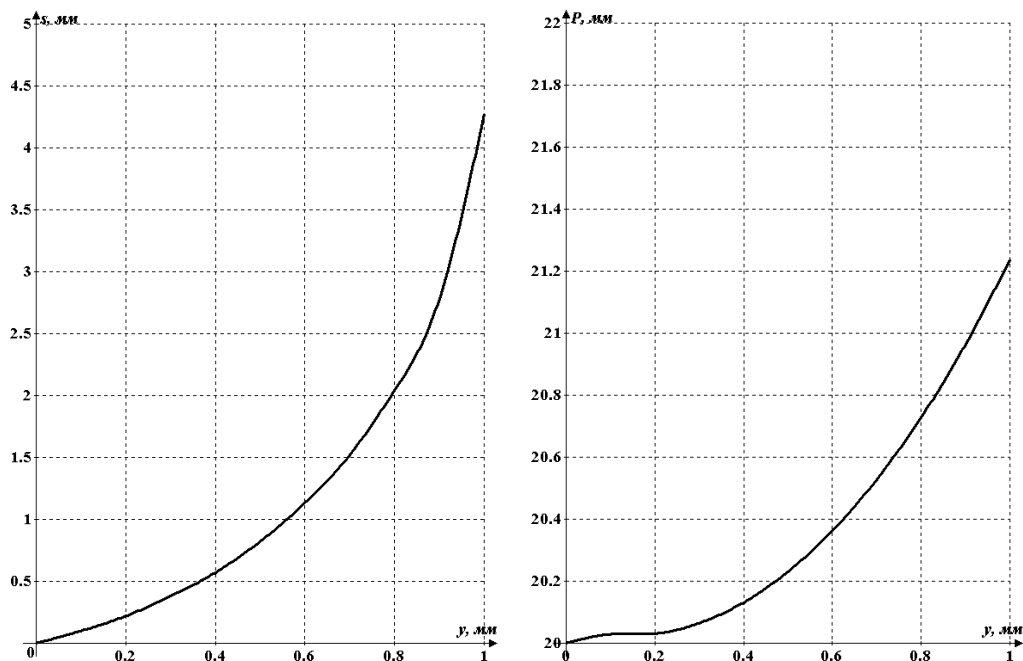


Рис.7.2.5.3. Значення координати s і натягу нитки P для бавовняної крученої пряжі, 23,6Т

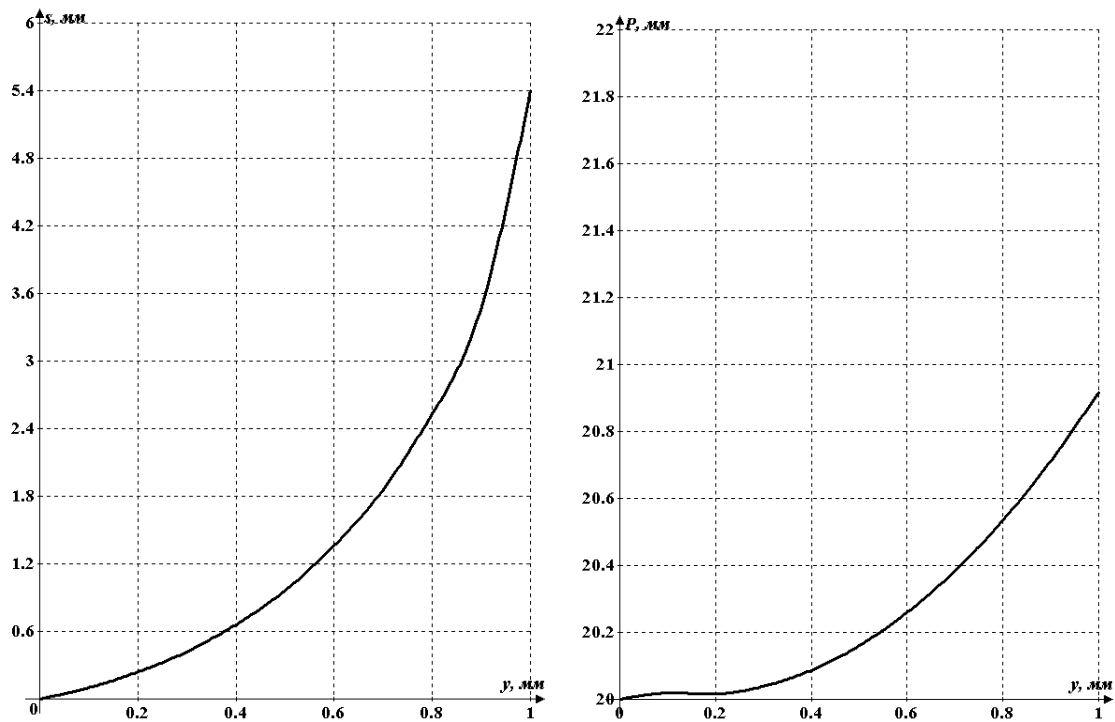


Рис.7.2.5.4. Значення координати y і натягу нитки P для вовняної крученої пряжі, 36т

Аналіз отриманих залежностей показує, що більш полого траєкторію осі і мінімальний натяг мають нитки з меншими значеннями коефіцієнтів поздовжнього і поперечного тертя.

7.3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОЦЕСУ ВЗАЄМОДІЇ НИТОК З НАПРАВЛЯЮЧИМИ З УРАХУВАННЯМ АНІЗОТРОПІЇ ТЕРТЯ

Для перевірки коректності зроблених припущень при теоретичному дослідженні процесу взаємодії комплексних ниток і пряжі з направляючими великої і малої кривизни з урахуванням анізотропії тертя,

а також для визначення величини помилки при вимірюванні величини натягу в роботі були реалізовані дві серії експериментальних досліджень.

Перша серія включала в себе експериментальні дослідження по визначенню впливу геометричних параметрів направляючої і швидкості руху нитки на величину натягу і форму осі нитки в процесі її переробки. Для її реалізації була розроблена спеціальна електротензометрична установка.

Друга серія включала в себе експериментальні дослідження по визначенню впливу геометричних параметрів направляючої і швидкості руху нитки на величину питомого розривного навантаження і відносного розривного подовження. Мається на увазі, що при взаємодії нитки з направляючими відбувається збільшення натягу. Залежно від геометричних параметрів направляючої і швидкості докладання деформації, за рахунок зміни взаємного розташування окремих філаментів відносно один одного в нитці і їх руйнування, відбувається зниження міцності нитки при зміні її відносного розривного подовження. Ця обставина може призвести до обриву нитки при її затягнення в тканину або трикотаж на наступних технологічних переходах.

Експериментальні дослідження проводилися на розривній машині РМ-30-1 Іванівського виробничого об'єднання «Точприбор», що дозволило змінювати швидкість розтягування нитки (швидкість руху нижнього затискача) в межах від 80 мм/хв до 800 мм/хв (рис.7.3.1).

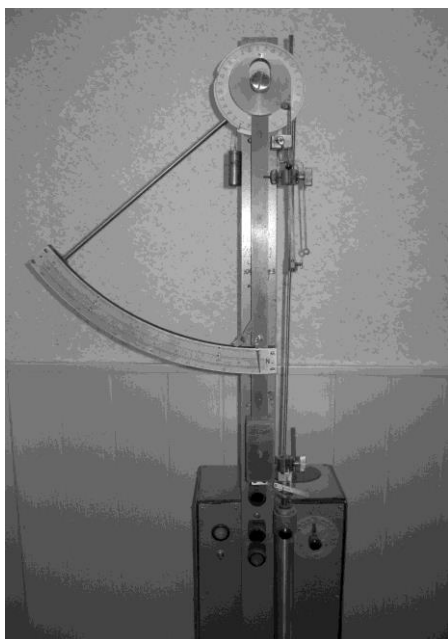


Рис.7.3.1. РМ-30-1

Діапазон варіювання розривного навантаження задається відповідними противагами. Величина першого діапазону змінюється в межах від 10 до 50 Н. Величина другого діапазону змінюється в межах від 20 до 100 Н. Величина третього діапазону змінюється в межах від 60 до 300 Н.

Розробка електротензометричної установки для першої серії дослідів обумовлена тим, що використовувані ручні механічні тензометри володіють низькою точністю вимірювання. Рухома нитка має подовжню пульсацію швидкості і натягу, що веде до нестационарної установки стрілки ручного вимірювального приладу. Крім того, в процесі вимірювань перепад середніх натягів нитки досягає великої величини, що вимагає перемикання коефіцієнта посилення ручного механічного тензометра. Враховуючи швидкість руху вузла, таке перемикання в динамічному режимі дуже складно здійснити.

Таким чином, для передбачуваних досліджень найбільш раціональною буде експериментальна установка на основі електротензометричної станції. Необхідно зазначити, що всі прилади проходили спеціальну тестову перевірку на предмет своєї функціональної придатності та надійності в роботі.

Розробка електротензометричної установки для проведення експериментальних досліджень

Загальний вигляд установки наведено на рис.7.3.2. На станині 5 жорстко укріплені перший 2 і другий 3 вимірювальні вузли. Між

вимірювальними вузлами 2 і 3 встановлено пристрій 4 для вимірювання фрикційних властивостей ниток з урахуванням анізотропії.

Нитка, сходячи з циліндричної шпулі 1, проходить через тарілчастий нитконатяжний прилад, який повідомляє їй попередній натяг (натяг веденої гілки).

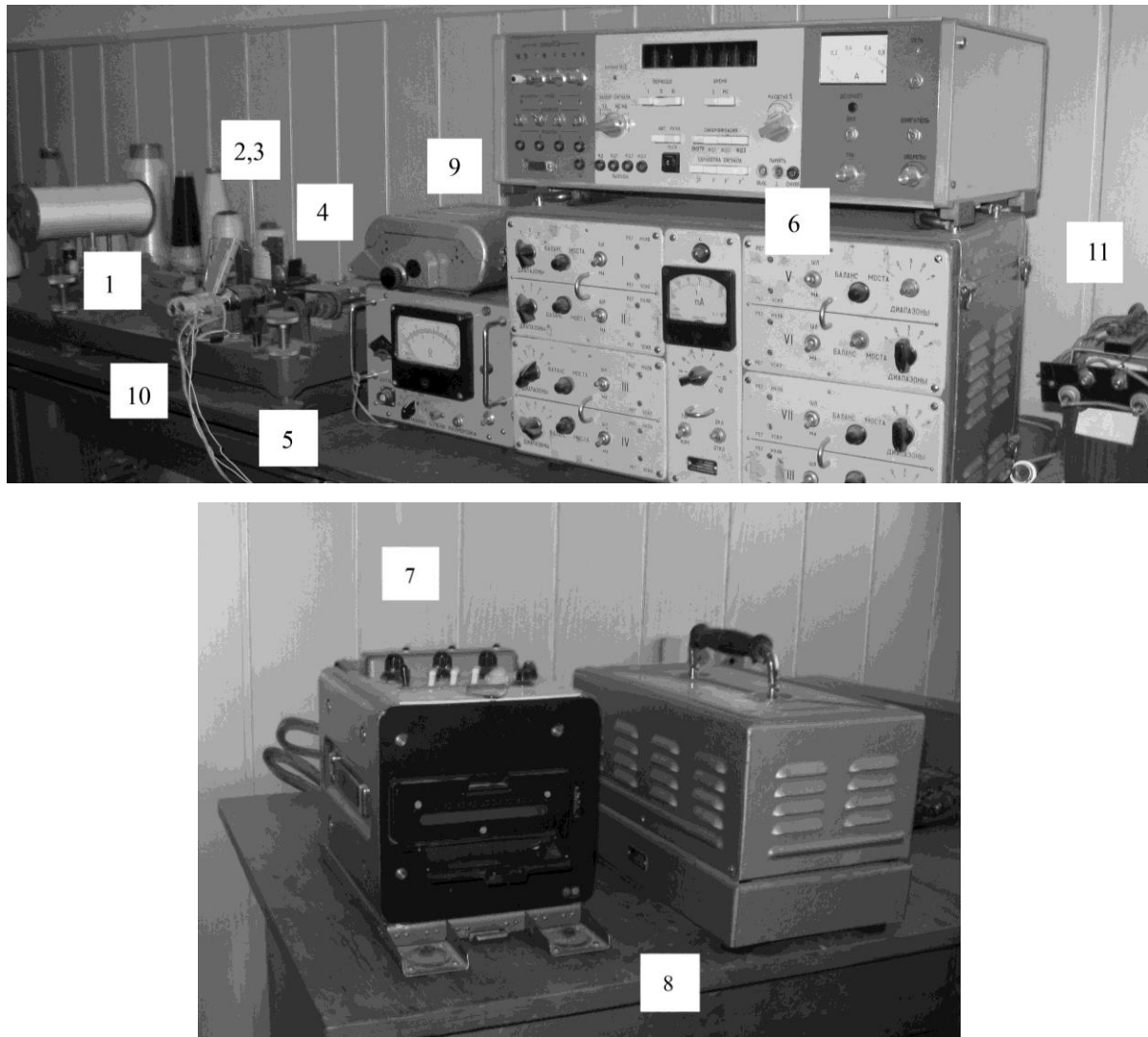


Рис.7.3.2. Загальний вигляд установки

На рис.7.3.3 представлена схема експериментальної установки. Послідовно проходячи через перший вимірювальний вузол 2, тарілчастий нитконатяжний прилад, пристрій 4 для вимірювання фрикційних

властивостей з урахуванням анізотропії нитка надходить у пристрій для змотування 10.

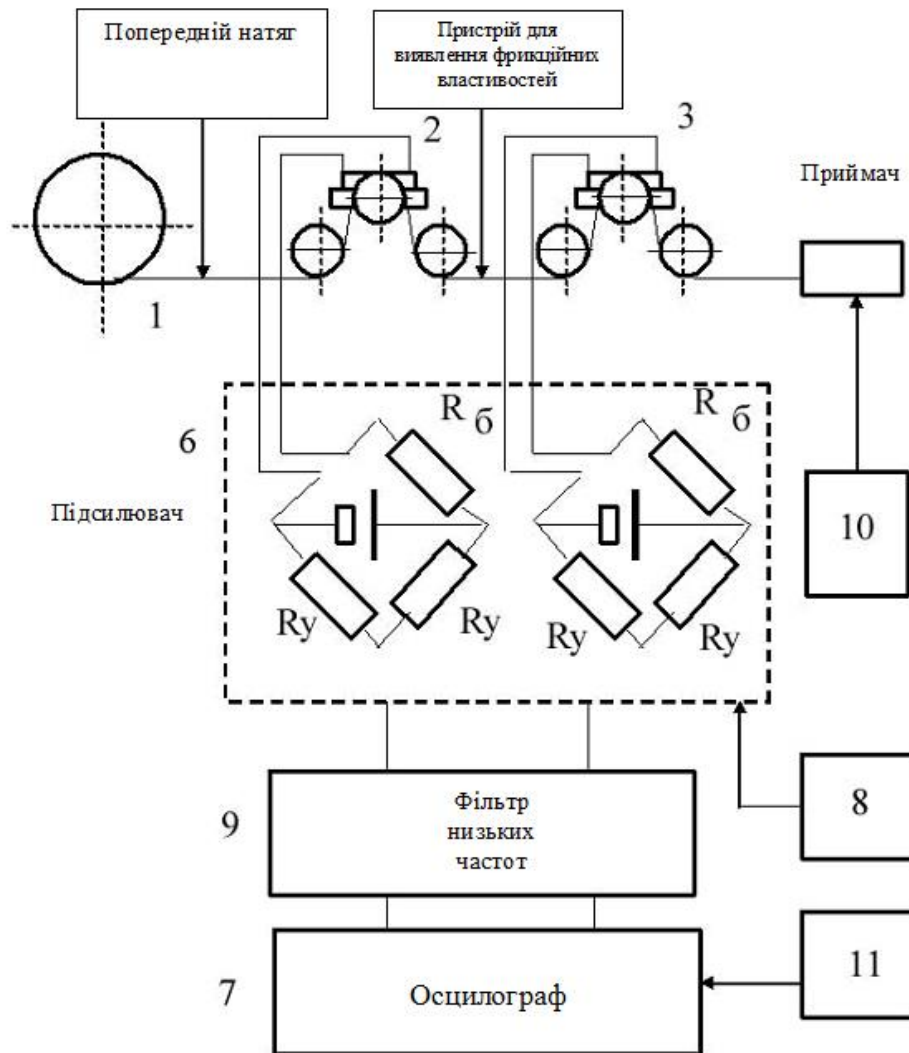


Рис.7.3.3. Схема експериментальної установки

Електричні сигнали з першого вимірювального вузла 2 і другого вимірювального вузла 3 надходять на 8-и каналний підсилювач 8АНЧ-7М (позиція 6). Далі через фільтр низьких частот 9 сигнали надходять у осцилограф 7 і далі на відповідні гальванометри осцилографа К12-22.

У таблиці 7.3.1 представлені характеристики гальванометрів осцилографа К12-22 (аналіз показав, що найбільш підходящими гальванометри для експерименту є гальванометри I і II типу).

Гальванометр є основним елементом осцилографа, функція якого полягає в реєстрації електричного сигналу, що надходить з підсилювача, на фотопапері.

Таблиця 7.3.1

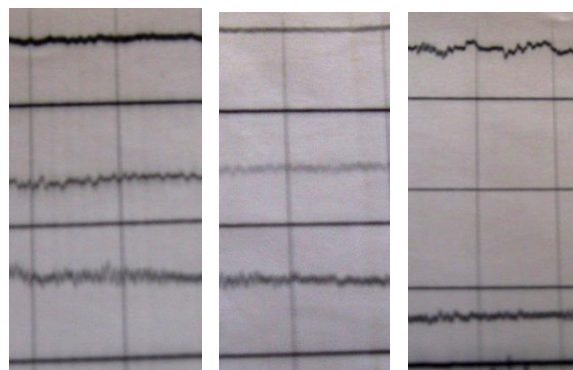
Характеристики гальванометрів осцилографа К12-22

Характеристика гальванометрів	Типи гальванометрів				
	I	II	III	IIIa	IV
Чутливість, мм/мА, не менше	0,21	0,57	2,37	9,5	32,2
Електричний опір, Ом	16±3	36±7	8±1,5	17±4,5	22±3,5
Максимально допустиме значення постійного струму, мА	30	30	15	3,5	2
Діапазон реєстрації змінних процесів, Гц	Від 0 до 500	Від 0 до 300	Від 0 до 150	Від 0 до 100	Від 0 до 30

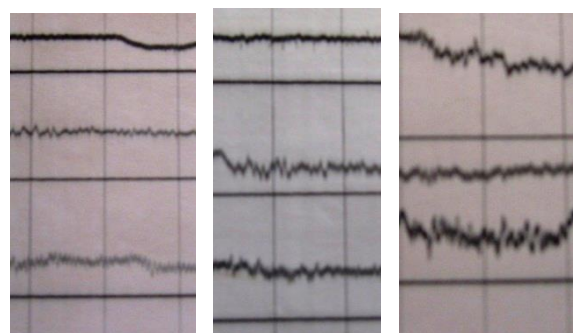
Для експерименту використовувалася осцилографний фотопапір МС-2-135, ТУ 6-17-1112-80 шириною 120 мм. Визначальними факторами при виборі гальванометрів осцилографа є частотна характеристика досліджуваного процесу і вихідний струм з відповідного каналу підсилювача на осцилограф.

В експериментальну установку також входили блоки живлення 8, 11 відповідно для підсилювача 8АНЧ-7М і осцилографа К12-22.

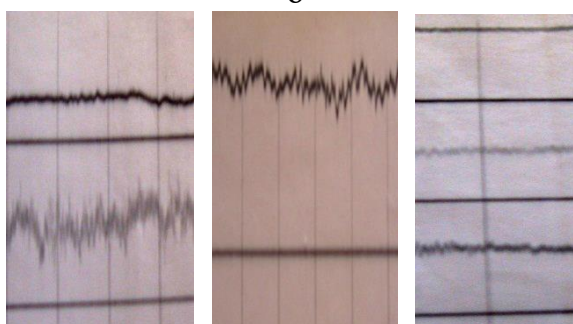
Зразки осцилограм наведені на рис.7.3.



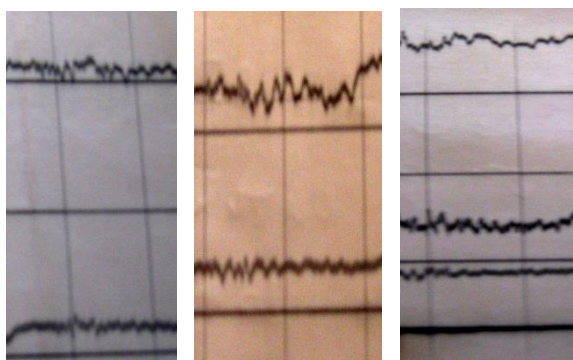
а



б



в



г

Рис.7.3.Зразки осцилограм

Осцилограми а (рис.7.3.4) отримані для віскозної комплексної нитки 16,7 Текс для швидкостей $V = 1, 2, 3$ м/с; осцилограми б (рис.7.3.4б)

отримані для капронової комплексної нитки 31,2 Текс для швидкостей $V = 1, 2, 3$ м/с; осцилограми в (рис.7.3.4в) отримані для бавовняної гребінній крученої пряжі 23,6 Текс для швидкостей $V = 1, 2, 3$ м/с; осцилограми γ (рис.7.3.4г) отримані для вовняної гребінної крученої пряжі 36 Текс для швидкостей $V = 1, 2, 3$ м/с.

Вимірювальні вузли 2 і 3 (див. Рис.7.3.2 ÷ 7.3.3) служать для вимірювання натягу веденої і ведучої гілок нитки до і після приладу 4 для визначення фрикційних властивостей з урахуванням анізотропії тертя. Їх конструкція однакова.

На рис.7.3.5 представлена схема вимірювального вузла, який включає в себе два ролика 1 і 2, які встановлені в підшипниках на нерухомих осях. Третій ролик 3 встановлений на консольно закріплену балочку таким чином, що внутрішнє кільце підшипника закріплено на ній, а з зовнішнім кільцем підшипника жорстко закріплений ролик взаємодіє з ниткою. Силами тертя в підшипниках нехтуємо.

Поперечні і поздовжні розміри балочки обрані такими, що частота власних коливань балочки дорівнює $f = 1400$ Гц. Ця частота у багато разів перевищує частоту найбільш високої складової натягу.

На балочку наклеєний тензодатчик з базою в 5 мм, які в парі з іншим тензодатчиком (наклеєним на пластину з того ж матеріалу) утворюють одну половину вимірювального моста. Друга половина розташовується на відповідному каналі в підсилювачі 8АНЧ-7М.

На схемі прийняті наступні позначення: r - радіус роликів, з якими взаємодіє нитка; l, h - відповідно відстань між осями нерухомого і рухомого роликів по горизонталі і вертикалі; P - натяг нитки; F - згинальна балочку сила; α - кут нахилу нитки до горизонталі у вимірювальному вузлі.

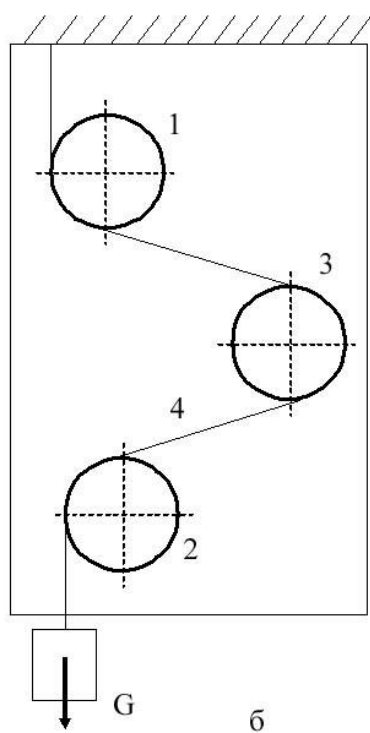
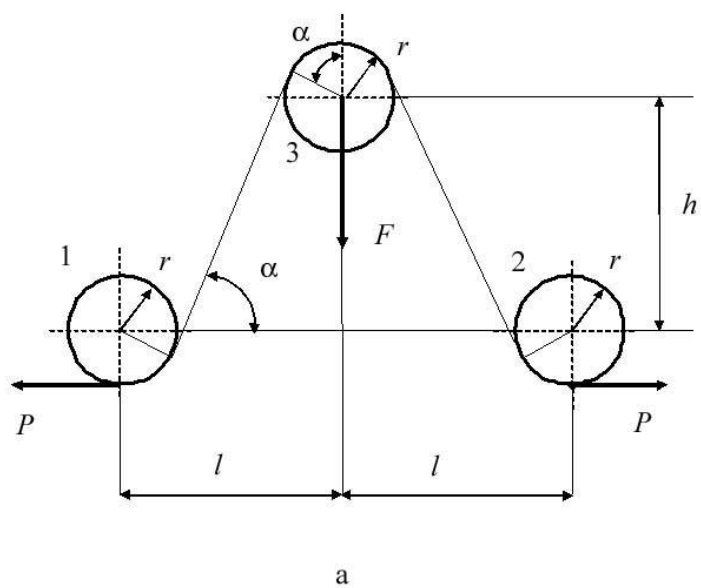


Рис.7.3.5. Схема вимірювального вузла 2 і 3 і тарировочна схема

Після отримання осцилограми зміни величини F від часу необхідно перейти безпосередньо до значення натягу P . Для цього необхідно отримати формулу для перерахунку. Як впливає зі схеми (рис.7.3.5) залежність між F і P має вигляд

$$F = 2P \sin \alpha . \quad (7.3.1)$$

Вираз, що пов'язує між собою геометричні параметри, має вигляд

$$2r \sin \alpha + \sqrt{l^2 + h^2 - 4r^2} \cos \alpha = l . \quad (7.3.2)$$

Рівняння (7.3.2) являє собою тригонометричне рівняння, яке вирішуємо методом додаткового кута. Запишемо (7.3.2) у вигляді

$$\frac{2r}{\sqrt{l^2 + h^2}} \sin \alpha + \frac{\sqrt{l^2 + h^2 - 4r^2}}{\sqrt{l^2 + h^2}} \cos \alpha = \frac{l}{\sqrt{l^2 + h^2}} , \quad (7.3.3)$$

і приймемо

$$\frac{2r}{\sqrt{l^2 + h^2}} = \cos \alpha' , \quad \frac{\sqrt{l^2 + h^2 - 4r^2}}{\sqrt{l^2 + h^2}} = \sin \alpha' , \quad (7.3.4)$$

де додатковий кут визначається як

$$\alpha' = \arcsin \frac{\sqrt{l^2 + h^2 - 4r^2}}{\sqrt{l^2 + h^2}} . \quad (7.3.5)$$

Тоді, з урахуванням (7.3.4), вираз (7.3.3) прийме вигляд

$$\sin \alpha \cos \alpha' + \cos \alpha \sin \alpha' = \frac{l}{\sqrt{l^2 + h^2}} , \quad (7.3.6)$$

або

$$\sin(\alpha + \alpha') = \frac{l}{\sqrt{l^2 + h^2}} .$$

Після перетворення отримаємо

$$\alpha = \arcsin \frac{l}{\sqrt{l^2 + h^2}} - \arcsin \frac{\sqrt{l^2 + h^2 - 4r^2}}{\sqrt{l^2 + h^2}} . \quad (7.3.7)$$

Використовуючи формулу синуса різниці двох кутів, отримаємо

$$\sin\alpha = \frac{l}{\sqrt{l^2 + h^2}} \sqrt{1 - \frac{l^2 + h^2 - 4r^2}{l^2 + h^2}} - \sqrt{1 - \frac{l^2}{l^2 + h^2}} \frac{\sqrt{l^2 + h^2 - 4r^2}}{\sqrt{l^2 + h^2}}.$$

Остаточно отримаємо

$$\sin\alpha = \frac{2rl - h\sqrt{l^2 + h^2 - 4r^2}}{l^2 + h^2}. \quad (7.3.8)$$

Підставляємо (7.3.8) в (7.3.1), отримуємо формулу для перерахунку величини натягу

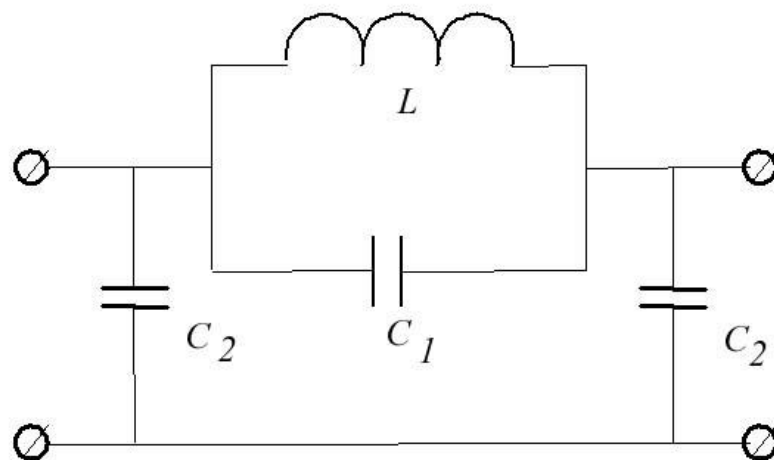
$$P = \frac{F}{2\left(\frac{2rl - h\sqrt{l^2 + h^2 - 4r^2}}{l^2 + h^2}\right)}. \quad (7.3.9)$$

Останній вираз (7.3.9) дозволяє визначати істинне значення натягу після розшифрування осцилограм.

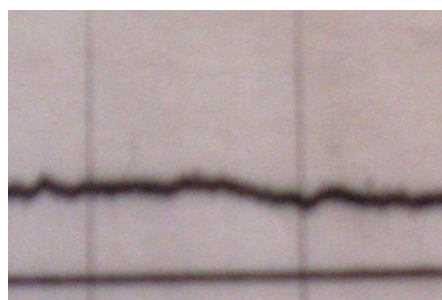
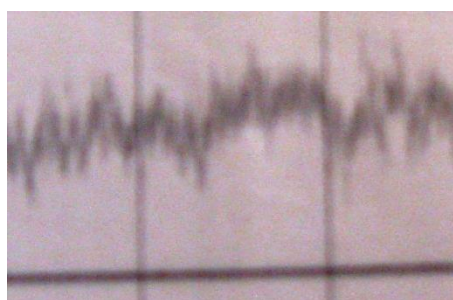
На рис.7.3.5б представлена тарировочна схема вимірювального вузла. Вимірювальний вузол жорстко закріплюється однією зі сторін. Нитка 4 заправляється між нерухомими 1 і 2 і рухомими 3 роликками. До кінця нитки 4 послідовно прикріплюються вантажі відомої ваги. Показання величини сили фіксуються на фотопапері. За отриманими точками будується тарировочна залежність. Проведений аналіз дозволив встановити, що тарировочні залежності мають лінійний характер.

На рис.7.3.6 представлена схема електричного фільтру низьких частот, який встановлюється перед осцилографом і використовується в процесі запису коливань натягу нитки у вузькому діапазоні частот від 0 ÷ 50 Гц. Даний діапазон частот близький до робочого для розглянутих типів нитконатягувачів. Елементи фільтра розраховані за відомими формулами і склали $L = 38,4$; $C_1 = 171$ мкф; $C_2 = 192$ мкф. На рис.7.3.6 б,в представлені осцилограми вимірювання натягу нитки без фільтру низьких частот (б) і з застосуванням фільтру низьких частот (в).

На рис.7.3.6г представлена амплітудно-частотна характеристика зазначеного фільтру.

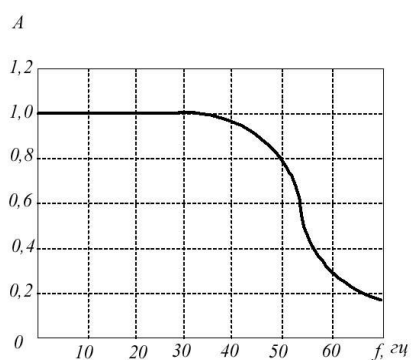


а



б

в



г

Рис.7.3.6. Фільтр низьких частот і його амплітудно-частотна характеристика

Пристрій для змотування 10 включає в себе нерухомо укріплений електродвигун, на валу якого укріплений шків. Зв'язок із ступінчастим шківом на відомому валу здійснюється за допомогою пасової передачі.

Електродвигун підключається до мережі через трансформатор Tr і випрямляч, зібраний на діодах Д7Ж (рис.7.3.7).

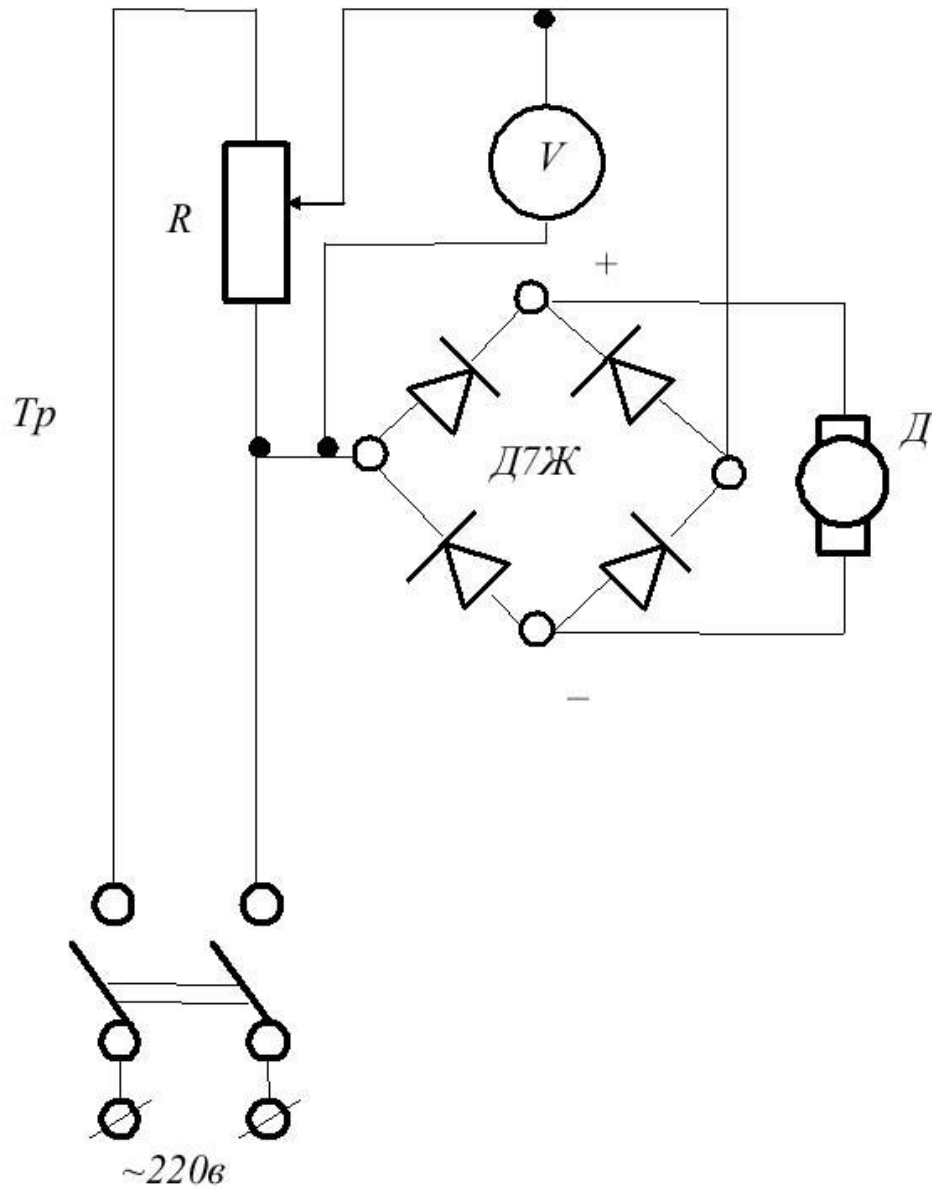


Рис.7.3.7. Електрична схема управління кутовою швидкістю обертання двигуна

Така схема дозволяє плавно змінювати швидкість обертання шківів і встановлювати лінійну швидкість руху нитки в діапазоні від 1 до 4 м/с.

Пристрій для визначення фрикційних властивостей ниток в подовжньому і поперечному напрямку з урахуванням анізотропії тертя

Для визначення фрикційних властивостей ниток в подовжньому і поперечному напрямку з урахуванням анізотропії тертя було розроблено спеціальний пристрій. Винахід відноситься до галузі дослідження матеріалів механічними способами, зокрема для визначення фрикційних властивостей гнучких ниток.

Відомий прилад для визначення коефіцієнта тертя ниток в подовжньому напрямку [8]. Прилад містить шків, призначений для розміщення на ньому зразка матеріалу, фрикційні властивості якого досліджуються, вантаж і динамометр, які з'єднані гнучкою стрічкою, яка має високу жорсткість в подовжньому напрямку, на нижній поверхні якої, з боку шківа, прикріплюється зразок матеріалу, що досліджується.

Але використання цього приладу не дозволяє визначити величину коефіцієнта тертя в поперечному напрямку (перпендикулярно волокнам), що знижує діапазон вимірювання величин і потребує додаткових витрат на визначення параметрів, які визначають анізотропію сил тертя.

Найбільш близьким по суті і ефекту, який досягається, до заявленого технічного рішення є прилад для визначення коефіцієнта тертя ниток [5].

Прилад містить шків, який встановлений з можливістю горизонтального переміщення щодо осі обертання, причому одна його торцева сторона має нахил і впирається в ролик, а інша контактує з пружиною. Гнучка стрічка з жорсткістю в подовжньому і поперечному напрямках, яка з'єднана з вантажем з одного боку і робочим рухомим органом динамометра з іншого боку, має на нижній поверхні з боку шківа прикріплений зразок нитки, фрикційні властивості якої досліджуються, і

розташовується між трьома парами роликів, які закріплені на двох кронштейнах. Кронштейни роликів з'єднані з робочими рухливими органами динамометрів. В якості рухомих органів динамометрів для визначення зусиль в поздовжньому напрямку використовуються електротензометричні пристрої, описані в підрозділі 7.3.1.

Але розвиток сучасного машинобудування та технологій в легкій і текстильній промисловості призводить до різкого збільшення швидкості руху нитки. Як відомо зі збільшенням швидкості руху нитки змінюється і величина коефіцієнта тертя [4]. Це потребує припасування при розрахунку натягу ведучої гілки нитки в робочій зоні технологічного обладнання, коли нитка взаємодіє з тарілчастими нитконатяжними пристроями [5]. Крім того, наявність вищої кінематичної пари (кулачкової), яку утворюють ролик і шків з похилою торцевою стороною і пружини для силового замикання, не дозволяє використовувати його при високих швидкостях обертання шківа. Це пояснюється наявністю розривів безперервності в графіках прискорення і похідних від прискорення будь-яких порядків, які мають місце в кулачкових механізмах [5-6].

Це призводить до того, що в процесі роботи виникають удари, відривання шківа від ролика. Це служить причиною нестаціонарного руху шківа в поперечному напрямку щодо осі обертання і не дозволяє визначити коефіцієнт тертя в поперечному напрямку при великих швидкостях руху нитки.

Завданням запропонованого винаходу є розширення технологічних можливостей за рахунок збільшення діапазону вимірювання фрикційних властивостей ниток при великих швидкостях руху.

Поставлена мета досягається тим, що пристрій для визначення фрикційних властивостей нитки, який містить катушкотримач, підпружинений ролик, три вимірювальних вузла з тензодатчиками, два рухливих ролика закріплених на корпусі і нерухому циліндричну

направляючу, яка закріплена на кронштейнах жорстко закріплених на корпусі, приймальний барабан для нитки і ступінчастий привід, згідно запропонованої моделі містить горизонтальну циліндричну направляючу на поверхні якої, за твірними, щільно розташовані нитки, кінці циліндричної направляючої квадратної форми в перетині з пристроєм фіксації розташовані на вертикальних кронштейнах в двох горизонтальних пазах, висота яких дорівнює довжині сторони квадрата кінця циліндричної направляючої, покажчик і кутомір [10].

Порівняльний аналіз з прототипом показує, що заявлений пристрій для визначення фрикційних властивостей нитки відрізняється тим, що містить горизонтальну циліндричну направляючу на поверхні якої, за твірними, щільно розташовані нитки, кінці циліндричної направляючої квадратної форми в перетині з пристроєм фіксації розташовані на вертикальних кронштейнах в двох горизонтальних пазах, висота яких дорівнює довжині сторони квадрата кінця циліндричної направляючої, покажчик і кутомір. Ознаки, які були перераховані вище, дозволяють вимірювати фрикційні властивості ниток з урахуванням анізотропії в разі ковзання по поверхні інших ниток, які виконують функцію направляючих, яка розширює технологічні можливості за рахунок збільшення асортименту матеріалів направляючих поверхонь з якими взаємодіє нитка в процесі своєї переробки.

На рис.7.3.8 і 7.3.9 зображена схема приладу.

Пристрій для визначення фрикційних властивостей нитки містить котушкотримач, підпружинений ролик 1, три вимірювальних вузла 2-3 з тензодатчиками, два рухливих ролика 4, закріплених на корпусі 5 і нерухому циліндричну направляючу 6, яка закріплена на кронштейнах 11 жорстко закріплених на корпусі 5, приймальний барабан 9 для нитки і ступінчастий привід 8, горизонтальну циліндричну направляючу на поверхні якої, за твірними, щільно розташовані нитки, кінці циліндричної

направляючої квадратної форми в перетині з пристроєм фіксації розташовані на вертикальних кронштейнах в двох горизонтальних пазах, висота яких дорівнює довжині сторони квадрата кінця циліндричної направляючої, показчик і кутомір.

Пристрій працює наступним чином. На першому етапі щільно розміщуємо нитку 12, яка виконує функцію направляючої, по утворюючим циліндрової направляючої 6. Встановлюємо циліндричну направляючу 6 в горизонтальні пази 10 кронштейнів 11, таким чином, щоб квадратні кінці циліндричної направляючої 6 з показчиком 14 розташувалися в горизонтальній площині, а показчик 14 був суміщений з 0-ою відміткою показника кутoměра 15. Нитка з котушки проходить тарілчастий нитконатягувачі, підпружинений ролик 1, три вимірювальних вузла 2-3 з тензодатчиками, два рухливих ролика 4 закріплених на корпусі 5 і циліндричну направляючу 6, яка розташована на кронштейнах 11 жорстко закріплених на корпусі 5, приймальний барабан 9 для нитки. Включає ступінчастий привід 8, наводимо досліджувану нитку в рух. Значення натягу фіксуються вимірювальними вузлами 2-3. Після цього, визначають фрикційні характеристики. Для оцінки анізотропії фрикційних властивостей відключають пристрої фіксації і повертають циліндричну направляючу 6 в горизонтальних пазах 10 кронштейнів 11 і включають пристрої фіксації. Кут обороту контролюють показником кутoměра 15 і показчиком 17.3. Після чого виконують попередні операції.

Використання запропонованого пристрою дозволяє розширити технологічні можливості за рахунок збільшення асортименту матеріалів направляючих поверхонь, з якими взаємодіє нитка в процесі своєї переробки з урахуванням їх анізотропії, що дозволяє удосконалити технологічні процеси та обладнання легкої та текстильної промисловості.

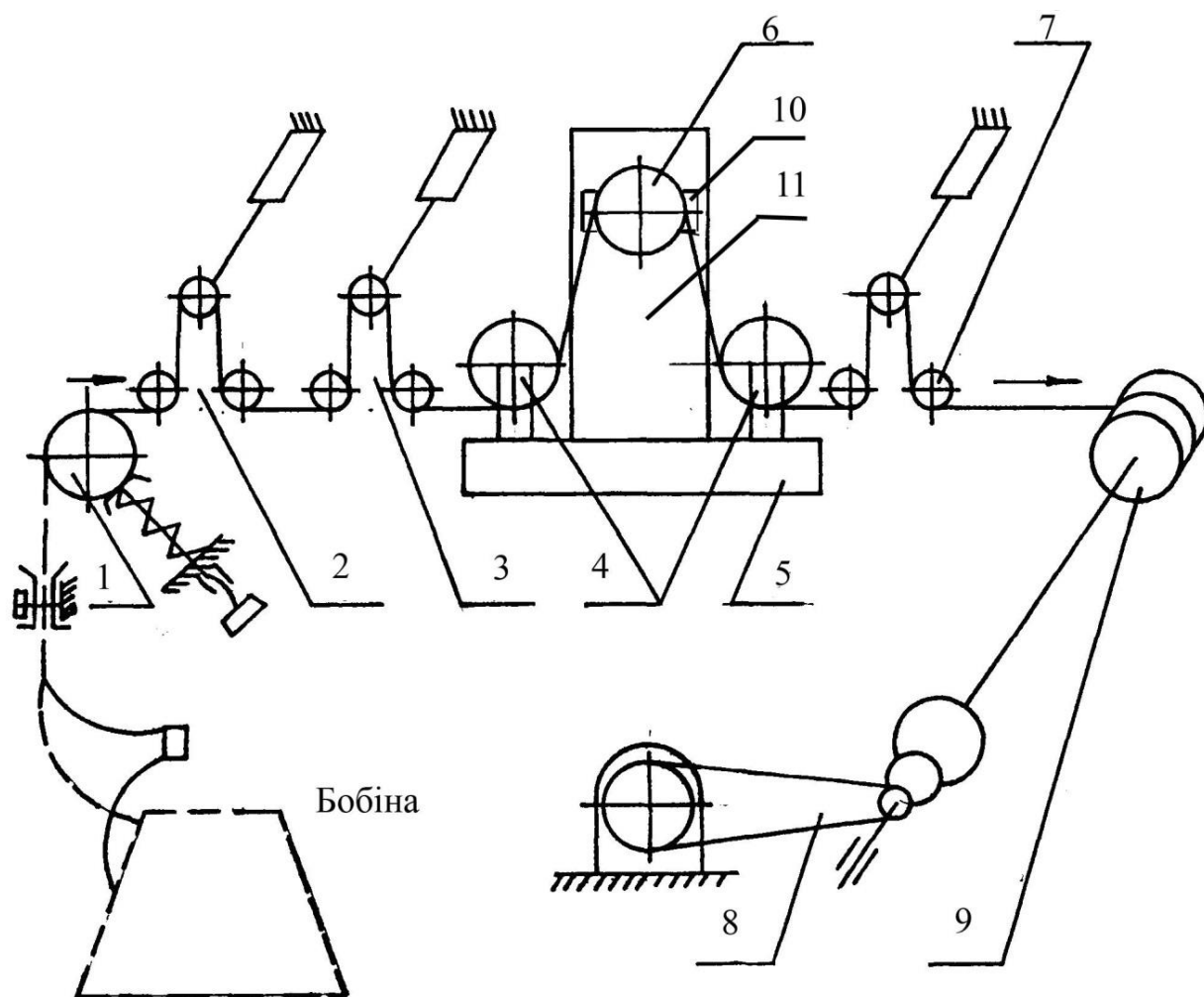


Рис.7.3.8. Схема приладу для визначення коефіцієнта тертя ниток

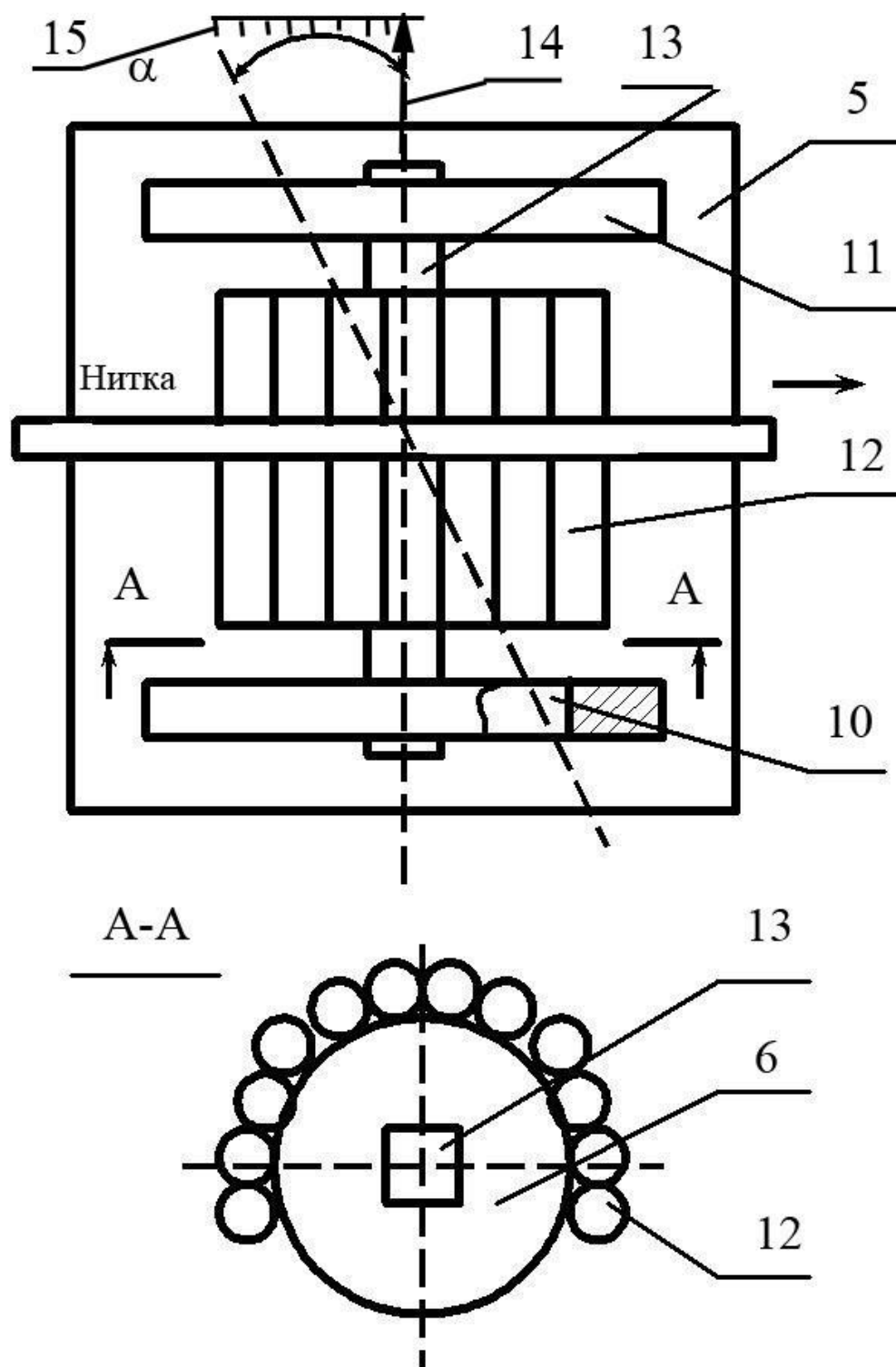


Рис.7.3.9. Схема вузла для визначення фрикційних властивостей нитки з урахуванням анізотропії (пристрої фіксації умовно не показані)

Плани експериментальних досліджень

При проведенні експериментальних досліджень, нами було поставлено завдання щодо визначення впливу радіуса кривизни направляючої поверхні R , натягу веденої гілки P_0 , кута охоплення φ на натяг величини коефіцієнта тертя ниток в подовжньому і поперечному напрямку в залежності від матеріалу ниток.

Для дослідження впливу матеріалу нитки на величину коефіцієнтів тертя в подовжньому і поперечному напрямках були обрані віскозна комплексна нитка 16,7 Текс, капронова комплексна нитка 31,2 Текс, бавовняна гребінна кручена пряжа 23,6 Текс, вовняна гребінна кручена пряжа 36 Текс.

Вибір даного асортименту, в першу чергу, пояснюється приблизно однаковим тексом ниток. Крім того, цей вид сировини широко використовується для виготовлення текстильних і трикотажних виробів [10, 11].

При вивченні анізотропії тертя текстильних ниток, нами було проведено дві серії дослідів (як змінюється величина коефіцієнта тертя при русі нитки в подовжньому і поперечному напрямках).

Відбір проб і лабораторні випробування за визначенням фізико-механічних властивостей (питомої розривного навантаження, відносного розривного подовження) ниток РМ-30-1. Швидкість розтягування нитки, відповідно до вимог, для одиночної і крученої пряжі з усіх видів волокон дорівнює 10 ± 1 мм/с, для всіх хімічних ниток 20 ± 2 мм/с. Відстань між зажимами розривної машини була однаковою 500 ± 1 мм.

Перед випробуваннями з пакування відмотуємо 5 м нитки, між випробуваннями - 1м. Нитка заправлялася в затискачі з попереднім натягом. Питоме розривне навантаження і відносне розривне подовження

брали як середній показник 10 розривів нитки. Середнє арифметичне має укладатися в необхідні допуски.

Після визначення чинників, що впливають на збільшення натягу ведучої гілки нитки, обґрунтування і вибору сировини для випробувань, необхідно скласти план проведення експерименту.

Для кожної з ниток даним планом (таблиця 7.3.2) передбачається 15 дослідів. По кожному з дослідів передбачалося 3 повторних.

Вибір розмірів направляючої поверхні при проведенні експерименту, в пропонованому діапазоні від 3 мм до 7 мм, пояснюється тим, що цей інтервал широко охоплює асортимент ниткопрямувачів, робочих органів текстильних і трикотажних машин.

Інтервал зміни величини натягу веденої гілки нитки від 5 до 15 сН був вибраний виходячи з умов переробки ниток на технологічному устаткуванні.

Кут обхвату направляючої поверхні, для кожного конкретного випадку, є величиною змінною. Зважаючи на специфіку переробки ниток на текстильному устаткуванні, де мають місце великі перегини ниток при взаємодії з робочими органами, нами був вибраний інтервал зміни від 90° до 180° .

У таблиці 7.3.2 наведений план проведення експерименту по визначенню впливу радіусу кривизни направляючої поверхні R , натягу веденої гілки P_0 , кута обхвату φ на величину коефіцієнта тертя в подовжньому та поперечному напрямку ниток чотирьох видів при діапазоні зміни радіусу кривизни напрямної від 3 до 7 мм.

При плануванні експерименту, при побудові моделей другого порядку нами було використано ортогональне планування другого порядку. Хоча існують інші оптимальні плани, ортогональне планування використовувалося із-за достатньої простоти і можливості застосування вже розроблених пакетів програм на ЕОМ при обробки отриманих

експериментальних даних. Крім цього, при використанні ортогональних планів другого порядку, порівняно з рототабельним плануванням другого порядку, в центрі матриці експерименту проводиться менше дослідів.

У нульовій точці величина радіуса кривизни становила 5 мм. Натяг веденої гілки нитки становив 10 сН. Величина кута обхвату ниткою направляючої дорівнювала 135° .

Таблиця 7.3.2.

План проведення експерименту

Визначення коефіцієнта тертя в поздовжньому (поперечному) напрямках, kt							
№ варіанту	Досліджувані фактори						Характерні зони матриці
	x1	R	x2	Po	x3	φ	
i- 1	-1	R+h1	+1	Po+ h2	+1	$\varphi+ h3$	Точки ядра
i- 2	+1	R-h1	-1	Po+ h2	+1	$\varphi+ h3$	
i- 3	-1	R+h1	+1	Po- h2	-1	$\varphi+ h3$	
i- 4	+1	R-h1	+1	Po- h2	+1	$\varphi+ h3$	
i- 5	+1	R+h1	+1	Po+ h2	-1	$\varphi- h3$	
i- 6	+1	R-h1	-1	Po+ h2	-1	$\varphi- h3$	
i- 7	-1	R+h1	-1	Po- h2	-1	$\varphi- h3$	
i- 8	-1	R-h1	-1	Po- h2	+1	$\varphi- h3$	
i- 9	-1,215	R-*	0	Po	0	φ	Зіркові точки
i- 10	+1,215	R+*	0	Po	0	φ	
i- 11	0	R	-1,215	Po-*	0	φ	
i- 12	0	R	+1,215	Po+*	0	φ	
i- 13	0	R	0	Po	-1,215	$\varphi-*$	
i- 14	0	R	0	Po	+1,215	$\varphi+*$	
i- 15	0	R	0	Po	0	φ	Нульова точка

Сумісний вплив радіусу кривизни направляючої, натягу веденої гілки, кута обхвату на величину коефіцієнту тертя в подовжньому напрямку руху нитки

Реалізуюючи план експериментальних досліджень (табл.7.3.2) за визначенням впливу радіусу кривизни направляючої поверхні, натягу веденої гілки нитки, кута обхвату на величину коефіцієнта тертя в подовжньому напрямку руху нитки, нами були отримані ряд даних, представлених в таблиці 7.3.3.

Вище ми зупинялися на тих міркуваннях, які привели до вибору ортогонального плану. Останні вимоги застосування того або іншого плану, що пред'являються до оцінки ефективності, привели до того, що основними критеріями необхідно вважати: максимальну величину дисперсії передбачених значень критерію оптимізації в області експерименту; об'єм еліпсоїда розсіяння оцінок параметрів та ін.

При проведенні експерименту іменовані величини необхідно замінити на кодовані. В нашому випадку радіус кривизни напрямної R позначимо через $x1$. В нульовій точці радіус кривизни дорівнював 3 мм. Довжина плеча складала 2 мм. Натяг веденої гілки нитки P_0 позначимо через $x2$. В нульовій точці натяг веденої гілки нитки дорівнює 10 сН. Довжина плеча складала 5 сН. Кут охоплення ниткою напрямної φ позначимо через $x3$. В нульовій точці кут охоплення ниткою напрямної дорівнює $3\pi/4$ рад. Довжина плеча складала $\pi/4$ рад.

Тоді система рівнянь для переходу від іменованих до кодованих значень чинників буде мати вигляд

$$x1 = \frac{(R-3)}{2}, \quad x2 = \frac{P_0-10}{5}, \quad x3 = \frac{\varphi - \frac{3\pi}{4}}{\frac{\pi}{4}}. \quad (7.3.10)$$

Таблиця 7.3.3

Сумісний вплив радіусу кривизни направляючої, натягу веденої гілки, кута обхвату на величину коефіцієнту тертя в подовжньому напрямку руху
НИТКИ

Поздовжнє ковзання							
Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt
1- 1	0,146	2-1	0,131	3- 1	0,115	4- 1	0,151
1- 2	0,184	2- 2	0,164	3- 2	0,137	4- 2	0,169
1- 3	0,158	2- 3	0,144	3- 3	0,122	4- 3	0,151
1- 4	0,161	2- 4	0,145	3- 4	0,124	4- 4	0,158
1- 5	0,168	2- 5	0,151	3- 5	0,131	4- 5	0,167
1- 6	0,190	2- 6	0,170	3- 6	0,150	4- 6	0,181
1- 7	0,171	2- 7	0,154	3- 7	0,133	4- 7	0,172
1- 8	0,166	2- 8	0,149	3- 8	0,127	4- 8	0,166
1- 9	0,153	2- 9	0,137	3- 9	0,119	4- 9	0,154
1- 10	0,170	2- 10	0,152	3- 10	0,135	4- 10	0,168
1- 11	0,179	2- 11	0,158	3- 11	0,141	4- 11	0,175
1- 12	0,158	2- 12	0,139	3- 12	0,120	4- 12	0,154
1- 13	0,170	2- 13	0,152	3- 13	0,132	4- 13	0,170
1- 14	0,163	2- 14	0,147	3- 14	0,125	4- 14	0,162
1- 15	0,163	2- 15	0,148	3- 15	0,127	4- 15	0,162

На рис.7.3.10 представлена поверхня відгуку для капронової нитки.

Рівняння регресії для капронової нитки буде мати вигляд

$$kt = 0,163 + 0,007x_1 - 0,009x_2 - 0,0035x_3 - 0,0007x_1^2 - 0,0015x_1x_2 + 0,0005x_1x_3 + 0,0035x_2^2 - 0,001x_2x_3 + 0,0023x_3^2. \quad (7.3.11)$$

$x_3=0,0$

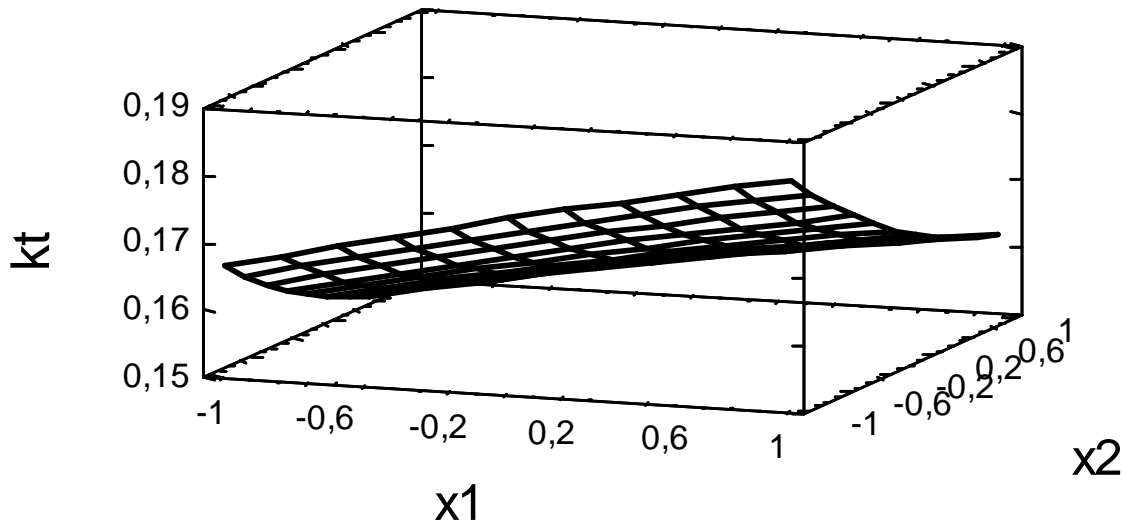


Рис.7.3.10. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.11) отримаємо відповідну діаграму, яка представлена на рис.7.3.11. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1^2 .

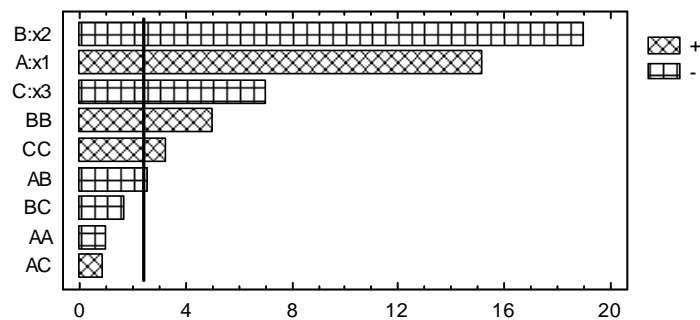


Рис.7.3.11. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.11), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,207 - 0,0007P_0 + 0,011R - 0,047\phi - 0,0000P_0^2 + 0,008\phi^2 - 0,0003P_0R. \quad (7.3.12)$$

На рис.7.3.12 представлена поверхня відгуку для віскозної комплексної нитки. Рівняння регресії для віскозної комплексної нитки буде мати вигляд

$$kt = 0,146 + 0,006x_1 - 0,0079x_2 - 0,0032x_3 - 0,0005x_1^2 - 0,0013x_1x_2 + 0,00075x_1x_3 + 0,0019x_2^2 - 0,001x_2x_3 + 0,0026x_3^2. \quad (7.3.13)$$

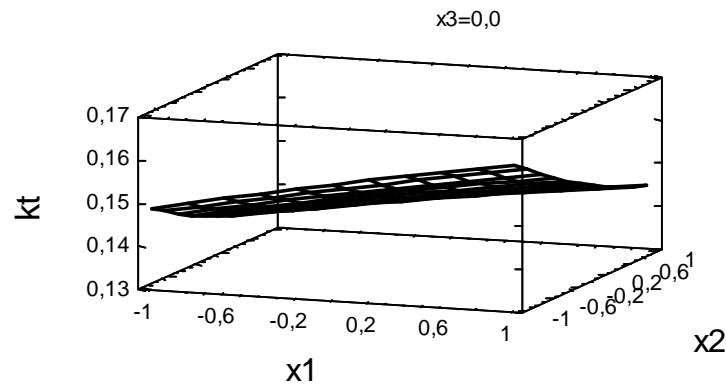


Рис.7.3.12. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.13) отримаємо відповідну діаграму, яка представлена на рис.7.3.13. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_2^2 , x_1^2 .

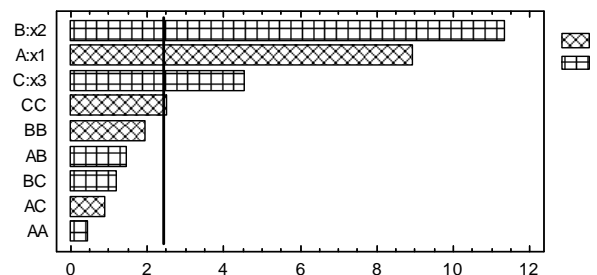


Рис.7.3.13. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.13), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,209 - 0,0032P_0 + 0,0065R - 0,046\phi + 0,008\phi^2. \quad (7.3.14)$$

На рис.7.3.14 представлена поверхня відгуку для вовняної пряжі. Рівняння регресії для вовняної пряжі буде мати вигляд

$$kt = 0,127 + 0,0058x_1 - 0,0073x_2 - 0,0037x_3 - 0,00006x_1^2 - 0,0011x_1x_2 - 0,00088x_1x_3 + 0,0021x_2^2 + 0,0006x_2x_3 + 0,00084x_3^2. \quad (7.3.15)$$

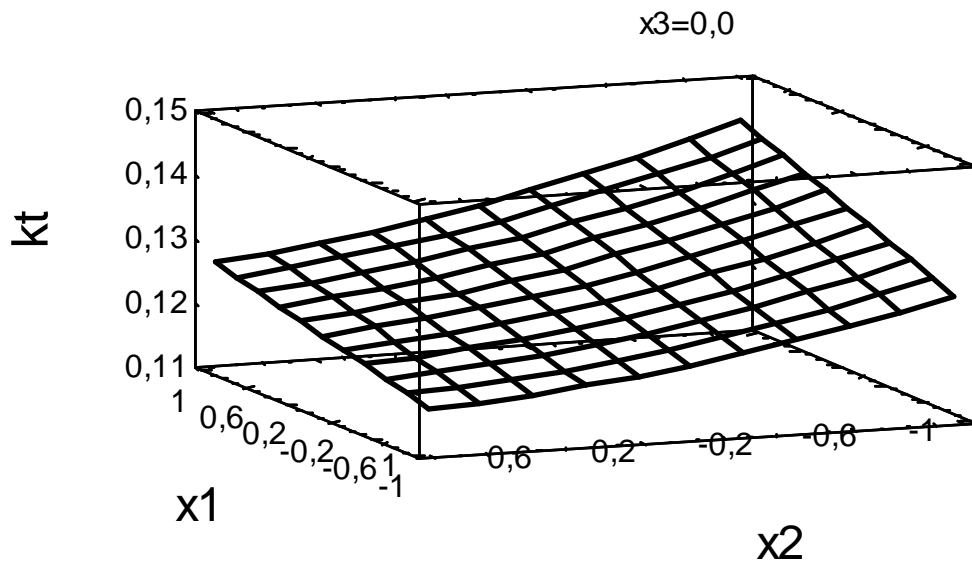


Рис.7.3.17.3. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.15) отримаємо відповідну діаграму, яка представлена на рис.7.3.15. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_3^2 , x_1^2 .

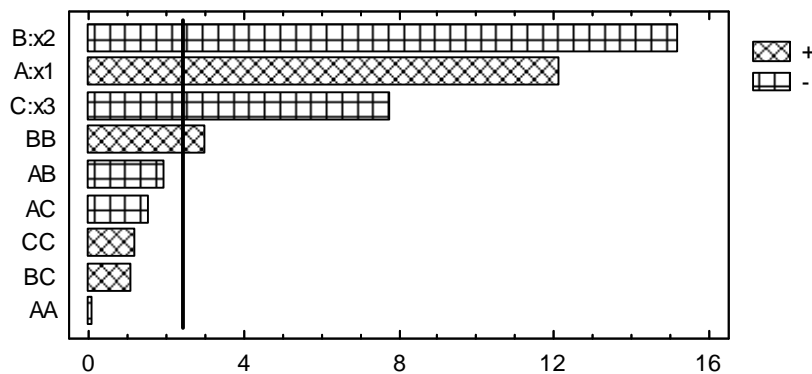


Рис.7.3.15. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.15), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,164 - 0,0062P_0 + 0,006R - 0,0089\phi + 0,0002P_0^2. \quad (7.3.16)$$

На рис.7.3.16 представлена поверхня відгуку для бавовняної пряжі. Рівняння регресії для бавовняної пряжі буде мати вигляд

$$kt = 0,162 + 0,0049x_1 - 0,0078x_2 - 0,0033x_3 - 0,001x_1^2 + 0,0014x_1x_2 - 0,0019x_1x_3 + 0,0011x_2^2 + 0,0011x_2x_3 + 0,002x_3^2. \quad (7.3.17)$$

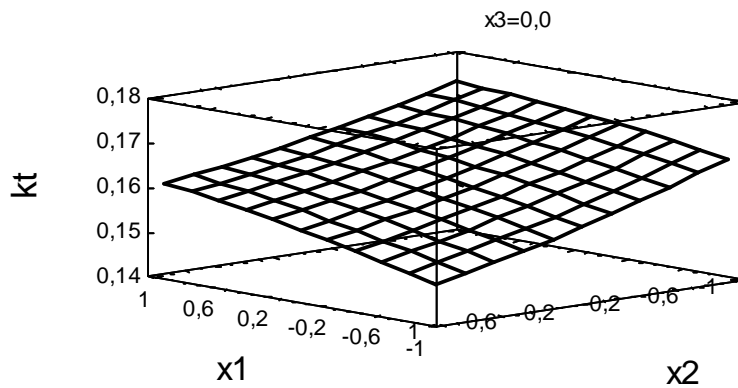


Рис.7.3.16. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.17) отримаємо відповідну діаграму, яка представлена на рис.7.3.17. Аналіз даної діаграми показує, що не значущих коефіцієнтів немає.

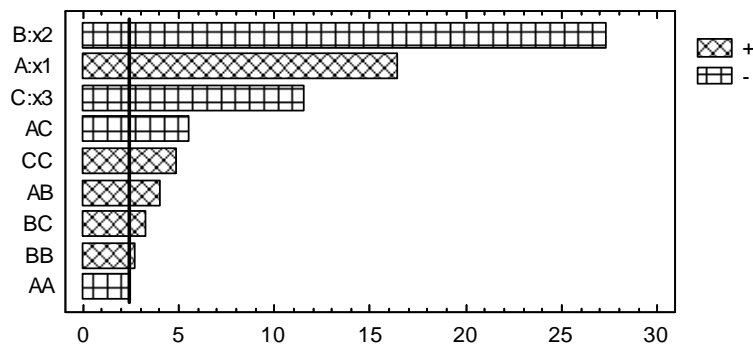


Рис.7.3.17. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.17), а

також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,28 - 0,0075P_0 + 0,00075R - 0,057\phi + 0,00008P_0^2 - 0,0005R^2 + 0,0064\phi^2 + 0,000275P_0R + 0,00057\phi + 0,0025R\phi. \quad (7.3.18)$$

Розрахунки по формулам (7.3.12), (7.3.14), (7.3.16), (7.3.18) дозволили отримати масив даних (дивись додаток А), використовуючи який були побудовані графічні залежності представлені на рис.7.3.18-7.3.20.

Аналіз графічних залежностей на рис.7.3.18-7.3.20 показав, що при збільшенні радіусу кривизни напрямної коефіцієнт тертя в подовжньому напрямку руху нитки зростає. Для бавовняної пряді після 5 мм починається його зменшення, що може бути пов'язано зі зміною структури нитки. При збільшенні натягу веденої гілки (рис.7.3.19) та кута обхвату (рис.7.3.20) простежується зворотня картина.

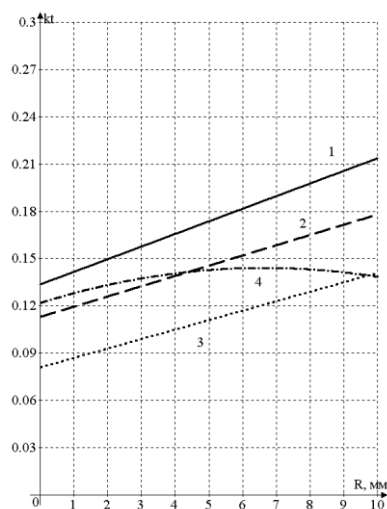


Рис.7.3.18. Залежність коефіцієнту тертя в подовжньому напрямку руху нитки від радіусу кривизни

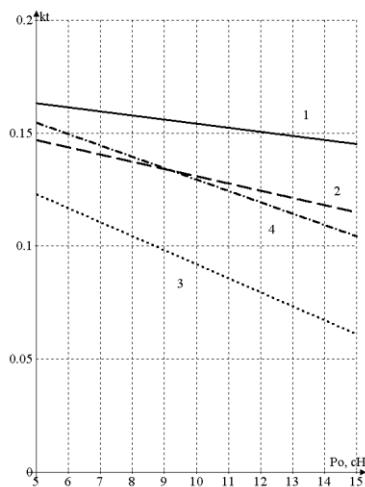


Рис.7.3.19. Залежність коефіцієнту тертя в подовжньому напрямку руху нитки від натягу веденої гілки нитки

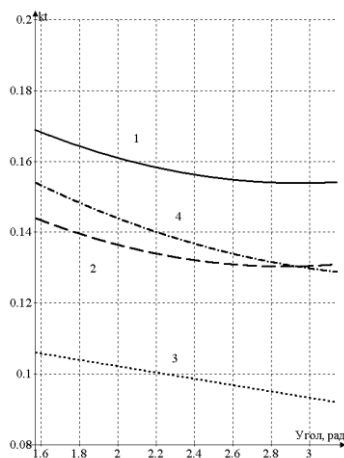


Рис.7.3.20. Залежність коефіцієнту тертя в подовжньому напрямку руху нитки від кута обхвату

Коефіцієнту тертя в подовжньому напрямку руху нитки при зростанні цих величин починає падати. Це пояснюється тим, що окремі мікрволокна в структурі нитки чи пряжі приймають форму близьку до прямої форми осі[2, 4]. Ці результати необхідно враховувати при проектуванні технологічних процесів легкої та текстильної промисловості.

Сумісний вплив радіусу кривизни направляючої, натягу веденої гілки, кута обхвату на величину коефіцієнту тертя в поперечному напрямі руху нитки

Реалізуюючи план експериментальних досліджень (табл.7.3.2) за визначенням впливу радіусу кривизни направляючої поверхні, натягу веденої гілки нитки, кута обхвату на величину коефіцієнту тертя в поперечному напрямі руху нитки, нами були отримані ряд даних, представлених в таблиці 7.3.7.3.

При проведенні експерименту іменовані величини необхідно замінити

на кодовані. В нашому випадку радіус кривизни напрямної R позначимо через $x1$. В нульовій точці радіус кривизни дорівнював 5 мм. Довжина плеча складала 2 мм. Натяг веденої гілки нитки P_0 позначимо через $x2$. В нульовій точці натяг веденої гілки нитки дорівнює 10 сН. Довжина плеча складала 5 сН. Кут охоплення ниткою напрямної φ позначимо через $x3$. В нульовій точці кут охоплення ниткою напрямної дорівнює $3\pi/4$ рад. Довжина плеча складала $\pi/4$ рад. Система рівнянь для переходу від іменованих до кодованих значень чинників буде мати вигляд (7.3.10).

Таблиця 7.3.4

Сумісний вплив радіусу кривизни направляючої, натягу веденої гілки, кута обхвату на величину коефіцієнту тертя в поперечному напрямі руху нитки

Поздовжнє ковзання							
Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt	Варіант	Коефіцієнт тертя, kt
1- 1	0,281	2-1	0,219	3- 1	0,210	4- 1	0,247
1- 2	0,265	2- 2	0,225	3- 2	0,187	4- 2	0,241
1- 3	0,275	2- 3	0,215	3- 3	0,203	4- 3	0,241
1- 4	0,305	2- 4	0,235	3- 4	0,228	4- 4	0,254
1- 5	0,295	2- 5	0,221	3- 5	0,218	4- 5	0,249
1- 6	0,248	2- 6	0,222	3- 6	0,179	4- 6	0,235
1- 7	0,234	2- 7	0,205	3- 7	0,165	4- 7	0,228
1- 8	0,249	2- 8	0,208	3- 8	0,176	4- 8	0,233
1- 9	0,261	2- 9	0,202	3- 9	0,190	4- 9	0,219
1- 10	0,290	2- 10	0,216	3- 10	0,211	4- 10	0,235
1- 11	0,259	2- 11	0,203	3- 11	0,185	4- 11	0,217

1- 12	0,294	2- 12	0,218	3- 12	0,209	4- 12	0,236
1- 13	0,261	2- 13	0,201	3- 13	0,188	4- 13	0,220
1- 14	0,285	2- 14	0,215	3- 14	0,212	4- 14	0,234
1- 15	0,274	2- 15	0,209	3- 15	0,197	4- 15	0,225

На рис.7.3.21 представлена поверхня відгуку для капронової комплексної нитки. Рівняння регресії для капронової комплексної нитки

буде мати вигляд

$$kt = 0,277 + 0,0098x_1 + 0,018x_2 + 0,0069x_3 - 0,002x_1^2 + 0,0018x_1x_2 + 0,0008x_1x_3 - 0,0014x_2^2 - 0,002x_2x_3 - 0,0035x_3^2. \quad (7.3.19)$$

$$x_3=0,0$$

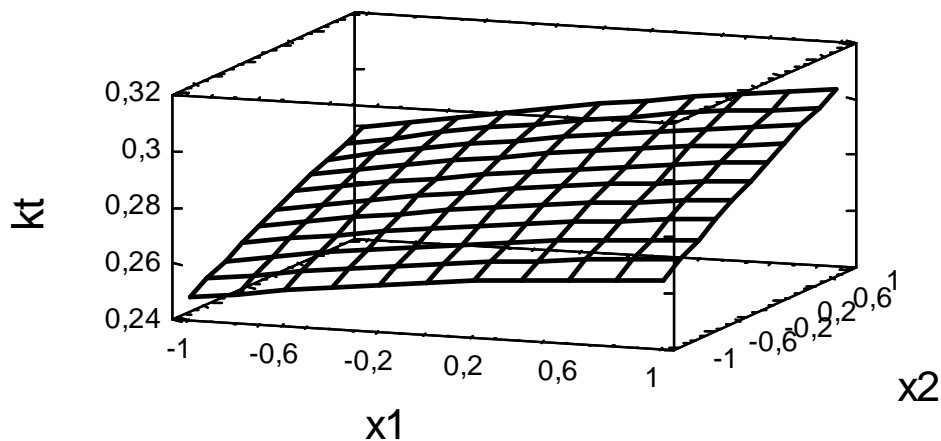


Рис.7.3.21. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.19) отримаємо відповідну діаграму, яка представлена на рис.7.3.22. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_3^2 , x_2^2 , x_1^2 .

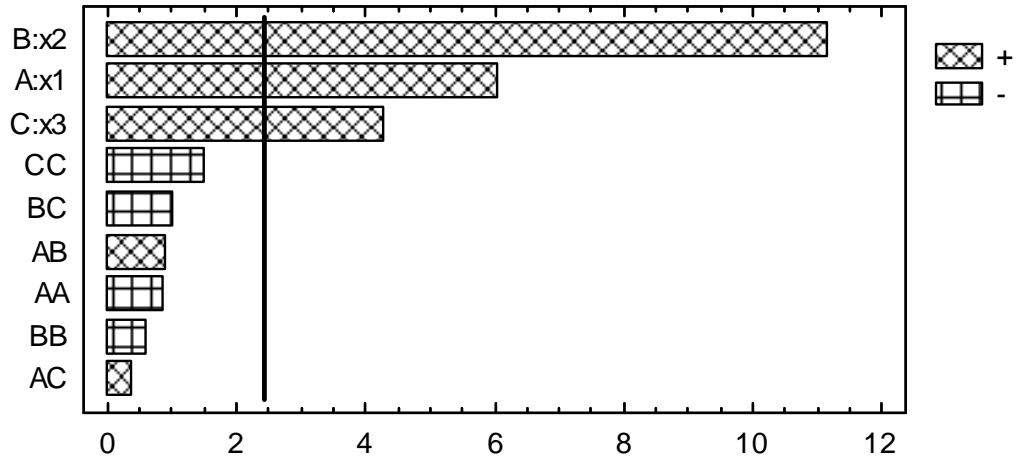


Рис.7.3.22. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.19), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,134 + 0,0072P_0 + 0,0095R + 0,018\phi. \quad (7.3.20)$$

На рис.7.3.23 представлена поверхня відгуку для віскозної комплексної нитки. Рівняння регресії для віскозної комплексної нитки буде мати вигляд

$$kt = 0,205 + 0,0065x_1 + 0,0044x_2 + 0,0037x_3 + 0,004x_1^2 - 0,0015x_1x_2 + 0,00125x_1x_3 + 0,0049x_2^2 + 0,0015x_2x_3 + 0,0034x_3^2. \quad (7.3.21)$$

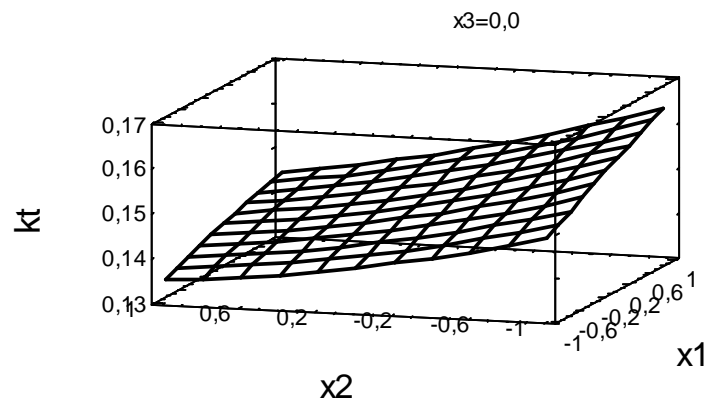


Рис.7.3.23. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.21) отримаємо відповідну діаграму, яка представлена на рис.7.3.27.3. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_2^2 , x_1^2 .

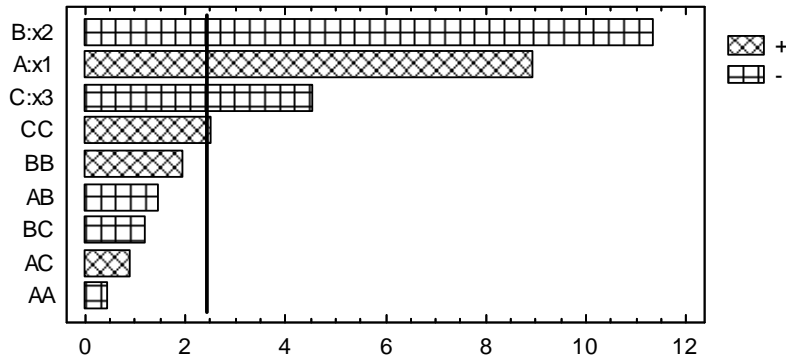


Рис.7.3.27.3. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.21), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,183 - 0,0054P_0 + 0,0065R + 0,0089\phi + 0,00036P_0^2. \quad (7.3.22)$$

На рис.7.3.25 представлена поверхня відгуку для вовняної пряжі. Рівняння регресії для вовняної пряжі буде мати вигляд

$$kt = 0,199 + 0,0075x_1 + 0,016x_2 + 0,0059x_3 - 0,00009x_1^2 + 0,001x_1x_2 + 0,00000125x_1x_3 - 0,0022x_2^2 - 0,00025x_2x_3 - 0,00039x_3^2. \quad (7.3.23)$$

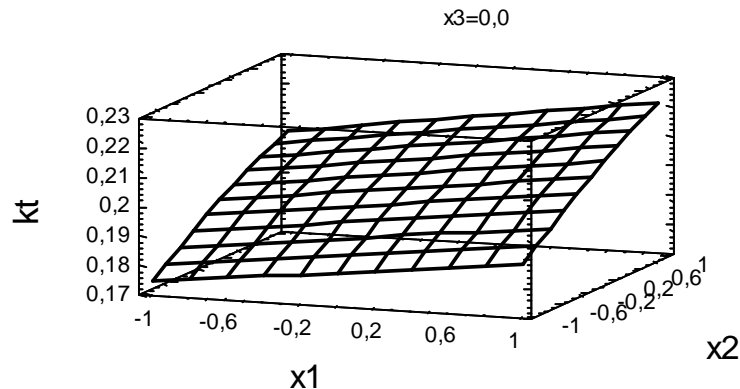


Рис.7.3.25. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.23) отримаємо відповідну діаграму, яка представлена на рис.7.3.26. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_3^2 , x_2^2 , x_1^2 .

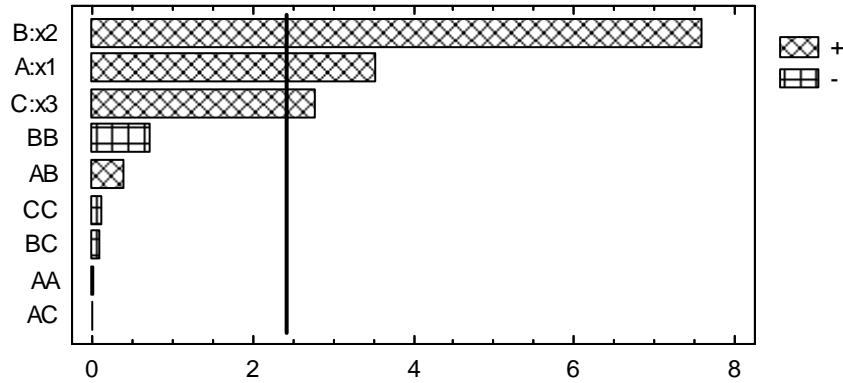


Рис.7.3.26. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.23), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,0765 + 0,0064P_0 + 0,0075R + 0,015\phi. \quad (7.3.24)$$

На рис.7.3.27 представлена поверхня відгуку для бавовняної пряжі. Рівняння регресії для бавовняної пряжі буде мати вигляд

$$kt = 0,219 + 0,0045x_1 + 0,0069x_2 + 0,0035x_3 + 0,0066x_1^2 + 0,0000001x_1x_2 + 0,00000001x_1x_3 + 0,0063x_2^2 + 0,00000001x_2x_3 + 0,0066x_3^2. \quad (7.3.25)$$

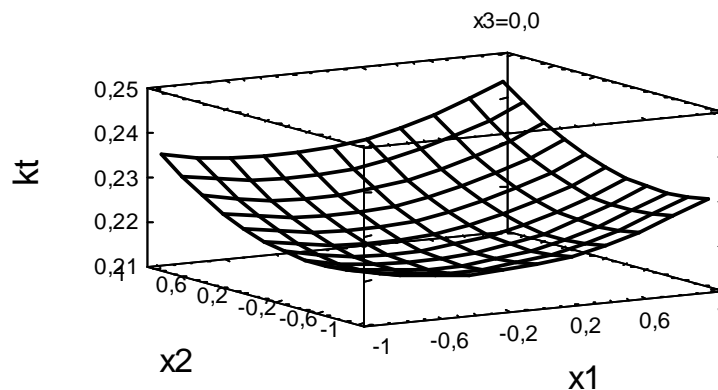


Рис.7.3.27. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (7.3.25) отримаємо відповідну діаграму, яка представлена на рис.7.3.28. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_2x_3 , x_1x_2 , x_3^2 .

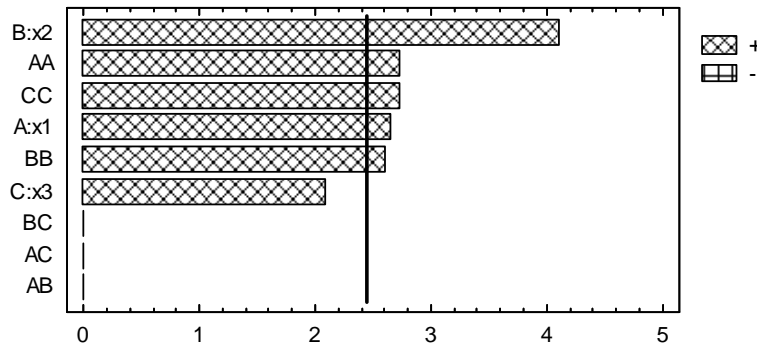
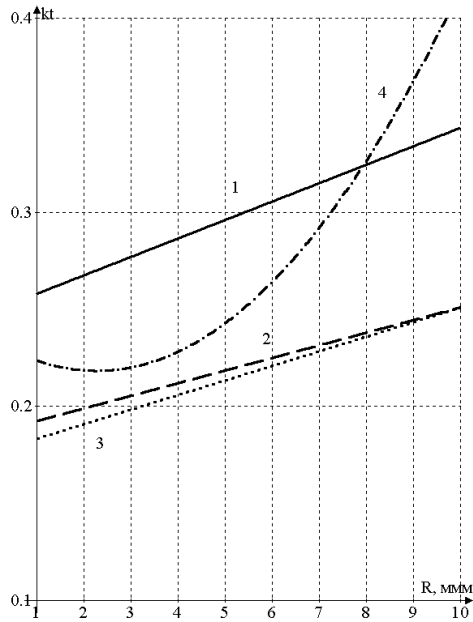


Рис.7.3.28. Діаграма значущості коефіцієнтів в рівнянні регресії

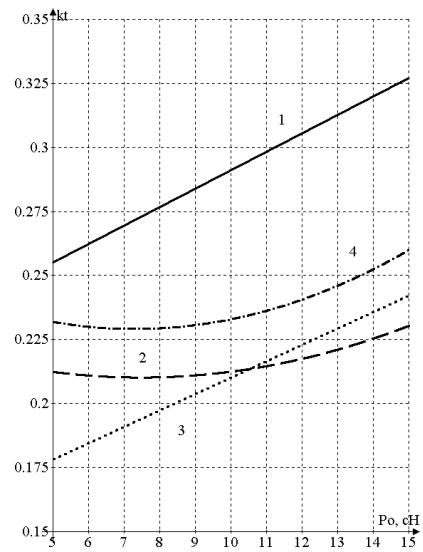
З урахуванням значущості коефіцієнтів в рівнянні регресії (7.3.25), а також з урахуванням системи (7.3.10), перейдемо до іменованих значень, остаточно отримаємо

$$kt = 0,376 - 0,0076P_0 - 0,015R - 0,099\phi + 0,00052P_0^2 + 0,0033R^2 + 0,021\phi^2. \quad (7.3.26)$$

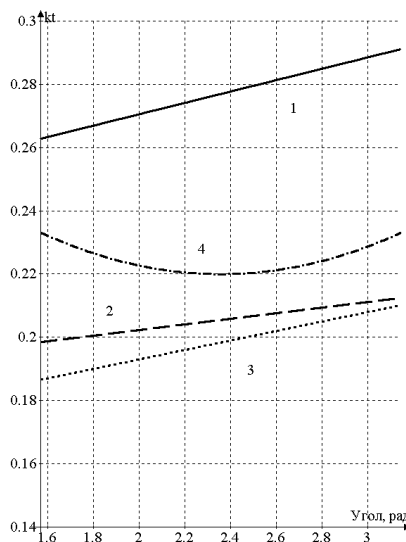
Розрахунки по формулам (7.3.20), (7.3.22), (7.3.24), (7.3.26) дозволили отримати масив даних (дивись додаток А), використовуючи який були побудовані графічні залежності представлені на рис.7.3.29-7.3.31.



7.3.29. Залежність коефіцієнту тертя в поперечному напрямку руху нитки від радіусу кривизни



7.3.30. Залежність коефіцієнту тертя в поперечному напрямку руху нитки від натягу веденої гілки нитки



7.3.31. Залежність коефіцієнту тертя в поперечному напрямку руху нитки від кута обхвату

Аналіз графічних залежностей на рис.7.3.29-7.3.31 показав, що при збільшенні радіусу кривизни напрямної, натягу веденої гілки (рис.7.3.30) та кута обхвату (рис.7.3.31) коефіцієнт тертя в подовжньому напрямку руху

нитки зростає. Це пояснюється тим, що окремі мікрОВОлокна в структурі нитки чи пряжі займають таке положення, при якому їх вісь буде перпендикулярна до вектора швидкості [110- 113]. Ці результати необхідно враховувати при проектуванні технологічних процесів легкої та текстильної промисловості.

7.4. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ТРИКОТАЖНОГО ТА ТКАЦЬКОГО ВИРОБНИЦТВА З УРАХУВАННЯМ АНІЗОТРОПІЇ ТЕРТЯ НИТОК

Поперечне ковзання ниток зустрічається в багатьох технологічних процесах текстильної і трикотажної промисловості [7, 8].

Поперечне ковзання ниток по напрямних з постійною кривиною поверхні має місце при формуванні елемента тканини, при взаємодії ниток з напрямними органами трикотажних та текстильних машин. Визначення натягу та форми вісі, з урахуванням матеріалу та анізотропії фрикційних властивостей ниток, дозволяє на початковому періоді проектування технологічних процесів обрати їх оптимальні параметри. Таким чином, тема даної статті є актуальною, яка має важливе значення для удосконалення технологічних процесів текстильної та трикотажної промисловості.

Удосконалення технологічних процесів легкої та текстильної промисловості, як об'єктів дослідження, повинно базуватися на комплексних дослідженнях процесу взаємодії ниток з напрямними поверхнями малої кривини. Отримання теоретичних залежностей дозволить отримати рівняння для визначення натягу та форми вісі нитки з урахуванням матеріалу та анізотропії її фрикційних властивостей в зоні формування тканини та трикотажу, зменшити відсоток обривів ниток, поліпшити якість готової продукції [5-8].

Визначення впливу будови поверхонь робочих органів трикотажних і текстильних машин на форму і натяг ниток

Будова поверхонь робочих органів трикотажних і текстильних машин (рис.7.4.1), а також самих ниток (рис.7.4.2) свідчать про те, що величина сили тертя в ортогональних напрямках (наприклад дотичній і нормалі до точки на осі нитки) відрізнятимуться. Це дозволяє говорити про необхідність врахування анізотропії фрикційних властивостей [10-12].

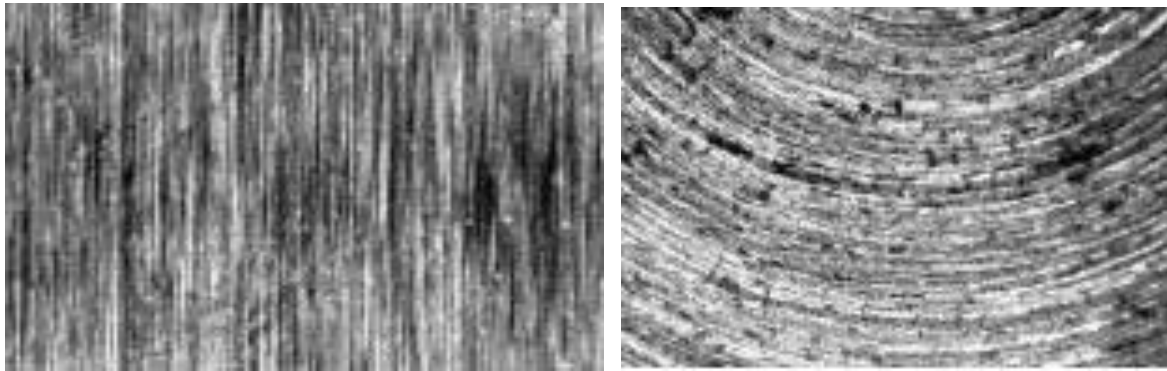


Рис.7.4.1. Будова поверхні робочих органів

У загальному випадку величина сили тертя визначається з виразу [11]

$$F = N\sqrt{k_1^2 + k_2^2}, \quad (7.4.1)$$

де F - сила тертя;

N - сила нормального тиску;

$\sqrt{k_1^2 + k_2^2}$ - ефективний коефіцієнт тертя [5].

Для розкриття його суті представимо силу тертя як функцію швидкості

$$\vec{F} = -\frac{NA\vec{v}}{v}, \quad (7.4.2)$$

де A - матриця, яка задає лінійне переміщення;

v - швидкість точки на осі нитки.

Для двох ортогональних напрямів (дотична і нормаль) вираз для даної матриці матиме вигляд

$$A = \begin{pmatrix} \mu_1 \cos \varphi & 0 \\ 0 & \mu_2 \sin \varphi \end{pmatrix}, \quad (7.4.3)$$

де μ_1, μ_2 - коефіцієнти тертя нитки в двох ортогональних напрямках [110-115];

φ - кут, який утворює вектор швидкості з дотичної до осі нитки.

Тоді ефективний коефіцієнт тертя визначимо по формулі

$$\sqrt{k_1^2 + k_2^2} = \sqrt{\mu_1^2 \cos^2 \varphi + \mu_2^2 \sin^2 \varphi}. \quad (7.4.4)$$

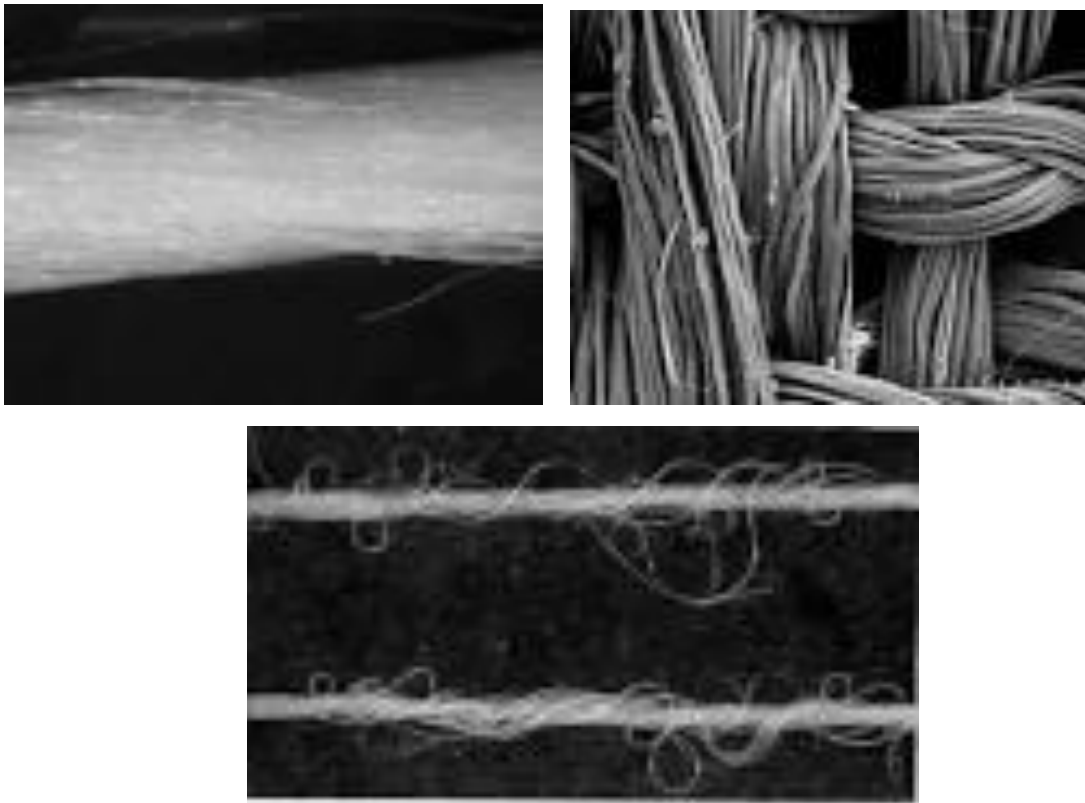


Рис.7.4.2. Будова комплексних ниток та пряжі

Для теоретичного дослідження впливу матеріалу нитки і анізотропії тертя на натяг і форму її осі були вибрані віскозна комплексна нитка 16,7 Текс, капронова комплексна нитка 31,2 Текс, бавовняна гребінна кручена пряжа 23,6 Текс, шерстяна гребінна кручена пряжа 36 Текс.

Вибір даного асортименту, в першу чергу, пояснюється приблизно однаковим тексом ниток. Крім того, даний вид сировини широко використовується для виготовлення текстильних і трикотажних виробів. За наслідками експериментальних досліджень коефіцієнти тертя в подовжньому і поперечному напрямках для даного виду сировини відповідно дорівнювали: віскозна комплексна нитка $\mu_1 = 0,16$, $\mu_2 = 0,21$; капронова комплексна нитка $\mu_1 = 0,18$, $\mu_2 = 0,24$; бавовняна гребінна кручена пряжа $\mu_1 = 0,17$, $\mu_2 = 0,22$; шерстяна гребінна кручена пряжа $\mu_1 = 0,14$, $\mu_2 = 0,18$.

Визначення сили прибою для одношарових тканин в залежності від щільності по основі та з урахуванням анізотропії основи та утоку

Сила прибою є основним технологічним параметром, що визначає напруженість процесу формування тканини [116, 117]. На рис.7.4.3 представлена розрахункова для визначення величини кута охоплення основної нитки уточної. Половина кута охоплення визначатиметься за формулою

$$\beta = \arcsin\left(\frac{d_o P_o}{100}\right), \quad (7.4.5)$$

де β - половина кута охоплення основної нитки уточної;

d_o - розрахунковий діаметр основної нитки;

P_o - щільність тканини по основі, н/дм.

У процесі прибою вигнута уточнююча нитка ковзає по поверхні основних ниток. Виникаючий при цьому опір буде визначати величину сили прибою. При цьому, можна вважати, що утоки нитки будуть розташовуватися в площині, нахиленої до твірної циліндричної поверхні основної нитки під кутом

$$\gamma = \arctan\left(\frac{1}{\mu_2}\right). \quad (7.4.6)$$

де μ_2 - коефіцієнт тертя уточної нитки по поверхні основної в поперечному напрямку.

Для дослідження процесу прибою будемо використовувати циліндричну систему координат [107, 108]. Для спрощення будемо вважати, що основні і утовкові нитки мають циліндричну форму. Тоді основні співвідношення між циліндричними координатами будуть мати вигляд

$$\begin{aligned} r &= r_0 + r_y, \quad r \frac{d\varphi}{ds} = \sin \gamma, \\ \frac{dz}{ds} &= \cos \gamma, \end{aligned} \quad (7.4.7)$$

де r_0, r_y - розрахункові радіуси поперечного перетину основної і уточної нитки;

φ - центральний кут, який визначає положення радіуса r в перетині основної нитки;

z - координата, що визначає положення поточної точки на осі уточної нитки щодо осі основної нитки;

s - дугова координата.

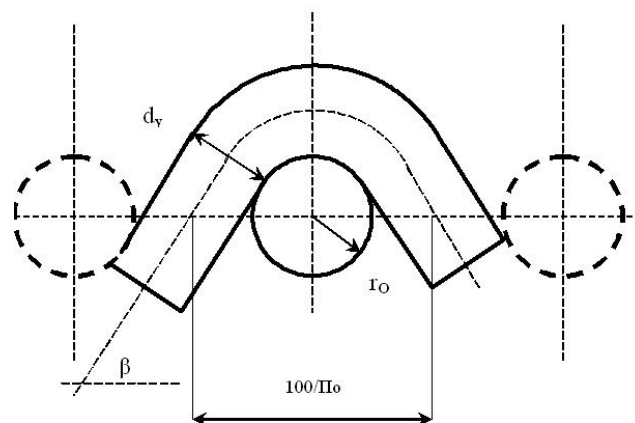


Рис.7.4.3. Розрахунковим схема

Основна система диференціальних рівнянь, що описують рівновагу уточної нитки на поверхні основи (див. розділ 2), з урахуванням (7.4.4), буде мати вигляд [107, 108]

$$\begin{aligned} \frac{d}{ds} \left(P \frac{dr}{ds} \right) - Pr \left(\frac{d\varphi}{ds} \right)^2 + N &= 0, \\ \frac{d}{ds} \left(Pr \frac{d\varphi}{ds} \right) &= 0, \\ \frac{d}{ds} \left(P \frac{dz}{ds} \right) - \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N &= 0, \\ \left(\frac{dr}{ds} \right)^2 + r^2 \left(\frac{d\varphi}{ds} \right)^2 + \left(\frac{dz}{ds} \right)^2 &= 1, \end{aligned} \tag{7.4.8}$$

де P - натяг нитки, H ;

N - нормальна реакція поверхні, H .

Враховуючи, що в якості направляючої поверхні був обраний циліндр ($r = const$), система диференціальних рівнянь (7.4.8), з урахуванням (7.4.5) - (7.4.7), прийме вигляд

$$\begin{aligned} -Pr \left(\frac{d\varphi}{ds} \right)^2 + N &= 0, \\ \frac{d}{ds} \left(Pr \frac{d\varphi}{ds} \right) &= 0, \\ \frac{d}{ds} \left(P \frac{dz}{ds} \right) - \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N &= 0, \\ r^2 \left(\frac{d\varphi}{ds} \right)^2 + \left(\frac{dz}{ds} \right)^2 &= 1. \end{aligned} \tag{7.4.9}$$

Підставимо в останню систему диференціальних рівнянь (7.4.9) вираження (7.4.7), отримаємо

$$\begin{aligned} -Pr \left(\frac{\sin \gamma}{r} \right)^2 + N &= 0, \\ \frac{d}{ds} (P \sin \gamma) &= 0, \\ \frac{d}{ds} (P \cos \alpha) - \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N &= 0. \end{aligned} \tag{7.4.10}$$

Перетворимо систему диференціальних рівнянь (7.4.10) до виду

$$N = Pr\left(\frac{\sin \gamma}{r}\right)^2,$$

$$\frac{d}{ds}(P \sin \gamma) = 0, \quad (7.4.11)$$

$$\frac{d}{ds}(P \cos \gamma) = \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N.$$

Підставляємо з першого рівняння системи диференціальних рівнянь (7.4.9) значення нормальної реакції поверхні в третє рівняння, отримаємо

$$\cos \alpha \frac{dP}{ds} = P \frac{\sin^2 \alpha}{r} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma}.$$

Перетворимо останнє диференціальне рівняння до виду

$$\frac{dP}{P} = \frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} ds.$$

Останнє диференціальне рівняння є диференціальним рівнянням із розділюваними перемінними. Зінтегруємо його

$$\frac{dP}{P} = \frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} ds.$$

$$\int_{P_0}^{P_1} \frac{dP}{P} = \frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} \int_0^s ds.$$

Або

$$\ln \frac{P_1}{P_0} = \frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} s,$$

де P_1 - натяг утокової нитки в точці сходу з основи, Н;

P_0 - натяг утокової нитки в точці перегину на траєкторії (см. рис. 7.4.3), Н;

s - довжина дуги нитки від точки сходу з направляючої до точки перегину,

м.

В остаточному вигляді отримаємо

$$P_1 = P_0 e^{\frac{\sin^2 \alpha}{r \cos \alpha} \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} s}. \quad (7.4.12)$$

Вираз для визначення елементарної сили тертя, що діє на утокову нитку має вигляд

$$d(P \cos \gamma) = \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N ds.$$

Зінтегруємо його

$$F_{TP\Sigma} = \int_0^s \sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} N ds = 2 \cos \gamma \int_{P_0}^P dP = 2 \cos \gamma (P - P_0).$$

В остаточному вигляді матимемо

$$F_{TP\Sigma} = 2 \cos \gamma P_0 \left(e^{\sqrt{\mu_1^2 \cos^2 \gamma + \mu_2^2 \sin^2 \gamma} \frac{\sin^2 \gamma}{r \cos \gamma} s} - 1 \right). \quad (7.4.13)$$

Зв'язок між змінними, з урахуванням (7.4.7), визначатиметься з виразу

$$s = \frac{\beta(d_0 + dy)}{\sin \gamma}, \quad d_0 = 2r_0.$$

З урахуванням (7.4.13) вираз для визначення результуючої сили прибою визначатиметься з виразу

$$P_{ПП} = \sum_{i=1}^n 2 \cos \gamma_i P_{0i} \left(e^{\sqrt{\mu_1^2 \cos^2 \gamma_i + \mu_2^2 \sin^2 \gamma_i} \frac{\sin^2 \gamma_i}{r \cos \gamma_i} s_i} - 1 \right), \quad (7.4.14)$$

де n - число ниток основи в заправці верстата.

На рис.7.4.4 - 7.4.5 представлені діаграми залежності сили прибою від виду переплетення, виду ниток отримані з урахуванням (7.4.14). Порівняння показує, що більше значення матимемо у випадку тканини полотняного переплетення. Це пояснюється тим, що для полотняного переплетення кут охоплення основи утоків в рапорті буде більшим, ніж у випадку переплетення репс основний. Ця різниця, в перерахунку на одну нитку, буде для $P_0=110$ н/дм – 0,5 сН; для $P_0=250$ н/дм – 2 сН; для $P_0=470$ н/дм – 6 сН.

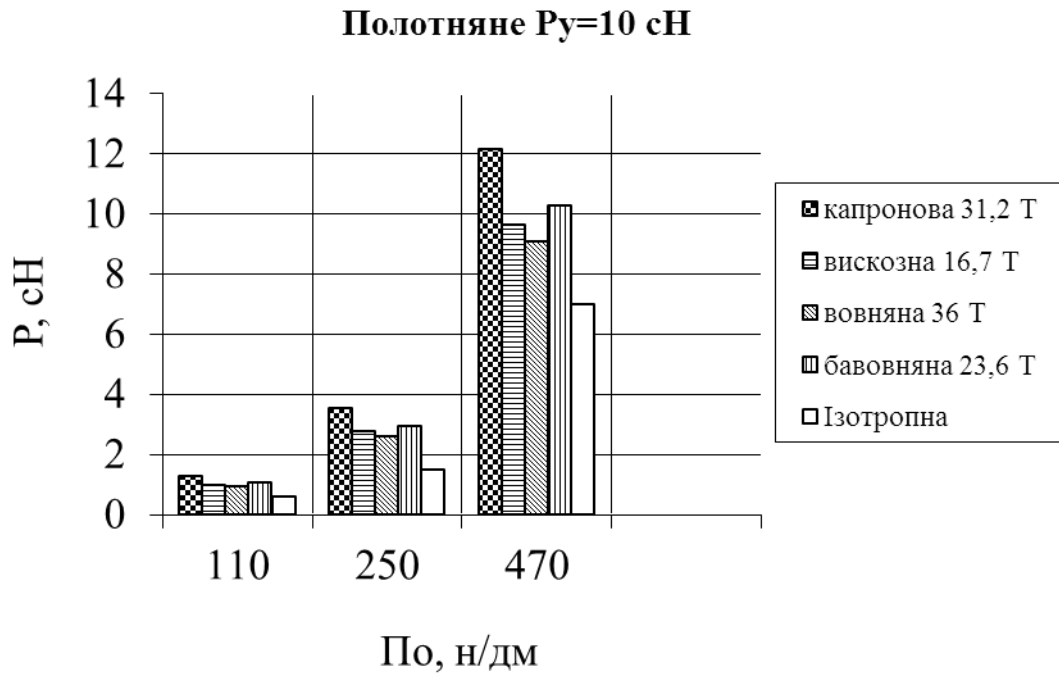


Рис.7.4.4.Залежність сили прибою для тканини полотняного переплетення залежно від щільності по основі

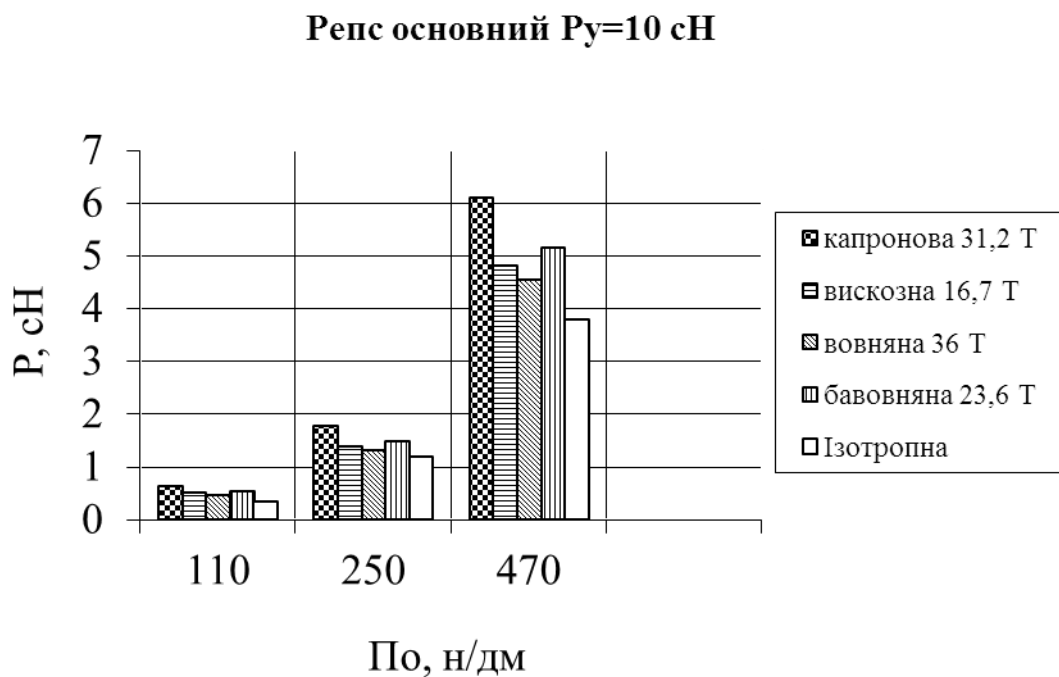


Рис.7.4.5. Залежність сили прибою для тканини переплетення репс основний залежно від щільності по основі

На діаграмах 7.4.4-7.4.5 представлено також значення сили прибою для випадку, коли нитка має ізотропні фрикційні властивості. В даному випадку різниця між анізотропною і ізотропною моделями становить 40-50%.

Визначення поверхнево заповнення одношарової тканини в залежності від умів прибою

В процесі формування тканини утокова нитка під дією зубів берда переміщується до опушки тканини. При цьому, в залежності від номера берда та числа ниток основи пробраних в один зуб, нитки утока будуть вигинатися як показано на рис.7.4.6 [15-18].

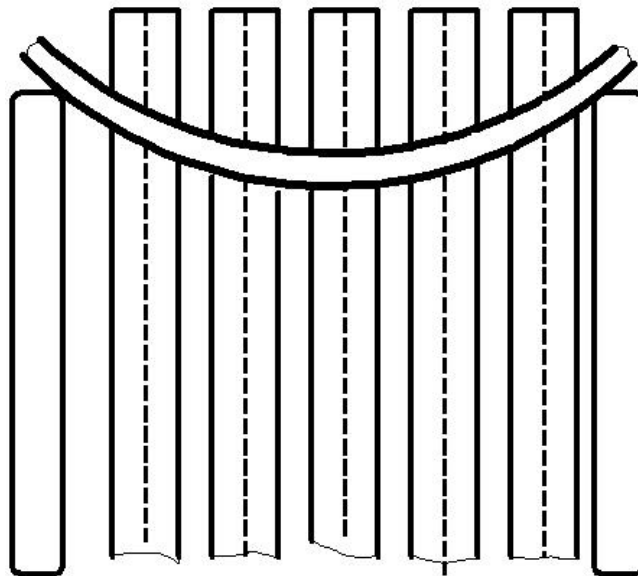


Рис.7.4.6. Розрахункова схема

За рахунок цього сумарна довжина утока збільшується і у вільному стані, коли тканина знята з верстата і фаза будови тканини буде близькою до п'ятої [11, 14], фактично буде змінюватися щільність тканини по основі (в даному випадку будемо рахувати, що щільність тканини по утоку залишається незмінною).

Визначимо як впливає вигин нитки при пробою з урахуванням анізотропії фрикційних властивостей. На рис.7.4.7 показаний переріз тканини вздовж утокових ниток. Визначимо основні геометричні параметри переплетення як функції щільності тканини по основі.

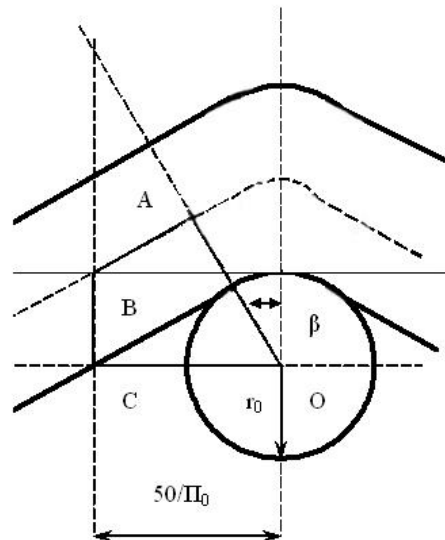


Рис.7.4.7.Переріз тканини вздовж утокових ниток

Визначаємо дві сторони чотирикутника $OABC$. Сторона BO буде дорівнювати

$$BO = \sqrt{r_0^2 + \left(\frac{50}{\Pi_0}\right)^2}. \quad (7.4.15)$$

де r_0 - радіус основи;

Π_0 - щільність тканини по основі.

Сторона AB буде дорівнювати

$$AB = \sqrt{BO^2 - (r_0 + r_y)^2} = \sqrt{r_0^2 + \left(\frac{50}{\Pi_0}\right)^2 - (r_0 + r_y)^2}. \quad (7.4.16)$$

де r_y - радіус уточної нитки.

Проектуючи сторони чотирикутника $OABC$ на горизонталь, з урахуванням (7.4.14)-(7.4.16), отримаємо

$$\begin{aligned} (r_O + r_y) \sin \beta + AB \cos \beta &= \frac{50}{\Pi_O}, \\ (r_O + r_y) \sin \beta + \sqrt{r_O^2 + \left(\frac{50}{\Pi_O}\right)^2 - (r_O + r_y)^2} \cos \beta &= \frac{50}{\Pi_O}. \end{aligned} \quad (7.4.17)$$

Для вирішення рівняння (7.4.17) будемо використовувати метод додаткового кута [5, 6]

$$\frac{(r_O + r_y)}{\sqrt{(r_O + r_y)^2 + AB^2}} \sin \beta + \frac{\sqrt{r_O^2 + \left(\frac{50}{\Pi_O}\right)^2 - (r_O + r_y)^2}}{\sqrt{(r_O + r_y)^2 + AB^2}} \cos \beta = \frac{50}{\Pi_O \sqrt{(r_O + r_y)^2 + AB^2}}$$

Введемо наступні позначення

$$\cos \beta' = \frac{(r_O + r_y)}{\sqrt{(r_O + r_y)^2 + AB^2}}, \quad \sin \beta' = \frac{\sqrt{r_O^2 + \left(\frac{50}{\Pi_O}\right)^2 - (r_O + r_y)^2}}{\sqrt{(r_O + r_y)^2 + AB^2}}. \quad (7.4.18)$$

З другого рівняння системи (7.4.18) знайдемо

$$\beta' = \arcsin \left[\frac{\sqrt{r_O^2 + \left(\frac{50}{\Pi_O}\right)^2 - (r_O + r_y)^2}}{\sqrt{(r_O + r_y)^2 + AB^2}} \right]. \quad (7.4.19)$$

Підставляємо (7.4.18) в рівняння (7.4.17), отримаємо

$$\sin \beta \cos \beta' + \cos \beta \sin \beta' = \frac{50}{\Pi_O \sqrt{(r_O + r_y)^2 + AB^2}}. \quad (7.4.20)$$

Використовуючи основні тригонометричні співвідношення знайдемо

$$\sin(\beta + \beta') = \frac{50}{\Pi_O \sqrt{(r_O + r_y)^2 + AB^2}}.$$

Остаточний вираз для визначення кута нахилу гілки утокової нитки до горизонтальної площини буде мати вигляд

$$\beta = \arcsin \left[\frac{50}{\Pi_o \sqrt{(r_o + r_y)^2 + AB^2}} \right] - \arcsin \left[\frac{\sqrt{r_o^2 + \left(\frac{50}{\Pi_o} \right)^2 - (r_o + r_y)^2}}{\sqrt{(r_o + r_y)^2 + AB^2}} \right]. \quad (7.4.21)$$

З використанням (7.4.15), (7.4.16), (7.4.21) знайдемо вираз для визначення довжини утокової нитки в рапорті (ідеальна модель, коли нитка утока буде розташована в дотичній площині до перерізу тканини)

$$L = 2\beta(r_o + r_y) + 2\sqrt{r_o^2 + \left(\frac{50}{\Pi_o} \right)^2 - (r_o + r_y)^2}. \quad (7.4.22)$$

Визначимо реальну довжину вигнутої уточної нитки. Для цього скористуємося результатами, отриманими в 2 розділі. Інтегрування диференційного рівняння дозволило отримати наступний вираз

$$\int_0^{y_0} \left[1 + \left(\frac{dy}{dx} \right)^2 \right]^{\frac{k_1}{2k_2} - 1} d \left(\frac{dy}{dx} \right) = \int_0^{\Pi_o} \frac{k_2 \mu_0 g}{P_0} dx. \quad (7.4.23)$$

Знайти інтеграл (7.4.23) у квадратурах неможливо. Для його рішення використовувалися чисельні методи та спеціально розроблене програмне забезпечення (див. розділ 2). Отримані результати апроксимувалися за допомогою відповідних методик. Для віскозної комплексної нитки 16,7 Т, з урахуванням анізотропії фрикційних властивостей (розділ 3), був отриманий вираз для визначення першої похідної

$$\frac{dy}{dx} = ax^2 - bx + c. \quad (7.4.24)$$

Тоді вираз для визначення довжини утокової нитки між зубами берда [115-118] буде мати вигляд

$$L1 = 2 \int_0^{\frac{50}{\Pi_o}} \sqrt{1 + \left(\frac{dy}{dx} \right)^2} dx,$$

де $L1$ - довжини утокової нитки між зубами берда.

Для визначення реальної щільності тканини, з урахуванням (7.4.21)-(7.4.22), ми отримали систему двох трансцендентних рівнянь

$$\beta_1 = \arcsin \left[\frac{50}{\Pi_{O1} \sqrt{(r_O + r_y)^2 + AB1^2}} \right] - \arcsin \left[\frac{\sqrt{r_O^2 + \left(\frac{50}{\Pi_{O1}} \right)^2 - (r_O + r_y)^2}}{\sqrt{(r_O + r_y)^2 + AB1^2}} \right], \quad (7.4.25)$$

$$L1 = 2\beta_1(r_O + r_y) + 2\sqrt{r_O^2 + \left(\frac{50}{\Pi_{O1}} \right)^2 - (r_O + r_y)^2},$$

де Π_{O1} - реальна щільність тканини по основі з урахуванням прогину утокової нитки між зубами берда;

β_1 - реальний кут нахилу гілки утокової нитки до горизонтальної площини.

Значення $AB1$ знайдемо з урахуванням (7.4.16)

$$AB1 = \sqrt{r_O^2 + \left(\frac{50}{\Pi_{O1}} \right)^2 - (r_O + r_y)^2}. \quad (7.4.26)$$

Для рішення системи трансцендентних рівнянь (7.4.25)-(7.4.26) будемо використовувати чисельні методи алгоритм та програма для його реалізації наведені в 2 розділі.

В результаті розрахунків по формулах (7.4.21)-(7.4.22) для віскозних комплексних ниток 16,7 Т для щільності тканини по основі $\Pi_O = 200$ н/дм (без врахування вигину утокової нитки між зубами берда) отримали $L = 0,527$ мм, $\beta = 0,346$ рад. Чисельне рішення системи трансцендентних рівнянь (7.4.25)-(7.4.26) дозволило визначити реальні значення довжини утокової нитки між зубами берда, кута нахилу гілки утокової нитки до горизонтальної площини та реальної щільності по основі, відповідно $L1 = 0,579$ мм, $\beta = 0,306$ рад, $\Pi_{O1} = 180$ н/дм [15-18].

Визначимо поверхневе заповнення тканини z_{II} для цих двох випадків за формулами [11, 25]

$$\begin{aligned} z_{II} &= z_O + z_Y - \frac{z_O z_Y}{100}, \\ z_O &= 0,0316 P_O k \sqrt{T_O}, \\ z_Y &= 0,0316 P_Y k \sqrt{T_Y}, \end{aligned} \tag{7.4.27}$$

де P_Y - щільність тканини по утку (приймалася постійною та рівною $P_Y=150$ н/дм);

k - постійні коефіцієнти, для визначення діаметра нитки (для віскозної комплексної нитки $k = 0,0411$ [9]).

У результаті розрахунків отримали $z_{II} = 1,845$ та $z_{II} = 1,74$ (для щільності по основі $P_O=180$ н/дм). Таким чином зниження заповнення тканини склало 5,7%.

Таке зменшення заповнення тканини призвело до збільшення відносної повітропроникності B на 6,1% та зменшенню відносної стійкості до стирання I на 10,9 % (розрахунки виконувалися за відомою методикою [14, 25]). На рис.7.4.8 представлені відповідні діаграми.

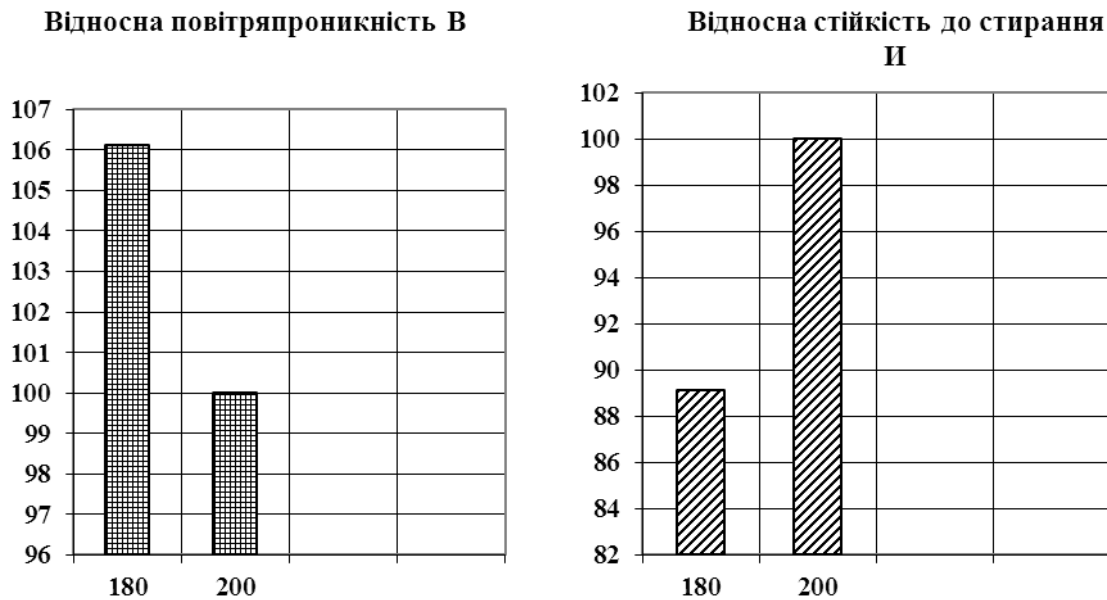


Рис.7.4.8. Діаграми значень відносних повітропроникності та стійкості до стирання

Вплив анізотропії фрикційних властивостей матеріалу нитки на кут нахилу петлі при переміщенні голки при завершенні

Операція завершення спрямована на перевід старої петлі на стрижень трикотажної голки. Закінчується ця операція коли трикотажна голка опиниться в верхньому положенні. Процес завершення ускладнюється тим, що стара петля під дією сил тертя розташовується під деяким кутом до відбійної площини. Визначення цього кута дозволить оптимізувати величину робочого руху трикотажної гільки.

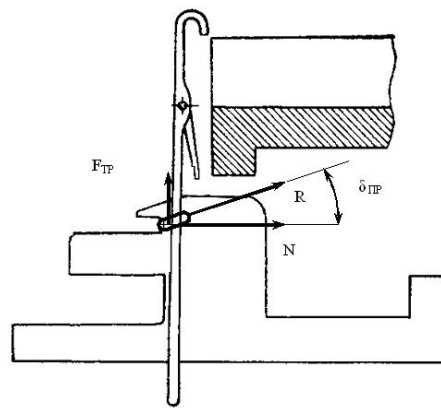


Рис.7.4.9. Схема взаємодії старої петлі зі стрижнем голки

Значення поточного натягу нитки в нормальній площині при взаємодії з стрижнем трикотажної гільки визначається за формулою

$$P = P_0 e^{-\mu l \alpha}, \quad (7.4.28)$$

де P - поточний натяг нитки;

P_0 - зусилля відтяжки полотна, віднесене до однієї з гілок петлі;

μl - коефіцієнт тертя нитки в подовжньому напрямку;

α - поточний кут обхвату ниткою стрижня трикотажної голки.

Елементарний тиск нитки на стрижень трикотажної голки буде рівним

$$dN_1 = P d\alpha, \quad (7.4.29)$$

де N_1 - елементарний тиск на стрижень трикотажної голки в нормальній площині.

Тиск на трикотажну гілку елементарної ділянки нитки визначається як проекція на головну нормаль усіх діючих на неї сил

$$dN = dN_1 \cos \delta_{\text{ПР}},$$

де N - елементарний тиск на стрижень трикотажної голки;

$\delta_{\text{ПР}}$ - кут нахилу трикотажної петлі.

Вираз для визначення елементарної сили тертя, яка діє вздовж трикотажної голки, з урахуванням (7.4.28)-(7.4.29), має вигляд

$$dF_{\text{ТР}} = \mu 2dN = \mu 2P_0 e^{-\mu l \alpha} \cos \delta_{\text{ПР}} d\alpha, \quad (7.4.30)$$

де $\mu 2$ - коефіцієнт тертя нитки в поперечному напрямі.

Інтегруємо (7.4.30), отримаємо

$$F_{\text{ТР}} = 2 \int_0^{\frac{\pi}{2}} \mu 2P_0 e^{-\mu l \alpha} \cos \delta_{\text{ПР}} d\alpha.$$

В остаточному вигляді будемо мати

$$F_{\text{ТР}} = \frac{2\mu 2P_0 (1 - e^{-\mu l \frac{\pi}{2}}) \cos \delta_{\text{ПР}}}{\mu l}. \quad (7.4.31)$$

Тангенс кута нахилу старої петлі буде дорівнювати відношенню сили тертя до нормальної реакції. З використанням (7.4.28), (7.4.29) та (7.4.31), отримаємо

$$\sin \delta_{\text{ПР}} = \frac{\mu 2 \cos \delta_{\text{ПР}}}{\mu l} (1 - e^{-\mu l \frac{\pi}{2}}).$$

Остаточно знаходимо значення кута нахилу петлі

$$\delta_{\text{ПР}} = \arctan \left[\frac{\mu 2}{\mu l} (1 - e^{-\mu l \frac{\pi}{2}}) \right]. \quad (7.4.32)$$

На рис.7.4.10 представлені значення кута нахилу трикотажної петлі з урахуванням анізотропії та для ізотропної нитки

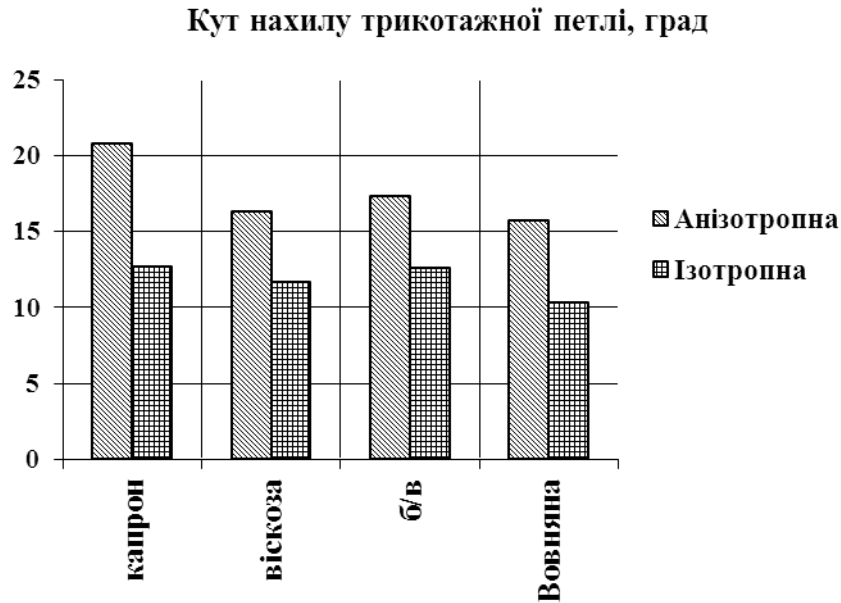


Рис.7.4.10. Діаграма кута нахилу трикотажної петлі

Аналіз отриманих даних дозволив встановити що у випадку, коли враховуються анізотропні властивості ниток значення кута нахилу старої трикотажної петлі до відбійної площини при завершенні зростає для капронової комплексної нитки 31,2 Т на 63,8% ($20,8^\circ$ для анізотропної, $12,7^\circ$ для ізотропної), віскозної комплексної нитки 16,7 Т на 39% ($16,3^\circ$ для анізотропної, $11,7^\circ$ для ізотропної), бавовняної пряжі 23,6 Т на 37,3% ($17,3^\circ$ для анізотропної, $12,6^\circ$ для ізотропної), вовняної пряжі 36 Т на 52,4% ($15,7^\circ$ для анізотропної, $10,3^\circ$ сН для ізотропної). Різниця при визначені кута нахилу трикотажної петлі з урахуванням анізотропії фрикційних властивостей та без урахування (нитка ізотропна) складає близько 48% [118].

Цю обставину необхідно враховувати при оптимізації величини робочого руху трикотажної гілки.

Розрахунок тарільчатого нитконатягувача з урахуванням анізотропії фрикційних властивостей та матеріалу нитки

Тарільчаті нитконатягувачі широко використовуються на сучасному технологічному обладнанні, що пов'язано з їх високими технічними характеристиками [9]. Їх подальша модернізація буде йти по шляху розширення технологічних можливостей, врахуванні реальних фізико-механічних показників ниток що переробляються на відповідному обладнанні. Зокрема, при розрахунках вихідного натягу не враховуються анізотропні властивості нитки. Це може призвести до значних похибок при визначенні вихідного натягу і, як наслідок, до порушення нормального протікання технологічного процесу.

Проведемо розрахунок вихідного натягу нитки з урахуванням її анізотропних фрикційних властивостей [11-12].

Значення тиску в точках контакту тарілок з ниткою визначимо із наступних рівнянь [9]

$$p = \frac{Q}{2[1 + \sin(\beta - \alpha)]}, \quad 0 \leq \frac{\alpha}{2} \leq \beta,$$

$$p = \frac{Q}{2[1 + \sin(\alpha - \beta)]}, \quad \frac{\alpha}{2} > \beta, \quad (7.4.33)$$

$$\beta = \arcsin\left(\frac{r}{R}\right),$$

де p - тиск в точках контакту тарілок з ниткою;

Q - тиск пружини нитконатягувача на верхню тарілку;

r - радіус стрижня;

R - радіус контактного кола;

α - кут охоплення ниткою стрижня.

Натяг нитки після першої точки контакту, з урахуванням (7.4.33), буде визначатися по формулі

$$P_1 = P_0 + 2\sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} p, \quad (7.4.34)$$

де P_0 - натяг нитки до нитконатягувача;

μ_1 - коефіцієнт тертя нитки в подовжньому напрямі;

μ_2 - коефіцієнт тертя нитки в поперечному напрямі.

Натяг нитки P_2 після контакту з циліндричним стрижнем буде визначатися по формулі

$$P_2 = P_1 e^{\mu_1 \alpha}. \quad (7.4.35)$$

Натяг нитки після другої точки контакту буде визначатися по формулі

$$P = P_2 + 2 \sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} p. \quad (7.4.36)$$

Спільно вирішуючи рівняння (7.4.34)-(7.4.36), отримаємо

$$P = (P_0 + 2 \sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} p) e^{\mu_1 \alpha} + 2 \sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} p. \quad (7.4.37)$$

Підставляємо в (7.4.37) значення тиску в точках контакту тарілок з ниткою (7.4.33) та отримаємо для випадку $0 \leq \frac{\alpha}{2} \leq \beta$

$$P = P_0 e^{\mu_1 \alpha} + \sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} \left\{ \frac{Q}{[1 + \sin(\beta - \alpha)]} \right\} (1 + e^{\mu_1 \alpha}) \quad (7.4.38)$$

Для випадку $\frac{\alpha}{2} > \beta$ отримаємо

$$P = P_0 e^{\mu_1 \alpha} + \sqrt{\mu_1^2 \sin^2 \frac{\alpha}{2} + \mu_2^2 \cos^2 \frac{\alpha}{2}} \left\{ \frac{Q}{[1 + \sin(\alpha - \beta)]} \right\} (1 + e^{\mu_1 \alpha}) \quad (7.4.39)$$

На рис.7.4.11 представлена діаграма значень натягу нитки після тарільчатого нитконатягувача з урахуванням анізотропії фрикційних властивостей нитки та без урахування.

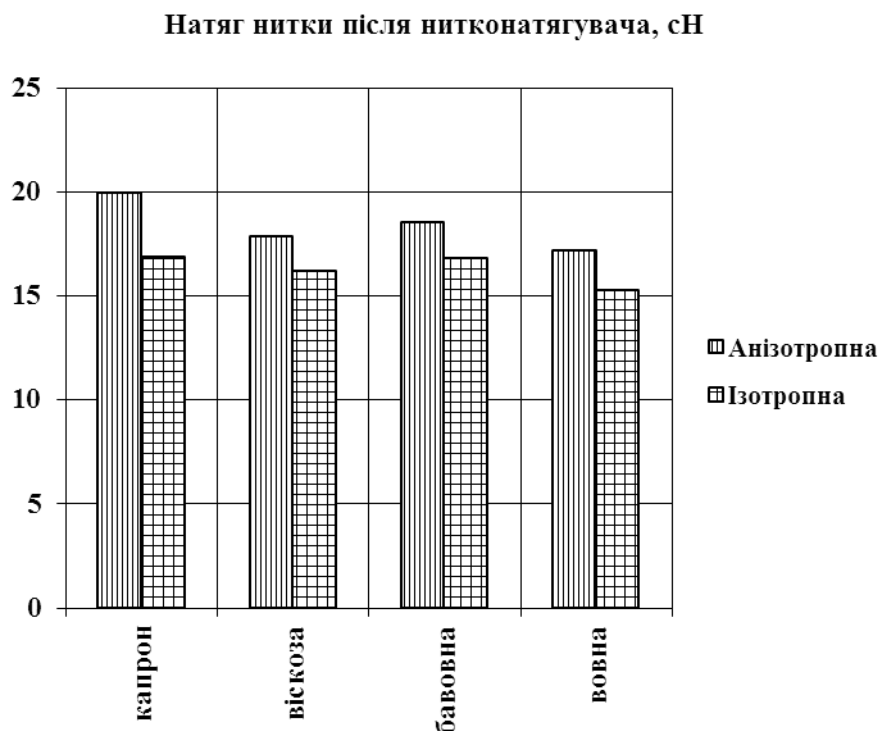


Рис.7.4.11. Діаграма натягу нитки після нитконатягувача

Аналіз отриманих даних дозволив встановити що у випадку, коли враховуються анізотропні властивості ниток вихідний натяг зростає для капронової комплексної нитки 31,2 Т на 18% (19,96 сН для анізотропної, 16,86 сН для ізотропної), віскозної комплексної нитки 16,7 Т на 10% (17,86 сН для анізотропної, 16,18 сН для ізотропної), бавовняної пряжі 23,6 Т на 10% (18,55 сН для анізотропної, 16,82 сН для ізотропної), вовняної пряжі 36 Т на 12,5% (17,17 сН для анізотропної, 15,26 сН для ізотропної). З урахуванням анізотропії натяг нитки після нитконатягувача, в середньому, на 17 % більше ніж для ізотропної нитки.

Цю обставину необхідно враховувати при проектуванні систем ниткоподачі для забезпечення мінімізації натягу в процесі переробки ниток на технологічному обладнанні [2-4].

8. УДОСКОНАЛЕННЯ ПРУЖНОЇ СИСТЕМИ ЗАПРАВКИ НИТОК НА КРУГЛОВ'ЯЗАЛЬНИХ МАШИНАХ НА ОСНОВІ СТАБІЛІЗАЦІЇ НАТЯГУ

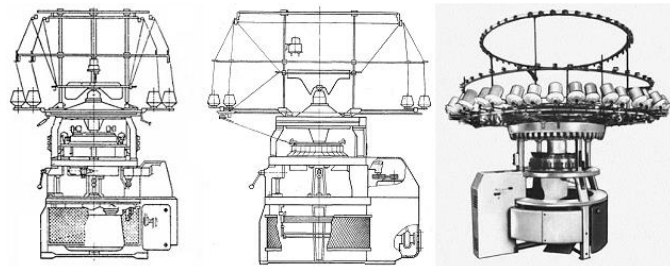
8.1. МАТЕМАТИЧНІ, АЛГОРИТМІЧНІ ТА ПРОГРАМНІ КОМПОНЕНТИ СИСТЕМИ ПРОЕКТУВАННЯ ФОРМИ ЛІНІЇ ЗАПРАВКИ НИТОК НА КРУГЛОВ'ЯЗАЛЬНИХ МАШИНАХ

Круглов'язальні машини, завдяки своїй продуктивності та широкому асортименту трикотажних полотен що виробляються на них, знаходять широке розповсюдження на трикотажних підприємствах та фірмах. В наш час широко використовуються круглов'язальні машини з утримувачем бобін, розташованим на підлозі. Це пов'язано з тим, що перенесення утримувача бобін на підлогу розвантажило корпус в'язальної машини, підвищило її стан рівноваги, зменшило динамічні навантаження, що, у свою чергу, дозволило підвищити швидкість обертання циліндра, збільшити масу та кількість бобін в заправці.

На рис.8.1.1 представлені різні види в'язальних машин. На рис.8.1.1а представлені машини з конструкцією утримувача бобін у вигляді парасольки. На рис.8.1.1б представлені машини з утримувачем бобін, розташованим на підлозі. На рис.8.1.1с представлені структурні схеми лінії заправки нитки, де 1 позначена схема заправки з конструкцією утримувача бобін у вигляді парасольки, а 2 позначена схема заправки з утримувачем бобін, розташованим на підлозі.

Аналіз структурної схеми лінії заправки нитки показує її дуже складну конфігурацію, як у площині так і у просторі габаритних розмірів в'язальної машини. В точках зламу лінії заправки має місце взаємодія нитки з направляючими отворами спрямовувачів нитки, пристроями що забезпечують натяг нитки.

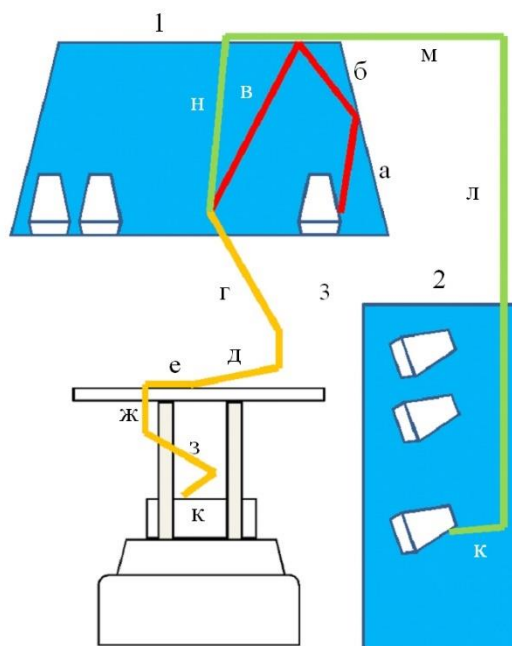
Величина кутів охоплення отворів спрямовувачів нитки та пристроїв, що забезпечують натяг нитки, впливає на величину натягу після цих пристроїв.



а



б



в

Рис.8.1.1. Лінії заправки нитки на круглов'язальних машинах

Для машини з конструкцією утримувача бобін у вигляді парасольки нитка на ділянках *аб*, *бв*, *вг* має злам у отворах спрямовувачів нитки

(червона лінія). На ділянці gd нитка проходить через отвір спрямовувача, нитконатягувальний пристрій та поступає у зону e дії механізму активної подачі. На ділянках $eж$, $жз$, $зк$ нитка має злам у отворах спрямовувачів нитки, після чого вона поступає в зону в'язання.

Для машин з утримувачем бобін, розташованим на підлозі різниця у формі лінії заправки (зелена лінія) полягає у наступному. На ділянках $кл$, $лм$, $мн$ лінія заправки нитки має злами проходячи скрізь отвір у синтетичній трубці, за допомогою якою вона транспортується до в'язальної машини. Після ділянки $нг$ лінія заправки повторює форму, яку вона мала і у першому випадку (лінія жовтого кольору).

Аналіз цієї інформації показує, що задача обрання оптимальної форми лінії заправки є дуже складною, яка залежить від багатьох чинників і її розв'язання не можливе без використання ЕОМ та спеціальних алгоритмів.

Теоретичне обґрунтування обчислювальної схеми алгоритму послідовної оптимізації, що мінімізує пошуки в дереві варіантів

У роботі [11] пропонується обґрунтування алгоритмів розв'язку задачі дискретного програмування, що базується на обчислювальній схемі гілок та границь, що є алгоритмами сортування дерева варіантів поєднаними з, алгоритмами елімінації недопустимих та неоптимальних варіантів, а також алгоритмів розділення дискретних множин, визначених в евклідовому просторі. Показано, що є можливою обчислювальна схема, що не потребує пошуку в дереві варіантів та обчислення границь.

Алгоритми аналізу задач дискретного програмування, що базуються на ідеології гілок і границь та динамічного програмування (на цих обчислювальних схемах базуються майже всі відомі алгоритми за винятком алгоритмів відтинання Гоморрі) є алгоритмами перебору

варіантів. Пошук найбільш перспективних варіантів разом з обчисленням границь є одним з найважливіших факторів, що визначають ефективність алгоритму. Тому представляється доцільним формулювання задачі визначення оптимального розв'язку таким чином, щоб зменшити затрати обчислень для розрахунків значень оцінок цільової функції та позбутись трудомісткої громіздкої операції сортування висячих вершин дерев розв'язків.

Постановка задачі та загальна схема методу.

Розглядається задача мінімізації функції $f_0(x)$ дискретного аргументу

$x, x \in E_n, x^t = \{x_1, \dots, x_j, \dots, x_n\}$, або відшукування

$$x^* = \arg \min f_0(x). \quad (8.1.1)$$

$x \in G$ на скінченій множині G , яка задається таким чином

$$G = Q \cap G_1 \cap G_2 \cap \dots \cap G_i \cap \dots \cap G_m, \quad (8.1.2)$$

$$Q = [x / x_j \in Q_j, Q_j = \{h_{j1}, h_{j2}, \dots, h_{jN}\}], G_i = [x / f_i^H \leq f_i(x) \leq f_i^b]. \quad (8.1.3)$$

На функції $f_i(x)$, $i=0, \dots, m$, накладається наступне додаткове обмеження: j -ті компоненти будь-яких векторів

$$x^{1i}, x^{2i}, x^{li} = \arg \min f_i(x), x^{2i} = \arg \max f_i(x), i = 0, \dots, m, \\ x \in Q.$$

Співвідношення $x \in Q$ приймають одне й те саме значення при будь-яких значеннях інших компонент, $x^{1i}, x^{2i} \in Q$. Такій умові задовольняють, наприклад, монотонні та сепарабельні функції. Для легкої промисловості в такому вигляді, наприклад, формулюються задачі пошуку оптимальної траєкторії заправки нитки. В цьому випадку x_j координата елемента

системи ниткоподачі для j - відрізка траєкторії. Обмеження (8.1.3) гарантують, що кількість елементів системи ниткоподачі буде відповідати необхідній комплектації.

Визначимо множину G^k таким чином

$$G^k = G0^k \cap G, G0^k = [x/ f0^{nk} \leq f0(x) \leq f0^{ek}, x \in Q^k], \quad (8.1.4)$$

де Q^k – деяка підмножина множини Q .

На k - тій ітерації алгоритму визначається деякий елемент $\bar{x}^k, \bar{x}^k \in G^k$, значення функції цілі для якого не може бути покращено за допомогою покоординатного спуску. Назвемо формально задачею k задачу, яка полягає у відшуванні елемента \bar{x}^k , який задовольняє наступним умовам: $x^k \in G^k; \exists x^{*k}, x^{*k} \in G_k$, такого, що усі компоненти x_j^{*k} , крім одного, дорівнюють відповідним компонентам вектору \bar{x}^k , та $f0(x^{*k}) > f0(\bar{x}^k)$.

Запропонований метод розв'язування задачі (8.1.1)-(8.1.3) зводиться до послідовного формування та розв'язування задач $k, k = 1, \dots, r$. Алгоритм розв'язування цієї допоміжної послідовності задач побудований так, що розв'язок задачі $k-1$ виконується для зменшення області G^k . Область G^k зменшується двома способами: зменшується інтервал зміни функції $f0(x)$; в множині Q^k відокремлюють та відкидають підмножини, які мають в собі неприпустимі плани задачі (не задовольняють умові 1), а також плани, які не задовольняють умові 8.1. Зрізання області G^k виконується так, що розв'язування задачі (8.1.1)-(8.1.3), якщо воно існує, належить $G^k, k = 1, \dots, r-1$.

Позначимо: V^{1k} - множина планів $x, x \in Q$, які у ході розв'язку k -тої задачі були відокремлені як неприпустимі,

$$V^{1k} \in \{ * x / x \in Q \setminus G^k \}, V^{-1k} = V^{1,1} \cup \dots \cup V^{1,2} \cup \dots \cup V^{1,k},$$

де $V^{1,2}$ - множина планів, які у ході розв'язку задачі k були відокремлені як не її розв'язок, тобто з

$$x \in V^{2,k}, \text{ слідує } \exists x', x' \in G^k, f_0(x') > f_0(x),$$

$$V^{-2k} = V^{2,1} \cup \dots \cup V^{2,2} \cup \dots \cup V^{2,k}.$$

На основі введених позначень маємо можливість описати процес формування задач k таким чином

$f_0^{H1} = f_0^H$, f_0^H - деяка оцінка мінімального значення функції цілі $f_0(x)$ на множині G

$$f_0^H \leq \min_{x \in G} f_0(x);$$

$$x \in G$$

$f_0^{B1} = f_0^B$, f_0^B - деяка оцінка максимального значення функції цілі $f_0(x)$ на множині G

$$V^{-2k} = V^{2,1} \cup \dots \cup V^{2,2} \cup \dots \cup V^{2,k},$$

$$f_0^B \geq \max_{x \in G} f_0(x), \quad (8.1.5)$$

$$f_0^{Hk} = f_0(x^{-k-1}), f_0^{Bk} = f_0^{B1}, k = 2, \dots, r, \quad)$$

$$Q^1 = Q, Q^k = \{ x / x \in Q, x \notin V^{-1,k-1}, x \notin V^{2,k-1} \}, r = \min k / G^k = \emptyset.$$

Для зручності посилення назвемо лемами наступні твердження.

Лема 1. Існує таке число $k, k < |Q|$, що $G_k = \emptyset$ (тут $|Q|$ - число елементів множини Q).

Лема 8.1. Якщо при $k = 1$, не існує елементу x^{-k} , який задовольняє умовам 1-2, то розв'язок задачі (8.1.1)-(8.1.3) також не існує.

Теорема 3. (Ознака оптимальності) У послідовності розв'язання допоміжних задач $x^{-1}, \dots, x^{-k}, \dots, x^{-\tau-1}$, що вироблюється алгоритмом, елемент $x^{\tau-1}$ є розв'язком задачі (8.1.1)-(8.1.3).

Доведення теореми проведемо від протилежного. Припустимо, що існує вектор $x', x' \in G$, який є оптимальним планом (8.1.1)-(8.1.3) та $f(x') > f(x^{\tau-1})$.

З способу визначення f_0^b та $f_0^{h\tau}$ і останньої нерівності слідує, що $x' \in G \cap G_{op}$ де

$$G_{op} = [x | f_0^{h\tau} \leq f(x) \leq f_0^b, x \in Q] = G_0^\tau \cup V^{-1, \tau-1} \cup V^{-2, \tau-1}.$$

По визначенню $G^\tau = \emptyset$, отже $x' \in V^{-1, \tau-1} \cup V^{-2, \tau-1}$. До цього множина $V^{-1, \tau-1}$ була визначена як множина, що містить тільки ті елементи, що не входять у множину G , отже $x' \in V^{-2, \tau-1}$. Але тоді у відповідності з означенням множини $V^{-2, \tau-1}$ існує елемент $x'', x'' \in G$, такий що $f_0(x'') > f_0(x')$. Ми прийшли до суперечності отже теорема стверджується. Сформульовані твердження - леми 1, 2, теорема 3 встановлюють еквівалентність задачі (8.1.1)-(8.1.3) і послідовність задач k .

Для суттєво дискретних задач алгоритми гілок і границь залишаються найбільш уживаними. Вважається що ці алгоритми базуються на процедурах упорядкування вершин дерева варіантів що висять. Сформульовані твердження можуть бути використані як основа для побудови алгоритмів що не потребують сортування дерева варіантів та обчислення границь.

Обчислювальна реалізація алгоритму послідовної оптимізації, що мінімізує пошуки в дереві варіантів

Пропонується алгоритм розв'язку задачі дискретного програмування[22]. У ході розв'язку задача, яка аналізується, замінюється послідовністю задач пошуку точок, які належать деяким підмножинам множини планів початкової задачі. Описано рекурентний спосіб побудови послідовності розв'язків допоміжних задач. Доведено, що ця скінченна послідовність збігається до розв'язку початкової задачі[11, 12].

Для задач знаходження мінімуму нелінійної функції що визначена на деякій дискретній множині елементів евклідового простору таких, що задовольняють системі нелінійних обмежень найбільш поширеними є обчислювальні схеми на основі схеми гілок і границь. В цьому випадку методи гілок та границь є алгоритмами сортування дерева варіантів що є поєднаними з алгоритмами елімінації недопустимих та неоптимальних варіантів та а також алгоритмах розділення дискретних множин, визначених в евклідовому просторі.

В підрозділі 8.1.1 встановлена еквівалентність задачі максимізації функції $f_0(x)$ дискретного аргументу та послідовністю задач ($k=1, \dots$), які полягають у відшуканні елементу \bar{x}^k , що задовольняє наступним умовам: $x^k \in G^k$; $\exists \bar{x}^{*k}$, $x^{*k} \in G_k$, такого, що усі компоненти x_j^{*k} , крім одного, дорівнюють відповідним компонентам вектора \bar{x}^k , та $f_0(x^{*k}) > f_0(\bar{x}^k)$.

Алгоритм розв'язування задач k побудований на послідовному розбитті початкової множини Q на підмножини Q^s

$$Q^s = [x/x_j \in Q_j^s, Q_j^s = \{h^s_{j1}, h^s_{j2}, \dots, h^s_{jN_s}\}, Q_j^s \in Q_j], \quad (8.1.6)$$

та на відокремленні у підмножини з максимальним індексом S елементів, які не можуть бути розв'язком задачі k , яка аналізується. У ході аналізу задачі k виконуються наступні операції.

1. Розгалуження. Якщо існує індекс j , такий, що $h^S_{j1} \neq h^S_{Ns}$, множина Q^S ділиться на дві підмножини Q^{s1} та Q^{s2} таким чином

$$Q^{s1} = [x/x \in Q^S, h^S_{j1} \leq x_j \leq h^S_{j2}],$$

$$Q^{s2} = [x/x \in Q^S, h^S_{j,l+1} \leq x_j \leq h^S_{j,Ns}].$$

Розбиттю підлягає множина з максимальним індексом s , $s = \check{s}$. Після того як розгалуження проведено, вважаємо $s1 = \check{s}$ $s2 = \check{s} + 1$.

Надалі будемо вважати, що множина $Q^{sj'}$ впорядкована по зростанню значення функції $f_0(x)$, тобто з $l_1 < l_2$, для будь яких x^1, x^2 що задовольняють умові

$$x^1_{j'} = h^S_{j',l_1}, x^2_{j'} = h^S_{j',l_2}, x^1_j = x^2_j, j = 1, \dots, j'-1, j+1, \dots, n,$$

тобто

$$f_0(x^1) > f_0(x^2). \quad (8.1.7)$$

Введена умова надає додаткове направлення процесу розгалуження.

Для $x^1, x^1 \in Q^{s1}$, та $x^2, x^2 \in Q^{s2}$ нерівність (8.1.7) можна розглядати як співвідношення між грубими оцінками функції цілі. У процесі роботи алгоритму розгалуженню підлягає множина Q^{s2} з максимальною оцінкою.

8.1. Виділення планів, які не можуть бути розв'язком задачі k . Не дивлячись на велике розмаїття схем послідовного аналізу варіантів, загальним для них є наявність операторів, які забезпечують виключення з подальшого розгляду підмножин планів, що не мають в собі розв'язку задачі (8.1.1). Наявність таких процедур дозволяє, в деяких випадках, суттєво, скоротити число варіантів, які аналізуються, і, отже, підвищити,

ефективність розрахунків. Алгоритм, що пропонується передбачає застосування двох операторів α_1, α_2 , перший з яких не дозволяє виключити з розглядання варіанти які не задовольняють умові 1, другий - варіанти, що не задовольняють умові 8.1.

Нехай $\Phi_s, \Phi_s = \{s+1, s+2, \dots\}$ множина індексів, що не відповідають підмножинам, які не можуть бути отримані з Q^s в результаті розгалуження. Оператори α ($\alpha = \alpha_1$ або α_2) мають такі властивості:

$$1. \quad \alpha(Q_s) \subseteq Q_s$$

$$8.1. \quad (\alpha(Q_s)) = \alpha(Q_s)$$

$$3. \quad \text{Якщо } Q_s \cap G = \emptyset, \text{ то } \exists j, j \in \Phi_s, Q_j \neq \emptyset, \alpha_1(Q_j) = \emptyset.$$

Послідовне застосування операторів розгалуження та відокремлення з максимальним індексом S підмножин, які не мають в собі розв'язку задачі k , дозволяє не деякому кроці розгалуження отримати множину Q^S , $Q^S = \alpha(Q^S)$, яка або має тільки один елемент, або пуста. В останньому випадку можливі ситуації:

$S > 1$ в такому випадку необхідно перейти до альтернативної множини $S-1$,

$S = 1$, в такому випадку розв'язок задачі k не існує (справедливість цього твердження прямує з визначення операторів α_1 та α_2), $G_k = \emptyset$.

Розглянемо більш уважно ситуацію, для якої множина Q^S має тільки один елемент x^S . Справедливо наступне твердження.

Лема 4. (Ознака оптимальності задач k). Нехай $s' = \min s \mid Q^s = x^s$. В такому разі $x^{s'}$ є розв'язком задачі k .

З властивості 3 оператора α_1 , виходить що $x^{s'} \in G$ і, отже задовольняє умові 1 для розв'язку задачі k . Покажемо, що $x^{s'}$ задовольняє також умові

8.1. Доведення проведемо від протилежного. Нехай існує план x' , $x' \in G^k$, такий що $x_j^{s'} = x_j^s$, $j=1, \dots, j'-1, \dots, j'+1, \dots, n, j \neq j'$, та $f_0(x') > f_0(x^s)$. З способу побудови послідовності множин Q^s прямує існування індексу s'' такого, що

$$s'' = \min s / x^{s'}, x' \in Q^s \cup Q^{s-1} \cup V^s, x^s \in Q^s, x' \notin Q^s,$$

де $V^{s''}$ - множина елементів, які не були відокремлені у множині $Q^{s''}$, $Q^{s''} = \alpha_2(\alpha_1(\bar{Q}^{s''}))$. У відповідність алгоритму множини $Q^{s''-1}$ та $\bar{Q}^{s''}$ отримано в результаті розгалуження множини $\check{Q}^{s''}$ $\check{Q}^{s''} = \bar{Q}^{s''} \cup Q^{s''-1}$.

З означення операторів α_1 та α_2 прямує, що $x' \in V^s$. Тобто $x' \in Q^{s''-1}$.

Але це неможливо, так як в такому випадку не виконується умова (8.1.7).

Ми прийшли до суперечності, отже, твердження виконується.

3. Оператори α_1 та α_2 . Раніше відмічалось, що розрахункові процедури, що не дозволяють виключати з множини варіантів що розглядаються, варіанти, які не можуть бути розв'язком задачі, є невід'ємним атрибутом багатьох розрахункових схем послідовної оптимізації. Наприклад, у роботі [12] наведено ряд тестів відокремлення неприпустимих варіантів, які можуть бути використані при побудові оператора α_1 . Наведені нижче розрахункові процедури оператора α_1 та α_2 є модифікацією, що застосовується до задач k , алгоритмів відокремлення підмножин, що не мають припустимих оптимальних планів [12].

Позначимо

$$x^{i \min} (Q^s | \bar{x}^{j'}) = \arg \min f_i(x_1, \dots, x_{j'} = \bar{x}^{j'}, \dots, x_n),$$

$$x_j \in Q_j^s, j \in J(j'),$$

$$x^{i \max} (Q^s | \bar{x}^{j'}) = \arg \max f_i(x_1, \dots, x_{j'} = \bar{x}^{j'}, \dots, x_n),$$

$$x_j \in Q_j^s, j \in J(j'),$$

де $J(j') = \{1, \dots, j, \dots, j'-1, j'+1, \dots, n\}$.

Оператор $\bar{\omega}_1$ визначимо наступним чином

$$\bar{\omega}_1(\bar{Q}^s) = \{x / x_j \in \bar{Q}_j^s \setminus \bar{V}_j^s\},$$

де $\bar{V}_j^s = \{x_j / f_i(x^{i \max}(\bar{Q}^s / x_j)) < f_i^H\}$, або

$$f_i(x^{i \min}(\bar{Q}^s / x_j)) > f_i^B, \exists i = 0, \dots, m\}. \quad (8.1.8)$$

Розрахункова процедура, яку реалізує оператор $\bar{\omega}_1$, полягає в послідовному використанні до \bar{Q}^s оператору $\bar{\omega}_1$ до тих пір, поки не буде отримана множина Q^s така, що

$$\bar{\omega}_1(Q^s) = Q^s.$$

Безпосередньою перевіркою легко переконатись в тому, визначений таким чином оператор $\bar{\omega}_1$ задовольняє властивостям 1-3, а з (8.1.8) маємо

$$\bar{Q}^s \setminus Q^s \not\subset G^k.$$

Позначимо: $\theta_j(\tilde{Q}^s)$ множина індексів елементів множини \tilde{Q}_j^s , яка не задовольняє наступній умові

$$\theta_j(\tilde{Q}^s) = \{l / f_i(x^{i \max}(\tilde{Q}^s / x_j = h_{jl})) \leq f_i^B$$

$$f_i(x^{i \max}(\tilde{Q}^s / x_j = h_{jl})) \leq f_i^H, \forall i, i = 0, \dots, m\},$$

позначимо також $\tilde{l} = \max l / l \in \theta_j(\tilde{Q}^s)$.

Оператор $\bar{\omega}_2$ позначимо так

$$\bar{\omega}_2(\bar{Q}^s) = \{x / x_j \in \bar{Q}_j^s \setminus \bar{V}_j^s\},$$

де $\bar{V}_j^s = \{h_{jl}^s / l < \tilde{l}\}$.

Також як і оператор α_1 оператор α_2 стоїть у послідовному використанні до множини \tilde{Q}^S оператора $\bar{\alpha}_1$ до тих пір поки не буде отримана множина Q^S така, що

$$\bar{\alpha}_2(Q^S) = Q^S.$$

Справедливим є наступне твердження - з $\tilde{x} \in Q^S \setminus \alpha_2(Q^S)$ $\tilde{x} \in G^k$ випливає, що \tilde{x} не задовольняє умові 8.1.

Дійсно, що з того що $\tilde{x} \in Q^S \setminus \alpha_2(Q^S)$ слідує існування такої компоненти \tilde{x}_j , плану \tilde{x} , що $\tilde{x}_j = h_{j',l}, l < \tilde{l}$. Розглянемо вектор $\bar{x} = \bar{x}_j = \tilde{x}_j, j = 1, \dots, j'-1, j'+1, \dots, n, \bar{x}_j = h_{j',l}$. Так як раніше прийнято припущення, що будь яка множина Q_j^S впорядкована у порядку зростання функції $f_0(x)$ тобто $f_0(x) > f_0(x)$. З (9) випливає, що будь який $x \in Q^S$ $\bar{x}_j = h_{j',l}$, задовольняє системі умов (4) і, отже, $\bar{x} \in G^k$ Це свідчить про те що \tilde{x} не задовольняє умові 8.1.

Перш за все, треба виділити, що основний об'єм роботи розрахунків, необхідний для реалізації запропонованого алгоритму, робиться операторами розгалуження та відокремлення планів задачі, які не можуть бути її розв'язком. Застосування різноманітних тестів відбору варіантів, які є припустимими дозволяє суттєво підвищити ефективність алгоритму типу "тілки та границі" та послідовного аналізу варіантів [11, 12]. Однак, поряд з операторами, які реалізують відбір неприпустимих варіантів, алгоритми "тілок та границь" та послідовного аналізу варіантів потребують розрахунків оцінок функцій цілі для кожної підмножини, що аналізується, та пошуку серед висячих вершин дерев розв'язків вершини з максимальною оцінкою функцій цілі. Алгоритм, що пропонується,

передбачає розрахунок оцінок значень функцій цілі тільки 2 рази, а сортування висячих вершин дерев розв'язків не потребує зовсім.

Задача синтезу системи подачі нитки на круглов'язальних машинах для випадку перешкод у вигляді вертикальних ліній

При проектуванні апаратів легкої промисловості виникає завдання визначення траєкторії нитки так, щоб сумарна сила тертя, яка діє на нитку, була мінімальною. Початкові умови завдання містять координати початкової точки (джерела) траєкторії, кінцевої точки (цілі) і координати перешкод, мимо яких повинна проходити нитку. Якщо траєкторія огинає перешкоду, виникає сила тертя, яка залежить від кута обхвату тієї, що направляє. Таким чином, завдання полягає у визначенні траєкторії з мінімальною сумою кутів обхвату ниткою тієї, що направляє.

Відрізками найбільшої товщини зображені перешкоди, відрізки середньої товщини відображають одну з можливих траєкторій нитки. Точки s і t відображають відповідно джерело і ціль. Кожною перешкодою p_i є вертикальний відрізок і характеризується позицією точок – верхньою u_i і нижньою l_i . Приведена на рис.8.1.2 траєкторія $T = (s, u_2, l_3, l_5, t)$ змінює напрям руху нитки в точках u_2, l_3, l_5 . Кути зміни напрямку відповідно складають α, β, γ , а оцінка цієї траєкторії, або значення цільової функції на ній складає

$$F(t) = \alpha + \beta + \gamma. \quad (8.1.9)$$

Надалі використовуватимемо кут нахилу відрізка траєкторії, під яким розумітимемо кут між направленим відрізком траєкторії у напрямі руху нитки і віссю X . При цьому враховуватимемо знак кута. Так, наприклад, відрізок (s, u_2) має позитивний кут нахилу, а (u_2, l_3) – негативний.

Відмітимо, що при обгинанні перешкоди у верхній точці кут нахилу суміжних відрізків траєкторії зменшується, а в нижній – зростає.

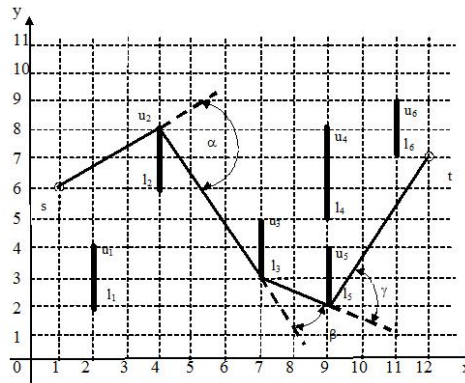


Рис.8.1.8.1. Приклад початкових умов завдання і допустимої траєкторії

Для визначення оптимальної траєкторії нитки пропонується скористатися моделлю у вигляді орієнтованого графа $G = (V, E)$, де $V = \{v_1, v_2 \dots v_{2n+2}\}$ – множина вершин, $E = \{e_1, e_2 \dots e_m\}$ – множина ребер. Одна вершина v_s графа G відповідає джерелу, друга вершина v_t – цілі, а всі інші – кінцевим точкам перешкод. Граф G містить ребро (v_i, v_j) , якщо можливе просування нитки від точки, яка відображається вершиною v_i до точки, яка відображається вершиною v_j . Граф G для початкових умов, показаних на рис.8.1.2, представлений на рис. 8.1.3. Для вершин графа застосовані позначення відповідних точок, а ребра зображені тонкими відрізками прямих. На малюнку навмисне залишені перешкоди (товсті відрізки), щоб наочно показати умови, яким задовольняють ребра графа G :

- 1) ребра не перетинають перешкод;
- 2) для кожного ребра, що виходить з верхньої вершини u_i , існує ребро e_l , що входить в u_i , і кут нахилу ребра не більш ніж кут нахилу e_i ;
- 3) для кожного ребра, що виходить з нижньої вершини u_i , існує ребро e_l , що входить в u_i , і кут нахилу ребра не менше ніж кут нахилу e_l .

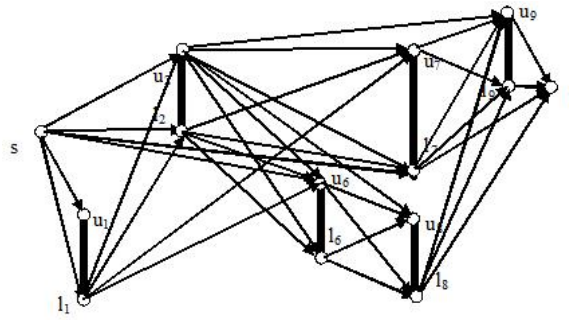


Рис.8.1.3. Граф G для початкових умов

Завдання пошуку шляху з вершини v_s у вершину v_t в графі G належить до класу завдань пошуку оптимального шляху графа. Відповідно до загальної методики вирішення завдань представленого класу необхідно визначити клас, до якого належить цільова функція. Приклад, приведений на рис.8.1.4, показує, що в завданні, яке розглядається, цільова функція належить до γ -класу. Дійсно, оптимальним (s, t) -шляхом для представленого графа є шлях $r = (s, u_1, u_2, t)$, якому відповідає значення цільової функції $F(r) = \pi/8.1$. Разом з тим, підшлях цього шляху $q = (s, u_1, u_2)$ із значенням цільової функції $F(q)$ не є оптимальним, оскільки існує інший шлях $w = (s, l_1, u_2)$ з меншим значенням цільової функції $F(w) < F(q)$, а для шляху $z = (s, l_1, u_2, t)$ маємо $F(z) > 2\pi$.

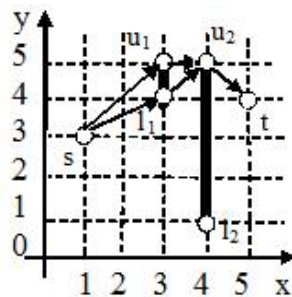


Рис.8.1.4. Визначення класу цільової функції

Як наголошувалося раніше, при цільовій функції, що належить до γ -класу, в процесі пошуку оптимального (s, t) -шляху треба запам'ятовувати всі шляхи в проміжні вершини, які не домінуються іншими шляхами. Це приводить до факторіальної, або експоненціальною, оцінці

обчислювальної і асимптотичній складності алгоритму. В той же час, як буде показано далі, облік специфічних властивостей представленого завдання дозволяє побудувати алгоритм з поліноміальною оцінкою.

Очевидною властивістю графа G є спрямованість кожного ребра (v_i, v_j) від вершини v_i з меншим значенням координати X до вершини з великим значенням цієї координати. X -координата вершини v_s має найменше значення, а v_t - найбільше.

Значення цільової функції на довільному шляху $p = (v_s, v_1, v_2, \dots, v_i, \dots, v_{k-1}, v_k)$ визначається як

$$F(p) = \sum_{i=1}^{k-1} |\alpha_i|,$$

де α_i - кут між ребрами шляху p , які інцидентні вершині v_i .

Тому при доповненні шляху p новим ребром (v_k, v_{k+1}) ($p' = p \cup (v_k, v_{k+1})$) значення цільової функції збільшується на значення кута між ребрами (v_{k-1}, v_k) і (v_k, v_{k+1})

$$F(p') = F(p) + |\alpha_k|.$$

Отже, приріст цільової функції при доповненні шляху ребром однозначно визначається координатами нової точки v_{k+1} , координатами останньої точки v_k і кутом нахилу останнього ребра (v_{k-1}, v_k) шляху p .

Припустимо, існує інший шлях $p' = (v_s, v'_1, v'_2, \dots, v'_i, \dots, v'_r, v_k)$, якому відповідає значення цільової функції $F(p')$, і $p' \cup (v_k, v_{k+1})$. Тоді, якщо

$$F(p) + |\alpha_k| \leq F(p') + |\alpha'_k|, \quad (8.1.10)$$

де α'_k - кут між ребрами (v'_r, v_k) і (v_k, v_{k+1}) .

Будь-яке доповнення шляху p шляхом q має не більше значення цільової функції, чим доповнення p' - шляхом q . Це означає, що набір параметрів Q , якими визначається домінування шляхів, для

представленого завдання містить два елементи: значення цільової функції і останнє ребро.

Кожне ребро (v_k, v_{k+1}) можна розглядати як останнє ребро (v_s, v_{k+1}) - шляху. Враховуючи (8.1.10), ребру (v_k, v_{k+1}) доцільно поставити у відповідність мінімальне значення цільової функції (v_s, v_{k+1}) - шляху, який закінчується цим ребром. Позначимо це значення як $f(v_k, v_{k+1})$. Якщо відомі значення функції f на всіх ребрах, які входять у вершину v_k , то $f(v_k, v_{k+1})$ визначається наступним шляхом

$$f(v_k, v_{k+1}) = \min\{f(v_r, v_k) + \alpha((v_r, v_k), (v_k, v_{k+1}))\}, \forall (v_r, v_k) \in E, \quad (8.1.11)$$

де $\alpha((v_r, v_k), (v_k, v_{k+1}))$ – кут між ребрами (v_r, v_k) і (v_k, v_{k+1}) .

Співвідношення (8.1.11) зв'язує значення цільової функції на довгому шляху (траєкторії) і значення цільової функції на менш коротких шляхах і таким чином є рівняннями Белмана для представленого завдання. Оскільки на ребрах, які виходять з вершини v_s функція f приймає нульове значення, починаючи з вершини v_s і розглядаючи вершини графа G по порядку зростання координати X , можна визначити значення f на всіх ребрах графа G . Тоді оптимальне значення цільової функції F_{min} на (s, t) - шляху визначається як

$$F_{min} = \min\{f(v_r, v_t)\}, \forall (v_r, v_t) \in E.$$

Алгоритм пошуку оптимальної траєкторії нитки складається з двох етапів: 1) побудова графа G і 2) визначення ребра, інцидентного вершині vt , з мінімальним значенням функції f . Оскільки на обох етапах треба розглядати кінцеві точки перешкод в порядку зростання координати X , етапи виконуються не послідовно, а паралельно. Покроковий опис алгоритму має такий вигляд.

1. Упорядкувати перешкоди по неспаданню координати X , а перешкоди з однакою координатою X упорядкувати по неспаданню координати Y верхньої точки і помістити впорядковану сукупність перешкод в список L . Доповнити L двома виродженими перешкодами, які складаються з однієї точки - джерелом і ціллю. Джерело – перший елемент списку, а ціль – останній.

2. Для кожної вершини v , яка належить L , сформувати порожній список $I(v)$ вхідних ребер; покласти $H(v_s) = 2 * NP \pi$ (NP – кількість перешкод) для всіх вершин окрім v_s . Покласти $H(v_s) = 0$ і $R = 2 * NP \pi$.

3. Вилучити перший елемент-вершину списку L і запам'ятати цю вершину v .

4. Якщо $v = v_t$, перейти до кроку 18.1.

5. Якщо $H(v_L) \geq R$, перейти до кроку 3.

6. Для вершини v_L визначити вершину v_N таку, що відрізок (v_L, v_N) розглядається вперше і не перетинає ні єдину перешкоду. Якщо така вершина не існує, перейти до кроку 3.

7. Якщо список $I(v_L)$ порожній, додати ребро (v_L, v_N) до графа G , покласти $f(v_L, v_N) = 0$ і $H(v_L) = 0$, внести (v_L, v_N) до списку $I(v_N)$ і перейти до кроку 6. Інакше покласти $f(v_L, v_N) = 2 * NP \pi$ і встановити покажчик q поточного вхідного ребра на перший елемент списку $I(v_L)$.

8. Вибрати поточний елемент (v_P, v_L) із списку $I(v_L)$, і якщо для ребер (v_P, v_L) і (v_L, v_N) виконується умова 2) або 3), а також $f(v_P, v_L) + \alpha((v_P, v_L), (v_L, v_N)) < f(v_L, v_N)$, виконати наступні дії:

8.а) покласти $f(v_L, v_N) = f(v_P, v_L) + \alpha((v_P, v_L), (v_L, v_N))$;

8.б) запам'ятати вершину v_P , яка використовувалася при визначенні (v_L, v_N) , – $P((v_L, v_N)) = v_P$;

8.в) якщо $f(v_L, v_N) < H(v_N)$, покласти $H(v_N) = f(v_L, v_N)$.

9. Якщо існує наступний елемент списку $I(v_L)$, встановити на нього покажчик q і перейти до кроку 8.
10. Якщо $f(v_L, v_N) = 2 * NP \pi$, перейти до кроку 6.
11. Якщо $v_N = v_t$ і $f(v_L, v_N) < R$, покласти $R = f(v_L, v_N)$ і запам'ятати передостанню вершину шуканого шляху – $v_f = v_L$. Перейти до кроку 6.
12. Визначити ребро шуканого шляху $e = (v_f, v_t)$.
13. Якщо $v_f = v_s$ - кінець алгоритму. Інакше покласти $v = P((v_f, v_t))$, $v_t = v_f$, $v_f = v$ і перейти до кроку 18.1.

Зробимо зауваження, які відносяться до приведеного алгоритму. Оскільки в кожній точці значення кута зміни напрямку не перевищує π , а кількість точок дорівнює подвійній кількості перешкод, то величина $2 * NP \pi$ є верхньою оцінкою цільової функції на будь-якому шляху. Функція $H(v)$ приймає поточне мінімальне значення цільової функції на (v_s, v) -шляху і після аналізу всіх ребер, які входять у вершину v , дорівнює мінімальній вартості досягнення v .

На кроці 6 для поточної вершини v_L визначаються всі початкові ребра. Для визначення кожного ребра розглядається кожна вершина v , у якої координата X більше, ніж в v_L . Якщо ребро (v_L, v) задовольняє умові 2) або 3), перевіряються перетини цього ребра зі всіма перешкодами, що мають координату X більшу, ніж у вершини v_L і меншу, ніж у вершини v . При цьому вважається, що ребро (v_L, v) перетинає перешкода $p = (v_u, v_l)$, якщо пряма, проведена через точки v_L, v , містить точку з координатами $(X(p), Y)$, де $X(p)$ – X -координата перешкоди p , і $Y(v_l) < Y < Y(v_u)$ ($Y(v_l), Y(v_u)$ – координати точок v_l, v_u відповідно). У разі відсутності перетинів (v_L, v) включається в множину E ребер графу G . Багаторазове виконання кроків 12, 13 забезпечує прохід від цілі до джерела за допомогою функції $P(e)$, тобто по ребрах, що створюють оптимальний (s, t) -шлях.

Проаналізуємо трудомісткість алгоритму. Перший крок впорядкування перешкод вимагає кількості операцій, яка складає $O(NPlgNP)$. Кількість операцій на другому кроці - $O(NP)$. Кроки 3-11 входять в цикл, кількість повторень якого дорівнює кількості вершин графа G , тобто складає $O(NP)$.

На кожній ітерації цього циклу на кроці 6 здійснюється аналіз кожної вершини v_N , X -координата якої більше X -координати v_L . Кількість таких вершин - $O(NP)$. Оскільки для кожної вершини v_N при визначенні $f((v_L, v_N))$ розглядаються всі ребра, які входять у вершину v_L , кількість яких також $O(NP)$, кількість операцій для визначення $f((v_L, v_N))$ складає $O(NP)$.

Для перевірки перетинів ребра (v_L, v_N) з перешкодами на кроці 6 аналізуються всі перешкоди, розташовані лівіше v_N і правіше v_L . Кількість таких перешкод буде рівна $O(NP)$ і оскільки визначення $f((v_L, v_N))$ і перевірка перетинів виконуються послідовно, кількість операцій обробки однієї вершини складає $O(NP)$. Тому кількість операцій обробки однієї вершини v_L складає $O(NP^2)$, а всіх вершин списку L - $O(NP^3)$.

Кроки 12, 13 очевидно вимагають $O(NP)$ операцій. Отже, загальна асимптотична обчислювальна складність алгоритму складає $O(NP^3)$.

Оцінимо витрати пам'яті, або асимптотичну ємкісну складність алгоритму. Для збереження списку L витрати пам'яті складають $O(NP)$. Оскільки для кожної вершини треба пам'ятати список вхідних ребер, кількість яких дорівнює $O(NP)$, загальна асимптотична ємкість складності алгоритму - $O(NP^2)$.

Розглянемо приклад застосування алгоритму до вхідних умов, показаних на рис.8.1.8.1. Після завершення аналізу вершини-джерела, побудована частина графа G , яка показана на рис.8.1.5.

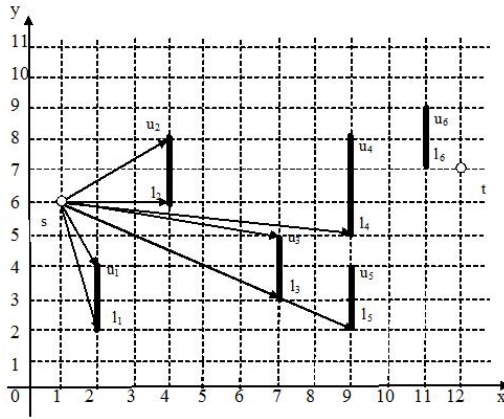


Рис.8.1.5. Частина графу з вершиною-джерела

Значення функції f для всіх ребер на рис.8.1.5 дорівнює нулю. Після розкриття вершин з координатою $X \leq 7$ отримуємо ситуацію, зображену на рис.8.1.6.

Як показано на рис.8.1.6, при розкритті вершини l_3 досягається ціль із значенням цільової функції $F = \arctg(4/5) + \arctg(1/2) \approx 1,14$. При цьому $R \approx 1,14$ і $P((l_3, t)) = s$. На наступному етапі маємо $H(u_6) = \arctg(2/3) - \arctg(1/7)$, $H(l_6) = \arctg(2/3) + \arctg(1/2)$, $R = H(t) = \arctg(1/8) + \arctg(2/3) \approx 0,71$. Оскільки $H(l_6) > R$, вершина l_6 в подальшому не розкривається. Розкриття вершин u_6 і l_5 показує, що набутти меншого значення R неможливо. Тому шлях (s, l_4, t) – оптимальний.

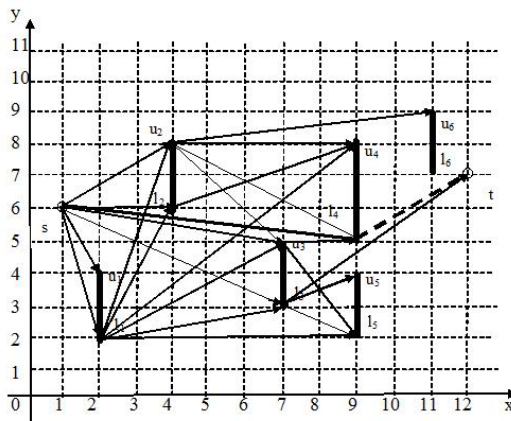


Рис.8.1.6. Кінцевий вигляд оптимальної траєкторії нитки

Викладені результати можуть бути застосовані при формах перешкод, які відрізняються від вертикальних відрізків. Наприклад, якщо перешкодами є кола різних радіусів. В цьому випадку траєкторія нитки складатиметься з ділянок дотичних до двох кіл і відповідних дуг. Загальна схема алгоритму при цьому, в цілому, зберігається і відрізняється від описаної вище схеми технічними деталями.

На рис.8.1.7а та 8.1.7б показаний фрагмент реалізації програмного забезпечення, яке приведене в додатку Б.

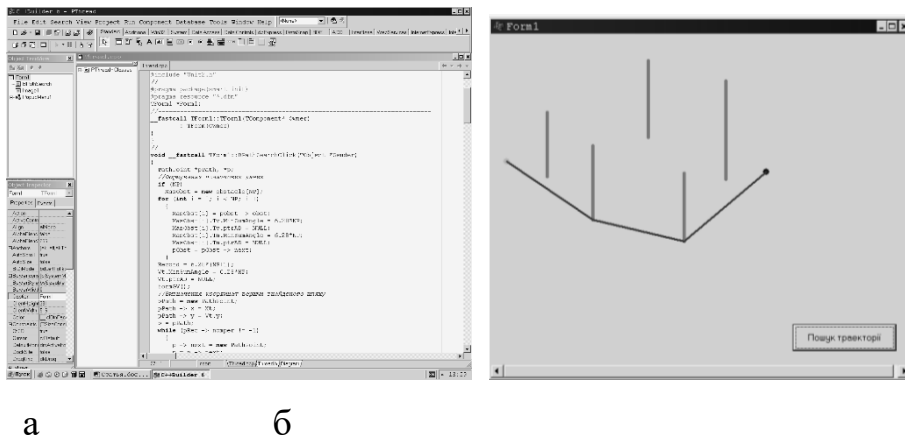


Рис.8.1.7. Програмна реалізація алгоритму

Оптимізація плоскої траєкторії нитки в круглов'язальних машинах у випадку перешкод у вигляді кіл

В даному підрозділі розглядається задача визначення траєкторії нитки у системі подачі нитки круглов'язальних машин. Дана траєкторія являє собою послідовність відрізків прямої та дуг, що з'єднують вузол (джерело) первісного подавання нитки із вузлом (ціллю) кінцевого приймання. За умови, що траєкторія нитки лежить у одній площині, джерело, ціль та перешкоди представляються колами різних радіусів. Тоді прямолінійні ділянки нитки представляються відрізками дотичних до двох відповідних кіл, а ділянки, що обгинають перешкоди, - дугами відповідних радіусів.

Запропоновано алгоритм визначення траєкторії, за якою сумарна сила тертя, що діє на нитку, набуває мінімального значення[121, 129].

При проектуванні системи подачі нитки на круглов'язальних машинах виникає задача визначення траєкторії нитки таким чином, щоб сумарна сила тертя, що діє на нитку, була мінімальною. Вихідні умови задачі містять координати джерела, цілі та перешкоди, які являють собою кола поданих радіусів. При обгинанні ниткою кожної перешкоди виникає сила тертя, пропорційна куту зміни напрямку руху [4]. Тому представляється доцільним формулювання задачі визначення оптимальної траєкторії нитки як задачі пошуку траєкторії з мінімальною сумою кутів зміни напрямку руху. Предметом дослідження в поданій роботі є властивості запропонованої цільової функції і побудова алгоритму розв'язання сформульованої задачі.

Нитка, траєкторію якої треба визначити, умовно розглядається як абсолютно гнучка, нерозтяжна та невагома. Траєкторія нитки складається з сукупності дотичних до кіл і дуг. Приклад вихідних умов і можливої траєкторії показаний на рис.8.1.8, де джерело має позначення s , ціль - t , а перешкоди - 1 і 2. Отже, подані координати центрів і радіуси кіл s , t , 1 і 2. Серед усіх можливих траєкторій треба знайти таку, якій відповідає мінімальна сума кутів між усіма послідовними відрізками. Для траєкторії на рис.8.1.8 цільова функція дорівнює сумі двох кутів: між дотичними до кіл s , 1 та 1, 2 і 1, 2 та 2, t .

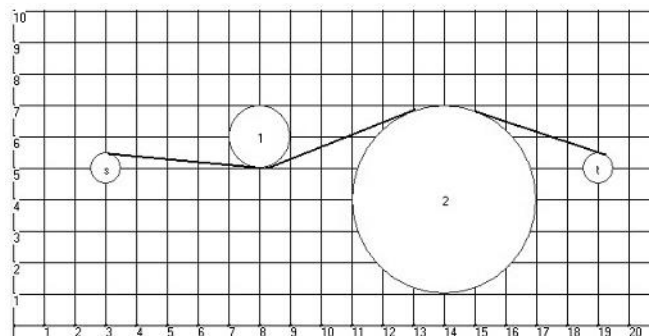


Рис.8.1.8. Вихідні умови задачі

Розв'язок задачі пошуку оптимальної траєкторії можна отримати, скориставшись моделлю у вигляді неорієнтованого графа $G = (V, E)$

де V – множина вершин,

E – множина ребер.

Множина вершин V складається з вершини s , яка відображає джерело, вершини t , яка відображає ціль, та сукупності вершин, які взаємно однозначно відповідають перешкодам. Множина ребер E взаємно однозначно відповідає відрізкам дотичних, що не перетинають перешкод.

Кожній перешкоді v ставиться у відповідність дві вершини графа – при огинанні ліворуч v_1 і при огинанні праворуч $v_{8.1}$. Відрізки шляху задовольняють наступним умовам:

1. Для кожного ребра e_k , що входить у вершину v_i зліва, існує ребро e_1 , що виходить з v_i і кут нахилу ребра e_k не більший ніж кут нахилу ребра e_1 . Для кожного ребра e_k , що входить у вершину v_i справа, існує ребро e_1 , що виходить з v_i і кут нахилу ребра e_k не менший ніж кут нахилу ребра e_1 . Тобто при огинанні перешкоди зліва кут нахилу суміжних відрізків траєкторії спадає, а справа – зростає.
2. Ребра не перетинають перешкод.

Задача пошуку шляху з вершини s у вершину t у графі G належить до класу задач пошуку оптимального шляху графа [19, 21]. Як показано в [22], у задачі, що розглядається, цільова функція належить до γ - класу. Це видно з того, що оптимальний шлях може містити підшлях, який не є оптимальним. Приклад оптимального шляху, що містить неоптимальний підшлях показано на рис.8.1.9.

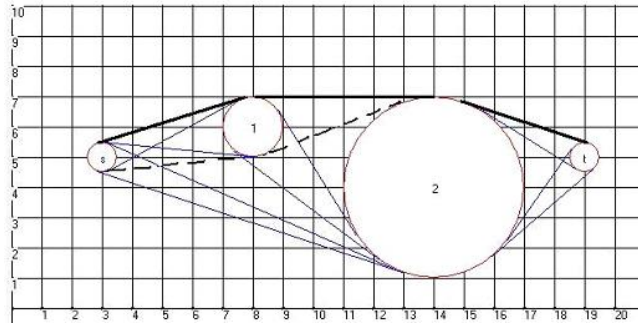


Рис.8.1.9. Шлях, що містить неоптимальний підшлях

Тому, у процесі пошуку оптимального (s,t) - шляху треба запам'ятовувати усі шляхи у проміжні вершини, які не домінуються іншими шляхами. Значення цільової функції на довільному шляху $p=(s, v_1, v_2, \dots, v_i, \dots, v_{k-1}, v_k)$ визначається таким чином

$$F(p) = \sum_{i=1}^{k-1} |\alpha_i|, \quad (8.1.12)$$

де α_i – кут між ребрами шляху p , що інцидентні вершині v_i .

Тому при доповненні шляху p новим ребром (v_k, v_{k+1}) , $(p_1 = p \cup (v_k, v_{k+1}))$ значення цільової функції збільшується на значення кута між ребрами (v_{k-1}, v_k) і (v_k, v_{k+1}) .

$$F(p_1) = F(p) + |\alpha_k|. \quad (8.1.13)$$

Отже, приріст цільової функції при доповненні шляху ребром однозначно визначається координатами нової вершини v_{k+1} , координатами останньої вершини v_k і кутом нахилу останнього ребра (v_{k-1}, v_k) шляху p .

Припустимо, існує інший шлях $p'=(s, v_1', v_2', \dots, v_i', \dots, v_{k-1}', v_k)$, якому відповідає значення цільової функції $F(p')$, і $p_1' = p' \cup (v_k, v_{k+1})$. Тоді, якщо

$$F(p) + |\alpha_k| \leq F(p') + |\alpha_k'|, \quad (8.1.14)$$

де α_k' – кут між ребрами (v_{k-1}', v_k) і (v_k, v_{k+1}) , будь-яке доповнення шляху p_1 шляхом q має не більше значення цільової функції, ніж доповнення p_1' шляхом q .

Це означає, що набір параметрів Q , за яким визначається домінування шляхів, для поданої задачі містить два елементи: значення цільової функції і останнє ребро.

Кожне ребро (v_k, v_{k+1}) можна розглядати як останнє ребро (s, v_{k+1}) - шляху. Враховуючи (8.1.14), ребру (v_k, v_{k+1}) доцільно поставити у відповідність мінімальне значення цільової функції на (s, v_{k+1}) - шляху, що закінчується цим ребром. Позначимо це значення як $f(v_k, v_{k+1})$. Якщо відомі значення функції f на всіх ребрах, що входять у вершину v_k , то $f(v_k, v_{k+1})$ визначається таким чином

$$f(v_k, v_{k+1}) = \min\{f(v_r, v_k) + \alpha((v_r, v_k), (v_k, v_{k+1}))\} \forall (v_r, v_k) \in E, \quad (8.1.15)$$

де $\alpha((v_r, v_k), (v_k, v_{k+1}))$ – кут між ребрами (v_r, v_k) і (v_k, v_{k+1}) .

Співвідношення (8.1.15) зв'язує значення цільової функції на довільному шляху та значення цільової функції на коротших шляхах. Оскільки на ребрах, що виходять з вершини s , функція f приймає нульове значення, починаючи з вершини s і розглядаючи вершини графа G в певній послідовності, можна визначити значення f на всіх ребрах графа G . Тоді оптимальне значення цільової функції F_{min} на (s, t) - шляху визначається як

$$F_{min} = \min\{f(v_r, v_t)\} \forall (v_r, v_t) \in E \quad (8.1.16)$$

Будь-який підшлях (s, v_i) оптимального шляху (s, t) при заданому куті нахилу ребра (v_{i-1}, v_i) оптимальний. Тому для кожної досягнутої вершини v_i в процесі пошуку достатньо запам'ятовувати оцінку ребра (v_{i-1}, v_i) і його кут нахилу.

В [19, 22] запропоновано застосування розглянутого алгоритму для розв'язання задачі визначення траєкторії з мінімальною сумою кутів зміни напрямку руху нитки, коли перешкоди представляються відрізками прямих перпендикулярних до переважного напрямку. Обчислювальна складність алгоритму, запропонованого в [21] дорівнює $O(n^3)$, де n – кількість вершин моделі, або кількість перешкод.

Покажемо, що для розв'язання розглянутої задачі можна застосувати і іншу модель, основна позитивна риса якої полягає в тому, що цільова функція належить до класу α -функцій. Це дає змогу застосовувати класичні алгоритми пошуку оптимального шляху в графі. Побудуємо модель у вигляді орієнтованого графа $G^* = (V^*, E^*)$, де V^* – множина вершин, E^* – множина ребер. Множина вершин V^* взаємно однозначно відповідає відріzkам дотичних, які не перетинають перешкод. Дві вершини v_i, v_j з'єднуються ребром тоді і тільки тоді, коли відповідні відрізки дотичні до спільної перешкоди p , і перехід від відрізка, що відповідає v_i , до відрізка v_j задовольняє умові обгинання p . Кожному ребру множини E присвоюється вартість, яка дорівнює куту між відповідними дотичними.

В моделі G^* вартість будь-якого шляху w дорівнює сумі вартостей ребер, що належать w . Тому задача пошуку оптимальної траєкторії зводиться до визначення в G^* найкоротшого шляху, що з'єднує деяку вершину, якій відповідає відрізок дотичний до джерела, з вершиною, якій відповідає відрізок дотичний до цілі. Іншими словами отримуємо задачу пошуку найкоротшого шляху в графі з декількома джерелами і декількома цілями.

Для розв'язання останньої задачі можна скористатися відомими алгоритмами і, зокрема, алгоритмом Дейкстри, обчислювальна складність якого дорівнює $O(n^2)$, де n – кількість вершин моделі G^* , або кількість відрізків дотичних до перешкод.

Для прикладу візьмемо вихідні умови задачі, зображені на рис.8.1.8. Модель для розв'язання задачі показана на рис.8.1.10.

Модель, побудовану вказаним способом, можна спростити. Спочатку виконаємо операцію злиття тих вершин графа, які відповідають відріzkам прямих, що виходять з джерела. А потім тих вершин, які відповідають

відрізкам прямих, що входять в ціль. Спрощений таким чином граф зображено на рис.8.1.11.

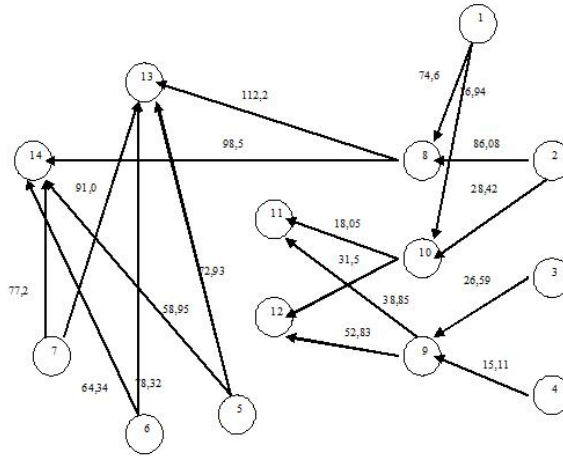


Рис.8.1.10. Модель для розв'язання задачі

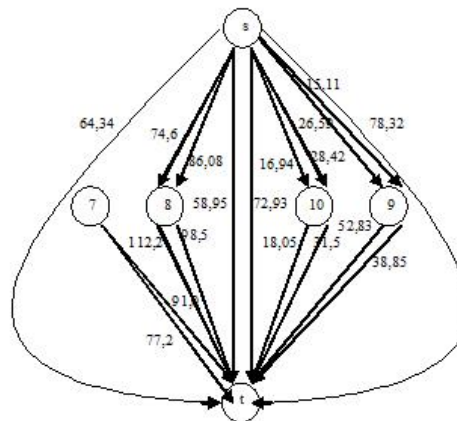


Рис.8.1.11. Модель для розв'язання задачі після виконання операції злиття вершин

Якщо в отриманому мультиграфі з вершини v_m у вершину v_n ведуть декілька ребер, то на другому етапі спрощення моделі серед цих ребер вибирається лише те, яке має мінімальну вартість. Спрощений таким чином граф зображено на рис.8.1.18.1.

Далі, застосовуючи алгоритм Дейкстри, отримаємо найкоротший шлях з s в t , який для даної моделі дорівнює 34,99. На основі загального підходу оптимізації шляхів у графі розроблено алгоритм з асимптотичною

обчислювальною складністю $O(n^3)$ для розв'язання задачі пошуку оптимальної траєкторії нитки на площині за умови, що перешкоди представляються колами. Запропоновано модель для розв'язання задачі визначення траєкторії нитки, за якої сила тертя, що діє на нитку, набуває мінімального значення. Для побудованої моделі застосовано алгоритм Дейкстри, який має обчислювальну складність $O(n^2)$. Отримані результати можуть бути узагальнені на випадок перешкод довільної форми, коли можливі відрізки траєкторії нитки між сусідніми перешкодами визначаються однозначно формою цих перешкод.

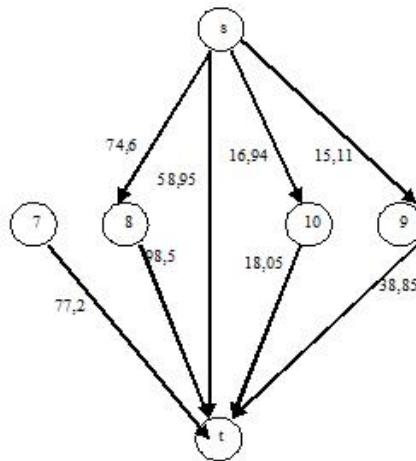


Рис.8.1.18.1. Остаточний вигляд моделі для розв'язання задачі

Програмна реалізація алгоритму пошуку оптимальної траєкторії заправки нитки в круглов'язальних машинах

В додатку Б представлений програмний код. Дана програма складається з одного модуля MainUnit.pas, що відповідає головній формі програми. Для зручності користування програмою створено меню за допомогою компонента MainMenu. Для відображення графічної інформації використано компонент Image, а прокрутка зображення реалізована за

допомогою елементів керування ScrollBar. Для реалізації можливостей завантаження даних з файлу та збереження їх у файл використано елементи OpenFileDialog та SaveDialog відповідно.

Для збереження даних та їх використання під час розрахунків використані такі структури даних як TList та Record.

З метою зменшення програмного коду створено процедури та функції, які неодноразово використовуються у програмі. Основні з них описано нижче.

- Функція

FindAngle(x1_1,y1_1,x1_2,y1_2,x2_1,y2_1,x2_2,y2_2:real):real шукає кут між двома дотичними. Параметрами функції є координати точок дотику першої та другої дотичних.

- Функція FindNaklon(x1,y1,x2,y2:real):real розраховує кут нахилу дотичної.

- Функція FindCross(x1,y1,x2,y2,xc,yc,R:real):boolean перевіряє перетин відрізка дотичної з колом.

- Процедура DrawAxis(sx,fx,sy,fy:integer) рисує на компоненті Image вісі координат та координатну сітку.

- Процедура DrawMyEllipse(xc,yc,R:real;mytype:string) рисує коло з заданими параметрами (координати центра та радіус) та ставить номер цього кола.

- Процедура ReDrawEllipse(shift_x,shift_y:integer) перерисовує кола, які вже існують, дотичні (якщо вони побудовані) та шлях (якщо він знайдений). Ця процедура використовується під час прокрутки зображення, зміни масштабу, побудові моделі, редагуванні та в інших випадках.

• Процедура `BuildTangent(e11,e12:string;xc1,yc1,R1,xc2,yc2,R2:real)` знаходить координати точок дотику спільної дотичної до двох кіл та заносить ці дані у список дотичних.

• Процедура `Koord(u,v:real)` переводить математичні координати до координат приладу.

• Процедура `MathKoord(ks,et:real)` переводить координати приладу до математичних координат.

Для ефективного використання програми та запобігання виникненню проблем необхідно ознайомитися з даною інструкцією.

Запуск програми здійснюється файлом `Project1.exe`.

Для роботи з програмою створене меню, зображене на рис. 5.1, яке складається з наступних пунктів:

Файл; Вихідні дані; Пошук; Убрати модель; Розміри поля; Редагування; Масштаб; Очистити.



Рис.8.1.13. Меню для роботи з програмою

Розглянемо меню Файл. Поки не задані вихідні умови задачі, деякі пункти меню, а саме: Пошук, Убрати модель, Редагування, знаходяться в неактивному стані. Для зручності вихідні умови задачі можна завантажити з файлу, в якому зберігаються координати джерела, цілі і перешкод та їх радіуси. Для завантаження вихідних даних з файлу необхідно вибрати в меню “Файл” підпункт “Відкрити”. Коли вибрано цей пункт, на екрані з’являється діалогове вікно “Відкрити”, де можна обрати необхідний файл з даними. Для можливості збереження даних створено підпункт “Зберегти”. Після обрання цього пункту умови задачі зберігаються в обраному файлі.

У будь-який час роботи з програмою є можливість вивести на принтер зображення форми. Для цього необхідно вибрати підпункт “Друкування” та у вікні, що з’явиться, підтвердити бажання вивести форму на друк. Коректний вихід із програми здійснюється кнопкою “Вихід”.

Розміри поля можна задати, вибравши в меню пункт “Розміри поля”. При цьому з’являється вікно, яке показано на рис. 8.1.14, для введення максимального значення координати X та Y.

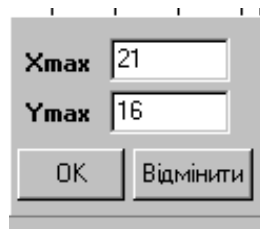


Рис.8.1.14. Вікно для завдання розмірів поля

Пункт меню “Вихідні дані” дозволяє вручну вводити вихідні дані задачі. Підпункти “Джерело” та “Ціль” дозволяють задати координати джерела та цілі відповідно, а також їх радіуси. Для унеможливлення введення більш ніж одного джерела або цілі ці підпункти меню стають неактивними. Якщо дані були завантажені з файлу, в якому вже були вказані характеристики джерела та цілі, то ці підпункти стають неактивними.

Додати перешкоду можна за допомогою кнопки “Коло”. При цьому з’являється панель, яка зображена на рис. 8.1.15, з полями, в які необхідно вводити дані, а саме: координати центра кола та його радіус. Для того щоб нарисувати це коло необхідно натиснути кнопку “Нарисувати”. Також є можливість відмінити введення даних, натиснувши кнопку “Відмінити”. Після введення даних автоматично здійснюється перевірка на перетин цього кола з іншими колами. Якщо це коло перетинає будь-яке коло, яке було задано раніше, то з’явиться попередження про неможливість додання

такого кола. Підпункт меню “Відмінити” дозволяє відмінити введення останнього кола.



Рис.8.1.15. Вікно для введення параметрів кола

Коли виникає необхідність внести зміни у розташування джерела, цілі або перешкод, то немає необхідності відмінити введення. Для цього випадку в програмі передбачена можливість редагування координат центру. В меню “Редагування”, яке стає активним після введення хоча б одної перешкоди, необхідно вибрати пункт “Почати”. У верхньому правому куті з'явиться напис “Редагування”. Після цього, навівши курсор на коло, перетягти його у інше місце. Після наведення курсору на коло він змінює свій вигляд. Програмою передбачена перевірка на перетинання колом інших кіл. Якщо перетягти коло так, щоб воно перетинало інше, то це коло повернеться на своє початкове місце розташування. Режим редагування зображений на рис. 8.1.16. Для закінчення редагування необхідно вибрати пункт “Закінчити”. Після цього змінювати розташування кіл стає неможливим.

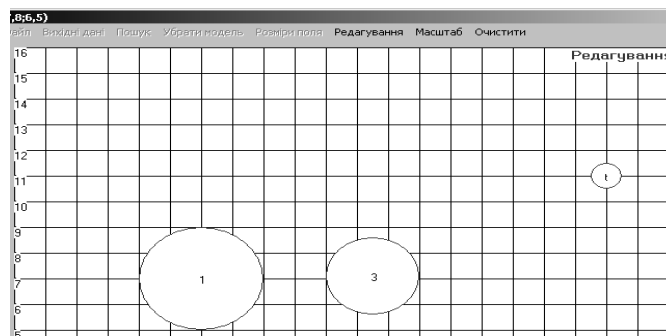


Рис.8.1.16. Режим редагування

Розглянемо “Меню Пошук”. Перед тим, як знайти шлях необхідно побудувати модель. Підпункт “Модель” стає активним, якщо були задані координати джерела та цілі. Після натискання цієї кнопки будується та з'являється на екрані модель для розв'язання задачі. Після цього є можливість знайти розв'язок задачі, вибравши підпункт “Шлях”. Якщо розв'язок задачі знайдено, то на екрані цей шлях буде виділено зеленим кольором. В лівому нижньому куті екрану з'явиться напис “Сумарний кут” та число, що є розв'язком задачі. Приклад знайденого шляху зображено на рис. 8.1.17.

Для зручності перегляду результатів є можливість убрати модель для розв'язання задачі, натиснувши кнопку “Убрати модель”. При цьому на екрані залишаються лише перешкоди та знайдений шлях, як показано на рис. 8.1.18.

Якщо у користувача виникають сумніви щодо правильності знаходження шляху, то дана програма передбачає знаходження значення цільової функції по вибраним вручну ребрам. Така можливість з'являється після побудови моделі і нею можна скористатися до пошуку шляху або після його знаходження. Для виділення ребер в меню “Редагування” треба вибрати пункт “Ребра”. Після цього з'являється можливість вибрати необхідні ребра, клацнувши по ребру мишею. При цьому змінюється вигляд курсору і ребро виділяється червоним кольором. Після виділення всього шляху можна знайти значення цільової функції, вибравши в меню “Редагування” пункт “Пошук по ребрам”. Знайдене значення з'являється в лівому нижньому куті екрану.

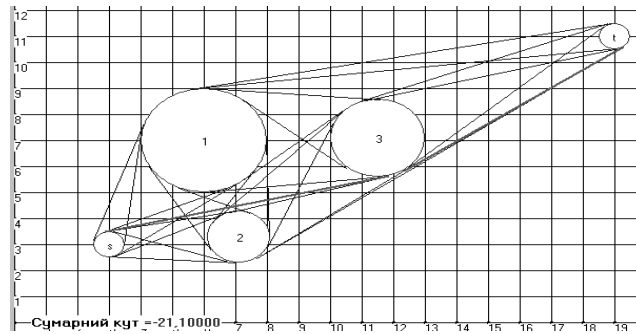


Рис.8.1.17. Результат роботи програми

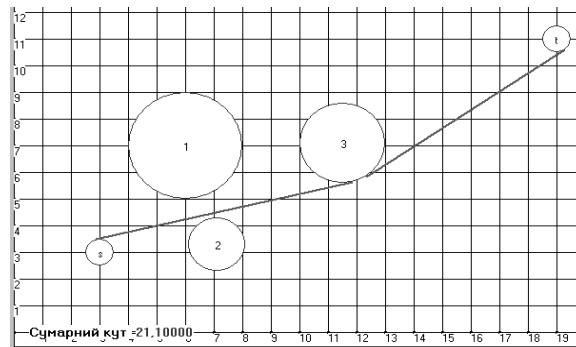


Рис.8.1.18. Результат роботи програми після прибрання моделі

Якщо картинка повністю не відображається на екрані, то можна зменшити масштаб, вибравши відповідний пункт в меню “Масштаб”. При бажанні розглянути знайдену траєкторію більш детально, масштаб можна збільшити. Для продовження роботи з програмою можна вибрати пункт меню “Очистити” і вводити нові дані.

Математичне та програмне забезпечення системи проектування лінії заправки в круглов’язальних машинах у просторі

Пружна система заправки є складовою частиною системи подачі нитки технологічного устаткування. Недосконалість структури пружної системи заправки не дозволяє здійснювати мінімізацію натягу в процесі роботи технологічного устаткування і призводить до порушення технологічного режиму, обривам. В першу чергу, на це впливає недосконалість форми пружної системи заправки, нитконапрямних елементів, великими кутами охоплення ниткою напрямних,

нерівномірність нитки за діаметром, що зумовлено її структурою і матеріалом, специфікою виготовлення нитки, довжиною окремих філаментів та їх взаємним розташуванням відносно один до одного, а також нерівномірність вхідного натягу[117, 129].

Розробка оптимальної форми пружної системи заправки в'язальних машин можна ефективно здійснювати з використанням ЕОМ. На рис.8.1.19 представлена загальна графічна схема робочої зони в'язальної машини. Її розміри визначаються як її конструкцією, так і функціональним призначенням. Розробка програмного забезпечення, блок-схема якого представлена на рис.8.1.20, необхідно з визначення габаритних розмірів пружної системи заправки, що проектується. Її розміри визначаються як її конструкцією, так і функціональним призначенням.

Наступним етапом є визначення координат закритих зон. Під цим терміном мається на увазі ті ділянки, через які нитка не може проходити. Це різні елементи станини в'язальної машини, допоміжні механізми. До цих зон можна віднести ділянки загального простору робочої зони в'язальної машини, де необхідно забезпечити нормальні умови обслуговування.

До вхідних даних, при розробці основ математичного та програмного забезпечення САПР форми пружної системи заправки в'язальних машин, необхідно віднести величину натягу нитки перед зоною в'язання, довжину нитки в зоні подачі нитки, координати точок входу та виходу з робочої зони в'язальної машини.

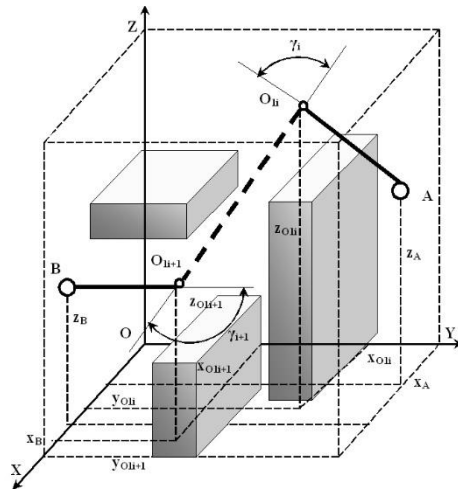


Рис.8.1.19. Графічна схема робочої зони в'язальної машини

При побудові форми пружної системи заправки в'язальних машин, на початковому етапі, приймаємо нульовий варіант, коли нитка після вихідного отвору А (після балону) не має точок перегину і потрапляє в точку В (зона в'язання). У випадку, коли пряма АВ перетинає закриті зони, програма обирає із сформованого масиву першу точку перегину. Після цього знову будується нова пряма, яка єднає останню точку з точкою виходу В. Знову виконується перевірючий розрахунок. Коли умова не виконується, обирається наступна точка перегину.

Отримана форма пружної системи заправки в'язальних машин перевіряється на предмет заданої довжини нитки. Цю вимогу необхідно виконати для забезпечення необхідної жорсткості нитки при розтягненні. При малій довжині приведений коефіцієнт жорсткості буде більшим, а при більшій довжині збільшуються розміри зони подачі нитки.

Після формування пружної системи заправки обираються нитконапрямні та нитконатягувальні елементи системи подачі нитки. Для цього використовуються підпрограми визначення вихідного натягу для нитконатягувачів.

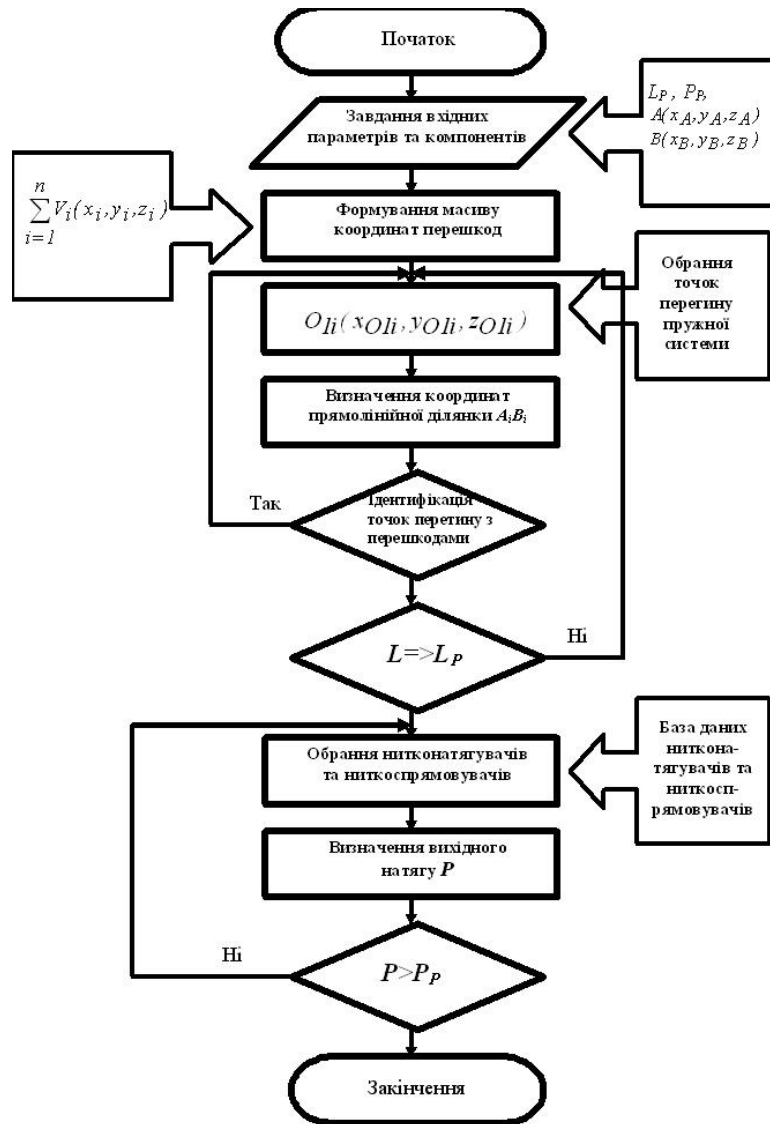


Рис.8.1.20. Блок-схема алгоритму побудови форми пружної системи заправки

Далі, згідно з наведеною блок-схемою (рис.8.1.20), здійснюється перевірка отриманого значення вихідного натягу в точці В з заданим. У випадку його підвищеного значення необхідно зменшити величину кута охоплення отворів нитконатягувача чи змінити форму пружної системи заправки в'язальної машини. Число нитконатягувачів в розрахунках приймалося рівним одному. Це пояснюється тим, що існуючі нитконатягувачі можуть змінити величину натягу в досить широких межах. На рис.8.1.21 представлена розрахункова схема для визначення

кута охоплення для i - направляючої в точці перегину. Система рівнянь для його визначення має вигляд

$$\begin{aligned}
 O_{li}E_i &= z_{Oli} - z_{Bi}, \quad O_{li}K_i = z_{Oli} - z_{Ai}, \\
 \beta_i &= \beta_{li} + \beta_{2i}, \quad \alpha_i = \alpha_{li} + \alpha_{2i}, \quad \gamma_i = \alpha_i + \beta_i, \\
 E_iB_i &= \sqrt{(x_{Oli} - x_{Bi})^2 + (y_{Bi} - y_{Oli})^2}, \\
 K_iA_i &= \sqrt{(x_{Ai} - x_{Oli})^2 + (y_{Oli} - y_{Ai})^2}, \\
 O_{li}B_i &= \sqrt{(x_{Oli} - x_{Bi})^2 + (y_{Bi} - y_{Oli})^2 + (z_{Oli} - z_{Bi})^2}, \\
 O_{li}A_i &= \sqrt{(x_{Oli} - x_{Ai})^2 + (y_{Ai} - y_{Oli})^2 + (z_{Oli} - z_{Ai})^2}, \\
 O_{li}B_i &= \frac{R_i}{\sin \beta_{li}} = \frac{O_{li}E_i}{\sin \beta_{2i}}, \quad O_{li}A_i = \frac{R_i}{\sin \alpha_{li}} = \frac{O_{li}K_i}{\sin \alpha_{2i}}, \quad \alpha_{li} = \arcsin\left(\frac{R_i}{O_{li}A_i}\right), \\
 \alpha_{2i} &= \arcsin\left(\frac{O_{li}K_i}{O_{li}A_i}\right), \quad \beta_{li} = \arcsin\left(\frac{R_i}{O_{li}B_i}\right), \quad \beta_{2i} = \arcsin\left(\frac{O_{li}E_i}{O_{li}B_i}\right),
 \end{aligned} \tag{8.1.17}$$

$x_{Oli}, y_{Oli}, z_{Oli}$ - координати i – циліндричної напрямної;

x_{Ai}, y_{Ai}, z_{Ai} - координати i – точки перегину;

x_{Bi}, y_{Bi}, z_{Bi} - координати $i+1$ – точки перегину;

R_i - радіус i – циліндричної напрямної;

γ_i - сумарний кут охоплення ниткою i – циліндричного ниткопрямовувача.

Система (8.1.17) використовується для реалізації алгоритму при побудові точок перегину. Реалізація алгоритму дозволила удосконалити форму пружної системи заправки для круглов'язальних машин.

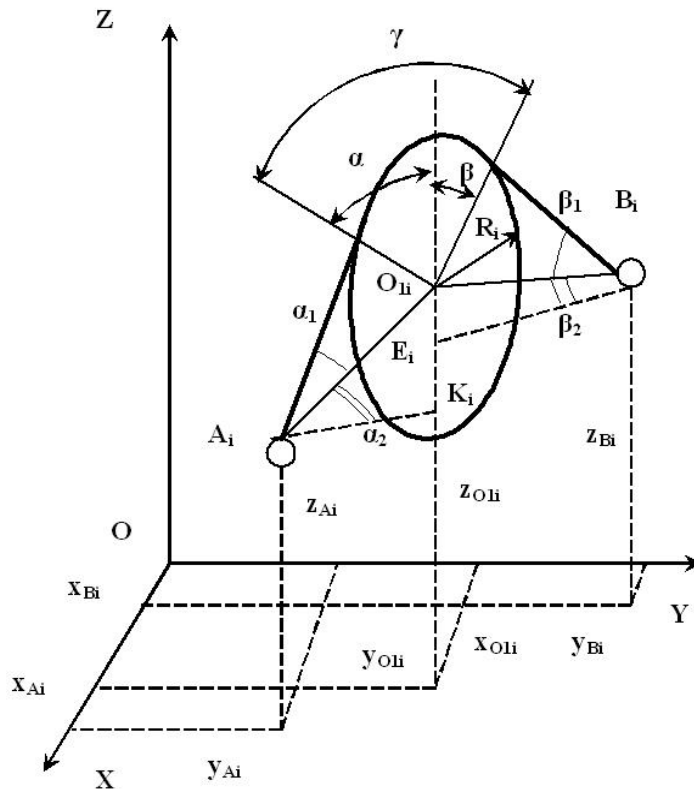


Рис.8.1.21. Розрахункова схема

Висновки по розділу

1. Аналіз структурних схем лінії заправки нитки круглов'язальних машин показав, що вони мають дуже складну конфігурацію, як у площині так і у просторі габаритних розмірів. В точках зламу лінії заправки має місце взаємодія нитки з направляючими отворами спрямовувачів нитки, пристроями що забезпечують натяг нитки.
2. Вперше проведено теоретичне обґрунтування обчислювальної схеми алгоритму послідовної оптимізації, що мінімізує пошуки в дереві варіантів, що дало змогу виконати обчислювальну реалізацію алгоритму послідовної оптимізації, що мінімізує пошуки в дереві варіантів.
3. На основі реалізації розробленого алгоритму вирішена задача синтезу системи подачі нитки на круглов'язальних машинах для випадку перешкод у вигляді вертикальних ліній та розроблено програмне забезпечення для пошуку оптимальної форми заправки нитки на круглов'язальній машині.

4. На основі застосованого алгоритму Дейкстри виконана оптимізація плоскої траєкторії нитки в круглов'язальних машинах у випадку перешкод у вигляді кіл.
5. Виконана програмна реалізація алгоритму пошуку оптимальної траєкторії заправки нитки в круглов'язальних машинах.
6. На основі системного аналізу отримане математичне та програмне забезпечення системи проектування форми лінії заправки в круглов'язальних машинах у просторі, що дозволило оптимізувати конструктивні параметри.

8.2. УДОСКОНАЛЕННЯ КОНСТРУКТИВНИХ ЕЛЕМЕНТІВ СИСТЕМИ НИТКОПОДАЧІ КРУГЛОВ'ЯЗАЛЬНИХ МАШИН

На другому етапі, після визначення оптимальної форми заправки нитки, необхідно визначити вплив ниткоспрямовувачів, нитконатягувачів в точках зламу лінії заправки на натяг нитки перед робочою зоною в'язальної машини та визначити їх оптимальні параметри.

На рис.8.2.1 представлений загальний вигляд ниткоспрямовувачих елементів.

Аналіз умов взаємодії нитки з направляючими в точках зламу лінії заправки показує, що контакт між ниткою та направляючою поверхнею відбувається по деякій площині. Нитка рухається у жолобі, який створений внутрішньою поверхнею ниткоспрямовувача. Цю обставину необхідно враховувати при складенні систем диференціальних рівнянь, які описують рівновагу елемента нитки на напрямній поверхні.



Рис.8.2.1. Загальний вигляд ниткоспрямовувачих елементів

Загальні диференційні рівняння рівноваги нитки на напрямній поверхні

Для спрощення процесу складання систем диференційних рівнянь, які описують рівновагу елемента нитки на напрямній поверхні цілком доцільно використовувати моделі ниток, які по своїй будові і вигляду найбільш наближені до реальних [5, 11]. На рис. 8.2.2 представлені найбільш поширені схеми для випадків, коли реальну нитку чи пряжу замінюють одним циліндричним тілом, або реальну нитку представляють як об'єкт, який складається з системи окремих циліндричних тіл, діаметр яких наближений до діаметру окремих філаментів реальної комплексної нитки чи пряжі [1-5, 11, 22]. В даному випадку маємо справу з системою матеріальних тіл. Використовуючи основні принципи методу кінетостатики необхідно для досягнення умовної рівноваги до активно

заданих сил і моментів додати відповідні реакції зв'язків і сили інерції, які діють на тіло або систему тіл.

Для складання систем диференціальних рівнянь, які описують рівновагу елемента нитки на напрямній поверхні, необхідно всі зовнішні сили і реакції зв'язків, нитки, що діють на елемент нескінченно малої довжини dS , представити у вигляді головного вектора R_0 і головного моменту M_0 . Дані вектора приведемо до точки A_* - центру мас виділеного елемента нитки. До даної точки приводиться і результуюча сила інерції Φ_* і вектор моменту інерції M_ϕ . Головний трикутник $\tau_* n_* b_*$, з вершиною в точці A_* , служить координатною системою, щодо якої розглядається умовна рівновага всіх сил і моментів[2].

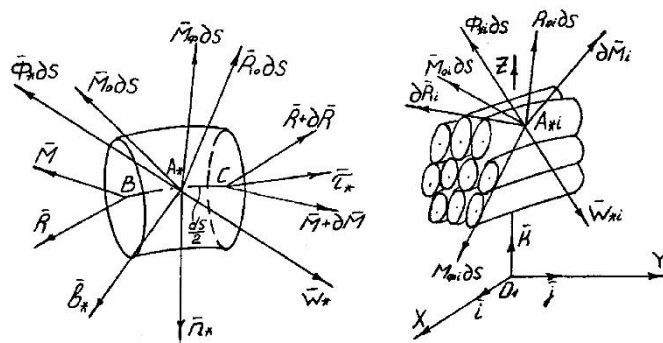


Рис.8.2.2. Фізичні моделі мононитки та комплексної нитки та пряжі

Виділення елемента [5, 116] нескінченно малої товщини BC шляхом перетину нитки двома площинами, перпендикулярними осі нитки в точках B і C , дозволяє всі внутрішні силові чинники в кожному з них привести до головного вектора і головного моменту: до точки B (M, R) - для лівого перетину; до точки C ($R+\partial R, M+\partial M$) - для правого перетину. При побудові даної схеми вважали, що рух відбувається зліва направо[16, 17, 22].

Векторні рівняння

$$\frac{\partial \vec{R}}{\partial S} + \vec{R}_0 - \vec{\Phi}_* = 0; \quad \frac{\partial \vec{M}}{\partial S} + \vec{M}_0 + \vec{\tau}_* \times \vec{Q} - \vec{M}_\phi = 0, \quad (8.2.1)$$

описують рух нескінченно малого елемента нитки.

У випадку, коли реальну нитку представляють як об'єкт, який складається з системи окремих циліндричних тіл, діаметр яких наближений до діаметру окремих філаментів реальної комплексної нитки чи пряжі, для складання системи диференціальних рівнянь двома площинами, перпендикулярними осі комплексної нитки або пряжі, відсікаємо елемент нескінченно малої довжини ds , що складається з n елементарних філаментів [16, 22]. Відповідно до приведеного вище методу кінетостатики [4, 5] необхідно до кожного i -го елементарного філаменту прикласти вказані вище сили.

Центром приведення у кожного i -го елементарного волокна буде точка A_{*i} - центр мас. Всі активно задані сили і відповідні реакції зв'язків приведемо до головного вектора R_{0i} і головному моменту M_{0i} . По аналогії з першим варіантом внутрішні зусилля в двох перетинах також приведемо до центру мас [2]. Їх дію на елементарний філамент замінює головний вектор dR_i і головний момент dM_i .

Сили інерції приведемо до головного вектора Φ_{*i} і головному моменту інерції $M_{\phi i}$ [5, 4].

Для i -го філаменту векторні рівняння рівноваги матимуть вигляд [2]

$$\frac{\partial \vec{R}_i}{\partial S} + \vec{R}_{0i} - \vec{\Phi}_{*i} = 0; \quad \frac{\partial \vec{M}_i}{\partial S} + \vec{M}_{0i} - \vec{M}_{\phi i} = 0. \quad (8.2.2)$$

Просумуємо дані залежності для n елементарних волокон, з яких складається комплексна нитка або пряжа, то при цьому реакції зв'язків (окрім внутрішніх зусиль) знищаться як взаємно урівноважені сили для кожної пари волокон [4].

Спроекуємо систему диференціальних рівнянь (8.2.2) на координатні осі X, Y, Z , при цьому отримаємо систему шести диференціальних рівнянь

$$\begin{aligned} \sum_{i=1}^n \left(\frac{\partial R_{ix}}{\partial S} + R_{0ix}^- - \Phi_{*ix} \right) &= 0; \quad \sum_{i=1}^n \left(\frac{\partial M_{ix}}{\partial S} + M_{0ix}^- - M_{\phi ix} \right) = 0; \\ \sum_{i=1}^n \left(\frac{\partial R_{iy}}{\partial S} + R_{0iy}^- - \Phi_{*iy} \right) &= 0; \quad \sum_{i=1}^n \left(\frac{\partial M_{iy}}{\partial S} + M_{0iy}^- - M_{\phi iy} \right) = 0; \\ \sum_{i=1}^n \left(\frac{\partial R_{iz}}{\partial S} + R_{0iz}^- - \Phi_{*iz} \right) &= 0; \quad \sum_{i=1}^n \left(\frac{\partial M_{iz}}{\partial S} + M_{0iz}^- - M_{\phi iz} \right) = 0, \end{aligned}$$

де відповідний індекс x , y або z указує проекцію на відповідну координатну вісь.

Проте коли реальну нитку представляють як об'єкт, який складається з системи окремих циліндричних тіл, діаметр яких наближений до діаметру окремих філаментів реальної комплексної нитки чи пряжі, виникають проблеми з адекватністю опису процесу взаємодії нитки з напрямною поверхнею. Моделі даного типу є досить громіздкими [2, 4]. Наприклад, при її побудові філаменти вважають розташованими прямолінійно. Насправді, елементарні волокна переплітаються між собою, а це істотним чином міняє характер взаємодії між ними. Крім того, при вирішенні приведеної вище системи рівнянь стикаються з дуже серйозними труднощами. Це стосується складності у визначенні проекцій векторів сил і моментів на відповідні осі координат.

Таким чином, враховуючи все сказане вище, найбільш прийнятною, з погляду простоти отримуваних результатів, є модель нитки, у якої весь об'єм рівномірно заповнений матеріалом.

Головний вектор R рівний

$$\vec{R} = P\vec{\tau}_* + Q_2\vec{n}_* + Q_3\vec{b}_*, \quad (8.2.3)$$

а головний момент M рівний

$$\vec{M} = M_k\vec{\tau}_* + M_{u2}\vec{n}_* + M_{u3}\vec{b}_*. \quad (8.2.4)$$

Для отримання системи диференціальних рівнянь необхідно векторні рівняння (8.2.3)-(8.2.4) спроектувати на осі головного тригранника $\tau_*n_*b_*$ (рис.8.2.2).

Диференціюємо вирази (8.2.3) і (8.2.4) по дуговій координаті

$$\begin{aligned} \frac{\partial \vec{R}}{\partial S} &= \left(\frac{\partial P}{\partial S} - Q_2 q_1 + Q_3 p_1 \right) \vec{\tau}_* + \left(\frac{\partial Q_2}{\partial S} + P q_1 + Q_3 r_1 \right) \vec{n}_* + \left(\frac{\partial Q_3}{\partial S} - P p_1 + Q_2 r_1 \right) \vec{b}_*, \\ \frac{\partial \vec{M}}{\partial S} &= \left(\frac{\partial M_k}{\partial S} - M_{u2} q_1 + M_{u3} p_1 \right) \vec{\tau}_* + \left(\frac{\partial M_{u2}}{\partial S} + M_k q_1 + M_{u3} r_1 \right) \vec{n}_* + \\ &+ \left(\frac{\partial M_{u3}}{\partial S} - M_k p_1 + M_{u2} r_1 \right) \vec{b}_*, \end{aligned} \quad (8.2.5)$$

де P - натяг нитки; Q_2, Q_3 - проекції перерізуючої сили на нормаль і бінормаль; M_k - крутячий момент; M_{u2}, M_{u3} - згинаючі моменти у відповідних перетинах нитки.

Сила інерції визначиться по формулі [2, 4]

$$\begin{aligned} \vec{\Phi}_* &= T \vec{W}_* = T \times \\ &\times \left\{ \left[\frac{\partial (V_\tau + U'_\tau)}{\partial t} + (\omega_{0n} + \omega_{un})(V_b + U'_b) - (\omega_{0b} + \omega_{ub})(V_n + U'_n) \right] \vec{\tau}_* + \right. \\ &+ \left[\frac{\partial (V_n + U'_n)}{\partial t} - (\omega_{0\tau} + \omega_{u\tau})(V_b + U'_b) + (\omega_{0b} + \omega_{ub})(V_\tau + U'_\tau) \right] \vec{n}_* + \\ &\left. + \left[\frac{\partial (V_b + U'_b)}{\partial t} + (\omega_{0\tau} + \omega_{u\tau})(V_n + U'_n) - (\omega_{0n} + \omega_{un})(V_\tau + U'_\tau) \right] \vec{b}_* \right\}, \end{aligned} \quad (8.2.6)$$

де T - лінійна щільність нитки.

Для визначення величини M_ϕ скористаємося відомим з теоретичної механіки співвідношенням [1-5, 11], тоді для елемента нитки

$$\vec{M}_\phi = \frac{J}{\partial S} \frac{\partial \vec{\omega}}{\partial t}, \quad (8.2.7)$$

де J - тензор інерції (враховуючи, що осі головного тригранника, розташованого в центрі мас елемента BC (рис.8.2.2), є осями симетрії і головними осями інерції, вважаємо, що відцентрові моменти інерції рівні нулю і значущими будуть тільки розташовані на головній діагоналі матриці тензора інерції).

Складові тензора інерції, розташовані на головній діагоналі матриці, визначаються по формулах [5]

$$\begin{aligned} J_{\tau^*} &= \gamma_n [J_{\tau_0} \pm \Delta_{\tau}(\vec{U})] \rho S, \\ J_{n^*} &= \gamma_n [J_{n_0} \pm \Delta_n(\vec{U})] \rho S, \\ J_{b^*} &= \gamma_n [J_{b_0} \pm \Delta_b(\vec{U})] \rho S, \end{aligned} \quad (8.2.8)$$

де J_{τ^*} , J_{n^*} , J_{b^*} - відповідно моменти інерції виділеного елемента щодо осей головного тригранника;

J_{τ_0} , J_{n_0} , J_{b_0} - геометричні моменти інерції щодо осей головного тригранника до зм'ятої нитки;

$\Delta_{\tau}(U)$, $\Delta_n(U)$, $\Delta_b(U)$ - функціональні коефіцієнти, що визначають зміну геометричних моментів інерції за рахунок змінання в зоні контакту;

γ_n - об'ємна щільність.

З урахуванням залежності (8.2.8) вираз (8.2.7) прийме вигляд

$$\vec{M}_{\phi} = \gamma_n \left\{ [J_{\tau_0} \pm \Delta_{\tau}(\vec{U})] \varepsilon_1 \vec{\tau}_* + [J_{n_0} \pm \Delta_n(\vec{U})] \varepsilon_2 \vec{n}_* + [J_{b_0} \pm \Delta_b(\vec{U})] \varepsilon_3 \vec{b}_* \right\}, \quad (8.2.9)$$

де ε_1 , ε_2 , ε_3 - проекції вектора кутового прискорення елемента нитки ε_* на осі головного тригранника.

Проекції головного вектора всіх зовнішніх сил, для випадку взаємодії нитки з тією, що направляє, можна виразити таким чином

$$\vec{R}_0 = F_{\tau} \vec{\tau}_* + F_n \vec{n}_* + F_b \vec{b}_*,$$

де F_{τ} , F_n , F_b - проекції вектора R_0 на осі головного тригранника.

На рис.8.2.3 представлена схема взаємодії нитки з направляючими в точках зламу лінії заправки

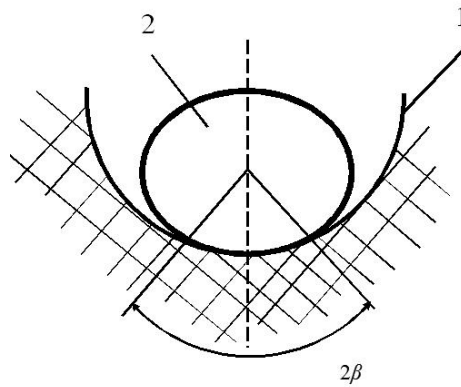


Рис.8.2.8.2. Схема взаємодії нитки з направляючими в точках зламу лінії заправки

Тоді рівняння для головного вектора всіх зовнішніх сил прийме вигляд [6, 11]

$$\begin{aligned} \vec{R}_0 &= F_\tau \vec{\tau}_* + F_n \vec{n}_* + F_b \vec{b}_*, \\ F_\tau &= \mu N, \quad F_n = N, \\ \mu &= \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b}, \end{aligned} \quad (8.2.10)$$

де μ – коефіцієнт тертя;

N – питома нормальна реакція поверхні;

β – половина кута дуги контакту нитки з направляючими в точках зламу лінії заправки;

a, b – постійні коефіцієнти для конкретного виду комплексної нитки чи пряжі (для капронової нитки 28 текс ($a= 0,1765, b= 0,1186$), віскозній комплексній нитки 16,7 текс ($a= 0,1580, b= 0,1160$), вовняної пряжі 29,9 ($a= 0,1330, b= 0,0910$), бавовняної пряжі 27,6 текс ($a= 0,1656, b= 0,0590$), віскозної штапельної пряжі 28,8 текс ($a= 0,1470, b= 0,0690$));

P_0 – натяг веденої гілки нитки;

R – радіус кривизни направляючої.

Залежність для головного моменту зовнішніх сил M_0 має вигляд

$$\vec{M}_0 = M_\tau \vec{\tau}_* + M_n \vec{n}_* + M_b \vec{b}_*, \quad (8.2.11)$$

де M_τ , M_n , M_b - відповідні проекції на осі головного тригранника.

Результуюча система диференціальних рівнянь в проекції на осі головного тригранника, що описує рух ниток з урахуванням їх реальних фізико-механічних властивостей по направляючій поверхні, з урахуванням (8.2.1) -(8.2.11) має вигляд [11, 12, 13]

$$\begin{aligned}
 & \frac{\partial P}{\partial S} - Q_2 q_1 + Q_3 p_1 - \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N \cos \varphi_t = T \times \\
 & \times \left[\frac{\partial(V_\tau + U'_\tau)}{\partial t} + (\omega_{0n} + \omega_{un})(V_b + U'_b) - (\omega_{0b} + \omega_{ub})(V_n + U'_n) \right]; \\
 & \frac{\partial Q_2}{\partial S} + P q_1 + Q_3 r_1 - N = T \times \\
 & \times \left[\frac{\partial(V_n + U'_n)}{\partial t} - (\omega_{0\tau} + \omega_{u\tau})(V_b + U'_b) + (\omega_{0b} + \omega_{ub})(V_\tau + U'_\tau) \right]; \\
 & \frac{\partial Q_3}{\partial S} - P p_1 + Q_2 r_1 + \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N \sin \varphi_t = T \times \\
 & \times \left[\frac{\partial(V_b + U'_b)}{\partial t} + (\omega_{0\tau} + \omega_{u\tau})(V_n + U'_n) - (\omega_{0n} + \omega_{un})(V_\tau + U'_\tau) \right]; \\
 & \frac{\partial M_k}{\partial S} - M_{u2} q_1 + (c_0 + c_1 K_r + c_2 K_r^2) q_1 p_1 + M_\tau = \gamma_n [J_{\tau_0} \pm \Delta_\tau(\bar{U})] \varepsilon_1; \\
 & \frac{\partial M_{u2}}{\partial S} + M_k q_1 + (c_0 + c_1 K_r + c_2 K_r^2) q_1 r_1 - Q_3 + M_n = \gamma_n [J_{n_0} \pm \Delta_n(\bar{U})] \varepsilon_2; \\
 & \frac{\partial[(c_0 + c_1 K_r + c_2 K_r^2) q_1]}{\partial S} - M_k p_1 + M_{u2} r_1 + Q_2 + \\
 & + r_x \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N = \gamma_n [J_{b_0} \pm \Delta_b(\bar{U})] \varepsilon_3, \\
 & N = b_2 E_1 \delta (1 - b_3 \delta^{b_4}) + \eta \delta^{b_5} (1 - b_5 \delta^{b_6}), \quad \delta = \frac{r - r_x}{r}, \quad \dot{\delta} = \frac{\partial U_v}{r \partial t}, \\
 & q_1 = \frac{\cos \Psi_0}{\rho_0} + \frac{\partial \beta}{\partial S} + \frac{\alpha}{\rho_{01}} + \alpha \frac{\partial \Psi_0}{\partial S}, \quad p_1 = \frac{\sin \Psi_0}{\rho_0} + \frac{\partial \alpha}{\partial S} - \frac{\beta}{\rho_{01}} - \beta \frac{\partial \Psi_0}{\partial S}; \\
 & r_1 = \frac{1}{\rho_0} + \frac{\partial \Psi_0}{\partial S} - \alpha \frac{\cos \Psi_0}{\rho_0} + \frac{\sin \Psi_0}{\rho_0} \beta, \quad \gamma = \frac{\partial U_\tau}{\partial S} - \frac{\cos \Psi_0}{\rho_0} U_v + \frac{\sin \Psi_0}{\rho_0} U_\beta; \\
 & \beta = \frac{\partial U_v}{\partial S} + \frac{\cos \Psi_0}{\rho_0} U_\tau - \frac{U_\beta}{\rho_{01}} - U_\beta \frac{\partial \Psi_0}{\partial S}, \quad \alpha = \frac{\sin \Psi_0}{\rho_0} U_\tau - \frac{\partial U_\beta}{\partial S} - \frac{U_v}{\rho_{01v}} - U_v \frac{\partial \Psi_0}{\partial S}.
 \end{aligned} \tag{8.2.12}$$

де r_x - відстань від точки A^* осі елемента нитки до направляючої ($r_x = r - U_v$, де r - розрахунковий радіус перетину нитки);

ρ_0, ρ_{01} - радіуси кривизни і геометричного крутіння осі нитки;

ψ_0 - кут Сен-Венана;

$U\tau$, $U\nu$, $U\beta$ – проєкції вектора поперечної деформації перетину на осі натурального тригранника[5, 11, 12];

b_2 - ширина сліду контакту;

δ' - швидкість відносної деформації поперечного перетину нитки;

η - коефіцієнт, що характеризує в'язкі властивості нитки при її поперечній деформації;

E_1 - поточний модуль жорсткості нитки на деформацію її перетину;

b_3 , b_4 , b_5 , b_6 - експериментальні коефіцієнти, які визначаються з діаграми "навантаження - деформація"[2];

c_0 , c_1 , c_3 - експериментальні коефіцієнти, які визначаються з діаграми "навантаження - деформація" при згинанні нитки на відповідний кут;

K_r – число повних обертань нитки відносно своєї вісі на одиницю довжини[5].

Декілька слів необхідно сказати про залежність величини питомого тиску від швидкості деформації і величини відносної деформації поперечного перетину нитки. Натяг нитки із збільшенням швидкості зменшується. При зменшенні розмірів поперечного перетину нитки вплив швидкості її руху на натяг зменшується. Отримані результати справедливі лише для моониток. Для комплексних ниток і пряжі спостерігається протилежна картина. Як буде показано нижче, при збільшенні швидкості руху нитки її натягнення зростає. Така ж тенденція спостерігається і при збільшенні відносної деформації поперечного перетину нитки. На наш погляд пояснення даного явища необхідно шукати при визначенні швидкості розповсюдження деформацій.

Отримана система є замкнутою нелінійною системою з 12 диференціальних і 4 алгебраїчних рівнянь. Дана система містить 16 невідомих функцій щодо аргументів S і t .

Інтегрування даної системи диференціальних рівнянь пов'язане з великими труднощами і можливе тільки із застосуванням ЕОМ. Складність інтегрування полягає у виборі граничних і початкових умов для похідних відповідних ступенів (вище другої).

Дослідження взаємодії нитки з направляючими в точках зламу лінії заправки з урахуванням жорсткості на згин та зминання

Такі реальні фізико-механічні властивості притаманні більшості комплексних ниток та пряжі, які переробляються на круглов'язальних машинах. Задача буде зведена до визначення натягу нитки після проходження по поверхні ниткоспрямовувача з урахуванням жорсткості на згин та зминання в зоні контакту. Будемо рахувати, що нитка розташована в спрямляючій площині. Силами інерції можна знехтувати. На нитку будуть діяти тільки сили тертя та питомий нормальний тиск в зоні контакту. На рис.8.2.4 представлена схема взаємодії нитки з направляючими в точках зламу лінії заправки з урахуванням жорсткості на згин та зминання.

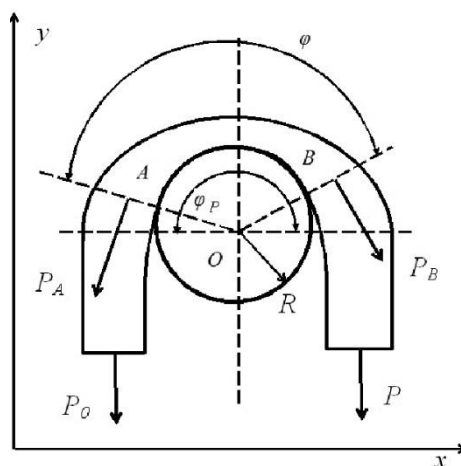


Рис.8.2.4. Схема взаємодії нитки з направляючими в точках зламу лінії заправки з урахуванням жорсткості на згин та зминання

Пошук взаємозв'язку між натягом ведучої P та веденої P_0 гілки нитки необхідно починати з встановлення залежності між натягом P_0 та натягом в точці P_A та натягом P та натягом в точці P_B . Визначимо зв'язок між натягом P_0 та P_A , а також між P та P_B . На цих ділянках вільні гілки нитки підкорюються законам пружних на згин одновимірних об'єктів. Нитка огинає циліндричну напрямну радіуса R .

Рівняння рівноваги пружної лінії запишемо у вигляді [2, 5]

$$\frac{y''(c_0 + c_1K_r + c_2K_r^2)}{(1 + y'^2)^{\frac{3}{2}}} = M, \quad M = P(x_A - x), \quad (8.2.13)$$

де $(c_0 + c_1K_r + c_2K_r^2)$ - коефіцієнт жорсткості нитки на згин;

y', y'' - перша та друга похідна координати y по x ;

M - згинаючий момент;

P - натяг нитки;

x_A - координата точки прикладення сили натягу P_0 .

Диференційне рівняння (8.2.13) запишемо у вигляді

$$\frac{y''(c_0 + c_1K_r + c_2K_r^2)}{(1 + y'^2)^{\frac{3}{2}}} = P(x_A - x). \quad (8.2.14)$$

Розділяємо в диференційному рівнянні (8.2.14) змінні і отримаємо

$$\frac{(c_0 + c_1K_r + c_2K_r^2)d(y')}{(1 + y'^2)^{\frac{3}{2}}} = P(x_A - x)dx.$$

Інтегруємо останнє диференційне рівняння

$$\int \frac{(c_0 + c_1K_r + c_2K_r^2)d(y')}{(1 + y'^2)^{\frac{3}{2}}} = \int P(x_A - x)dx.$$

Отримаємо

$$\frac{(c_0 + c_1 K_r + c_2 K_r^2) y'}{(1 + y'^2)^{\frac{1}{2}}} = P(x_A x - \frac{x^2}{2}) + C,$$

де C - будь яка стала інтегрування.

Враховуючи що $y' = tg \varphi$, де φ - кут нахилу дотичної до вісі x , останнє диференціальне рівняння можна представити у вигляді

$$(c_0 + c_1 K_r + c_2 K_r^2) \sin \varphi = P(x_A x - \frac{x^2}{2}) + C. \quad (8.2.15)$$

При $x=0, \varphi=90-\gamma_0$, де γ_0 - кут, на який зменшується загальний кут охоплення ниткою напрямної за рахунок жорсткості нитки на згин для лівої гілки. Тоді стала інтегрування визначається з виразу

$$C = (c_0 + c_1 K_r + c_2 K_r^2) \sin(90 - \gamma_0), \quad C = (c_0 + c_1 K_r + c_2 K_r^2) \cos \gamma_0.$$

Тоді рівняння (8.2.15) прийме вигляд

$$(c_0 + c_1 K_r + c_2 K_r^2) (\sin \varphi - \cos \gamma_0) = P(x_A x - \frac{x^2}{2}). \quad (8.2.16)$$

При $x = x_A, \varphi = \frac{\pi}{2}$. З урахуванням цього рівняння (8.2.16) представимо у вигляді

$$\cos \gamma_0 = 1 - \frac{P x_A^2}{2(c_0 + c_1 K_r + c_2 K_r^2)}.$$

Значення координати x_A знайдемо з умови рівноваги моментів сил відносно точки O

$$(c_0 + c_1 K_r + c_2 K_r^2) / [R + r(1 - \delta_0)] = P x_A, \Rightarrow x_A = \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{P_0 [R + r(1 - \delta_0)]},$$

де δ_0 - початкова деформація перетину нитки в точці A ;

r - розрахунковий радіус перетину нитки.

$$\cos \gamma_0 = 1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2 P_0 [R + r(1 - \delta_0)]^2}. \quad (8.2.17)$$

Виконуючи аналогічні процедури отримаємо вираз для визначення кута γ для правої гілки нитки

$$\cos \gamma = 1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P[R + r(1 - \delta)]^2}, \quad (8.2.18)$$

де δ – деформація перетину нитки в точці B .

Використовуючи залежності (8.2.17) та (8.2.18) визначимо натяг нитки в точках A та B [119]

$$P_B = P \cos \gamma = P \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P[R + r(1 - \delta)]^2} \right], \quad P_A = P_0 \cos \gamma_0 = P_0 \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0[R + r(1 - \delta_0)]^2} \right] \quad (8.2.19)$$

Перейдемо для визначення взаємозв'язку між натягом нитки у точках A та B . На цій ділянці нитка буде взаємодіяти з циліндричною напрямною поверхнею. Враховуючи це можна зробити висновок, що згинаючий момент в будь якому перетині буде постійним. Сумарний кут охоплення ниткою циліндричної напрямної дорівнює $\varphi = \varphi_P + \varphi_{см1} + \varphi_{см2} - \varphi_{иж1} - \varphi_{иж2}$, де φ_P - кут охоплення ниткою напрямної без урахування зминання та жорсткості на згин ($\varphi_P = \pi$); $\varphi_{см1}, \varphi_{см2}$ - кути, на які збільшується кут φ_P за рахунок деформації зминання в зоні контакту нитки з напрямною; $\varphi_{иж1}, \varphi_{иж2}$ - кути, на які зменшується кут φ_P за рахунок наявності жорсткості нитки на згин.

З урахуванням зроблених припущень основна система диференціальних рівнянь (8.2.12) прийме вигляд

$$\begin{aligned} \frac{dP}{dS} - \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N &= 0, \\ P q_1 - N &= 0, \\ N = b_2 E_1 \delta, \quad \delta &= \frac{r - r_x}{r}, \\ q_1 &= \frac{1}{R + r(1 - \delta)}. \end{aligned} \tag{8.2.20}$$

З системи (8.2.20) виключаємо значення питомої нормальної реакції

$$\frac{P}{[R + r(1 - \delta)]} = b_2 E_1 \delta. \tag{8.2.21}$$

Підставляємо (8.2.21) в перше рівняння системи (8.2.20), отримаємо

$$\frac{d}{ds} \{ [R + r(1 - \delta)] b_2 E_1 \delta \} = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b b_2 E_1 \delta.$$

Розкриваємо ліву частину останнього рівняння

$$b_2 E_1 \frac{d\delta}{ds} R + b_2 E_1 \frac{d\delta}{ds} r - 2b_2 E_1 \delta \left(\frac{d\delta}{ds} \right) = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b b_2 E_1 \delta.$$

Спростимо останнє рівняння

$$\frac{d\delta}{ds} (R + r - 2\delta) = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \delta.$$

Представимо диференціал дугової координати у вигляді [5, 6]

$$[R + r(1 - \delta)] d\varphi = ds. \tag{8.2.22}$$

З урахуванням (8.2.22), представимо останнє диференціальне рівняння у вигляді

$$\frac{d\delta}{[R + r(1 - \delta)] d\varphi} (R + r - 2\delta) = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \delta.$$

або

$$\frac{(R+r-2\delta)d\delta}{[R+r(1-\delta)]\delta} = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b d\varphi.$$

Інтегруємо останній вираз

$$\int_{\delta_0}^{\delta} \frac{(R+r-2\delta)d\delta}{[R+r(1-\delta)]\delta} = \int_0^{\varphi} \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b d\varphi.$$

Отримаємо

$$\left[\ln(\delta) - \frac{(r-2)}{r} \ln\left(-\frac{R+r-r\delta}{r}\right) \right]_{\delta_0}^{\delta} = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi \Big|_0^{\varphi}.$$

Підставимо верхні та нижні границі визначених інтегралів

$$\begin{aligned} & \ln(\delta) - \frac{(r-2)}{r} \ln\left(-\frac{R+r-r\delta}{r}\right) - \ln(\delta_0) + \\ & + \frac{(r-2)}{r} \ln\left(-\frac{R+r-r\delta_0}{r}\right) = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi. \end{aligned}$$

Спростимо останній вираз

$$\ln\left[\frac{\delta}{\left(-\frac{R+r-r\delta}{r}\right)^{\frac{r-2}{r}}}\right] - \ln\left[\frac{\delta_0}{\left(-\frac{R+r-r\delta_0}{r}\right)^{\frac{r-2}{r}}}\right] = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi.$$

Використовуючи властивості логарифмів, отримаємо

$$\ln\left[\frac{\delta \left(-\frac{R+r-r\delta_0}{r}\right)^{\frac{r-2}{r}}}{\delta_0 \left(-\frac{R+r-r\delta}{r}\right)^{\frac{r-2}{r}}}\right] = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi.$$

Або

$$\ln\left[\frac{\delta (R+r-r\delta_0)^{\frac{r-2}{r}}}{\delta_0 (R+r-r\delta)^{\frac{r-2}{r}}}\right] = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi.$$

Нехтуючи в останньому виразі добутком малих величин, отримаємо

$$\ln\left(\frac{\delta}{\delta_0}\right) = \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi.$$

Тоді, остаточний вираз закону зміни деформації перетину нитки при взаємодії з ниткоспрямовувачем в'язальної машини має вигляд

$$\delta = \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi}. \quad (8.2.23)$$

З другого рівняння системи (8.2.20) отримаємо

$$P = b_2 E_1 \delta [R + r(1 - \delta)]. \quad (8.2.24)$$

Підставляємо (8.2.23) в перше рівняння системи (8.2.20)

$$\frac{dP}{ds} = \frac{\beta}{\sin \beta} \frac{a R^b}{P_0^b} b_2 E_1 \delta.$$

Або з урахуванням (8.2.23)

$$\frac{dP}{ds} = \frac{\beta}{\sin \beta} \frac{a R^b}{P_0^b} b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi}.$$

Розділяємо змінні в останньому диференційному рівнянні, з урахуванням (8.2.22), отримаємо

$$dP = \frac{\beta}{\sin \beta} \frac{a R^b}{P_0^b} b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} \left[R + r(1 - \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi}) \right] d\varphi.$$

Інтегруємо останнє диференційне рівняння

$$\int_{P_A}^{P_B} dP = \int_0^\varphi (R + r) \frac{\beta}{\sin \beta} \frac{a R^b}{P_0^b} b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} d\varphi - \int_0^\varphi \frac{\beta}{\sin \beta} \frac{a R^b}{P_0^b} b_2 E_1 \delta_0^2 e^{\frac{2\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} d\varphi.$$

В останньому інтегральному рівнянні можна знехтувати квадратами малих величин

$$\int_{P_A}^{P_B} dP = \int_0^\varphi (R+r) \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} d\varphi.$$

Після інтегрування отримаємо

$$P \Big|_{P_A}^{P_B} = (R+r) b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} \Big|_0^\varphi.$$

Підставляємо верхні та нижні значення меж інтегрування

$$P_B - P_A = (R+r) b_2 E_1 \delta_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - (R+r) b_2 E_1 \delta_0.$$

Представимо останнє рівняння у вигляді

$$P_B = P_A \left[1 + \frac{(R+r) b_2 E_1 \delta_0}{P_A} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right].$$

З урахування (8.2.24), закон зміни натягу нитки на дузі охоплення ниткоспрямовувача AB прийме вигляд

$$P_B = P_A \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right]. \quad (8.2.25)$$

Вирішуємо сумісно (8.2.25) та (8.2.19), отримаємо

$$P \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P[R+r(1-\delta)]^2} \right] = P_0 \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0[R+r(1-\delta_0)]^2} \right] \times \\ \times \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right].$$

В остаточному вигляді маємо

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right] + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right] - \\ - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right]. \quad (8.2.26)$$

Кут $\varphi = \varphi_P + \varphi_{cm1} + \varphi_{cm2} - \varphi_{уж1} - \varphi_{уж2}$, можна розрахувати з використанням залежностей [5, 6, 116]

$$\begin{aligned} \varphi_{cm1} &= \arccos\left[1 - \delta_0 \left(\frac{2r}{R}\right)^2\right], \quad \varphi_{cm2} = \arccos\left[1 - \delta \left(\frac{2r}{R}\right)^2\right], \\ \varphi_{уж1} &= \arccos\left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0(R+r)^2}\right], \\ \varphi_{уж2} &= \arccos\left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P(R+r)^2}\right]. \end{aligned} \quad (8.2.27)$$

В таблиці 8.2.1 представлені результати розрахунків по (8.2.26) вихідного натягу нитки після ниткоспрямовувача в залежності від вхідного натягу та радіусу кривизни поверхні ниткоспрямовувача в зоні контакту. Для розрахунку обрано: капронову комплексну нитку 28 текс ($a = 0,1110$, $b = 0,1186$), $B_{KH} = c_0 + c_1 K_r + c_2 K_r^2 = 0,22$ сН·мм², $r = 0,1$ мм, $\beta = 5^\circ$, $b_2 = 0,012$ мм, $E_1 = 584,4$ сН/мм²; віскозні комплексні нитки 16,7 текс ($a = 0,1111$, $b = 0,1160$), $B_{BH} = c_0 + c_1 K_r + c_2 K_r^2 = 0,2$ сН·мм², $r = 0,085$ мм, $\beta = 5^\circ$, $b_2 = 0,014$ мм, $E_1 = 474,9$ сН/мм²; вовняну пряжу 29,9 текс ($a = 0,1090$, $b = 0,0910$), $B_{ВП} = c_0 + c_1 K_r + c_2 K_r^2 = 0,1$ сН·мм², $r = 0,115$ мм, $\beta = 5^\circ$, $b_2 = 0,015$ мм, $E_1 = 599,0$ сН/мм²; бавовняну пряжу 27,6 текс ($a = 0,1060$, $b = 0,0590$), $B_{ВП} = c_0 + c_1 K_r + c_2 K_r^2 = 0,7$ сН·мм², $r = 0,11$ мм, $\beta = 5^\circ$, $b_2 = 0,012$ мм, $E_1 = 712,2$ сН/мм²; віскозну штапельну пряжу 28,8 текс ($a = 0,107$, $b = 0,0690$), $B_{ВШП} = c_0 + c_1 K_r + c_2 K_r^2 = 0,4$ сН·мм², $r = 0,11$ мм, $\beta = 5^\circ$, $b_2 = 0,014$ мм, $E_1 = 1254,8$ сН/мм² [5, 116]. Вхідний натяг змінювався в межах $P_0 = 5 \div 40$ сН, а радіус кривизни напрямної змінювався у межах $R = 0,6 \div 11,2$ мм.

Таблиця 8.2.1

Залежність натягу нитки після ниткоспрямовувача в залежності від вхідного натягу та радіусу кривизни поверхні ниткоспрямовувача в зоні контакту

R , мм	Натяг нитки після ниткоспрямовувача P , сН	
	Натяг веденої гілки $P_0 = 10$ сН	Натяг веденої гілки $P_0 = 20$ сН

	<i>КН</i>	<i>ВН</i>	<i>ВП</i>	<i>БП</i>	<i>ВШ</i> <i>П</i>	<i>КН</i>	<i>ВН</i>	<i>ВП</i>	<i>БП</i>	<i>ВШ</i> <i>П</i>
0,6	19.1 4	17.4 8	16.5 6	18.1 4	15.75	58.2.8 1	45.8 6	40.4 8	49.79	35.47
1,1	16.6	15.6 3	14.9 4	16.4 7	15.13	34.94	32.5 1	30.9 2	35.72	31.21
1,6	16.2 5	15.3 9	14.6 8	16.1 8	15.08	32.77	30.9 3	29.5 6	38.2.5 1	30.46
2,1	16.2 2	15.3 9	14.6 3	16.1 4	15.13	32.18	30.5 2	29.1 6	32.81	30.27
2,6	16.2 8	15.4 6	14.6 5	16.1 6	15.19	32.02	30.4 2	29.0 3	32.54	30.24
3,1	16.3 7	15.5 3	14.6 8	16.2 1	15.26	32.02	30.4 4	29	32.44	30.27
3,6	16.4 6	15.6 2	14.7 3	16.2 6	15.32	32.08	30.5 1	29.0 2	32.42	30.32
4,1	16.5 5	15.7	14.7 7	16.3 1	15.38	32.17	30.6	29.0 6	32.43	30.39
4,6	16.6 4	15.7 8	14.8 2	16.3 6	15.43	32.28	30.7	29.1 1	32.46	30.45
5,1	16.7 3	15.8 5	14.8 6	16.4	15.48	32.4	30.8	29.1 6	32.5	30.52
5,6	16.8 2	15.9 2	14.9	16.4 5	15.52	32.52	30.9 1	29.2 2	32.55	30.58
6,1	16.8 9	15.9 9	14.9 4	16.4 9	15.57	32.63	31.0 1	29.2 8	32.6	30.65
6,6	16.9 7	16.0 6	14.9 8	16.5 3	15.61	32.75	31.1 1	29.3 3	32.65	30.71
7,1	17.0 4	16.1 2	15.0 1	16.5 7	15.64	32.86	31.2	29.3 9	32.7	30.77
7,6	17.1 1	16.1 7	15.0 5	16.6 1	15.68	32.96	31.2 9	29.4 4	32.75	30.82
8,1	17.1 8	16.2 3	15.0 8	16.6 4	15.71	38.2.0 7	31.3 8	29.5	32.79	30.88
8,6	17.2 4	16.2 8	15.1 1	16.6 7	15.74	38.2.1 7	31.4 7	29.5 5	32.84	30.93
9,1	17.3	16.3 3	15.1 4	16.7	15.77	38.2.2 7	31.5 5	29.6	32.89	30.98
9,6	17.3 6	16.3 8	15.1 7	16.7 3	15.8	38.2.3 6	31.6 3	29.6 5	32.93	31.02

10, 1	17.4 2	16.4 3	15.2	16.7 6	15.83	38.2.4 6	31.7 1	29.6 9	32.98	31.07
10, 6	17.4 7	16.4 8	15.2 3	16.7 9	15.85	38.2.5 5	31.7 9	29.7 4	38.2.0 2	31.11

По даним таблиці 8.2.1 побудовані графічні залежності натягу нитки після ниткоспрямовувача в залежності від вхідного натягу та радіусу кривизни поверхні ниткоспрямовувача в зоні контакту, які представлені на рис.8.2.5.

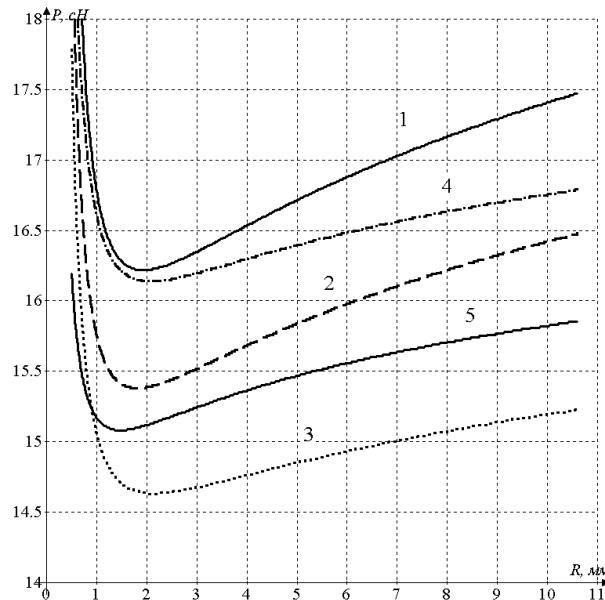


Рис.8.2.5. Графічні залежності натягу нитки після ниткоспрямовувача: 1- капронова комплексна нитка 28 текс ; 2 - віскозній комплексній нитки 16,7 текс ; 3 - вовняної пряжі 29,9 текс ; 4 - бавовняної пряжі 27,6 текс ; 5 - віскозної штапельної пряжі 28,8 текс

Аналіз графічних залежностей показав, що для кожної нитки вихідний натяг при збільшенні значення радіусу кривизни зменшується на ділянці від 0.5 мм до деякого значення, конкретного для кожної нитки чи пряжі. Після цього плавно зростає[2]. Така поведінка для першої ділянки пояснюється впливом зминання нитки в зоні контакту. Плавне зростання

натягу на другій ділянці пов'язане зі зростанням площі контакту[2, 5, 6, 11]. Такий закон зміни натягу ставить питання про оптимізацію натягу нитки на в'язальній машині за рахунок підбору таких геометричних параметрів ниткопрямувачів, при яких натяг був мінімальним. В додатку А представлені результати визначення натягу в залежності від натягу веденої гілки P_0 , радіусу кривизни робочої поверхні ниткопрямувача R та кута радіального охоплення β .

Представляють інтерес дослідження про вплив кута радіального охоплення β на значення натягу нитки після ниткопрямувача (рис.8.2.3). На рис.8.2.6 представлені графічні залежності зміни натягу для різних значень цього кута.

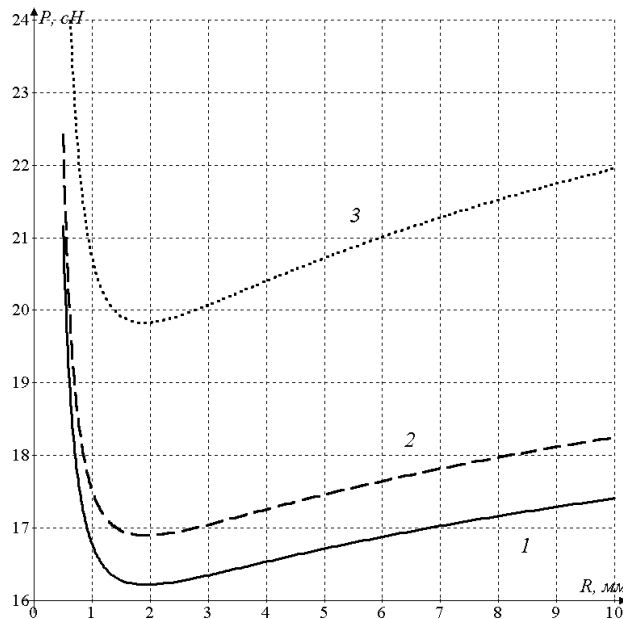


Рис.8.2.8.2. Графічні залежності зміни натягу в залежності від кута радіального охоплення β для капронової комплексної нитки 28 текс : 1- $\beta = 5^\circ$;
 2 - $\beta = 40^\circ$; 3 - $\beta = 80^\circ$

Аналіз отриманих даних показав, що для оптимального значення радіусу кривизни, коли натяг має мінімальне значення, для кута $\beta = 5^\circ$ натяг вихідної гілки дорівнював 16.22 сН, для кута $\beta = 40^\circ$ натяг вихідної гілки дорівнював 16.9 сН; для кута $\beta = 80^\circ$ натяг вихідної гілки дорівнював

19.83 сН. Таким чином, зростання складо 22%. Виходячі з геометрії конструкції ниткопрямовувача (рис.8.2.3) з одного боку зменшення геометричних розмірів призведе до зростання вихідного натягу, а з другого боку при зростанні геометричних розмірів будуть виникати додаткові затрати матеріалу на виготовлення ниткопрямовувача.

Математична формалізація критерію оптимізації розмірів ниткопрямовувача

Оптимізація геометричних параметрів ниткопрямовувачів повинна базуватися на комплексних теоретико-експериментальних дослідженнях процесу взаємодії ниток з направляючими і робочими органами в'язальних машин, які ляжуть в основу вибору оптимальних технологічних параметрів заправки, що дозволить стабілізувати технологічний процес вироблення трикотажного полотна, характер взаємодії ниток у робочій зоні, а отже, зменшити обривність ниток.

Вихідна система рівнянь для визначення оптимальних розмірів ниткопрямовувачів, з урахуванням (8.2.26)-(8.2.27), має вигляд

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right] + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right]. \quad (8.2.28)$$

$$\varphi_{см1} = \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{см2} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right],$$

$$\varphi_{уж1} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0(R+r)^2} \right], \quad \varphi_{уж2} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P(R+r)^2} \right].$$

Для знаходження екстремуму необхідно знайти похідну системи рівнянь (8.2.28), яка повинна буде дорівнювати нулю та розв'язати отримане рівняння відносно R

$$\frac{dP}{dR} = 0. \quad (8.2.29)$$

Для цього представимо систему (8.2.28) у вигляді

$$\begin{aligned}
 P &= P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right] + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right] - \\
 &- \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right], \\
 m1(R) &= P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right], \\
 m2(R) &= \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right], \\
 m3(R) &= \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right], \\
 m4(R) &= \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right],
 \end{aligned} \tag{8.2.30}$$

$$\varphi = \varphi_P + \varphi_{CM1} + \varphi_{CM2} - \varphi_{ИЖ1} - \varphi_{ИЖ2},$$

$$\varphi_{cm1} = \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{cm2} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right],$$

$$\varphi_{иж1} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0(R+r)^2} \right], \quad \varphi_{иж2} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P(R+r)^2} \right].$$

Тоді цільовий функціонал (8.2.30) прийме вигляд

$$P = m1(R) + m2(R) - m3(R)m4(R).$$

Знайдемо похідну від останньої функції та прирівняємо її до нуля

$$\frac{dP}{dR} = \frac{d[m1(R)]}{dR} + \frac{d[m2(R)]}{dR} - m3(R) \frac{d[m4(R)]}{dR} - \frac{d[m3(R)]}{dR} m4(R) = 0 \tag{8.2.31}$$

Для визначення оптимальних розмірів ниткоспрямовувача необхідно знайти відповідні похідні від функцій $m1(R)$, $m2(R)$, $m3(R)$, $m4(R)$ та підставити їх у рівняння (8.2.31).

Для функції $m1(R)$ отримаємо наступну систему

$$\begin{aligned}
 \frac{d[m1(R)]}{dR} &= P_0 \left\{ \frac{e^{W(R)\varphi(R)} - 1}{R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right]} - \right. \\
 &\quad \left. - \frac{\left[\frac{P_0 r}{E_1 b_2 (R+r)^2} + 1 \right] (e^{W(R)\varphi(R)} - 1) (R+r)}{\left[R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right] \right]^2} + \right. \\
 &\quad \left. + \frac{W(R) \frac{d[\varphi(R)]}{dR} e^{W(R)\varphi(R)} (R+r) [W(R) \frac{d[\varphi(R)]}{dR} + \frac{d[W(R)]}{dR} \varphi(R)]}{R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right]} \right\}, \\
 W(R) &= \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b, \\
 \frac{dW(R)}{dR} &= \frac{\beta R^{b-1} a b}{P_0^b \sin \beta}, \\
 \frac{d[\varphi(R)]}{dR} &= \frac{2(c_0 + c_1 K_r + c_2 K_r^2)}{P_0 (R+r)^3 \sqrt{1 - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0 (R+r)^2} - 1 \right]^2}} - \\
 &\quad - \frac{2 \left[\frac{4P_0 r^2}{E_1 R^2 b_2 (R+r)^2} + \frac{8P_0 r^2}{E_1 R^3 b_2 (R+r)} \right]}{\sqrt{1 - \left[\frac{4P_0 b_2}{E_1 R^2 r^2 (R+r)} - 1 \right]^2}}.
 \end{aligned} \tag{8.2.32}$$

Для функції $m2(R)$ отримаємо

$$\frac{d[m2(R)]}{dR} = \frac{(c_0 + c_1 K_r + c_2 K_r^2) \left[\frac{P_0 r}{E_1 b_2 (R+r)^2} + 1 \right]}{\left\{ R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right] \right\}^3}. \tag{8.2.33}$$

Для функції $m3(R)$ отримаємо

$$\frac{d[m3(R)]}{dR} = \frac{(c_0 + c_1 K_r + c_2 K_r^2) \left[\frac{P_0 r}{E_1 b_2 (R+r)^2} + 1 \right]}{\left\{ R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right] \right\}^3}. \tag{8.2.34}$$

Для функції $m4(R)$ отримаємо

$$\begin{aligned}
 \frac{d[m_4(R)]}{dR} = & \left\{ \frac{e^{W(R)\varphi(R)} - 1}{R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right]} - \right. \\
 & \frac{\left[\frac{P_0 r}{E_1 b_2 (R+r)^2} + 1 \right] (e^{W(R)\varphi(R)} - 1) (R+r)}{\left[R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right]^2} + \right. \\
 & \left. \left. + \frac{W(R) \frac{d[\varphi(R)]}{dR} e^{W(R)\varphi(R)} (R+r) \left[W(R) \frac{d[\varphi(R)]}{dR} + \frac{d[W(R)]}{dR} \varphi(R) \right]}{R - r \left[\frac{P_0}{E_1 b_2 (R+r)} - 1 \right]} \right\}, \\
 W(R) = & \frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b, \\
 \frac{dW(R)}{dR} = & \frac{\beta R^{b-1} a b}{P_0^b \sin \beta}, \\
 \frac{d[\varphi(R)]}{dR} = & \frac{2(c_0 + c_1 K_r + c_2 K_r^2)}{P_0 (R+r)^3 \sqrt{1 - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0 (R+r)^2} - 1 \right]^2}} - \\
 & \frac{2 \left[\frac{4P_0 r^2}{E_1 R^2 b_2 (R+r)^2} + \frac{8P_0 r^2}{E_1 R^3 b_2 (R+r)} \right]}{\sqrt{1 - \left[\frac{4P_0 b_2}{E_1 R^2 r^2 (R+r)} - 1 \right]^2}}.
 \end{aligned} \tag{8.2.35}$$

Отримані залежності (8.2.30), (8.2.32)-(8.2.35) підставляємо в (8.2.31). Отримаємо трансцендентне рівняння відносно радіуса R кривизни ниткоспрямовувачів. Знаходити корені трансцендентного рівняння будемо за допомогою метода ділення відрізка $[a, b]$ навпіл (метод дихотомії)[5, 6, 116].

Суть методу полягає в побудові ітераційної послідовності вкладених один в одного відрізків $[a_n, b_n]$, кінці яких представляють монотонні послідовності $\{a_n\}, \{b_n\}$, при чому

$$a_n \leq \xi, \quad b_n \geq \xi, \quad n = 1, 2, \dots$$

де ξ - корінь трансцендентного рівняння на відрізку $[a, b]$.

Збіжність даного методу повільна. Проте при будь-якій ширині відрізку $[a, b]$ збіжність гарантована [116].

Будемо рахувати, що $f(a)f(b) < 0$ и $f(a) < 0, f(b) > 0$. Тоді візьмемо середню точку відрізку $[a, b]$

$$\xi_1 = \frac{a + b}{2}.$$

Обчислюємо в даній точці значення функції $f(x)$. Припустимо, що $f(\xi_1) = 0$. В цьому випадку процес визначення кореня трансцендентного рівняння закінчується. Якщо $f(\xi_1) \neq 0$, то розглядаємо два відрізка $[a, \xi_1]$ и $[\xi_1, b]$. Вибираємо той з них, де виконується умова $F(a)f(\xi_1) < 0$ або $f(\xi_1)f(b) < 0$. Вибраний відрізок знову ділимо навпіл приймаючи $a_1 = a, b_1 = \xi_1$ або $a_1 = \xi_1, b_1 = b$

$$\xi_2 = \frac{a_1 + b_1}{2}.$$

Необмежене продовження процесу ділення дозволяє отримати ітераційну послідовність вкладених один в одного відрізків $[a_n, b_n]$, причому

$$a_n \leq a_{n+1} < b_{n+1} \leq b_n.$$

Ліві кінці відрізків утворюють ітераційну, монотонну послідовність $\{a_n\}$, яка в межі представлятиме деяку величину z_1

$$\lim_{n \rightarrow \infty} a_n = z_1,$$

а праві кінці відрізків утворюють ітераційну, монотонну послідовність $\{b_n\}$, яка в межі представлятиме деяку величину z_2

$$\lim_{n \rightarrow \infty} b_n = z_2.$$

Очевидно, що

$$\begin{aligned} a_n &\leq z_1 \leq z_2 \leq b_n, \\ z_2 - z_1 &\leq b_n - a_n = \frac{b-a}{2^n}. \end{aligned} \quad (8.2.36)$$

З останнього рівняння системи (8.2.36) отримуємо $z_2 - z_1 = 0$, тобто дана різниця менше будь-якого наперед заданого позитивного числа. Звідки матимемо $z_2 = z_1 = \xi$. Отже

$$f(\xi) = \lim_{n \rightarrow \infty} f(a_n) = \lim_{n \rightarrow \infty} f(b_n).$$

З урахуванням (8.2.36)

$$a_n \leq \xi \leq b_n.$$

Аналіз останньої нерівності показує, що величина a_n визначає шуканий корінь ξ з недоліком, а величина b_n з лишком [11]. Помилка при цьому не перевищує довжину відрізка $b_n - a_n$ і прагне до нуля при збільшенні n за законом геометричної прогресії із знаменником $1/2$. Число ітерацій N , необхідне для досягнення заданої точності ε визначається з виразу

$$N \cong \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2}.$$

В результаті розрахунків отримано наступні оптимальні значення радіусу кривизни ниткопрямувачів: 1,91 мм – для капронової комплексної нитки 28 текс; 1,8 мм – для віскозної комплексної нитки 16,7

текс; 2,18 мм – для вовняної пряжі 29,9 текс; 2,02 мм – для бавовняної пряжі 27,6 текс; 1,59 мм – для віскозної штапельної пряжі 28,8 текс (рис.8.2.4).

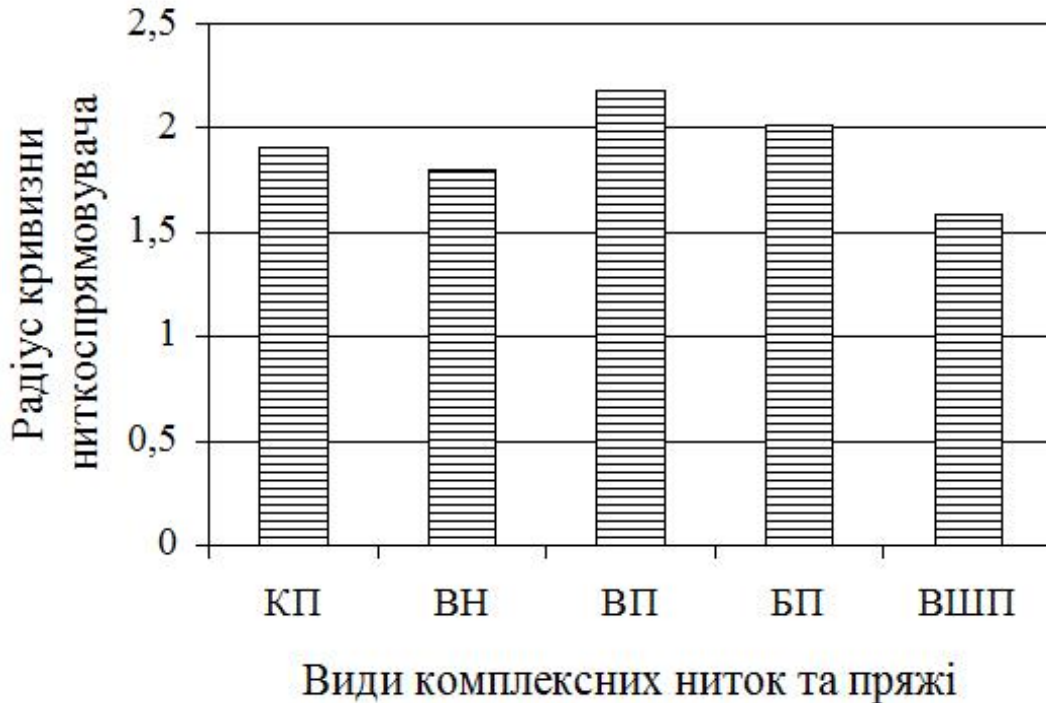


Рис.8.2.4. Оптимальні значення радіуса кривизни ниткопрямувачів для різних ниток

Дослідження взаємодії нитки з направляючими в точках зламу лінії заправки з урахуванням змінання

Результуюча система диференціальних рівнянь (8.2.12) в проекції на осі головного тригранника, що описує рух комплексних ниток та пряжі малого крутіння, з урахуванням їх реальних фізико-механічних властивостей, по направляючій поверхні має вигляд [11, 12, 22]

$$\begin{aligned}
 & \frac{\partial P}{\partial S} - \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N \cos \varphi_t = T \times \\
 & \times \left[\frac{\partial(V_\tau + U'_\tau)}{\partial t} + (\omega_{0n} + \omega_{un})(V_b + U'_b) - (\omega_{0b} + \omega_{ub})(V_n + U'_n) \right]; \\
 & P q_1 - N = T \times \\
 & \times \left[\frac{\partial(V_n + U'_n)}{\partial t} - (\omega_{0\tau} + \omega_{u\tau})(V_b + U'_b) + (\omega_{0b} + \omega_{ub})(V_\tau + U'_\tau) \right]; \\
 & P p_1 + \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N \sin \varphi_t = T \times \\
 & \times \left[\frac{\partial(V_b + U'_b)}{\partial t} + (\omega_{0\tau} + \omega_{u\tau})(V_n + U'_n) - (\omega_{0n} + \omega_{un})(V_\tau + U'_\tau) \right]; \\
 & \frac{\partial M_\kappa}{\partial S} + M_\tau = \gamma_n [J_{\tau_0} \pm \Delta_\tau(\vec{U})] \varepsilon_1; \\
 & \frac{\partial M_{u_2}}{\partial S} + M_k q_1 + M_n = \gamma_n [J_{n_0} \pm \Delta_n(\vec{U})] \varepsilon_2; \\
 & -M_k p_1 + r_x \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N = \gamma_n [J_{b_0} \pm \Delta_b(\vec{U})] \varepsilon_3, \\
 & N = b_2 E_1 \delta (1 - b_3 \delta^{b_4}) + \eta \delta^{b_5} (1 - b_5 \delta^{b_6}), \quad \delta = \frac{r - r_x}{r}, \quad \dot{\delta} = \frac{\partial U_v}{r \partial t}, \\
 & q_1 = \frac{\cos \Psi_0}{\rho_0} + \frac{\partial \beta}{\partial S} + \frac{\alpha}{\rho_{01}} + \alpha \frac{\partial \Psi_0}{\partial S}, \quad p_1 = \frac{\sin \Psi_0}{\rho_0} + \frac{\partial \alpha}{\partial S} - \frac{\beta}{\rho_{01}} - \beta \frac{\partial \Psi_0}{\partial S}; \\
 & r_1 = \frac{1}{\rho_0} + \frac{\partial \Psi_0}{\partial S} - \alpha \frac{\cos \Psi_0}{\rho_0} + \frac{\sin \Psi_0}{\rho_0} \beta, \quad \gamma = \frac{\partial U_\tau}{\partial S} - \frac{\cos \Psi_0}{\rho_0} U_v + \frac{\sin \Psi_0}{\rho_0} U_\beta; \\
 & \beta = \frac{\partial U_v}{\partial S} + \frac{\cos \Psi_0}{\rho_0} U_\tau - \frac{U_\beta}{\rho_{01}} - U_\beta \frac{\partial \Psi_0}{\partial S}, \quad \alpha = \frac{\sin \Psi_0}{\rho_0} U_\tau - \frac{\partial U_\beta}{\partial S} - \frac{U_v}{\rho_{01v}} - U_v \frac{\partial \Psi_0}{\partial S}.
 \end{aligned} \tag{8.2.37}$$

Перейдемо для визначення взаємозв'язку між натягом нитки у точках набігання та збігання з напрямної поверхні з урахуванням зминання. Крутіння комплексних ниток та пряжі низьке, тому жорсткістю нитки на згин можна знехтувати [2, 5]. Силами інерції, враховуючи масу елемента нитки, яка взаємодіє з направляючою, можна знехтувати. На цій ділянці нитка буде взаємодіяти з циліндричною напрямною поверхнею ниткопрямовувача з радіальним охопленням в спрямляючій площині. Враховуючи це можна зробити висновок, що згинаючий момент буде рівним нулю. Сумарний кут охоплення ниткою циліндричної напрямної

дорівнює $\varphi = \varphi_P + \varphi_{CM1} + \varphi_{CM2}$, де φ_P - кут охоплення ниткою напрямної без урахування зминання та жорсткості на згин ($\varphi_P = \pi$); $\varphi_{CM1}, \varphi_{CM2}$ - кути, на які збільшується кут φ_P за рахунок деформації зминання в зоні контакту нитки з напрямною.

З урахуванням зроблених припущень основна система диференціальних рівнянь (8.2.37) прийме вигляд

$$\begin{aligned} \frac{dP}{dS} - \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N &= 0, \\ P q_1 - N &= 0, \\ N &= b_2 E_1 \delta, \quad \delta = \frac{r - r_x}{r}, \\ q_1 &= \frac{1}{R + r(1 - \delta)}. \end{aligned} \tag{8.2.38}$$

Використовуючи методику, яка викладена в підрозділі 8.2.2, з системи диференціальних рівнянь (8.2.38) отримаємо залежність для натягу вихідної гілки нитки з урахуванням зминання в зоні контакту

$$\begin{aligned} P &= P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} \left(e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1 \right) \right], \\ \varphi &= \varphi_P + \varphi_{CM1} + \varphi_{CM2}, \\ \varphi_{CM1} &= \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{CM2} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right]. \end{aligned} \tag{8.2.39}$$

Математична формалізація критерію оптимізації розмірів ниткоспрямовувача з урахуванням зминання в зоні контакту

Для знаходження математичної форми критерію оптимізації геометричних розмірів ниткоспрямовувача необхідно вихідну систему рівнянь (8.2.39), для визначення оптимальних розмірів ниткоспрямовувачів

з урахуванням змінання в зоні контакту, диференціювати та прирівняти отриманий вираз нулю. Представимо систему рівнянь (8.2.39) у вигляді

$$\begin{aligned} P &= P_0 \left[1 + \frac{(R+r)}{m11} m22 \right], \\ m11 &= R + r(1 - \delta_0), \\ m22 &= e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} - 1, \end{aligned} \quad (8.2.40)$$

$$\varphi(R) = \varphi_P + \varphi_{CM1} + \varphi_{CM2},$$

$$\varphi_{cm1} = \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{cm2} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right].$$

Знайдемо похідну від останньої функції та прирівняємо її до нуля

$$\frac{dP}{dR} = P_0 \left[\frac{m22(R)}{m11(R)} + \frac{\frac{d[m22(R)]}{dR} (R+r)}{m11(R)} - \frac{\frac{d[m11(R)]}{dR} m22(R) (R+r)}{m11(R)^2} \right] = 0. \quad (8.2.41)$$

Для визначення оптимальних розмірів ниткоспрямовувача необхідно знайти відповідні похідні від функцій $m11(R)$, $m22(R)$ та підставити їх у рівняння (8.2.41).

Для функції $m11(R)$ отримаємо наступну систему

$$\frac{d[m11(R)]}{dR} = 1 + \frac{P_0 r}{E_1 b_2 (R+r)^2}. \quad (8.2.42)$$

Для функції $m22(R)$ отримаємо наступну систему

$$\begin{aligned} \frac{d[m22(R)]}{dR} &= e^{\frac{R^b a \beta \varphi(R)}{P_0^b \sin \beta}} \left[\frac{R^b a \beta \frac{d[\varphi(R)]}{dR}}{P_0^b \sin \beta} + \frac{R^{b-1} a \beta b \varphi(R)}{P_0^b \sin \beta} \right], \\ \frac{d[\varphi(R)]}{dR} &= - \frac{2 \left[\frac{4 P_0 r^2}{E_1 R^2 b_2 (R+r)^2} + \frac{8 P_0 r^2}{E_1 R^3 b_2 (R+r)} \right]}{\sqrt{1 - \left[\frac{4 P_0 b_2}{E_1 R^2 r^2 (R+r)} - 1 \right]^2}}. \end{aligned} \quad (8.2.43)$$

Отримані залежності (8.2.42) та (8.2.43) підставляємо в (8.2.41). Отримаємо трансцендентне рівняння відносно радіуса R кривизни ниткоспрямовувачів. Знаходити корені трансцендентного рівняння будемо за допомогою метода ділення відрізка навпіл [2].

Взаємодія жорсткої на згин нитки з направляючими з урахуванням жорсткості на згин

В процесі переробки монониток та комплексних ниток та пряжі високого крутіння, при визначенні натягу нитки після направляючих, необхідно враховувати коефіцієнт жорсткості на згин, а змінанням нехтувати. Виходячи з цього, з урахуванням зроблених припущень основна система диференціальних рівнянь (8.2.12) прийме вигляд

$$\begin{aligned} \frac{dP}{dS} - \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} N &= 0, \\ P q_1 - N &= 0, \\ q_1 &= \frac{1}{R+r}. \end{aligned} \tag{8.2.44}$$

Сумісно вирішуючи друге та третє рівняння системи (8.2.44) визначимо

$$N = \frac{P}{R+r}.$$

Підставляємо отриманий результат в перше рівняння системи (8.2.44)

$$\frac{dP}{dS} = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \frac{P}{(R+r)}.$$

Отримали диференціальне рівняння з розподіленими змінними. Розділяємо змінні та отримаємо наступне диференціальне рівняння

$$\frac{dP}{P} = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \frac{ds}{(R+r)}.$$

Враховуючи, що [116]

$$ds = (R+r)d\varphi,$$

де φ – поточний кут охоплення жорсткою на згин напрямної поверхні, представимо диференційне рівняння у вигляді

$$\frac{dP}{P} = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} d\varphi.$$

Інтегруємо останнє диференційне рівняння

$$\int_{P_A}^{P_B} \frac{dP}{P} = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \int_0^\varphi d\varphi.$$

де P_A, P_B – натяг нитки в точках входу та сходу з напрямної поверхні (див. рис. 8.2.4) [2, 5].

Після інтегрування маємо

$$\ln(P)|_{P_A}^{P_B} = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \varphi|_0^\varphi.$$

Підставляємо верхню та нижню межу інтегрування, отримаємо

$$\ln(P_B) - \ln(P_A) = \frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \varphi. \quad (8.2.45)$$

Визначимо з останнього рівняння залежність між натягами в точках A та B

$$P_B = P_A e^{\frac{\beta}{\sin \beta} \frac{aR^b}{P_0^b} \varphi}. \quad (8.2.45)$$

Для знаходження величини натягу нитки після ниткоспрямовувача необхідно скористатися рівняннями (8.2.19). Вирішуючи сумісно (8.2.45) та (8.2.19), отримаємо

$$P \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P(R+r)^2} \right] = P_0 \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0(R+r)^2} \right] e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi}.$$

В остаточному вигляді маємо

$$P = P_0 e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi} + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2(R+r)^2} \right] (1 - e^{\frac{\beta}{\sin \beta} \frac{a}{P_0^b} R^b \varphi}). \quad (8.2.46)$$

Кут $\varphi = \varphi_P - \varphi_{уж1} - \varphi_{уж2}$, можна розрахувати з використанням залежностей [5, 6, 116]

$$\varphi_{уж1} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0(R+r)^2} \right], \quad (8.2.47)$$

$$\varphi_{уж2} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P(R+r)^2} \right].$$

Результати розрахунків за формулами (8.2.46) та (8.2.47) представлені в таблиці 8.2.2 та на рис.8.2.5. В таблиці показані результати розрахунку натягу нитки в залежності від коефіцієнту жорсткості на згин, вхідного натягу та радіусу кривизни поверхні в зоні контакту.

Розрахунки проводилися для капронової комплексної нитки 28 текс з різною ступеню крутіння. У роботі [5] встановлена квадратична залежність коефіцієнту жорсткості на згин від крутіння нитки. В таблиці в стовпчику за номером 1 наведені значення натягу для коефіцієнту жорсткості на згин $B=0$ сНмм² (коефіцієнт крутіння $K_r=0$), 2 - $B=0,5$ сНмм², 3 - $B=3$ сНмм², 4 - $B=5$ сНмм², 5 - $B=11,2$ сНмм² (це значення коефіцієнту жорсткості відповідає капроновій мононитці). Розрахунки проводилися для наступних даних $a=0,1765$, $b=0,1186$, $r=0,1$ мм, $\beta=5^\circ$.

Аналіз графічних залежностей на рис.8.2.5 показує, що зростання коефіцієнту жорсткості на згин призводить до зменшення натягу при зростанні радіусу кривизни. Це пов'язано зі зменшенням кута охоплення ниткою напрямної, як це видно з формули (8.2.47). Подальше збільшення коефіцієнту жорсткості на згин призведе к неможливості здійснювати

охоплення ниткою напрямної. При збільшенні вхідного натягу P_0 контакт буде забезпечений.

Удосконалення конструкції шайбового нитконатягувача, як складового елементу системи ниткоподачі круглов'язальної машини

Нитконатягувальні пристрої є складовою частиною системи ниткоподачі на круглов'язальних машинах[1-6, 28]. Вони призначені для забезпечення відповідного натягу нитки при вході у робочу зону в'язання. Найбільш поширеними є шайбові нитконатягувальні пристрої[28]. Їх конструкція складається з стрижня, на який одягнені дві фігурні шайби. Відповідний тиск на шайби забезпечується за допомогою спіральної пружини та гайки, яка накручується на стрижень. Закручуючи гайку збільшуємо силу тиску верхньої шайби на нижню.

Таблиця 8.2.2

Залежність натягу жорсткої на згин капронової нитки 28 текс нитки після ниткоспрямовувача від коефіцієнту жорсткості на згин, вхідного натягу та радіусу кривизни поверхні в зоні контакту

R , мм	Натяг нитки після ниткоспрямовувача P , сН									
	Натяг веденої гілки $P_0=5$ сН					Натяг веденої гілки $P_0=10$ сН				
	1	2	3	4	5	1	2	3	4	5
0,6	7.7 1	6.6 2	5.2 3	5 4	5.0 1	14.8 8	18.2.5 3	11.49	10.68	10.05
1,1	7.9 5	7.2 7	6.2 3	5.7 6	5.0 8	15.3 3	14.49	18.2.1 6	12.5	11.2
1,6	8.1 2	7.6 2	6.8 3	6.4 5	5.6 9	15.6 3	15.01	14.04	18.2.5 4	12.46
2,1	8.2 5	7.8 5	7.2 2	6.9 1	6.2 4	15.8 6	15.37	14.59	14.2	18.2.3 2
2,6	8.3 6	8.0 2	7.5	7.2 3	6.6 5	16.0 5	15.64	14.99	14.66	18.2.9 3
3,1	8.4 5	8.1 6	7.7 1	7.4 8	6.9 7	16.2 1	15.85	15.3	15.02	14.38

3,6	8.5 3	8.2 7	7.8 8	7.6 8	7.2 3	16.3 5	16.04	15.55	15.3	14.75
4,1	8.6	8.3 7	8.0 2	7.8 4	7.4 3	16.4 8	16.19	15.76	15.54	15.04
4,6	8.6 6	8.4 5	8.1 3	7.9 7	7.6 1	16.5 9	16.33	15.94	15.74	15.29
5,1	8.7 2	8.5 3	8.2 4	8.0 9	7.7 6	16.7	16.46	16.1	15.92	15.5
5,6	8.7 8	8.6	8.3 3	8.1 9	7.8 8	16.7 9	16.57	16.24	16.07	15.69
6,1	8.8 3	8.6 6	8.4 1	8.2 8	8	16.8 8	16.67	16.37	16.21	15.86
6,6	8.8 7	8.7 2	8.4 8	8.3 7	8.1	16.9 6	16.77	16.48	16.34	16.01
7,1	8.9 2	8.7 7	8.5 5	8.4 4	8.1 9	17.0 4	16.86	16.59	16.45	16.14
7,6	8.9 6	8.8 2	8.6 1	8.5 1	8.2 7	17.1 2	16.94	16.69	16.56	16.27
8,1	9	8.8 7	8.6 7	8.5 7	8.3 5	17.1 9	17.02	16.78	16.66	16.38
8,6	9.0 4	8.9 1	8.7 3	8.6 3	8.4 2	17.2 5	17.1	16.87	16.75	16.49
9,1	9.0 7	8.9 5	8.7 8	8.6 9	8.4 8	17.3 2	17.17	16.95	16.84	16.59
9,6	9.1 1	8.9 9	8.8 2	8.7 4	8.5 4	17.3 8	17.23	17.02	16.92	16.68
10, 1	9.1 4	9.0 3	8.8 7	8.7 9	8.6	17.4 3	17.3	17.1	16.99	16.77
10, 6	9.1 7	9.0 7	8.9 1	8.8 3	8.6 5	17.4 9	17.36	17.16	17.07	16.85
11, 1	9.2	9.1	8.9 5	8.8 8	8.7 1	17.5 4	17.42	17.23	17.14	16.93

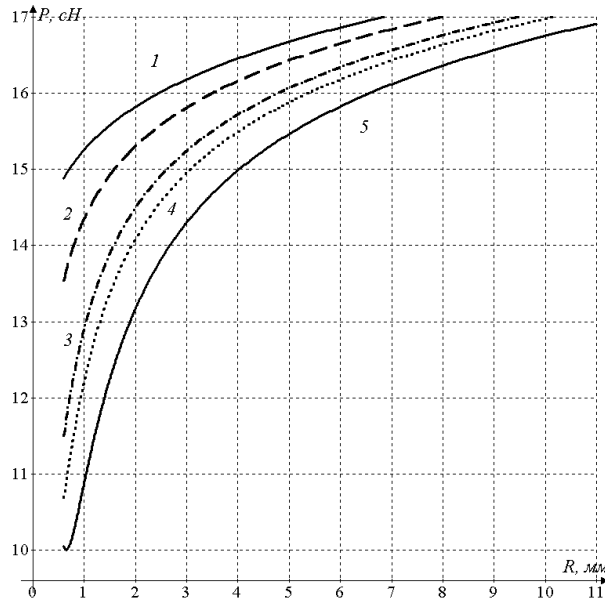


Рис.8.2.5. Залежність натягу жорсткої на згин капронової нитки 28 текс нитки після ниткопрямувача від коефіцієнту жорсткості на згин та радіусу кривизни напрямної поверхні: 1 - $B=0$ сНмм² (коефіцієнт крутіння $K_r=0$), 2 - $B=0,5$ сНмм², 3 - $B=3$ сНмм², 4 - $B=5$ сНмм², 5 - $B=11,2$ сНмм²(мононитка)

Відповідна сила натягу нитки досягається за рахунок сил тертя. Сили тертя виникають в точках контакту нитки з фігурними виступами шайб та при огинанні направляючого стрижня. Крім того, в точках контакту з фігурними виступами шайб відбувається деформація змінання об'єму нитки, що призводить до стрибкового зростання натягу на виході після нитконатягувача.

Натуральні комплексні нитки та пряжа, які складають левову частку асортименту ниток, які перероблюються на круглов'язальних машинах, мають багато дефектів. К ним можна віднести місця з неоднорідністю щільністю по масі, потовщення, вузли. В результаті взаємодії з робочими органами шайбового нитконатягувача в точках контакту нитки з фігурними виступами шайб та при огинанні направляючого стрижня виникають значні сили супротиву переміщенню нитки в зону в'язання. При збільшенні швидкості руху нитки дія цих вузлів та потовщень буде

носити ударний характер. Це може привести до обриву ниток, враховуючи наявність на них слабких місць та потоншень.

Одним з технічних рішень, яке дозволяє частково стабілізувати натяг нитки на в'язальній машині, є шайбовий нитконатягувач, який містить стрижень з надягнутими на нього двома тарілочками. Тиск між тарілочками створюється пружиною. Силу тиску можна змінювати за допомогою гайки. Для стабілізації натягу пристрій оснащено двоплечовим важелем. Один кінець важеля взаємодіє з пружиною, а з іншим взаємодіє рухома нитка. Вісь двоплечового важеля і стрижень тарілкового нитконатягувального пристрою перехрещуються під кутом 90° . При зміні натягу відбувається обертання двоплечового важеля, що призводить до зміни ступеня стискання пружини, а це у свою чергу, призводить до зміни тиску тарілочок на нитку в зоні їх контакту. При взаємодії шишок, потовщень та вузлів з двоплечовим важелем відбувається ударний вплив, що призводить до різкого збільшення натягу та обриву нитки. Це спричинено тим, що радіус кривизни направляючої поверхні двоплечового важеля спів розмірний з діаметром вузла, шишки або потовщення. Крім того, наявність двоплечового важеля не виключає ударну взаємодію вузла, шишки або локального потовщення з поверхнею тарілочок нитконатягувального пристрою. Ця обставина негативно впливає на нормальне проходження технологічного процесу [20, 31, 63].

Для виключення цього явища, враховуючи те, що недостатній рівень розвитку підготовчого виробництва не гарантує різкого підвищення якості сировини та готових комплексних ниток та пряжі, у роботі була запропонована конструкція шайбового нитконатягувача, яка захищена патентом України [5].

На рис.8.2.6 представлений шайбовий нитконатягувальний пристрій. На рис.8.2.6а показаний вид збоку, на рис.8.2.6б показаний вид зверху, а

на рис.8.2.6в показаний перетин А-А. Нитконатягувальний пристрій містить стрижень 1 з двома фігурними шайбами 2 і 3, що надягнуті на нього, пружину 4 і гайку 5, що забезпечує необхідний ступінь стиску пружини 4. На вертикальній осі 6 розташовується фігурний виделковий одноплечий важіль 7, що прикріплений до пружини стиску 8 і має два клиноподібних штовхачі 9. Обмежувач 10 переміщення верхньої шайби виконаний у вигляді гвинтової пари. Робоча поверхня важеля має ввігнуту форму, яка створена двома півколами, а в точці контакту з тарілочками з неробочої сторони має два клиноподібних штовхачі.

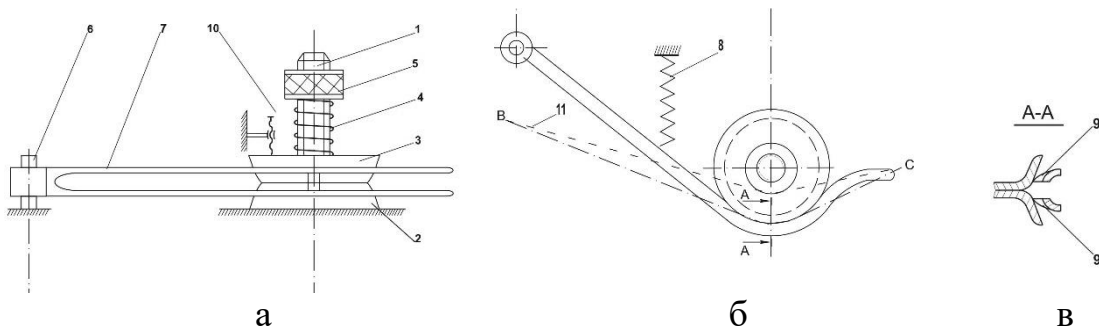


Рис.8.2.6. Нитконатягувальний пристрій (Патент України №30205)

Робоча поверхня важеля має ввігнуту форму, що утворена двома півколами, а в точці контакту з шайбами з неробочої сторони має два клиноподібні штовхачі, обмежувач. Форма фігурного виделкового одноплечого важеля утворюється з відрізка прямої та двох дуг кола, радіус якого дорівнює зовнішньому радіусу тарілочок.

Нитка 11, виходячи із точки, заправляється між зубцями фігурного виделкового одноплечого важеля 7, далі проходить під шайбами 2, 3, огинає стрижень 1 і надходить у точку С. При відсутності вузлів і потовщень нитка має форму, яка показана пунктирною лінією. При наявності вузла або шишки, останні, при переміщенні нитки, починають давити на поверхню одноплечого важеля 7. При цьому, відстань між зубцями вилки менше діаметру вузла. Момент, що виникає від сили

впливу вузла на робочу поверхню одноплечого важеля 7, викликає стискання пружини 8, при цьому два клиноподібних штовхачі розсовують шайби 2 і 3 і вивільняють нитку 11. Величина переміщення верхньої тарілки регулюється обмежувачем 10. При цьому нитка 11 має поточну форму, показану штрих-пунктирною лінією. Вузол або шишка рухаються по робочій поверхні одноплечого важеля 7 не потрапляючи під шайби. Після сходу вузла або нитки з робочої поверхні одноплечого важеля 7 під дією пружини стиску у вихідний стан.

Використання на круглов'язальних машинах запропонованого нитконатягувального пристрою дозволяє виключити стрибкоподібне збільшення натягу нитки, що виникає за рахунок ударного впливу при проходженні вузлів, шишок і локальних потовщень і виключити обрив ниток. Це дозволить значно підвищити продуктивність технологічного устаткування за рахунок зниження часу простоїв для ліквідації обриву й підвищити якість продукції, що випускається.

Розрахунок шайбового нитконатягувального пристрою круглов'язальних машин необхідно здійснювати з урахуванням змінання нитки в зоні контакту з напрямною, її жорсткість на згин. В якості бази необхідно скористуватися відомою методикою [6, 11, 15]. Виходячи з цього для розрахунку вихідного натягу будемо враховувати результати отримані в підрозділі 8.2.2. Тоді, з використанням формул (8.2.26)-(8.2.27), отримаємо

$$\begin{aligned}
 P = P_0 \{ & [1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\frac{a}{P_0} R^b \varphi} - 1)] + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right] - \\
 & - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\frac{a}{P_0} R^b \varphi} - 1) \right] + \\
 & + \frac{\mu_1 N}{1 + \sin(\beta - \frac{\varphi}{2})} \left(1 + \left[1 + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\frac{a}{P_0} R^b \varphi} - 1) \right] + \right. \\
 & + \left. \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta)]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R+r(1-\delta_0)]^2} \right] \left[1 + \right. \right. \\
 & \left. \left. + \frac{(R+r)}{[R+r(1-\delta_0)]} (e^{\frac{a}{P_0} R^b \varphi} - 1) \right] \right\}, \tag{8.2.48}
 \end{aligned}$$

$$\sin \beta = \frac{r_C}{R_C},$$

$$\varphi_{cm1} = \arccos \left[1 - \delta_0 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{cm2} = \arccos \left[1 - \delta \left(\frac{2r}{R} \right)^2 \right],$$

$$\varphi_{уж1} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P_0 (R+r)^2} \right],$$

$$\varphi_{уж2} = \arccos \left[1 - \frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2P (R+r)^2} \right].$$

де N – тиск пружини на рухому верхню шайбу;

R_C – радіус кола, на якому розташовані точки контакту рухомої шайби з ниткою;

r_C – радіус стрижня шайбового нитконатягувача;

μ_1 – коефіцієнт тертя між поверхнями нитки та рухомою шайбою.

Розрахунки вихідного натягу по базовій методиці відрізняються від даних отриманих при розрахунках з урахуванням змінання та жорсткості нитки на згин в середньому на 15-22%. Це підтвердили і результати експериментальних досліджень [22, 30]. Таке занижене значення натягу може призвести до суттєвих помилок при проектуванні системи ниткоподачі на круглов'язальних машинах.

Використовуючи викладену вище методику у роботі був розроблений нитконапрямник [26], який дозволяє відтворювати функцію подачі нитки у

робочу зону, що призводить до стабілізації натягу нитки за весь цикл формування елемента полотна та зменшує кількість обривів нитки.

Висновки по розділу

1. Проведений структурний аналіз системи ниткоподачі круглов'язальних машин, що дозволило встановити вирішальний вплив ниткоспрямовувачів та шайбових нитконатягувачів на величину натягу нитки перед робочою зоною.
2. На основі дослідження взаємодії нитки в динамічних умовах з поверхнею ниткоспрямовувача отримана система диференціальних рівнянь, яка пов'язує між собою основні силові, фізико-механічні, конструктивні параметри, які впливають на натяг.
3. На основі розв'язання системи диференціальних рівнянь, яка описує умови взаємодії жорсткої на згин, зминаємої нитки в зоні контакту з поверхнею ниткоспрямовувача, отримані залежності вихідного натягу та кута охоплення від фізико-механічних властивостей комплексних ниток і пряжі та конструктивних параметрів ниткоспрямовувача.
4. В процесі формалізації критерію оптимізації конструктивних параметрів ниткоспрямовувача отримана система трансцендентних рівнянь, рішення якої з використанням чисельних методів (методом дихотомії) та ЕОМ дозволило визначити геометричні розміри напрямних поверхонь в ниткоспрямовувачах, з урахуванням жорсткості на згин та зминання в зоні контакту, при яких вихідний натяг буде мінімальним.
5. На основі розв'язання системи диференціальних рівнянь, яка описує умови взаємодії комплексних ниток та пряжі малого крутіння в зоні контакту з поверхнею ниткоспрямовувача, отримані залежності вихідного натягу та кута охоплення від фізико-механічних властивостей та конструктивних параметрів ниткоспрямовувача.

6. Для комплексних ниток та пряжі малого крутіння отримані математичні залежності критерію оптимізації конструктивних параметрів ниткопрямовувача, розв'язання яких з використанням чисельних методів та ЕОМ дозволило визначити геометричні розміри напрямних поверхонь в ниткопрямовувачах, з урахуванням зминання в зоні контакту, при яких вихідний натяг буде мінімальним.
7. Отримана система диференційних рівнянь для жорсткої на згин нитки, інтегрування якої дозволило отримати залежності натягу та кута охоплення напрямної від конструктивних параметрів ниткопрямовувача, які були використані для оптимізації конструктивних параметрів системи ниткоподачі круглов'язальних машин.
8. Проведений аналіз роботи шайбових нитконатягувачів та нитконапрямників, яких дозволив встановити їх основні конструктивні недоліки, що дозволило розробити нові конструкції нитконатягувальних пристроїв та нитконапрямників, на які отримані патенти України №30205, №45940 та виконати основні розрахунки для визначення впливу їх конструктивних параметрів на вихідний натяг нитки.

8.3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ВПЛИВУ ЕЛЕМЕНТІВ СИСТЕМИ ПОДАЧІ НИТКИ НА ЇЇ НАТЯГ

У четвертому розділі наведено результати досліджень з визначення впливу конструктивних параметрів фідерних та кільцевих спрямовувачів на вихідний натяг нитки при її взаємодії з направляючою поверхнею фідерів та спрямовувачів. Дослідження процесу взаємодії ниток з направляючими поверхнями вимагає проведення цілої серії дослідів для підтвердження коректності зроблених допущень при виведенні теоретичних залежностей в 3 розділі і визначення меж використання

останніх[29-34].

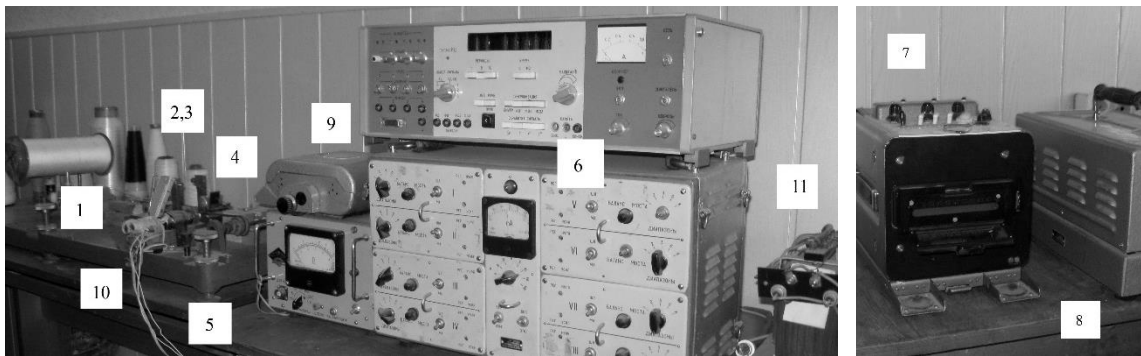
Активне планування експерименту, використання методів математичної статистики дозволяє отримати регресійні залежності, які описують процес з необхідною точністю, при цьому число чинників може досягати значної величини. Застосування ЕОМ дозволяє легко визначати величину коефіцієнтів в рівняннях регресії, їх значущість. Отримані залежності, після відповідних перетворень, можна використовувати для оптимізації процесу взаємодії нитки з структурними елементами системи подачі нитки.

Обґрунтування вибору вимірювально-реєструючої апаратури.

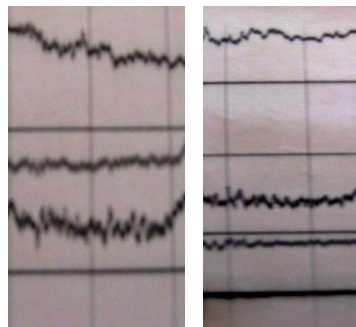
Розробка установок для проведення експерименту

Для реалізації плану експериментальних досліджень потрібно розробити спеціальні установки для визначення впливу конструктивних параметрів фідерних та кільцевих спрямовувачів на вихідний натяг. Велике значення має обґрунтування і вибір вимірювально-реєструючої апаратури[116, 129].

Загальний вигляд установки для визначення впливу конструктивних параметрів фідерних та кільцевих спрямовувачів на вихідний натяг наведено на рис.8.3.1а. На станині 5 жорстко укріплені перший 2 і другий 3 вимірювальні вузли. Між вимірювальними вузлами 2 і 3 встановлено пристрій 4 для вимірювання фрикційних властивостей ниток з урахуванням анізотропії. Нитка, сходячи з циліндричної шпулі 1, проходить через тарілчастий нитконатяжний прилад, який надає їй попередній натяг (натяг веденої гілки).



а



б

Рис.8.3.1. Загальний вигляд установки

Послідовно проходячи через перший вимірювальний вузол 2, шайбовий нитконатяжний прилад, пристрій 4 для вимірювання фрикційних властивостей з урахуванням анізотропії нитка надходить у пристрій для змотування 10[129].

Електричні сигнали з першого вимірювального вузла 2 і другого вимірювального вузла 3 надходять на 8-и каналний підсилювач 8АНЧ-7М (позиція 6). Далі через фільтр низьких частот 9 сигнали надходять у осцилограф 7 і далі на відповідні гальванометри осцилографа К12-22.

У таблиці 8.3.1 представлені характеристики гальванометрів осцилографа К12-22 (аналіз показав, що найбільш підходящими гальванометри для експерименту є гальванометри I і II типу).

Таблица 8.3.1

Характеристики гальванометрів осцилографа К12-22

Характеристика гальванометрів	Типи гальванометрів				
	I	II	III	IIIa	IV
Чутливість, мВ/мА, не менше	0,21	0,57	2,37	9,5	32,2
Електричний опір, Ом	16±3	36±7	8±1,5	17±4,5	22±3,5
Максимально допустиме значення постійного струму, мА	30	30	15	3,5	2
Діапазон реєстрації змінних процесів, Гц	Від 0 до 500	Від 0 до 300	Від 0 до 150	Від 0 до 100	Від 0 до 30

В експериментальну установку також входили блоки живлення 8, 11 відповідно для підсилювача 8АНЧ-7М і осцилографа К12-22. Зразки осцилограм наведені на рис.8.3.1б.

На рис.8.3.2 представлена схема вимірювального вузла, який включає в себе два ролика 1 і 2, які встановлені в підшипниках на нерухомих осях. Третій ролик 3 встановлений на консольно закріплену балочку таким чином, що внутрішнє кільце підшипника закріплено на ній, а з зовнішнім кільцем підшипника жорстко закріплений ролик взаємодіє з ниткою. Поперечні і поздовжні розміри балочки обрані такими, що частота власних коливань балочки дорівнює $f = 1400$ Гц. Ця частота у багато разів перевищує частоту найбільш високої складової натягу. На балочку наклеєний тензодатчик з базою в 5 мм, які в парі з іншим тензодатчиком (наклеєним на пластину з того ж матеріалу) утворюють одну половину вимірювального моста. Друга половина розташовується на відповідному каналі в підсилювачі 8АНЧ-7М.

Установка включає третій вимірювальний вузол 3. Це зроблено для того, щоб можна було одночасно з експериментальним дослідженням умов взаємодії нитки з напрямною 6, встановленою нерухомо в корпусі 5, проводити експеримент за визначенням умов роботи різних нитконатягувачів, які встановлювалися між 2 і 3 вимірювальними вузлами.

Крім того, для варіювання швидкості руху нитки був встановлений ступінчастий привод 8, який забезпечує необхідну швидкість обертання приймальному барабану 9.

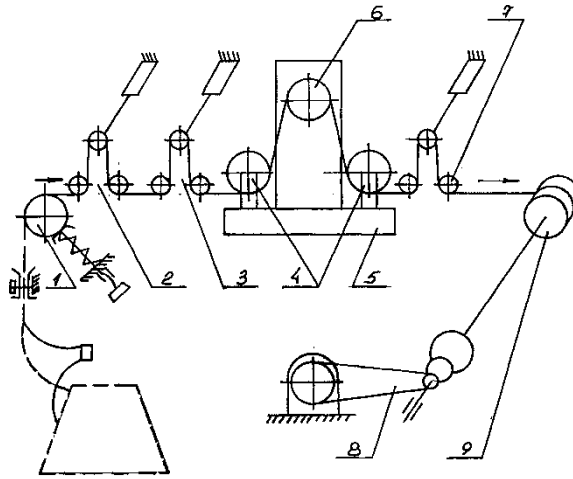


Рис.8.3.2. Електротензометрична установка

У приладі передбачалася подача нитки безпосередньо з конічної бобіни, для чого в його конструкцію були введені ниткоспрямовувачі і шайбовий нитконатягувач. Натяг веденої гілки нитки перед напрямною 6 реєструвався вимірювальним вузлом 3. Натяг ведучої гілки нитки - вимірювальним вузлом 7. У приладі була передбачена можливість заміни направляючої 6 з метою визначення впливу радіусу її кривизни на умови взаємодії з ниткою.

Для здійснення експериментальних досліджень з визначення впливу конструктивних параметрів фідерних та кільцевих спрямовувачів на вихідний натяг розроблено прилади для визначення натягу ведучої гілки нитки; для визначення фрикційних властивостей ниток в умовах, наближених до реальних, таких, що виникають на технологічному устаткуванні. Три з цих установок захищені патентами України (патент України №31888, патент України №45939, патент України №80199).

Задачею запропонованої нижче моделі є розширення технологічних можливостей за рахунок збільшення діапазону вимірювання фрикційних властивостей ниток при великих швидкостях руху[124]. На рис.8.3.3 зображена схема приладу.

Прилад для визначення коефіцієнта тертя ниток містить шків 1, встановлений з можливістю горизонтального переміщення відносно осі обертання ведучого вала 2, гнучку стрічку із жорсткістю в повздовжньому й поперечному напрямку, з'єднаний з вантажем з однієї сторони й робочим рухомим органом динамометра з іншої сторони, на нижній поверхні якого з боку шківа закріплений зразок нитки, що досліджується, й розташовується між трьома парами роликів, укріпленими на двох кронштейнах, які з'єднані з робочими рухомими органами динамометрів, який має два конічні зубчасті колеса 3, що жорстко закріплені на перетнутих валах, що розташовуються в опорах, один із яких є ведучим 2, а другий веденим. На веденому валу розташованій кривошип 5, кінематично зв'язаний з шатуном 5, а шатун зв'язаний з повзуном 7, розташованим у горизонтальних направляючих.

Вилкоподібний кронштейн 8, на кінцях якого розміщені двосторонні опорні підшипники ковзання 9, які зв'язані із шипами 10, що закріплені з торців шківа 1, жорстко закріплений на повзуні 7, шипи 10 розташовуються з боків шківа 1.

Використання запропонованого приладу дозволяє значно розширити діапазон вимірюваних величин шляхом визначення коефіцієнта тертя ниток при більших швидкостях руху.

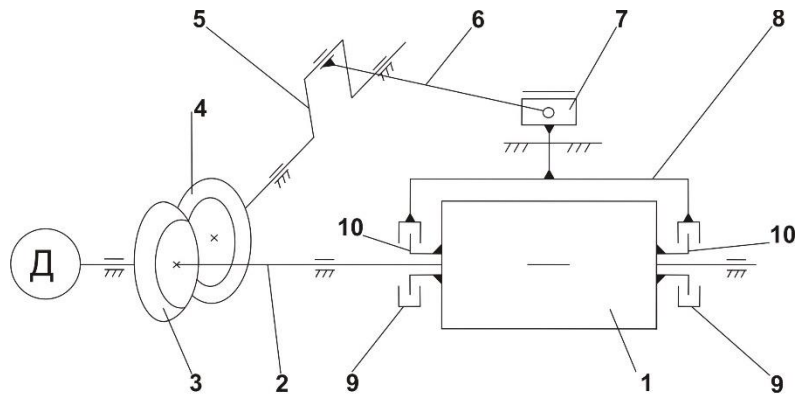


Рис.8.3.3. Схема вимірювального вузла (патент України №31888)

Для розширення технологічних можливостей по зміні кута охоплення направляючих поверхонь фідерів та кільцевих спрямовувачів нитки був розроблений наступний прилад[125]. На Рис.8.3.4 зображена схема пристрою (центральна каретка вимірювального вузла показана в загальному вигляді).

Пристрій для визначення фрикційних властивостей нитки містить в корпусі 1 приладу бобінотримач 2, нитконапрямник 3, нитконатягувач 4, вимірювальний вузол в вигляді каретки 5. Каретка 5 встановлена в направляючій 7 корпусу 1 та містить індикаторну стрілку 8 та фігурний кронштейн 9. Нитка отримує рух від пристрою 11. Індикаторна лінійка 10 промаркована в одиницях, які характеризують фрикційні властивості нитки. На корпусі приладу змонтовані рухомі рамки 12-14 для переміщення нитки. Направляюча 7 встановлена з можливістю повздовжнього переміщення за допомогою механізму 15 для переміщення відносно нерухомо змонтованих підшипником ковзання 16. Нижче направляючої 7 на корпусі 1 змонтовані два шківів 17 та 18 з нескінченною гнучкою металевою стрічкою 19, на якій закріплена металева пластинка 20 з контактом 21 та пружиною 22. Два рухомих шківів 23 розташовуються на

осях з пристроями фіксації та вказівниками 25, які розміщені в наскрізному горизонтальному пазу 26 каретки 5.

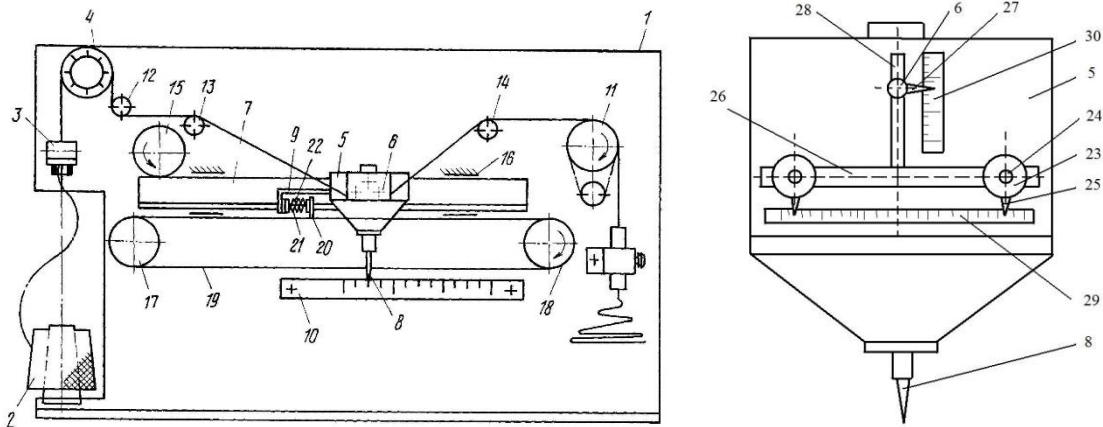


Рис.8.3.8.3. Схема вимірювального пристрою (патент України №45939)

Нитконапрямок 6 з пристроєм фіксації та вказівником 27 розташовані в наскрізному вертикальному пазу 28 каретки 5. Горизонтальна 29 і вертикальна 30 шкали мають метричну шкалу.

Для визначення фрикційних властивостей ниток в подовжньому і поперечному напрямку з урахуванням анізотропії тертя розроблено спеціальний пристрій[28]. Винахід відноситься до галузі дослідження матеріалів механічними способами, зокрема для визначення фрикційних властивостей гнучких ниток. Завданням запропонованого винаходу є розширення технологічних можливостей за рахунок збільшення діапазону вимірювання фрикційних властивостей ниток при великих швидкостях руху. На рис.8.3.5 зображена схема приладу.

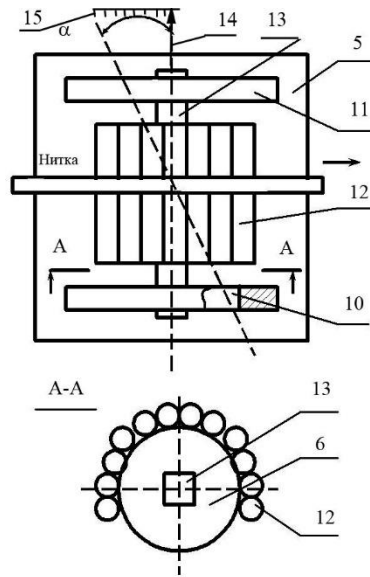


Рис.8.3.5. Схема вузла для визначення фрикційних властивостей нитки з урахуванням анізотропії (патент України №80199)

Пристрій для визначення фрикційних властивостей нитки містить котушкотримач, підпружинений ролик 1, три вимірювальних вузла 2-3 з тензодатчиками, два рухливих ролика 4, закріплених на корпусі 5 і нерухому циліндричну направляючу 6, яка закріплена на кронштейнах 11 жорстко закріплених на корпусі 5, приймальний барабан 9 для нитки і ступінчастий привід 8, горизонтальну циліндричну направляючу на поверхні якої, за твірними, щільно розташовані нитки, кінці циліндричної направляючої квадратної форми в перетині з пристроєм фіксації розташовані на вертикальних кронштейнах в двох горизонтальних пазах, висота яких дорівнює довжині сторони квадрата кінця циліндричної направляючої, показчик і кутомір.

Розробка планів проведення експерименту

Визначення основних чинників конструктивних параметрів фідерних та кільцевих спрямовувачів нитки, які обумовлюють зміну натягу нитки,

що рухається по направляючій поверхні фідерних та кільцевих спрямовувачів, має значення при виборі стратегії проведення експерименту.

Наукова інформація, яка накопичилася до теперішнього часу, з даної проблеми дозволяє побудувати середню апріорну діаграму рангів. По одній осі відкладаються чинники – конструктивні і технологічні параметри фідерних та кільцевих спрямовувачів, а по іншій суми рангів. До даних чинників були віднесені: внутрішній радіус фідера, радіус зовнішньої кривизни фідера та спрямовувача, натяг веденої гілки, кут охоплення ниткою фідера, швидкість руху нитки, температура навколишнього середовища, вологість повітря, коефіцієнт тертя між ниткою і напрямними поверхнями фідерних та кільцевих спрямовувачів, характер обробки напрямної поверхні, матеріал і структура нитки.

При проведенні експерименту обрано: капронову комплексну нитку 28 текс(КН); віскозну комплексну нитку 16,7(ВН); вовняну пряжу 29,9 текс(ВП); бавовняну пряжу 27,6 текс(БП); віскозну штапельну пряжу 28,8 текс(ВШП).

Аналіз середньої апріорної діаграми рангів дозволив встановити, що для внутрішнього радіусу фідера, радіусу зовнішньої кривизни фідера та спрямовувача, натягу веденої гілки, кута охоплення ниткою фідера розподіл відповідних сум рангів рівномірний[29].

Таким чином, при проведенні експериментальних досліджень, поставлено завдання з визначення впливу внутрішнього радіусу фідера та кільцевих спрямовувачів r , радіус кривизни фідера R , натягу веденої гілки P_0 , кута охоплення φ на натяг ведучої гілки P нитки.

Після визначення чинників, що впливають на збільшення натягу ведучої гілки нитки, обґрунтування і вибору сировини для випробувань, необхідно скласти план проведення експерименту. Це дозволить виробити

певну стратегію в пошуку оптимального поєднання чинників, що впливають на даний процес.

Для кожної з ниток даним планом (таблиця 8.3.2 і таблиця 8.3.3) передбачається 15 дослідів. По кожному з дослідів передбачалося 5 повторних.

Одна серія експериментів проводилася для дослідження впливу внутрішнього радіусу фідера r , радіусу кривизни фідера R , кута охоплення φ на натяг ведучої гілки P нитки. Друга серія експериментів проводилася для дослідження впливу внутрішнього радіусу кільцевих спрямовувачів r , натягу веденої гілки P_0 , кута охоплення φ на натяг ведучої гілки P нитки.

Інтервали зміни величини натягу веденої гілки нитки (від 5 до 15 сН) були вибрані виходячи з умов переробки ниток на в'язальних машинах.

Кут охоплення направляючої поверхні, для кожного конкретного випадку, є величиною змінною. Зважаючи на специфіку переробки ниток на в'язальному устаткуванні, де мають місце великі перегини ниток при взаємодії з робочими органами, був вибраний інтервал зміни від 30° до 110° (для кожної серії інтервал зміни кута обирався в цьому діапазоні)[2].

При плануванні експерименту, при побудові моделей другого порядку використано ортогональне планування другого порядку. Хоча існують інші оптимальні плани, ортогональне планування використовувалося через свою достатню простоту і можливість застосування вже розроблених пакетів програм для ЕОМ з обробки отриманих експериментальних даних[1].

Таблиця 8.3.2

План проведення експерименту ($P_0 = 5 \text{ сН}$) (внутрішня поверхня фідера)

№ варіанта	Досліджувані чинники						Характерні зони матриці
	x1	r	x2	R	x3	φ	
1- 1	-1	r-h1	+1	R+ h2	+1	$\varphi+ h3$	Точки ядра
1- 2	+1	r+h1	-1	R- h2	+1	$\varphi+ h3$	
1- 3	-1	r-h1	+1	R+h2	-1	$\varphi- h3$	
1- 4	+1	r+h1	+1	R+ h2	+1	$\varphi+ h3$	
1- 5	+1	r+h1	+1	R+ h2	-1	$\varphi- h3$	
1- 6	+1	r+h1	-1	R- h2	-1	$\varphi- h3$	
1- 7	-1	r-h1	-1	R- h2	-1	$\varphi- h3$	
1- 8	-1	r-h1	-1	R- h2	+1	$\varphi+ h3$	
1- 9	-1,215	r-*	0	R	0	φ	
1- 10	+1,215	r+*	0	R	0	φ	
1- 11	0	r	-1,215	R-*	0	φ	
1- 12	0	r	+1,215	R+*	0	φ	
1- 13	0	r	0	R	-1,215	$\varphi-^*$	
1- 14	0	r	0	R	+1,215	$\varphi+^*$	
1- 15	0	r	0	R	0	φ	Нульова точка

Дослідження впливу конструктивних параметрів фідерних спрямовувачів нитки на її натяг

Реалізуючи план експериментальних досліджень (табл.8.3.2) за визначенням впливу конструктивних параметрів фідерних спрямовувачів круглов'язальних машин отримано ряд даних, представлених в таблиці 8.3.8.3.

Вище ми зупинялися на тих міркуваннях, які привели до вибору ортогонального плану. Останні вимоги застосування того або іншого

плану, що пред'являються до оцінки ефективності, привели до того, що основними критеріями необхідно вважати: максимальну величину дисперсії передбачених значень критерію оптимізації в області експерименту; об'єм еліпсоїда розсіяння оцінок параметрів та ін.[4].

Таблиця 8.3.3

План проведення експерименту ($R = 3$ мм) (внутрішня поверхня керамічного спрямовувача)

№ варіанта	Досліджувані чинники						Характерні зони матриці
	x1	r	x2	Po	x3	φ	
2- 1	-1	r-h1	+1	Po+ h2	+1	$\varphi+$ h3	Точки ядра
2- 2	+1	r+h1	-1	Po- h2	+1	$\varphi+$ h3	
2- 3	-1	r-h1	+1	Po+ h2	-1	$\varphi-$ h3	
2- 4	+1	r+h1	+1	Po+ h2	+1	$\varphi+$ h3	
2- 5	+1	r+h1	+1	Po+ h2	-1	$\varphi-$ h3	
2- 6	+1	r+h1	-1	Po- h2	-1	$\varphi-$ h3	
2- 7	-1	r-h1	-1	Po- h2	-1	$\varphi-$ h3	
2- 8	-1	r-h1	-1	Po- h2	+1	$\varphi+$ h3	
2- 9	-1,215	r-*	0	Po	0	φ	
2- 10	+1,215	r+*	0	Po	0	φ	
2- 11	0	r	-1,215	Po-*	0	φ	
2- 12	0	r	+1,215	Po+*	0	φ	
2- 13	0	r	0	Po	-1,215	$\varphi-$ *	
2- 14	0	r	0	Po	+1,215	$\varphi+$ *	
2- 15	0	r	0	Po	0	φ	Нульова точка

При проведенні експерименту іменовані величини необхідно замінити на кодовані. В нашому випадку внутрішній радіус кільцевих спрямовувачів r позначимо через $x1$. В нульовій точці внутрішній радіус

кільцевих спрямовувачів дорівнював 3 мм. Довжина плеча складала 1 мм. Радіус кривизни фідера R позначимо через x_2 . В нульовій точці радіус кривизни фідера дорівнює 10 мм. Довжина плеча складала 5 мм. Кут охоплення ниткою напрямної φ позначимо через x_3 . В нульовій точці кут охоплення ниткою напрямної дорівнює 90^0 . Довжина плеча складала 10^0 .

Таблиця 8.3.4

Результати проведення експерименту

$P_0 = 5$ сН (внутрішня поверхня фідера)								
№ варіанта	Досліджувані чинники			Натяг нитки, сН				
	x_1	x_2	x_3	КН	ВН	ВП	БП	ВШП
1- 1	-1	+1	+1	13,20	12,90	9,40	12,40	10,40
1- 2	+1	-1	+1	8,67	11,50	8,41	10,80	8,41
1- 3	-1	+1	-1	10,40	12,20	8,90	11,82	9,80
1- 4	+1	+1	+1	9,85	12,50	9,10	11,75	9,35
1- 5	+1	+1	-1	7,55	8,90	8,51	10,50	8,67
1- 6	+1	-1	-1	8,01	8,18	7,58	8,28	7,85
1- 7	-1	-1	-1	11,42	10,10	8,32	9,85	8,91
1- 8	-1	-1	+1	12,01	11,90	8,98	11,93	9,54
1- 9	-1,215	0	0	13,50	11,20	9,22	11,30	9,98
1- 10	+1,215	0	0	9,10	9,80	8,21	9,28	8,88
1- 11	0	-1,215	0	9,76	10,10	8,11	10,01	8,76
1- 12	0	+1,215	0	10,65	10,90	8,79	11,15	9,81
1- 13	0	0	-1,215	10,55	9,95	8,24	9,68	9,05
1- 14	0	0	+1,215	11,40	11,01	8,81	10,85	9,53
1- 15	0	0	0	10,90	10,54	8,49	10,34	9,12

Тоді система рівнянь для переходу від іменованих до кодованих значень чинників має вигляд

$$x_1 = r - 3, \quad x_2 = \frac{R - 10}{5}, \quad x_3 = \frac{\varphi - 90}{10}. \quad (8.3.1)$$

На рис.8.3.6 для визначення впливу конструктивних параметрів фідерних спрямовувачів на вихідний натяг представлена поверхня відгуку для бавовняної пряжі 27,6 текс. Рівняння регресії для бавовняної пряжі має вигляд

$$P_{BP} = 10,14 - 0,64x_1 + 0,63x_2 + 0,70x_3 + 0,19x_1^2 + 0,09x_1x_2 + 0,14x_1x_3 + 0,36x_2^2 - 0,35x_2x_3 + 0,17x_3^2. \quad (8.3.2)$$

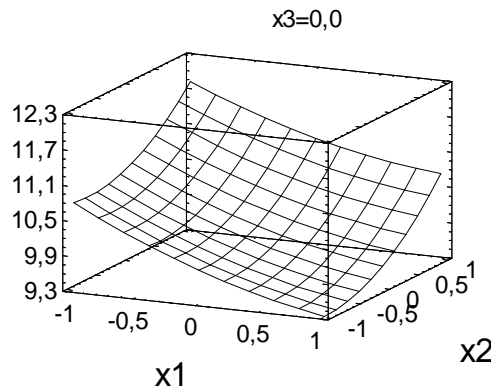


Рис.8.3.6. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.2) отримаємо відповідну діаграму, яка представлена на рис.8.3.7. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_3 , x_1x_2 , x_1^2 , x_2^2 , x_3^2 .

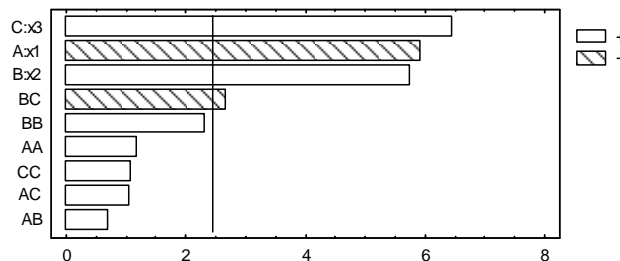


Рис.8.3.7. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.2), а

також з урахуванням системи (8.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P_{БП} = 10,75R + 0,14\varphi - 0,64r - 0,007R\varphi - 1,73. \quad (8.3.3)$$

На рис.8.3.8 для визначення впливу конструктивних параметрів фідерних спрямовувачів на вихідний натяг представлена поверхня відгуку для віскозної штапельної пряжі 28,8 текс. Рівняння регресії для віскозної штапельної пряжі має вигляд

$$P_{ВШП} = 9,28 - 0,51x_1 + 0,43x_2 + 0,27x_3 + 0,019x_1^2 + 0,0013x_1x_2 + 0,0013x_1x_3 - 0,068x_2^2 + 0,011x_2x_3 - 0,065x_3^2. \quad (8.3.4)$$

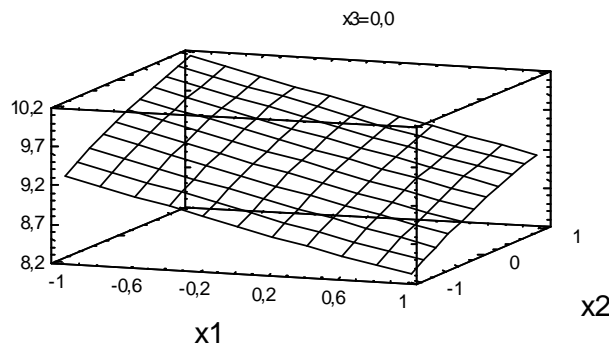


Рис.8.3.8. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.4) отримаємо відповідну діаграму, яка представлена на рис.8.3.9. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_2 , x_2x_3 , x_1x_3 , x_1^2 , x_2^2 , x_3^2 .

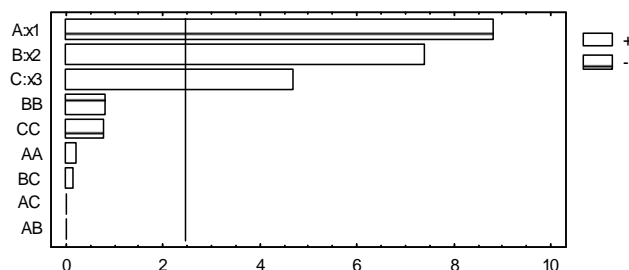


Рис.8.3.9. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.4), а

також з урахуванням системи (8.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P_{ВШП} = 7,497 + 0,086R + 0,027\varphi - 0,51r. \quad (8.3.5)$$

На рис.8.3.10 для визначення впливу конструктивних параметрів фідерних спрямовувачів на вихідний натяг представлена поверхня відгуку для віскозній комплексній нитки 16,7текс. Рівняння регресії для віскозної комплексної нитки має вигляд

$$P_{ВН} = 10,31 - 0,70x_1 + 0,51x_2 + 0,94x_3 + 0,22x_1^2 - 0,16x_1x_2 + 0,57x_1x_3 + 0,22x_2^2 - 0,09x_2x_3 + 0,21x_3^2. \quad (8.3.6)$$

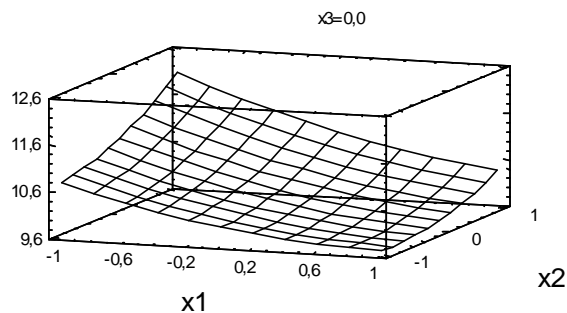


Рис.8.3.10. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.6) отримаємо відповідну діаграму, яка представлена на рис.8.3.11. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_1x_2, x_2x_3, x_1^2, x_2^2, x_3^2$.

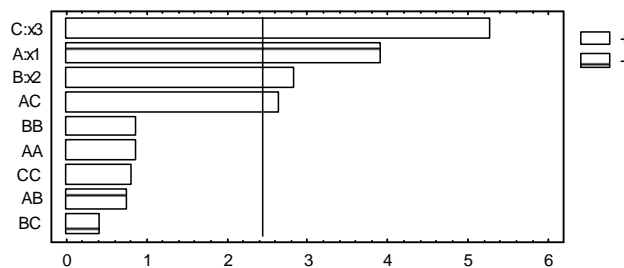


Рис.8.3.11. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.6), а також з урахуванням системи (8.3.1), перейдемо до іменованих значень,

остаточно отримаємо

$$P_{BH} = 18,32 + 0,102R - 0,077\varphi - 5,83r + 0,057\varphi r. \quad (8.3.7)$$

На рис.8.3.12 для визначення впливу конструктивних параметрів фідерних спрямовувачів на вихідний натяг представлена поверхня відгуку для вовняної пряжі 29,9 текс. Рівняння регресії для вовняної пряжі має вигляд

$$P_{BH} = 8,48 - 0,29x_1 + 0,31x_2 + 0,29x_3 + 0,15x_1^2 + 0,08x_1x_2 + 0,03x_1x_3 - 0,01x_2^2 - 0,05x_2x_3 + 0,03x_3^2. \quad (8.3.8)$$

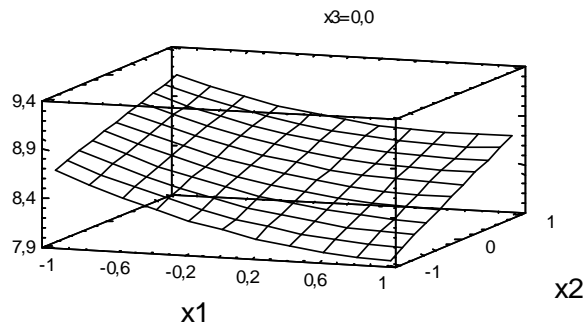


Рис.8.3.12. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.8) отримаємо відповідну діаграму, яка представлена на рис.8.3.13. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_1x_2, x_2x_3, x_1x_3, x_2^2, x_3^2$.

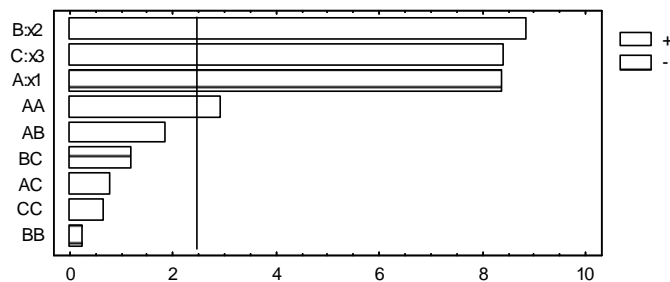


Рис.8.3.13. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.8), а також з урахуванням системи (8.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P_{ВП} = 7,47 - 1,19r + 0,06R + 0,03\phi + 0,15r^2. \quad (8.3.9)$$

На рис.8.3.14 для визначення впливу конструктивних параметрів фідерних спрямовувачів на вихідний натяг представлена поверхня відгуку для капронова комплексна нитка 28 текс. Рівняння регресії для капронової комплексної нитки має вигляд

$$P_{КН} = 11,12 - 1,59x_1 + 0,24x_2 + 0,59x_3 + 0,009x_1^2 + 0,15x_1x_2 - 0,14x_1x_3 - 0,65x_2^2 + 0,39x_2x_3 - 0,19x_3^2. \quad (8.3.10)$$

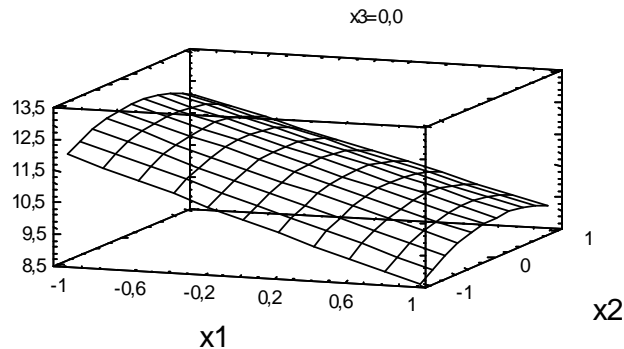


Рис.8.3.18.3. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.10) отримаємо відповідну діаграму, яка представлена на рис.8.3.15. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_2, x_1x_2, x_1x_3, x_1^2, x_3^2$.

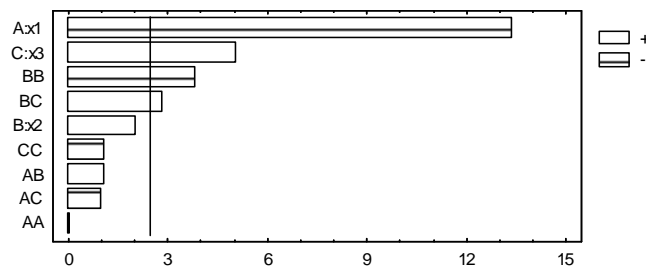


Рис.8.3.15. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.10), а також з урахуванням системи (8.3.1), перейдемо до іменованих значень, остаточно отримаємо

$$P_{KH} = 15,0 - 0,019\varphi - 0,18R - 1,59r + 0,008R\varphi - 0,03R^2. \quad (8.3.11)$$

Дослідження впливу конструктивних параметрів кільцевих спрямовувачів нитки на її натяг

Реалізуючи план експериментальних досліджень (табл.8.3.3) за визначенням впливу конструктивних параметрів кільцевих спрямовувачів круглов'язальних машин отримано ряд даних, представлених в таблиці 8.3.5.

При проведенні експерименту іменовані величини необхідно замінити на кодовані. В нашому випадку внутрішній радіус кільцевих спрямовувачів r позначимо через $x1$. В нульовій точці внутрішній радіус кільцевих спрямовувачів дорівнював 3 мм. Довжина плеча складала 1 мм. Натяг веденої гілки нитки P_0 позначимо через $x2$. В нульовій точці натяг веденої гілки нитки дорівнює 10 сН. Довжина плеча складала 5 сН. Кут охоплення ниткою напрямної φ позначимо через $x3$. В нульовій точці кут охоплення ниткою напрямної дорівнює 60^0 . Довжина плеча складала 30^0 . Тоді система рівнянь для переходу від іменованих до кодованих значень чинників має вигляд

$$x1 = r - 3, \quad x2 = \frac{P_0 - 10}{5}, \quad x3 = \frac{\varphi - 60}{30}. \quad (8.3.12)$$

Таблиця 8.3.5

Результати проведення експерименту

R = 3 мм (внутрішня поверхня керамічного спрямовувача)								
№ варіанта	Досліджувані чинники			Натяг нитки, сН				
	x1	x2	x3	КН	ВН	ВП	БП	ВШП
2- 1	-1	+1	+1	30,50	28,29	26,20	30,70	27,90
2- 2	+1	-1	+1	19,25	18,41	16,53	19,04	18,01
2- 3	-1	+1	-1	28,16	25,72	23,75	27,95	25,14

2- 4	+1	+1	+1	25,72	22,17	21,08	25,15	21,46
2- 5	+1	+1	-1	18,00	18,44	17,14	17,89	17,89
2- 6	+1	-1	-1	8,40	7,95	7,46	8,10	7,69
2- 7	-1	-1	-1	20,17	19,58	18,67	19,93	19,02
2- 8	-1	-1	+1	26,38	23,65	21,98	25,89	22,99
2- 9	-1,215	0	0	22,45	22,74	21,05	22,03	22,35
2- 10	+1,215	0	0	15,62	11,35	10,04	15,14	10,97
2- 11	0	-1,215	0	14,38	11,84	11,10	13,97	11,19
2- 12	0	+1,215	0	19,16	19,17	17,53	18,64	18,73
2- 13	0	0	-1,215	15,42	12,89	10,99	14,87	12,38
2- 14	0	0	+1,215	19,11	18,77	17,86	18,85	18,25
2- 15	0	0	0	17,28	16,41	15,37	17,15	16,06

На рис.8.3.16 для визначення впливу конструктивних параметрів кільцевих спрямовувачів на вихідний натяг представлена поверхня відгуку для бавовняної пряжі 27,6 текс. Рівняння регресії для бавовняної пряжі має вигляд

$$R_{БП} = 15,33 - 3,81x_1 + 3,07x_2 + 2,83x_3 + 2,79x_1^2 + 0,38x_1x_2 + 1,19x_1x_3 + 1,41x_2^2 - 0,86x_2x_3 + 1,75x_3^2. \quad (8.3.13)$$

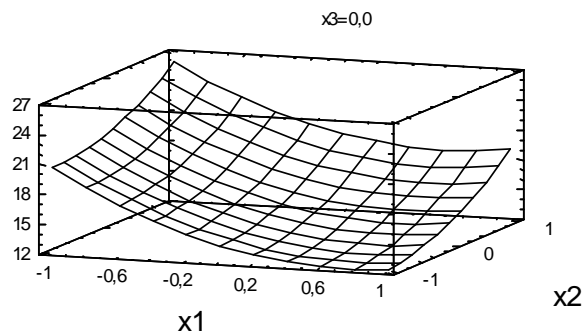


Рис.8.3.16. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії

(8.3.13) отримаємо відповідну діаграму, яка представлена на рис.8.3.17. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_1x_2, x_1x_3, x_2x_3, x_2^2, x_3^2$.

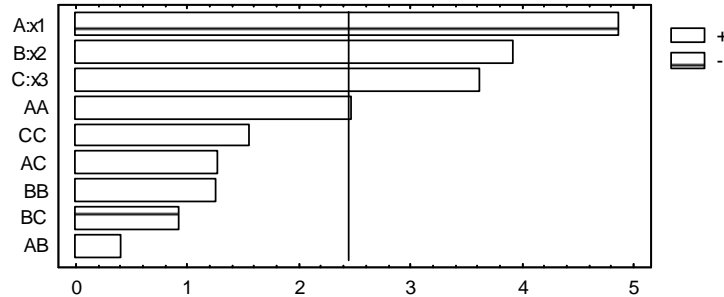


Рис.8.3.17. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.13), а також з урахуванням системи (8.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P_{БП} = 40,07 + 0,09\varphi + 0,61P_0 - 20,55r + 2,79r^2. \quad (8.3.14)$$

На рис.8.3.18 для визначення впливу конструктивних параметрів кільцевих спрямовувачів на вихідний натяг представлена поверхня відгуку для віскозної штапельної пряжі 28,8 текс. Рівняння регресії для віскозної штапельної пряжі має вигляд

$$P_{ВШП} = 14,15 - 3,95x_1 + 3,04x_2 + 2,49x_3 + 2,37x_1^2 + 0,33x_1x_2 + 0,89x_1x_3 + 1,34x_2^2 - 0,99x_2x_3 + 1,56x_3^2. \quad (8.3.15)$$

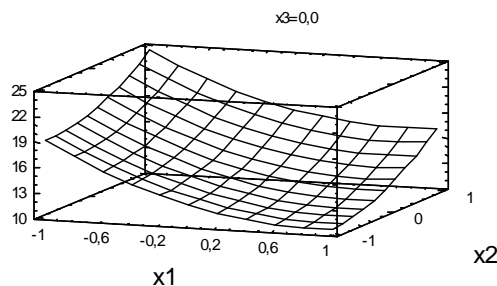


Рис.8.3.18. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.15) отримаємо відповідну діаграму, яка представлена на рис.8.3.19.

Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: x_1x_2 , x_2x_3 , x_1x_3 , x_1^2 , x_2^2 , x_3^2 .

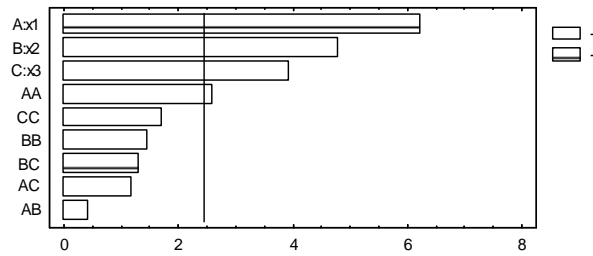


Рис.8.3.19. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.15), а також з урахуванням системи (8.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P_{ВШП} = 36,27 + 0,08\varphi + 0,61P_0 - 18,17r + 2,37r^2. \quad (8.3.16)$$

На рис.8.3.20 для визначення впливу конструктивних параметрів кільцевих спрямовувачів на вихідний натяг представлена поверхня відгуку для віскозній комплексній нитки 16,7текс. Рівняння регресії для віскозної комплексної нитки має вигляд

$$P_{ВН} = 14,53 - 3,97x_1 + 3,05x_2 + 2,51x_3 + 2,36x_1^2 + 0,43x_1x_2 + 0,94x_1x_3 + 1,43x_2^2 - 1,03x_2x_3 + 1,63x_3^2. \quad (8.3.17)$$

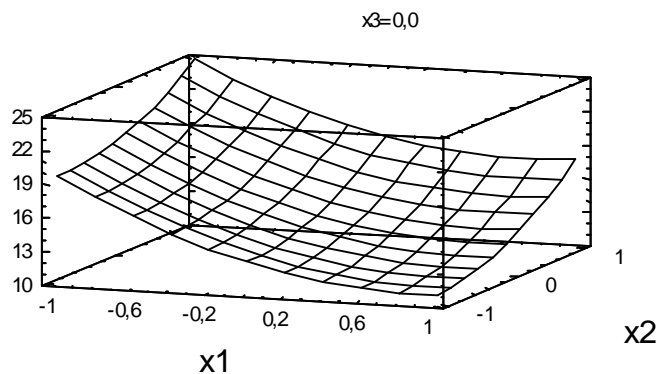


Рис.8.3.20. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.17) отримаємо відповідну діаграму, яка представлена на рис.8.3.21.

Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_1x_2, x_2x_3, x_1^2, x_2^2, x_3^2$.

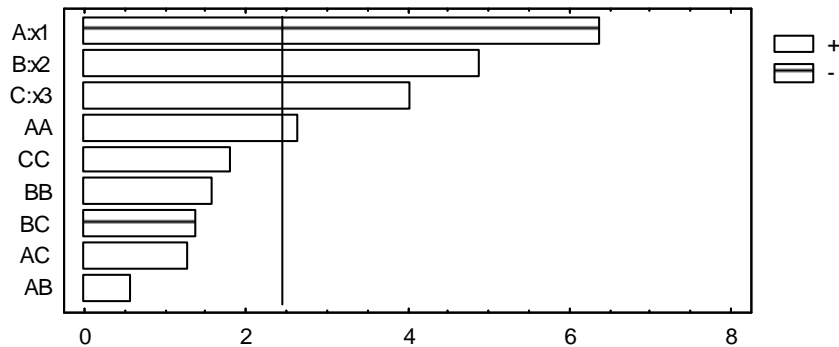


Рис.8.3.21. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.17), а також з урахуванням системи (8.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P_{BH} = 36,58 + 0,08\varphi + 0,61P_0 - 18,13r + 2,36r^2. \quad (8.3.18)$$

На рис.8.3.22 для визначення впливу конструктивних параметрів кільцевих спрямовувачів на вихідний натяг представлена поверхня відгуку для вовняної пряжі 29,9 текс. Рівняння регресії для вовняної пряжі має вигляд

$$P_{ВП} = 13,31 - 3,94x_1 + 2,63x_2 + 2,62x_3 + 2,28x_1^2 + 0,87x_1x_2 + 0,65x_1x_3 + 1,53x_2^2 - 1,00x_2x_3 + 1,59x_3^2. \quad (8.3.19)$$

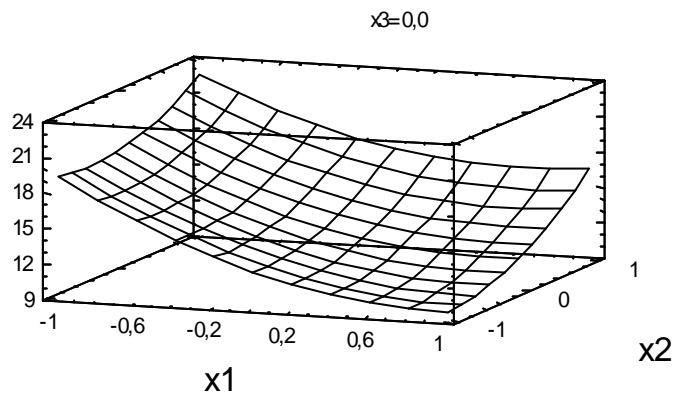


Рис.8.3.22. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.19) отримаємо відповідну діаграму, яка представлена на рис.8.3.23. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_1x_2, x_2x_3, x_1x_3, x_2^2, x_3^2$.

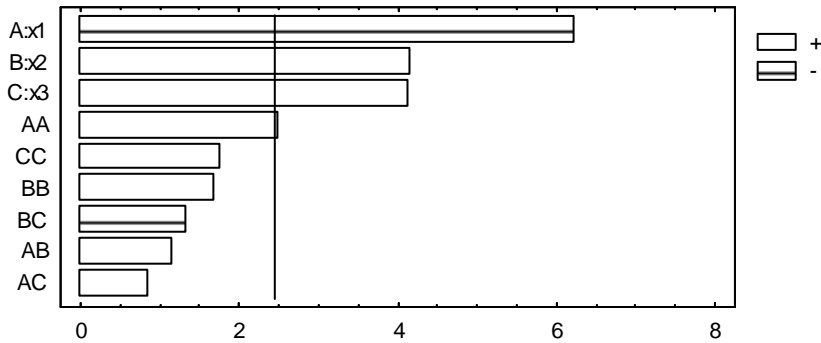


Рис.8.3.23. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.19), а також з урахуванням системи (8.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P_{ВП} = 35,15 + 0,09\varphi + 0,53P_0 - 17,62r + 2,28r^2. \quad (8.3.20)$$

На рис.8.3.24 для визначення впливу конструктивних параметрів кільцевих спрямовувачів на вихідний натяг представлена поверхня відгуку для капронова комплексна нитка 28 текс. Рівняння регресії для капронової комплексної нитки має вигляд

$$P_{КН} = 15,64 - 3,77x_1 + 3,03x_2 + 2,82x_3 + 2,79x_1^2 + 0,49x_1x_2 + 1,25x_1x_3 + 1,42x_2^2 - 0,88x_2x_3 + 1,72x_3^2. \quad (8.3.21)$$

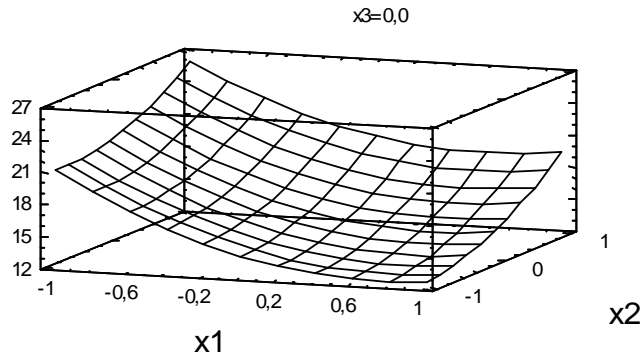


Рис.8.3.28.3. Поверхня відгуку

В результаті оцінки значущості коефіцієнтів в рівнянні регресії (8.3.21) отримаємо відповідну діаграму, яка представлена на рис.8.3.25. Аналіз даної діаграми показує, що не значущими коефіцієнтами будуть коефіцієнти при наступних комбінаціях: $x_2, x_1x_2, x_1x_3, x_1^2, x_3^2$.

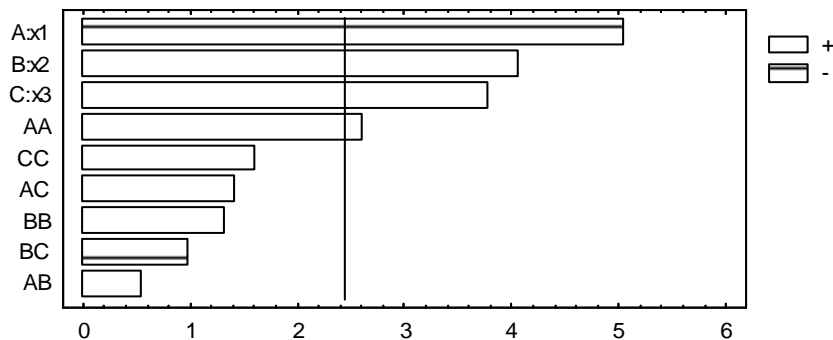


Рис.8.3.25. Діаграма значущості коефіцієнтів в рівнянні регресії

З урахуванням значущості коефіцієнтів в рівнянні регресії (8.3.21), а також з урахуванням системи (8.3.12), перейдемо до іменованих значень, остаточно отримаємо

$$P_{KH} = 40,34 + 0,09\varphi + 0,61P_0 - 20,51r + 2,79r^2. \quad (8.3.22)$$

Висновки по розділу

1. Для дослідження взаємодії комплексних ниток та пряжі з

конструктивними елементами системи подачі ниток на круглов'язальних машинах розроблені пристрої для визначення коефіцієнту тертя та їх натягу, на які отримано патенти України №31888, №45939, №80199.

2. Сконструйовано електро-тензометричну установку для дослідження процесу взаємодії комплексних ниток та пряжі з конструктивними елементами системи подачі ниток в умовах, які максимально наближені до реальних на круглов'язальних машинах при виготовленні трикотажу.

3. Проведено вибір комплексних ниток та пряжі для експериментальних досліджень, при цьому обрані такі зразки, що за розрахунковим діаметром співпадають між собою та найбільш широко використовуються у трикотажному виробництві. Для експерименту обрано як натуральні так і хімічні нитки та пряжу: бавовняну пряжу 27,6 текс; вовняну пряжу 29,9 текс; віскозну штапельну пряжу 28,8 текс; капронову комплексну нитку 28 текс; віскозну комплексну нитку 16,7 текс.

4. Визначені фактори, які впливають на натяг комплексних ниток та пряжі при їх переробці на круглов'язальних машинах і розроблені відповідні плани проведення експериментальних досліджень, при цьому використовували активне планування експерименту при реалізації ортогональних планів 2-го порядку. Для обробки даних використовувалося спеціально розроблене програмне забезпечення.

5. Досліджено вплив конструктивних параметрів фідерних спрямовувачів нитки на її натяг, при цьому отримані відповідні регресійні залежності натягу комплексних ниток та пряжі як функцій від внутрішнього радіусу фідера, радіусу зовнішньої кривизни фідера, кута охоплення ниткою фідера, побудовані відповідні поверхні відгуку, проведена оцінка значущості коефіцієнтів в рівняннях регресії.

6. Досліджено вплив конструктивних параметрів кільцевих спрямовувачів нитки на її натяг, при цьому отримані відповідні регресійні залежності

натягу комплексних ниток та пряжі як функцій від внутрішнього радіусу спрямовувача, натягу веденої гілки, кута охоплення ниткою спрямовувача, побудовані відповідні поверхні відгуку, проведена оцінка значущості коефіцієнтів в рівняннях регресії.

8.4. УДОСКОНАЛЕННЯ СИСТЕМИ ПОДАЧІ НИТОК КРУГЛОВ'ЯЗАЛЬНИХ МАШИН

Удосконалення трикотажного обладнання, зокрема круглов'язальних машин, повинно базуватися на проведенні комплексних теоретико-експериментальних досліджень умов праці окремих вузлів в процесі виготовлення трикотажу. Система подачі ниток відіграє в цьому процесі дуже важливе значення. Це пов'язано з тим, що недосконалість конструкції системи подачі нитки призводить до зростання її натягу і, як наслідок, виникнення обривів.

Вирішення питання удосконалення системи подачі ниток на круглов'язальних машинах потрібно здійснювати у двох напрямках: удосконалення форми лінії заправки нитки (з використанням матеріалів досліджень і програмного забезпечення, які наведені в другому розділі); аналіз умов взаємодії нитки з конструктивними елементами, які входять в систему подачі нитки (спрямовувачі нитки, нитконатягувальні пристрої) та, на базі цього, оптимізація граничних умов на вході та виході з цих елементів таких параметрів як кути охоплення направляючих поверхонь, радіуси кривизни цих поверхонь (це питання розглядалося в третьому та четвертому розділах).

При проведенні цих досліджень за першим напрямом необхідно лінію заправки нитки в системі подачі розбивати на характерні ділянки, які включають в себе відповідні конструктивні елементи. Для більшості

круглов'язальних машин ці ділянки будуть мати подібний характер і відрізнятимуться тільки кількістю і видами конструктивних елементів.

Особливий характер будуть мати системи подачі нитки у в'язальних машин, які мають шпулярники розташовані на підлозі. В даному випадку, форма заправки нитки буде суттєво відрізнятися від класичної схеми, коли використовують шпулярники парасолькового типу.

Оптимізація пружної системи заправки ниток круглов'язальної машини КТ-3П

На рис.8.4.1 представлена форма лінії заправки ниток круглов'язальної машини КТ-3П, яка призначена для виготовлення трикотажних полотен футерованого одинарного, подвійного та платинованого переплетення.

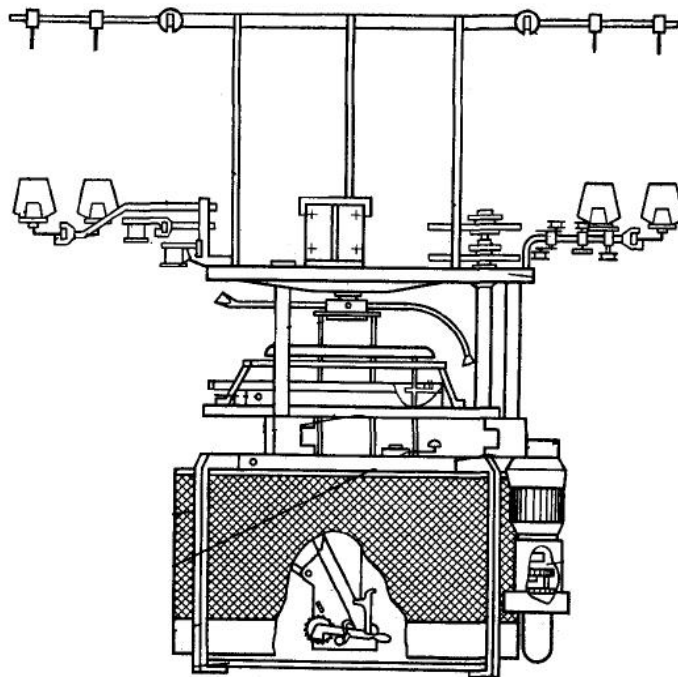


Рис.8.4.1. Форма лінії заправки ниток круглов'язальної машини КТ-3П

Лінію заправки нитки на круглов'язальній машині КТ-3П можна умовно розбити на 11 ділянок: 1 ділянка – від бобіни до спрямовувача

нитки I; 2 ділянка – від спрямовувача нитки I до вхідного отвору II торцевого кронштейну пальцевого натягувача нитки[1]; 3 ділянка – від вхідного отвору II торцевого кронштейну пальцевого натягувача нитки до циліндричної поверхні пальцевого натягувача III; 4 ділянка – від циліндричної поверхні пальцевого натягувача III до вихідного отвору IV торцевого кронштейну пальцевого натягувача нитки; 5 ділянка – від вихідного отвору IV торцевого кронштейну пальцевого натягувача нитки до вхідного отвору V контролера обриву нитки; 6 ділянка – від вхідного отвору V горизонтального контролера обриву нитки до вихідного отвору VI горизонтального контролера обриву нитки; 7 ділянка – від вихідного отвору VI горизонтального контролера обриву нитки до вертикального накопичувача нитки VII барабанного типу; 8 ділянка – від вертикального накопичувача нитки VII барабанного типу до вхідного отвору VIII вертикального контролера обриву нитки; 9 ділянка – від вхідного отвору VIII вертикального контролера обриву нитки до вихідного отвору IX вертикального контролера обриву нитки; 10 ділянка – від вихідного отвору IX вертикального контролера обриву нитки до вхідного отвору водія нитки X; 11 ділянка – від вхідного отвору водія нитки X до кулірного колеса XI.

Натяг нитки на ділянці 1 буде відповідати натягу, який має нитка при сході з бобіни [1, 4]. Взагалі ця величина має змінний характер і залежить від багатьох факторів. Для спрощення розрахунків будемо вважати її постійною. При розрахунках приймали $P1 = 5$ сН (це значення відповідає середньому значенню натягу для бавовняної пряжі 27,6 текс, з якої виготовляється значна кількість асортименту в'язаних полотен з бавовняної пряжі побутового призначення).

Натяг нитки, для круглов'язальної машини КТ-3П, при проходженні конструктивних елементів II-X(різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{kt(i+1)} = & P_{kt(i)} \left[1 + \frac{(R_{kt(j)} + r)}{[R_{kt(j)} + r(1 - \delta_{0kt(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{kt(j)}}{\sin \beta_{kt(j)}} \frac{a}{P_{kt(i)}^b} R_{kt(j)}^b \varphi_{kt(j)} - 1 \right) + \right. \\
 & \left. + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{kt(j)} + r(1 - \delta_{kt(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{kt(j)} + r(1 - \delta_{0kt(j)})]^2} \right] \times \right. \\
 & \left. \times \left[1 + \frac{(R_{kt(j)} + r)}{[R_{kt(j)} + r(1 - \delta_{0kt(j)})]} \left(e^{\frac{\beta_{kt(j)}}{\sin \beta_{kt(j)}} \frac{a}{P_{kt(i)}^b} R_{kt(j)}^b \varphi_{kt(j)} - 1 \right) \right], \right. \\
 & \left. \right) \tag{8.4.1}
 \end{aligned}$$

де $P_{kt(i+1)}$ – натяг нитки після j конструктивного елемента машини КТ-3П;

$P_{kt(i)}$ – натяг нитки до j конструктивного елемента машини КТ-3П;

$R_{kt(j)}$ – радіус кривизни поверхні j конструктивного елемента машини КТ-3П;

$\delta_{0kt(j)}$ – початкова деформація перетину нитки при набіганні на j конструктивний елемент машини КТ-3П;

$\delta_{kt(j)}$ – кінцева деформація перетину нитки при збіганні з j конструктивного елемента машини КТ-3П;

$\beta_{kt(j)}$ – кут радіального охоплення нитки поверхнею j конструктивного елемента машини КТ-3П;

$\varphi_{kt(j)}$ – реальний кут охоплення ниткою j конструктивного елемента машини КТ-3П.

Натяг нитки після вертикального накопичувача нитки VII барабанного типу визначається різницею лінійних швидкостей подачі та споживання нитки у робочій зоні. Він має змінний, пульсуючий характер. Для розрахунків можна вважати його близьким до натягу на ділянці 1.

Натяг нитки після огинання циліндричної поверхні пальцевого натягувача III можна визначити за формулою

$$P_{kt4} = P_{kt3} e^{2\pi k l \sin \varphi_{ktIII}}, \quad (8.4.2)$$

де P_{kt4} – натяг нитки перед циліндричною поверхнею пальцевого натягувача III;

P_{kt3} – натяг нитки після циліндричної поверхні пальцевого натягувача III;

$k l$ – кількість повних обертів нитки навколо циліндричної поверхні пальцевого натягувача III;

φ_{ktIII} – кут нахилу нитки до осі пальцевого натягувача III.

Використовуючи залежності (8.4.1) та (8.4.2) була отримана діаграма зміни відносного натягу нитки у зонах заправки, яка представлена на рис.8.4.2. Аналіз діаграми показав, що до вертикального накопичувача нитки VII барабанного типу натяг зростає за рахунок взаємодії нитки з поверхнями конструктивних елементів системи заправки.

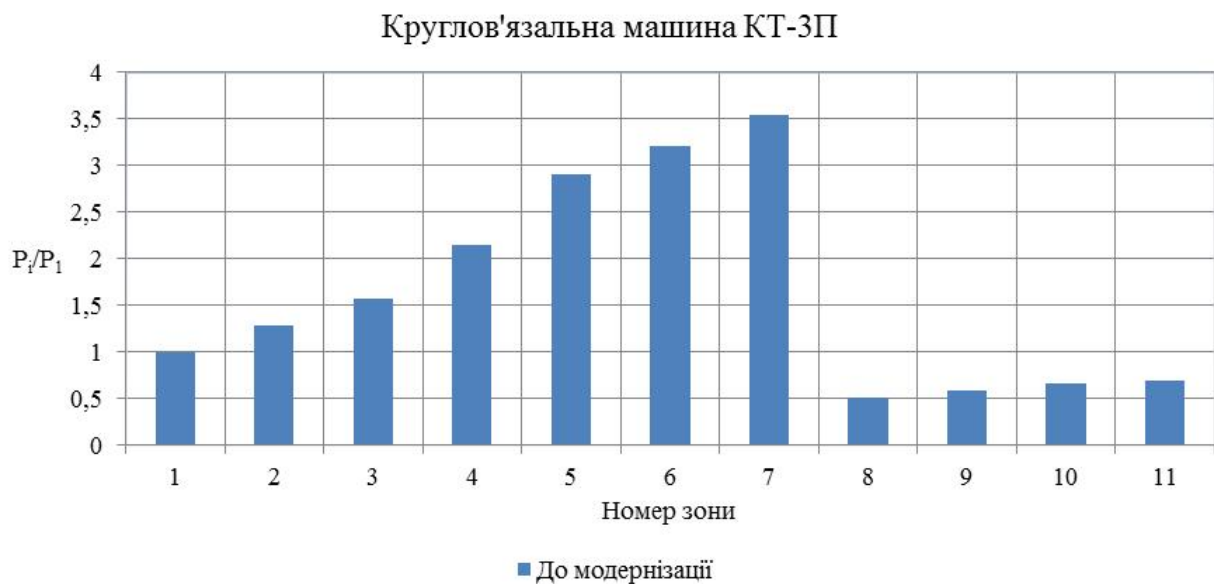


Рис.8.4.2. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини КТ-3П

Після проходження вертикального накопичувача нитки VII барабанного типу натяг починає знову зростати і досягає свого максимуму в зоні в'язання. Максимальне значення складає майже 3,5 перед 8 зоною. Виходячи з цього можна стверджувати, що накопичувач нитки VII, виконуючи свою функцію по зменшенню натягу, не виключає можливості обриву на ділянці між 1 та 7 зонами, як це видно з рис.8.4.2.

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини КТ-3П, яка наведена на рис.8.4.3.

Аналіз результатів з визначення зміни відносного натягу нитки у зонах показав, що тенденція зберігається як і у випадку до оптимізації (рис.8.4.4). Проте значення відносного натягу значно зменшилися. Так у 7 зоні відносний натяг вже складав 2,5, що майже на 18% менше ніж у випадку до оптимізації.

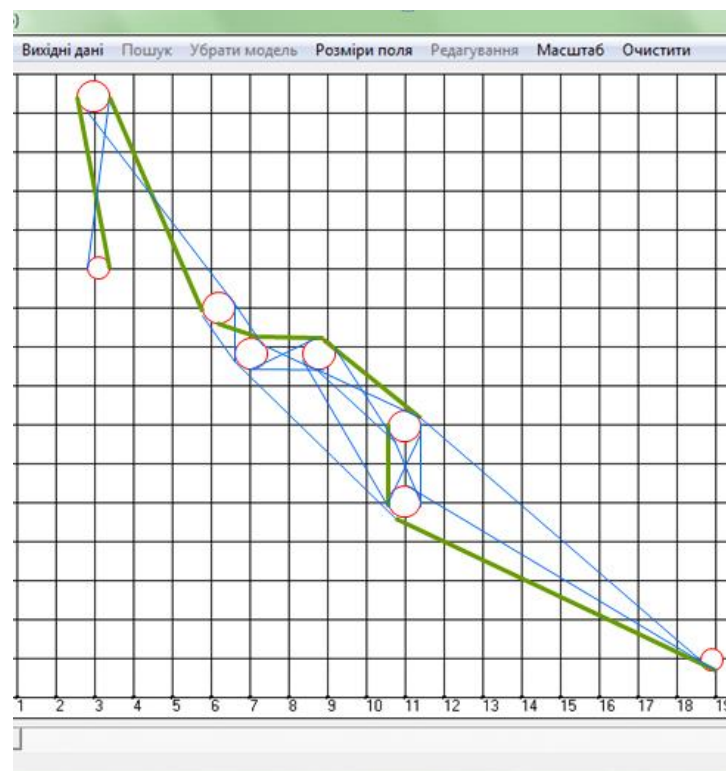


Рис.8.4.3. Оптимальна форма лінії заправки нитки для круглов'язальної машини КТ-3П

Отримані результати свідчать про те, що виникає необхідність або скорочувати довжину лінії заправки нитки, або розробляти нові конструкції структурних елементів, де в контактних парах замінювати тертя ковзання на тертя обертання.

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для круглов'язальної машини КТ-3П і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,5 сН до 7,1 сН (зменшення натягу на 17,5%).



Рис.8.4.4. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини КТ-3П після оптимізації форми лінії заправки
Удосконалення системи подачі ниток круглов'язальної машини МС

– 9

На рис.8.4.5 представлена форма лінії заправки ниток однофонтурної, круглов'язальної машини 22 класу МС – 9, яка призначена для виготовлення трикотажних полотен гладких та платинованих переплетень.

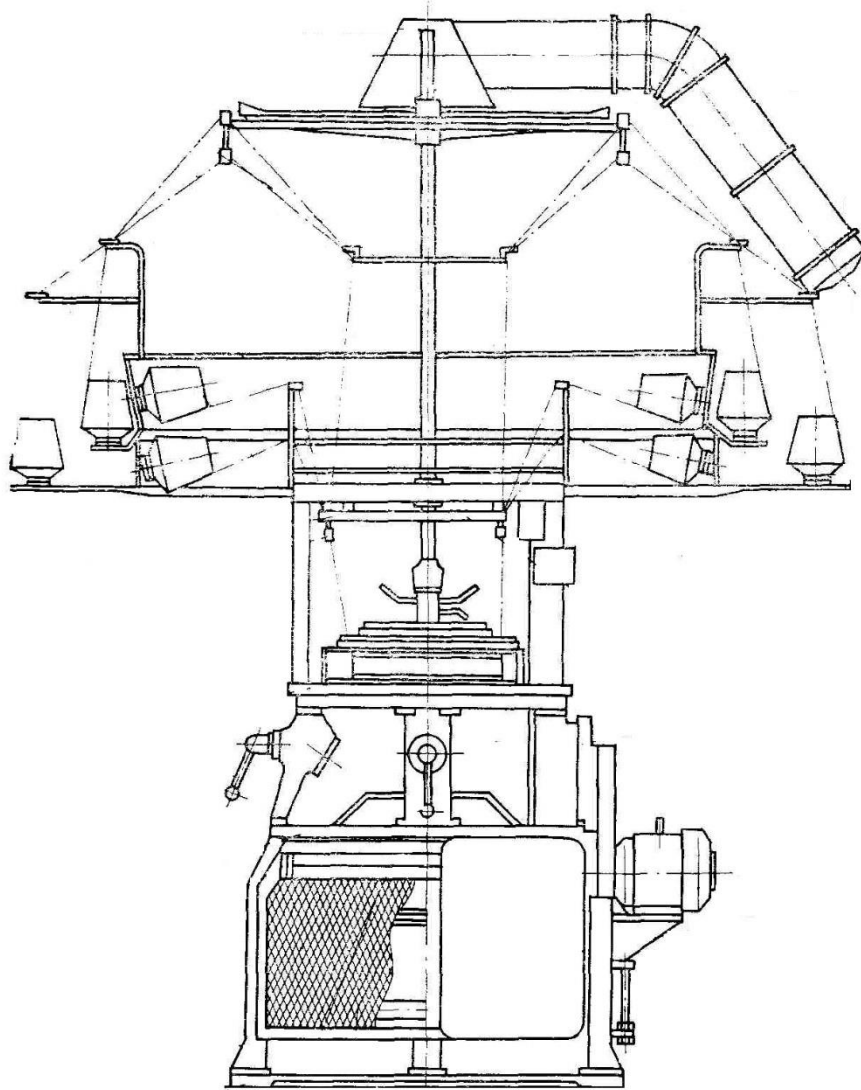


Рис.8.4.5. Форма лінії заправки ниток круглов'язальної машини МС – 9

Лінію заправки нитки на круглов'язальній машині МС – 9 можна умовно розбити на 11 ділянок: 1 ділянка – від бобіни до спрямовувача нитки I; 2 ділянка – від спрямовувача нитки I до вхідного отвору II контролера обриву нитки; 3 ділянка – від вхідного отвору II контролера обриву нитки до вихідного отвору контролера обриву нитки III; 4 ділянка – від вихідного отвору контролера обриву нитки III до спрямовувача нитки IV; 5 ділянка – від спрямовувача нитки IV до вхідного отвору на поверхні

рухомої рейки механізму розкладання нитки V; 6 ділянка – від вхідного отвору на поверхні рухомої рейки механізму розкладання нитки V до поверхні циліндричного подаючого барабана, вкритого гумою, механізму активної подачі нитки; 7 ділянка – від поверхні циліндричного подаючого барабана, вкритого гумою, механізму активної подачі нитки до вхідного отвору на поверхні рухомої рейки механізму розкладання нитки VI; 8 ділянка – від вхідного отвору на поверхні рухомої рейки механізму розкладання нитки VI до вхідного отвору VII контролера обриву нитки; 9 ділянка – від вхідного отвору VII контролера обриву нитки до вхідного отвору VIII обриву нитки; 10 ділянка – від вхідного отвору VIII контролера обриву нитки до вхідного отвору водія нитки IX; 11 ділянка – від вхідного отвору водія нитки IX до зони в'язання[1].

Для круглов'язальної машини МС – 9 натяг нитки на ділянці 1 має змінний характер і він буде відповідати натягу, який має нитка при сході з бобіни. Для спрощення розрахунків будемо вважати його постійним. При розрахунках приймали $P1 = 5$ сН (це значення відповідає середньому значенню натягу для бавовняної пряжі 27,6 текс, з якої виготовляється значна кількість асортименту в'язаних полотен з бавовняної пряжі побутового призначення на круглов'язальній машині МС – 9).

Натяг нитки, для круглов'язальної машини МС - 9, при проходженні конструктивних елементів I-IX (різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{ms(i+1)} = & P_{ms(i)} \left[1 + \frac{(R_{ms(j)} + r)}{[R_{ms(j)} + r(1 - \delta_{0ms(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{ms(j)}}{\sin \beta_{ms(j)}} \frac{a}{P_{ms(i)}^b} R_{ms(j)}^b \varphi_{ms(j)} - 1 \right) + \right. \\
 & \left. + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{ms(j)} + r(1 - \delta_{ms(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{ms(j)} + r(1 - \delta_{0ms(j)})]^2} \right] \times \right. \\
 & \left. \times \left[1 + \frac{(R_{ms(j)} + r)}{[R_{ms(j)} + r(1 - \delta_{0ms(j)})]} \left(e^{\frac{\beta_{ms(j)}}{\sin \beta_{ms(j)}} \frac{a}{P_{ms(i)}^b} R_{ms(j)}^b \varphi_{ms(j)} - 1 \right) \right], \right. \\
 & \left. \right) \quad (8.4.3)
 \end{aligned}$$

де $P_{ms(i+1)}$ – натяг нитки після j конструктивного елемента машини МС - 9;

$P_{ms(i)}$ – натяг нитки до j конструктивного елемента машини МС - 9;

$R_{ms(j)}$ – радіус кривизни поверхні j конструктивного елемента машини МС - 9;

$\delta_{0ms(j)}$ – початкова деформація перетину нитки при набіганні на j конструктивний елемент машини МС - 9;

$\delta_{ms(j)}$ – кінцева деформація перетину нитки при збіганні з j конструктивного елемента машини МС - 9;

$\beta_{ms(j)}$ – кут радіального охоплення нитки поверхнею j конструктивного елемента машини МС - 9;

$\varphi_{ms(j)}$ – реальний кут охоплення ниткою j конструктивного елемента машини МС - 9.

Натяг нитки після циліндричного подаючого барабана, вкритого гумою, механізму активної подачі нитки машини МС - 9 визначається різницею лінійних швидкостей точок на поверхні барабану та споживання нитки у робочій зоні. Коли лінійна швидкість точок на поверхні барабану буде більше швидкості споживання нитки у робочій зоні натяг буде

нижчим по співвідношенню до випадку, коли швидкість точок на поверхні барабану буде менше швидкості споживання нитки у робочій зоні.

Тому натяг нитки на ділянці 7, після огинання циліндричного подаючого барабана, вкритого гумою, механізму активної подачі нитки машини МС – 9, можна визначити за формулою

$$P_{ms7} = E(V_{msII} - V_{msS})t, \quad (8.4.4)$$

де P_{ms7} – натяг нитки на ділянці 7, після огинання циліндричного подаючого барабана, вкритого гумою, механізму активної подачі нитки машини МС – 9;

E – лінійний модуль пружності нитки та пряжі при її розтягуванні;

V_{msII} – швидкість точок на поверхні барабану;

V_{msS} – швидкість споживання нитки у робочій зоні;

t – час.

Використовуючи залежності (8.4.3) та (8.4.4) була отримана діаграма зміни відносного натягу нитки у зонах заправки, яка представлена на рис.8.4.6. Аналіз діаграми показав, що до механізму активної подачі нитки з горизонтальними барабанами машини МС - 9 натяг зростає за рахунок взаємодії нитки з поверхнями конструктивних елементів системи заправки.



Рис.8.4.6. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини МС – 9

Після проходження механізму активної подачі нитки з горизонтальними барабанами машини МС - 9 натяг починає знову зростати і досягає свого максимуму в зоні в'язання. Максимальне значення складає 1,6 перед 7 зоною. Виходячи з цього можна стверджувати, що механізм активної подачі нитки з горизонтальними барабанами машини МС - 9, виконуючи свою функцію по зменшенню натягу, не виключає можливості обриву на ділянці між 1 та 6 зонами, як це видно з рис.8.4.6.

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини МС - 9, яка наведена на рис.8.4.7.

Аналіз результатів з визначення зміни відносного натягу нитки у зонах показав, що тенденція зберігається як і у випадку до оптимізації. Проте значення відносного натягу значно зменшилися. Так у 6 зоні відносний натяг вже складав 1,5, що майже на 3 % менше ніж у випадку до оптимізації (рис.8.4.8).

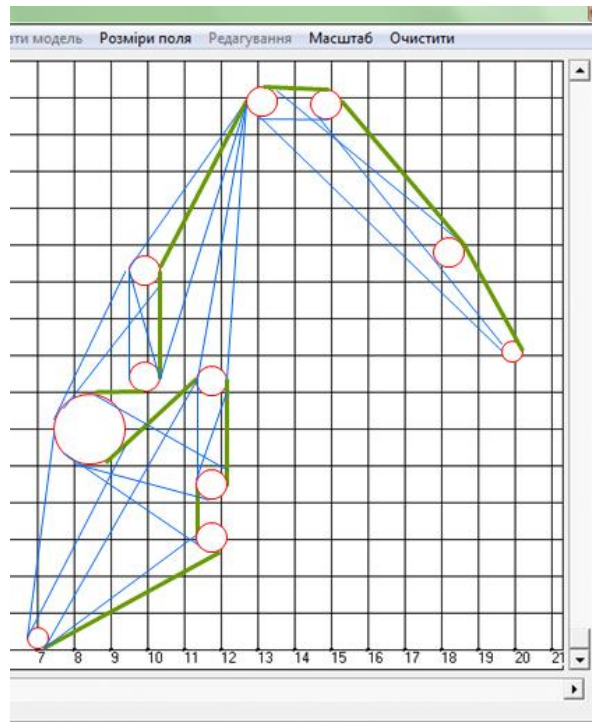


Рис.8.4.7. Оптимальна форма лінії заправки нитки для круглов'язальної машини МС – 9

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для круглов'язальної машини МС - 9 і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 6,3 сН до 6,0 сН (зменшення натягу на 3%).



Рис.8.4.8. Діаграма зміни відносного натягу нитки у зонах заправки

круглов'язальної машини МС – 9 після оптимізації форми лінії заправки

Оптимізація пружної системи заправки ниток круглов'язальної машини ДЛ – 4М

На рис.8.4.9 представлена форма лінії заправки ниток круглов'язальної, двофонтурної машини ДЛ–4М, яка призначена для в'язання полотна дволастичним переплетенням, з якого виготовляють вироби білизна та спортивний одяг.

Лінію заправки нитки на круглов'язальній машині ДЛ–4М можна умовно розбити на 14 ділянок: 1 ділянка – від бобіни до спрямовувача нитки I; 2 ділянка – від спрямовувача нитки I до вхідного отвору контролера обриву нитки II; 3 ділянка – від вихідного отвору контролера обриву нитки III до спрямовувача нитки IV; 4 ділянка – від спрямовувача нитки IV до спрямовувача нитки V; 5 ділянка – від спрямовувача нитки V до вхідного отвору VI торцевого кронштейну пальцевого натягувача нитки[133]; 6 ділянка – від вхідного отвору VI торцевого кронштейну пальцевого натягувача нитки до циліндричної поверхні пальцевого натягувача; 7 ділянка – від циліндричної поверхні пальцевого натягувача до вихідного отвору VII торцевого кронштейну пальцевого натягувача нитки; 8 ділянка – від вихідного отвору VII торцевого кронштейну пальцевого натягувача нитки до вхідного отвору VIII контролера обриву нитки; 9 ділянка – від вхідного отвору VIII контролера обриву нитки до вихідного отвору IX контролера обриву нитки; 10 ділянка – від вихідного отвору IX контролера обриву нитки до вертикального накопичувача нитки барабанного типу; 11 ділянка – від вертикального накопичувача нитки барабанного типу до вхідного отвору X контролера обриву нитки; 12 ділянка – від вхідного отвору X контролера обриву нитки до вихідного отвору XI контролера обриву нитки; 13 ділянка – від вихідного отвору XI

контролера обриву нитки до вхідного отвору водія нитки XII; 14 ділянка – від вхідного отвору водія нитки XII до зони в'язання.

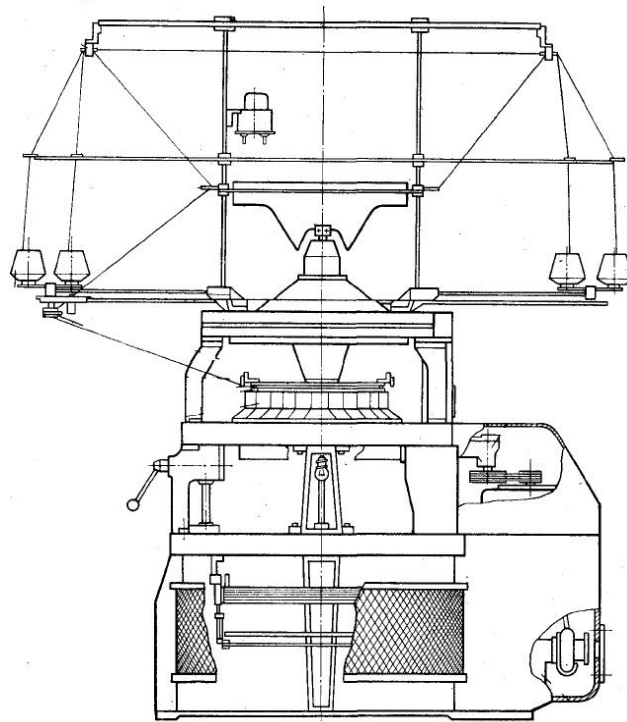


Рис.8.4.9. Форма лінії заправки ниток круглов'язальної машини ДЛ-4М

Натяг нитки на ділянці 1, як це зазначалося для попередніх круглов'язальних машин, буде відповідати натягу, який має нитка при сході з бобіни . Взагалі ця величина має змінний характер і залежить від багатьох факторів. При розрахунках приймали $P1 = 5$ сН.

Натяг нитки, для круглов'язальної машини ДЛ-4М, при проходженні конструктивних елементів I-XII (різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{dl(i+1)} = & P_{dl(i)} \left[1 + \frac{(R_{dl(j)} + r)}{[R_{dl(j)} + r(1 - \delta_{0dl(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{dl(j)}}{\sin \beta_{dl(j)}} \frac{a}{P_{dl(i)}^b} R_{dl(j)}^b \varphi_{dl(j)} - 1 \right) + \right. \\
 & \left. + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{dl(j)} + r(1 - \delta_{dl(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{dl(j)} + r(1 - \delta_{0dl(j)})]^2} \right] \times \right. \\
 & \left. \times \left[1 + \frac{(R_{dl(j)} + r)}{[R_{dl(j)} + r(1 - \delta_{0dl(j)})]} \left(e^{\frac{\beta_{dl(j)}}{\sin \beta_{dl(j)}} \frac{a}{P_{dl(i)}^b} R_{dl(j)}^b \varphi_{dl(j)} - 1 \right) \right], \right. \\
 & \left. \right) \tag{8.4.5}
 \end{aligned}$$

де $P_{dl(i+1)}$ – натяг нитки після j конструктивного елемента машини ДЛ–4М;

$P_{dl(i)}$ – натяг нитки до j конструктивного елемента машини ДЛ–4М;

$R_{dl(j)}$ – радіус кривизни поверхні j конструктивного елемента машини ДЛ–4М;

$\delta_{0dl(j)}$ – початкова деформація перетину нитки при набіганні на j конструктивний елемент машини ДЛ–4М;

$\delta_{dl(j)}$ – кінцева деформація перетину нитки при збіганні з j конструктивного елемента машини ДЛ–4М;

$\beta_{dl(j)}$ – кут радіального охоплення нитки поверхнею j конструктивного елемента машини ДЛ–4М;

$\varphi_{dl(j)}$ – реальний кут охоплення ниткою j конструктивного елемента машини ДЛ–4М.

Натяг нитки після вертикального накопичувача нитки барабанного типу машини ДЛ–4М визначається різницею лінійних швидкостей подачі та споживання нитки у робочій зоні. Для розрахунків можна вважати його близьким до натягу на ділянці 1.

Натяг нитки на ділянці 7 після огинання циліндричної поверхні пальцевого натягувача можна визначити за формулою

$$P_{dl7} = P_{dl6} e^{2\pi k l \sin \varphi_{dl7}}, \quad (8.4.6)$$

де P_{dl7} – натяг нитки перед циліндричною поверхнею пальцевого натягувача;

P_{dl6} – натяг нитки після циліндричної поверхні пальцевого натягувача;

$k l$ – кількість повних обертів нитки навколо циліндричної поверхні пальцевого натягувача;

φ_{dl7} – кут нахилу нитки до осі пальцевого натягувача.

Використовуючи залежності (8.4.5) та (8.4.6) була отримана діаграма зміни відносного натягу нитки для круглов'язальної машини ДЛ-4М у зонах заправки, яка представлена на рис.8.4.10. Аналіз діаграми показав, що до вертикального накопичувача нитки барабанного типу натяг зростає за рахунок взаємодії нитки з поверхнями конструктивних елементів системи заправки.



Рис.8.4.10. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини ДЛ-4М

Після проходження вертикального накопичувача нитки барабанного типу натяг починає знову зростати і досягає свого максимуму в зоні в'язання. Максимальне значення складає майже 3,4 перед 11 зоною.

Виходячи з цього можна стверджувати, що накопичувач нитки барабанного типу, виконуючи свою функцію по зменшенню натягу, не виключає можливості обриву на ділянці між 1 та 10 зонами, як це видно з рис.8.4.10.

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини ДЛ-4М, яка наведена на рис.8.4.11.

Аналіз результатів з визначення зміни відносного натягу нитки у зонах показав, що тенденція зберігається як і у випадку до оптимізації. Проте значення відносного натягу значно зменшилися. Так у 10 зоні відносний натяг вже складав 2,7, що майже на 9% менше ніж у випадку до оптимізації (рис.8.4.12).

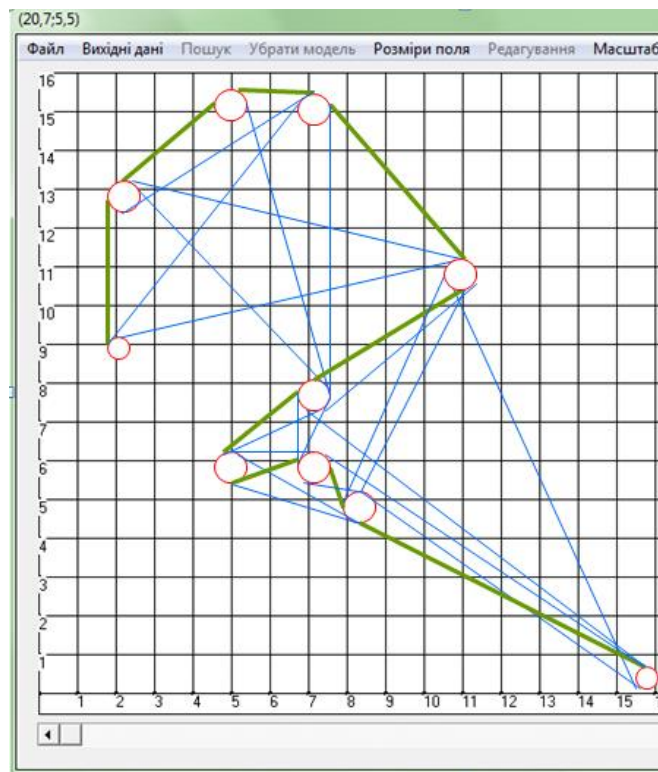


Рис.8.4.11. Оптимальна форма лінії заправки нитки для круглов'язальної машини ДЛ-4М

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для

круглов'язальної машини ДЛ-4М і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 9,3 сН до 8,5 сН (зменшення натягу на 9 %).

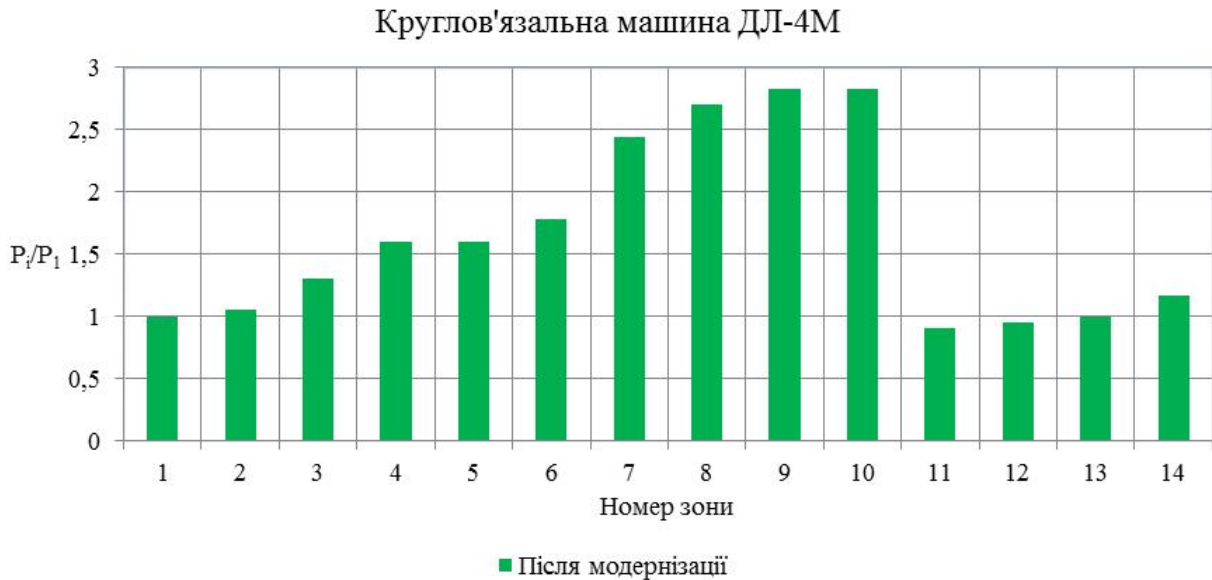


Рис.8.4.12. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини ДЛ-4М після оптимізації форми лінії заправки

Удосконалення системи подачі ниток круглов'язальної машини КЛК – 10

На рис.8.4.13 представлена форма лінії заправки ниток круглов'язальної, двофонтурної машини КЛК – 10, яка призначена для в'язання купонних верхніх виробів.

Лінію заправки нитки на круглов'язальній машині КЛК – 10 можна умовно розбити на 9 ділянок: 1 ділянка – від бобіни до вхідного отвору I торцевого кронштейну пальцевого натягувача нитки; 2 ділянка - від вхідного отвору I торцевого кронштейну пальцевого натягувача нитки до циліндричної поверхні пальцевого натягувача II; 3 ділянка – від циліндричної поверхні пальцевого натягувача II до вихідного отвору III торцевого кронштейну пальцевого натягувача нитки; 4 ділянка – від

вихідного отвору III торцевого кронштейну пальцевого натягача нитки до вхідного отвору контролера обриву нитки IV; 5 ділянка – від вхідного отвору контролера обриву нитки IV до вихідного отвору контролера обриву нитки V; 6 ділянка – від вихідного отвору контролера обриву нитки V до спрямовувача нитки VI; 7 ділянка – від спрямовувача нитки VI до спрямовувача нитки VII; 8 ділянка – від спрямовувача нитки VII до вхідного отвору водія нитки VIII; 9 ділянка – від вхідного отвору водія нитки VIII до зони в'язання.

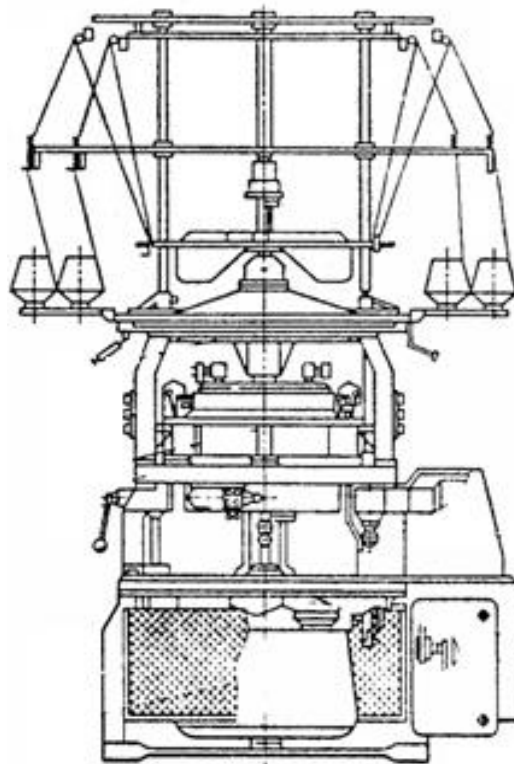


Рис.8.4.13. Форма лінії заправки ниток круглов'язальної машини КЛК – 10

Вхідний натяг нитки на ділянці 1 дорівнював $P_1 = 5$ сН. Натяг нитки, для круглов'язальної машини КЛК – 10, при проходженні конструктивних елементів I, III-VIII (різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{kl(i+1)} = & P_{kl(i)} \left[1 + \frac{(R_{kl(j)} + r)}{[R_{kl(j)} + r(1 - \delta_{0kl(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{kl(j)}}{\sin \beta_{kl(j)}} \frac{a}{P_{kl(i)}^b} R_{kl(j)}^b \varphi_{kl(j)}} - 1 \right) + \right. \\
 & + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{kl(j)} + r(1 - \delta_{kl(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{kl(j)} + r(1 - \delta_{0kl(j)})]^2} \right] \times \\
 & \left. \times \left[1 + \frac{(R_{kl(j)} + r)}{[R_{kl(j)} + r(1 - \delta_{0kl(j)})]} \left(e^{\frac{\beta_{kl(j)}}{\sin \beta_{kl(j)}} \frac{a}{P_{kl(i)}^b} R_{kl(j)}^b \varphi_{kl(j)}} - 1 \right) \right], \right. \quad (8.4.7) \\
 & \left. \right)
 \end{aligned}$$

де $P_{kl(i+1)}$ – натяг нитки після j конструктивного елемента машини КЛК – 10;

$P_{kl(i)}$ – натяг нитки до j конструктивного елемента машини КЛК – 10;

$R_{kl(j)}$ – радіус кривизни поверхні j конструктивного елемента машини КЛК – 10;

$\delta_{0kl(j)}$ – початкова деформація перетину нитки при набіганні на j конструктивний елемент машини КЛК – 10;

$\delta_{kl(j)}$ – кінцева деформація перетину нитки при збіганні з j конструктивного елемента машини КЛК – 10;

$\beta_{kl(j)}$ – кут радіального охоплення нитки поверхнею j конструктивного елемента машини КЛК – 10;

$\varphi_{kl(j)}$ – реальний кут охоплення ниткою j конструктивного елемента машини КЛК – 10.

Натяг нитки на ділянці 3 після огинання циліндричної поверхні пальцевого натягувача можна визначити за формулою

$$P_{kl3} = P_{kl2} e^{2\pi k l \sin \varphi_{kIII}}, \quad (8.4.8)$$

де P_{kl3} – натяг нитки перед циліндричною поверхнею пальцевого натягувача;

P_{kl2} – натяг нитки після циліндричної поверхні пальцевого натягувача;

$k1$ – кількість повних обертів нитки навколо циліндричної поверхні пальцевого натягувача;

φ_{kl1} – кут нахилу нитки до осі пальцевого натягувача.

Використовуючи залежності (8.4.7) та (8.4.8) була отримана діаграма зміни відносного натягу нитки для круглов'язальної машини КЛК – 10 у зонах заправки, яка представлена на рис.8.4.14. Аналіз діаграми показав, що натяг зростає за рахунок взаємодії нитки з поверхнями конструктивних елементів системи заправки.

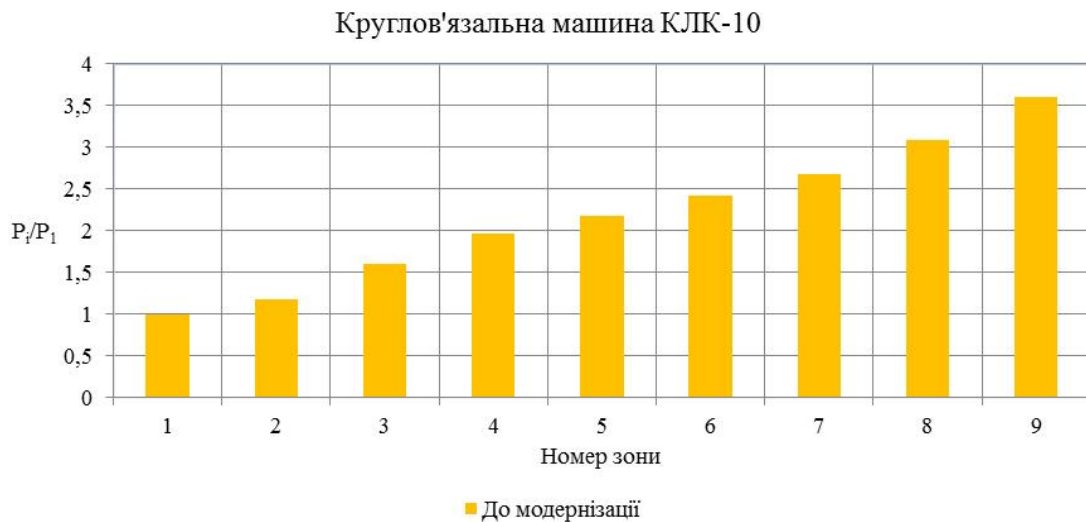


Рис.8.4.14. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини КЛК – 10

Максимальне значення відносного натягу складає майже 3,7 в 9 зоні. Виходячи з цього можна стверджувати, що не виключається можливість обриву на ділянці між 1 та 9 зонами, як це видно з рис.8.4.14. Все це характерно для пасивної подачі нитки на круглов'язальній машині КЛК – 10.

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини КЛК – 10, яка наведена на рис.8.4.18.4.

Аналіз результатів з визначення зміни відносного натягу нитки, після оптимізації конструктивних параметрів системи подачі нитки, у зонах показав, що тенденція зберігається як і у випадку до оптимізації. Проте значення відносного натягу значно зменшилися. Так у 9 зоні відносний натяг вже складав 2,8, що майже на 14% менше ніж у випадку до оптимізації (рис.8.4.16).

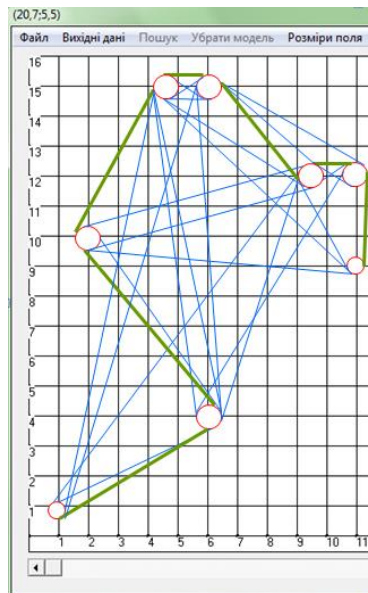


Рис.8.4.18. Оптимальна форма лінії заправки нитки для круглов'язальної машини КЛК – 10

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для круглов'язальної машини КЛК – 10 і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 11,7 сН до 10,1 сН (зменшення натягу на 13,2 %).



Рис.8.4.16. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини КЛК – 10 після оптимізації форми лінії заправки

Оптимізація пружної системи заправки ниток круглов'язальної машини MAYER

На рис.8.4.17 представлена форма лінії заправки ниток круглов'язальної машини Mayer, яка може бути однофонтурна чи двофонтурна і включати у свою конструкцію до 114 транспортуючих нитку фідерів. Вона призначена для виготовлення трикотажних жакардових полотен, покриваючих тканин.

Лінію заправки нитки на круглов'язальній машині Mayer можна умовно розбити на 12 ділянок: 1 ділянка – від бобіни до вхідного отвору I фідера; 2 ділянка – від вхідного отвору I фідера до перехідного коліна фідера II ; 3 ділянка – від перехідного коліна фідера II до перехідного коліна фідера III; 4 ділянка – від перехідного коліна фідера III до шайбового натягувача нитки IV; 5 ділянка – від шайбового натягувача нитки IV до вхідного отвору V спрямовувача нитки; 6 ділянка – від вхідного отвору V спрямовувача нитки до вертикального накопичувача нитки VI барабанного типу; 7 ділянка – від вертикального накопичувача нитки VI барабанного типу до вхідного отвору VII спрямовувача нитки; 8

ділянка – від вхідного отвору VII спрямовувача нитки до вхідного отвору VIII спрямовувача нитки; 9 ділянка – від вхідного отвору VIII спрямовувача нитки до вхідного отвору IX контролера обриву нитки; 10 ділянка – від вхідного отвору IX контролера обриву нитки до вихідного отвору X контролера обриву нитки; 11 ділянка – від вихідного отвору X контролера обриву нитки до вхідного отвору XI водія нитки; 12 ділянка – від вхідного отвору XI водія нитки до зони в'язання.



Рис.8.4.17. Форма лінії заправки ниток круглов'язальної машини Мауер

Натяг нитки на ділянці 1 буде відповідати натягу, який має бавовняна пряжа 27,6 текс при сході з бобіни. При розрахунках приймали $P1 = 5$ сН.

Натяг нитки, для круглов'язальної машини Мауер, при проходженні конструктивних елементів I-III, V, VII-XI(різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{ma(i+1)} = & P_{ma(i)} \left[1 + \frac{(R_{ma(j)} + r)}{[R_{ma(j)} + r(1 - \delta_{0ma(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{ma(j)}}{\sin \beta_{ma(j)}} \frac{a}{P_{ma(i)}^b} R_{ma(j)}^b \varphi_{ma(j)} - 1 \right) + \right. \\
 & + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{ma(j)} + r(1 - \delta_{ma(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{ma(j)} + r(1 - \delta_{0ma(j)})]^2} \right] \times \\
 & \left. \times \left[1 + \frac{(R_{ma(j)} + r)}{[R_{ma(j)} + r(1 - \delta_{0ma(j)})]} \left(e^{\frac{\beta_{ma(j)}}{\sin \beta_{ma(j)}} \frac{a}{P_{ma(i)}^b} R_{ma(j)}^b \varphi_{ma(j)} - 1 \right) \right], \right. \quad (8.4.9)
 \end{aligned}$$

де $P_{ma(i+1)}$ – натяг нитки після j конструктивного елемента машини Mayer;

$P_{ma(i)}$ – натяг нитки до j конструктивного елемента машини Mayer;

$R_{ma(j)}$ – радіус кривизни поверхні j конструктивного елемента машини Mayer;

$\delta_{0ma(j)}$ – початкова деформація перетину нитки при набіганні на j конструктивний елемент машини Mayer;

$\delta_{ma(j)}$ – кінцева деформація перетину нитки при збіганні з j конструктивного елемента машини Mayer;

$\beta_{ma(j)}$ – кут радіального охоплення нитки поверхнею j конструктивного елемента машини Mayer;

$\varphi_{ma(j)}$ – реальний кут охоплення ниткою j конструктивного елемента машини Mayer.

Натяг нитки після вертикального накопичувача нитки VI барабанного типу машини Mayer визначається різницею лінійних швидкостей подачі та споживання нитки у робочій зоні. Він має змінний, пульсуючий характер. Для розрахунків можна вважати його близьким до натягу на ділянці 1.

Натяг нитки після проходження шайбового натягувача IV можна визначити за формулою

$$P_{ma5} = P_{ma4} e^{\mu_2 \alpha_{maIV}} + \frac{\mu_1 N (1 + e^{\mu_2 \alpha_{maIV}})}{[1 + \sin(\frac{\alpha_{maIV}}{2} - \arcsin \frac{r}{R})]}, \quad (8.4.10)$$

де P_{ma5} – натяг нитки після шайбового натягувача IV;

P_{ma4} – натяг нитки до шайбового натягувача IV;

μ_1 – коефіцієнт тертя між ниткою та поверхнею шайб;

μ_2 – коефіцієнт тертя між ниткою та циліндричною поверхнею стрижня, на який одягнені шайби;

N – сила нормального тиску шайб на нитку в точках контакту;

r – радіус стрижня;

R – радіус контактного кола;

α_{maIV} – кут охоплення ниткою стрижня.

Використовуючи залежності (8.4.9) та (8.4.10) була отримана діаграма зміни відносного натягу нитки у зонах заправки, яка представлена на рис.8.4.18. Аналіз діаграми показав, що до вертикального накопичувача нитки VI барабанного типу натяг зростає за рахунок взаємодії нитки з поверхнями фідерів та конструктивних елементів системи заправки.

Після проходження вертикального накопичувача нитки VI барабанного типу круглов'язальної машини Mayer натяг починає знову зростати і досягає свого максимуму в зоні в'язання. Максимальне значення відносного натягу складає майже 3,8 перед 7 зоною. Виходячи з цього можна стверджувати, що накопичувач нитки VI круглов'язальної машини Mayer, виконуючи свою функцію по зменшенню натягу, не виключає можливості обриву на ділянці між 1 та 6 зонами, як це видно з рис.8.4.18.



Рис.8.4.18. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини Mayer

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини Mayer, яка наведена на рис.8.4.19.

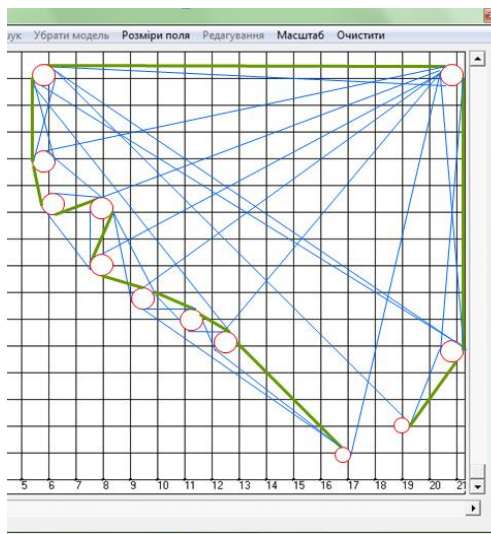


Рис.8.4.19. Оптимальна форма лінії заправки нитки для круглов'язальної машини Mayer

Аналіз результатів з визначення зміни відносного натягу нитки у зонах після оптимізації показав, що тенденція зберігається як і у випадку до оптимізації (рис.8.4.20). Проте значення відносного натягу значно зменшилися. Так у 6 зоні відносний натяг вже складав 3,4, що майже на 9% менше ніж у випадку до оптимізації.

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для круглов'язальної машини Mayer і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,6 сН до 7,8 сН (зменшення натягу на 8,5%).



Рис.8.4.20. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини Mayer після оптимізації форми лінії заправки

Удосконалення системи подачі ниток круглов'язальної машини PaiLung

На рис.8.4.21 представлена форма лінії заправки ниток круглов'язальної машини PaiLung. Вона призначена для виготовлення трикотажних полотен кулірною гладдю.

Лінію заправки нитки на круглов'язальній машині PaiLung можна умовно розбити на 12 ділянок: 1 ділянка – від бобіни до вхідного отвору I фідера машини PaiLung; 2 ділянка – від вхідного отвору I фідера до перехідного коліна фідера II; 3 ділянка – від перехідного коліна фідера II до перехідного коліна фідера III машини PaiLung; 4 ділянка – від

перехідного коліна фідера III до шайбового натягувала нитки IV; 5 ділянка – від шайбового натягувача нитки IV до вхідного отвору V спрямовувача нитки; 6 ділянка – від вхідного отвору V спрямовувача нитки до вертикального накопичувача нитки VI барабанного типу машини PaiLung; 7 ділянка – від вертикального накопичувача нитки VI барабанного типу до вхідного отвору VII спрямовувача нитки; 8 ділянка – від вхідного отвору VII спрямовувача нитки до вхідного отвору VIII спрямовувача нитки; 9 ділянка – від вхідного отвору VIII спрямовувача нитки до вхідного отвору IX контролера обриву нитки; 10 ділянка – від вхідного отвору IX контролера обриву нитки до вихідного отвору X контролера обриву нитки машини PaiLung; 11 ділянка – від вихідного отвору X контролера обриву нитки до вхідного отвору XI водія нитки; 12 ділянка – від вхідного отвору XI водія нитки до зони в'язання машини PaiLung.

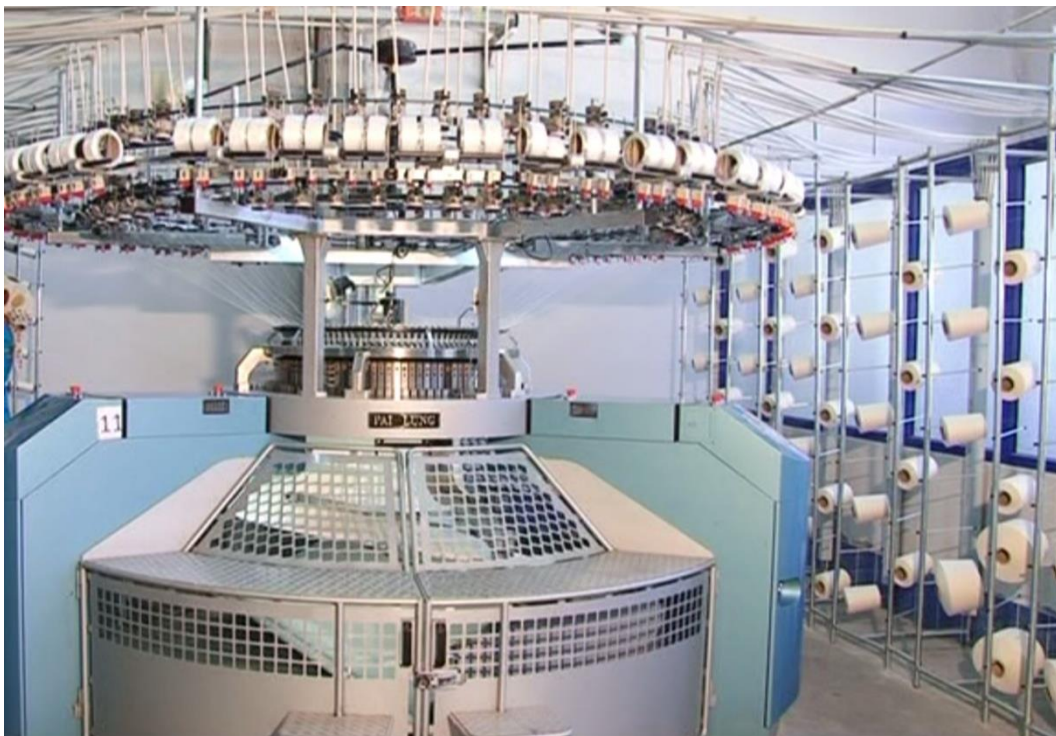


Рис.8.4.21. Форма лінії заправки ниток круглов'язальної машини PaiLung

Натяг нитки на ділянці 1 буде відповідати натягу, який має бавовняна пряжа 27,6 текс при сході з бобіни. При розрахунках приймали $P_1 = 5$ сН.

Натяг нитки, для круглов'язальної машини PaiLung, при проходженні конструктивних елементів I-III, V, VII-XI(різного типу спрямовувачі нитки), з урахуванням (8.2.26)-(8.2.27), визначимо з рівняння

$$\begin{aligned}
 P_{pa(i+1)} = & P_{pa(i)} \left[1 + \frac{(R_{pa(j)} + r)}{[R_{pa(j)} + r(1 - \delta_{0pa(j)})]} \times \right. \\
 & \left. \times \left(e^{\frac{\beta_{pa(j)}}{\sin \beta_{pa(j)}} \frac{a}{P_{pa(i)}^b} R_{pa(j)}^b \varphi_{pa(j)} - 1 \right) + \right. \\
 & \left. + \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{pa(j)} + r(1 - \delta_{0pa(j)})]^2} \right] - \left[\frac{(c_0 + c_1 K_r + c_2 K_r^2)}{2[R_{pa(j)} + r(1 - \delta_{0pa(j)})]^2} \right] \times \right. \\
 & \left. \times \left[1 + \frac{(R_{pa(j)} + r)}{[R_{pa(j)} + r(1 - \delta_{0pa(j)})]} \left(e^{\frac{\beta_{pa(j)}}{\sin \beta_{pa(j)}} \frac{a}{P_{pa(i)}^b} R_{pa(j)}^b \varphi_{pa(j)} - 1 \right) \right], \right. \quad (8.4.1) \\
 & \left. 1) \right.
 \end{aligned}$$

де $P_{pa(i+1)}$ – натяг нитки після j фідеру чи конструктивного елементу машини PaiLung;

$P_{pa(i)}$ – натяг нитки до j фідеру чи конструктивного елементу машини PaiLung;

$R_{pa(j)}$ – радіус кривизни поверхні j фідеру чи конструктивного елементу машини PaiLung;

$\delta_{0pa(j)}$ – початкова деформація перетину нитки при набіганні на j фідер чи конструктивний елемент машини PaiLung;

$\delta_{pa(j)}$ – кінцева деформація перетину нитки при збіганні з j фідеру чи конструктивного елементу машини PaiLung;

$\beta_{pa(j)}$ – кут радіального охоплення нитки поверхнею j фідеру чи конструктивного елементу машини PaiLung;

$\varphi_{pa(j)}$ – реальний кут охоплення ниткою j фідеру чи конструктивного елемента машини PaiLung.

Натяг нитки після вертикального накопичувача нитки VI барабанного типу машини PaiLung визначається різницею лінійних швидкостей подачі та споживання нитки у робочій зоні. Він має змінний, пульсуючий характер. Для розрахунків можна вважати його близьким до натягу на ділянці 1.

Натяг нитки після проходження шайбового натягувача IV можна визначити за формулою

$$P_{pa5} = P_{pa4} e^{\mu_2 \alpha_{paIV}} + \frac{\mu_1 N (1 + e^{\mu_2 \alpha_{paIV}})}{[1 + \sin(\frac{\alpha_{paIV}}{2} - \arcsin \frac{r}{R})]}, \quad (8.4.1)$$

2)

де P_{pa5} – натяг нитки після шайбового натягувача IV;

P_{pa4} – натяг нитки до шайбового натягувача IV;

μ_1 – коефіцієнт тертя між ниткою та поверхнею шайб;

μ_2 – коефіцієнт тертя між ниткою та циліндричною поверхнею стрижня, на який одягнені шайби;

N – сила нормального тиску шайб на нитку в точках контакту;

r – радіус стрижня;

R – радіус контактної точки;

α_{paIV} – кут охоплення ниткою стрижня.

Використовуючи залежності (8.4.11) та (8.4.12) була отримана діаграма зміни відносного натягу нитки у зонах заправки, яка представлена на рис.8.4.22. Аналіз діаграми показав, що до вертикального накопичувача нитки VI барабанного типу натяг зростає за рахунок взаємодії нитки з поверхнями фідерів та конструктивних елементів системи заправки.

Після проходження вертикального накопичувача нитки VI барабанного типу круглов'язальної машини PaiLung натяг починає знову

зростати і досягає свого максимуму в зоні в'язання. Максимальне значення відносного натягу складає майже 3,6 перед 7 зоною. Виходячи з цього можна стверджувати, що накопичувач нитки VI круглов'язальної машини PaiLung, виконуючи свою функцію по зменшенню натягу, не виключає можливості обриву на ділянці між 1 та 6 зонами, як це видно з рис.8.4.22.



Рис.8.4.22. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини PaiLung

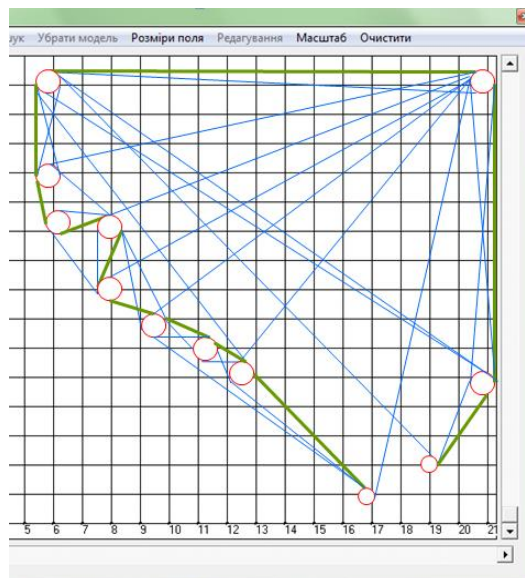


Рис.8.4.23. Оптимальна форма лінії заправки нитки для круглов'язальної машини PaiLung

Використовуючи алгоритм та програмне забезпечення, які розглядалися в другому розділі, отримана оптимальна форма лінії заправки нитки для круглов'язальної машини PaiLung, яка наведена на рис.8.4.23.

Аналіз результатів з визначення зміни відносного натягу нитки у зонах після оптимізації показав, що тенденція зберігається як і у випадку до оптимізації (рис.8.4.24). Проте значення відносного натягу значно зменшилися. Так у 6 зоні відносний натяг вже складав 3,3, що майже на 8,5% менше ніж у випадку до оптимізації.

Проведені дослідження дозволили обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки для круглов'язальної машини PaiLung і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,4 сН до 7,7 сН (зменшення натягу на 8,48%).



Рис.8.4.24. Діаграма зміни відносного натягу нитки у зонах заправки круглов'язальної машини PaiLung після оптимізації форми лінії заправки

Висновки по розділу

1. Проведений аналіз системи подачі нитки на круглов'язальній однофонтурній машині КТ-3П дозволив вперше отримати систему рівнянь, яка описує залежність натягу в зоні в'язання від форми заправки нитки та

конструктивних параметрів структурних елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, що дозволило обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,5 сН до 7,1 сН (зменшення натягу на 17,5%) та зменшити обривність нитки на 7 %.

2. Аналіз системи подачі нитки на круглов'язальній однофонтурній машині МС – 9 дозволив вперше отримати систему рівнянь, яка описує залежність натягу в зоні в'язання в залежності від форми заправки нитки та конструктивних параметрів структурних елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, що дозволило обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 6,3 сН до 6,0 сН (зменшення натягу на 3 %) та зменшити обривність нитки на 3%.

3. Вперше отримана система рівнянь, яка описує залежність натягу в зоні в'язання на круглов'язальній машині ДЛ – 4М, що дозволило в залежності від форми заправки нитки та конструктивних параметрів структурних елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 9,3 сН до 8,5 сН (зменшення натягу на 9 %) та зменшити обривність нитки на 4 %.

4. Проведений аналіз системи подачі нитки на круглов'язальній однофонтурній машині КЛК – 10 дозволив вперше отримати систему рівнянь, яка описує залежність натягу в зоні в'язання в залежності від форми заправки нитки та конструктивних параметрів структурних

елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, що дозволило обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 11,7 сН до 10,1 сН (зменшення натягу на 13,2 %) та зменшити обривність нитки на 6%.

5. Аналіз системи подачі нитки на круглов'язальній однофонтурній машині Мауер дозволив вперше отримати систему рівнянь, яка описує залежність натягу в зоні в'язання в залежності від форми заправки нитки та конструктивних параметрів структурних елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, що дозволило обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,6 сН до 7,8 сН (зменшення натягу на 8,5%) та зменшити обривність нитки на 5%.

6. Вперше отримана система рівнянь, яка описує залежність натягу в зоні в'язання на круглов'язальній машині PaiLung, що дозволило в залежності від форми заправки нитки та конструктивних параметрів структурних елементів системи подачі з урахуванням реальних кутів радіального охоплення та фізико-механічних властивостей нитки, обрати оптимальну форму та геометричні розміри структурних елементів системи подачі нитки і знизити нерівномірність натягу для бавовняної пряжі 27,6 текс з 8,4 сН до 7,7 сН (зменшення натягу на 8,48%) та зменшити обривність нитки на 5%.

9. КОМП'ЮТЕРНИЙ ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ ВИЗНАЧЕННЯ ТЕХНОЛОГІЧНИХ ЗУСИЛЬ НА ОСНОВІ РЕКУРСІЇ

9.1. СТРУКТУРА КОМП'ЮТЕРНОЇ ПРОГРАМИ РЕАЛІЗАЦІЇ АЛГОРИТМУ РЕКУРСІЇ ДЛЯ ВИЗНАЧЕННЯ ТЕХНОЛОГІЧНИХ ЗУСИЛЬ

Актуальність. Різке збільшення натягу призводить до порушення нормального проходження технологічного процесу переробки нитки[3,4]. Недосконалість структурних елементів системи подачі, пристроїв для натягу нитки та спрямовувачів нитки на трикотажних, текстильних та швейних машинах не дозволяють здійснювати переробку ниток при стрибкому зростанні вхідного натягу[4]. Низька якість сировини, недосконалість лінії заправки технологічного устаткування при збільшенні натягу по глибині заправки приводить до обривів. Простої устаткування, пов'язані з ліквідацією обриву, складають в даний час 75-80% від загального часу простоїв[4, 7, 9]. Аналіз дефектів, що приводять до обриву нитки при переробці, показав, що найбільш вагомими є: слабкі місця на нитці (стоншування), шишки і потовщення, моховитість [4, 8-11]. Якщо виникнення перших двох дефектів пояснюється низькою якістю початкової сировини і порушенням технологічного процесу при виробництві, то останній, як наголошувалося вище, виникає безпосередньо при взаємодії ниток з робочими органами технологічного устаткування[4, 6-10]. Розробка нових схем системи подачі нитки потребує оперативної оцінки значення натягу перед роботою зоною. Для цього ефективно використовувати спеціально розроблені програми для виконання обчислювального експерименту[3,4].

Постановка завдання

На основі реалізації алгоритму визначення натягу нитки по зонах заправки на технологічному обладнанні, з використанням рекурсії, розробити комп'ютерну програму.

Основна частина

На рис.9.1.1 а представлена структурна схема заправки нитки на технологічному обладнанні. Практично для будь-якої технологічної машини (швейної, текстильної, трикотажної) схема заправки включає зону 0, в якій нитка сходить з бобіни, шпулі, ткацького навою, катушки основов'язальної машини. В зонах 1...N нитка взаємодіє з робочими елементами системи подачі. До робочих елементів відносяться різні спрямовувачі з отворами у формі тора[5,6,11], циліндричні спрямовувачі[5], пристрої для натягу нитки (шайбові, гребінчасті, пальцьові, оригінальної конструкції для конкретної технологічної машини[9]), компенсатори натягу нитки (з шайбами, з пальцями, гребінчастий, трубчатий[6], оригінальної конструкції для конкретної технологічної машини[10]). Зона ZP представляє собою робочу зону, де нитка переробляється (зона в'язання, зона формування тканини, зона зшивання матеріалів, зона формування ткацьких навоїв, зона формування бобін та ін.).

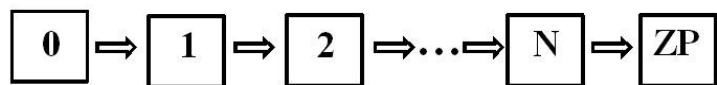


Рис.9.1.1. Структурна схема заправки нитки на технологічному обладнанні

Нормальне протікання технологічних процесів можливе у випадку, коли нитка перед зоною ZP буде мати мінімально необхідний натяг. При збільшенні натягу в системі подачі будуть виникати обриви, що призведе до зупинки технологічного обладнання та знизить його продуктивність. Крім того, розробка нових схем системи подачі нитки потребує

оперативної оцінки значення натягу перед роботою зоною ZP. Для цього ефективно використовувати спеціально розроблені програми для виконання обчислювального експерименту[3,4].

Характерною особливістю структурних схем заправки нитки на технологічному обладнанні є те, що вихідний натяг нитки на поточній зоні буде вхідним для наступної. Це дозволяє використовувати рекурсивних підхід для визначення натягу перед робочою зоною ZP. Лінію заправки необхідно розбити на зони, в кожній з котрих буде знаходитися один елементами системи подачі. Тоді система рівнянь, яка описує зміну натягу по зонам, буде мати вигляд

$$\begin{aligned}
 P_1 &= P_0 f_0(P_0), \\
 P_2 &= P_1 f_1(P_1), \\
 &\dots\dots\dots \\
 P_i &= P_{i-1} f_{i-1}(P_{i-1}), \\
 i &= 1, 2, \dots, n,
 \end{aligned}
 \tag{9.1.1}$$

де P_0 - натяг нитки в зоні входу в систему подачі на технологічній машині; $P_1, P_2 \dots P_i$ - натяг нитки при виході з відповідної зони; $f_0(P_0), f_1(P_1) \dots f_{i-1}(P_{i-1})$ - функції, які пов'язують натяг нитки до та після елемента системи подачі в кожній зоні; i - поточний номер зони; n - кількість елементів системи подачі ниток конкретної технологічної машини.

Використовуючи рекурсивний підхід для визначення натягу нитки перед роботою зоною ZP, при якому вихідний натяг нитки на поточній зоні буде вхідним для наступної (рис.1), представимо систему рівнянь (9.1.1) у вигляді

$$P_n = P_0 \prod_{i=0}^n f_i(P_i) .
 \tag{9.1.2}$$

На рис.9.1.2 представлена блок-схема комп'ютерної програми [1, 2] реалізації алгоритму рекурсії для визначення технологічних зусиль. Зупинимося на розшифруванні окремих блоків. Блок 1 відповідає початку виконання алгоритму. В блоці 2 задаються величини: $kivuz$ - кількість

вузлів; $v_{гух}$ - швидкість руху нитки; $d_{лінз}$ - довжина лінії заправки. В блоці 3 розраховується час руху нитки після сходу з бобіни до моменту входу в робочу зону за формулою

$$TRUXU = d_{лінз} / v_{гух}. \quad (9.1.3)$$

Блок 4 включає в себе три блока процедур: `procedure TForm2.N2Click`; `procedure TForm2.N3Click`; `procedure TForm2.N4Click`. Процедура `procedure TForm2.N2Click` забезпечує обрання виду сировини: процедура `procedure TForm2.N28Click` забезпечує введення даних для бавовняної пряжі; процедура `procedure TForm2.N29Click` забезпечує введення даних для вовняної пряжі; процедура `procedure TForm2.N36Click` забезпечує введення даних для шовкової нитки; процедура `procedure TForm2.N37Click` забезпечує введення даних для віскозної нитки; процедура `procedure TForm2.N38Click` забезпечує введення даних для капронової комплексної нитки; процедура `procedure TForm2.N39Click` забезпечує введення даних для лавсанової нитки; процедура `procedure TForm2.N40Click` забезпечує введення даних для льняної нитки; процедура `procedure TForm2.N30Click` забезпечує введення даних для капронової моноплетки. При обранні сировини задається текст нитки чи пряжі. На наступному етапі розраховується діаметр перетину.

Процедура `procedure TForm2.N3Click` забезпечує обрання діаметру сировини. В програмі закладена можливість використання трьох варіантів: діаметр нитки чи пряжі постійний (процедура `procedure TForm2.N31Click`); діаметр змінюється за періодичним законом (процедура `procedure TForm2.N12Click`); зміна діаметру сировини відбувається за законом, який запропонований користувачем (процедура `procedure TForm2.N14Click`). Для цього в програмі використовується модифікований транслятор на основі зворотних польських записів.

В блоці 5 обирається перший елемент системи подачі ниток конкретної технологічної машини. Блок 6 використовується для обрання матеріалу напрямної поверхні. Для цього реалізується процедура procedure TForm2.N43Click при обранні сталевий чи керамічної напрямної. В залежності від обрання матеріалу з бази застосовуються відповідні коефіцієнти тертя[7,8].

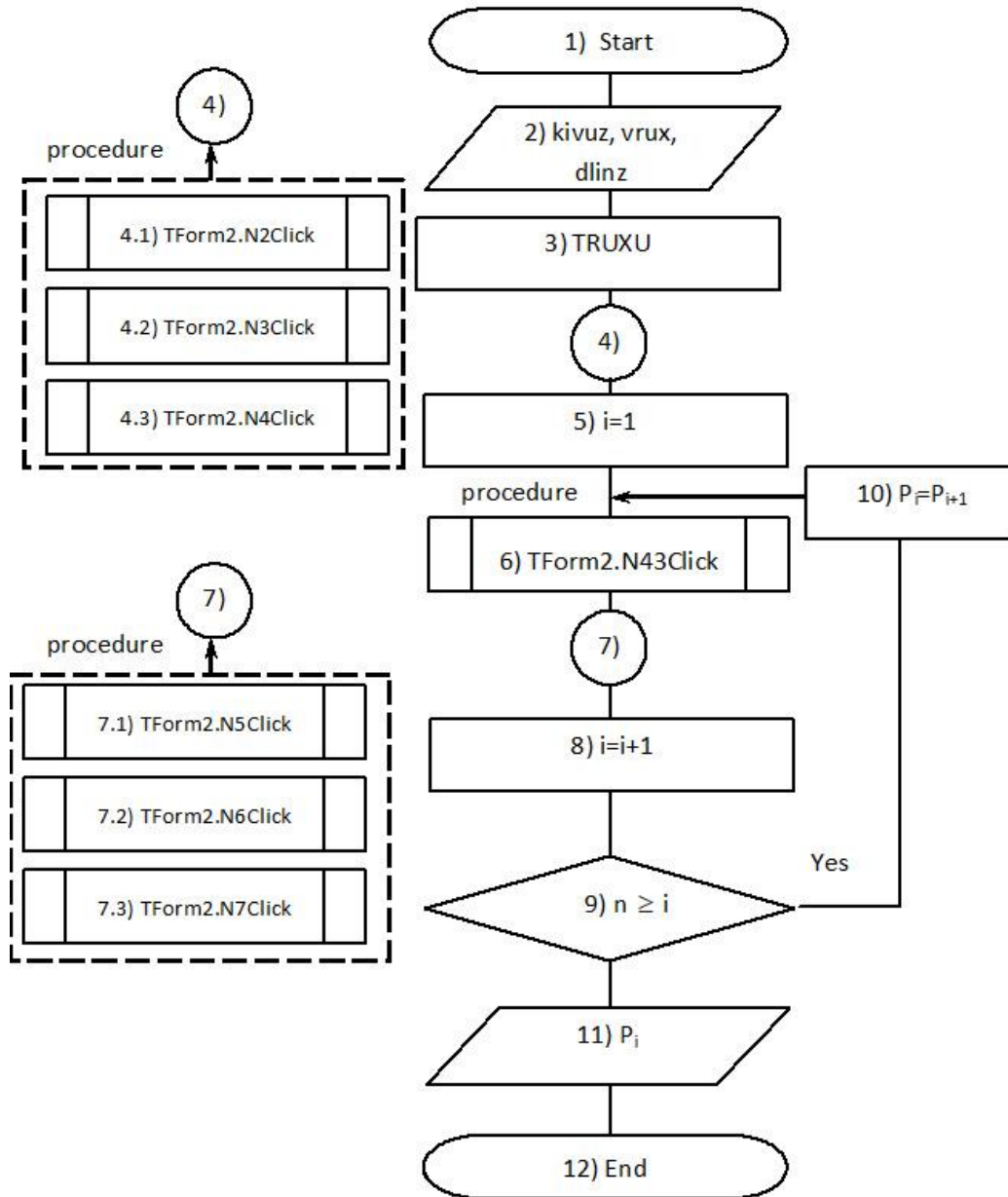


Рис.9.1.2. Блок-схема комп'ютерної реалізації алгоритму визначення натягу нитки з використанням рекурсії

Блок 7 процедур включає в себе три блока процедур: процедура `procedure TForm2.N5Click`, яка використовує пристрої для натягу нитки (шайбові – процедура `procedure TForm2.N19Click`, гребінчасті – процедура `procedure TForm2.N20Click`, пальцьові – процедура `procedure TForm2.N21Click`, оригінальної конструкції для конкретної технологічної машини – процедура `procedure TForm2.N45Click`); процедура `procedure TForm2.N6Click`, яка використовує компенсатори натягу нитки (з шайбами – процедура `procedure TForm2.N22Click`, з пальцями – процедура `procedure TForm2.N23Click`, гребінчастий – процедура `procedure TForm2.N24Click`, трубчатий – процедура `procedure TForm2.N25Click`, оригінальної конструкції для конкретної технологічної машини – процедура `procedure TForm2.N46Click`); процедура `procedure TForm2.N7Click`, яка використовує циліндричні спрямовувачі при спрямуванні нитки в робочу зону – процедура `procedure TForm2.N27Click` та спрямовувачі з отворами у формі тора – процедура `procedure TForm2.N26Click`.

В блоці 8 здійснюється перехід до наступного елементу системи подачі. Блок 9 використовується для перевірки номеру елементу системи подачі с заданим числом елементів системи подачі нити технологічної машини. У випадку виконання умови відбувається повернення до блоку 6. При цьому, в блоці 10 відбувається пере присвоювання, коли вихідний натяг нитки на поточній зоні буде вхідним для наступної. У випадку, коли умова не виконується відбувається перехід до блоку 11, де фіксується натяг нитки перед роботою зоною ZP. Блок 12 відповідає кінцю виконання алгоритму.

На рис.9.1.3 представлені перша та основна форма програми.

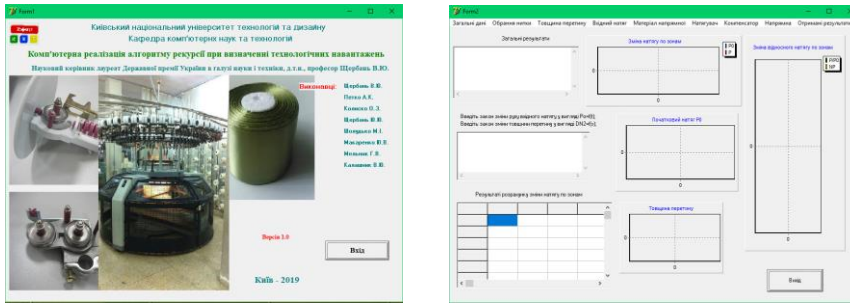


Рис.9.1.3. Перша та основна форма програми

На рис.9.1.4 представлені основні компоненти головного меню MainMenu1: TMainMenu програми. Графічна форма процедури procedure TForm2.N32Click загальних даних представлена на рис.9.1.4а. Графічна форма процедури procedure TForm2.N2Click обрання нитки представлена на рис.9.1.4б.

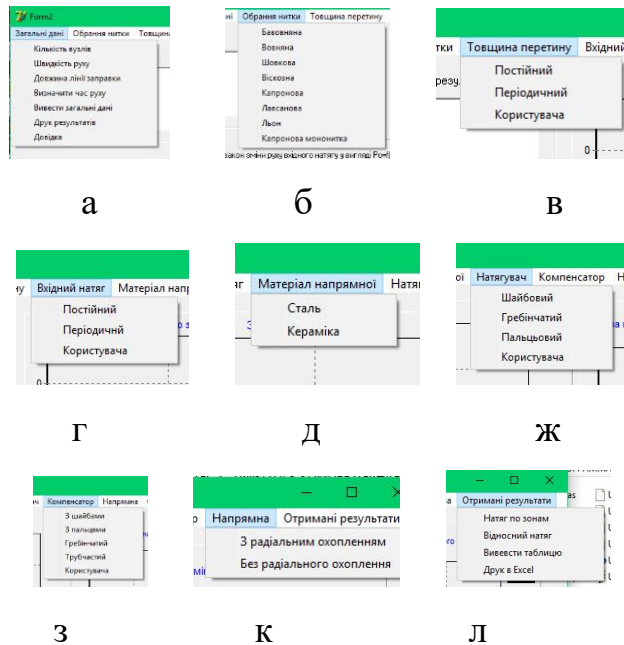


Рис.9.1.4. Основні компоненти головного меню MainMenu1: TMainMenu програми

Графічна форма процедури procedure TForm2.N3Click товщини перетину представлена на рис.9.1.4в. Графічна форма процедури procedure TForm2.N4Click вхідного натягу представлена на рис.9.4г. Графічна форма процедури procedure TForm2.N42Click матеріалу напрямної представлена на рис.9.1.4д. Графічна форма процедури

procedure TForm2.N5Click обрання пристрою для натягу нитки представлена на рис.9.1.4ж. Графічна форма процедури procedure TForm2.N6Click компенсатора натягу нитки представлена на рис.4з. Графічна форма процедури procedure TForm2.N7Click напрямної для нитки представлена на рис.4к. Графічна форма процедури procedure TForm2.N8Click розрахунку та представлення результату представлена на рис.9.1.4л.

На рис.9.1.5 представлені результати розрахунку натягу нитки по зонах для 4 елементів системи подачі нити технологічної машини. На графічних діаграмах представлені відносний натяг по зонах, що дозволяє оцінювати напруженість процесу взаємодії нитки з елементами системи подачі та вносити відповідні корективи в параметри їх заправки.

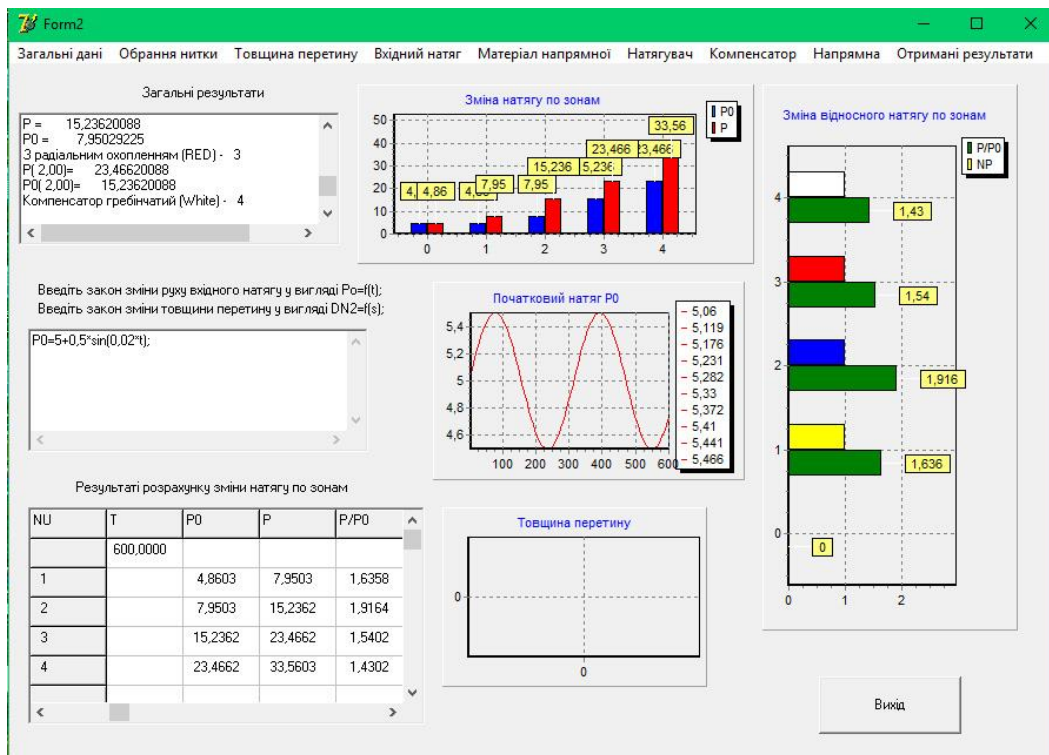


Рис.9.1.5. Результати розрахунку натягу нитки по зонах для 4 елементів системи подачі нити технологічної машини

9.2. ПРОГРАМНІ МОДУЛІ КОМП'ЮТЕРНОЇ ПРОГРАМИ РЕАЛІЗАЦІЇ АЛГОРИТМУ РЕКУРСІЇ ДЛЯ ВИПАДКУ ЗМІННОГО ВХІДНОГО НАТЯГУ

Актуальність. Основним параметром оптимізації системи подачі ниток на технологічному обладнанні текстильної та трикотажної промисловості є мінімально необхідний натяг в робочій зоні[3-8]. Збільшення натягу в робочій зоні призводить до обриву ниток і, як наслідок, до зупинки технологічного обладнання[3, 4, 9-12]. Простої устаткування, пов'язані з ліквідацією обриву, складають в даний час 75-80% від загального часу простоїв[4]. Це негативно впливає на продуктивність обладнання, зменшує якість продукції що випускається[6-8, 10]. Зменшення натягу по відношенню до мінімально необхідного також негативно впливає на нормальне протікання технологічного процесу[11]. Це пов'язано з порушенням технологічних умов взаємодії нитки з робочими органами в зоні формування тканини чи трикотажного полотна.

Натяг нитки збільшується при переході по зонах заправки системи подачі нитки на технологічному обладнанні[3, 4]. Це збільшення обумовлено взаємодією нитки з напрямними та пристроями для натягу. Максимального значення натягу буде перед робочою зоною. Характерною особливістю є зміна вхідного натягу при змотуванні нитки з бобіни[4, 5]. Визначення закону зміни натягу на вході в систему подачі нитки на технологічному обладнанні дозволить визначати натяг в робочій зоні з урахуванням реальних умов змотування.

Розробка спеціальних комп'ютерних програм [1-2] для визначення натягу в робочій зоні дозволяє оперативно визначати необхідні технологічні параметри, провадити корегування, як самої структури так і

складових компонентів системи подачі нитки для отримання мінімально необхідного натягу в робочій зоні.

Постановка завдання

На основі реалізації алгоритму рекурсії розробити модулі комп'ютерної програми для визначення натягу нитки в робочій зоні у випадку, коли закон зміни вхідного натягу представлений у вигляді гармонічної функції або у вигляді довільній функції користувача з використанням зворотних польських записів.

Основна частина

Розробку програмних модулів комп'ютерної програми реалізації алгоритму рекурсії визначення натягу в робочій зоні для випадку змінного вхідного натягу почнемо з формалізації законів їх зміни. При проведенні технологічних вимірювань натягу одиночної нитки використовувався електронний тензометричний прилад. В експерименті використовувалася одна випадково обрана бобіна бавовняної пряжі 32 текс (швидкість змотування - 5 мм / с). Виміри проводилися при змотуванні нитки з трьох діаметрів: максимального $130 \leq D \leq 150$, середнього $110 \leq D \leq 130$, мінімального $D \leq 110$.

Для кожного діаметру виміри проводилися 10 разів. В результаті були отримані експериментальні значення натягу одиночної нитки, які наведені в таблиці 1.

Таблиця 9.2.1

Виміри натягу одиночної нитки

№ п/п	Максимальний діаметр		Середній діаметр		Мінімальний діаметр	
	мм	сН	мм	сН	мм	сН
0	150	9.63	120	14.39	105	17.38
1	145	8.73	120	11.56	100	13.32

2	150	10.20	120	12.05	100	11.31
3	150	9.75	125	13.69	95	14.10
4	150	10.04	125	13.48	100	13.53
5	150	10.82	125	10.00	102	14.26
6	145	16.97	125	13.36	95	14.82
7	145	17.63	128	11.56	100	14.67
8	135	15.94	125	13.98	100	13.25
9	135	11.93	128	9.18	102	14.18

При визначенні натягу окремої нитки допускаємо, що існує функціональна залежність між величинами y (натяг при різних діаметрах бобіни) та x (діаметр бобіни). При цьому, функція $y = f(x)$ невідома, але на основі експериментальних досліджень встановлені практичні дані (таблиця 9.2.1). Задача полягає в тому, щоб: знайти функцію, по можливості більш простішу з обчислювальної точки зору, яка представляла невідому функцію $y = f(x)$ точно або наближено; визначити проміжні значення функції $y = f(x_i^*)$, де $x_i < x_i^* < x_{i+1}$.

Оскільки різниця між суміжними значеннями аргументу різна, то поставлена задача інтерполяції буде зведена до знаходження багаточлену $P(x)$ ступеню $\leq n$ за допомогою інтерполяційних формул Лагранжа та Ньютона (метод розподілених різностей).

Інтерполяційна формула Ньютона для усіх вузлів інтерполяції

$$P(x) = y_0 + (x - x_0)f(x_0, x_1) + (x - x_0)(x - x_1)f(x_0, x_1, x_2) + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})f(x_0, x_1, \dots, x_n). \quad (9.2.1)$$

Коли два чи більше аргументів співпадають, то поняття розподіленої різності зберігає свій зміст

$$f(x_0, x_1) = \lim_{x_1 \rightarrow x_0} \frac{f(x_0) - f(x_1)}{x_0 - x_1} = f'(x_0), \quad (9.2.2)$$

$$f(x_0, x_1, \dots, x_n) = \frac{1}{r!} \frac{d^r}{dx^r} f(x_0),$$

$$\begin{aligned} f(x, x_0, x_1, \dots, x_n) &= \lim_{x' \rightarrow x_0} \frac{f(x', x_0, x_1, \dots, x_n) - f(x, x_0, x_1, \dots, x_n)}{x' - x} = \\ &= \frac{d}{dx} f(x, x_0, x_1, \dots, x_n). \end{aligned} \quad (9.2.3)$$

В вузлах інтерполяції $P(x_i) = f(x_i) = y_i (i=1, 2, \dots, n)$ тому, що остаточно́ний член в них $R(x) = (x - x_0)(x - x_1) \dots (x - x_n) f(x, x_0, x_1, \dots, x_n) = 0$. Остаточно́ний член $R(x)$ дає можливість оцінити похибку на базі

$$\begin{aligned} R &= f(x, x_0, \dots, x_n) \prod_{i=0}^n (x - x_i), \\ \varepsilon &= |P_n(x) - f(x)| \leq f', \\ f' &= |(x - x_0)(x - x_1) \dots (x - x_n)| \frac{|f^{(n+1)}(\xi)|}{(n+1)!}, \end{aligned}$$

де $\frac{|f^{(n+1)}(\xi)|}{(n+1)!}$ - це $(n+1)$ похідна функції $f(x)$ в точці ξ з найменшого проміжку,

який зберігає усі точки $x_0, x_1, x_2, \dots, x_n$.

На рис.9.2.1 представлений графік поліноміальної функціональної залежності та експериментальні дані натягу в залежності від діаметру.

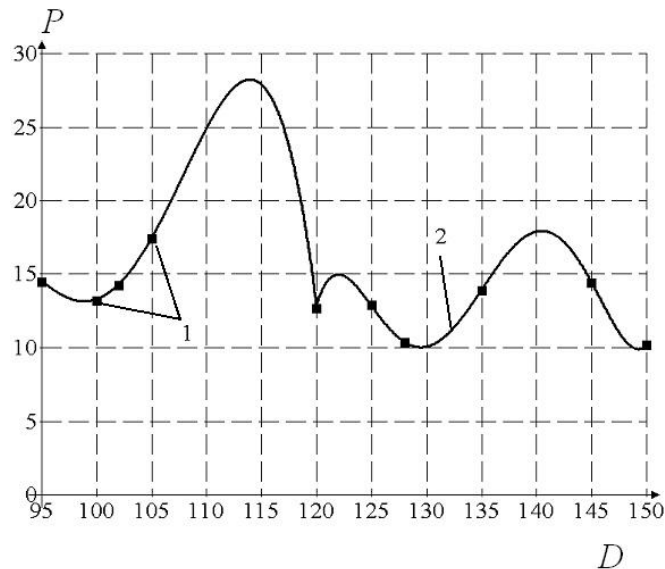


Рис.9.2.1. Графік поліноміальної функціональної залежності натягу в залежності від діаметру: 1 – експериментальні дані; 2 – графік функції $P(D)$

Аналіз графічної залежності показує, що для її апроксимації можна використовувати тригонометричні або алгебраїчні поліноми.

При розробці програмних модулів використовувалися закон зміни вхідного натягу представлений у вигляді гармонічної функції або у вигляді довільній функції користувача з використанням зворотних польських записів. На рис.9.2.2 представлені форми програмних модулів: 2а – закон зміни натягу у вигляді гармонічної функції; 2б - закон зміни натягу у вигляді довільній функції користувача з використанням зворотних польських записів.

а

б

Рис.9.2.2. Форми програмних модулів

Для завдання закону зміни вхідного натягу $P_0 = P_0(t)$ був використаний модуль unit Unit23. Для закону зміни вхідного натягу $P_0 = P_0(t)$ у вигляді довільній функції користувача був використаний

модуль Synt на основі розробленого транслятора з використанням зворотних польських записів (рис.3).

```

unit Synt;
interface
uses classes;
type
  TData = record
    Name: string;
    Data: real;
  end;
var
  NConst: integer = 100;
  ErrorList: TStringList;
  PZ: array of integer;
  DataList: array of TData;
const
  MConst = 2;
procedure SyntItem(S:string; First:boolean=false; Pos:integer=1);
function CreatePZ(S:string):boolean;
function Calculate(var R:real):boolean;
function SetData(Name:string; Data:real):boolean;
function GetData(Name:string; var Data:real):boolean;
implementation
uses Sysutils, Math, Dialogs, Unit2;
type
  TType = (None, Number, Divider, Ident, Func, Part, All);
  TSynt = record
    mode: TType;
    Number: real;
    Ident: string;
    Error: boolean;
    Pos1, Pos2: integer;
  end;
const
  SetNum: set of char = ['0'..'9', '.', '^'];
  SetDiv: set of char = ['+', '-', '/', '*', '^'];
  
```

Рис.9.2.3. Комп'ютерний модуль транслятора на основі зворотних польських записів

Натяг нитки перед роботою зоною, з використанням рекурсивного підходу, буде мати вигляд

$$P_n = P_0(t) \prod_{i=0}^n f_i(P_i), \quad (9.2.4)$$

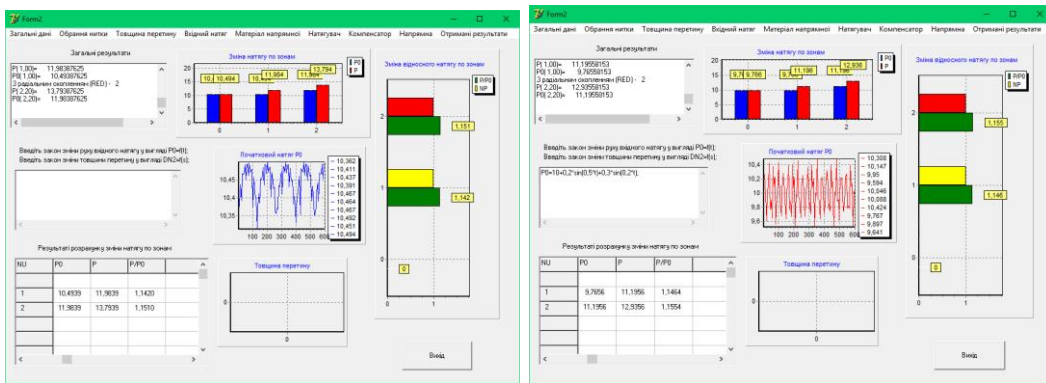
де $P_0(t)$ - закон зміни вхідного натягу нитки в зоні входу в систему подачі на технологічній машині; $f_i(P_i)$ - функції, які пов'язують натяг нитки до та після елемента системи подачі в кожній зоні; i - поточний номер зони; n - кількість елементів системи подачі ниток конкретної технологічної машини.

На рис.9.2.4 представлена основна форма програми з результатами (а – закон зміни натягу у вигляді гармонічної функції; б - закон зміни натягу у вигляді довільній функції користувача з використанням зворотних польських записів). При розрахунках приймали: кількість вузлів 2; швидкість руху нитки 5 мм/с; довжина лінії заправки 3 м.

Використовувалася бавовняна нитка 32 текс. Для періодичного закону були прийняті наступні константи $b_0=10$, $b_1=0.2$, $b_3=0.5$, $b_4=0.3$, $b_5=0.1$, $b_6=0.2$. При $t=300$ с початковий натяг $P_0=10.49$ сН. В якості напрямних в вузлах використовувалися напрямна без радіального охоплення ($\varphi=1.57$ рад., $R=1$ мм) та напрямна з радіальним охопленням ($\varphi=1.4$ рад., $R=2.2$ мм, $\beta=1.3$ рад.). В якості довільної функції користувача була прийнята функція

$$P_0 = 10 + 0.25\sin(0.5t) + 0.3\sin(0.2t) \cdot$$

Розрахунки за останньою формулою показали, що при $t=300$ с початковий натяг $P_0=9.77$ сН.



а

б

Рис.4. Основна форма програми з результатами: а – закон зміни натягу у вигляді гармонічної функції; б - закон зміни натягу у вигляді довільній функції користувача з використанням зворотних польських записів

9.3. КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ АЛГОРИТМУ РЕКУРСІЇ ДЛЯ ВИПАДКУ ЗМІННОГО ДІАМЕТРУ СИРОВИНИ

Актуальність. Головним параметром оптимізації системи подачі ниток на є мінімально необхідний натяг в робочій зоні. Обривність ниток при переробці на технологічному устаткуванні пов'язана з недосконалістю системи подачі нитки і її складових елементів[3]. Простої устаткування,

пов'язані з ліквідацією обриву, складають в даний час 75-80% від загального часу простоїв. Це негативно впливає на продуктивність обладнання, зменшує якість продукції що випускається. Найбільше число обривів випадає на зону розташування пристроїв натягу нитки[8-12]. Дослідження впливу конструкції пристрою натягу нитки на умови його взаємодії з ниткою з урахуванням її нерівномірності по діаметру поперечного перетину мають важливе значення при визначенні натягу нитки. Натяг нитки збільшується при переході по зонах заправки системи подачі нитки на технологічному обладнанні. Це збільшення обумовлено взаємодією нитки з напрямними та пристроями для натягу[4-9]. Максимального значення натягу буде перед робочою зоною. Мінімізація натягу перед робочою зоною має важливе значення для удосконалення технологічних процесів текстильної та швейної промисловості з позиції підвищення продуктивності технологічного устаткування та якості продукції що випускається. Таким чином, тема даної статті є актуальною, яка має важливе значення для удосконалення системи подачі ниток на технологічному обладнанні, конструкції існуючих пристроїв натягу нитки та розробки нових[10-12].

Розробка спеціальних комп'ютерних програм [1-2] для визначення натягу в робочій зоні з урахуванням її нерівномірності по діаметру поперечного перетину дозволяє оперативно визначати необхідні технологічні параметри, провадити корегування, як самої структури так і складових компонентів системи подачі нитки для отримання мінімально необхідного натягу в робочій зоні[3, 4-9].

Об'єкти і методи дослідження. Складовою частиною системи подачі нитки технологічного устаткування текстильної промисловості є пристрої натягу нитки. Недосконалість конструкції пристроїв натягу нитки призводить до коливання натягу нитки в процесі її переробки на

технологічному устаткуванні та порушенню технологічного режиму. В першу чергу, на це впливає нерівномірність нитки по діаметру, що зумовлено її структурою і матеріалом, специфікою виготовлення нитки, довжиною окремих елементарних волокон та їх взаємним розташуванням відносно один до одного. Теоретичною основою при вирішенні науково-технічної проблеми є праці провідних вчених в галузях технології текстильного та трикотажного виробництв, текстильного матеріалознавства, механіки нитки, теорії пружності, математичного моделювання. У теоретичних та експериментальних дослідженнях використано методи теоретичної механіки, опору матеріалів, планування експерименту та статистичної обробки результатів досліджень.

Постановка завдання

На основі реалізації алгоритму рекурсії розробити модулі комп'ютерної програми для визначення натягу нитки в робочій зоні у випадку нерівномірності по діаметру її поперечного перетину, яка може бути представлена у вигляді гармонічної функції або у вигляді довільній функції користувача з використанням зворотних польських записів.

Основна частина

На рис.1 представлена залежність поперечного перетину вовняної пряжі 29,9 текс довжиною 1 м. Аналіз показує, що апроксимувати нерівномірність поперечного перетину можна за допомогою наступного поліному [3]

$$d_H = 2r_H = \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t) , \quad (9.3.1)$$

де a_i, b_i - амплітуди зміни i -ї гармоніки поперечного перетину; ω_i - частота зміни поперечного перетину i -ї гармоніки ; n - число гармонік; t - час.

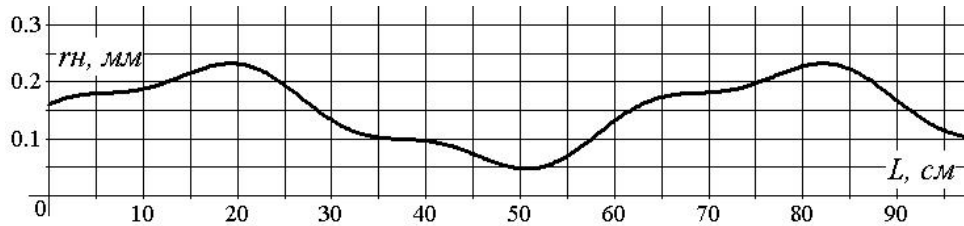


Рис.9.3.1. Залежність поперечного перетину вовняної пряжі 29,9 текс від довжини

Натяг нитки перед роботою зоною, з використанням рекурсивного підходу, буде мати вигляд

$$P_n = P_0(t) \prod_{i=0}^n f_i(P_i), \quad (9.3.2)$$

де $P_0(t)$ - закон зміни вхідного натягу нитки в зоні входу в систему подачі на технологічній машині; $f_i(P_i)$ - функції, які пов'язують натяг нитки до та після елемента системи подачі в кожній зоні; i - поточний номер зони; n - кількість елементів системи подачі ниток конкретної технологічної машини.

Для кожного структурного елемента натяг, з урахуванням залежності (1), буде визначатися за формулою

$$P_{i+1} = P_i \left[1 + \frac{(R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t))}{[R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t) (1 - \delta_{0j})]} (e^{\frac{\beta_j a}{\sin \beta_j P_i^b R_j^b \phi_j} - 1}) + \left[\frac{B}{2[R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t) (1 - \delta_j)]^2} \right] - \left[\frac{B}{2[R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t) (1 - \delta_{0j})]^2} \right] \times \left[1 + \frac{(R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t))}{[R_j + 0.5 \sum_{i=1}^n (a_i \sin \omega_i t + b_i \cos \omega_i t) (1 - \delta_{0j})]} (e^{\frac{\beta_j a}{\sin \beta_j P_i^b R_{ms(j)}^b \phi_j} - 1}) \right], \quad (9.3.3)$$

де P_{i+1} - натяг нитки після j конструктивного елемента; P_i - натяг нитки до j конструктивного елемента; R_j - радіус кривизни поверхні j конструктивного елемента; δ_{0j} - початкова деформація перетину нитки при набіганні на j конструктивний елемент; δ_j - кінцева деформація перетину нитки при збіганні з j конструктивного елемента; β_j - кут радіального

охоплення нитки поверхнею j конструктивного елементу; φ_j – реальний кут охоплення ниткою j конструктивного елементу.

Сумісне рішення системи рівнянь (9.3.1) - (9.3.3) дозволяє визначити значення натягу нитки в робочій зоні. На рис.9.3.2 представлені комп'ютерні програмні модулі для визначення натягу. Модуль unit Unit4(рис.9.3.2а) призначений для обрання сировини нитки чи пряжі. Активізуючи кнопку (Button1: TButton) модуля unit Unit4(рис.9.3.2б) здійснюємо розрахунок параметрів нитки. Процедура (procedure TForm2.N12Click(Sender: TObject)) модуля unit Unit4(рис.9.3.2в) призначена для обрання періодичного закону зміни діаметру поперечного перетину. Процедура (procedure TForm2.N14Click(Sender: TObject)) модуля unit Unit4(рис.9.3.2г) призначена для обрання періодичного закону зміни діаметру поперечного перетину.

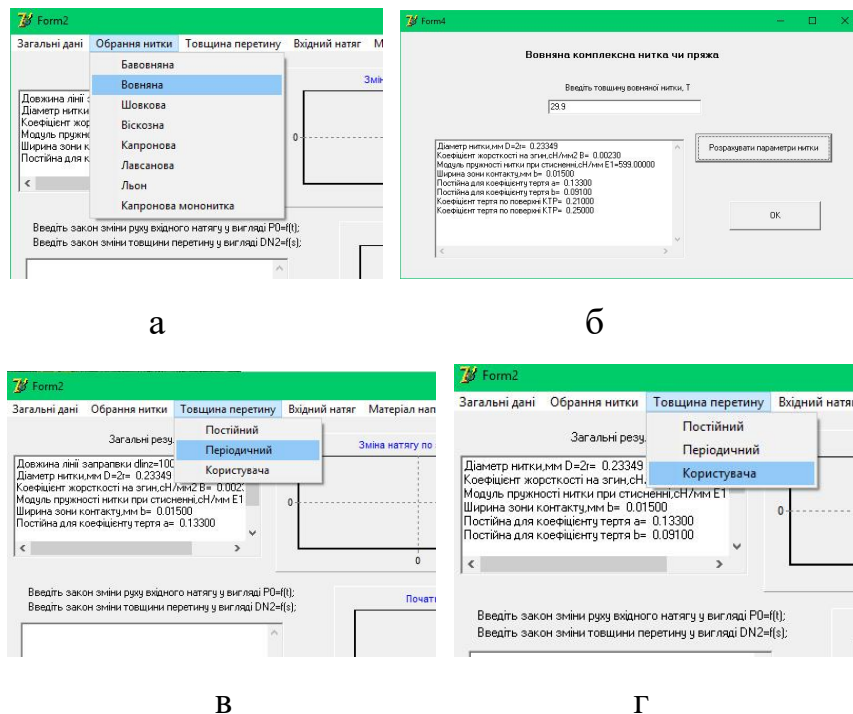


Рис.9.3.2. Комп'ютерні модулі програми для визначення натягу

На рис.3а представлений модуль Unit22 для реалізації закону зміни діаметру перетину нитки у вигляді гармонічної функції. Коефіцієнти апроксимації обираються на основі експериментальних досліджень. На

рис.3б представлена форма TForm2 = class(TForm) модуля Unit2 за допомогою якого можна задавати закон зміни діаметру перетину нитки у вигляді довільній функції користувача. Для цього в програмі використовується транслятор у вигляді модуля unit Synt з використанням зворотних польських записів .

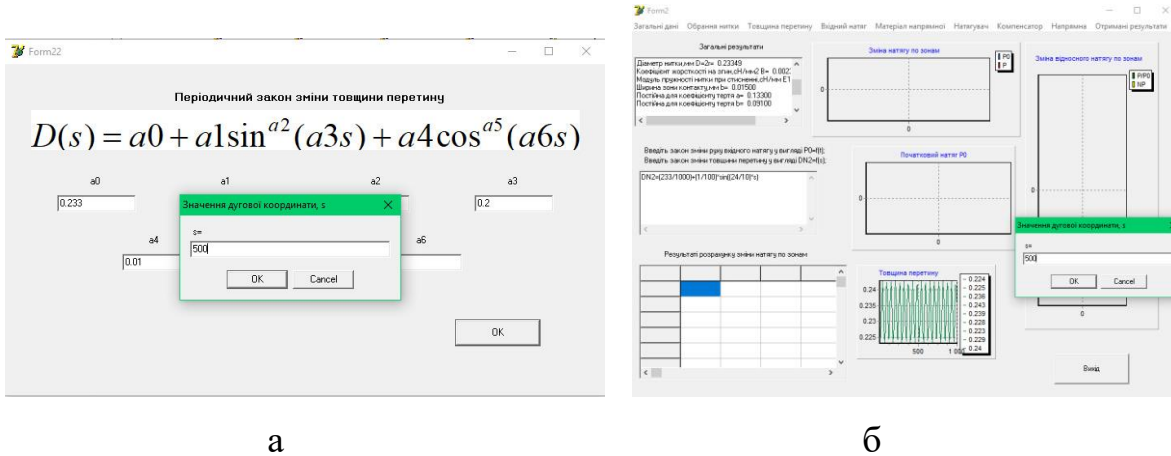


Рис.9.3.3. Форми модулів Unit22 та Unit2

Результати розрахунку значення натягу нитки в робочі зони технологічної машини для випадку двох вузлів представлені на рис.4.

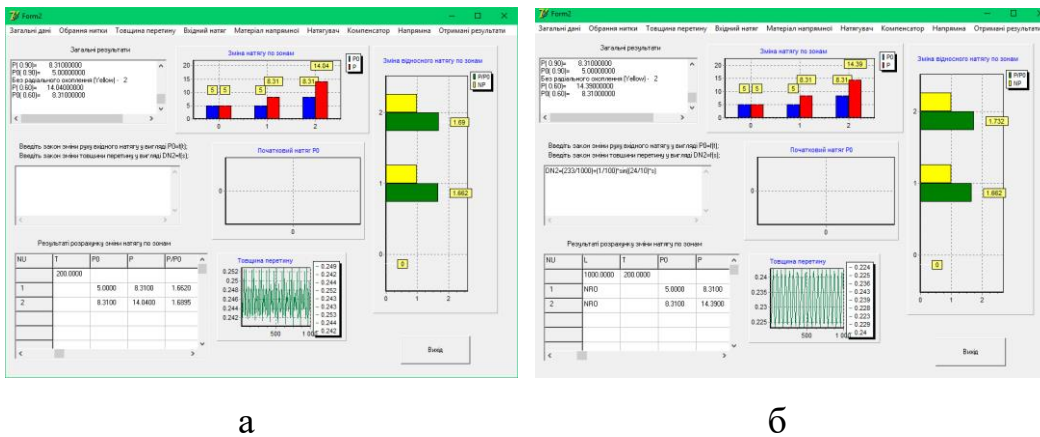


Рис.9.3.4. Результати розрахунку натягу для двох вузлів на технологічній машині

Випадок реалізації закону зміни діаметру перетину нитки у вигляді гармонічної функції представлений на рис.9.3.4а. Випадок реалізації закону зміни діаметру перетину нитки у вигляді довільній функції на рис.9.3.4б.

Висновки по розділу

1. На основі реалізації алгоритму визначення натягу нитки по зонах заправки на технологічному обладнанні, з використанням рекурсії, розроблена комп'ютерна програма, за допомогою якої, з урахуванням реальних фізико-механічних властивостей сировини, можна визначати натяг перед роботою зоною та оцінювати напруженість процесу по зонах заправки.
2. На основі реалізації алгоритму рекурсії розроблені модулі комп'ютерної програми для визначення натягу нитки в робочій зоні у випадку, коли закон зміни вхідного натягу представлений у вигляді гармонічної функції або у вигляді довільній функції користувача з використанням зворотних польських записів.
3. На основі реалізації алгоритму рекурсії розроблені модулі комп'ютерної програми для визначення натягу нитки в робочій зоні у випадку, коли діаметр нитки має змінне значення з використанням зворотних польських записів. Наведена структура комп'ютерної програми реалізації алгоритму рекурсії для визначення технологічних зусиль в робочій зоні текстильного обладнання для випадку, коли діаметр нитки має змінне значення. Представлені програмні модулі програми для визначення технологічних зусиль в робочій зоні на текстильних машинах з урахуванням зміни діаметру перетину нитки для широкого спектру комплексних ниток та пряжі. Програмними модулями передбачена можливість завдання закону зміни діаметру перетину нитки у вигляді гармонічної функції або довільній функції користувача з використанням зворотних польських записів у вигляді транслятора.

10. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ ВЗАЄМОДІЇ НИТКИ З НАПРАВЛЯЮЧИМИ ТРИКОТАЖНИХ МАШИН З УРАХУВАННЯМ ЇХ ФОРМИ ТА КОНСТРУКТИВНИХ ПАРАМЕТРІВ

Удосконалення системи подачі нитки на круглов'язальних машинах великого діаметру повинне базуватися на теоретичному і експериментальному дослідженні взаємодії ниток з циліндричними напрямними поверхнями великої та малої кривизни, спрямовувачами нитки у формі тора, пристроями та компенсаторами для натягу та його стабілізації в процесі переробки..

Проведення теоретичних досліджень повинне сприяти широкому освітленню процесів, які відбуваються в зоні контакту нитки з напрямними поверхнями великої та малої кривизни, спрямовувачами нитки у формі тора, пристроями та компенсаторами для натягу та його стабілізації. Головним чинником, який впливає на збільшення натягу є кут охоплення напрямної. В реальних умовах фактичне значення кута охоплення ниткою напрямних буде залежить від жорсткості нитки на згин та ступеню її зминання в зоні контакту [1]. При збільшенні жорсткості нитки на згин реальний кут охоплення буде меншим ніж розрахунковий, який визначається без урахування фізико-механічних характеристик нитки. При збільшенні ступеня зминання в зоні контакту реальний кут охоплення буде більшим ніж розрахунковий [2, 3]. Реальним пряжі та комплексним ниткам, які переробляються на круглов'язальних машинах великого діаметру, притаманні обидві ці властивості.

Величина жорсткості на згин та зминання в зоні контакту з напрямною залежить від величини крутіння нитки. Нитки та пряжа високого крутіння мають велику жорсткість на згин та малу ступінь деформації в зоні контакту та, навпаки, нитки та пряжа низького крутіння

мають низьку жорсткість на згин та велику ступінь деформації в зоні контакту.

Важливим фактором є форма та розміри площини контакту нитки з напрямними поверхнями великої та малої кривизни, спрямовувачами нитки у формі тора. Для циліндричних напрямних вона буде мати форму зігнутої площини. Для спрямовувачами нитки у формі тора вона буде мати складну просторову форму, що пов'язано з радіальним охопленням нитки поверхнею напрямної. Розміри площини контакту нитки з напрямними поверхнями та наявність радіального охоплення впливають на величину коефіцієнту тертя. При збільшенні площини контакту та кута радіального охоплення він буде зростати [1, 2].

Нерівномірність геометричних розмірів перетину нитки по довжині та вхідного натягу за часом суттєво впливають на значення натягу нитки після проходження напрямної [4-8]. При цьому, важливе значення має швидкість руху нитки. При збільшенні швидкості руху зростають значення динамічних навантажень в зоні взаємодії нитки з циліндричними напрямними поверхнями великої та малої кривизни, спрямовувачами нитки у формі тора.

Розв'язання цих питань потребує розробки основ статички, кінематики та динаміки процесу взаємодії нитки з направляючою у формі тора при урахуванні деформації перетину в зоні контакту, жорсткості на згин та зміни коефіцієнту тертя при радіальному охопленні нитки поверхнею напрямної.

10.1. ГЕОМЕТРИЧНІ, КІНЕМАТИЧНІ ТА ДИНАМІЧНІ КОМПОНЕНТИ ТОЧОК НИТКИ ПРИ ЇЇ ВЗАЄМОДІЇ З НАПРЯМНОЮ ВЕЛИКОЇ КРИВИЗНИ

Напруга, що виникає в довільному перетині нитки, можна привести до головного вектора \vec{P}_0 і головному моменту \vec{M}_0 , які прикладені в центрі

тяжіння перетину. В процесі переробки на круглов'язальних машинах великого діаметру, при взаємодії з направляючими великої кривизни у формі тора, відбувається зміна форми осі та перетину нитки [1, 2], що приводить до зміни кривизни і кручення останньої. При зминанні в зоні контакту відбувається деформація поперечного перетину ниток, що викликає зсув центру тяжіння перетину. В зоні контакту з направляючою великої кривизни у формі тора відбувається зміна положення осі нитки по відношенню до лінії, що визначає форму осі абсолютно жорсткої на зминання нитки. Таке переміщення приводить до зростання або зменшення радіусу кривизни осі нитки в зоні контакту з направляючою у формі тора залежно від її фізико-механічних характеристик. При складанні диференціальних рівнянь рівноваги нескінченно малого елемента нитки що зминається на направляючій поверхні у формі тора, необхідно знати закон зміни кривизни осі нитки залежно від деформації перетину в отворі напрямної.

На рис.10.1а представлена розрахункова схема деформації у зоні контакту з направляючою у формі тора.

Для опису форми осі деформованої в перетині нитки з направляючою у формі тора необхідно ввести три координатні системи [1]. Дві системи рухомі: натурального тригранника $\tau_*v_*\beta_*$ і головного тригранника $\tau_*n_*b_*$. Нерухома координатна система O_1XYZ утворюється перетином осей X , Y , Z , напрям яких визначається одиничними ортами i, j, k [1]. Дотична вісь τ_* натурального тригранника отримується шляхом перетину випрямляючої і дотичної площин [1]. Нормальна вісь v_* прямує до центру кривизни і отримується шляхом перетину нормальної і дотичної площин. Бінормаль β_* отримується шляхом перетину нормальної і випрямляючої площин. Нормальна вісь n_* головного тригранника $\tau_*n_*b_*$ прямує по

нормалі до направляючої поверхні в даній точці [1, 2]. У загальному випадку нормаль n_* головного тригранника і нормаль v_* натурального тригранника не співпадають.

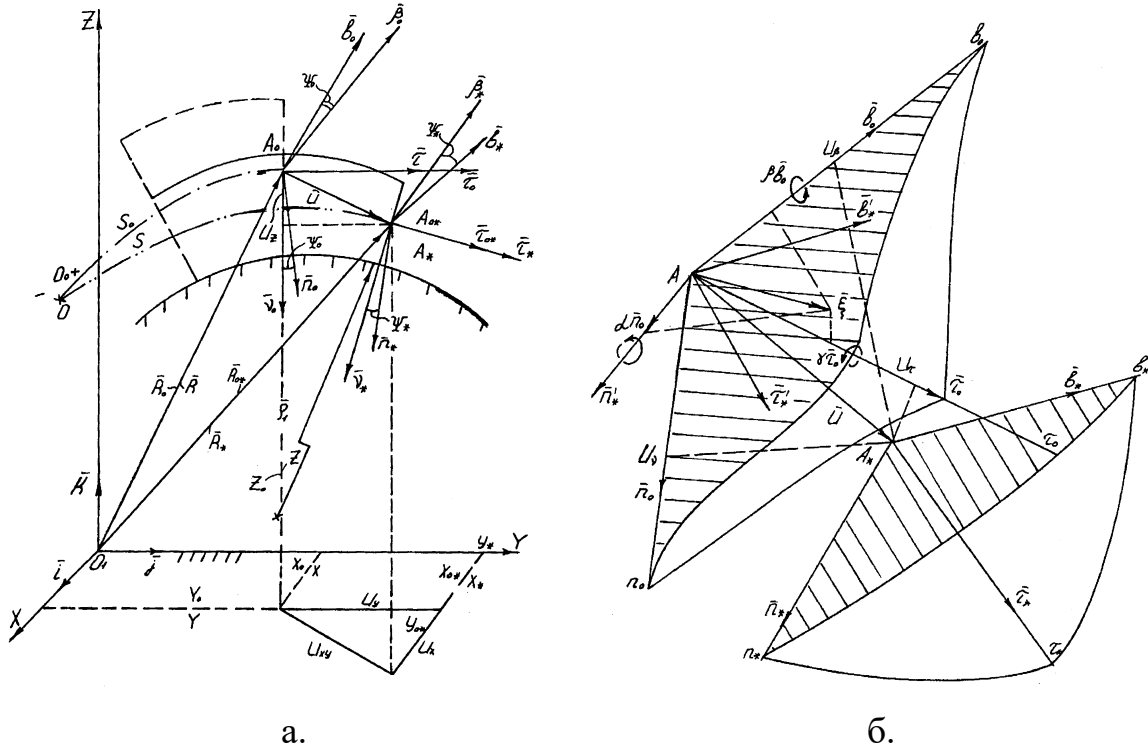


Рис.10.1. Схеми для визначення геометричних характеристик форми осі нитки при взаємодії з напрямними великої кривизни у формі тору та співвідношення між вектором переміщення та вектором обертунатурального тригранника

Радіус-вектор R_0 визначає положення точки A_0 на S_0 - лінії щодо координатної системи O_1XYZ [1]. Радіус-вектор R визначає у випадку радіального охоплення положення точки A_0 на S - лінії щодо нерухомої координатної системи. В результаті при зминанні нитки точка $A(A_0)$ переміститься в нове положення. Новий радіус-вектор R_0^* визначить положення точки A_0^* на S_0 - лінії, а радіус-вектор R_* - положення точки A_0^* на S - лінії. Вектор між точками A і A_* позначимо U_{Δ}^{β} . Цей вектор

характеризує зсув точок за рахунок деформації перетину нитки у випадку радіального охоплення .

Векторні вирази для визначення положення точок для направляючій у формі тора A_0^* та A_* в фізичних координатах S_0, t і A_* в геометричних координатах S, t матимуть вигляд

$$\vec{R}_{0*} = \vec{R}_0 + \vec{U}_\Delta^\beta, \vec{R}_* = \vec{R}_* + \vec{U}_\Delta^\beta. \quad (10.1)$$

Диференціюючи рівняння (10.1) по відповідних дугових координатах, отримаємо встановити залежність між одиничними ортами в різних системах координатних осей

$$\vec{\tau}_{0*} = \vec{\tau}_0 + \frac{\partial \vec{U}_\Delta^\beta}{\partial S_0}, \vec{\tau}_* = \vec{\tau} + \frac{\partial \vec{U}_\Delta^\beta}{\partial S}. \quad (10.2)$$

Для визначення залежності, що визначає одиничний орт головної нормалі залежно від вибору дугових координат у випадку радіального охоплення, диференціюючи рівняння (10.2) ще раз по відповідних фізичних S_0 і геометричних координатах S , отримаємо

$$\vec{\nu}_* = (1 + \varepsilon) \vec{\nu}_0 \frac{\rho_*}{\rho_{0*}} + (1 + \varepsilon) \rho_* \frac{\partial^2 \vec{U}_\Delta^\beta}{\partial S_0^2} + \frac{\partial \varepsilon}{\partial S_0} \rho_* \vec{\tau}_{0*}. \quad (10.3)$$

Для визначення одиничного орта бінормалі, у випадку радіального охоплення, скористуємося векторним рівнянням

$$\vec{\beta}_* = (1 + \varepsilon)^2 \left[\vec{\beta}_0 \frac{\rho_*}{\rho_{0*}} + \frac{\partial \vec{U}_\Delta^\beta}{\partial S_0} \times \vec{\nu}_0 \frac{\rho_*}{\rho_{0*}} + \vec{\tau}_0 \times \frac{\partial^2 \vec{U}_\Delta^\beta}{\partial S_0^2} \rho_* + \frac{\partial \vec{U}_\Delta^\beta}{\partial S_0} \times \frac{\partial^2 \vec{U}_\Delta^\beta}{\partial S_0^2} \rho_* \right].$$

Перейдемо до визначення кривизни осі нитки що деформується для направляючій у формі тора. На рис.10.1б представлено графічне зображення співвідношення між вектором переміщення та вектором оберту натурального тригранника у випадку радіального охоплення.

Представимо одиничні орти головного тригранника в точці A_* у вигляді

$$\vec{\tau}_* = \vec{\tau}_0 - \alpha \vec{b}_0 + \beta \vec{n}_0, \vec{n}_* = \vec{n}_0 - \beta \vec{\tau}_0 + \gamma \vec{b}_0, \vec{b}_* = \vec{b}_0 + \alpha \vec{\tau}_0 - \gamma \vec{n}_0, \quad (10.4)$$

де α, β, γ - проєкції вектора повороту ξ на осі головного тригранника $A\tau_0 n_0 b_0$ у випадку радіального охоплення.

Перші похідні одиничних ортів (10.4) по дуговій координаті через проекції, для направляючої у формі тора, можна представити у вигляді

$$\begin{aligned} \frac{\partial U_{\Delta\tau}^\beta}{\partial S} - q_0 U_{\Delta\nu}^\beta + p_0 U_{\Delta\beta}^\beta = 0, \quad \frac{\partial U_{\Delta\nu}^\beta}{\partial S} + q_0 U_{\Delta\tau}^\beta - r_0 U_{\Delta\beta}^\beta = \beta, \\ \frac{\partial U_{\Delta\beta}^\beta}{\partial S} - p_0 U_{\Delta\tau}^\beta + p_0 U_{\Delta\nu}^\beta = -\alpha, \quad q_0 = \frac{\cos \Psi_0}{\rho_0}, \quad p_0 = \frac{\sin \Psi_0}{\rho_0}, \quad r_0 = \frac{1}{\rho_{01}} + \frac{\partial \Psi_0}{\partial S}, \end{aligned} \quad (10.5)$$

ρ_{01} - радіус кривизни геометричного кручення осі нитки для направляючої у формі тора;

Ψ_0 - кут Сен-Венана у випадку радіального охоплення.

Якщо нехтувати проекцією вектора \vec{U}_Δ^β на дотичну вісь ($U_{\Delta\tau}^\beta = 0$) і вважати, що нитка розташовується в дотичній площині ($U_{\Delta\beta}^\beta = 0$), то тоді з (10.5) визначимо

$$\frac{1}{\rho_1} = \frac{1}{\rho_0} + \frac{\partial^2 U_{\Delta\nu}^\beta}{\partial S^2} - \frac{U_{\Delta\nu}^\beta}{\rho_0^2}.$$

Перейдемо до визначення швидкостей і прискорень точок осі нитки яка деформується в зоні контакту у випадку радіального охоплення. На рис. 2.2 представлена розрахункова схема для визначення швидкостей та прискорень нитки в зоні деформації для направляючої у формі тора. Положення точки A_* щодо нерухомої координатної системи O_1XYZ визначається радіус-вектором R_* . Положення точки A на осі незім'ятої нитки визначається радіус-вектором R у випадку радіального охоплення. При деформації поперечного перетину у випадку радіального охоплення в зоні контакту точка A переміститься в положення A_* . Вектор AA_* позначимо через U_Δ^β .

Диференціюючи векторне рівняння для радіус-вектора за часом, отримаємо

$$\vec{V}_* = \vec{V} + \frac{\partial \vec{U}_\Delta^\beta}{\partial t}, \quad (10.6)$$

де V - швидкість точки A у випадку радіального охоплення.

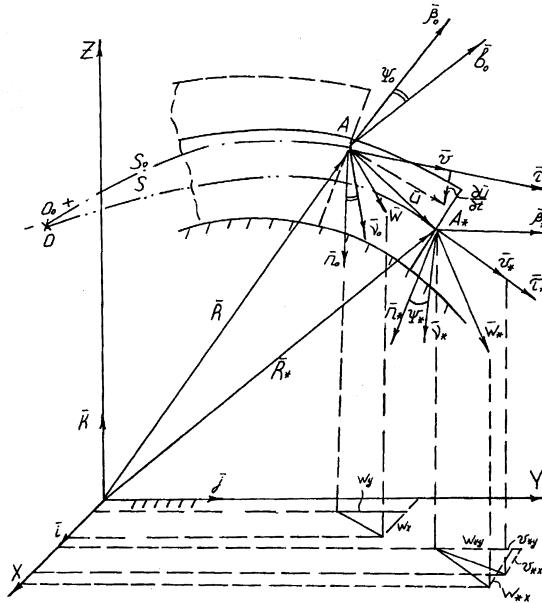


Рис.10.2. Розрахункова схема для визначення швидкостей та прискорень
нитки в зоні контакту у випадку радіального охоплення

Система рівнянь, для швидкостей точок осі нитки що деформується в зоні контакту, для направляючій у формі тора у випадку радіального охоплення матиме наступні вирази

$$\begin{aligned} \left(\frac{\partial \vec{V}_x}{\partial S}\right)_\tau &= \frac{\partial V_{x\tau}}{\partial S} + V_{xb} p_1 - V_{xn} q_1, & \left(\frac{\partial \vec{V}_x}{\partial t}\right)_\tau &= \frac{\partial V_{x\tau}}{\partial t} + \omega_2 V_{*b} - \omega_3 V_{*n}, \\ \left(\frac{\partial \vec{V}_x}{\partial S}\right)_n &= \frac{\partial V_{xn}}{\partial S} - V_{xb} r_1 + V_{x\tau} q_1, & \left(\frac{\partial \vec{V}_x}{\partial t}\right)_n &= \frac{\partial V_{xn}}{\partial t} - \omega_1 V_{*b} + \omega_3 V_{*\tau}, \\ \left(\frac{\partial \vec{V}_x}{\partial S}\right)_b &= \frac{\partial V_{xb}}{\partial S} + V_{xn} r_1 - V_{x\tau} p_1, & \left(\frac{\partial \vec{V}_x}{\partial t}\right)_b &= \frac{\partial V_{xb}}{\partial t} + \omega_1 V_{*n} - \omega_2 V_{*\tau}, \end{aligned} \quad (10.7)$$

де $r_1 = \frac{1}{\rho_{*1}} + \frac{\partial \psi_*}{\partial S}$, $p_1 = \frac{\sin \psi_*}{\rho_1}$, $q_1 = \frac{\cos \psi_*}{\rho_1}$ - складові вектора повної кривизни у

випадку радіального охоплення;

ψ_* - значення кута Сен-Венана в точці A_* ;

$V_{x\tau}, V_{xn}, V_{xb}$ - відповідно проекції вектора швидкості V_* на осі τ, n, b головного тригранника для направляючій у формі тора;

$\omega_1, \omega_2, \omega_3$ - відповідно проекції вектора кутової швидкості ω_e елемента нитки на осі головного тригранника у випадку радіального охоплення (з урахуванням кутової швидкості деформаційного кручення [1]).

Система диференціальних рівнянь (10.7), для направляючій у формі тора, дозволяє визначити закон зміни швидкості V_* залежно від часу t . Для визначення залежності між проекціями вектора прискорення від дугової координати у випадку радіального охоплення, диференціюємо систему (10.7) по t , отримаємо

$$\begin{aligned} \left(\frac{\partial \vec{W}_*}{\partial t}\right)_\tau &= \frac{\partial W_{*\tau}}{\partial t} + \frac{\partial \omega_2}{\partial S} V_{*b} + \omega_2 \frac{\partial V_{*b}}{\partial S} - \frac{\partial \omega_3}{\partial S} V_{*n} - \omega_3 \frac{\partial V_{*n}}{\partial S}, \\ \left(\frac{\partial \vec{W}_*}{\partial t}\right)_n &= \frac{\partial W_{*n}}{\partial S} + \frac{\partial \omega_3}{\partial S} V_{*\tau} + \omega_3 \frac{\partial V_{*\tau}}{\partial S} - \frac{\partial \omega_1}{\partial S} V_{*b} - \omega_1 \frac{\partial V_{*b}}{\partial S}, \\ \left(\frac{\partial \vec{W}_*}{\partial t}\right)_b &= \frac{\partial W_{*b}}{\partial S} + \frac{\partial \omega_1}{\partial S} V_{*n} + \omega_1 \frac{\partial V_{*n}}{\partial S} - \frac{\partial \omega_2}{\partial S} V_{*\tau} - \omega_2 \frac{\partial V_{*\tau}}{\partial S}. \end{aligned}$$

де $\left(\frac{\partial \vec{W}_*}{\partial t}\right)_\tau, \left(\frac{\partial \vec{W}_*}{\partial t}\right)_n, \left(\frac{\partial \vec{W}_*}{\partial t}\right)_b$ - проекції вектора першої похідної прискорення точки A_* по дуговій координаті на осі головного тригранника для направляючій у формі тора.

Для складання систем диференційних рівнянь, які описують рівновагу елемента нитки на напрямній поверхні у формі тора, необхідно всі зовнішні сили і реакції зв'язків, нитки, що діють на елемент нескінченно малої довжини dS , представити у вигляді головного вектора R_0 і головного моменту M_0 . Дані вектора приведемо до точки A_* - центру мас виділеного елемента нитки у випадку радіального охоплення. До даної точки приводиться і результуюча сила інерції Φ_* і вектор моменту інерції M_F . Головний трикутник τ_*, n_*, b_* , з вершиною в точці A_* , служить координатною системою у випадку радіального охоплення, щодо якої розглядається умовна рівновага всіх сил і моментів для направляючій у формі тора [1, 2]. Векторні рівняння

$$\frac{\partial \vec{R}}{\partial S} + \vec{R}_0 - \vec{\Phi}_* = 0, \quad \frac{\partial \vec{M}}{\partial S} + \vec{M}_0 + \vec{\tau}_* \times \vec{Q} - \vec{M}_\phi = 0, \quad (10.8)$$

описують рух нескінченно малого елемента нитки у випадку радіального охоплення. Головні вектор R та момент M , для направляючої у формі тора, визначаються наступними векторними рівняннями

$$\vec{R} = P\vec{\tau}_* + Q_2\vec{n}_* + Q_3\vec{b}_*, \quad \vec{M} = M_k\vec{\tau}_* + M_{u2}\vec{n}_* + M_{u3}\vec{b}_*. \quad (10.9)$$

Для отримання системи диференціальних рівнянь, для направляючої у формі тора, необхідно векторні рівняння (10.8) спроекувати на осі головного тригранника τ_* , n_* , b_* , з урахуванням (10.9) отримаємо

$$\begin{aligned} \frac{\partial P}{\partial S} - Q_2q_1 + Q_3p_1 - 4\frac{\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{aR^b}{P_0^b} N \cos \phi_t &= T \left[\frac{\partial(V_\tau + U_{\Delta\tau}^{\beta'})}{\partial t} + \right. \\ &+ (\omega_{0n} + \omega_{un})(V_b + U_{\Delta b}^{\beta'}) - (\omega_{0b} + \omega_{ub})(V_n + U_{\Delta n}^{\beta'}) \left. \right], \\ \frac{\partial Q_2}{\partial S} + Pq_1 + Q_3r_1 - N &= T \left[\frac{\partial(V_n + U_{\Delta n}^{\beta'})}{\partial t} - \right. \\ &- (\omega_{0\tau} + \omega_{u\tau})(V_b + U_{\Delta b}^{\beta'}) + (\omega_{0b} + \omega_{ub})(V_\tau + U_{\Delta\tau}^{\beta'}) \left. \right], \\ \frac{\partial Q_3}{\partial S} - Pp_1 + Q_2r_1 + 4\frac{\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{aR^b}{P_0^b} N \sin \phi_t &= T \left[\frac{\partial(V_b + U_{\Delta b}^{\beta'})}{\partial t} + \right. \\ &+ (\omega_{0\tau} + \omega_{u\tau})(V_n + U_{\Delta n}^{\beta'}) - (\omega_{0n} + \omega_{un})(V_\tau + U_{\Delta\tau}^{\beta'}) \left. \right], \\ \frac{\partial M_k}{\partial S} - M_{u2}q_1 + Bq_1p_1 + M_\tau &= \gamma_n \left[J_{\tau_0} \pm \Delta_\tau (\vec{U}_\Delta^\beta) \right] \varepsilon_1, \\ \frac{\partial M_{u2}}{\partial S} + M_kq_1 + Bq_1r_1 - Q_3 + M_n &= \gamma_n \left[J_{n_0} \pm \Delta_n (\vec{U}_\Delta^\beta) \right] \varepsilon_2, \\ \frac{\partial(Bq_1)}{\partial S} - M_kp_1 + M_{u2}r_1 + Q_2 + r_x &4\frac{\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{aR^b}{P_0^b} N = \gamma_n \times \\ \times \left[J_{b_0} \pm \Delta_b (\vec{U}_\Delta^\beta) \right] \varepsilon_3, \quad N &= b_2E_1\delta_1(1 - b_3\delta_1^{b_4}) + \eta\delta_1^{b_5}(1 - b_5\delta_1^{b_6}), \end{aligned} \quad (10.10)$$

та рівняння для визначення головних компонентів кривизни у випадку радіального охоплення

$$\begin{aligned}
 q_1 &= \frac{\cos \Psi_0}{\rho_0} + \frac{\partial \beta}{\partial S} + \frac{\alpha}{\rho_{01}} + \alpha \frac{\partial \Psi_0}{\partial S}, \quad p_1 = \frac{\sin \Psi_0}{\rho_0} + \frac{\partial \alpha}{\partial S} - \frac{\beta}{\rho_{01}} - \beta \frac{\partial \Psi_0}{\partial S}; \\
 r_1 &= \frac{1}{\rho_0} + \frac{\partial \Psi_0}{\partial S} - \alpha \frac{\cos \Psi_0}{\rho_0} + \frac{\sin \Psi_0}{\rho_0} \beta, \quad \gamma = \frac{\partial U_{\Delta\tau}^\beta}{\partial S} - \frac{\cos \Psi_0}{\rho_0} U_{\Delta\nu}^\beta + \frac{\sin \Psi_0}{\rho_0} U_{\Delta\beta}^\beta; \\
 \beta &= \frac{\partial U_{\Delta\nu}^\beta}{\partial S} + \frac{\cos \Psi_0}{\rho_0} U_{\Delta\tau}^\beta - \frac{U_{\Delta\beta}^\beta}{\rho_{01}} - U_{\Delta\beta}^\beta \frac{\partial \Psi_0}{\partial S}, \quad \alpha = \frac{\sin \Psi_0}{\rho_0} U_{\Delta\tau}^\beta - \frac{\partial U_{\Delta\beta}^\beta}{\partial S} - \frac{U_{\Delta\nu}^\beta}{\rho_{01}} - U_{\Delta\nu}^\beta \frac{\partial \Psi_0}{\partial S}.
 \end{aligned}
 \tag{10.11}$$

де B – коефіцієнт жорсткості нитки на згин;

ρ_0, ρ_{01} - радіуси кривизни і геометричного крутіння осі нитки у випадку радіального охоплення;

ψ_0 - кут Сен-Венана у випадку радіального охоплення;

$U_{\Delta\tau}^\beta, U_{\Delta\nu}^\beta, U_{\Delta\beta}^\beta$ – проекції вектора поперечної деформації перетину на осі натурального тригранника для направляючій у формі тора;

b - ширина сліду контакту для направляючій у формі тора;

δ_1' - швидкість відносної деформації поперечного перетину нитки у випадку радіального охоплення;

η - коефіцієнт, що характеризує в'язкі властивості нитки при її поперечній деформації у випадку радіального охоплення;

E_1 - поточний модуль жорсткості нитки на деформацію її перетину для направляючій у формі тора;

b_3, b_4, b_5, b_6 - експериментальні коефіцієнти, які визначаються з діаграми "навантаження - деформація" у випадку радіального охоплення [2].

10.2. ВЗАЄМОДІЯ НИТОК З НАПРАВЛЯЮЧИМИ СТРУКТУРНИМИ ЕЛЕМЕНТАМИ ВЕЛИКОЇ КРИВИЗНИ СИСТЕМИ ПОДАЧІ НИТКИ КРУГЛОВ'ЯЗАЛЬНОЇ МАШИНИ

В процесі переробки нитки взаємодіють з направляючими поверхнями великої кривизни у випадку радіального охоплення. На рис.10.3 а - д представлені такі випадки, коли має місце радіальне охоплення нитки

направляючою поверхнею[4-6]. Виникаючі при цьому навантаження в зоні контакту можуть призводити до обриву нитки.

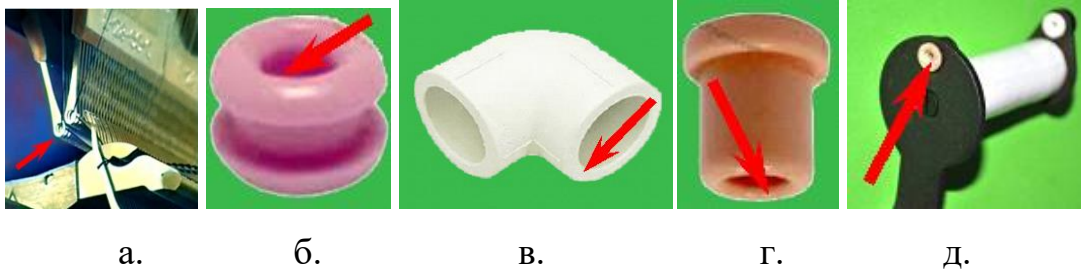


Рис.10.3. Робочі органи технологічних машин: а – отвори гребінок основов'язальної машини; б, г – спрямовувачі нитки у формі тора; в – перехідний елемент фідерного спрямовувача; д – пальцьовий натягувач нитки

На рис.10.4 показана загальна розрахункова схема взаємодії нитки з напрямною у формі тора у випадку радіального охоплення. Нитка огинає циліндричну напрямну поверхню у формі тора радіусу R . Вхідний натяг нитки дорівнює P_0 , а вихідний натяг нитки дорівнює P . Кут охоплення ниткою циліндричної напрямної у випадку радіального охоплення дорівнює

$$\varphi = \varphi_P + \varphi_{см1} + \varphi_{см2} - \varphi_{уж1} - \varphi_{уж2}, \quad (10.12)$$

де φ_P - кут охоплення ниткою напрямної у формі тора без урахування змінання та жорсткості на згин;

$\varphi_{см1}, \varphi_{см2}$ - кути, на які збільшується кут φ_P за рахунок деформації змінання в зоні контакту нитки з напрямною у формі тора;

$\varphi_{уж1}, \varphi_{уж2}$ - кути, на які зменшується кут φ_P за рахунок наявності жорсткості нитки на згин у випадку радіального охоплення.

Пошук взаємозв'язку між натягом ведучої P та веденої P_0 гілки нитки необхідно починати з встановлення залежності між натягом P_0 та натягом в точці P_A (точка входу нитки на напрямну у формі тора) та

натягом P та натягом в точці P_B (точка сходу нитки з напрямної у формі тора у випадку радіального охоплення).

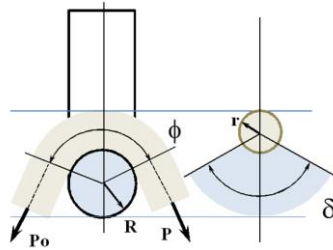


Рис.10.4. Загальна розрахункова схема взаємодії нитки з напрямною у формі тора

На цих ділянках вільні гілки нитки підпорядковуються законам пружних на згин одновимірних об'єктів. Завдяки наявності жорсткості нитки на згин реальний кут охоплення напрямної у формі тора буде становити

$$\varphi = \varphi_P - \gamma_0 - \gamma,$$

де γ_0, γ - кути жорсткості, які визначаються нахилом дотичних в точках A та B напрямної у формі тора до вертикальної прямої (або вісі y) у випадку радіального охоплення.

Таким чином, необхідно визначити значення кутів γ_0, γ . Після розв'язання рівняння рівноваги пружної лінії для направляючої у формі тора отримаємо наступні залежності [3-5]

$$\cos \gamma_0 = 1 - \frac{B_0}{2P_0(R+r)^2}, \quad \cos \gamma = 1 - \frac{B_0}{2P(R+r)^2}. \quad (10.13)$$

де $B_0 = EI$ - коефіцієнт жорсткості нитки на згин у випадку радіального охоплення;

E - модуль пружності нитки на розтягування;

$I = \pi d^4 / 64$ - осьовий момент інерції перетину нитки;

d - розрахунковий діаметр комплексної нитки чи пряжі.

Перейдемо для розгляду питання, яке спрямовано на визначення закону зміни натягу нитки на ділянці AB поверхні напрямної у формі тора.

Система диференціальних рівнянь у випадку радіального охоплення, яка описує рівновагу нескінченно малого елемента нитки $dS = (R+r)d\varphi$ (S - дугова координата), має вид [4, 5, 7]

$$\frac{dP}{ds} = F_{mp}, \quad \frac{P}{[R+r(1-\delta_1)]} = N, \quad N = bE_1\delta_1, F_{mp} = fN^n, f = \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{a}{b\phi^{n_1}}, \quad (10.14)$$

де P - натяг нити для направляючої у формі тора;

F_{mp} - сила тертя, яка діє на нескінченно малого елемента нитки;

N - питома нормальна реакція направляючої поверхні у формі тора;

S - дугова координата у випадку радіального охоплення;

b - ширина сліду контакту нитки з направляючою поверхнею;

де - a, b, n, n_1 деякі константи, значення яких залежить від виду матеріалів нитки та направляючої у формі тора та умов взаємодії між ними;

E_1 - модуль пружності нитки при стисненні на направляючій поверхні у формі тора [2] .

З системи (10.14), розділяючи змінні, отримаємо наступне диференціальне рівняння

$$\frac{d}{ds} \{ [R+r(1-\delta_1)] bE_1\delta_1 \} = \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{a}{b\phi^{n_1}} (bE_1)^n (\delta_1)^n .$$

Проінтегруємо останнє диференціальне рівняння. В остаточному вигляді отримаємо формулу у випадку радіального охоплення

$$(\delta_1)^{1-n} = (\delta_{01})^{1-n} + \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{(1-n)a(bE_1)^{n-1}}{b(1-n_1)} \phi^{1-n_1} . \quad (10.15)$$

При $n \rightarrow 1$, отримаємо остаточну

$$\delta_1 = \delta_{01} e^{\frac{4\sin(\frac{\delta}{2}) a \phi^{1-n_1}}{\delta + \sin(\delta) b^{(1-n_1)} [bE_1(\delta_0)]^{1-n}}} \quad (10.16)$$

Для випадку, коли коефіцієнт тертя для направляючої у формі тора є постійною величиною, враховуючи рівняння (10.14), при $n_1 = 0$, отримаємо

$$\delta_1 = \delta_{01} e^{\frac{4\sin(\frac{\delta}{2}) a \phi}{\delta + \sin(\delta) b [bE_1(\delta_0)]^{1-n}}} \quad (10.17)$$

Підставляємо (10.17) в перше рівняння системи (10.14), після інтегрування будемо мати

$$P = P_0 \left[1 + \frac{(R+r)}{[R+r(1-\delta_{01})]} \left(e^{\frac{4\sin(\frac{\delta}{2}) a \phi}{\delta + \sin(\delta) b}} - 1 \right) \right] + \left[\frac{B_0}{2[R+r(1-\delta_1)]^2} \right] - \left[\frac{B_0}{2[R+r(1-\delta_{01})]^2} \right] \left[1 + \frac{(R+r)}{[R+r(1-\delta_{01})]} \left(e^{\frac{4\sin(\frac{\delta}{2}) a \phi}{\delta + \sin(\delta) b}} - 1 \right) \right] \quad (10.18)$$

Кут φ , для направляючої у формі тора, можна визначити з використанням формул

$$\varphi_{см1} = \arccos \left[1 - \delta_{01} \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{см2} = \arccos \left[1 - \delta_1 \left(\frac{2r}{R} \right)^2 \right], \quad \varphi_{умс1} = \arccos \left[1 - \frac{B_0}{2P_0(R+r)^2} \right], \quad \varphi_{умс2} = \arccos \left[1 - \frac{B_0}{2P(R+r)^2} \right] \quad (10.19)$$

Рівняння (10.18) необхідно вирішувати з використанням систем (10.16), (10.19). Отримана залежність для направляючої у формі тора представляє трансцендентне рівняння відносно P . Для його вирішення використовувалися чисельні методи з використанням алгоритму дихотомії (ділення відрізка навпіл). Для цього було розроблене спеціальне програмне забезпечення.

В таблиці 10.1 представлені результати розрахунків по (10.18) залежності натягу ведучої гілки нитки P для: поліамідної комплексної нитки (ПАН) 174 Т (1 крива), $B=1,3\text{сН}\cdot\text{мм}^2$, $P_0=10\text{ сН}$, $r=0,24\text{ мм}$, $R=0,5 -$

10 мм, $\varphi_p=3,14$ рад, $n=0,82$, $\delta=0,488$ рад; вовняної пряжі (ВП) 93,5 Т (2 крива), $B=0,3$ сН·мм², $P_0=10$ сН, $r=0,2$ мм, $R=0,5 - 10$ мм, $\varphi_p=3,14$ рад, $n=0,85$, $\delta=0,488$ рад; бавовняної пряжі (БП) 100Т(3 крива), $B=2,5$ сН·мм², $P_0=10$ сН, $r=0,21$ мм, $R=0,5 - 10$ мм, $\varphi_p=3,14$ рад, $n=0,78$, $\delta=0,488$ рад.

По даним таблиці 10.1 побудовані графічні залежності натягу ведучої гілки нитки P в залежності від натягу веденої гілки P_0 , радіусу кривизни спрямовувача у формі тора R та кута радіального охоплення δ , які представлені на рис.10.5.

Аналіз графічних залежностей на рис.2.5 показав, що натяг ведучої гілки нитки для трьох обраних зразків змінюється при зростанні радіусу кривини напрямної. На першій ділянці, при зміні радіусу з 0.5 до 2-5 мм, натяг зменшується. Це пов'язано зі зміною сумарного кута охоплення. Мінімальне значення натягу для поліамідної комплексної нитки 174 Т буде коли радіус дорівнюватиме 2.6 мм, для віскозної штапельної пряжі 93,5 Т коли радіус дорівнюватиме 3.8 мм та для бавовняної пряжі 100Т коли радіус дорівнюватиме 2.9 мм.

Таблиця 10.1

Залежність натягу нитки після спрямовувача у формі тора в залежності від вхідного натягу та радіусу кривизни поверхні тора в зоні контакту

$R, \text{мм}$	Натяг нитки після спрямовувача у формі тора $P, \text{сН}$ $\delta=28^0$								
	Натяг веденої гілки $P_0=5 \text{ сН}$			Натяг веденої гілки $P_0=10 \text{ сН}$			Натяг веденої гілки $P_0=15 \text{ сН}$		
	ПАН	ВП	БП	ПАН	ВП	БП	ПАН	ВП	БП
2.01	8.78	7.65	8.34	17.64	15.34	17.17	26.79	23.14	26.32
3.61	8.83	7.69	8.42	17.26	15.16	16.87	25.62	22.59	25.36
4.01	8.87	7.7	8.45	17.27	15.16	16.87	25.59	22.57	25.31
5.21	8.98	7.76	8.52	17.37	15.22	16.91	25.62	22.6	25.27

5.61	9.01	7.78	8.54	17.42	15.24	16.93	25.66	22.62	25.28
6.01	9.05	7.8	8.56	17.46	15.27	16.95	25.7	22.65	25.29
7.21	9.15	7.85	8.62	17.6	15.34	17.02	25.85	22.73	25.35
7.61	9.18	7.86	8.63	17.64	15.36	17.04	25.91	22.75	25.37
8.01	9.21	7.88	8.65	17.69	15.39	17.06	25.96	22.78	25.4
9.61	9.33	7.93	8.71	17.86	15.47	17.15	26.18	22.9	25.49
10.01	9.35	7.95	8.72	17.91	15.49	17.17	26.23	22.92	25.52
11.21	9.43	7.98	8.76	18.03	15.56	17.22	26.38	23	25.59
12.01	9.48	8.00	8.78	18.1	15.59	17.26	26.48	23.05	25.63

При подальшому зростанні радіусу напрямної буде відбуватися поступове зростання натягу. Це пояснюється зміною орієнтації окремих елементарних волокон відносно утворюючої поверхні у формі тора [4, 7]. Таким чином, постає питання про оптимізацію розмірів направляючих машин текстильної промисловості у випадку радіального охоплення.

В додатку А представлені результати визначення натягу в залежності від натягу веденої гілки P_0 , радіусу кривизни спрямовувача у формі тора R та кута радіального охоплення δ .

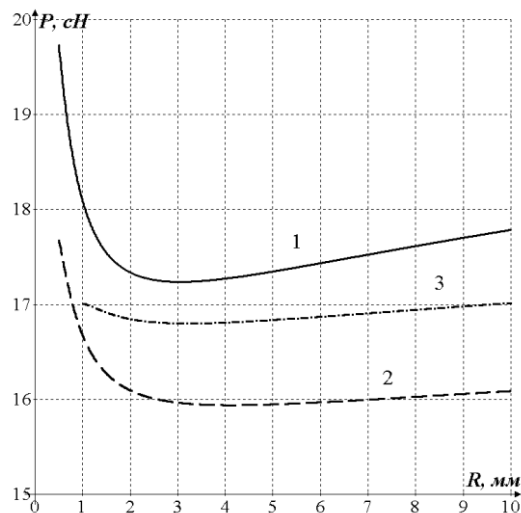


Рис.10.5. Залежності натягу ведучої гілки нитки P в залежності від натягу веденої гілки P_0 , радіусу кривизни спрямовувача у формі тора R та кута радіального охоплення δ

10.3. ВИЗНАЧЕННЯ ПРИВЕДЕНОГО КОЕФІЦІЄНТУ ТЕРТЯ ДЛЯ СТРУКТУРНИХ ЕЛЕМЕНТІВ СИСТЕМИ ПОДАЧІ НИТКИ КРУГЛОВ'ЯЗАЛЬНИХ МАШИН У ФОРМІ ТОРА

Недосконалість конструкції кільцевих та трубчастих спрямовувачів нитки не дозволяє здійснювати мінімізацію натягу в процесі роботи круглов'язальної машини [5-7]. Особливості процесу взаємодії нитки з направляючими та робочими органами круглов'язальної машини машин, коли має місце радіальне охоплення поверхнею напрямного елемента нитки, призводить до додаткового зростання її натягу до 30%, що визиває обриви [3]. Це негативно впливає на продуктивність устаткування та якість продукції що випускається. Вирішення цієї задачі має важливе значення для удосконалення технологічних процесів трикотажної промисловості з позиції підвищення продуктивності технологічного устаткування та якості продукції що випускається [3, 5-7]. Таким чином, тема даної статті є актуальною, яка має важливе значення для удосконалення конструкції існуючих кільцевих та трубчастих спрямовувачів та розробки нових [5].

Приведений коефіцієнт тертя, при взаємодії нитки з поверхнею кільцевих та трубчастих спрямовувачів, коли має місце радіальне охоплення поверхні нитки напрямною визначається за формулою [3]

$$f = 4 \frac{\sin\left(\frac{\delta}{2}\right)}{\delta + \sin(\delta)} f_0, \quad (10.20)$$

де f - приведений коефіцієнт тертя;

δ - радіальний кут охоплення поверхні нитки направляючою ;

f_0 - коефіцієнт тертя нитки по направляючій поверхні при відсутності радіального охоплення.

На рис.10.6 показана загальна розрахункова схема. При вирішенні цього питання, для комплексних ниток та пряжі, можна рахувати, що при взаємодією з внутрішньою поверхнею кільцевих та трубчастих

спрямовувачів окремі елементарні волокна будуть займати таке положення, при якому їх натяг буде мати мінімальне значення [3]. У цьому випадку можна рахувати, що перетин комплексної нитки чи пряжі деформується і прийме форму сегмента.

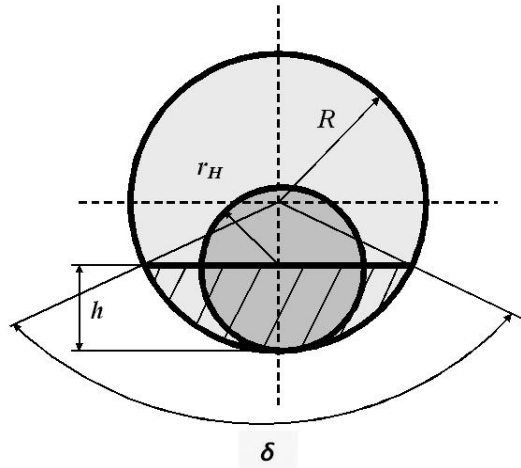


Рис.10.6. Загальна розрахункова схема

Площа сегмента буде визначатися за формулою

$$S_C = R^2 \arccos\left(\frac{R-h}{R}\right) - (R-h)\sqrt{2Rh-h^2}, \quad (10.21)$$

де S_C - площа сегмента;

R - внутрішній радіус кільцевих та трубчастих спрямовувачів;

h - висота сегмента.

Площа сегмента буде визначатися за формулою

$$S_C = \pi r_H^2, \quad (10.22)$$

де - r_H - радіус поперечного перетину нитки.

З урахуванням рівняння (2.3.3) рівняння (2.3.2) буде мати вигляд

$$\pi r_H^2 = R^2 \arccos\left(\frac{R-h}{R}\right) - (R-h)\sqrt{2Rh-h^2}. \quad (10.23)$$

Рівняння (10.23) представляє собою трансцендентне рівняння відносно h . Для його вирішення необхідно використовувати чисельні методи. Нами був обраний метод дихотомії.

Цільова функція буде мати вигляд

$$g = \pi r_H^2 - R^2 \arccos\left(\frac{R-h}{R}\right) - (R-h)\sqrt{2Rh-h^2}, \quad (10.24)$$

де g - значення цільової функції.

Перейдемо до розгляду методу ділення відрізка $[h_i, h_{i+1}]$ навпіл (метод дихотомії). Суть методу полягає в побудові ітераційної послідовності вкладених один в одного відрізків $[h_{in}, h_{i+1n}]$, кінці яких представляють монотонні послідовності $\{h_{in}\}, \{h_{i+1n}\}$, причому

$$x_i = \frac{h_i + h_{i+1}}{2}, \quad g_i = \pi r_H^2 - R^2 \arccos\left(\frac{R-x_i}{R}\right) - (R-x_i)\sqrt{2Rx_i-x_i^2} > 0,$$

$$h_{in} \leq \xi, \quad h_{i+1n} \geq \xi, \quad n = 1, 2, \dots$$

де ξ - корінь трансцендентного рівняння (10.23) на відрізку $[h_i, h_{i+1}]$.

Збіжність даного методу повільна. Проте при будь-якій ширині відрізання $[h_i, h_{i+1}]$ збіжність гарантована. Для реалізації алгоритму методу дихотомії була розроблена спеціальна програма на мові Object Pascal в середовищі Delphi.

Величина радіальний кут охоплення поверхні нитки направляючою буде визначатися за формулою

$$\delta = 2 \arccos\left(\frac{R-h}{R}\right). \quad (10.25)$$

Результати розрахунків за формулами (10.20), (10.23), (10.25) представлені в таблицях 10.2, 10.3 та на рис.10.7.

В результаті проведеного дослідження розроблені математичні моделі для визначення приведенного коефіцієнту тертя при взаємодії нитки з кільцевими та трубчастими спрямовувачами нитки трикотажних машин з урахуванням їх реальних фізико – механічних властивостей, структури та умов переробки на технологічному обладнанні.

Для розв'язання трансцендентного рівняння (10.24) розроблений алгоритм методу половинного розділення відрізка при розв'язанні трансцендентних рівнянь та програмне забезпечення для визначення

висоти сегменту заповненого матеріалом нитки, радіального кута охопту поверхні нитки напрямною поверхнею та приведенного коефіцієнту тертя.

Таблиця 10.2

Результати визначення висоти h сегмента заповненого матеріалом нитки

r_H	R=1.5	R=2	R=2.5	R=3	R=3.5	R=4	R=4.5	R=5
0,1	0,0572	0,0481	0,0430	0,0396	0,0370	0,0349	0,0334	0,0320
0,2	0,1449	0,1216	0,1085	0,0998	0,0933	0,0882	0,0840	0,0806
0,3	0,2507	0,2096	0,1868	0,1715	0,1603	0,1515	0,1444	0,1385
0,4	0,3712	0,3090	0,2749	0,2522	0,2356	0,2227	0,2122	0,2034
0,5	0,5047	0,4180	0,3712	0,3403	0,3178	0,3002	0,2859	0,2741
0,6	0,6509	0,5356	0,4748	0,4349	0,4058	0,3832	0,3650	0,3499
0,7	0,8095	0,6615	0,5851	0,5354	0,4993	0,4713	0,4488	0,4300
0,8	0,9813	0,7950	0,7016	0,6412	0,5976	0,5641	0,5369	0,5144
0,9	0,9999	0,9362	0,8241	0,7523	0,7007	0,6610	0,6290	0,6025
1,0	0,9999	0,9999	0,9521	0,8681	0,8079	0,7618	0,7248	0,6942

Встановлені графічні залежності приведенного коефіцієнту тертя від співвідношення внутрішнього радіусу кільцевих та трубчастих спрямовувачів та розрахункового радіусу поперечного перетину нитки, що дозволяє оптимізувати геометричні параметри робочих органів системи подачі ниток на трикотажному обладнанні з позиції мінімізації натягу в робочій зоні.

Таблиця 10.3

Значення приведенного коефіцієнту тертя

r_H	R=1.5	R=2	R=2.5	R=3	R=3.5	R=4	R=4.5	R=5
0,1	0,5643	0,4025	0,3211	0,2713	0,2369	0,2114	0,1921	0,1766
0,2	0,8710	0,6350	0,5101	0,4319	0,3777	0,3375	0,3065	0,2817
0,3	1,0863	0,8181	0,6638	0,5644	0,4947	0,4428	0,4025	0,3701
0,4	1,2307	0,9665	0,7944	0,6793	0,5973	0,5357	0,4875	0,4486
0,5	1,3128	1,0863	0,9067	0,7808	0,6891	0,6194	0,5644	0,5200
0,6	1,3375	1,1808	1,0035	0,8711	0,7720	0,6957	0,6352	0,5859
0,7	1,3080	1,2519	1,0863	0,9515	0,8475	0,7660	0,7007	0,6471

0,8	1,2271	1,3011	1,1562	1,0230	0,9161	0,8307	0,7614	0,7043
0,9	1,2158	1,3294	1,2140	1,0863	0,9785	0,8905	0,8182	0,7580
1,0	1,2158	1,3354	1,2603	1,1418	1,0351	0,9456	0,8711	0,8085

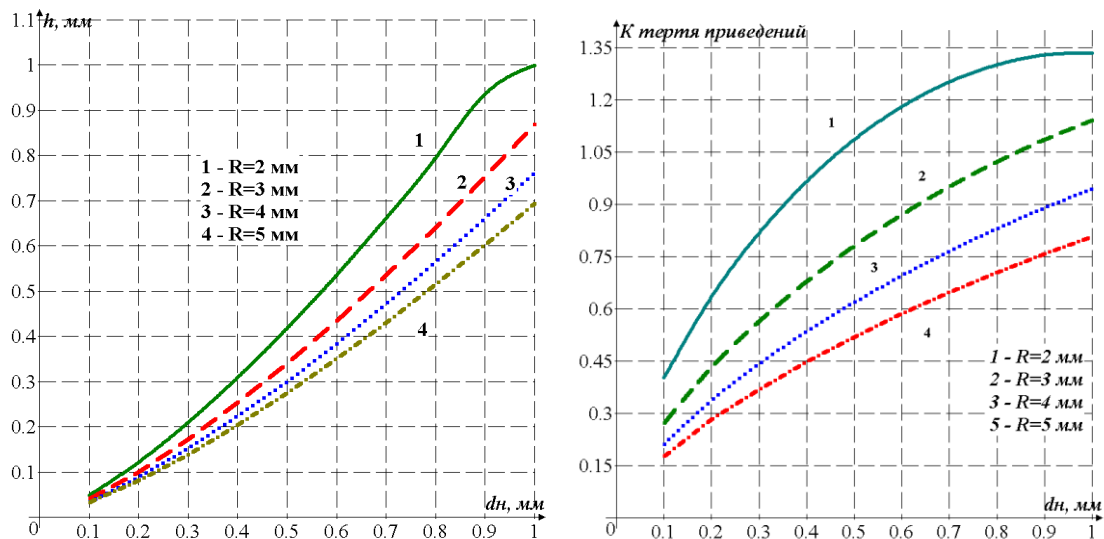


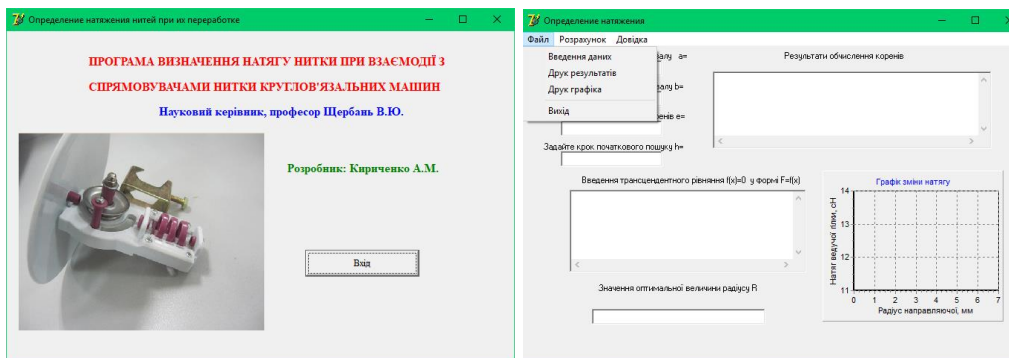
Рис.10.7. Графічні залежності приведенного коефіцієнту тертя та висоти h сегменту

10.4. АЛГОРИТМІЧНЕ ТА КОМП'ЮТЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ НАТЯГУ НИТКИ ПІСЛЯ НАПРЯМНОЇ ВЕЛИКОЇ КРИВИЗНИ

Програма для визначення натягу нитки після напрямної великої кривизни у формі тора включає п'ять форм: $Tfrm1NN1 = class(TForm)$; $Tfrm1NN2 = class(TForm)$; $Tfrm1NN3 = class(TForm)$; $Tfrm1NN4 = class(TForm)$; $TF Errors = class(TForm)$. На першій формі програми (рис.2.8а) розташовані наступні елементи: $btn1NN1: TButton$; $lbl1NN1: TLabel$; $lbl2NN1: TLabel$; $lbl3NN1: TLabel$; $lbl4NN1: TLabel$; $Image1: TImage$.

На основній формі програми $Tfrm1NN2 = class(TForm)$ (рис.2.8б) розташоване головне меню програми $MainMenu1: TMainMenu$. Воно включає три розділи меню: файл; розрахунок; довідка. Розділ меню файл включає наступні підрозділи: введення даних; друк результатів; друк графіка; вихід. Розділ розрахунок включає наступні підрозділи:

розрахувати натяг; побудувати графік; розрахувати оптимум. На цій формі розташовані п'ять одно строкових вікон для введення даних та виводу результатів розрахунків: edt1NN2: TEdit; edt2NN2: TEdit; edt3NN2: TEdit; edt4NN2: TEdit; edt5NN2: TEdit. Одно строкове вікно edt1NN2: TEdit призначене для завдання початкової границі інтервалу пошуку кореня трансцендентного рівняння для визначення натягу нитки після напрямної великої кривизни у формі тора. Одно строкове вікно edt2NN2: TEdit призначене для завдання кінцевої границі інтервалу пошуку кореня трансцендентного рівняння. Одно строкове вікно edt3NN2: TEdit призначене для завдання точності обчислення коренів трансцендентного рівняння.



а.

б.

Рис.10.8. Перша форма Tfrm1NN1 = class (TForm) програми та основна форма програми для визначення натягу нитки після напрямної у формі тора

Одно строкове вікно edt4NN2: TEdit призначене для завдання кроку початкового в ітераційному процесі. Одно строкове вікно edt5NN2: TEdit призначене для виводу значення оптимальної величини радіусу R напрямної великої кривизни у формі тора.

Багато строкове вікно Memo1: TMemo призначено для введення трансцендентного рівняння (10.18) у формі $F = f(P)$. Багато строкове вікно mem1NN2: TMemo призначене для визначення результатів

обчислень коренів трансцендентного рівняння (10.18). В цьому вікні виводяться значення натягу ведучої гілки нитки після напрямної великої кривизни у формі тора в залежності від радіусу кривизни R .

При активації підрозділу введення даних, розділу меню файл, виконується процедура `procedure Tfrm1NN2.n2NN2Click(Sender: TObject)`. При цьому активується форма `Tfrm1NN4 = class(TForm)`. Ця форма призначена для введення необхідних даних для обчислення натягу нитки(рис.10.9).

Одно строкове вікно `edt2NN4: TEdit` призначене для введення натягу веденої гілки P_0 до напрямної у формі тора. Одно строкове вікно `edt3NN4: TEdit` призначене для введення значення ширини сліду контакту b_1 нитки на напрямній у формі тора. Одно строкове вікно `edt4NN4: TEdit` призначене для введення значення розрахункового радіусу r перетину нитки чи пряжі. Одно строкове вікно `edt5NN4: TEdit` призначене для введення значення модуля пружності нитки чи пряжі при стисненні E_1 . Одно строкове вікно `edt6NN4: TEdit` призначене для введення значення модуля жорсткості нитки чи пряжі при вигині B_0 . Одно строкове вікно `edt7NN4: TEdit` призначене для введення значення розрахункового значення кута охоплення напрямної у формі тора φ_P .

Рис.10.9. Форма `Tfrm1NN4 = class(TForm)` для введення даних для обчислення натягу нитки

Цей кут визначається при умові відсутності деформації перетину нитки чи пряжі в зоні контакту з напрямною у формі тора та жорсткості нитки на згин. Одно строкове вікно Edit1: TEdit призначене для введення значення кута радіального охоплення напрямною поверхнею нитки в зоні контакту δ . Одно строкові вікна edt8NN4: TEdit, edt9NN4: TEdit призначені для введення значення емпіричних коефіцієнтів a_2, b_2 .

Процедура `procedure Tfrm1NN2.n7NN2Click(Sender: TObject)` призначена для виконання розрахунку натягу нитки.

В якості алгоритму для визначення кореня трансцендентного рівняння (10.18) будемо використовувати метод ділення відрізка навпіл (метод дихотомії). На рис.10.10 представлена блок-схема алгоритму ділення відрізка навпіл. Необхідно побудувати ітераційну послідовність вкладених один в одного відрізків $[a_n, b_n]$, кінці яких представляють монотонні послідовності. Необхідні умови для ітераційної послідовності мають вигляд

$$a_n \leq \xi, b_n \geq \xi, n = 1, 2, \dots$$

де ξ - корінь трансцендентного рівняння (2.2.28) на відріжку $[a, b]$.

Будемо вважати, що $f(a)f(b) < 0$ та $f(a) < 0, f(b) > 0$. Беремо середню точку відрізка $[a, b]$

$$\xi_1 = \frac{a+b}{2}.$$

Обчислюємо в даній точці значення функції $f(x)$. Припустимо, що $f(\xi_1) = 0$. В цьому випадку процес визначення кореня трансцендентного рівняння закінчується. Якщо $f(\xi_1) \neq 0$, то розглядаємо два відрізка $[a, \xi_1]$ та $[\xi_1, b]$. Вибираємо той з них, де виконується умова $f(a)f(\xi_1) < 0$ або $f(\xi_1)f(b) < 0$. Вибраний відрізок знову поділяємо навпіл приймаючи

$a_1 = a, b_1 = \xi_1$ або $a_1 = \xi_1, b_1 = b$. Обчислюємо нове значення центру відрізка

$$\xi_2 = \frac{a_1 + b_1}{2}.$$

Продовження процесу ділення дозволяє отримати ітераційну послідовність вкладених один в одного відрізків $[a_n, b_n]$. Необхідна система нерівностей має вигляд

$$a_n \leq a_{n+1} < b_{n+1} \leq b_n.$$

Аналіз останньої нерівності показує, що величина a_n визначає корінь ξ трансцендентного рівняння з лівого боку обраного відрізка, а величина b_n з правого боку обраного відрізка. Помилка при цьому не перевищує поточну довжину відрізка $b_n - a_n$ і прагне до нуля при збільшенні n за законом геометричної прогресії із знаменником 0,5. Число ітерацій N , необхідне для досягнення заданої точності ε , визначається з виразу

$$N \approx \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2}.$$

Програма дозволяє визначати натяг нитки після направляючої у формі тора для різних значень радіусу кривизни R напрямної великої кривизни у формі тора. В програмі реалізовані два цикли. Перший дозволяє визначати натяг нитки після напрямної великої кривизни у формі тора для фіксованого значення радіусу кривизни R напрямної великої кривизни у формі тора. Зовнішній цикл забезпечує зміну радіусу кривизни на обраному інтервалі, величина якого обумовлена конструкцією окремих спрямовувачів круглов'язальної машини великого діаметру.

При виконанні процедури `procedure Tfrm1NN2.n7NN2Click(Sender: TObject)` результати послідовно відображаються в багато строковому вікні `mem1NN2: TMemo`.

Виконання процедури `procedure Tfrm1NN2.n8NN2Click(Sender: TObject)` (рис.10.11a) дозволяє будувати графічну залежність величини натягу ведучої гілки нитки в залежності від радіусу кривизни R напрямної великої кривизни у формі тора. Величина інтервалу зміни радіусу направляючої може змінюватися програмно.

Комп'ютерна програма дозволяє визначити радіусу кривизни R напрямної великої кривизни у формі тора, при якому натяг ведучої гілки нитки буде мінімальним. Для цього реалізована процедура `procedure Tfrm1NN2.n9NN2Click(Sender: TObject)`.

Введення вхідних даних для реалізації алгоритму ділення відрізка навпіл здійснюється за допомогою наступних програмних операторів: `Val(edt1NN2.Text,a,code)`; `Val(edt2NN2.Text,b,code)`; `Val(edt3NN2.Text,e,code)`; `Val(edt4NN2.Text,h,code)`.

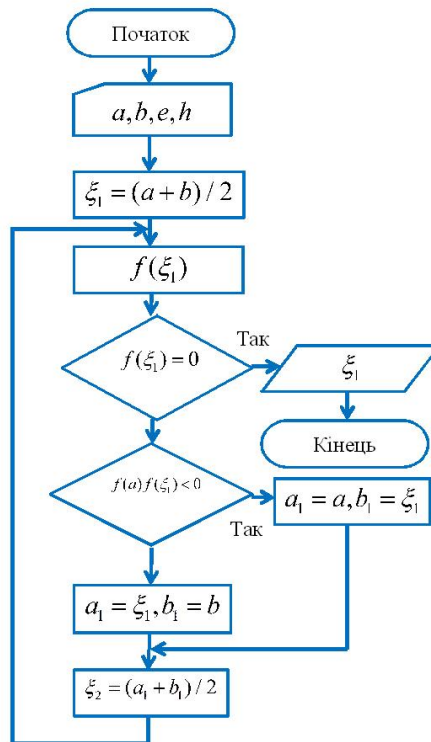


Рис.10.10. Блок-схема алгоритму ділення відрізка навпіл

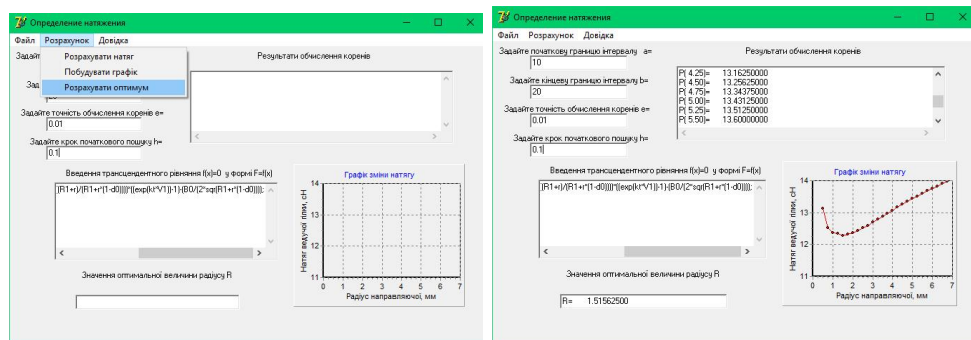
При введенні трансцендентного рівняння в багато строкове вікно Мето1: ТМето трансцендентного рівняння (10.18) у формі $F = f(P)$ в

програмі реалізований транслятор з використанням зворотних польських записів. Це здійснюється за допомогою окремого модуля unit Synt. Цей модуль в програмі реалізований без форми.

Для контролю введення інформації у багато строкове вікно Memo1: TMemo та виправлення помилок при вводі використовується форма TFErrors = class(TForm).

На рис.10.11б представлені результати роботи програми для визначення величини натягу ведучої гілки нитки в залежності від радіусу кривизни напрямної великої кривизни у формі тора.

В додатку наведений лістинг програми для визначення значення натягу при взаємодії нитки з напрямною у формі тора з використанням зворотних польських записів.



а.

б.

Рис.10.11. Процедура procedure Tfrm1NN2.n8NN2Click(Sender: TObject) та результати роботи програми для визначення величини натягу ведучої гілки

НИТКИ

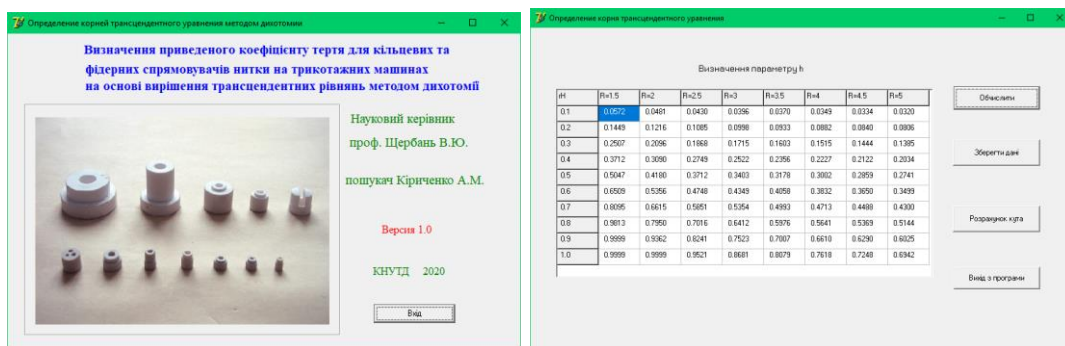
10.5. АЛГОРИТМІЧНЕ ТА КОМП'ЮТЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ПРИВЕДЕНОГО КОЕФІЦІЄНТУ ТЕРТЯ ДЛЯ СТРУКТУРНИХ ЕЛЕМЕНТІВ СИСТЕМИ ПОДАЧІ НИТКИ КРУГЛОВ'ЯЗАЛЬНИХ МАШИН У ФОРМІ ТОРА

Програма для визначення приведенного коефіцієнту тертя для структурних елементів системи подачі нитки круглов'язальних машин у

формі тора включає п'ять форм: $TfrmTUDI1 = class(TForm)$; $TfrmTUDI2 = class(TForm)$; $TForm1 = class(TForm)$; $TForm2 = class(TForm)$; $TForm3 = class(TForm)$. На рис.10.12а представлена перша форма $TfrmTUDI1 = class(TForm)$ програми.

На цій формі розташовані наступні компоненти: $lb111TUDI: TLabel$; $lb112TUDI: TLabel$; $lb113TUDI: TLabel$; $lb114TUDI: TLabel$; $btn111TUDI: TButton$; $Label1: TLabel$; $Label2: TLabel$; $Label3: TLabel$; $Label5: TLabel$; $GroupBox1: TGroupBox$; $Image1: TImage$.

Визначення приведенного коефіцієнту тертя включає три послідовні кроки: визначення висоти сегменту; обчислення радіального кута охоплення нитки поверхнею напрямної; визначення приведенного коефіцієнту тертя.



а.

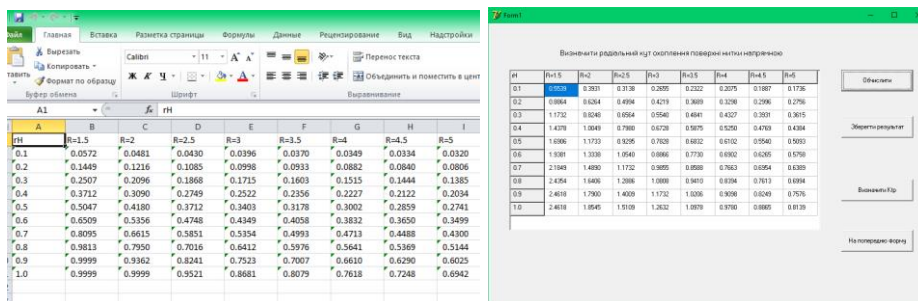
б.

Рис.10.12. Перша форма $TfrmTUDI1 = class(TForm)$ програми для визначення приведенного коефіцієнту тертя та друга форма $TfrmTUDI2 = class(TForm)$ для розрахунку висоти сегменту h

Друга форма $TfrmTUDI2 = class(TForm)$ (рис.10.12б) призначена для розрахунку висоти сегменту h . Для чого необхідно з використанням чисельним методів знайти корені трансцендентного рівняння (10.24). В якості алгоритму для визначення кореня трансцендентного рівняння (10.24) використовували метод ділення відрізка навпіл. Необхідно побудувати ітераційну послідовність для цільової функції

$$g_i = \pi r_H^2 - R^2 \arccos\left(\frac{R - x_i}{R}\right) - (R - x_i) \sqrt{2Rx_i - x_i^2} > 0.$$

На формі TfrmTUDI2 = class(TForm) розташовані наступні компоненти: btn21TUDI2: TButton; btn23TUDI2: TButton; StringGrid1: TStringGrid; Button1: TButton; Button2: TButton; Label1: TLabel. За допомогою компонента btn21TUDI2: TButton визивається процедура procedure btn21TUDI2Click (Sender: TObject) для розрахунку висоти сегменту h . Активація компонента Button1: TButton призводить до виконання процедури procedure Button1Click(Sender: TObject) збереження результатів розрахунку в таблиці Microsoft Excel (рис.10.13а).



а.

б.

Рис.10.13. Результати розрахунку висоти сегменту h в таблиці Microsoft Excel та обчислення радіального кута охоплення нитки поверхнею напрямної у формі тора

Активація компонента Button2: TButton призводить до виконання процедури procedure Button2Click(Sender: TObject) переходу на форму TForm1 = class(TForm) для обчислення радіального кута охоплення нитки поверхнею напрямної у формі тора.

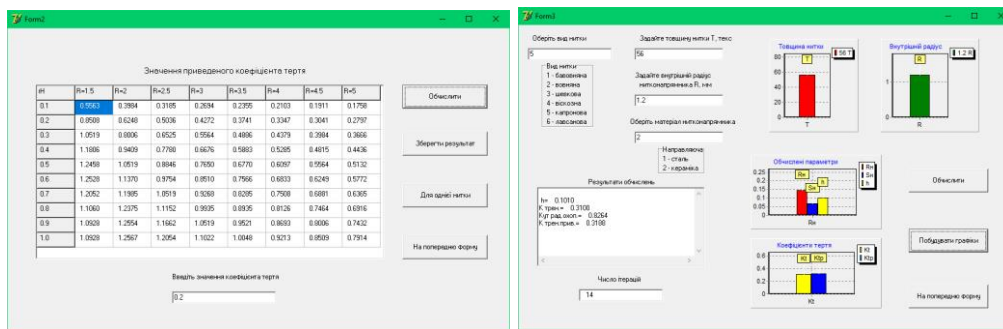
Активація компонента btn23TUDI2: TButton призводить до виконання процедури procedure btn23TUDI2Click(Sender: TObject) закінчення роботи комп'ютерної програми.

На формі TForm1 = class(TForm) (рис.10.13б) розташовані наступні компоненти: Button1: TButton; StringGrid1: TStringGrid; Button2: TButton; Button3: TButton; Button4: TButton; Label1: TLabel.

За допомогою компонента Button2: TButton визивається процедура procedure Button2Click(Sender: TObject) обчислення радіального кута охоплення нитки поверхнею напрямної у формі тора. Натиснення на кнопку Button3: TButton призводить до виконання процедури procedure Button3Click(Sender: TObject) збереження результатів розрахунку в таблиці Microsoft Excel. Активація компонента Button4: TButton призводить до виконання процедури procedure Button4Click(Sender: TObject) переходу на форму TForm2 = class(TForm) для визначення приведенного коефіцієнту тертя.

Активація компонента Button1: TButton призводить до виконання процедури procedure Button1Click(Sender: TObject) переходу до попередньої форми TfrmTUDI2 = class(TForm) комп'ютерної програми.

На формі TForm2 = class(TForm) (рис.10.14а) розташовані наступні компоненти: Button1: TButton; Button2: TButton; StringGrid1: TStringGrid; Button3: TButton; Button4: TButton; Label1: TLabel; Edit1: TEdit; Label2: TLabel.



а.

б.

Рис.10.14. Четверта форма TForm2 = class(TForm) для визначення приведенного коефіцієнту тертя та п'ята форма TForm3 = class(TForm) для визначення приведенного коефіцієнту тертя для конкретної комплексної нитки чи пряжі

За допомогою компонента Button2: TButton визивається процедура procedure Button2Click(Sender: TObject) визначення приведенного

коефіцієнту тертя. Активація компонента Button3: TButton призводить до виконання процедури procedure Button3Click(Sender: TObject) збереження результатів розрахунку в таблиці Microsoft Excel.

Активація компонента Button4: TButton призводить до виконання процедури procedure Button4Click(Sender: TObject) переходу на форму TForm3 = class(TForm) для визначення приведенного коефіцієнту тертя для конкретної комплексної нитки чи пряжі.

Активація компонента Button1: TButton призводить до виконання процедури procedure Button1Click(Sender: TObject) переходу до попередньої форми TForm1 = class(TForm) комп'ютерної програми.

На формі TForm3 = class(TForm) (рис.10.146) розташовані наступні компоненти: Button1: TButton; Label1: TLabel; Edit1: TEdit; Label2: TLabel; Edit2: TEdit; Label3: TLabel; Label4: TLabel; Edit3: TEdit; Label5: TLabel; Edit4: TEdit; GroupBox1: TGroupBox; Label6: TLabel; Label7: TLabel; Label8: TLabel; Label9: TLabel; Label10: TLabel; Label11: TLabel; GroupBox2: TGroupBox; Label12: TLabel; Label13: TLabel; Label14: TLabel; Label15: TLabel; Edit5: TEdit; Button2: TButton; Memo1: TMemo; Button3: TButton; Chart1: TChart; Chart2: TChart; Chart3: TChart; Chart4: TChart; Series1: TBarSeries; Series2: TBarSeries; Series3: TBarSeries; Series4: TBarSeries; Series5: TBarSeries; Series6: TBarSeries; Series7: TBarSeries.

Одно строкове вікно Edit1: TEdit призначене для обрання виду нитки чи пряжі: 1 – бавовняна; 2 – вовняна; 3 – шовкова; 4 – віскозна; 5 – капронова (поліамідна); 6 – лавсанова. Одно строкове вікно Edit2: TEdit призначене для завдання товщини нитки в текстах. Одно строкове вікно Edit3: TEdit призначене для введення значення внутрішнього радіусу напрямної у формі тора. Одно строкове вікно Edit4: TEdit призначене для обрання матеріалу напрямної: 1 – сталь; 2 – кераміка. Одно строкове вікно

Edit5: TEdit призначене для виведення числа ітерацій при чисельному визначенні коренів трансцендентного рівняння (10.24).

Багато строкове вікно Memo1: TMemo призначене для виведення результатів обчислень: висоти сегменту; радіального кута охоплення нитки поверхнею напрямної; приведенного коефіцієнту тертя.

Компоненти Chart1: TChart, Chart2: TChart, Chart3: TChart, Chart4: TChart призначені для побудови графічних залежностей введених та обчислених параметрів, товщини нитки.

За допомогою компонента Button2: TButton визивається процедура procedure Button2Click(Sender: TObject) для визначення висоти сегменту, радіального кута охоплення нитки поверхнею напрямної, приведенного коефіцієнту тертя для конкретної комплексної нитки чи пряжі. Активація компонента Button3: TButton призводить до виконання процедури procedure Button3Click(Sender: TObject) побудови графічних залежностей. Активація компонента Button1: TButton призводить до виконання процедури procedure Button1Click(Sender: TObject) переходу до попередньої форми TForm2 = class(TForm) комп'ютерної програми.

В додатку Б1 наведений лістинг програми для визначення приведенного коефіцієнту тертя при реалізації алгоритму дихотомії для трансцендентних рівнянь.

Висновки по розділу

1. Визначені геометричні параметри осі нитки, швидкості і прискорення точок вісі в процесі взаємодії нитки з направляючою у формі тора при урахуванні деформації перетину в зоні контакту, жорсткості на згин та зміни коефіцієнту тертя при радіальному охопленні нитки поверхнею напрямної.
2. Отримана система диференціальних рівнянь, що описує динаміку руху нитки жорсткої на вигин та кручення по направляючій поверхні великої

кривизни у формі тора з урахуванням деформації в зоні контакту, змінного вхідного натягу та діаметру перетину при радіальному охопленні нитки поверхнею напрямної.

3. Встановлені залежності вихідного натягу комплексної нитки чи пряжі від радіусу кривини направляючої поверхні великої кривини у формі тора з урахуванням вхідного натягу, кута охоплення, виду сировини, співвідношення радіусу перетину нитки та внутрішнього радіусу спрямовувача у формі тора, жорсткості на згин, деформації в зоні контакту, нелінійної залежності коефіцієнту тертя.

4. В результаті проведеного дослідження розроблені математичні моделі для визначення приведенного коефіцієнту тертя при взаємодії нитки з кільцевими та трубчастими спрямовувачами нитки трикотажних машин з урахуванням їх реальних фізико – механічних властивостей, структури та умов переробки на технологічному обладнанні.

5. Розроблені алгоритм та комп'ютерна програма для визначення значення натягу при взаємодії нитки з напрямними у формі тора системи подачі круглов'язальних машин на основі чисельної реалізації алгоритму половинного розділення відрізка при розв'язанні трансцендентних рівнянь та розроблений транслятор з використанням зворотних польських записів.

6. Розроблений алгоритм методу половинного розділення відрізка при розв'язанні трансцендентних рівнянь та програмне забезпечення для визначення висоти сегменту заповненого матеріалом нитки, радіального кута охоплення поверхні нитки напрямною поверхнею та приведенного коефіцієнту тертя.

ЛІТЕРАТУРА

1. Щербань В.Ю. Алгоритмічні, програмні та математичні компоненти САПР в індустрії моди/ В.Ю.Щербань, О.З.Колиско, М.І.Шолудько, В.Ю.Калашник. – К.:Освіта України, 2017. – 745 с.
2. Прогнозування процесів на основі моделювання часових рядів: навч. Посіб./П.І.Бідюк, В.Ю.Щербань, Є.О.Демківський, Т.І.Демківська.- К.:КНУТД, 2017.-324 с.
3. Слізков А.М., Щербань В.Ю., Кизимчук О.П. Механічна технологія текстильних матеріалів. Частина II. (Ткацьке, трикотажне та неткане виробництво): підручник / А.М.Слізков, В.Ю.Щербань, О.П.Кизимчук. – К.:КНУТД, 2018. – 276 с.
4. Щербань В.Ю. Механіка нитки/В.Ю.Щербань. – К.:Видавництво «Укрбланковидав». – 2018. – 533 с.
5. Shcherban' V., Melnyk G. , Sholudko M. and Kalashnyk V. Warp yarn tension during fabric formation/V.Shcherban' , G.Melnyk , M.Sholudko, V.Kalashnyk // *Fibres and Textiles*. – 2018. – volume 25. - №2. – pp.7-14.
6. Vasilchenko V.N. Steady motion of a textile yarn with two anchoring points over a rough surface / V.N. Vasilchenko, V.Yu. Shcherban, Ts.V. Apokin // *Technology of the textile industry*. - 1985. - № 4. - P.54-56.
7. Vasilchenko V.N. Equilibrium of a filament of a root base in the zone of formation of a multilayer technical fabric / V.N. Vasilchenko, V.Yu. Shcherban // *Technology of the textile industry*. - 1986. - № 5. - P.44-47.
8. Vasilchenko V.N. Influence of the twist of a capron complex filament on the value of its flexural rigidity / V.N. Vasilchenko, V.Yu. Shcherban // *Technology of the textile industry*. - 1986. - №4. - P.8-9.
9. Scherban V.Yu. Determination of the geometric characteristics of the shape of the filament axis moving along the deformable guide surface / V.Yu. Shcherban // *Technology of the textile industry*. - 1990. - №6. - P.52-55.

10. Scherban V.Yu. Determination of technological efforts in the process of surf during the formation of multilayer technical fabric / V.Yu. Shcherban // Technology of the textile industry. - 1990. - №3. - P.44-47.
11. Scherban V.Yu. Investigation of the process of duck surf during the formation of multilayer technical fabric / V.Yu. Shcherban // Technology of the textile industry. - 1990. - №4. - P.41-44.
12. Yakubitskaya I.A. Dynamic analysis of layout conditions on the end sections of the groove of the winding drum / I.A. Yakubitskaya, V.V. Chugin, V.Yu. Shcherban // Technology of the textile industry. - 1997. - №5. - P.33-37.
13. Yakubitskaya I.A. Differential equations of the relative motion of the filament element on the end sections of the coil of the winding drum / I.A. Yakubitskaya, V.V. Chugin, V.Yu. Shcherban // Technology of the textile industry. - 1997. - №6. - P.50-54.
14. Shcherban' V., Melnyk G. , Sholudko M. and Kalashnyk V. Warp yarn tension during fabric formation/V.Shcherban' , G.Melnyk , M.Sholudko, V.Kalashnyk // Fibres and Textiles. – 2018. – volume 25. - №2. – pp.7-14.
15. Ресурсоощадні технології виробництва текстилю, одягу та взуття: монографія: в 2 т. Т.1/Теоретичні основи та методи розроблення ресурсоощадних технологій та обладнання для виробництва текстилю, одягу та взуття/ В.Ю.Щербань, Б.Ф.Піпа, В.В.Чабан та ін. – К.:КНУТД, 2016. – 373 с.
16. Ресурсоощадні технології виробництва текстилю, одягу та взуття: монографія: в 2 т. Т.2/Підвищення надійності ресурсоощадних виробництв текстилю, одягу і взуття на основі новітніх технологій та системного управління/ В.Ю.Щербань, Б.Ф.Піпа, В.В.Чабан та ін. – К.:КНУТД, 2016. – 214 с.
17. Ресурсоощадні технології та обладнання швейної та текстильної промисловості: монографія: в 2 ч. Ч.1/Наукові основи та інженерні методи

проектування ресурсощадних технологій і обладнання швейної та текстильної промисловості/ В.Ю.Щербань, Г.Б.Параска, Б.В.Орловський та ін. – К.:КНУТД, 2015. – 339 с.

18. Ресурсощадні технології та обладнання швейної та текстильної промисловості: монографія: в 2 ч. Ч.2/Шляхи підвищення ефективності швейної та текстильної галузей України на базі новітніх технологій та управління/ В.Ю.Щербань, Г.Б.Параска, Б.В.Орловський та ін. – К.:КНУТД, 2015. – 270 с.

19. Прогнозування фізико-механічних властивостей текстильних матеріалів побутового призначення/А.М. Слізков , В.Ю. Щербань ,С.М. Краснитський , О.Б. Демківський. –К.:КНУТД, 2013. – 223 с.

20. ЩЕРБАНЬ В. Ю.Інформаційні технології в науці, виробництві та підприємстві/В.Ю.Щербань.-К.:КНУТД, 2016. – 184 с.

21. Scherban V. Interaction yarn guide surface/V.Scerban, M. Sholudko, V. Kalashnik, O. Kolisko//Intellectual Archive, Toronto: Shiny World Corp., Richmond Hill, Ontario, Canada. – May 2015. – Volume 4.- Number 3. – P. 10-15.

22. Scherban V. Kinematics of threads cooperates with the guiding surfaces of arbitrary profile /V. Scherban, N. Murza, O. Kolisko, M. Sheludko, I. Semenova //Intellectual Archive, Toronto: Shiny World Corp., Richmond Hill, Ontario, Canada. – May/June - 2016. – Volume 5.- Number 3. – P. 23-27.

23. Scherban V. Basic parameters of curvature and torsion of the deformable thread in contact with runner/V. Scherban, N.Murza, A.Kirichenko, O. Kolisko, M. Sholudko //Intellectual Archive, Toronto: Shiny World Corp., Richmond Hill, Ontario, Canada. – Nov/Des - 2016. – Volume 10.- Number 2. – P. 18-23.

24. Scherban V. Equalizations of dynamics of filament interactive with surface/V. Scherban, G. Melnik, A.Kirichenko, O. Kolisko, M.

- Sheludko//Intellectual Archive, Toronto: Shiny World Corp., Richmond Hill, Ontario, Canada. – January/February 2017. – Volume 6.- Number 1. – P. 22-26.
25. Shcerban V.Yj. Determination of the transfer function of the comb tensioner of the thread/V.Yj.Shcerban , I.S.Semenova , O.Z.Kolysko , M.I.Sholudko , N.I.Murza //Вісник КНУТД, 2016.- №3.- С.44-49.
26. Щербань В.Ю. Порівняльний аналіз роботи нитконатягувачів текстильних машин/ В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько//Вісник ХНУ.- 2016.-№6(243). - с.18-21.
27. Щербань В.Ю. Ефективність роботи компенсаторів натягу нитки трикотажних машин/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько //Вісник ХНУ.- 2017.-№1(245). - с.83-86.
28. Щербань В.Ю. Динамічні навантаження рукавичних автоматів зі зворотно-поступальним рухом кареток/В.Ю.Щербань, Б.Ф.Пипа, О.В.Чабан//Вісник КНУТД, 2016.- №3.- С.118-125.
29. Колиско М.І. Визначення оптимальних геометричних параметрів циліндричних нитконапрямних машин легкої промисловості/ М.І.Колиско, В.Ю.Щербань// Вісник КНУТД. – 2014. - №1. – С. 119-124.
30. Колиско М.І. Дослідження взаємодії нитки з циліндричною напрямною поверхнею/ М.І.Колиско, В.Ю.Щербань// Вісник КНУТД. – 2013. - №5. – С44-47.
31. Колиско М.І. Визначення впливу змінання та жорсткості на згин на натяг нитки при її взаємодії з циліндричною напрямною/ М.І.Колиско, В.Ю.Щербань// Вісник ХНУ. – 2013. - №6. – С.10-13.
32. Щербань В.Ю. Дослідження впливу матеріалу нитки і анізотропії тертя на її натяг і форму осі/ В.Ю.Щербань, В.Ю.Калашник, О.З.Колиско, М.І.Шолудько // Вісник ХНУ. – 2015. - №2. - С.25-29.
33. Щербань В.Ю. Оптимізація процесу взаємодії нитки з напрямними з урахуванням анізотропії фрикційних властивостей/ В.Ю.Щербань,

М.І.Шолудько, О.З.Колиско, В.Ю.Калашник // Вісник ХНУ. – 2015. - №3. - С.30-33.

34. Патент 80199 UA, МКИ D04B 15/38/ Пристрій для визначення фрикційних властивостей нитки/ Мельник Г.В., Калашник В.Ю., Колиско М.І., Щербань В.Ю., Колиско О.З. (UA). – №и 2013 04588; заявл.12.04.2013; опубл. 13.05.2013, Бюл.№ 9.

35. Патент 110982 UA, МКИ D05B 47/00/ Нитконатягувальний пристрій/ Щербань В.Ю., Мурза Н.І., Колиско О.З., Шолудько М.І., Семенова І.С. (UA). – №и 2016 04374; заявл.21.04.2016; опубл. 25.10.2016, Бюл.№ 20.

36. Патент 110548 UA, МКИ D04B 15/38/ Система ниткоподачі/ Щербань В.Ю., Мурза Н.І., Колиско О.З., Шолудько М.І., Семенова І.С. (UA). – №и 2016 04693; заявл.26.04.2016; опубл. 10.10.2016, Бюл.№ 19.

ДОДАТОК

PROGRAM PRE_MAT_MAX_SUM;

{ Програма перетворення квадратної матриці на основі послідовного підсумовування модулів всіх членів матриці по рядках і стовпцях і відповідних перестановок елементів по убутанню цих сум }

uses

```

  Crt; Var N,i,j,k,i1,w,j1:integer; v,v1,s1,v3:real;A:array[1..50,1..50] of real;
  B:array[1..50] of real; s:array[1..50] of real; Begin    ClrScr; Write('Задайте
число рівнянь вхідних в систему N='); Read(N); for i:=1 to N do    begin
  WriteLn('Введення коефіцієнтів для',i,'-го рівняння матриці');
for j:=1 to N do    begin Write('Введіть коефіцієнт матриці
A(',i,',',j)=');
Read(A[i,j]); END; begin Write('Введіть значення для матриці вільних
членів B(',i)='); Read(B[i]);    END;    END;
{ Блок перетворення матриці коефіцієнтів по убутанню суми
абсолютних значень коефіцієнтів по рядках } for i:=1 to N do begin
s1:=0; for j:=1 to N do begin s1:=s1+abs(A[i,j]); s[i]:=s1; end;
end; for i:=1 to N do begin k:=i; for i1:=i+1 to N do begin if s[i1]>s[k] then
k:=i1;
end; v3:=s[i];s[i]:=s[k];s[k]:=v3; for j:=1 to N do begin
v:=A[i,j];A[i,j]:=A[k,j];A[k,j]:=v; end; v1:=B[i];B[i]:=B[k];B[k]:=v1; end;
{ Блок перетворення матриці коефіцієнтів по убутанню суми
абсолютних значень коефіцієнтів по стовпцях } for j:=1 to N do
begin  s1:=0; for i:=1 to N do begin s1:=s1+abs(A[i,j]); s[j]:=s1;
end; end; for j:=1 to N do begin k:=j; for j1:=j+1 to N do begin if s[j1]>s[k] then
k:=j1; end; v3:=s[j];s[j]:=s[k];s[k]:=v3; for i:=1 to N do begin
v:=A[i,j];A[i,j]:=A[i,k];A[i,k]:=v; end; end; for i:=1 to N do begin
for j:=1 to N do  begin Write(A[i,j]:4:0); end; WriteLn(B[i]:4:0); end;
END.
```


PROGRAM PRE_MATR_VGE;

```
{ Програма перетворення квадратної матриці на основі принципу вибору
головного(найбільшого по модулю) елемента з подальшою перестановкою
рядків і стовпців}uses Crt; Var N,i,j,i5,w,w1:integer; v,v1,max:real;
A:array[1..50,1..50] of real; B:array[1..50] of real;Begin ClrScr;
Write('Задайте число рівнянь вхідних в систему N='); Read(N); for i:=1 to N
do begin WriteLn('Введення коефіцієнтів для',i,'-го рівняння матриці'); for
j:=1 to N do begin Write('Введіть коефіцієнт матриці A(',i,',',j)=');
Read(A[i,j]); END; begin Write('Введіть значення для матриці вільних
членів B(',i)='); Read(B[i]); END; END; for i5:=1 to N-1 do begin
max:=abs(A[i5,i5]);w:=i5;w1:=i5; for i:=i5 to N do begin for j:=i5 to N do
begin if abs(A[i,j]) >max then begin w:=i;w1:=j;max:=abs(A[w,w1]);
end; end; end;
{ Блок перестановки рядків матриці по убаванню залежно від значення
модуля головного елемента на кожному кроці} for j:=1 to N do begin
v:=A[i5,j];A[i5,j]:=A[w,j];A[w,j]:=v; end;
v1:=B[i5];B[i5]:=B[w];B[w]:=v1;
{ Блок перестановки стовпців матриці по убаванню залежно від значення
модуля головного елемента на кожному кроці} for i:=1 to N do begin
v:=A[i,i5];A[i,i5]:=A[i,w1];A[i,w1]:=v; end; end; for i:=1 to N do begin
for j:=1 to N do begin Write(A[i,j]:4:0); end; WriteLn(B[i]:4:0); end;
END.
```

PROGRAM S_G_PE;

USES Crt;

```
{ Вирішення системи з N лінійних рівнянь методом Гауса послідовного
виключення невідомих }
```

```
var N,i,j,u,m,k,z:integer; v,v1,H:real; A:array[1..50,1..50] of real;
B:array[1..50] of real; X:array[1..50] of real; begin ClrScr;
```

```

Write('Задайте число рівнянь вхідних в систему N='); Read(N); for i:=1 to N
do begin WriteLn('Введення коефіцієнтів для',i,'-го рівняння матриці');for
j:=1 to N do begin Write('Введіть коефіцієнт матриці A('i','j')=');
Read(A[i,j]); END; begin
Write('Введіть значення для матриці вільних членів B('i')='); Read(B[i]);
END;END;{ begin u:=1; for m:=1 to N do begin for i:=1*u to N do begin
k:=i; for j:=i+1 to N do if abs(A[j,m]) >abs(A[k,m]) then k:=j;
for z:=1 to N do begin v:=A[i,z];A[i,z]:=A[k,z];A[k,z]:=v;END;
v1:=B[i];B[i]:=B[k];B[k]:=v1; END; u:=u+1; END; END;}
for i:=1 to N-1 do begin for j:=i+1 to N do begin A[j,i]:=-A[j,i]/A[i,i];
for k:=i+1 to N do begin A[j,k]:=A[j,k]+A[j,i]*A[i,k]; END;
B[j]:=B[j]+A[j,i]*B[i]; END; END; X[N]:=B[N]/A[N,N]; for i:=N-1 downto
1 do
begin H:=B[i]; for j:=i+1 to N do begin H:=H-X[j]*A[i,j]; END;
X[i]:=H/A[i,i]; END; WriteLn('Коріння системи лінійних рівнянь');
for i:=1 to N do begin WriteLn('X('i')=',X[i]:10:4); END; END.

```

PROGRAM S_G_VGE;

USES Crt;

{ Вирішення системи з N лінійних рівнянь методом Гауса з вибором
головного елементу }

var N,i,j,u,m,k,z,N1,k1,L,j1:integer; v,v1,H,S:real; A:array[1..50,1..50] of
real;

B:array[1..50] of real; X:array[1..50] of real; C:array[1..50,1..50] of real;

G:array[1..50] of real; begin ClrScr;

Write('Задайте число рівнянь вхідних в систему N='); Read(N);

for i:=1 to N do begin WriteLn('Введення коефіцієнтів для',i,'-го рівняння
матриці'); for j:=1 to N do begin Write('Введіть коефіцієнт матриці
A('i','j')='); Read(A[i,j]); END; begin

Write('Введіть значення для матриці вільних членів B('i')=');

```

Read(B[i]); END; END; begin u:=1; for m:=1 to N do begin
for i:=1*u to N do begin k:=i; for j:=i+1 to N do if abs(A[j,m]) >abs(A[k,m])
then k:=j; for z:=1 to N do begin v:=A[i,z];A[i,z]:=A[k,z];A[k,z]:=v; END;
v1:=B[i];B[i]:=B[k];B[k]:=v1; END; u:=u+1; END; END; N1:=N-1; for k:=1
to N1 do begin if abs(A[до, до]) >0 then begin G[k]:=B[k]/A[до,
до];k1:=k+1;
for i:=k1 to N do begin B[i]:=B[i]-A[i,k]*G[k]; for j1:=k to N do
begin j:=N-j1+k;C[k,j]:=A[k,j]/A[до, до]; A[i,j]:=A[i,j]-A[i,k]*C[k,j];
END; END; END else begin k1:=k+1; for m:=k1 to N do
begin if abs(A[m,k]) >0 then for L:=1 to N do begin
v:=A[k,L];A[k,L]:=A[m,L];A[m,L]:=v; END; END;
begin v:=B[k];B[k]:=B[m];B[m]:=v; G[k]:=B[k]/A[до, до];k1:=k+1;
for i:=k1 to N do begin B[i]:=B[i]-A[i,k]*G[k];
for j1:=k to N do begin j:=N-j1+k;C[k,j]:=A[k,j]/A[до, до];
A[i,j]:=A[i,j]-A[i,k]*C[k,j]; END; END;END; END; END;
m:=N;X[m]:=B[m]/A[m,m]; for m:=N-1 downto 1 do begin
S:=0; for L:=m to N1 do begin S:=S+C[m,L+1]*X[L+1]; END;
X[m]:=G[m]-S; END; for i:=1 to N do begin WriteLn('X('i)=';X[i]:10:4);
END;END.

```

PROGRAM S_I_PRO;

USES Crt;

```

{ Вирішення системи з N лінійних рівнянь методом простих ітерацій}
var N,i,j,u,k,z1,m:integer; S,E,v,v1:real; A:array[1..50,1..50] of real;
B:array[1..50] of real; X:array[1..50] of real; z:array[1..50] of real;
BEGIN ClrScr; Write("Задайте число рівнянь вхідних в систему N=");
Read(N); Write("Задайте погрішність обчислення E="); Read(E);
for i:=1 to N do begin WriteLn("Введення коефіцієнтів для',i,'-го рівняння
матриці'); for j:=1 to N do begin Write("Введіть коефіцієнт матриці
A('i','j)='); Read(A[i,j]); END; begin Write("Введіть значення для

```

```

матриці вільних членів B('i')='); Read(B[i]); END; END; begin u:=1; for
m:=1 to N do
begin for i:=1*u to N do begin k:=i; for j:=i+1 to N do begin if abs(A[j,m])
>abs(A[k,m]) then k:=j; END; for z1:=1 to N do begin
v:=A[i,z1];A[i,z1]:=A[k,z1];A[k,z1]:=v; END; v1:=B[i];B[i]:=B[k];B[k]:=v1;
END; u:=u+1; END; END; S:=0; for i:=1 to N do begin
WriteLn("Задайте початкове наближення для кореня XO('i')=");
Read(z[i]); end; repeat k:=0; for i:=1 to N do begin X[i]:=-B[i];
for j:=1 to N do begin X[i]:=X[i]+A[i,j]*z[j]; end;
if abs(X[i]/A[i,i]) >=E then k:=1; X[i]:=z[i]-(X[i]/A[i,i]); end; for i:=1 to N do
begin z[i]:=X[i]; end; S:=S+1; until k<>1; WriteLn('Коріння системи
лінійних рівнянь'); for i:=1 to N do begin WriteLn('X('i')=',X[i]:10:4); end;
WriteLn('Число итерацій',S:10:0); END.

```

PROGRAM S_I_ZEID;

USES

```

Crt; { Вирішення системи з N лінійних рівнянь методом Зейделя}
var N,i,j,u,k,z1,m:integer; S,E,v,v1:real; A:array[1..50,1..50] of real;
B:array[1..50] of real; X:array[1..50] of real; z:array[1..50] of real;
BEGIN ClrScr; Write("Задайте число рівнянь вхідних в систему N=");
Read(N); Write("Задайте погрішність обчислення E="); Read(E);
for i:=1 to N do begin WriteLn("Введення коефіцієнтів для,i',i'-го рівняння
матриці"); for j:=1 to N do begin Write("Введіть коефіцієнт матриці
A('i','j')="); Read(A[i,j]); END; begin
Write("Введіть значення для матриці вільних членів B('i')=");
Read(B[i]); END; END; begin u:=1; for m:=1 to N do
begin for i:=1*u to N do begin k:=i; for j:=i+1 to N do
begin if abs(A[j,m]) >abs(A[k,m]) then k:=j; END; for z1:=1 to N do

```

```

begin v:=A[i,z1];A[i,z1]:=A[k,z1];A[k,z1]:=v; END;
v1:=B[i];B[i]:=B[k];B[k]:=v1; END; u:=u+1; END; END; S:=0;for i:=1 to N
do begin WriteLn('Задайте початкове наближення для кореня X0('i')=');
Read(z[i]); end; repeat k:=0; for i:=1 to N do begin X[i]:=-B[i];
for j:=1 to N do begin X[i]:=X[i]+A[i,j]*z[j]; end; if abs(X[i]/A[i,i]) >=E then
k:=1;
X[i]:=z[i]-(X[i]/A[i,i]); z[i]:=X[i]; end; S:=S+1; until k<>1;
WriteLn('Коріння системи лінійних рівнянь'); for i:=1 to N do
begin WriteLn('X('i')=',X[i]:10:4); end; WriteLn('Число ітерацій',S:10:0);
END.

```

PROGRAM S_M_VRAS;

USES Crt;

{ Вирішення системи з N лінійних рівнянь методом обертання }

var N,i,j,u,m1,k,z:integer; L,R,v,v1,m:real; A:array[1..50,1..50] of real;

B:array[1..50] of real; B1:array[1..50] of real;BEGIN ClrScr;

Write('Задайте число рівнянь вхідних в систему N='); Read(N);m:=0;

for i:=1 to N do begin

WriteLn('Введення коефіцієнтів для',i,'-го рівняння матриці');

for j:=1 to N do begin Write('Введіть коефіцієнт матриці A('i','j')=');

Read(A[i,j]); END; begin

Write('Введіть значення для матриці вільних членів B('i')=');

Read(B[i]); END; END; begin u:=1; for m1:=1 to N do begin for i:=1*u to N

do

begin k:=i; for j:=i+1 to N do begin if abs(A[j,m1]) >abs(A[k,m1]) then k:=j;

END; for z:=1 to N do begin v:=A[i,z];A[i,z]:=A[k,z];A[k,z]:=v; END;

v1:=B[i];B[i]:=B[k];B[k]:=v1; END; u:=u+1; END; END; for i:=1 to N-1 do

begin for k:=i+1 to N do begin if A[i,i]=A[k,i] then begin if A[k,i]=0 then

begin m:=1;L:=0; end else begin m:=sqrt((A[i,i]*A[i,i])+(A[k,i]*A[k,i]));

```

L:=-A[k,i]/m;m:=A[i,i]/m; end; end else begin
m:=sqrt((A[i,i]*A[i,i])+(A[k,i]*A[k,i])); L:=-A[k,i]/m;m:=A[i,i]/m;
end; for j:=1 to N do begin R:=(m*A[i,j])-L*A[k,j];
A[k,j]:=L*A[i,j]+(m*A[k,j]);
A[i,j]:=R; end; R:=(m*B[i])-L*B[k]; B[k]:=(L*B[i])+m*B[k];
B[i]:=R; end; end; for i:=N downto 1 do begin m:=0; for k:=0 to N-i-1 do
begin m:=m+B1[N-k]*A[i,N-k]; end; B1[i]:=(B[i]-m)/A[i,i];
WriteLn('X(',i)=' ',B1[i]:10:3); end;END.

```

PROGRAM KOR_POLIN;

```

{ Визначення всього коріння полінома  $A_n(X^n)+A_{n-1}(X^{n-1})+\dots+A_1X+A_0=0$ 
з дійсними коефіцієнтами}USES Crt; var
e,C,T,A1,S,R,P,Q,H,M,L,U,D,V,Y,Z,X,B,W,F:real; N,j,K:integer;
A:array[1..50] of Real;procedure V1; begin N:=N-2;A1:=0;
if sqrt((S-(R*C/2))*(S-(R*C/2))+R*R*abs(H)) <=e then begin A1:=1; end;
B:=0; if N>=2 then begin B:=1; end;end;procedure V2;begin P:=-C/2;
Q:=sqrt(abs(H)); if H>=0 then begin M:=P+Q;P:=P-Q;Q:=0; end else
begin M:=P; end;end;procedure V3;label M1;begin begin repeat begin
repeat M1: if M<>10 then begin M:=M+1; H:=0;Q:=A[1]; P:=A[2]-C*Q;
L:=Q;
end else begin P:=C; M:=0; Q:=D; C:=U; D:=V; U:=P;V:=Q;Y:=C; Z:=D;
F:=-F;
M:=M+1; H:=0; Q:=A[1]; P:=A[2]-C*Q; L:=Q; end; for j:=3 to N do begin
R:=P; P:=A[j]-C*R-D*Q; Q:=R; R:=L; L:=Q-C*R-H*D; H:=R; end;
Q:=A[N+1]-D*Q; S:=L+C*R; if T=0 then begin X:=D*R; H:=R*X+S*L;
if H=0 then begin {Continue;} goto M1; end; end; until True; end;
C:=C+((P*S-Q*R)/H); D:=D+((P*X+Q*L)/H); if (C-Y+D-Z) <>0 then
begin end else begin if F=-W then begin WriteLn('Немає рішення');
Exit; end else begin W:=-F; end; end; H:=(C*C/4) -D;

```

```

if sqrt((Q-(P*C/2))*(Q-(P*C/2))+P*P*abs(H)) >e/N then begin { Continue;}
goto M1; end; until True; end; T:=0; A[2]:=A[2]-C*A[1]; for j:=3 to N-1 do
begin A[j]:=A[j]-C*A[j-1]-D*A[j-2]; end; V2;end;procedure V4;begin
for j:=3 to N do begin R:=P; P:=A[j]-C*R-D*Q; Q:=R; R:=L; L:=Q-C*R-
H*D;
H:=R; end; Q:=A[N+1]-D*Q; S:=L+C*R; if T=0 then begin X:=D*R;
H:=R*X+S*L; if H=0 then begin V3; end; end;
C:=C+((P*S-Q*R)/H); D:=D+((P*X+Q*L)/H); if (C-Y+D-Z) <>0 then
begin end else begin if F=-W then begin WriteLn('Немає рішення');
Exit; end else begin W:=-F; end; end; H:=(C*C/4) -D;
if sqrt((Q-(P*C/2))*(Q-(P*C/2))+P*P*abs(H)) >e/N then begin V3; end;
T:=0; A[2]:=A[2]-C*A[1]; for j:=3 to N-1 do begin A[j]:=A[j]-C*A[j-1]-
D*A[j-2];
end; V2;end;begin ClrScr; WriteLn('Задайте погрішність визначення
коріння'); Write('e='); ReadLn(e); WriteLn('Задайте ступінь полінома');
Write('N='); ReadLn(N); WriteLn('Введіть коефіцієнти полінома');
for j:=1 to N+1 do begin Write('A(',N+1-j)='); ReadLn(A[j]); end; K:=1;
repeat T:=1;C:=A[2]/A[1]; if N=1 then begin P:=-C; Q:=0;
WriteLn('X(',K)=';P:12:5;'+j*(',-Q:12:5');'); K:=K+1; if T<>0 then Break;
V1; end else begin if N=2 then begin H:=(C*C/4) -(A[3]/A[1]);
V2; WriteLn('X(',K)=';M:12:5;'+j*(';Q:12:5');'); K:=K+1;
WriteLn('X(',K)=';P:12:5;'+j*(',-Q:12:5');'); K:=K+1; if T<>0 then Break;
V1; end else begin M:=10; C:=4; D:=8; U:=4; V:=8; F:=1; W:=2; T:=0;
V3; WriteLn('X(',K)=';M:12:5;'+j*(';Q:12:5');'); K:=K+1;
WriteLn('X(',K)=';P:12:5;'+j*(',-Q:12:5');'); K:=K+1; if T<>0 then Break;
V1; end; end; if A1+B=2 then begin T:=1; V4;
WriteLn('X(',K)=';M:12:5;'+j*(';Q:12:5');'); K:=K+1;
WriteLn('X(',K)=';P:12:5;'+j*(',-Q:12:5');'); K:=K+1; if T<>0 then Break;V1;
end; until False;END.

```

PROGRAM SU_TR_U_M_N_M;

```
{ Вирішення системи трансцендентних рівнянь  $F_i(X_i)=0$  модифікованим
методом Ньютона}USES Crt; const
l1=300;l3=400;l2=400;h1=100;co=4;ct=28;n1=1;n2=7;lca=316.228; var
e,x1,h,lao,la1o,lod,lcao,lca1o,lodt,p1,p2,pt:Real; N,M,s,i,j,k,R:integer;
A:array[1..50,1..50] of Real; B:array[1..50] of Real; x:array[1..50] of Real;
F:array[1..50] of Real; procedure V;
{ Представляємо трансцендентні рівняння у вигляді  $f_i(X_i)=0$  і
підставляємо праві частини отриманих рівнянь у вирази
 $F(i)=f(X_i)$  } begin { Значення розтягнутої ділянки основи до ремізної рамки
lca:=sqrt(sqr(l1)+sqr(h1));} lao:=sqrt(sqr(l2-x[1])+sqr(h1-x[2]));
la1o:=sqrt(sqr(l2-x[1])+sqr(h1+x[2])); lod:=sqrt(sqr(l3+x[1])+sqr(x[2]));
lcao:=lca+lao-(l1+l2);lca1o:=lca+la1o-(l1+l2);lodt:=lod-l3;
p1:=n2*co*lcao;p2:=n1*co*lca1o;pt:=ct*lodt;
F[1]:=(-p1*(l2-x[1])/lao)+(-p2*(l2-x[1])/la1o)+(pt*(l3+x[1])/lod);
F[2]:=(p1*(h1-x[2])/lao)+(-p2*(h1+x[2])/la1o)+(-pt*x[2]/lod); end;begin
ClrScr; WriteLn('Задайте число рівнянь'); Write('N='); ReadLn(N);
WriteLn('Задайте максимальне число ітерацій'); Write('M='); ReadLn(M);
WriteLn('Задайте відносну погрішність'); Write('e='); ReadLn(e);
s:=0; WriteLn('Введіть початкові значення'); for i:=1 to N do begin
Write('X',i'(0)='); ReadLn(x[i]); end; repeat V; for i:=1 to N do begin
B[i]:=-F[i]; end; for j:=1 to N do begin x1:=x[j]; h:=e*abs(x1); x[j]:=x1+h;
V; for i:=1 to N do begin A[i,j]:=(F[i]+B[i])/h; end; x[j]:=x1; end; s:=s+1;
if s=M+1 then begin Write('Число ітерацій дорівнює s=',s); Break;
end; for i:=1 to N-1 do begin for j:=i+1 to N do begin A[j,i]:=-A[j,i]/A[i,i];
for k:=i+1 to N do begin A[j,k]:=A[j,k]+A[j,i]*A[i,k]; end;
B[j]:=B[j]+A[j,i]*B[i]; end; end; F[N]:=B[N]/A[N,N]; for i:=N-1 downto 1
do
begin h:=B[i]; for j:=i+1 to N do begin h:=h-F[j]*A[i,j]; end; F[i]:=h/A[i,i];
```



```
end; R:=0; for i:=1 to N do begin x[i]:=x[i]+F[i]; if abs(F[i]/x[i]) >e then
R:=1;
end; if R=1 then Continue; WriteLn('Коріння вирішення системи рівнянь');
for i:=1 to N do begin WriteLn('X('i)='x[i]:14:9); end;
WriteLn('Число ітерацій s=',s); Break; until false;END.
```

PROGRAM SU_TR_U_M_N_M;

```
{ Вирішення системи трансцендентних рівнянь  $F_i(X_i)=0$  модифікованим
методом Ньютона}USES Crt; var e,x1,h:Real; N,M,s,i,j,k,R:integer;
A:array[1..50,1..50] of Real; B:array[1..50] of Real; x:array[1..50] of Real;
F:array[1..50] of Real; procedure V;{ Представляємо трансцендентні
рівняння у вигляді  $f_i(X_i)=0$  і підставляємо праві частини отриманих
рівнянь у вирази
 $F(i)=f(X_i)$ } begin F[1]:=x[1]+3*(Ln(x[1]) /Ln(10))-x[2]*x[2];
F[2]:=2*x[1]*x[1]-x[1]*x[2]-5*x[1]+1; end;begin ClrScr;
WriteLn('Задайте число рівнянь'); Write('N='); ReadLn(N);
WriteLn('Задайте максимальне число ітерацій'); Write('M='); ReadLn(M);
WriteLn('Задайте відносну погрішність'); Write('e='); ReadLn(e); s:=0;
WriteLn('Введіть початкові значення'); for i:=1 to N do begin
Write('X,i'(0)='); ReadLn(x[i]); end; repeat V; for i:=1 to N do begin
B[i]:=-F[i]; end; for j:=1 to N do begin
x1:=x[j]; h:=e*abs(x1); x[j]:=x1+h; V; for i:=1 to N do begin
A[i,j]:=(F[i]+B[i])/h; end; x[j]:=x1; end; s:=s+1; if s=M+1 then begin
Write('Число ітерацій дорівнює s=',s); Break; end; for i:=1 to N-1 do
begin for j:=i+1 to N do begin A[j,i]:=-A[j,i]/A[i,i]; for k:=i+1 to N do
begin A[j,k]:=A[j,k]+A[j,i]*A[i,k]; end; B[j]:=B[j]+A[j,i]*B[i]; end; end;
F[N]:=B[N]/A[N,N]; for i:=N-1 downto 1 do begin h:=B[i]; for j:=i+1 to N
do
begin h:=h-F[j]*A[i,j]; end; F[i]:=h/A[i,i]; end; R:=0; for i:=1 to N do
begin
```

```
x[i]:=x[i]+F[i]; if abs(F[i]/x[i]) >e then R:=1; end; if R=1 then Continue;
WriteLn('Коріння вирішення системи рівнянь'); for i:=1 to N do begin
WriteLn('X('i)='x[i]:14:9); end; WriteLn('Число ітерацій s='s); Break;
until false;END.
```

PROGRAM TR_U_M_DI;

```
{ Вирішення трансцендентного рівняння F(X)=0 методом
дихотомії(ділення
відрізання навпіл)}USES Crt; var x,x1,a,b,e,F:real;
n:integer;procedure V;
{ Представляємо трансцендентне рівняння у вигляді F(x)=0 і
підставляємо ліву частину отриманого рівняння у вираз F=f(x)} begin
x1:=a; x:=(a+b)/2;{Ліва частина підставляється із знаком "+" при
аргументі x1=a} F:=x1-cos(x1); if F>=0 then begin{Ліва частина
підставляється із знаком "+" } F:=x-cos(x); end else begin {Ліва частина
підставляється із знаком "-" } F:=-x-cos(x)); end; end;begin ClrScr;
WriteLn('Задайте початкову координату інтервалу'); Write('a=');
ReadLn(a); WriteLn('Задайте кінцеву координату інтервалу'); Write('b=');
ReadLn(b); WriteLn('Задайте погрішність обчислення результату');
Write('e='); ReadLn(e); n:=0; repeat
V; if F>0 then begin a:=x; end else begin b:=x; end; n:=n+1; until b-a<e;
WriteLn('Корінь рівняння рівний'); WriteLn('X=',x:12:9);
WriteLn('Число кроків для визначення кореня рівняння із заданою
точністю');
WriteLn('n=',n);END.
```

PROGRAM TR_U_M_NJ;

```
{ Вирішення трансцендентного рівняння X=F(X) методом Ньютона}USES
Crt; var x0,x,e,F,F1,V3:real; n:integer; procedure V;
{ Представляємо трансцендентне рівняння у вигляді f(x)=0 і
підставляємо ліву частину отриманого рівняння у вираз
```

```
F=f(x)} begin F:=x-cos(x); end; procedure V1;
{ Визначаємо першу похідну рівняння f(x) і підставляємо
у вираз F1=df(x)/dx} begin F1:=1+sin(x); end;begin ClrScr;
WriteLn('Задайте початкове значення кореня рівняння'); Write('Xo=');
ReadLn(x); WriteLn('Задайте погрішність обчислення результату');
Write('e=');
ReadLn(e); n:=0; V; V1; V3:=F/F1; while abs(V3) >=e do begin x:=x-V3;
V; V1; V3:=F/F1; n:=n+1; end; WriteLn('Корінь рівняння рівний');
WriteLn('X=',x:12:9); WriteLn('Число кроків для визначення кореня
рівняння із заданою точністю'); WriteLn('n=',n);END.
```

PROGRAM TR_U_M_N_M;

```
{ Вирішення трансцендентного рівняння F(X)=0 модифікованим методом
Ньютона}USES Crt; var x0,x,e,F,L:real; n:integer; procedure V;
{ Представляємо трансцендентне рівняння у вигляді x=f(x) і
підставляємо праву частину отриманого рівняння у вираз F=f(x)} begin
F:=x-cos(x); end; procedure V1; begin L:=F; x:=x+e; end; procedure V2;
begin L:=e*L/(F-L); x:=x-L-e; end;begin ClrScr;
WriteLn('Задайте початкове значення кореня рівняння'); Write('Xo=');
ReadLn(x);
WriteLn('Задайте погрішність обчислення результату'); Write('e=');
ReadLn(e);
n:=0; V; V1; V; V2; while abs(L) >=e do begin V; V1; V; V2; n:=n+1;
end;
WriteLn('Корінь рівняння рівний'); WriteLn('X=',x:12:9);
WriteLn('Число кроків для визначення кореня рівняння із заданою
точністю');
WriteLn('n=',n);END.
```

PROGRAM TR_U_M_PI;

```
{ Вирішення трансцендентного рівняння  $X=F(X)$  методом простих ітерацій}
USES Crt; var x0,x,e,F:real; n:integer; procedure V;
{ Представляємо трансцендентне рівняння у вигляді  $x=f(x)$  і підставляємо праву частину отриманого рівняння у вираз  $F=f(x)$ } begin F:=sin(x)+0.25; end; procedure V1; begin x:=F; end; begin ClrScr; WriteLn('Задайте початкове значення кореня рівняння'); Write('Xo='); ReadLn(x); WriteLn('Задайте погрішність обчислення результату'); Write('e='); ReadLn(e); n:=0; V; while abs(F-x) >= e do begin V1; V; n:=n+1; end; WriteLn('Корінь рівняння рівний'); WriteLn('X=',x:12:9); WriteLn('Число кроків для визначення кореня рівняння із заданою точністю'); WriteLn('n=',n);END.
```

PROGRAM TR_U_M_PP;

```
{ Визначення всього коріння трансцендентного рівняння  $F(x)=0$  методом порозрядного наближення на інтервалі  $a,b$ }USES Crt; var x,a,b,e,F,h,c:real; k:integer; w:LongInt; procedure V;
{ Представляємо трансцендентне рівняння у вигляді  $F(x)=0$  і підставляємо ліву частину отриманого рівняння у вираз  $F=f(x)$ } begin F:=x-cos(x); end;begin ClrScr; WriteLn('Задайте початкову координату інтервалу'); Write('a='); ReadLn(a); WriteLn('Задайте кінцеву координату інтервалу'); Write('b='); ReadLn(b); WriteLn('Задайте погрішність обчислення результату'); Write('e='); ReadLn(e);
```

```
WriteLn('Задайте крок початкового пошуку'); Write('h='); ReadLn(h); c:=h;
k:=0; x:=a; V; w:=Trunc(F/abs(F)); repeat x:=x+c; if x-c>=b then Break;
V;
if F*w/c>0 then Continue; c:=-c/4; if abs(c)>(e/4) then Continue; k:=k+1;
WriteLn('X(',k)='x:12:8); c:=h; w:=-w; until False;END.
```

PROGRAM DIF_UR_SIS_M_RKM_AV;

```
{ Вирішення системи лінійних диференціальних рівнянь
методом Рунге-кутта-мерсона з автоматичним вибором кроку} USES
Crt; var N,j,d1: integer; x,h,xm,e1,e2,e3: real; y:array[1..50] of real;
w:array[1..50] of real; c:array[1..50] of real; F:array[1..50] of real;
a:array[1..50] of real; d:array[1..50] of real; e:array[1..50] of real;
Procedure V1;
{ Представляємо систему диференціальних рівнянь  $dy_i/dx=fi(x,y_i)$ 
ввиде  $F(i)=fi(x,y)$ } Begin F[1]:=y[1]+y[2]-x*x+x-2; F[2]:=-
2*y[1]+4*y[2]+2*x*x-4*x-7; end;begin ClrScr;
WriteLn('Задайте число диференціальних рівнянь першого порядку');
Write('N='); ReadLn(N); WriteLn('Введіть початкове значення
аргументу');
Write('X0='); ReadLn(x); WriteLn('Введіть кінцеве значення аргументу');
Write('X_MAX='); ReadLn(xm); WriteLn('Задайте погрішність обчислень');
Write('E='); ReadLn(e1); WriteLn('Введіть початкове значення');
for j:=1 to N do begin Write('Y0(',j)='); ReadLn(w[j]); y[j]:=w[j]; end;
WriteLn('Введіть початкове значення кроку інтеграції'); Write('h=');
ReadLn(h); repeat e3:=0; V1; d1:=0; for j:=1 to N do begin a[j]:=F[j]*h;
y[j]:=w[j]+(a[j]/3); end; x:=x+(h/3); V1; for j:=1 to N do begin
y[j]:=w[j]+((a[j]+F[j]*h)/6); end; V1; for j:=1 to N do begin z[j]:=F[j]*h;
y[j]:=w[j]+(a[j]/8)+0.375*c[j]; end; x:=x+(h/6); V1; for j:=1 to N do begin
d[j]:=F[j]*h; y[j]:=w[j]+(a[j]/2)-1.5*c[j]+2*d[j]; end; x:=x+h/2; V1; for
j:=1 to N do begin e[j]:=F[j]*h; y[j]:=w[j]+(a[j]+4*d[j]+e[j])/6; e2:=(abs(-
```

```

2*a[j]+9*c[j]-8*d[j]+e[j]))/30; if e2<=e1 then begin if e2<(e1/20) then
begin d1:=d1+1; end;
end else begin e3:=1; end; end; if e3=0 then begin if d1=N then begin
h:=h+h; end; Write('X=',x:5:2); for j:=1 to N do begin Write('
Y('j')=',y[j]:12:9);
w[j]:=y[j]; end; end else begin x:=x-h; for j:=1 to N do begin y[j]:=w[j];
end; h:=h/2; end; WriteLn; until x>xm;END.

```

PROGRAM DIF_UR_SIS_M_T;

```

{ Вирішення системи лінійних диференціальних рівнянь
методом трапецій} USES Crt; var N,j: integer; x,h,xm: real;
y:array[1..50] of real; w:array[1..50] of real; k:array[1..50] of real;
F:array[1..50] of real; Procedure V;
{ Представляємо систему диференціальних рівнянь  $dy_i/dx=f_i(x,y_i)$ 
ввиде  $F(i)=f_i(x,y)$ } Begin F[1]:=y[2]; F[2]:=(((y[1]/x)-y[2])/x)-y[1]; end;
begin ClrScr;
WriteLn('Задайте число диференціальних рівнянь першого порядку');
Write('N='); ReadLn(N); WriteLn('Введіть початкове значення
аргументу');
Write('X0='); ReadLn(x); WriteLn('Введіть кінцеве значення аргументу');
Write('X_MAX='); ReadLn(xm); WriteLn('Введіть початкове значення');
for j:=1 to N do begin Write('Y0('j')='); ReadLn(w[j]); y[j]:=w[j];
end; WriteLn('Введіть початкове значення кроку інтеграції');
Write('h='); ReadLn(h); repeat V; for j:=1 to N do begin
до[j]:=h*F[j]; y[j]:=w[j]+k[j]; end; x:=x+h; Write('X=',x:5:2); V; for j:=1 to
N do
begin y[j]:=w[j]+((до[j]+h*F[j])/2); Write(' Y('j')=',y[j]:12:9); w[j]:=y[j];
end;
WriteLn; until x>xm;END.

```

PROGRAM DIF_UR_SIS_M_RK4;

```

{ Вирішення системи лінійних диференціальних рівнянь
методом Рунге-Кутта четвертого порядку} USES Crt; var N,j: integer;
x,h,xm,v: real; balon: text; y:array[1..50] of real; w:array[1..50] of real;
k:array[1..50] of real; F:array[1..50] of real; a:array[1..50] of real; Procedure
V1;
{ Представляємо систему диференціальних рівнянь  $dy_i/dx=fi(x,y_i)$ 
ввиде  $F(i)=fi(x,y)$ } Begin F[1]:=(25/100000)*((sin(y[2])
/cos(y[2]))/cos(y[3]))-7*x; F[2]:=(1/y[1])*((sin(y[2])
/cos(y[2]))*(7*x+x*x*(sin(y[3]) /cos(y[3])))+(25/100000)/cos(y[3]));
F[3]:=((7*x*sin(y[3]) -(x*x) -
0.52*cos(y[2]))/(y[1]*cos(y[2])*cos(y[2])*cos(y[3])))-((sin(y[3]) /cos(y[3]))/x);
F[4]:=(sin(y[3]) /cos(y[3]))/x;
F[5]:=(sin(y[2]) /cos(y[2]))/cos(y[3]); end;begin ClrScr;
Assign(balon,'C:\balon.txt'); Rewrite(balon);
WriteLn('Задайте число диференціальних рівнянь першого порядку');
Write('N='); ReadLn(N); WriteLn('Введіть початкове значення аргументу');
Write('X0='); ReadLn(x); WriteLn('Введіть кінцеве значення аргументу');
Write('X_MAX='); ReadLn(xm); WriteLn('Введіть початкове значення');
for j:=1 to N do begin Write('Y0('j')='); ReadLn(w[j]); y[j]:=w[j]; end;
WriteLn('Введіть початкове значення кроку інтеграції'); Write('h=');
ReadLn(h); repeat V1; for j:=1 to N do begin v:=h*F[j]; до[j]:=v;
y[j]:=w[j]+(v/2);
end; x:=x+(h/2); V1; for j:=1 to N do begin v:=h*F[j]; до[j]:=k[j]+2*v;
y[j]:=w[j]+(v/2); end; V1; for j:=1 to N do begin v:=h*F[j]; до[j]:=k[j]+2*v;
y[j]:=w[j]+v; end; x:=x+(h/2); Write('X=',x:6:3); Write(balon,'X=',x:6:3);
V1; for j:=1 to N do begin y[j]:=w[j]+((до[j]+(h*F[j]))/6); Write(balon',
Y('j')=',y[j]:10:5); Write(' Y('j')=',y[j]:10:5); w[j]:=y[j]; end; WriteLn(balon);
WriteLn; until x>xm; Close(balon);END.

```

PROGRAM IN_M_PR;

```
{ Програма для обчислення певного інтеграла методом
прямокутників}
uses Crt; var a,b,m,h,x,y,s,io:real; i:integer; {Завдання підінтегральній
функції} PROCEDURE V; begin y:=sqrt(2*x+1) end;begin ClrScr;
WriteLn('Введення верхньої і нижньої меж інтеграції a,b');
Write('b='); Read(b); Write('a='); Read(a); WriteLn('Задайте число ділянок
розбиття'); Write('m='); Read(m); h:=(b-a)/m; x:=a; s:=0;i:=0; repeat
begin V; s:=s+y; x:=x+h; end; i:=i+1; until i>m-1; io:=s*h;
Write('I=',io:16:8);
END.
```

PROGRAM IN_M_SIM;

```
{ Програма для обчислення певного інтеграла методом
Сімпсона} uses Crt; var a,b,h,x,y,io,s,ia,ib:real; i,m:integer;
{Завдання підінтегральній функції} PROCEDURE V; begin
y:=sqrt(2*x+1)
end;begin ClrScr; WriteLn('Введення верхньої і нижньої меж
інтеграції a,b'); Write('b='); Read(b); Write('a='); Read(a); WriteLn('Задайте
число ділянок розбиття'); Write('m='); Read(m); h:=(b-a)/m; x:=a; V;
ia:=y*h/3; x:=b; V; ib:=y*h/3; s:=0;x:=a;s:=0;i:=1;
repeat begin x:=x+h; V; if Odd(i) then begin s:=s+4*y; end else begin
s:=s+2*y; end; end; i:=i+1 until i>m-1; io:=s*(h/3)+ia+ib;
Write('I=',io:16:8);
END.
```

PROGRAM IN_M_TR;

```
{ Програма для обчислення певного інтеграла методом
трапецій}
```



```

uses Crt; var a,b,h,x,y,io,s,ia,ib:real; i,m:integer; {Завдання
підінтегральній функції} PROCEDURE V; begin y:=sqrt(2*x+1)
end;begin ClrScr;
WriteLn('Введення верхньої і нижньої меж інтеграції a,b'); Write('b=');
Read(b); Write('a='); Read(a);
WriteLn('Задайте число ділянок розбиття'); Write('m='); Read(m); h:=(b-
a)/m; x:=a; V; ia:=y*h/2; x:=b; V; ib:=y*h/2; s:=0;x:=a;s:=0;i:=1; repeat
begin x:=x+h; V; s:=s+y; end; i:=i+1 until i>m-1; io:=s*h+ia+ib;
Write('I=',io:16:8);END.

```

PROGRAM AP_M_NK;

```

{ Апроксимація і інтерполяція результатів досліджень поліномом по
методу найменших квадратів з можливістю автоматичного вибору ступеня
полінома}
{Поліном має вигляд  $y(x)=a_0+a_1*x+a_2*x^2+a_3*x^3+...+a_n*x^n$ } uses Crt;
var
i,H,VV,N,j,k,N1,k1,j1,M,L,M2:integer; E1,R,F,U,S,E,p,Q1:real; X:array[1..50]
of real; Y:array[1..50] of real; Z:array[1..50] of real; B:array[1..50] of real;
G:array[1..50] of real; A:array[1..50,1..50] of real; C:array[1..50,1..50] of real;
D:array[1..100] of real;label M1;PROCEDURE V1V1; begin N:=N+1;
for j:=0 to 2*N-1 do begin if j>N then begin D[j]:=0; end else begin
B[j]:=0;
D[j]:=0; end; end; for i:=1 to H do begin R:=Y[i]; F:=1; for j:=1 to 2*N-1 do
begin if j>N then begin D[j]:=D[j]+F; F:=F*X[i]; end else begin
B[j]:=B[j]+R; R:=R*X[i]; D[j]:=D[j]+F; F:=F*X[i]; end; end; end; for i:=1 to
N do
begin k:=i; for j:=1 to N do begin A[i,j]:=D[k]; k:=k+1; end; end; N1:=N-1;
for k:=1 to N1 do begin if abs(A[до, до]) > 0 then begin end else begin
k1:=k+1;

```

```

for M:=k1 to N do begin if abs(A[M,k])=0 then begin end else begin for L:=1
to N do begin U:=A[k,L]; A[k,L]:=A[M,L]; A[M,L]:=U; end; end; end;
U:=B[k];
B[k]:=B[M]; B[M]:=U;end; G[k]:=B[k]/A[до, до]; k1:=k+1; for i:=k1 to N do
begin B[i]:=B[i]-A[i,k]*G[k]; for j1:=k to N do begin j:=N-j1+k;
C[k,j]:=A[k,j]/A[до, до]; A[i,j]:=A[i,j]-A[i,k]*C[k,j]; end; end; end;
M:=N; Z[M]:=B[M]/A[M,M]; repeat M:=M-1; S:=0; for L:=M to N1 do
begin S:=S+C[M,L+1]*Z[L+1]; end; Z[M]:=G[M]-S; until M<=1; E:=0;
for i:=1 to H do begin S:=Y[i]; R:=1; for j:=1 to N do begin S:=S-R*Z[j];
R:=R*X[i]; end; E:=E+S*S; end; E:=sqrt(E/H); end;PROCEDURE V1V2;
begin WriteLn('Ступінь полінома N= ',N-1:4); WriteLn('Погрішність
результатів E=',e); WriteLn('Коефіцієнти полінома'); for i:=1 to N do
begin WriteLn('A(',i-1)=' ,Z[i]:14:5); end; end;PROCEDURE V1V3; begin
WriteLn('Ступінь полінома N= ',N-1:4); WriteLn('Погрішність результатів
E=',e);
WriteLn('Якщо E більше заданою, збільшуємо N і повторюємо
обчислення');
WriteLn('Коефіцієнти полінома'); for i:=1 to N do begin WriteLn('A(',i-
1)=' ,Z[i]:14:5); end; end;begin ClrScr;
Write('Задайте число вузлів інтерполяції або апроксимації'); Write(' N=');
Read(N); WriteLn('Введення значень X(i) і Y(i) для кожного вузла ');
for i:=1 to H do begin Write('X(',i)='); Read(X[i]);
Write('Y(',i)=');Read(Y[i]);
end; N:=1; WriteLn('Введіть 0 - якщо задана погрішність апроксимації');
WriteLn(' 1 - якщо задається ступінь полінома'); Read(VV); if VV=0
then
begin WriteLn('Задайте середньоквадратичну погрішність');Write('E1=');
Read(E1); end else begin WriteLn('Введіть ступінь полінома M<N');

```

```

Write('M='); Read(N); end; V1V1; if VV=1 then begin V1V2; end else
begin
M1: if E<=E1 then begin V1V3; end else begin if N<H then begin V1V1;
goto M1; end else begin V1V3; end; end; end;
WriteLn('Перевірочне обчислення величини Y(X)'); Write('Введіть
значення X='); Read(p); Q1:=0; for i:=N downto 2 do begin
Q1:=(Q1+Z[i])*p;
end; Q1:=Q1+Z[1]; Write('Значення Y(X)=',Q1:14:5);END.

```

Програмне забезпечення кінематичного аналізу і синтезу механізмів і машин легкої промисловості.

```

program newton (input,output);{$i graph.p}{$i intg }{$i rea }label
10,20,30,40;type t506=array [1..50,1..6] of real;t503=array [1..50,1..3]
ofreal;t50=array [1..50] of real;t6=array [1..6] of real;t3=array [1..3]of
real;t2=array [1..2] of real;ti50=array [1..50] of integer;t=integer; var mu:array
[1..8] of integer;infileK,oufileK:text;fv:file;dg:array [1..200] of
byte;mexanzname:string[60];   infilenamK:string[14];
oufilenamK:string[14];   q:t506;p:t503;g1:t50;
a,b,c,d,e:t6;a1,p1,p2,p4:t3;p3,p5,p6:t2;jg,ja,jb,jc,jd,je,prinK:ti50;   i,j,i1,i2:t;
as,Hag,ugpre,u,z,sg,x0,y0,x1,y1,x2,y2,mgr,AX,AY,   cs:real;
error,kinem,filed,n,nd,nn,menuK,menuDel,element,Rkin,Nprin,Khag,ACPU,
menuKin,m,Rerror,l,page,outwrite,MaxN,mult,x,y,ii,jj,cont,z0,color,
kursor,sp,am,bm,xm,cia,cv,   stoik:integer;   OK,eoj:boolean;   ch:char;
overlay procedure gr1;   begin
textbackground(6);window(19,14,65,24);clrscr; repeat
textbackground(7);textcolor(0);window(17,13,63,23);clrscr;writeln; writeln('
1. Номера кинематических'); write('   пар для <A> и <B>   :
');ja[nn]:=intg(ja[nn]);write(' ');   jb[nn]:=intg(jb[nn]);writeln; write('   2.
Абсцисса стойки   ',ja[nn]:2,'   ');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;

```

```

write(' 3. Ордината стійки ',ja[nn]:2,' :
');q[ja[nn],2]:=rea(q[ja[nn],2]);writeln; write(' 4. Довжина ланки
',ja[nn]:2,'-',jb[nn]:2,' : '); p[nn,1]:=rea(p[nn,1]);writeln;writeln(' 5.
Початкове положення'); write(' кривошипа (у град.) :
');p[nn,2]:=rea(p[nn,2]);writeln; write(' 6. Кутова швидкість (с-1) :
');p[nn,3]:=rea(p[nn,3]);writeln;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write(' (Так-1/Нет-0)
: ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr; writeln(' Змініте помилкові
дані !'); write(' Для вірних даних - <ENTER>');read;end; until error=0; end;
overlay procedure gr2; begin
textbackground(6);window(19,17,65,24);clrscr; repeat
textbackground(7);textcolor(0);window(17,16,63,23);clrscr; writeln(' 1.
Номерів кінематичних'); write(' пара для <A>, <B>, <C> : ');
ja[nn]:=intg(ja[nn]);write(' ');jb[nn]:=intg(jb[nn]);write('
');jc[nn]:=intg(jc[nn]);writeln; write(' 2. Довжина ланки ',ja[nn]:2,'-
',jb[nn]:2,' : '); p[nn,1]:=rea(p[nn,1]);writeln; write(' 3. Довжина ланки
',jb[nn]:2,'-',jc[nn]:2,' : '); p[nn,2]:=rea(p[nn,2]);writeln; write(' 4. Знак
схеми збірки (-1/+1): ');p[nn,3]:=rea(p[nn,3]);writeln; writeln(' 5.
Позначення стійки <','ja[nn]:2,'>', ' '); write(' або <','jc[nn]:2,'> . Немає --
> 0 : ');stoik:=intg(stoik);writeln; if stoik=ja[nn] then begin
clrscr;writeln;writeln;q[jc[nn],3]:=1.; write(' 6. Абсциса стійки ',ja[nn]:2',
: ');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln; write(' 7. Ордината стійки
',ja[nn]:2,' : ');q[ja[nn],2]:=rea(q[ja[nn],2]);writeln; end;if stoik=jc[nn] then
begin clrscr;writeln;writeln;q[ja[nn],3]:=1.; write(' 6. Абсциса стійки
',jc[nn]:2,' : ');q[jc[nn],1]:=rea(q[jc[nn],1]);writeln; write(' 7. Ордината
стіжки ',jc[nn]:2,' : ');q[jc[nn],2]:=rea(q[jc[nn],2]);writeln;end; if stoik=0

```

```

then begin q[ja[nn],3]:=1.0;q[jc[nn],3]:=1.0;end;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных - <ENTER>');read;end;
until error=0; end; overlay procedure gr3; begin
textbackground(6);window(13,19,71,24);clrscr; repeat
textbackground(7);textcolor(0);window(11,18,69,23);clrscr; writeln(' 1.
Выбрать координаты точек <D> и <E> так чтобы положение'); writeln('
направляющей определяло положение шатуна АВ :'); writeln(' - для АВС
и АВ1С1 - DE ; - для АВ2С2 и АВ3С3 - DE1 .'); writeln(' 2. Выбрать знак
для поводка ВС по скалярному произведению'); write(' DE и ВС : для СВ
и С2В2 { + }; для С1В1 и С3В3 { - }.'); write(' Выбор сделан
');textcolor(0+16);write('?');textcolor(0); write(' "Да" - введите < ENTER
>');read;clrscr; writeln(' 3. Номера кинематических пар: <A> <B> <C>
<D> <E>'); write(' ');ja[nn]:=intg(ja[nn]);write(' ');
jb[nn]:=intg(jb[nn]); write(' ');jc[nn]:=intg(jc[nn]);write('
');jd[nn]:=intg(jd[nn]);write(' ');je[nn]:=intg(je[nn]);writeln; write(' 4. Длина
звена 'ja[nn]:2,-',jb[nn]:2,' : '); p[nn,1]:=rea(p[nn,1]);writeln; write('
5. Длина поводка 'jb[nn]:2,-',jc[nn]:2,' (со знаком) : ');
p[nn,2]:=rea(p[nn,2]);writeln; write(' 6. Направляющая 'jd[nn]:2,-',je[nn]:2,'
неподвижна '); textcolor(0+16);writeln('? ');textcolor(0); write(' (Да-
1/Нет-0) : ');stoik:=intg(stoik);if stoik=1 then begin clrscr;write(' 7.
Координаты точек : <'jd[nn]:2,> по оси OX : ');
q[jd[nn],1]:=rea(q[jd[nn],1]);writeln; write(' <'je[nn]:2,> по
оси OX : '); q[je[nn],1]:=rea(q[je[nn],1]);writeln; write('
<'jd[nn]:2,> по оси OY : '); q[jd[nn],2]:=rea(q[jd[nn],2]);writeln; write('

```

```

<',je[nn]:2,> по оси OY : '); q[je[nn],2]:=rea(q[je[nn],2]);writeln;end else
begin q[jd[nn],3]:=1.;q[je[nn],3]:=1.;end;
textbackground(4);textcolor(7);window(13,20,35,21);clrscr; write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(15,21,43,22);clrscr;
textbackground(4);window(13,20,41,21);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных - <ENTER>');read;end;
until error=0; end; overlay procedure gr4; begin
textbackground(6);window(13,20,71,24);clrscr; repeat
textbackground(7);textcolor(0);window(11,19,69,23);clrscr; writeln(' 1.
Номера кинематических пар: <A> <B> <C> <D> <E>'); write('
');ja[nn]:=intg(ja[nn]);write(' '); jb[nn]:=intg(jb[nn]); write('
');jc[nn]:=intg(jc[nn]);write(' ');jd[nn]:=intg(jd[nn]);write(' ');
je[nn]:=intg(je[nn]);writeln;clrscr; write(' 2. Длина плеча АЕ кулисы
',ja[nn]:2,'-',je[nn]:2,' (со знаком) : '); p[nn,1]:=rea(p[nn,1]);writeln; write(' 3.
Длина поводка ВС ',jb[nn]:2,'-',jc[nn]:2,' (со знаком) : ');
p[nn,2]:=rea(p[nn,2]);writeln; write(' 4. Направляющая ED кулисы ',je[nn]:2,'-
',jd[nn]:2,' (со знаком) : '); p[nn,3]:=rea(p[nn,3]);writeln; writeln(' 5.
Обозначение стойки <',ja[nn]:2,> или <',jc[nn]:2,> ,'); write(' если в
группе стойка отсутствует ---> 0 : ');stoik:=intg(stoik); if stoik=ja[nn] then
begin clrscr; write(' 6. Абсцисса стойки ',ja[nn]:2,' : ');
q[ja[nn],1]:=rea(q[ja[nn],1]);writeln; write(' 7. Ордината стойки
',ja[nn]:2,' : '); q[ja[nn],2]:=rea(q[ja[nn],2]);q[jc[nn],3]:=1.;writeln; end;if
stoik=jc[nn] then begin clrscr; write(' 6. Абсцисса стойки ',jc[nn]:2,'
: '); q[jc[nn],1]:=rea(q[jc[nn],1]); write(' 7. Ордината стойки
',jc[nn]:2,' : '); q[jc[nn],2]:=rea(q[jc[nn],2]);q[ja[nn],3]:=1.;writeln;end; if
stoik=0 then begin q[ja[nn],3]:=1.0;q[jc[nn],3]:=1.0;end;
textbackground(4);textcolor(7);window(13,20,35,21);clrscr; write('

```

```

Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(15,21,43,22);clrscr;
textbackground(4);window(13,20,41,21);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных - <ENTER>');read;end;
until error=0; end; overlay procedure gr5; begin
textbackground(6);window(19,18,65,23);clrscr; repeat
textbackground(7);textcolor(0);window(17,17,63,22);clrscr; writeln(' 1. Номера
кинематических'); write(' пар для <A>, <B>, <C> : ');
ja[nn]:=intg(ja[nn]);write(' ');jb[nn]:=intg(jb[nn]);write('
');jc[nn]:=intg(jc[nn]);writeln; write(' 2. Длина звена ',ja[nn]:2,'-',jb[nn]:2,'
: ');p[nn,1]:=rea(p[nn,1]);writeln; writeln(' 3. Угол между плечами АВ и АС');
write(' в градусах (со знаком) : ');p[nn,2]:=rea(p[nn,2]);writeln;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных - <ENTER>');read;end;
until error=0; end; overlay procedure gr6; begin
textbackground(6);window(17,18,67,23);clrscr; repeat
textbackground(7);textcolor(0);window(15,17,65,22);clrscr; writeln(' 1. Номера
кинематических пар <A> <B> <D> <E>'); write('
');ja[nn]:=intg(ja[nn]);write(' '); jb[nn]:=intg(jb[nn]);write('
');jd[nn]:=intg(jd[nn]);write(' ');je[nn]:=intg(je[nn]);writeln; write(' 2. Длина
звена ',ja[nn]:2,'-',jb[nn]:2,' : ');p[nn,1]:=rea(p[nn,1]);writeln; writeln(' 3.
Угол между прямыми АВ и ED'); write(' в градусах (со знаком) :
');p[nn,2]:=rea(p[nn,2]);
textbackground(4);textcolor(7);window(16,19,38,20);clrscr; write('

```

```

Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write(
(Да-1/Нет-0) : ');error:=(error);if error=1 then begin
textbackground(1);window(18,20,46,21);clrscr;
textbackground(4);window(16,19,44,20);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных - <ENTER>');read;end;
until error=0; end; overlay procedure d1; begin gotoxy(24,3);
writeln('ГРУППА ПЕРВОГО ВИДА'); write(' | Порядковый номер
группы равен : ');textbackground(7); textcolor(0);writeln(' ',nd:2,'
');textbackground(1);textcolor(7);writeln; if element=1 then begin write(' | 1.
Длина звена ',ja[nd]:2,'-',jb[nd]:2); write(' : ');
writeln(p[nd,1]:7:2); write(' | 2. Начальное положение кривошипа (в град.)
: '); writeln(p[nd,2]:7:2); write(' | 3. Угловая скорость (с-1) . : ');
writeln(p[nd,3]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(52,8);p[nd,3]:=rea(p[nd,3]);end else begin gotoxy(15,6);write('1.
Абсцисса стойки ',ja[nd]:2,' : '); writeln(q[ja[nd],1]:7:2);
gotoxy(15,7);write('2. Ордината стойки ',ja[nd]:2,' : ');
writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]); end;end; overlay procedure d2;
begin gotoxy(24,3); writeln('ГРУППА ВТОРОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2,'-',jb[nd]:2);
write(' : '); writeln(p[nd,3]:4:1); textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(53,8);p[nd,3]:=rea(p[nd,3]); end else begin as:=0.;cs:=0.;for i:=3 to 6
do begin as:=as+q[ja[nd],i]; cs:=cs+q[jc[nd],i];end;if as=0.0 then begin
gotoxy(15,6);write('1. Абсцисса стойки ',ja[nd]:2,' : ');

```



```
writeln(q[ja[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки    ',ja[nd]:2,'
: '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]);end else if cs=0.0 then begin
gotoxy(15,6);write('1. Абсцисса стойки    ',jc[nd]:2,'    ');
writeln(q[jc[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки    ',jc[nd]:2,'
: '); writeln(q[jc[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[jc[nd],1]:=rea(q[jc[nd],1]);
gotoxy(49,7);q[jc[nd],2]:=rea(q[jc[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln; end;end; end; overlay procedure d3; begin
gotoxy(23,3); writeln('ГРУППА ТРЕТЬЕГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' |    1. Длина звена    ',ja[jg[nd]]:2,'-
',jb[jg[nd]]:2); write('    '); writeln(p[nd,1]:7:2); write(' |    2. Длина
звена (со знаком) ', jb[nd]:2,'-',jc[nd]:2);write('    ');writeln(p[nd,2]:7:2);
textbackground(7);textcolor(0); gotoxy(55,6);readln(p[nd,1]);
gotoxy(55,7);readln(p[nd,2]); end else begin as:=0.;for i:=3 to 6 do begin
as:=as+q[je[nd],i];end; if as>=0.0 then begin gotoxy(15,6);write('1. Абсцисса
стойки    ',jd[nd]:2,'    '); writeln(q[jd[nd],1]:7:2); gotoxy(15,7);write('2.
Ордината стойки    ',jd[nd]:2,'    '); writeln(q[jd[nd],2]:7:2);
gotoxy(15,8);write('3. Абсцисса стойки    ',je[nd]:2,'    ');
writeln(q[je[nd],1]:7:2); gotoxy(15,9);write('4. Ордината стойки    ',je[nd]:2,'
: '); writeln(q[je[nd],2]:7:2); textbackground(7);textcolor(0);
gotoxy(49,6);q[jd[nd],1]:=rea(q[jd[nd],1]);
gotoxy(49,7);q[jd[nd],2]:=rea(q[jd[nd],2]);
gotoxy(49,8);q[je[nd],1]:=rea(q[je[nd],1]);
gotoxy(49,9);q[je[nd],2]:=rea(q[je[nd],2]);end else begin
```

```

gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln; end;end; end; overlay procedure d4; begin
gotoxy(22,3); writeln('ГРУППА ЧЕТВЕРТОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2, ' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина плеча кулисы ',ja[nd]:2,'-
',je[nd]:2,' (со знаком) : '); writeln(p[nd,1]:7:2); write(' | 2. Длина поводка
',jb[nd]:2,'-',jc[nd]:2,' (со знаком) : '); writeln(p[nd,2]:7:2); write(' | 3.
Направляющая кулисы ',je[nd]:2,'-',jd[nd]:2,' (со знаком) : ');
writeln(p[nd,3]:7:2); textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(52,7);p[nd,3]:=rea(p[nd,3]); end else begin as:=0.;cs:=0.;for i:=3 to 6
do begin as:=as+q[ja[nd],i]; cs:=cs+q[jc[nd],i];end;if as>=0.0 then begin
gotoxy(15,6);write('1. Абсцисса стойки ',ja[nd]:2, ' : ');
writeln(q[ja[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки ',ja[nd]:2,
: '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]);end else if cs=0.0 then begin
gotoxy(15,6);write('1. Абсцисса стойки ',jc[nd]:2, ' : ');
writeln(q[jc[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки ',jc[nd]:2,
: '); writeln(q[jc[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[jc[nd],1]:=rea(q[jc[nd],1]);
gotoxy(49,7);q[jc[nd],2]:=rea(q[jc[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln; end;end; end; overlay procedure d5; begin
gotoxy(24,3); writeln('ГРУППА ПЯТОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2, ' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2,'-',jb[nd]:2,

```

```

: '); writeln(p[nd,1]:7:2); writeln(' | 2. Угол между плечами ',ja[nd]:2,'-
',jb[nd]:2,' и ', ja[nd]:2,'-',jc[nd]:2); write(' | в градусах (со знаком)
: '); writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end
else begin gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В
ГРУППЕ НЕТ СТОЕК ! ');readln;end; overlay procedure d6; begin
gotoxy(24,3); writeln('ГРУППА ШЕСТОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2,'-',jb[nd]:2,'
: '); writeln(p[nd,1]:7:2); writeln(' | 2. Угол между прямыми ',ja[nd]:2,'-
',jb[nd]:2,' и ', jd[nd]:2,'-',je[nd]:2); write(' | в градусах (со знаком)
: '); writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end
else begin gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В
ГРУППЕ НЕТ СТОЕК ! ');readln;end; end; overlay procedure v1; begin {
ГРУППА ПЕРВОГО ВИДА } write(' | Порядокковый номер группы
равен : ');textbackground(7); textcolor(0);writeln(' ',i1:2,'
');textbackground(1);textcolor(7);writeln; write(' | 1. Длина звена
',ja[i1]:2,'-',jb[i1]:2); write(' : '); writeln(p[i1,1]:7:2); write(' | 2.
Начальное положение (в град.) : '); writeln(p[i1,2]:7:2); write(' | 3.
Угловая скорость (с-1) : '); writeln(p[i1,3]:7:2); write(' | 4.
Абсцисса стойки ',ja[i1]:2,' : '); writeln(q[ja[i1],1]:7:2); write(' | 5.
Ордината стойки ',ja[i1]:2,' : '); writeln(q[ja[i1],2]:7:2); end; overlay
procedure v2; begin { ГРУППА ВТОРОГО ВИДА } write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',i1:2,' ');textbackground(1);textcolor(7);writeln; write(' |
1. Длина звена ',ja[i1]:2,'-',jb[i1]:2); write(' : ');
writeln(p[i1,1]:6:2); write(' | 2. Длина звена ', jb[i1]:2,'-

```

```

',jc[i1]:2);write('      : ');writeln(p[i1,2]:6:2); write(' | 3. Знак схеми
сборки группы (-1/+1)      : '); writeln(p[i1,3]:4:1); begin as:=0.;cs:=0.;for
i:=3 to 6 do begin as:=as+q[ja[i1],i]; cs:=cs+q[jc[i1],i];end;if as=0.0 then begin
write(' | 4. Абсцисса стойки 'ja[i1]:2,' : '); writeln(q[ja[i1],1]:7:2);
write(' | 5. Ордината стойки 'ja[i1]:2,' : ');
writeln(q[ja[i1],2]:7:2);end else if cs=0.0 then begin write(' | 4. Абсцисса
стойки 'jc[i1]:2,' : '); writeln(q[jc[i1],1]:7:2); write(' | 5. Ординат
стойки 'jc[i1]:2,' : '); writeln(q[jc[i1],2]:7:2);end;end; end; overlay
procedure v3; begin { ГРУППА ТРЕТЬЕГО ВИДА } write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ,i1:2,' ');textbackground(1);textcolor(7);writeln; write(' |
1. Длина звена 'ja[i1]:2,'-',jb[i1]:2); write(' : ');
writeln(p[i1,1]:7:2); write(' | 2. Длина звена (со знаком) ', jb[i1]:2,'-
',jc[i1]:2);write(' : ');writeln(p[i1,2]:7:2); begin as:=0.;for i:=3 to 6 do begin
as:=as+q[je[i1],i];end; if as=0.0 then begin write(' | 3. Абсцисса стоек
',jd[i1]:2,' и ',je[i1]:2,' : '); write(q[jd[i1],1]:7:2);write(' и
');writeln(q[je[i1],1]:7:2); write(' | 4. Ордината стоек 'jd[i1]:2,' и
',je[i1]:2,' : '); write(q[jd[i1],2]:7:2);write(' и ');writeln(q[je[i1],2]:7:2);end;end;
end; overlay procedure v4; begin { ГРУППА ЧЕТВЕРТОГО ВИДА }
write(' |      Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ,i1:2,' ');textbackground(1);textcolor(7);writeln; write(' |
1. Длина плеча кулисы 'ja[i1]:2,'-',je[i1]:2,' (со знаком) : ');
writeln(p[i1,1]:7:2); write(' | 2. Длина поводка 'jb[i1]:2,'-',jc[i1]:2,' (со
знаком) : '); writeln(p[i1,2]:7:2); write(' | 3. Направляющая кулисы
',je[i1]:2,'-',jd[i1]:2,' (со знаком) : '); writeln(p[i1,3]:7:2); begin
as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[i1],i]; cs:=cs+q[jc[i1],i];end;if
as=0.0 then begin write(' | 4. Абсцисса стойки 'ja[i1]:2,' : ');
writeln(q[ja[i1],1]:7:2); write(' | 5. Ордината стойки 'ja[i1]:2,' : ');
writeln(q[ja[i1],2]:7:2); end else if cs=0.0 then begin write(' | 4. Абсцисса

```

```

стойки 'jc[i1]:2,' :'); writeln(q[jc[i1],1]:7:2); write(' | 5. Ордината
стойки 'jc[i1]:2,' :'); writeln(q[jc[i1],2]:7:2); end;end; end; overlay
procedure v5; begin { ГРУППА ПЯТОГО ВИДА } write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ,i1:2,' ');textbackground(1);textcolor(7);writeln; write(' |
1. Длина звена 'ja[i1]:2,-',jb[i1]:2,' :'); writeln(p[i1,1]:7:2);
writeln(' | 2. Угол между плечами 'ja[i1]:2,-',jb[i1]:2,' и ', ja[i1]:2,-
',jc[i1]:2); write(' | в градусах (со знаком) : ');
writeln(p[i1,2]:7:2); end; overlay procedure v6; begin { ГРУППА
ШЕСТОГО ВИДА } write(' | Порядковый номер группы равен :
');textbackground(7); textcolor(0);writeln(' ,i1:2,'
');textbackground(1);textcolor(7);writeln; write(' | 1. Длина звена
',ja[i1]:2,-',jb[i1]:2,' :'); writeln(p[i1,1]:7:2); writeln(' | 2. Угол
между прямыми 'ja[i1]:2,-',jb[i1]:2,' и ', jd[i1]:2,-',je[i1]:2); write(' | в
градусах (со знаком) : '); writeln(p[i1,2]:7:2); end; overlay
procedure graf1; begin textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,3,70,9);clrscr;
textbackground(3);window(9,2,68,8);clrscr;textcolor(0);writeln; writeln('
РЕЖИМ ГРАФИЧЕСКОГО ИССЛЕДОВАНИЯ :'); write(' 1.
Построение ');textbackground(7);textcolor(0); write(' ФУНКЦИИ
ПОЛОЖЕНИЯ ');textbackground(3);textcolor(0); writeln(' точек
механизма. '); write(' 2. Построение ');textbackground(7);textcolor(0);
write(' СКОРОСТИ ');textbackground(3);textcolor(0); writeln(' точек
механизма. '); write(' 3. Построение ');textbackground(7);textcolor(0);
write(' УСКОРЕНИЯ ');textbackground(3);textcolor(0); writeln(' точек
механизма. '); write(' Номер ПОЗИЦИИ : ');
textbackground(7);window(10,3,16,7);clrscr;textcolor(4); write(' актив. ');for
i:=1 to Rkin-1 do writeln;textcolor(0+16); write(' ---->');
textbackground(3);window(48,7,68,8);clrscr;textbackground(7);

```

```

textcolor(0);gotoxy(1,1);write(' ');gotoxy(2,1);Rkin:=intg(Rkin); end;
overlay procedure graf2; begin textbackground(1);
window(9,12,74,23);clrscr;textbackground(3);textcolor(0);
window(7,11,72,22);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 11 do begin
gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;gotoxy(3,11);for i:=1 to 62 do
write('_'); gotoxy(20,10);textbackground(7);textcolor(0); writeln(' Нет
корректировки - <ENTER>');textbackground(3);textcolor(0); gotoxy(22,3);
writeln(' ИСХОДНЫЕ ДАННЫЕ :');writeln; gotoxy(8,5);write('1. Общее
');textcolor(0);write('КОЛИЧЕСТВО ТОЧЕК'); textcolor(0);write(' для
исследования :'); writeln(Nprin:2); gotoxy(8,6);write('2.
');textcolor(0);write('ШАГ');textcolor(0); write(' расчета кинематических
характеристик :'); writeln(Наg:5:1); gotoxy(8,7);write('3. Общее
');textcolor(0);write('ЧИСЛО ШАГОВ'); textcolor(0);write(' исследования
:'); writeln(Кhag:2);textcolor(0); gotoxy(8,8);write('4. Инициализация
режима');textcolor(0); write(' МУЛЬТИПЛИКАЦИИ');
textcolor(0);write('(0..9) : '); writeln(mult:1);textbackground(7);textcolor(0);
gotoxy(55,5);Nprin:=intg(Nprin); gotoxy(56,6);Наg:=rea(Наg);
gotoxy(55,7);Кhag:=intg(Кhag); gotoxy(56,8);mult:=intg(mult); end;
overlay procedure graf3; begin textbackground(5);window(1,1,80,25);clrscr;
textbackground(1); window(9,7,74,18);clrscr;textbackground(3);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(7);textcolor(0); writeln(' Нет корректировки -
<ENTER>');textbackground(3);textcolor(0); gotoxy(18,3);
writeln('ВВЕДИТЕ НОМЕРА ИССЛЕДУЕМЫХ ТОЧЕК');writeln;
i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then begin
i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;

```

```

textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then begin
i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end; for i:=Nprin+1
to 50 do prinK[i]:=0; end; overlay procedure graf4; begin if Rkin=1 then
begin textbackground(5);window(1,1,80,25);clrscr; textbackground(2);
window(9,7,74,18);clrscr;textbackground(7);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0); writeln(' Нет корректировки -
<ENTER>');textbackground(7);textcolor(0); gotoxy(4,3); writeln('
ОБЛАСТЬ ИССЛЕДОВАНИЯ ФУНКЦИИ ПОЛОЖЕНИЯ ТОЧЕК
МЕХАНИЗМА');writeln; gotoxy(8,5);write('1. Абсцисса ЛЕВОЙ нижней
вершины области : '); writeln(x0:7:2); gotoxy(8,6);write('2. Ордината
ПРАВОЙ нижней вершины области : '); writeln(y0:7:2);
gotoxy(8,7);write('3. ГРАНИЦА области исследования (Ox - Oy) : ');
writeln(sg:7:2); gotoxy(8,8);write('4. Описание работы КУРСОРОМ (Да-
1/Нет-0) : '); writeln(kursor:1); textbackground(7);textcolor(0);
gotoxy(52,5);x0:=rea(x0); gotoxy(52,6);y0:=rea(y0);
gotoxy(52,7);sg:=rea(sg); gotoxy(55,8);kursor:=intg(kursor); end; end;
overlay procedure graf5; begin if Rkin>1 then begin
textbackground(5);window(1,1,80,25);clrscr; textbackground(2);
window(9,7,74,18);clrscr;textbackground(7);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0); writeln(' Нет корректировки -
<ENTER>');textbackground(7);textcolor(0); gotoxy(7,3); writeln('
ИССЛЕДОВАНИЕ СКОРОСТИ И УСКОРЕНИЯ ТОЧЕК
МЕХАНИЗМА');writeln; gotoxy(8,5);write('1. Определите ДИАПОЗОН

```

```

области исследования '); gotoxy(8,6);write(' скорости или ускорения
точек механизма. '); gotoxy(8,7);write(' В первом приближении
ДИАПОЗОН равен : '); if Rkin=2 then sg:=2*p[1,3]*p[1,1]/z else
sg:=2*p[1,3]*p[1,3]*p[1,1]/z; writeln(sg:7:2); gotoxy(8,8);write('2.
Описание работы КУРСОРОМ (Да-1/Нет-0) : '); writeln(kursor:1);
textbackground(7);textcolor(0); gotoxy(52,7);sg:=rea(sg);
gotoxy(55,8);kursor:=intg(kursor); end; end; overlay procedure graf6;
begin if kursor>0 then begin repeat
clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');
textbackground(3);textcolor(0+16);gotoxy(29,18);write('<ENTER>');
textbackground(1);textcolor(7); gotoxy(19,4);writeln('ОПИСАНИЕ
УПРАВЛЕНИЯ КУРСОРОМ'); gotoxy(14,6);writeln('1 - перемещение по
диагонали вниз влево'); gotoxy(14,7);writeln('2 - перемещение вниз ');
gotoxy(14,8);writeln('3 - перемещение по диагонали вниз вправо');
gotoxy(14,9);writeln('4 - перемещение влево'); gotoxy(14,10);writeln('6 -
перемещение вправо'); gotoxy(14,11);writeln('7 - перемещение по
диагонали вверх влево'); gotoxy(14,12);writeln('8 - перемещение вверх');
gotoxy(14,13);writeln('9 - перемещение по диагонали вверх вправо');
gotoxy(14,14);writeln('0 - маркер начала участка исследования');
gotoxy(14,15);writeln('5 - маркер конца участка исследования');
gotoxy(14,16);writeln('+ - установка дополнительных осей'); read;for ii:=6
to 18 do begin gotoxy(14,ii); writeln(' ');end;
gotoxy(14,6);writeln('S - останов режима "построение графика"');
gotoxy(14,7);writeln('P - включение режима "построение графика"');
gotoxy(14,8);writeln('V - включение/выключение следа на экране');

```



```

gotoxy(14,9);writeln('Q - желтая палитра графики');
gotoxy(14,10);writeln('W - белая палитра графики');
gotoxy(14,11);writeln('B - быстрое передвижение курсора по экрану');
gotoxy(14,12);writeln('M - пошаговое передвижение курсора по экрану');
gotoxy(14,13);writeln('N - ВЫХОД из графического режима исследования');
textbackground(3);textcolor(0);gotoxy(16,16); writeln(' ПОВТОРИТЬ ? (Нет -
введите : N) '); read(kbd,ch);until ch=#110; end; {конец внутрен. цикла }
end; overlay procedure indate; begin repeat
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(0); writeln('
_____ '); writeln(' Введите массив исходных ');
writeln(' данных по механизму'); write(' имя файла :
');textbackground(7);textcolor(0);write(' ');
gotoxy(23,4);readln(infilenamK); assign(infileK,infilenamK); {$I-}
reset(infileK);{$I+} ОК :=(IOresult=0); if not ОК then begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(27,11,57,15);clrscr;
textbackground(4);window(25,10,55,14);clrscr;textcolor(7); writeln;
writeln(' Файл исходных данных'); write(' не найден -
');textcolor(0+16);writeln('ОШИБКА !'); textcolor(7); write(' Повторите
ввод <ENTER>');readln;end;until ОК; end; overlay procedure outdate;
begin repeat textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(0); writeln('
_____ '); writeln(' Введите имя файла для ');
writeln(' массива ВЫХОДНЫХ данных '); write(' имя файла :
');textbackground(7);textcolor(0); write(' ');
gotoxy(23,4);readln(oufilenamK); assign(oufileK,oufilenamK); {$I-}

```

```

reset(oufileK);{$I+}   ОК :=(IOresult=0);   if not ОК then outwrite:=1 else
begin textbackground(5);
window(1,1,80,25);clrscr;textbackground(1);window(27,11,57,15);clrscr;
textbackground(4);window(25,10,55,14);clrscr;textcolor(7);   writeln;
writeln('   Имя файла данных ');   write(' СУЩЕСТВУЕТ -
');textcolor(0+16);writeln('ПЕРЕЗАПИСЬ ?');   textcolor(7);   write(' (Да-
1/Нет-0) КОМАНДА:');outwrite:=intg(outwrite);end;until outwrite>0;   end;
overlay procedure inname;   begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;
textbackground(2);window(9,10,72,14);clrscr;textcolor(0);   writeln('
_____ ');   writeln(' *
Наименование исследуемого механизма машины * ');   writeln(' -----
----- ');   write('
');textbackground(7);textcolor(0);   for i:=1 to 60 do write(' ');if
mexanzname='' then begin   gotoxy(14,4);write('НОВЫЙ исследуемый
механизм');textcolor(0+16);   write(' <ENTER>');textcolor(0);end else
begin
gotoxy(3,4);write(mexanzname:length(mexanzname));end;read(kbd,ch);   if
ch=#26 then cia:=0;   end;   overlay procedure outname;   begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;
textbackground(2);window(9,10,72,14);clrscr;textcolor(0);   writeln('
_____ ');   writeln('
Введите наименование исследуемого механизма машины');   writeln('
-----');   write('
');textbackground(7);textcolor(0);   for i:=1 to 60 do write('
');gotoxy(3,4);write(mexanzname:length(mexanzname));   read(kbd,ch);if
ch<>#13 then begin gotoxy(3,4);for i:=1 to 60 do write(' ');gotoxy(3,4);

```

```

read(mexanzname);                                end;    end;    overlay procedure
tailt;    begin    graphcolormode;clearscreen;
palette(2);graphbackground(1);    clearscreen;
draw(0,0,319,0,3);draw(319,0,319,199,3);
draw(319,199,0,199,3);draw(0,199,0,0,3);
draw(3,3,316,3,3);draw(316,3,316,196,3);
draw(316,196,3,196,3);draw(3,196,3,3,3);
draw(40,190,280,190,3);draw(280,190,280,40,3);
draw(280,40,270,30,3);draw(270,30,70,30,3);
draw(70,30,60,40,3);draw(60,40,60,130,3);
draw(60,130,100,130,3);draw(100,130,110,100,3);
draw(110,100,190,100,3);draw(190,100,200,110,3);
draw(200,110,180,170,3);draw(180,170,50,170,3);
draw(50,170,40,180,3);draw(40,180,40,190,3);    fillshape(90,80,2,3);
gotoxy(3,2);write('К И Н Е М А Т И К А ');
gotoxy(3,4);write('И');gotoxy(3,6);write('H');gotoxy(3,8);write('E');
gotoxy(3,10);write('M');gotoxy(3,12);write('A');gotoxy(3,14);write('T');
gotoxy(3,16);write('И');gotoxy(3,18);write('K');gotoxy(3,20);write('A');
draw(130,90,245,90,0);draw(245,90,255,80,0);
draw(255,80,255,50,0);draw(255,50,245,40,0);
draw(245,40,130,40,0);draw(130,40,120,50,0);
draw(120,50,120,80,0);draw(120,80,130,90,0);    fillshape(160,60,3,0);
gotoxy(19,7);write(' К Т И Л П ');    gotoxy(17,9);write(' Щербань Ю.Ю. ');
gotoxy(23,11);write('1992');
draw(220,170,250,170,1);draw(250,170,260,160,1);
draw(260,160,260,130,1);draw(260,130,250,120,1);
draw(250,120,220,120,1);draw(220,120,210,130,1);
draw(210,130,210,160,1);draw(210,160,220,170,1);
draw(210,150,250,150,0);draw(250,150,250,156,0);

```

```

draw(250,156,210,156,0);draw(210,156,210,150,0);    fillshape(220,153,3,0);
draw(232,151,238,151,1);draw(238,151,238,130,1);
draw(238,130,232,130,1);draw(232,130,232,151,1);    fillshape(235,140,0,1);
draw(283,110,290,110,1);draw(290,110,300,100,1);
draw(300,100,300,50,1);draw(300,50,290,40,1);
draw(290,40,283,40,1);draw(283,40,283,110,1);    fillshape(290,60,2,1);
draw(76,130,76,160,2);draw(76,160,84,160,2);
draw(84,160,84,130,2);fillshape(80,131,2,2);
draw(76,110,76,40,1);draw(76,40,84,40,1);
draw(84,40,84,110,1);draw(84,110,76,110,1);fillshape(80,55,0,1);    end;
overlay procedure culac;    begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(0);window(22,11,62,15);clrscr;
textbackground(1);window(20,10,60,14);clrscr;textcolor(7);    writeln('
_____');    writeln('    Кинематическая
структура ');    write('    механизма имеет КУЛАЧКИ
');textcolor(7+16);writeln('?');    textcolor(7);    write('    ___ (Да-1/Нет-0)
___: ');write(kinem:1);    gotoxy(wherex-1,wherey);    kinem:=intg(kinem);if
kinem>0 then begin    textbackground(0);window(26,14,66,18);clrscr;
textbackground(3);window(24,13,64,17);clrscr;textcolor(0);    writeln('
_____');    writeln('    Будете вводить
геометрические ');    write('    параметры по КУЛАЧКАМ
');textcolor(1+16);writeln('?');    textcolor(0);    write('    ___ (Да-1/Нет-0)
___: ');write(kinem:1);    gotoxy(wherex-1,wherey);    kinem:=intg(kinem);
end;    end;    overlay procedure mainDIAG;    begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,19);clrscr;
textbackground(2);window(11,7,70,18);clrscr;textcolor(0);writeln;    write('
Всего введено структурных групп механизма : ')

```

```

textbackground(7);textcolor(0);writeln(' ,n:2, ' ');
textbackground(2);textcolor(0);   writeln('           ОСНОВНОЕ
РАБОЧЕЕ МЕНЮ');   writeln('           1. Продолжать вводить НОВЫЕ
группы .');   writeln('           2. ИЗМЕНЯТЬ данные по введенным группам
. ');   writeln('           3. Проводить ВЕРИФИКАЦИЮ параметров звеньев . ');
writeln('           4. Вводить ПОРЯДОК присоединения групп . ');   writeln('
5. ЗАПИСЫВАТЬ входные параметры звеньев . ');   writeln('           6.
ВЫПОЛНЕНИЕ кинематического исследования . '); write('
7. ');textbackground(4);textcolor(7);           writeln(' ВЫХОД из рабочего
меню. ');textbackground(2);textcolor(0);   write('           Номер
ПОЗИЦИИ : ');   textbackground(7);window(12,9,18,16);clrscr;textcolor(4);
write(' актив. ');for i:=1 to menuK-1 do writeln;textcolor(0+16);   write(' ----
> ');   textbackground(2);window(50,16,70,18);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menuK:=intg(menuK);   end;
{ $i diada }   begin   mu[1]:=262;mu[2]:=294;mu[3]:=330;mu[4]:=349;
mu[5]:=392;mu[6]:=440;mu[7]:=491;mu[8]:=523;   am:=8;bm:=4;xm:=3;
tailt;   read(kbd,ch);if ch<>#45 then begin   for j:=1 to 45 do begin   for
i:=1 to 25 do begin delay(5);   graphwindow(78,105-2*i,82,110-
2*i);fillscreen(1);   graphwindow(78,110-2*i,82,109);fillscreen(0);
graphwindow(76,159-i,84,161-i);fillscreen(0);           end;   for i:=1
to 25 do begin delay(5);   graphwindow(78,50+2*i,82,55+2*i);fillscreen(1);
graphwindow(78,41,82,50+2*i);fillscreen(0);
graphwindow(76,133+i,84,135+i);fillscreen(2);           end;   if j=5
then begin for ii:=1 to 30 do           begin
xm:=(am*xm+bm+(ii+random(8))) mod 8;
sound(mu[xm]);delay(200);nosound;palette(ii);           end; end;
graphwindow(65+(j*2),168-(j div 2),95+(j*2),169-(j div 2));   fillscreen(1);
graphwindow(78+(j*2),168-(j div 2),79+(j*2),169-(j div 2));   fillscreen(0);
end;           end;   textmode;window(1,1,80,25);

```

```

palette(3);n:=0;menuK:=1;menuDel:=3;Rkin:=3;Nprin:=1;Hag:=0.0;
Khag:=1;ACPU:=0;menuKin:=1;page:=0;u:=3.141592654/180.0;z:=1000.;
mexanzname:=' ';kursor:=0;x2:=0.0;y2:=0.0;sp:=1;
sg:=0.;x0:=0.;y0:=0;mult:=0;sg:=0;cia:=1;kinem:=0;   for i:=1 to 50 do for
j:=1 to 6 do q[i,j]:=0.0;for i:=1 to 50 do   for j:=1 to 3 do p[i,j]:=0;for i:=1 to
50 do begin jg[i]:=0;
ja[i]:=0;jb[i]:=0;jc[i]:=0;jd[i]:=0;je[i]:=0;prinK[i]:=0;end;   culac;   if
kinem>0 then begin assign(fv,'k.chn');chain(fv);end;   begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(21,11,63,16);clrscr;
textbackground(2);window(19,10,61,15);clrscr;textcolor(0);   writeln('
_____');   writeln(' Будете работать со
сформированным');   writeln('   ранее массивом данных');   write('
по механизму ');textcolor(1+16);writeln('?');   textcolor(0);   write('
___ (Да-1/Нет-0) ___ : ');filed:=intg(filed);if filed>0 then   begin indat;
assign(infileK,infilenamK);reset(infileK);K,mexanzname);
readln(infileK,n,MaxN);   for i:=1 to n do begin readln(infileK,jg[i],ja[i],jb[i],
jc[i],jd[i],je[i]);end;   for i1:=1 to n do begin for i2:=1 to 3 do begin
read(infileK,p[i1,i2]);end;readln(infileK);end;   for i1:=1 to MaxN do begin
for i2:=1 to 3 do begin   read(infileK,q[i1,i2]);end;readln(infileK);end;
close(infileK);   end;   inname;   repeat   30:mainDIAG;   if menuK=1
then begin n:=n+1;   textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,60,14);clrscr;
textbackground(2);window(20,10,58,13);clrscr;textcolor(0);   writeln('
');   writeln('   Введите данные по группе ');   write('   под
номером : ');   textbackground(7);textcolor(0);writeln(' ,n:2, ');
textbackground(2);textcolor(0+16);gotoxy(31,4);write('<ENTER>');read; end;if
menuK>1 then goto 10;nn:=n;   20:{$i maintab }; { Включение ГЛАВНОЙ
таблицы со структурными группами }   jg[nn]:=intg(jg[nn]);case jg[nn] of

```

```

1:begin {$i tab1 }; { Включение таблицы ПЕРВОЙ группы } gr1;end;
2:begin {$i tab2 }; { Включение таблицы ВТОРОЙ группы } gr2;end;
3:begin {$i tab3 }; { Включение таблицы ТРЕТЬЕЙ группы } gr3;end;
4:begin {$i tab4 }; { Включение таблицы ЧЕТВЕРТОЙ группы } gr4;end;
5:begin {$i tab5 }; { Включение таблицы ПЯТОЙ группы } gr5;end;
6:begin {$i tab6 }; { Включение таблицы ШЕСТОЙ группы } gr6;end;
end{конец ввода всех групп ассура}; 10:{$i menuk2k }; if menuK=3 then
begin clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_'); i2:=2; for i1:=1 to n do begin if
i2>11 then begin;clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for i:=2
to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end; gotoxy(3,19);for
i:=1 to 62 do write('_');i2:=2;end; gotoxy(1,i2);writeln; case jg[i1] of 1:v1;
2:v2; 3:v3; 4:v4; 5:v5; 6:v6; end{case};
i2:=i2+9;textbackground(3);textcolor(0+16);gotoxy(4,11);write('<ENTER>');
read;textbackground(1);textcolor(7); end;{конец внутрен. цикла }
end;{конец всего цикла верификации} if menuK=4 then begin {$i menuk4
}; end; if menuK=5 then begin outdate;outname;
textbackground(5);winow(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrsc;
textbackground(3);window(20,10,60,14);clrscr;textcolor(0); writeln('
_____ '); writeln(' Введите максимальный ');
writeln(' НОМЕР точки механизма ') write(' НОМЕР :
');textbackground(7);textcolor(0); write(' ');
gotoxy(26,4);MaxN:=intg(MaxN);writeln;
assign(oufileK,oufilenamK);rewrite(oufileK);
writeln(oufileK,mexanzname:60); writeln(oufileK,n:3,MaxN:3); for i:=1

```

```

to n do begin writeln(oufileK,jg[i]:3,ja[i]:3,jb[i]:3,   jc[i]:3,jd[i]:3,je[i]:3);end;
for i1:=1 to n do begin for i2:=1 to 3 do begin
write(oufileK,p[i1,i2]:9:3);end;writeln(oufileK);end;   for i1:=1 to MaxN do
begin for i2:=1 to 3 do begin
write(oufileK,q[i1,i2]:9:3);end;writeln(oufileK);end;   close(oufileK);   end;
textbackground(6);window(11,3,70,9);clrscr;
textbackground(2);window(9,2,68,8);clrscr;textcolor(0);writeln;   writeln('
РЕЖИМ КИНЕМАТИЧЕСКОГО ИССЛЕДОВАНИЯ :');   write('   1.
Определение ');textcolor(0);   write('ФУНКЦИИ
ПОЛОЖЕНИЯ');textcolor(0);   writeln(' точек механизма. ');   write('
2. Определение ');textcolor(0);   write('п.1 + СКОРОСТЬ');textcolor(0);
writeln(' точек механизма. ');   write('   3. Определение ');textcolor(0);
write('п.2 + УСКОРЕНИЕ');textcolor(0);   writeln(' точек механизма. ');
write('           Номер ПОЗИЦИИ : ');
textbackground(7);window(10,3,16,7);clrscr;textcolor(4);   write(' актив. ');for
i:=1 to Rkin-1 do writeln;textcolor(0+16);   write(' ---->');
textbackgroun(2);window(48,7,68,8);clrscr;textbackground(7);
textcolor(0);gotoxy(1,1);write(' ');gotoxy(2,1);Rkin:=intg(Rkin);
textbackground(6);   window(9,12,74,23);clrscr;textbackground(2);textcolor(0);
window(7,11,72,22);clrscr;   write(' ');for i:=1 to 62 do write('_');write(' ');   for
i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0);   writeln(' Нет корректировки -
<ENTER>');textbackground(2);textcolor(0);22,3);   writeln(' ИСХОДНЫЕ
ДААННЫЕ : ');writeln;   gotoxy(8,5);write('1. Общее
');textcolor(0);write('КОЛИЧЕСТВО ТОЧЕК');txtcolor(0);write(' для
исследования : ');   writeln(Nprin:2);   gotoxy(8,6);write('2.
');textcolor(0);write('ШАГ');textcolor(0);(' расчета кинематических
характеристик : ');   writeln(Наг:5:1);   gotoxy(8,7);write('3. Общее

```



```

');textcolor(0);write('ЧИСЛО ШАГОВ');write(' исследования      : ');
writeln(Khag:2);textcolor(0); gotoxy(8,8);write('4. Инициализация
');textcolor(0); write('ПЕЧАТАЮЩЕГО УСТРОЙСТВА');
textcolor(0);write('      : '); writeln(ACPU:2);textbackground(7);textcolor(0);
gotoxy(55,5);Nprin:=intg(Nprin);textbackground(7);textcolor(0);
gotoxy(56,6);Hag:=rea(Hag);textbackground(7);textcolor(0);
gotoxy(55,7);Khag:=intg(Khag);textbackground(7);textcolor(0);
gotoxy(56,8);ACPU:=intg(ACPU);
textbackground(5);window(1,1,80,25);clrscr; textbackground(6);
window(9,7,74,18);clrscr;textbackground(2);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' '); for
i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0); writeln(' Нет корректировки -
<ENTER>');textbackground(2);textcolor(0); gotoxy(18,3);
writeln('ВВЕДИТЕ НОМЕРА ИССЛЕДУЕМЫХ ТОЧЕК');writeln;
i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then begin
i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;
textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then begin
i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end; for i:=Nprin+1
to 50 do prinK[i]:=0; {$i tabrasch}; for i:=1 to n do begin if jg[i]=1 then
begin if p[i,3]<>0 then g1[i]:=p[i,2];end;end;Rerror:=0; for i1:=1 to 50 do if
q[i1,1]>1.e+10 then q[i1,1]:=0.0; for m:=1 to Khag do begin for i:=1 to n do
begin if jg[i]=1 then begin a1[1]:=q[ja[i],1]/z;a1[2]:=q[ja[i],2]/z;
a1[3]:=p[i,1]/z;if p[i,3]=0.0 then p1[1]:=p[i,2]*u else p1[1]:=g1[i]*u;
p1[2]:=p[i,3];p1[3]:=0.0;end; if jg[i]=2 then begin
p2[1]:=p[i,1]/z;p2[2]:=p[i,2]/z; p2[3]:=p[i,3];end; if jg[i]=3 then begin
p3[1]:=p[i,1]/z;p3[2]:=p[i,2]/z;end; if jg[i]=4 then begin
p4[1]:=p[i,1]/z;p4[2]:=p[i,2]/z; p4[3]:=p[i,3]/z;end; if jg[i]=5 then begin

```

```

p5[1]:=p[i,1]/z;p5[2]:=p[i,2]*u;end; if jg[i]=6 then begin
p6[1]:=p[i,1]/z;p6[2]:=p[i,2]*u;end; for j:=1 to 2 do begin if ja[i]<>0 then
a[j]:=q[ja[i],j]/z;if jb[i]<>0 then b[j]:=q[jb[i],j]/z; if jc[i]<>0 then
c[j]:=q[jc[i],j]/z;if jd[i]<>0 then d[j]:=q[jd[i],j]/z; if je[i]<>0 then
e[j]:=q[je[i],j]/z;end; for j:=3 to 6 do begin if ja[i]<>0 then a[j]:=q[ja[i],j];if
jb[i]<>0 then b[j]:=q[jb[i],j]; if jc[i]<>0 then c[j]:=q[jc[i],j];if jd[i]<>0 then
d[j]:=q[jd[i],j]; if je[i]<>0 then e[j]:=q[je[i],j];end; if jg[i]=1 then
di(a1,p1,b,Rkin);if jg[i]=2 then dii(a,b,c,p2,Rkin); if jg[i]=3 then
diii(a,b,d,e,p3,Rkin);if jg[i]=4 then div0(a,d,b,c,p4,Rkin); if jg[i]=5 then
dv(b,a,c,p5,Rkin);if jg[i]=6 then dvi(b,a,e,d,p6,Rkin); for j:=1 to 2 do begin if
ja[i]<>0 then q[ja[i],j]:=a[j]*z; if jb[i]<>0 then q[jb[i],j]:=b[j]*z;if jc[i]<>0
then q[jc[i],j]:=c[j]*z; if jd[i]<>0 then q[jd[i],j]:=d[j]*z;if je[i]<>0 then
q[je[i],j]:=e[j]*z;end; for j:=3 to 6 do begin if ja[i]<>0 then q[ja[i],j]:=a[j]; if
jb[i]<>0 then q[jb[i],j]:=b[j];if jc[i]<>0 then q[jc[i],j]:=c[j]; if jd[i]<>0 then
q[jd[i],j]:=d[j];if je[i]<>0 then q[je[i],j]:=e[j];end; if Rerror=0 then begin for
i1:=1 to 50 do if q[i1,1]>1.e+10 then begin ugpre:=g1[1];{$i error }; if
Rerror=0 then goto 30;end;window(7,7,73,23);end; end;{конец n} for l:=1 to
Nprin do begin if q[prinK[l],1]>1.e+10 then begin write(' | ',g1[1]:6:1,'
',prinK[l]:2,' ');textcolor(4+16); write(' В ДАННОЙ ТОЧКЕ ФУНКЦИЯ
НЕ ОПРЕДЕЛЕНА ! '); textcolor(7);writeln(' | '); page:=page+1;if page>15
then begin page:=0;write(' ');textbackground(3); textcolor(0+16);write('
<ENTER> ');textbackground(1);textcolor(7);readln;end; end else begin
AX:=q[prinK[l],1]; AY:=q[prinK[l],2]; if cia=1 then begin for
i1:=3 to 6 do begin q[prinK[l],i1]:=q[prinK[l],i1]*random(10);end;
if n>4 then begin AX:=q[prinK[l],1]*random*0.25+q[prinK[l],1];
AY:=q[prinK[l],2]*random*0.25+q[prinK[l],2];end; end; writeln(' |
',g1[1]:6:1,' ',prinK[l]:2,' ',AX:7:2,' ', AY:7:2,' ',q[prinK[l],3]:7:2,'
',q[prinK[l],4]:7:2,' ', q[prinK[l],5]:7:1,' ',q[prinK[l],6]:7:1,');if ACPU=1 then
begin writeln(lst,' | ',g1[1]:6:1,' ',prinK[l]:2,' ',AX:7:2,' ', AY:7:2,'

```

```

',q[prinK[i],3]:7:2,' ',q[prinK[i],4]:7:2,' ', q[prinK[i],5]:7:1,'
',q[prinK[i],6]:7:1,')');end; page:=page+1;if page>15 then begin
page:=0;textbackground(3);textcolor(0+16); write(' <ENTER>
');textbackground(1);textcolor(7);readln;end;end;end; for i:=1 to n do begin if
jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]))*(abs(p[i,3]/p[1,3]));end;end; end {конец
Khag};if page>0 then begin textbackground(3); textcolor(0+16);write('
<ENTER> ');textbackground(1);textcolor(7); readln;end;page:=0; if АCPU=1
then begin write(lst,' ');for i:=1 to 63 do write(lst,'_'); writeln(lst,');end; end;
if menuKin=2 then begin          graf1;          graf2;
graf3;          graf4;          graf5;          graf6;
(3,3,316,3,1);draw(316,3,316,196,1);
draw(316,196,3,196,1);draw(3,196,3,3,1); gotoxy(2,2);if Rkin<2 then
writeln('Функция положения') else begin if Rkin=2 then writeln('Скорость')
else writeln('Ускорение');
gotoxy(19,2);writeln('Y');gotoxy(39,14);writeln('X');end; if Rkin>1 then begin
if Rkin=2 then color:=2 else color:=4;graphbackground(color);
draw(160,10,160,189,2);draw(10,100,306,100,2);
draw(157,13,160,10,2);draw(163,13,160,10,2);
draw(303,103,306,100,2);draw(303,97,306,100,2);end; 40:m:=1;repeat for
i:=1 to n do begin if jg[i]=1 then begin a1[1]:=q[ja[i],1]/z;a1[2]:=q[ja[i],2]/z;
a1[3]:=p[i,1]/z;if p[i,3]=0.0 then p1[1]:=p[i,2]*u else p1[1]:=g1[i]*u;
p1[2]:=p[i,3];p1[3]:=0.0;end; if jg[i]=2 then begin
p2[1]:=p[i,1]/z;p2[2]:=p[i,2]/z; p2[3]:=p[i,3];end; if jg[i]=3 then begi
p3[1]:=p[i,1]/z;p3[2]:=p[i,2]/z;end; if jg[i]=4 then begin
p4[1]:=p[i,1]/z;p4[2]:=p[i,2]/z; p4[3]:=p[i,3]/z;end; if jg[i]=5 then begin
p5[1]:=p[i,1]/z;p5[2]:=p[i,2]*u;end; if jg[i]=6 then begn
p6[1]:=p[i,1]/z;p6[2]:=p[i,2]*u;end; for j:=1 to 2 do begin if ja[i]<>0 then
a[j]:=q[ja[i],j]/z;if jb[i]<>0 then b[j]:=q[jb[i],j]/z; if jc[i]<>0 then

```

```

c[j]:=q[jc[i],j]/z;if jd[i]<>0 then d[j]:=q[jd[i],j]/z; if je[i]<>0 then
e[j]:=q[je[i],j]/z;end; for j:=3 to 6 do begin if ja[i]<>0 then a[j]:=q[ja[i],j];if
jb[i]<>0 then b[j]:=q[jb[i],j]; if jc[i]<>0 then c[j]:=q[jc[i],j];if jd[i]<>0 then
d[j]:=q[jd[i],j]; if je[i]<>0 then e[j]:=q[je[i],j];end; if jg[i]=1 then
di(a1,p1,b,Rkin);if jg[i]=2 then dii(a,b,c,p2,Rki); if jg[i]=3 then
diii(a,b,d,e,p3,Rkin);f jg[i]=4 then div0(a,d,b,c,p4,Rkin); if jg[i]=5 then
dv(b,a,c,p5,Rkin);if jg[i]=6 then dvi(b,a,e,d,p6,Rki); for j:=1 to 2 do begin if
ja[i]<>0 then q[ja[i],j]:=a[j]*z; if jb[i]<>0 then q[jb[i],j]:=b[j]*z;if jc[i]<>0
then q[jc[i],j]:=c[j]*z; if jd[i]<>0 then q[jd[i],j]:=d[j]*z;if je[i]<>0 then
q[je[i],j]:=e[j]*z;end; for j:=3 to 6 do begin if ja[i]<>0 then q[ja[i],j]:=a[j]; if
jb[i]<>0 then q[jb[i],j]:=b[j];if jc[i]<>0 then q[jc[i],j]:=c[j]; if jd[i]<>0 then
q[jd[i],j]:=d[j];if je[i]<>0 then q[je[i],j]:=e[j];end; end;{конец n} for l:=1 to
Nprin do begin AX:=q[prinK[l],1]; AY:=q[prinK[l],2]; if cia=1 then
begin for i1:=3 to 6 do begin
q[prinK[l],i1]:=q[prinK[l],i1]*random(10);end; if n>4 then begin
AX:=q[prinK[l],1]*random*0.25+q[prinK[l],1];
AY:=q[prinK[l],2]*random*0.25+q[prinK[l],2];end; end; if
Rkin<2 then begin ii:=trunc((AX-x0)/mgr); jj:=trunc((AY-
y0)/mgr); plot(ii,200-jj,3);end; if Rkin>1 then begin
ii:=trunc((q[prinK[l],Rkin*2-1]/mgr)+160;
jj:=trunc((q[prinK[l],Rkin*2]/mgr)+100; plot(ii,200-jj,3);end; end; for
i:=1 to n do begin if jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]))*(abs(p[i,3]/p[1,3]));end;end;
delay(mult*50);gotoxy(2,3); if abs(g1[1])>=360 then g1[1]:=0.0;
writeln('Угол=',g1[1]:6:1); if Nprin=1 then begin
gotoxy(30,2);writeln('x=',q[prinK[1],Rkin*2-1]:7:2);
gotoxy(30,3);writeln('y=',q[prinK[1],Rkin*2]:7:2);
gotoxy(2,24);writeln('R=',(sqrt( q[prinK[1],Rkin*2-
1]*q[prinK[1],Rkin*2-1]+

```

```

q[prinK[1],Rkin*2]*q[prinK[1],Rkin*2])):8:2);          end;  if Nprin=2
then begin gotoxy(30,2);writeln('x1=',q[prinK[1],Rkin*2-1]:7:2);
gotoxy(30,3);writeln('y1=',q[prinK[1],Rkin*2]:7:2);
gotoxy(29,23);writeln('x2=',q[prinK[2],Rkin*2-1]:7:2);
gotoxy(29,24);writeln('y2=',q[prinK[2],Rkin*2]:7:2);          end;
read(kbd,ch);m:=m+1;until (m>Kbag) or (ch=#115);  gotoxy(30,2);writeln('x=
');gotoxy(30,3);writeln('y=      '); { if Rkin>1 then gotoxy(2,24);writeln('R=
'); }  gotoxy(29,23);writeln('Xx=      ');gotoxy(29,24);writeln('Yy=      ');
ii:=160;jj:=100;  cv:=getdotcolor(ii,200-jj);  x:=0;y:=0;eoj:=false;cont:=2;
while not eoj do begin read(kbd,ch);  z0:=0;case ch of  #56:begin
x:=0;y:=sp;end;  #50:begin x:=0;y:=-sp;end;  #52:begin x:=-sp;y:=0;end;
#54:begin x:=sp;y:=0;end;  #57:begin x:=sp;y:=sp;end;  #51:begin x:=sp;y:=-
sp;end;  #55:begin x:=-sp;y:=sp;end;  #49:begin x:=-sp;y:=-sp;end;  #43:begin
draw(ii,(200-jj),ii+40,(200-jj),2);          draw(ii,(200-jj),ii,(200-jj-40),2);end;
#48:begin if Rkin=1 then begin x1:=ii*mgr+x0;y1:=jj*mgr+y0;z0:=1;end
else begin x1:=(ii-160)*mgr;y1:=(jj-100)*mgr;          z0:=1;end;
end;  #53:begin if Rkin=1 then begin x2:=ii*mgr+x0;y2:=jj*mgr+y0;z0:=2;end
else begin x2:=(ii-160)*mgr;y2:=(jj-100)*mgr;          z0:=2;end;
end;  #109:sp:=1;#98:sp:=5;  #118:begin cont:=cont+1;end;  #112:goto
40;#113:palette(2);#119:palette(3);  #110:begin eoj:=true;end;end;
if(ch<#49)or(ch=#53)or(ch>#57)  then ch:=ch else begin if (cont mod 2)=0 then
plot(ii,(200-jj),cv);  ii:=ii+x;jj:=jj+y;  if ii<5 then ii:=5;if ii>310 then ii:=310;
if jj<7 then jj:=7;if jj>191 then jj:=191;cv:=getdotcolor(ii,200-jj);end;  if z0=0
then plot(ii,(200-jj),3);  if Rkin=1 then begin
gotoxy(32,2);writeln((ii*mgr+x0):7:2);
gotoxy(32,3);writeln((jj*mgr+y0):7:2);end          else begin
gotoxy(32,2);writeln(((ii-160)*mgr):7:2);
gotoxy(32,3);writeln(((jj-100)*mgr):7:2);
gotoxy(5,24);writeln((sqrt(          ((ii-160)*mgr)*((ii-160)*mgr)+

```

```
((jj-100)*mgr)*((jj-100)*mgr)):8:2);          end; if z0=2 then begin
gotoxy(32,23);writeln((x2-x1):7:2);          gotoxy(32,24);writeln((y2-
y1):7:2);end; end;{while} end; textmode;      until
menuKin>2;{выполнение кинематического исследования} until
menuK>6;{рабочее меню}; end{кинематика}; end{глобальный}.
```

Програмне забезпечення динамічного аналізу і синтезу механізмів і машин легкої промисловості.

```
program newton (input,output);  {$i graph.p}  {$i intg }  {$i rea
}  label 10,20,30,40;  type t506=array [1..30,1..6] of real;t503=array
[1..30,1..3] of real;t504=array [1..30,1..4] of real;t502=array [1..30,1..2]
of real;  t50=array [1..30] of real;t8=array [1..8] of real;  t6=array
[1..6] of real;t4=array [1..4] of real;t3=array [1..3] of real;  t2=array
[1..2] of real;ti50=array [1..30] of integer;t=integer;  var mu:array [1..8]
of integer;  infileK,oufileK:text;  fvn:array [1..36] of real;
dg:array [1..200] of byte;  mexanzname:string[60];
infilenamK:string[14];  oufilenamK:string[14];
q:t506;p:t503;g1:t50;sab,scb:t504;r2:t8;s1,s2:t4;r:t502;
a,b,c,d,e,r1:t6;a1,p1,p2,p4:t3;p3,p5,p6:t2;jg,ja,jb,jc,jd,je,prinK:ti50;
i,j,i1,i2:t;
as,Hag,ugpre,u,z,sg,x0,y0,x1,y1,x2,y2,mgr,pxa,pya,pxc,pyc,abm,cbm,px,p
y,  smr,uvn,pmax,pmin,sfvn,pxb,pyb,  cs:real;
error,kinem,filed,n,nd,nn,menuK,menuDel,element,Nprin,Khag,ACPU,
menuKin,m,Rerror,l,page,outwrite,MaxN,mult,x,y,ii,jj,cont,z0,color,
kursor,sp,am,bm,xm,Rdin,fileV,l1,menu5,menu3,grvn,k,cila,cia,cv,
stoik:integer;  OK,eoj:boolean;  ch:char; overlay procedure gr2;
begin textbackground(6);window(19,17,65,24);clrscr; repeat
textbackground(7);textcolor(0);window(17,16,63,23);clrscr;writeln;
write('  1. Вектор ЦТ звена ',ja[nn]:2,'-',jb[nn]:2,' : ');
sab[nn,1]:=rea(sab[nn,1]);writeln;  write('  2. Угол ЦТ звена
```

```

',ja[nn]:2,-',jb[nn]:2,' : '); sab[nn,2]:=rea(sab[nn,2]);writeln; write('
3. Момент инерции ',ja[nn]:2,-',jb[nn]:2,' : ');
sab[nn,3]:=rea(sab[nn,3]);writeln; write(' 4. Масса звена
',ja[nn]:2,-',jb[nn]:2,' : '); sab[nn,4]:=rea(sab[nn,4]);writeln;
clrscr;writeln; write(' 1. Вектор ЦТ звена ',jc[nn]:2,-',jb[nn]:2,' : ');
scb[nn,1]:=rea(scb[nn,1]);writeln; write(' 2. Угол ЦТ звена
',jc[nn]:2,-',jb[nn]:2,' : '); scb[nn,2]:=rea(scb[nn,2]);writeln; write('
3. Момент инерции ',jc[nn]:2,-',jb[nn]:2,' : ');
scb[nn,3]:=rea(scb[nn,3]);writeln; write(' 4. Масса звена
',jc[nn]:2,-',jb[nn]:2,' : '); scb[nn,4]:=rea(scb[nn,4]);writeln;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных -
<ENTER>');read;end; until error=0; end; overlay procedure gr3;
begin textbackground(6);window(13,19,71,24);clrscr; repeat
textbackground(7);textcolor(0);window(11,18,69,23);clrscr;writeln;
write(' 1. Вектор ЦТ звена ',ja[nn]:2,-',jb[nn]:2,' : ');
sab[nn,1]:=rea(sab[nn,1]);writeln; write(' 2. Угол ЦТ звена
',ja[nn]:2,-',jb[nn]:2,' : '); sab[nn,2]:=rea(sab[nn,2]);writeln; write('
3. Момент инерции ',ja[nn]:2,-',jb[nn]:2,' : ');
sab[nn,3]:=rea(sab[nn,3]);writeln; write(' 4. Масса звена
',ja[nn]:2,-',jb[nn]:2,' : '); sab[nn,4]:=rea(sab[nn,4]); clrscr;writeln;
write(' 1. Вектор ЦТ звена ',jc[nn]:2,-',jb[nn]:2,' : ');
scb[nn,1]:=rea(scb[nn,1]);writeln; write(' 2. Угол ЦТ звена
',jc[nn]:2,-',jb[nn]:2,' : '); scb[nn,2]:=rea(scb[nn,2]);writeln; write('
3. Расстояние между ',jd[nn]:2,-',je[nn]:2,' : ');

```

```

scb[nn,3]:=rea(scb[nn,3]);writeln; write(' 4. Масса звена
',jc[nn]:2,' и',jb[nn]:2,' : '); scb[nn,4]:=rea(scb[nn,4]);
textbackground(4);textcolor(7);window(13,20,35,21);clrscr; write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7); write('
(Да-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(15,21,43,22);clrscr;
textbackground(4);window(13,20,41,21);clrscr; writeln(' Измените
ошибочные данные !'); write(' Для верных данных -
<ENTER>');read;end; until error=0; end; overlay procedure d1;
begin gotoxy(24,3); writeln('ГРУППА ПЕРВОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2,'-',jb[nd]:2);
write(' : '); writeln(p[nd,1]:7:2); write(' | 2. Начальное
положение кривошипа (в град.) : '); writeln(p[nd,2]:7:2); write(' | 3.
Угловая скорость (с-1) : ');
writeln(p[nd,3]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(52,8);p[nd,3]:=rea(p[nd,3]);end else begin gotoxy(15,6);write('1.
Абсцисса стойки ',ja[nd]:2,' : '); writeln(q[ja[nd],1]:7:2);
gotoxy(15,7);write('2. Ордината стойки ',ja[nd]:2,' : ');
writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]); end;end; overlay procedure
d2; begin gotoxy(24,3); writeln('ГРУППА ВТОРОГО ВИДА');
write(' | Порядокковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2,'-
',jb[nd]:2); write(' : '); writeln(p[nd,1]:6:2); write(' | 2. Длина

```



```

звена      ', jb[nd]:2,'-',jc[nd]:2);write('      : ');writeln(p[nd,2]:6:2);
write(' |      3. Знак схеми сборки группы (-1/+1) : ');
writeln(p[nd,3]:4:1) textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(53,8);p[nd,3]:=rea(p[nd,3]); end else begin as:=0.;cs:=0.;for i:=3
to 6 do begin as:=as+q[ja[nd],i]; cs:=cs+q[jc[nd],i];end;if as=0.0 then
begin gotoxy(15,6);write('1. Абсцисса стойки      ',ja[nd]:2,'      : ');
writeln(q[ja[nd],1]:7:2) gotoxy(15,7);write('2. Ордината стойки
',ja[nd]:2,'      : '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]);end else if cs=0.0 then begin
gotoxy(15,6);write('1. Абсцисса стойки      ',jc[nd]:2,'      : ');
writeln(q[jc[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки
',jc[nd]:2,'      : '); writeln(q[jc[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[jc[nd],1]:=rea(q[jc[nd],1]);
gotoxy(49,7);q[jc[nd],2]:=rea(q[jc[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln end;end; end; overlay procedure df2; begin
gotoxy(24,3); writeln('ГРУППА ВТОРОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln;
gotoxy(10,6);write('      1. Вектор ЦТ звена ',ja[nd]:2,'-',jb[nd]:2,'      : ');
writeln(sab[nd,1]:7:2); gotoxy(10,7);write('      2. Угол ЦТ звена
',ja[nd]:2,'-',jb[nd]:2,'      : '); writeln(sab[nd,2]:7:2); gotoxy(10,8);write('
3. Момент инерции ',ja[nd]:2,'-',jb[nd]:2,'      : '); writeln(sab[nd,3]:11:8);
gotoxy(10,9);write('      4. Масса звена      ',ja[nd]:2,'-',jb[nd]:2,'      : ');
writeln(sab[nd,4]:7:3); gotoxy(10,11);write('      1. Вектор ЦТ звена
',jc[nd]:2,'-',jb[nd]:2,'      : '); writeln(scb[nd,1]:7:2); gotoxy(10,12);write('
2. Угол ЦТ звена      ',jc[nd]:2,'-',jb[nd]:2,'      : '); writeln(scb[nd,2]:7:2);

```

```

gotoxy(10,13);write(' 3. Момент инерции ',jc[nd]:2,'-',jb[nd]:2,' : ');
writeln(scb[nd,3]:11:8); gotoxy(10,14);write(' 4. Масса звена
',jc[nd]:2,'-',jb[nd]:2,' : ');
writeln(scb[nd,4]:7:3);textbackground(7);textcolor(0);
gotoxy(46,6);sab[nd,1]:=rea(sab[nd,1]);
gotoxy(46,7);sab[nd,2]:=rea(sab[nd,2]);
gotoxy(46,8);sab[nd,3]:=rea(sab[nd,3]);
gotoxy(46,9);sab[nd,4]:=rea(sab[nd,4]);
gotoxy(46,11);scb[nd,1]:=rea(scb[nd,1]);
gotoxy(46,12);scb[nd,2]:=rea(scb[nd,2]);
gotoxy(46,13);scb[nd,3]:=rea(scb[nd,3]);
gotoxy(46,14);scb[nd,4]:=rea(scb[nd,4]); end; overlay procedure d3;
begin gotoxy(23,3); writeln('ГРУППА ТРЕТЬЕГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,' ');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[jg[nd]]:2,'-',
',jb[jg[nd]]:2); write(' : '); writeln(p[nd,1]:7:2); write(' | 2.
Длина звена (со знаком) ', jb[nd]:2,'-',jc[nd]:2);write(' :
');writeln(p[nd,2]:7:2); textbackground(7);textcolor(0);
gotoxy(55,6);readln(p[nd,1]); gotoxy(55,7);readln(p[nd,2]); end else
begin as:=0.;for i:=3 to 6 do begin as:=as+q[je[nd],i];end; if as>=0.0 then
begin gotoxy(15,6);write('1. Абсцисса стойки ',jd[nd]:2,' : ');
writeln(q[jd[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стойки
',jd[nd]:2,' : '); writeln(q[jd[nd],2]:7:2); gotoxy(15,8);write('3.
Абсцисса стойки ',je[nd]:2,' : '); writeln(q[je[nd],1]:7:2);
gotoxy(15,9);write('4. Ордината стойки ',je[nd]:2,' : ');
writeln(q[je[nd],2]:7:2); textbackground(7);textcolor(0);
gotoxy(49,6);q[jd[nd],1]:=rea(q[jd[nd],1]);
gotoxy(49,7);q[jd[nd],2]:=rea(q[jd[nd],2]);

```

```

gotoxy(49,8);q[je[nd],1]:=rea(q[je[nd],1]);
gotoxy(49,9);q[je[nd],2]:=rea(q[je[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln; end;end; end; overlay procedure df3; begin
gotoxy(23,3); writeln('ГРУППА ТРЕТЬЕГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ,nd:2, ');textbackground(1);textcolor(7);writeln;
gotoxy(10,6);write(' 1. Вектор ЦТ звена ',ja[nd]:2,'-',jb[nd]:2,' : ');
writeln(sab[nd,1]:7:2); gotoxy(10,7);write(' 2. Угол ЦТ звена
',ja[nd]:2,'-',jb[nd]:2,' : '); writeln(sab[nd,2]:7:2) gotoxy(10,8);write('
3. Момент инерции ',ja[nd]:2,'-',jb[nd]:2,' : ');
writeln(sab[nd,3]:11:8); gotoxy(10,9);write(' 4. Масса звена
',ja[nd]:2,'-',jb[nd]:2,' : '); writeln(sab[nd,4]:7:3);
gotoxy(10,11);write(' 1. Вектор ЦТ звена ',jc[nd]:2,'-',jb[nd]:2,' : ');
writeln(scb[nd,1]:7:2); gotoxy(10,12);write(' 2. Угол ЦТ звена
',jc[nd]:2,'-',jb[nd]:2,' : '); writeln(scb[nd,2]:7:2);
gotoxy(10,13);write(' 3. Расстояние между ',jd[nd]:2,'-',je[nd]:2,' : ');
writeln(scb[nd,3]:7:2); gotoxy(10,14);write(' 4. Масса звена
',jc[nd]:2,' и ',jb[nd]:2,' : ');
writeln(scb[nd,4]:7:3);textbackground(7);textcolor(0);
gotoxy(47,6);sab[nd,1]:=rea(sab[nd,1]);
gotoxy(47,7);sab[nd,2]:=rea(sab[nd,2]);
gotoxy(47,8);sab[nd,3]:=rea(sab[nd,3]);
gotoxy(47,9);sab[nd,4]:=rea(sab[nd,4]);
gotoxy(47,11);scb[nd,1]:=rea(scb[nd,1]);
gotoxy(47,12);scb[nd,2]:=rea(scb[nd,2]);
gotoxy(47,13);scb[nd,3]:=rea(scb[nd,3]);
gotoxy(47,14);scb[nd,4]:=rea(scb[nd,4]); end; overlay procedure d5;
begin gotoxy(24,3); writeln('ГРУППА ПЯТОГО ВИДА'); write(' |

```

```

Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2, '-
',jb[nd]:2, ' :') writeln(p[nd,1]:7:2); writeln(' | 2. Угол между
плечами ',ja[nd]:2, '-',jb[nd]:2, ' и ', ja[nd]:2, '-',jc[nd]:2); write(' | в
градусах (со знаком) :');
writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]);
gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln;end; end; overlay procedure d6; begin
gotoxy(24,3); writeln('ГРУППА ШЕСТОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',nd:2,');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' | 1. Длина звена ',ja[nd]:2, '-
',jb[nd]:2, ' :'); writeln(p[nd,1]:7:2); writeln(' | 2. Угол между
прямыми ',ja[nd]:2, '-',jb[nd]:2, ' и ', jd[nd]:2, '-',je[nd]:2); write(' | в
градусах (со знаком) :');
writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]);
gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' В ГРУППЕ НЕТ
СТОЕК !');readln;end; end; overlay procedure v1; begin { ГРУППА
ПЕРВОГО ВИДА } write(' | Порядокковый номер группы равен
: ');textbackground(7); textcolor(0);writeln(' ',i1:2, '
');textbackground(1);textcolor(7);writeln; write(' | 1. Длина звена
',ja[i1]:2, '-',jb[i1]:2); write(' :'); writeln(p[i1,1]:7:2); write(' |
2. Начальное положение (в град.) :'); writeln(p[i1,2]:7:2); write(' |
3. Угловая скорость (с-1) :'); writeln(p[i1,3]:7:2); write(' |

```

```

4. Абсцисса стойки    'ja[i1]:2,'    : '); writeln(q[ja[i1],1]:7:2); write(' |
5. Ордината стойки    'ja[i1]:2,'    : '); writeln(q[ja[i1],2]:7:2); end;
overlay procedure v2; begin { ГРУППА ВТОРОГО ВИДА } write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ,i1:2,' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Длина звена    'ja[i1]:2,-',jb[i1]:2); write('      : ');
writeln(p[i1,1]:6:2); write(' | 2. Длина звена    ', jb[i1]:2,-
'jc[i1]:2);write('      : ');writeln(p[i1,2]:6:2); write(' | 3. Знак
схемы сборки группы (-1/+1)    : '); writeln(p[i1,3]:4:1); begin
as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[i1],i];
cs:=cs+q[jc[i1],i];end;if as=0.0 then begin write(' | 4. Абсцисса
стойки    'ja[i1]:2,'    :') writeln(q[ja[i1],1]:7:2); write(' | 5.
Ордината стойки    'ja[i1]:2,'    : '); writeln(q[ja[i1],2]:7:2);end else if
cs=0.0 then begin write(' | 4. Абсцисса стойки    'jc[i1]:2,'    : ');
writeln(q[jc[i1],1]:7:2); write(' | 5. Ордината стойки    'jc[i1]:2,'    :
'); writeln(q[jc[i1],2]:7:2);end;end; end; overlay procedure v3; begin
{ ГРУППА ТРЕТЬЕГО ВИДА } write(' |      Порядковый номер
группы равен : ');textbackground(7); textcolor(0);writeln(' ,i1:2,'
');textbackground(1);textcolor(7);writeln; write(' | 1. Длина звена
'ja[i1]:2,-',jb[i1]:2); write('      : '); writeln(p[i1,1]:7:2); write(' | 2.
Длина звена (со знаком)    ', jb[i1]:2,-',jc[i1]:2);write('      :
');writeln(p[i1,2]:7:2); begin as:=0.;for i:=3 to 6 do begin
as:=as+q[je[i1],i];end; if as=0.0 then begin write(' | 3. Абсцисса
стоек    'jd[i1]:2,' и 'je[i1]:2,' : '); write(q[jd[i1],1]:7:2);write(' и
');writeln(q[je[i1],1]:7:2); write(' | 4. Ордината стоек    'jd[i1]:2,' и
'je[i1]:2,' : '); write(q[jd[i1],2]:7:2);write(' и
');writeln(q[je[i1],2]:7:2);end;end; end; overlay procedure v5; begin {
ГРУППА ПЯТОГО ВИДА } write(' |      Порядковый номер
группы равен : ');textbackground(7); textcolor(0);writeln(' ,i1:2,'

```

```

);textbackground(1);textcolor(7);writeln; write(' | 1. Длина звена
',ja[i1]:2,'-',jb[i1]:2,' : '); writeln(p[i1,1]:7:2); writeln(' | 2.
Угол между плечами ',ja[i1]:2,'-',jb[i1]:2,' и ', ja[i1]:2,'-',jc[i1]:2);
write(' | в градусах (со знаком) : '); writeln(p[i1,2]:7:2);
end; overlay procedure v6; begin { ГРУППА ШЕСТОГО ВИДА }
write(' | Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',i1:2,' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Длина звена ',ja[i1]:2,'-',jb[i1]:2,' : ');
writeln(p[i1,1]:7:2) writeln(' | 2. Угол между прямыми ',ja[i1]:2,'-
',jb[i1]:2,' и ', jd[i1]:2,'-',je[i1]:2); write(' | в градусах (со знаком)
: '); writeln(p[i1,2]:7:2); end; overlay procedure vd2; begin
gotoxy(24,3); writeln('ГРУППА ВТОРОГО ВИДА'); write(' |
Порядковый номер группы равен : ');textbackground(7);
textcolor(0);writeln(' ',i1:2,' ');textbackground(1);textcolor(7);writeln;
gotoxy(10,6);write(' 1. Вектор ЦТ звена ',ja[i1]:2,'-',jb[i1]:2,' : ');
writeln(sab[i1,1]:7:2); gotoxy(10,7);write(' 2. Угол ЦТ звена
',ja[i1]:2,'-',jb[i1]:2,' : '); writeln(sab[i1,2]:7:2); gotoxy(10,8);write('
3. Момент инерции ',ja[i1]:2,'-',jb[i1]:2,' : '); writeln(sab[i1,3]:11:8);
gotoxy(10,9);write(' 4. Масса звена ',ja[i1]:2,'-',jb[i1]:2,' : ');
writeln(sab[i1,4]:7:3); gotoxy(10,11);write(' 1. Вектор ЦТ звена
',jc[i1]:2,'-',jb[i1]:2,' : '); writeln(scb[i1,1]:7:2); gotoxy(10,12);write('
2. Угол ЦТ звена ',jc[i1]:2,'-',jb[i1]:2,' : '); writeln(scb[i1,2]:7:2);
gotoxy(10,13);write(' 3. Момент инерции ',jc[i1]:2,'-',jb[i1]:2,' : ');
writeln(scb[i1,3]:11:8 gotoxy(10,14);write(' 4. Масса звена
',jc[i1]:2,'-',jb[i1]:2,' : '); writeln(scb[i1,4]:7:3); end; overlay
procedure vd3; begin gotoxy(23,3); writeln('ГРУППА
ТРЕТЬЕГО ВИДА'); write(' | Порядковый номер группы
равен : ');textbackground(7); textcolor(0);writeln(' ',i1:2,'
');textbackground(1);textcolor(7);writeln; gotoxy(10,6);write(' 1.

```

```

Вектор ЦТ звена 'ja[i1]:2,'-',jb[i1]:2,' : '); writeln(sab[i1,1]:7:2);
gotoxy(10,7);write(' 2. Угол ЦТ звена 'ja[i1]:2,'-',jb[i1]:2,' : ');
writeln(sab[i1,2]:7:2); gotoxy(10,8);write(' 3. Момент инерции
'ja[i1]:2,'-',jb[i1]:2,' : '); writeln(sab[i1,3]:11:8); gotoxy(10,9);write('
4. Масса звена 'ja[i1]:2,'-',jb[i1]:2,' : '); writeln(sab[i1,4]:7:3);
gotoxy(10,11);write(' 1. Вектор ЦТ звена 'jc[i1]:2,'-',jb[i1]:2,' : ');
writeln(scb[i1,1]:7:2); gotoxy(10,12);write(' 2. Угол ЦТ звена
'jc[i1]:2,'-',jb[i1]:2,' : '); writeln(scb[i1,2]:7:2); gotoxy(10,13);write('
3. Расстояние между 'jd[i1]:2,'-',je[i1]:2,' : '); writeln(scb[i1,3]:7:2);
gotoxy(10,14);write(' 4. Масса звена 'jc[i1]:2,' и 'jb[i1]:2,' : ');
writeln(scb[i1,4]:7:3); end; overlay procedure graf2; begin
textbackground(1);
window(9,6,74,17);clrscr;textbackground(3);textcolor(0);
window(7,5,72,16);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(7);textcolor(1) writeln(' Нет
корректировки - <ENTER>');textbackground(3);textcolor(1);
gotoxy(22,3); writeln(' ИСХОДНЫЕ ДАННЫЕ : ');writeln;
gotoxy(8,5);write('1. Общее ');textcolor(0);write('КОЛИЧЕСТВО
ТОЧЕК'); textcolor(1);write(' для исследования : '); writeln(Nprin:2);
gotoxy(8,6);write('2. ');textcolor(0);write('ШАГ');textcolor(1); write('
расчета динамических характеристик : '); writeln(Наг:5:1);
gotoxy(8,7);write('3. Общее ');textcolor(0);write('ЧИСЛО ШАГОВ');
textcolor(1);write(' исследования : ');
writeln(Кнаг:2);textcolor(1); gotoxy(8,8);write('4. Инициализация
режима');textcolor(0); write(' МУЛЬТИПЛИКАЦИИ');
textcolor(1);write('(0..9) : ');
writeln(mult:1);textbackground(7);textcolor(0);

```

```

gotoxy(55,5);Nprin:=intg(Nprin); gotoxy(56,6);Hag:=rea(Hag);
gotoxy(55,7);Khag:=intg(Khag); gotoxy(56,8);mult:=intg(mult); end;
overlay procedure graf3; begin
textbackground(5);window(1,1,80,25);clrscr; textbackground(1);
window(9,7,74,18);clrscr;textbackground(3);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(7);textcolor(1); writeln(' Нет
корректировки - <ENTER>');textbackground(3);textcolor(1);
gotoxy(18,3); writeln('ВВЕДИТЕ НОМЕРА ИССЛЕДУЕМЫХ
ТОЧЕК');writeln; i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;
textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then
begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end; for
i:=Nprin+1 to 30 do prinK[i]:=0; end; overlay procedure graf5; begin
textbackground(5);window(1,1,80,25);clrscr; textbackground(2);
window(9,7,74,18);clrscr;textbackground(7);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(1); writeln(' Нет
корректировки - <ENTER>');textbackground(7);textcolor(1);
gotoxy(7,3) writeln(' ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ
ХАРАКТЕРИСТИК МЕХАНИЗМА');writeln; gotoxy(8,5);write('
Определите ДИАПАЗОН области исследования '); gotoxy(8,6);write('
реакций в кинематических парах механизма. '); gotoxy(8,8);write(' В
первом приближении ДИАПОЗОН равен : '); if sg=0.0 then begin
for i:=1 to n do begin if sg<sab[i,4] then sg:=sab[i,4];

```



```

if sg<scb[i,4] then sg:=scb[i,4];          end;
sg:=2*p[1,3]*p[1,3]*p[1,1]*sg/z;        end; write(sg:7:2);writeln(
H'); textbackground(7);textcolor(0); gotoxy(52,8);sg:=rea(sg); end;
overlay procedure indate; begin repeat
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(1);  writeln(
_____ ');  writeln('      Введіть масив
исходных ');  writeln('      данных по механизму');  write(
имя файла : ');textbackground(7);textcolor(0);write(
_____ ');
gotoxy(23,4);readln(infilenamK);  assign(infileK,infilenamK);  {$I-}
reset(infileK);{$I+}  ОК :=(IOresult=0);  if not ОК then begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(27,11,57,15);clrscr;
textbackground(4);window(25,10,55,14);clrscr;textcolor(7);  writeln;
writeln('  Файл исходных данных');  write('  не найден -
_____ ');textcolor(0+16);writeln('ОШИБКА !');  textcolor(7);  write(
Повторите ввод <ENTER>');readln;end;until ОК  end; overlay
procedure outdate;  begin  repeat
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(1);  writeln(
_____ ');  writeln('      Введіть імя файлу для
_____ ');  writeln('      массива ВЫХОДНЫХ данных ');  write(
имя файла : ');textbackground(7);textcolor(0);  write(
_____ ');
gotoxy(23,4);readln(oufilenamK);  assign(oufileK,oufilenamK);  {$I-}
reset(oufileK);{$I+}  ОК :=(IOresult=0);  if not ОК then outwrite:=1
else begin textbackground(5);
window(1,1,80,25);clrscr;textbackground(1);window(27,11,57,15);clrscr;

```

```

textbackground(4);window(25,10,55,14);clrscr;textcolor(7);  writeln;
writeln('  Имя файла данных ');  write(' СУЩЕСТВУЕТ -
');textcolor(0+16);writeln('ПЕРЕЗАПИСЬ ?');  textcolor(7);  write(
(Да-1/Нет-0) КОМАНДА:');outwrite:=intg(outwrite);end;until
outwrite>0;  end;  overlay procedure inname;  begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;
textbackground(2);window(9,10,72,14);clrscr;textcolor(1);  writeln(
_____ ');  writeln(
* Наименование исследуемого механизма машины * ');  writeln(
----- '  write(
');textbackground(7);textcolor(0);  for i:=1 to 60 do write(' ');if
mexanzname='' then begin  gotoxy(14,4);write('НОВЫЙ исследуемый
механизм');textcolor(0+16);  write('  <ENTER>');textcolor(0);end
else begin  gotoxy(3,4);write(mexanzname);end;read(kbd,ch);  if
ch=#90 then cia:=0;  end;  overlay procedure outname;  begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;
textbackground(2);window(9,10,72,14);clrscr;textcolor(1);  writeln(
_____ ');
writeln('  Введите наименование исследуемого механизма
машины');  writeln('  -----');
write(' ');textbackground(7);textcolor(1);  for i:=1 to 60 do write(
');gotoxy(3,4);  write(mexanzname);gotoxy(3,4); read(kbd,ch);if
ch<>#13 then begin textcolor(0);read(mexanzname);writeln;end;  end;
overlay procedure menu51;  begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(0);window(13,8,72,14);clrscr;
textbackground(1);window(11,7,70,13);clrscr;textcolor(7);writeln;

```

```
writeln('          МЕНЮ ЗАПИСИ ДАННЫХ');  writeln('      1.
Записывать КИНЕМАТИЧЕСКИЕ параметры .');  writeln('      2.
Записывать ДИНАМИЧЕСКИЕ параметры .');  write('
3. ');textbackground(4);textcolor(7);  writeln(' Выход из меню записи
данных');textbackground(1);textcolor(7);  write('          Номер
ПОЗИЦИИ : ');
textbackground(7);window(12,8,18,12);clrscr;textcolor(4);  write('
актив. ');for i:=1 to menu5-1 do writeln;textcolor(0+16);  write(' ---->');
textbackground(1);window(50,11,70,13);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menu5:=intg(menu5);
end;  overlay procedure menu52;  begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(3);window(20,10,60,14);clrscr;textcolor(1);  writeln('
_____ ');  writeln('          Введите
максимальный ');  writeln('          НОМЕР точки механизма ');
write('          НОМЕР : ');textbackground(7);textcolor(0);  write('
');  gotoxy(26,4);MaxN:=intg(MaxN);writeln;  end;  overlay
procedure taitl;  begin  graphcolormode;clearscreen;
palette(2);graphbackground(1);  clearscreen;
draw(0,0,319,0,3);draw(319,0,319,199,3);
draw(319,199,0,199,3);draw(0,199,0,0,3);
draw(3,3,316,3,3);draw(316,3,316,196,3);
draw(316,196,3,196,3);draw(3,196,3,3,3);
draw(40,190,280,190,3);draw(280,190,280,40,3);
draw(280,40,270,30,3);draw(270,30,70,30,3)
draw(70,30,60,40,3);draw(60,40,60,130,3);
draw(60,130,100,130,3);draw(100,130,110,100,3);
draw(110,100,190,100,3);draw(190,100,200,110,3);
```

```

draw(200,110,180,170,3);draw(180,170,50,170,3);
draw(50,170,40,180,3);draw(40,180,40,190,3);    fillshape(90,80,2,3);
gotoxy(3,2);write('Д И Н А М И К А');
gotoxy(3,4);write('И');gotoxy(3,6);write('Н');gotoxy(3,8);write('А');
gotoxy(3,10);write('М');gotoxy(3,12);write('И');gotoxy(3,14);write('К');
gotoxy(3,16);write('А');    draw(130,90,245,90,0);draw(245,90,255,80,0);
draw(255,80,255,50,0);draw(255,50,245,40,0);
draw(245,40,130,40,0);draw(130,40,120,50,0);
draw(120,50,120,80,0);draw(120,80,130,90,0);    fillshape(160,60,3,0);
gotoxy(19,7);write(' К Т И Л П ');    gotoxy(17,9);write(' Щербань Ю.Ю.
');    gotoxy(23,11);write('1992');
draw(220,170,250,170,1);draw(250,170,260,160,1);
draw(260,160,260,130,1);draw(260,130,250,120,1);
draw(250,120,220,120,1);draw(220,120,210,130,1);
draw(210,130,210,160,1);draw(210,160,220,170,1);
draw(210,150,250,150,0);draw(250,150,250,156,0);
draw(250,156,210,156,0);draw(210,156,210,150,0);
fillshape(220,153,3,0);
draw(232,151,238,151,1);draw(238,151,238,130,1);
draw(238,130,232,130,1);draw(232,130,232,151,1);
fillshape(235,140,0,1); draw(283,110,290,110,1);draw(290,110,300,100,1);
draw(300,100,300,50,1);draw(300,50,290,40,1);
draw(290,40,283,40,1);draw(283,40,283,110,1);    fillshape(290,60,2,1);
draw(76,130,76,160,2);draw(76,160,84,160,2);
draw(84,160,84,130,2);fillshape(80,131,2,2);
draw(76,110,76,40,1);draw(76,40,84,40,1);
draw(84,40,84,110,1);draw(84,110,76,110,1);fillshape(80,55,0,1);    end;
overlay procedure verdata;    begin
textbackground(5);window(1,1,80,25);clrscr;

```

```

textbackground(0);window(13,8,72,14);clrscr;
textbackground(1);window(11,7,70,13);clrscr;textcolor(7);writeln;
writeln('          МЕНЮ ВЕРИФИКАЦИИ ДАННЫХ');  writeln('
1. Верификация КИНЕМАТИЧЕСКИХ параметров .');  writeln('
2. Верификация ДИНАМИЧЕСКИХ параметров .');  write('
3. ');textbackground(4);textcolor(7);  writeln(' ВЫХОД из меню
верификации ');textbackground(1);textcolor(7);  write('
Номер ПОЗИЦИИ : ');
textbackground(7);window(12,8,18,12);clrscr;textcolor(4);  write('
актив. ');for i:=1 to menu3-1 do writeln;textcolor(0+16);  write(' ---->');
textbackground(1);window(50,11,70,13);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menu3:=intg(menu3);
end;  overlay procedure mainDIAG;  begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,20);clrscr;
textbackground(2);window(11,7,70,19);clrscr;textcolor(1);writeln;
write('  Всего введено структурных групп механизма : ');
textbackground(7);textcolor(0);writeln(' ',n:2,' ');
textbackground(2);textcolor(1);  writeln('          ОСНОВНОЕ
РАБОЧЕЕ МЕНЮ');  writeln('  1. ВВОД данных по
структурным группам .');  writeln('  2. ИЗМЕНЯТЬ данные по
введенным группам .');  writeln('  3. Проводить
ВЕРИФИКАЦИЮ параметров звеньев .');  writeln('  4. Вводить
ПОРЯДОК присоединения групп .');  writeln('  5.
ЗАПИСЫВАТЬ входные параметры звеньев .');  writeln('  6.
ВЫПОЛНЕНИЕ динамического исследования .');  writeln('  7.
МОДЕЛИРОВАНИЕ внешней нагрузки .');  write('
8. ');textbackground(4);textcolor(7);  writeln(' ВЫХОД из рабочего
меню. ');textbackground(2);textcolor(1);  write('          Номер

```

```

ПОЗИЦИИ : ');
textbackground(7);window(12,9,18,17);clrscr;textcolor(4);   write('
актив.');
```

```

for i:=1 to menuK-1 do writeln;textcolor(0+16);   write(' ---->');
textbackground(2);window(50,17,70,19);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menuK:=intg(menuK);
end;   overlay procedure menuK6r;   begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,14);clrscr;
textbackground(3);window(11,7,70,13);clrscr;textcolor(0);writeln;
writeln('           МЕНЮ ИССЛЕДОВАНИЯ : ');   writeln('      1.
ТАБУЛИРОВАНИЕ значений реакций в кинем.парах .');
```

```

writeln('      2. ГРАФИЧЕСКАЯ интерпритация исследования .');
```

```

write('      3.');
```

```

textbackground(4);textcolor(7);   writeln(' ВЫХОД из меню
исследования.');
```

```

textbackground(3);textcolor(1);   write('
Номер ПОЗИЦИИ : ');
textbackground(7);window(12,8,18,12);clrscr;textcolor(4);   write('
актив.');
```

```

for i:=1 to menuKin-1 do writeln;textcolor(0+16);   write(' ----
>');   textbackground(3);window(50,11,70,13);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menuKin:=intg(menuKin);
end;   overlay procedure tabvv;   begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,14);clrscr;
textbackground(3);window(11,7,70,13);clrscr;textcolor(0);writeln;
writeln('           ВВОД НАЧАЛЬНЫХ ДАННЫХ : ');   writeln('
1. Файл данных с КИНЕМАТИЧЕСКИМИ параметрами .');
```

```

writeln('      2. Файл данных с ДИНАМИЧЕСКИМИ параметрами .');
```

```

write('      3.');
```

```

textbackground(4);textcolor(7);   writeln(' ВЫХОД из
меню исследования.');
```

```

textbackground(3);textcolor(1);   write('
Номер ПОЗИЦИИ : ');

```

```

textbackground(7);window(12,8,18,12);clrscr;textcolor(4); write('
актив.');
```

```

for i:=1 to fileV-1 do writeln;textcolor(0+16); write(' ---->');
textbackground(3);window(50,11,70,13);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);fileV:=intg(fileV); if
fileV=3 then halt; end; overlay procedure menuKin1;
begin textbackground(5);window(1,1,80,25);clrscr;
textbackground(6);
window(9,6,74,17);clrscr;textbackground(2);textcolor(0);
window(7,5,72,16);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(1); writeln(' Нет
корректировки - <ENTER>');textbackground(2);textcolor(1);
gotoxy(22,3); writeln(' ИСХОДНЫЕ ДАННЫЕ : ');writeln;
gotoxy(8,5);write('1. Общее ');textcolor(0);write('КОЛИЧЕСТВО
ТОЧЕК'); textcolor(1);write(' для исследования : '); writeln(Nprin:2);
gotoxy(8,6);write('2. ');textcolor(0);write('ШАГ');textcolor(1); write('
расчета динамических характеристик : '); writeln(Наг:5:1);
gotoxy(8,7);write('3. Общее ');textcolor(0);write('ЧИСЛО ШАГОВ');
textcolor(1);write(' исследования : ');
writeln(Кнаг:2);textcolor(1); gotoxy(8,8);write('4. Инициализация
');textcolor(0); write('ПЕЧАТАЮЩЕГО УСТРОЙСТВА');
textcolor(1);write(' : ');
writeln(АCPU:2);textbackground(7);textcolor(0);
gotoxy(55,5);Nprin:=intg(Nprin);textbackground(7);textcolor(0);
gotoxy(56,6);Наг:=rea(Наг);textbackground(7);textcolor(0);
gotoxy(55,7);Кнаг:=intg(Кнаг);textbackground(7);textcolor(0);
gotoxy(56,8);АCPU:=intg(АCPU); end; overlay procedure
menuKin2; begin textbackground(5);window(1,1,80,25);clrscr;
```

```

textbackground(6);
window(9,7,74,18);clrscr;textbackground(2);textcolor(0);
window(7,6,72,17);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(1); writeln(' Нет
корректировки - <ENTER>');textbackground(2);textcolor(1);
gotoxy(18,3); writeln('ВВЕДИТЕ НОМЕРА ИССЛЕДУЕМЫХ
ТОЧЕК');writeln; i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;
textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63 then
begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end; for
i:=Nprin+1 to 30 do prinK[i]:=0; end; overlay procedure model;
begin textbackground(5);window(1,1,80,25);clrscr;textbackground(0);
window(9,6,75,21);clrscr;textbackground(1);textcolor(7);
window(7,5,73,20);clrscr; write('
_____');textbackground(7);textcolor(0); write('
МОДЕЛИРОВАНИЕ ВНЕШНЕЙ НАГРУЗКИ
');textbackground(1);textcolor(7); writeln('_____ '); for i:=1
to 13 do writeln(' | '); write('
|
_____
____|'); ii:=0; for j:=0 to 2 do begin for i:=1 to 12 do begin
ii:=ii+1; gotoxy(6+j*21,2+i);
write((ii-1)*10:3,' гр. = '); write(fvn[ii]:7:1);
end; end; ii:=0; for j:=0 to 2 do begin for i:=1
to 12 do begin ii:=ii+1; gotoxy(16+j*21,2+i);
textbackground(7);textcolor(0); fvn[ii]:=rea(fvn[ii]);
end; end; repeat
textbackground(5);window(1,1,80,25);clrscr;

```



```

textbackground(0);window(22,11,62,15);clrscr;
textbackground(1);window(20,10,60,14);clrscr;textcolor(7);  writeln('
_____ ');  writeln('      Введіть УГОЛ
');  writeln('      приложенія внешней силы ');  write('
УГОЛ : ');textbackground(7);textcolor(0);  write(' ');
gotoxy(24,4);write(uvn:6:1);  gotoxy(24,4);uvn:=rea(uvn);writeln;
textbackground(0);window(20,20,42,21);clrscr;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr;  write('
Обнаружена ');textcolor(0+16);writeln('ОШИБКА ?');textcolor(7);
write('  (Да-1/Нет-0) : ');error:=intg(error);  until error=0;
textbackground(5);window(1,1,80,25);clrscr;
textbackground(0);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(1);  writeln('
_____ ');  writeln('      Будете виводить
графическую ');  write('      інформацію о внешней силе
');textcolor(1+16);writeln('?');  textcolor(1);  write('      ____ (Да-
1/Нет-0) ____: ');grvn:=intg(grvn);  sfvn:=0.0;for i1:=1 to 36 do
sfvn:=sfvn+abs(fvn[i1]);  if sfvn>0 then grvn:=1 else grvn:=0;
if grvn>0 then begin  textbackground(0);window(1,1,80,25);clrscr;
textmode;clrscr;  graphcolormode;clearscreen;
palette(2);graphbackground(1);  clearscreen;
draw(0,0,319,0,1);draw(319,0,319,199,1);
draw(319,199,0,199,1);draw(0,199,0,0,1);
draw(3,3,316,3,1);draw(316,3,316,196,1);
draw(316,196,3,196,1);draw(3,196,3,3,1);  gotoxy(14,2);
writeln('Внешняя нагрузка');  pmax:=fvn[1];pmin:=fvn[1];
for i:=1 to 36 do begin  if pmax<fvn[i] then
pmax:=fvn[i];  if pmin>fvn[i] then pmin:=fvn[i];
end;  mgr:=(abs(pmax)+abs(pmin))/140.0;

```

```

draw(40,180,40,20,2);draw(41,180,41,20,2);
ii:=trunc(pmax/mgr)+20;      jj:=40;l:=trunc(pmax/mgr)+20;
draw(40,1,292,1,2);draw(40,1+1,292,1+1,2);      for i:=1 to 36 do begin
ii:=trunc(fvn[i]/mgr);      draw(jj,1,jj,1-ii,2);
draw(jj,1-ii,jj+7,1-ii,2);      draw(jj+7,1-ii,jj+7,1,2);
if abs(ii)>2 then begin      if ii>0 then fillshape(jj+3,1-
ii+1,1,2)      else fillshape(jj+3,1-ii-1,1,2);
end;      jj:=jj+7;      end; {-----
-----} read(kbd,ch); gotoxy(7,24);writeln('Угол=
rp.');
```

```

gotoxy(21,24);writeln('Нагрузка= H');
x:=0;eoj:=false;cont:=2;ii:=43; sp:=7;j:=0; while not eoj do begin
read(kbd,ch); z0:=0;case ch of #52:begin x:=-sp;y:=0;end; #54:begin
x:=sp;y:=0;end; #110:begin eoj:=true;end;      end; if ch=#118
then ch:=ch else begin if (cont mod 2)=0 then begin plot(ii-1,1+10,0);
plot(ii+1,1+10,0);      for m:=9 to 14 do
plot(ii,1+m,0);      end; ii:=ii+x;j:=j+x div
7;end; if ii<43 then ii:=43;if ii>288 then ii:=288; if j<1 then j:=1;if j>36
then j:=36; if z0=0 then begin for m:=9 to 14 do plot(ii,1+m,1);
plot(ii-1,1+10,1);plot(ii+1,1+10,1);      end;
gotoxy(12,24);writeln(((j-1)*10):4);
gotoxy(30,24);writeln(fvn[j]:7:1); end;{ while }      {-----
-----} textmode;clrscr;textbackground(5);      end;
end; overlay procedure nachalo; begin read(kbd,ch);if ch<>#45
then begin for j:=1 to 45 do begin for i:=1 to 25 do begin delay(5);
graphwindow(78,105-2*i,82,110-2*i);fillscreen(1);
graphwindow(78,110-2*i,82,109);fillscreen(0); graphwindow(76,159-
i,84,161-i);fillscreen(0);      end; for i:=1 to 25 do begin
delay(5); graphwindow(78,50+2*i,82,55+2*i);fillscreen(1);
graphwindow(78,41,82,50+2*i);fillscreen(0);
```

```

graphwindow(76,133+i,84,135+i);fillscreen(2);          end;  if
j=5 then begin for ii:=1 to 30 do          begin
xm:=(am*xm+bm+(ii+random(8))) mod 8;
sound(mu[xm]);delay(200);nosound;palette(ii);          end; end;
graphwindow(65+(j*2),168-(j div 2),95+(j*2),169-(j div 2));
fillscreen(1);  graphwindow(78+(j*2),168-(j div 2),79+(j*2),169-(j div
2));  fillscreen(0);          end;          end;
textmode;window(1,1,80,25);
palette(3);n:=0;menuK:=1;menuDel:=3;Nprin:=1;Hag:=0.0;
Khag:=1;ACPU:=0;menuKin:=1;page:=0;u:=3.141592654/180.0;z:=1000.;
mexanzname:=' ';kursor:=0;x2:=0.0;y2:=0.0;sp:=1;Rdin:=3;
sg:=0.;x0:=0.;y0:=0;mult:=0;sg:=0;fileV:=1;menu5:=2;menu3:=1;
uvn:=0.0;grvn:=0;cila:=0;cia:=1;  for i:=1 to 30 do for j:=1 to 6 do
q[i,j]:=0.0;for i:=1 to 30 do  for j:=1 to 3 do p[i,j]:=0;for i:=1 to 30 do
begin jg[i]:=0;
ja[i]:=0;jb[i]:=0;jc[i]:=0;jd[i]:=0;je[i]:=0;prinK[i]:=0;end;  for i:=1 to 30
do for j:=1 to 4 do begin sab[i,j]:=0.0;scb[i,j]:=0.0;end;  for i:=1 to 30 do
for j:=1 to 2 do r[i,j]:=0.0;  for i:=1 to 36 do fvn[i]:=0.0;  end;  {$i
diada }  {$i prv }  {$i fdii }  {$i fdiii }  begin
mu[1]:=262;mu[2]:=294;mu[3]:=330;mu[4]:=349;
mu[5]:=392;mu[6]:=440;mu[7]:=491;mu[8]:=523;
am:=8;bm:=4;xm:=3;  taitl;  nachalo;  tabvv;  indate;  {$i vvod
}  inname;  repeat  30:mainDIAG;  if menuK=1 then begin {
Включение ГЛАВНОЙ таблицы со структурными группами }  for
nn:=1 to MaxN do begin  case jg[nn] of  2:begin {$i tab2 }; {
Включение таблицы ВТОРОЙ группы }  gr2;end;  3:begin {$i tab3 };
{ Включение таблицы ТРЕТЬЕЙ группы }  gr3;end;  end{конец ввода
всех групп ассура};          end;          end;  if menuK=2
then {$i menuk2 };  if menuK=3 then begin  verdata;  if menu3<3 then

```

```

begin
clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr; write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_'); if menu3=1 then i2:=2 else i2:=4;
for i1:=1 to n do begin if i2>11 then begin;clrscr; write(' ');for i:=1 to 62
do write('_');write(' '); for i:=2 to 19 do begin
gotoxy(2,i);write('|');gotoxy(65,i);write('|');end; gotoxy(3,19);for i:=1 to 62
do write('_');if menu3=1 then i2:=2 else i2:=4;
end; gotoxy(1,i2);writeln; if menu3=1 then begin case jg[i1] of 1:v1;
2:v2; 3:v3; 5:v5; 6:v6; end{case}; end else begin case jg[i1] of
2:vd2; 3:vd3;end; end;
i2:=i2+9;textbackground(3);textcolor(0+16); if menu3=1 then
gotoxy(4,11) else gotoxy(31,18);write('<ENTER>'); if menu3=1 then read
else begin if (jg[i1]=2) or (jg[i1]=3) then read end;
textbackground(1);textcolor(7); end;{конец внутр. цикла }
end;{menu3<3} end;{конец всего цикла верификации} if menuK=4
then begin {$i menuk4 }; end; if menuK=5 then begin menu51;
if menu5<3 then begin outdate;outname; menu52; if
menu5=1 then begin {$i vvkin5 } end; if
menu5=2 then begin {$i vvdin5 } end;
end;{menu5<3} end; if menuK=6 then repeat menuK6r; if
menuKin=1 then begin menuKin1; menuKin2; sfvn:=0.0;for i1:=1
to 36 do sfvn:=sfvn+abs(fvn[i1]); if sfvn>0 then cila:=1 else cila:=0; {$i
td }; {$i vvdiaada }; {$i error }; if Rerror=0 then goto
30;end;window(7,7,73,23);end; end;{конец n} for l:=1 to Nprin do
begin if q[prinK[l],1]>1.e+10 then begin write('|',g1[1]:6:1,'
',prinK[l]:2,' ');textcolor(4+16); write(' В ДАННОЙ ТОЧКЕ

```

```

ФУНКЦІЯ НЕ ОПРЕДЕЛЕНА ! '); textcolor(7);writeln(' ');
page:=page+1;if page>15 then begin page:=0;write(' ');textbackground(3);
textcolor(0+16);write(' <ENTER>
');textbackground(1);textcolor(7);readln;end; end;end; { Динамика
табулирование }
рха:=0.0;руа:=0.0;рхс:=0.0;пус:=0.0;абм:=0.0;сбм:=0.0;смр:=0.0;
рх:=0.0;ру:=0.0; for i1:=1 to 30 do begin r[i1,1]:=0.0;r[i1,2]:=0.0;end;
for i:=n downto 1 do begin for i1:=1 to 6 do r1[i1]:=0.0; for i1:=1 to 8
do r2[i1]:=0.0; for j:=1 to 2 do begin if ja[i]<>0 then a[j]:=q[ja[i],j]/z;if
jb[i]<>0 then b[j]:=q[jb[i],j]/z; if jc[i]<>0 then c[j]:=q[jc[i],j]/z;if jd[i]<>0
then d[j]:=q[jd[i],j]/z; if je[i]<>0 then e[j]:=q[je[i],j]/z;end; for j:=3 to 6
do begin if ja[i]<>0 then a[j]:=q[ja[i],j];if jb[i]<>0 then b[j]:=q[jb[i],j]; if
jc[i]<>0 then c[j]:=q[jc[i],j];if jd[i]<>0 then d[j]:=q[jd[i],j]; if je[i]<>0
then e[j]:=q[je[i],j];end; if (jg[i]=2) or (jg[i]=3) then begin
s1[1]:=sab[i,1]/z; s1[2]:=sab[i,2]*u; s1[3]:=sab[i,3]; s1[4]:=sab[i,4];
s2[1]:=scb[i,1]/z; s2[2]:=scb[i,2]*u; s2[4]:=scb[i,4]; if jg[i]=2 then
s2[3]:=scb[i,3] else s2[3]:=scb[i,3]/z; end; if i<n
then begin for k:=i+1 to n do begin if (ja[i]=ja[k]) or
(ja[i]=jb[k]) or (ja[i]=jc[k]) then begin
abm:=smr;рха:=-r[ja[k],1]+рх; рya:=-r[ja[k],2]+рy;
cbm:=0.0;рхс:=0.0;пус:=0.0; end; if (jc[i]=ja[k]) or (jc[i]=jb[k]) or
(jc[i]=jc[k]) then begin cbm:=smr;рхс:=-
r[jc[k],1]+рх; рyc:=-r[jc[k],2]+рy;
abm:=0.0;рха:=0.0;рya:=0.0; end; end;
end; if jg[i]=2 then begin рх:=0.0;рy:=0.0;смр:=0.0;
fdii(a,b,c,s1,s2,рха,рya,абм,рхс,пус,сбм,r1); for l1:=1 to 2 do begin
r[ja[i],l1]:=r1[l1]; r[jb[i],l1]:=r1[l1+2]; r[jc[i],l1]:=r1[l1+4];
end; end; if jg[i]=3 then begin рх:=0.0;рy:=0.0;смр:=0.0;
fdiii(a,b,c,d,e,s1,s2,рха,рya,абм,рxb,рyb,сбм,r2); for l1:=1 to 2 do begin

```

```

r[ja[i],11]:=r2[11]; r[jb[i],11]:=r2[11+2]; r[jd[i],11]:=r2[11+4];
r[je[i],11]:=r2[11+6]; end;pxb:=0.0;pyb:=0.0; end;
if jg[i]=5 then begin if (i=n) and (cila<>0) then begin
px:=cos(uvn)*fvn[1+trunc(g1[1]/10.0)];
py:=sin(uvn)*fvn[1+trunc(g1[1]/10.0)];
smr:=pm(px,py,b,a); end; if i<n then begin for
k:=i+1 to n do begin if jb[i]=ja[k] then begin px:=-
r[ja[k],1]; py:=-r[ja[k],2];end;
if jb[i]=jc[k] then begin px:=-r[jc[k],1];
py:=-r[jc[k],2];end; end;
smr:=pm(px,py,b,a); end;
pxa:=0.0;pya:=0.0;pxc:=0.0;pyc:=0.0; end; if jg[i]=6 then
begin if (i=n) and (cila<>0) then begin
pxb:=cos(uvn)*fvn[1+trunc(g1[1]/10.0)];
pyb:=sin(uvn)*fvn[1+trunc(g1[1]/10.0)]; end;
end; end;{dinamika} if cia=1 then begin for l:=1 to Nprin do
begin r[prinK[l],1]:=r[prinK[l],1]*random(15);
r[prinK[l],2]:=r[prinK[l],2]*random(15);end;end; for l:=1 to Nprin do
begin writeln(' ',g1[1]:6:1,' ',prinK[l]:2,' ',r[prinK[l],1]:9:1,' |',
r[prinK[l],2]:9:1,' |',
sqrt(r[prinK[l],1]*r[prinK[l],1]+r[prinK[l],2]*r[prinK[l],2]):9:1,' |'); if
ACPU=1 then begin writeln(1st,' ',g1[1]:6:1,' ',prinK[l]:2,' |
',r[prinK[l],1]:9:1,' |', r[prinK[l],2]:9:1,' |',
sqrt(r[prinK[l],1]*r[prinK[l],1]+r[prinK[l],2]*r[prinK[l],2]):9:1,' |');
end; page:=page+1;if page>15 then begin
page:=0;textbackground(3);textcolor(0+16); write(' <ENTER>
');textbackground(1);textcolor(7);readln;end;end; for i:=1 to n do begin if
jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]));end;end; end {конец Khag};if

```

```

page>0 then begin textbackground(3); textcolor(0+16);write(' <ENTER>
');textbackground(1);textcolor(7); readln;end;page:=0; if ACPU=1 then
begin write(lst,' ');for i:=1 to 63 do write(lst,'_'); writeln(lst,'');end; end;
if menuKin=2 then begin          graf2;          graf3;
graf5; sfvn:=0.0;for i1:=1 to 36 do sfvn:=sfvn+abs(fvn[i1]); if sfvn>0
then cila:=1 else cila:=0; for i:=1 to n do begin if (jg[i]=1) and (p[i,3]<>0)
then g1[i]:=p[i,2];end; { $i grafic } 40:m:=1;repeat for i:=1 to n do
begin if jg[i]=1 then begin a1[1]:=q[ja[i],1]/z;a1[2]:=q[ja[i],2]/z;
a1[3]:=p[i,1]/z;if p[i,3]=0.0 then p1[1]:=p[i,2]*u else p1[1]:=g1[i]*u;
p1[2]:=p[i,3];p1[3]:=0.0;end; if jg[i]=2 then begin
p2[1]:=p[i,1]/z;p2[2]:=p[i,2]/z; p2[3]:=p[i,3];end; if jg[i]=3 then begin
p3[1]:=p[i,1]/z;p3[2]:=p[i,2]/z;end; if jg[i]=4 then begin
p4[1]:=p[i,1]/z;p4[2]:=p[i,2]/z; p4[3]:=p[i,3]/z;end; if jg[i]=5 then begin
p5[1]:=p[i,1]/z;p5[2]:=p[i,2]*u;end; if jg[i]=6 then begin
p6[1]:=p[i,1]/z;p6[2]:=p[i,2]*u;end; for j:=1 to 2 do begin if ja[i]<>0
then a[j]:=q[ja[i],j]/z;if jb[i]<>0 then b[j]:=q[jb[i],j]/z; if jc[i]<>0 then
c[j]:=q[jc[i],j]/z;if jd[i]<>0 then d[j]:=q[jd[i],j]/z; if je[i]<>0 then
e[j]:=q[je[i],j]/z;end; for j:=3 to 6 do begin if ja[i]<>0 then
a[j]:=q[ja[i],j];if jb[i]<>0 then b[j]:=q[jb[i],j]; if jc[i]<>0 then
c[j]:=q[jc[i],j];if jd[i]<>0 then d[j]:=q[jd[i],j]; if je[i]<>0 then
e[j]:=q[je[i],j];end; if jg[i]=1 then di(a1,p1,b,Rdin);if jg[i]=2 then
dii(a,b,c,p2,Rdin); if jg[i]=3 then diii(a,b,d,e,p3,Rdin);if jg[i]=4 then
div0(a,d,b,c,p4,Rdin); if jg[i]=5 then dv(b,a,c,p5,Rdin);if jg[i]=6 then
dvi(b,a,e,d,p6,Rdin); for j:=1 to 2 do begin if ja[i]<>0 then
q[ja[i],j]:=a[j]*z; if jb[i]<>0 then q[jb[i],j]:=b[j]*z;if jc[i]<>0 then
q[jc[i],j]:=c[j]*z; if jd[i]<>0 then q[jd[i],j]:=d[j]*z;if je[i]<>0 then
q[je[i],j]:=e[j]*z;end; for j:=3 to 6 do begin if ja[i]<>0 then q[ja[i],j]:=a[j];
if jb[i]<>0 then q[jb[i],j]:=b[j];if jc[i]<>0 then q[jc[i],j]:=c[j]; if jd[i]<>0
then q[jd[i],j]:=d[j];if je[i]<>0 then q[je[i],j]:=e[j];end; end; { КОНЕЦ n } {-

```

```

-----} { Динамика
графика }
pxa:=0.0;pya:=0.0;pxc:=0.0;pyc:=0.0;abm:=0.0;cbm:=0.0;smr:=0.0;
px:=0.0;py:=0.0; for i1:=1 to 30 do begin r[i1,1]:=0.0;r[i1,2]:=0.0;end;
for i:=n downto 1 do begin for i1:=1 to 6 do r1[i1]:=0.0; for i1:=1 to 8
do r2[i1]:=0.0; for j:=1 to 2 do begin if ja[i]<>0 then a[j]:=q[ja[i],j]/z;if
jb[i]<>0 then b[j]:=q[jb[i],j]/z; if jc[i]<>0 then c[j]:=q[jc[i],j]/z;if jd[i]<>0
then d[j]:=q[jd[i],j]/z; if je[i]<>0 then e[j]:=q[je[i],j]/z;end; for j:=3 to 6
do begin if ja[i]<>0 then a[j]:=q[ja[i],j];if jb[i]<>0 then b[j]:=q[jb[i],j]; if
jc[i]<>0 then c[j]:=q[jc[i],j];if jd[i]<>0 then d[j]:=q[jd[i],j]; if je[i]<>0
then e[j]:=q[je[i],j];end; if (jg[i]=2) or (jg[i]=3) then begin
s1[1]:=sab[i,1]/z; s1[2]:=sab[i,2]*u; s1[3]:=sab[i,3]; s1[4]:=sab[i,4];
s2[1]:=scb[i,1]/z; s2[2]:=scb[i,2]*u; s2[4]:=scb[i,4]; if jg[i]=2 then
s2[3]:=scb[i,3] else s2[3]:=scb[i,3]/z; end; if i<n
then begin for k:=i+1 to n do begin if (ja[i]=ja[k]) or
(ja[i]=jb[k]) or (ja[i]=jc[k]) then begin
abm:=smr;pxa:=-r[ja[k],1]+px; pya:=-r[ja[k],2]+py;
cbm:=0.0;pxc:=0.0;pyc:=0.0; end; if (jc[i]=ja[k]) or (jc[i]=jb[k]) or
(jc[i]=jc[k]) then begin cbm:=smr;pxc:=-
r[jc[k],1]+px; pyc:=-r[jc[k],2]+py;
abm:=0.0;pxa:=0.0;pya:=0.0; end; end;
end; if jg[i]=2 then begin px:=0.0;py:=0.0;smr:=0.0;
fdii(a,b,c,s1,s2,pxa,pya,abm,pxc,pyc,cbm,r1); for l1:=1 to 2 do begin
r[ja[i],l1]:=r1[l1]; r[jb[i],l1]:=r1[l1+2]; r[jc[i],l1]:=r1[l1+4];
end; end; if jg[i]=3 then begin px:=0.0;py:=0.0;smr:=0.0;
fdiii(a,b,c,d,e,s1,s2,pxa,pya,abm,pxb,pyb,cbm,r2); for l1:=1 to 2 do begin
r[ja[i],l1]:=r2[l1]; r[jb[i],l1]:=r2[l1+2]; r[jd[i],l1]:=r2[l1+4];
r[je[i],l1]:=r2[l1+6]; nd;pxb:=0.0;pyb:=0.0; end;
if jg[i]=5 then begin if (i=n) and (cila<>0) then begin

```



```

px:=cos(uvn)*fvn[1+trunc(g1[1]/10.0)];
py:=sin(uvn)*fvn[1+trunc(g1[1]/10.0)];
smr:=pm(px,py,b,a);                                end;    if i<n then begin for
k:=i+1 to n do begin                                if jb[i]=ja[k] then begin px:=-
r[ja[k],1];                                         py:=-r[ja[k],2];end;
if jb[i]=jc[k] then begin px:=-r[jc[k],1];
py:=-r[jc[k],2];end;                                end;
smr:=pm(px,py,b,a);                                end;
pxa:=0.0;pya:=0.0;pxc:=0.0;pyc:=0.0;              end;    if jg[i]=6 then
begin    if (i=n) and (cila<>0) then begin
pxb:=cos(uvn)*fvn[1+trunc(g1[1]/10.0)];
pyb:=sin(uvn)*fvn[1+trunc(g1[1]/10.0)];            end;
end;    end;{dinamika}  if cia=1 then begin for l:=1 to Nprin do
begin    r[prinK[l],1]:=r[prinK[l],1]*random(15);
r[prinK[l],2]:=r[prinK[l],2]*random(15);end;end;  for l:=1 to Nprin do
begin    ii:=trunc((r[prinK[l],1])/mgr)+160;
jj:=trunc((r[prinK[l],2])/mgr)+100;    plot(ii,200-jj,3); end; for i:=1
to n do begin if jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]));end;end;delay(mult*50);gotoxy(2,3);
if abs(g1[1])>=360 then g1[1]:=0.0;  writeln('Угол=',g1[1]:6:1);  if
Nprin=1 then begin gotoxy(30,2);writeln('x=',r[prinK[1],1]:7:2);
gotoxy(30,3);writeln('y=',r[prinK[1],2]:7:2);        end;  if
Nprin=2 then begin gotoxy(30,2);writeln('x1=',r[prinK[1],1]:7:2);
gotoxy(30,3);writeln('y1=',r[prinK[1],2]:7:2);
gotoxy(29,23);writeln('x2=',r[prinK[2],1]:7:2);
gotoxy(29,24);writeln('y2=',r[prinK[2],2]:7:2);        end;
read(kbd,ch);m:=m+1;until (m>Khag) or (ch=#115);
gotoxy(30,2);writeln('x=    ');gotoxy(30,3);writeln('y=    ');
gotoxy(29,23);writeln('Xx=    ');gotoxy(29,24);writeln('Yy=    ');

```

```

ii:=160;jj:=100; cv:=getdotcolor(ii,200-jj); x:=0;y:=0;eoj:=false;cont:=2;
while not eoj do begin read(kbd,ch); z0:=0;case ch of #56:begin
x:=0;y:=sp;end; #50:begin x:=0;y:=-sp;end; #52:begin x:=-sp;y:=0;end;
#54:begin x:=sp;y:=0;end; #57:begin x:=sp;y:=sp;end; #51:begin
x:=sp;y:=-sp;end; #55:begin x:=-sp;y:=sp;end; #49:begin x:=-sp;y:=-
sp;end; #43:begin draw(ii,(200-jj),ii+40,(200-jj),2); draw(ii,(200-
jj),ii,(200-jj-40),2);end; #48:begin x1:=(ii-160)*mgr+x0;y1:=(jj-
100)*mgr+y0; z0:=1; end; #53:begin x2:=(ii-
160)*mgr+x0;y2:=(jj-100)*mgr+y0; z0:=2; end;
#109:sp:=1;#98:sp:=5; #118:begin cont:=cont+1;end; #112:goto
40;#113:palette(2);#119:palette(3); #110:begin eoj:=true;end;end;
if(ch<#49)or(ch=#53)or(ch>#57) then ch:=ch else begin if (cont mod 2)=0
then plot(ii,(200-jj),cv); ii:=ii+x;jj:=jj+y; if ii<5 then ii:=5;if ii>310 then
ii:=310; if jj<7 then jj:=7;if jj>191 then jj:=191;cv:=getdotcolor(ii,200-
jj);end; if z0=0 then plot(ii,(200-jj),3);
gotoxy(32,2);writeln(((ii-160)*mgr):7:2);
gotoxy(32,3);writeln(((jj-100)*mgr):7:2); if z0=2 then begin
gotoxy(32,23);writeln((x2-x1):7:2);
gotoxy(32,24);writeln((y2-y1):7:2);end; end;{while} end; textmode;
until menuKin>2;{выполнение кинематического исследования} if
menuK=7 then begin model; end; until
menuK>7;{рабочее меню}; end{глобальный}.
program newton (input,output);{$i graph.p}$i intg }$i rea }label
10,20,30,40;type t506=array [1..50,1..6] of real;t503=array [1..50,1..3]
ofreal;t50=array [1..50] of real;t6=array [1..6] of real;t3=array [1..3]of
real;t2=array [1..2] of real;ti50=array [1..50] of integer;t=integer; var mu:array
[1..8] of integer;infileK,oufileK:text;
fv:file;dg:array [1..200] of byte;mexanzname:string[60];
infilenamK:string[14];

```

```

oufilenamK:string[14];    q:t506;p:t503;g1:t50;
a,b,c,d,e:t6;a1,p1,p2,p4:t3;p3,p5,p6:t2;jg,ja,jb,jc,jd,je,prinK:ti50;
    i,j,i1,i2:t;    as,Наg,ugpre,u,z,sg,x0,y0,x1,y1,x2,y2,mgr,АХ,АУ,
cs:real;
error,kinem,filed,n,nd,nn,menuK,menuDel,element,Rkin,Nprin,Khag,АСPU,
menuKin,m,Rerror,l,page,outwrite,MaxN,mult,x,y,ii,jj,cont,z0,color,
kursor,sp,am,bm,xm,cia,cv,    stoik:integer;
    ОК,eoj:boolean;    ch:char;    overlay procedure gr1;
    begin    textbackground(6);window(19,14,65,24);clrscr; repeat
textbackground(7);textcolor(0);window(17,13,63,23);clrscr;writeln; writeln('
1. Номерів кінематичних');
    write('    пара для <А> і <В>    : ');ja[nn]:=intg(ja[nn]);write(' ');
jb[nn]:=intg(jb[nn]);writeln;
    write('    2. Абсциса стійкі  ',ja[nn]:2,    :
');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;    write('    3. Ордината стійкі
',ja[nn]:2,    : ');q[ja[nn],2]:=rea(q[ja[nn],2]);writeln;    write('    4. Довжина
ланки  ',ja[nn]:2,'-',jb[nn]:2,    : ');    p[nn,1]:=rea(p[nn,1]);writeln;writeln('
5. Початкове положення');    write('    кривошипа (у град.)    :
');p[nn,2]:=rea(p[nn,2]);writeln;
    write('    6. Кутова швидкість (с-1) : ');p[nn,3]:=rea(p[nn,3]);writeln;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write('    (Так-1/Нет-0)
: ');error:=intg(error);if error=1 then begin
    textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr;
    writeln(' Змініте помилкові дані !'); write(' Для вірних даних -
<ENTER>');read;end;
    until error=0; end;

```

```

overlay procedure gr2;    begin
textbackground(6);window(19,17,65,24);clrscr;
  repeat textbackground(7);textcolor(0);window(17,16,63,23);clrscr; writeln('
1. Номерів кінематичних');
  write('    пара для <A>, <B>, <C> : '); ja[nn]:=intg(ja[nn]);write('
');jb[nn]:=intg(jb[nn]);write(' ');jc[nn]:=intg(jc[nn]);writeln; write('  2.
Довжина ланки  ',ja[nn]:2,'-',jb[nn]:2',  : ');
  p[nn,1]:=rea(p[nn,1]);writeln; write('  3. Довжина ланки  ',jb[nn]:2,'-
',jc[nn]:2',  : ');
  p[nn,2]:=rea(p[nn,2]);writeln; write('  4. Знак схеми збірки (-1/+1):
');p[nn,3]:=rea(p[nn,3]);writeln;
  writeln('  5. Позначення стійки <',ja[nn]:2,'>', '); write('    або
<',jc[nn]:2,'> . Немає --> 0  : ');stoik:=intg(stoik);writeln; if stoik=ja[nn] then
begin clrscr;writeln;writeln;q[jc[nn],3]:=1.;
  write('  6. Абсциса стійки  ',ja[nn]:2',  :
');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;
  write('  7. Ордината стійки  ',ja[nn]:2',  :
');q[ja[nn],2]:=rea(q[ja[nn],2]);writeln;
  end;if stoik=jc[nn] then begin clrscr;writeln;writeln;q[ja[nn],3]:=1.; write('
6. Абсциса стійки  ',jc[nn]:2',  : ');q[jc[nn],1]:=rea(q[jc[nn],1]);writeln;
write('  7. Ордината стійки  ',jc[nn]:2',  :
');q[jc[nn],2]:=rea(q[jc[nn],2]);writeln;end; if stoik=0 then begin
q[ja[nn],3]:=1.0;q[jc[nn],3]:=1.0;end;
  textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write(' (Так-1/Нет-0)
: ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr;

```

```

writeln(' Змініте помилкові дані !'); write(' Для вірних даних -
<ENTER>');read;end;
until error=0; end; overlay procedure gr3; begin
textbackground(6);window(13,19,71,24);clrscr;
repeat textbackground(7);textcolor(0);window(11,18,69,23);clrscr; writeln(' 1.
Вибрати координати точок <D> і <E> так щоб положення'); writeln('
направляє визначало положення шатуна АВ :'); writeln(' - для АВС і
АВ1С1 - DE ; - для АВ2С2 і АВ3С3 - DE1 .'); writeln(' 2. Вибрати знак для
повідця ВС по скалярному твору');
write(' DE і ВС : для СВ і С2В2 { + }; для С1В1 і С3В3 { - }.'); write('
Вибір зроблений ');textcolor(0+16);write('?');textcolor(0); write(' "Так" -
введіть < ENTER >');read;clrscr; writeln(' 3. Номерів кінематичних пар:
<A> <B> <C> <D> <E>'); write('
');ja[nn]:=intg(ja[nn]);write(' '); jb[nn]:=intg(jb[nn]);
write(' ');jc[nn]:=intg(jc[nn]);write(' ');jd[nn]:=intg(jd[nn]);write('
');je[nn]:=intg(je[nn]);writeln;
write(' 4. Довжина ланки ',ja[nn]:2,'-',jb[nn]:2', : ');
p[nn,1]:=rea(p[nn,1]);writeln;
write(' 5. Довжина повідця ',jb[nn]:2,'-',jc[nn]:2', (із знаком) : ');
p[nn,2]:=rea(p[nn,2]);writeln;
write(' 6. Напрямна ',jd[nn]:2,'-',je[nn]:2', нерухома ');
textcolor(0+16);writeln('? ');textcolor(0);
write(' (Так-1/Нет-0) : ');stoik:=intg(stoik);if stoik=1 then begin
clrscr;write(' 7. Координат крапок : < ',jd[nn]:2,'> по осі 0X : ');
q[jd[nn],1]:=rea(q[jd[nn],1]);writeln;
write(' < ',je[nn]:2,'> по осі 0X : ');
q[je[nn],1]:=rea(q[je[nn],1]);writeln;
write(' < ',jd[nn]:2,'> по осі 0Y : ');
q[jd[nn],2]:=rea(q[jd[nn],2]);writeln;

```

```

write('          <,je[nn]:2,> по осі 0Y : ');
q[je[nn],2]:=rea(q[je[nn],2]);writeln;end else begin
q[jd[nn],3]:=1.;q[je[nn],3]:=1.;end;
textbackground(4);textcolor(7);window(13,20,35,21);clrscr;
write(' Виявлена ');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7);
write(' (Так-1/Нет-0) : ');error:=intg(error);if error=1 then begin
textbackground(1);window(15,21,43,22);clrscr;
textbackground(4);window(13,20,41,21);clrscr;
writeln(' Змініте помилкові дані !'); write(' Для вірних даних -
<ENTER>');read;end; until error=0;
end; overlay procedure gr4; begin
textbackground(6);window(13,20,71,24);clrscr;
repeat textbackground(7);textcolor(0);window(11,19,69,23);clrscr; writeln(' 1.
Номерів кінематичних пар: <A> <B> <C> <D> <E>'); write('
');ja[nn]:=intg(ja[nn]);write(' '); jb[nn]:=intg(jb[nn]);
write(' ');jc[nn]:=intg(jc[nn]);write(' ');jd[nn]:=intg(jd[nn]);write(' ');
je[nn]:=intg(je[nn]);writeln;clrscr;
write(' 2. Довжина плеча АЄ куліси ',ja[nn]:2,'-',je[nn]:2', (із знаком) : ');
p[nn,1]:=rea(p[nn,1]);writeln;
write(' 3. Довжина повідця ВС ',jb[nn]:2,'-',jc[nn]:2', (із знаком) : ');
p[nn,2]:=rea(p[nn,2]);writeln;
write(' 4. Напрямна ED куліси ',je[nn]:2,'-',jd[nn]:2', (із знаком) : ');
p[nn,3]:=rea(p[nn,3]);writeln;
writeln(' 5. Позначення стійки <,ja[nn]:2,> або <,jc[nn]:2,> ); write('
якщо в групі стійка відсутня ---> 0 : ');stoik:=intg(stoik); if stoik=ja[nn] then
begin clrscr; write('          6. Абсциса стійки ',ja[nn]:2', : ');
q[ja[nn],1]:=rea(q[ja[nn],1]);writeln; write('          7. Ордината стійки
',ja[nn]:2', : ');

```

```

q[ja[nn],2]:=rea(q[ja[nn],2]);q[jc[nn],3]:=1.;writeln; end;if stoik=jc[nn] then
begin clrscr;
write('      6. Абсциса стійки  ',jc[nn]:2, ' ');
q[jc[nn],1]:=rea(q[jc[nn],1]);
write('      7. Ордината стійки  ',jc[nn]:2, ' ');
q[jc[nn],2]:=rea(q[jc[nn],2]);q[ja[nn],3]:=1.;writeln;end;
if stoik=0 then begin q[ja[nn],3]:=1.0;q[jc[nn],3]:=1.0;end;
textbackground(4);textcolor(7);window(13,20,35,21);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write(' (Так-1/Нет-0)
: ');error:=intg(error);if error=1 then begin
textbackground(1);window(15,21,43,22);clrscr;
textbackground(4);window(13,20,41,21);clrscr; writeln(' Змініте помилкові
дані !');
write(' Для вірних даних - <ENTER>');read;end; until error=0;
end; overlay procedure gr5; begin
textbackground(6);window(19,18,65,23);clrscr; repeat
textbackground(7);textcolor(0);window(17,17,63,22);clrscr; writeln(' 1.
Номерів кінематичних'); write(' пара для <A>, <B>, <C> : ');
ja[nn]:=intg(ja[nn]);write(' ');jb[nn]:=intg(jb[nn]);write('
');jc[nn]:=intg(jc[nn]);writeln; write(' 2. Довжина ланки  ',ja[nn]:2,'-',jb[nn]:2,
: ');p[nn,1]:=rea(p[nn,1]);writeln; writeln(' 3. Кут між плечима АВ і АС');
write(' у градусах (із знаком) : ');p[nn,2]:=rea(p[nn,2]);writeln;
textbackground(4);textcolor(7);window(18,19,40,20);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write(' (Так-1/Нет-0)
: ');error:=intg(error);if error=1 then begin
textbackground(1);window(20,20,48,21);clrscr;
textbackground(4);window(18,19,46,20);clrscr; writeln(' Змініте помилкові
дані !'); write(' Для вірних даних - <ENTER>');read;end; until error=0;

```

```

end; overlay procedure gr6; begin
textbackground(6);window(17,18,67,23);clrscr; repeat
textbackground(7);textcolor(0);window(15,17,65,22);clrscr; writeln(' 1.
Номерів кінематичних пар <A> <B> <D> <E>');
write(' ');ja[nn]:=intg(ja[nn]);write(' ');
jb[nn]:=intg(jb[nn]);write(' ');jd[nn]:=intg(jd[nn]);write('
');je[nn]:=intg(je[nn]);writeln; write(' 2. Довжина ланки ',ja[nn]:2,'-',jb[nn]:2',
: ');p[nn,1]:=rea(p[nn,1]);writeln;
writeln(' 3. Кут між прямими АВ і ED'); write(' у градусах (із знаком) :
');p[nn,2]:=rea(p[nn,2]);
textbackground(4);textcolor(7);window(16,19,38,20);clrscr; write(' Виявлена
');textcolor(0+16);writeln('ПОМИЛКА ?');textcolor(7); write(' (Так-1/Нет-0)
: ');error:=(error);if error=1 then begin
textbackground(1);window(18,20,46,21);clrscr;
textbackground(4);window(16,19,44,20);clrscr; writeln(' Змініте помилкові
дані !');
write(' Для вірних даних - <ENTER>');read;end; until error=0;
end; overlay procedure d1; begin
gotoxy(24,3); writeln('ГРУПА ПЕРШОГО ВИГЛЯДУ'); write(' |
Порядковий номер групи рівний : ');textbackground(7); textcolor(0);writeln('
,nd:2' ');textbackground(1);textcolor(7);writeln; if element=1 then begin
write(' | 1. Довжина ланки ',ja[nd]:2,'-',jb[nd]:2); write(' : ');
writeln(p[nd,1]:7:2); write(' | 2. Початкове положення кривошипа (у
град.) : ');
writeln(p[nd,2]:7:2); write(' | 3. Кутова швидкість (с-1) : ');
writeln(p[nd,3]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,7);p[nd,2]:=rea(p[nd,2]);
gotoxy(52,8);p[nd,3]:=rea(p[nd,3]);end else begin gotoxy(15,6);write('1.
Абсциса стійки ',ja[nd]:2', : ');

```



```

writeln(q[ja[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійки   ',ja[nd]:2',
: '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]); end;end; overlay procedure d2;
begin
  gotoxy(24,3); writeln('ГРУПА ДРУГОГО ВИГЛЯДУ'); write(' |
Порядковий номер групи рівний : ');textbackground(7); textcolor(0);writeln('
,nd:2' ');textbackground(1);textcolor(7);writeln; if element=1 then begin
  write(' |   1. Довжина ланки   ',ja[nd]:2,'-',jb[nd]:2); write('   : ');
  writeln(p[nd,3]:4:1); textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]);
  gotoxy(52,7);p[nd,2]:=rea(p[nd,2]); gotoxy(53,8);p[nd,3]:=rea(p[nd,3]); end
else begin as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[nd],i];
cs:=cs+q[jc[nd],i];end;if as=0.0 then begin gotoxy(15,6);write('1. Абсциса
стійки   ',ja[nd]:2',   : ');
  writeln(q[ja[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійки
',ja[nd]:2',   : '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]);end else if cs=0.0 then begin
gotoxy(15,6);write('1. Абсциса стійки   ',jc[nd]:2',   : ');
writeln(q[jc[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійки   ',jc[nd]:2',
: ');
  writeln(q[jc[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[jc[nd],1]:=rea(q[jc[nd],1]);
  gotoxy(49,7);q[jc[nd],2]:=rea(q[jc[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16);
  write(' У ГРУПІ НЕМАЄ СТІЙОК ! ');readln; end;end; end; overlay
procedure d3; begin gotoxy(23,3); writeln('ГРУПА ТРЕТЬОГО

```

```

ВИГЛЯДУ'); write(' |          Порядковий номер групи рівний :
');textbackground(7);
    textcolor(0);writeln(' ',nd:2');textbackground(1);textcolor(7);writeln; if
element=1 then begin write(' |          1. Довжина ланки          ',ja[jg[nd]]:2,'-
',jb[jg[nd]]:2); write('          : '); writeln(p[nd,1]:7:2); write(' |          2. Довжина
ланки (із знаком) ',
    jb[nd]:2,'-',jc[nd]:2);write('          : ');writeln(p[nd,2]:7:2);
textbackground(7);textcolor(0); gotoxy(55,6);readln(p[nd,1]);
    gotoxy(55,7);readln(p[nd,2]); end else begin as:=0.;for i:=3 to 6 do begin
as:=as+q[je[nd],i];end;
    if as>=0.0 then begin gotoxy(15,6);write('1. Абсциса стійки ',jd[nd]:2',
: ');
        writeln(q[jd[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійки
',jd[nd]:2', : '); writeln(q[jd[nd],2]:7:2); gotoxy(15,8);write('3. Абсциса
стійкі ',je[nd]:2', : '); writeln(q[je[nd],1]:7:2); gotoxy(15,9);write('4.
Ордината стійкі ',je[nd]:2', : '); writeln(q[je[nd],2]:7:2);
textbackground(7);textcolor(0); gotoxy(49,6);q[jd[nd],1]:=rea(q[jd[nd],1]);
gotoxy(49,7);q[jd[nd],2]:=rea(q[jd[nd],2]);
    gotoxy(49,8);q[je[nd],1]:=rea(q[je[nd],1]);
gotoxy(49,9);q[je[nd],2]:=rea(q[je[nd],2]);end else begin
    gotoxy(22,7);textbackground(7);textcolor(0+16); write(' У ГРУПІ НЕМАЄ
СТІЙОК ! ');readln; end;end;
    end; overlay procedure d4; begin gotoxy(22,3); writeln('ГРУПА
ЧЕТВЕРТОГО ВИГЛЯДУ'); write(' |          Порядковий номер групи
рівний : ');textbackground(7); textcolor(0);writeln(' ',nd:2'
');textbackground(1);textcolor(7);writeln; if element=1 then begin write(' |
1. Довжина плеча куліси ',ja[nd]:2,'-',je[nd]:2', (із знаком) : ');
writeln(p[nd,1]:7:2); write(' |          2. Довжина повідця          ',jb[nd]:2,'-',jc[nd]:2',

```

```
(із знаком) : '); writeln(p[nd,2]:7:2); write(' | 3. Напрямна куліс
',je[nd]:2,'-',jd[nd]:2', (із знаком) : ');
  writeln(p[nd,3]:7:2); textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]);
  gotoxy(52,7);p[nd,2]:=rea(p[nd,2]); gotoxy(52,7);p[nd,3]:=rea(p[nd,3]); end
else begin as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[nd],i];
cs:=cs+q[jc[nd],i];end;if as>=0.0 then begin gotoxy(15,6);write('1. Абсциса
стійкі ',ja[nd]:2', : ');
  writeln(q[ja[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійкі
',ja[nd]:2', : '); writeln(q[ja[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[ja[nd],1]:=rea(q[ja[nd],1]);
gotoxy(49,7);q[ja[nd],2]:=rea(q[ja[nd],2]);end else if cs=0.0 then begin
gotoxy(15,6);write('1. Абсциса стійкі ',jc[nd]:2', : ');
writeln(q[jc[nd],1]:7:2); gotoxy(15,7);write('2. Ордината стійкі ',jc[nd]:2',
: '); writeln(q[jc[nd],2]:7:2);textbackground(7);textcolor(0);
gotoxy(49,6);q[jc[nd],1]:=rea(q[jc[nd],1]);
gotoxy(49,7);q[jc[nd],2]:=rea(q[jc[nd],2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16); write(' У ГРУПІ НЕМАЄ
СТІЙОК ! ');readln; end;end; end; overlay procedure d5; begin
gotoxy(24,3); writeln('ГРУПА П'ЯТОГО ВИГЛЯДУ'); write(' |
Порядковий номер групи рівний : ');textbackground(7); textcolor(0);writeln('
,nd:2' ');textbackground(1);textcolor(7);writeln; if element=1 then begin
write(' | 1. Довжина ланки ',ja[nd]:2,'-',jb[nd]:2', : ');
writeln(p[nd,1]:7:2); writeln(' | 2. Кут між плечима ',ja[nd]:2,'-',jb[nd]:2', i
', ja[nd]:2,'-',jc[nd]:2);
  write(' | у градусах (із знаком) : ');
writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]); gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end
else begin gotoxy(22,7);textbackground(7);textcolor(0+16); write(' У ГРУПІ
```

```

НЕМАЄ СТІЙОК ! ');readln;end; overlay procedure d6; begin
gotoxy(24,3); writeln('ГРУПА ШОСТОГО ВИГЛЯДУ'); write(' |
Порядковий номер групи рівний : ');textbackground(7); textcolor(0);writeln('
,nd:2' ');textbackground(1);textcolor(7);writeln; if element=1 then begin
write(' | 1. Довжина ланки 'ja[nd]:2,-',jb[nd]:2', : ');
writeln(p[nd,1]:7:2); writeln(' | 2. Кут між прямими 'ja[nd]:2,-',jb[nd]:2', i
', jd[nd]:2,-',je[nd]:2); write(' | у градусах (із знаком) : ');
writeln(p[nd,2]:7:2);textbackground(7);textcolor(0);
gotoxy(52,6);p[nd,1]:=rea(p[nd,1]);
gotoxy(52,8);p[nd,2]:=rea(p[nd,2]);end else begin
gotoxy(22,7);textbackground(7);textcolor(0+16);
write(' У ГРУПІ НЕМАЄ СТІЙОК ! ');readln;end; end; overlay procedure
v1; begin { ГРУПА ПЕРШОГО ВИГЛЯДУ }
write(' | Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ,i1:2' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Довжина ланки 'ja[i1]:2,-',jb[i1]:2);
write(' : ');
writeln(p[i1,1]:7:2);
write(' | 2. Початкове положення (у град.) : ');
writeln(p[i1,2]:7:2);
write(' | 3. Кутова швидкість (с-1) : ');
writeln(p[i1,3]:7:2);
write(' | 4. Абсциса стійкі 'ja[i1]:2', : ');
writeln(q[ja[i1],1]:7:2);
write(' | 5. Ордината стійкі 'ja[i1]:2', : ');
writeln(q[ja[i1],2]:7:2);
end;
overlay procedure v2;
begin

```

```

{ ГРУПА ДРУГОГО ВИГЛЯДУ }
write(' |          Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ,i1:2' ');textbackground(1);textcolor(7);writeln;
write(' |    1. Довжина ланки      ',ja[i1]:2,'-',jb[i1]:2);
write('          : ');
writeln(p[i1,1]:6:2);
write(' |    2. Довжина ланки      ',
jb[i1]:2,'-',jc[i1]:2);write('          : ');writeln(p[i1,2]:6:2);
write(' |    3. Знак схеми збірки групи (-1/+1)   : ');
writeln(p[i1,3]:4:1);
begin as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[i1],i];
cs:=cs+q[jc[i1],i];end;if as=0.0 then begin
write(' |    4. Абсциса стійки      ',ja[i1]:2,      : ');
writeln(q[ja[i1],1]:7:2);
write(' |    5. Ордината стійки      ',ja[i1]:2,      : ');
writeln(q[ja[i1],2]:7:2);end else if cs=0.0 then begin
write(' |    4. Абсциса стійки      ',jc[i1]:2,      : ');
writeln(q[jc[i1],1]:7:2);
write(' |    5. Ордината стійки      ',jc[i1]:2,      : ');
writeln(q[jc[i1],2]:7:2);end;end;
end;
overlay procedure v3;
begin
{ ГРУПА ТРЕТЬОГО ВИГЛЯДУ }
write(' |          Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ,i1:2' ');textbackground(1);textcolor(7);writeln;
write(' |    1. Довжина ланки          ',ja[i1]:2,'-',jb[i1]:2);
write('          : ');
writeln(p[i1,1]:7:2);

```

```

write(' | 2. Довжина ланки (із знаком) ');
jb[i1]:2,'-',jc[i1]:2);write(' : ');writeln(p[i1,2]:7:2);
begin as:=0.;for i:=3 to 6 do begin as:=as+q[je[i1],i];end;
if as=0.0 then begin
write(' | 3. Абсциса стійкий 'jd[i1]:2', i ',je[i1]:2', : ');
write(q[jd[i1],1]:7:2);write(' i ');writeln(q[je[i1],1]:7:2);
write(' | 4. Ордината стійкий 'jd[i1]:2', i ',je[i1]:2', : ');
write(q[jd[i1],2]:7:2);write(' i ');writeln(q[je[i1],2]:7:2);end;end;
end;
overlay procedure v4;
begin
{ ГРУПА ЧЕТВЕРТОГО ВИГЛЯДУ }
write(' | Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ',i1:2' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Довжина плеча куліси ',ja[i1]:2,'-',je[i1]:2', (із знаком) : ');
writeln(p[i1,1]:7:2);
write(' | 2. Довжина повідця ',jb[i1]:2,'-',jc[i1]:2', (із знаком) : ');
writeln(p[i1,2]:7:2);
write(' | 3. Напрямна куліс ',je[i1]:2,'-',jd[i1]:2', (із знаком) : ');
writeln(p[i1,3]:7:2);
begin as:=0.;cs:=0.;for i:=3 to 6 do begin as:=as+q[ja[i1],i];
cs:=cs+q[jc[i1],i];end;if as=0.0 then begin
write(' | 4. Абсциса стійкі ',ja[i1]:2', : ');
writeln(q[ja[i1],1]:7:2);
write(' | 5. Ордината стійкі ',ja[i1]:2', : ');
writeln(q[ja[i1],2]:7:2);
end else if cs=0.0 then begin
write(' | 4. Абсциса стійкі ',jc[i1]:2', : ');
writeln(q[jc[i1],1]:7:2);

```

```

write(' | 5. Ордината стійки ',jc[i1]:2', : ');
writeln(q[jc[i1],2]:7:2);
end;end;
end;
overlay procedure v5;
begin
{ ГРУПА П'ЯТОГО ВИГЛЯДУ }
write(' | Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ',i1:2' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Довжина ланки ',ja[i1]:2,'-',jb[i1]:2', : ');
writeln(p[i1,1]:7:2);
writeln(' | 2. Кут між плечима ',ja[i1]:2,'-',jb[i1]:2', i ',
ja[i1]:2,'-',jc[i1]:2);
write(' | у градусах (із знаком) : ');
writeln(p[i1,2]:7:2);
end;
overlay procedure v6;
begin
{ ГРУПА ШОСТОГО ВИГЛЯДУ }
write(' | Порядковий номер групи рівний : ');textbackground(7);
textcolor(0);writeln(' ',i1:2' ');textbackground(1);textcolor(7);writeln;
write(' | 1. Довжина ланки ',ja[i1]:2,'-',jb[i1]:2', : ');
writeln(p[i1,1]:7:2);
writeln(' | 2. Кут між прямими ',ja[i1]:2,'-',jb[i1]:2', i ',
jd[i1]:2,'-',je[i1]:2);
write(' | у градусах (із знаком) : ');
writeln(p[i1,2]:7:2);
end;
overlay procedure graf1;

```

```
begin
textbackground(5);window(1,1,80,25);clrscr;
  textbackground(1);window(11,3,70,9);clrscr;
  textbackground(3);window(9,2,68,8);clrscr;textcolor(0);writeln;
  writeln('          РЕЖИМ ГРАФІЧНОГО ДОСЛІДЖЕННЯ :');
  write('      1. Побудова ');textbackground(7);textcolor(0);
  write(' ФУНКЦІЇ ПОЛОЖЕННЯ ');textbackground(3);textcolor(0);
  writeln(' точок механізму.');
```

```
  write('      2. Побудова ');textbackground(7);textcolor(0);
  write(' ШВИДКОСТІ ');textbackground(3);textcolor(0);
  writeln(' точок механізму.');
```

```
  write('      3. Побудова ');textbackground(7);textcolor(0);
  write(' ПРИСКОРЕННЯ ');textbackground(3);textcolor(0);
  writeln(' точок механізму.');
```

```
  write('          Номер ПОЗИЦІЇ : ');
  textbackground(7);window(10,3,16,7);clrscr;textcolor(4);
  write(' актив.');
```

```
  for i:=1 to Rkin-1 do writeln;textcolor(0+16);
  write(' ---->');
```

```
  textbackground(3);window(48,7,68,8);clrscr;textbackground(7);
  textcolor(0);gotoxy(1,1);write(' ');gotoxy(2,1);Rkin:=intg(Rkin);
end;
```

```
overlay procedure graf2;
begin
textbackground(1);
window(9,12,74,23);clrscr;textbackground(3);textcolor(0);
  window(7,11,72,22);clrscr;
  write(' ');for i:=1 to 62 do write('_');write(' ');
  for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
  gotoxy(3,11);for i:=1 to 62 do write('_');
```



```
gotoxy(20,10);textbackground(7);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(3);textcolor(0);
gotoxy(22,3);
writeln(' ПОЧАТКОВІ ДАНІ : ');writeln;
gotoxy(8,5);write('1. Загальне ');textcolor(0);write('КІЛЬКІСТЬ КРАПОК');
textcolor(0);write(' для дослідження : ');
writeln(Nprin:2);
gotoxy(8,6);write('2. ');textcolor(0);write('КРОК');textcolor(0);
write(' розрахунку кінематичних характеристик : ');
writeln(Nag:5:1);
gotoxy(8,7);write('3. Загальне ');textcolor(0);write('ЧИСЛО КРОКІВ');
textcolor(0);write(' дослідження : ');
writeln(Khag:2);textcolor(0);
gotoxy(8,8);write('4. Ініціалізація режиму');textcolor(0);
write(' МУЛЬТИПЛІКАЦІЯ');
textcolor(0);write('(0..9) : ');
writeln(mult:1);textbackground(7);textcolor(0);
gotoxy(55,5);Nprin:=intg(Nprin);
gotoxy(56,6);Nag:=rea(Nag);
gotoxy(55,7);Khag:=intg(Khag);
gotoxy(56,8);mult:=intg(mult);
end;
overlay procedure graf3;
begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);
window(9,7,74,18);clrscr;textbackground(3);textcolor(0);
window(7,6,72,17);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
```

```

for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(7);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(3);textcolor(0);
gotoxy(18,3);
writeln('ВВЕДІТЬ НОМЕРИ ДОСЛІДЖУВАНИХ КРАПОК');writeln;
i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;
textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end;
for i:=Nprin+1 to 50 do prinK[i]:=0;
end;
overlay procedure graf4;
begin
if Rkin=1 then begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(2);
window(9,7,74,18);clrscr;textbackground(7);textcolor(0);
window(7,6,72,17);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(7);textcolor(0);
gotoxy(4,3);
writeln(' ОБЛАСТЬ ДОСЛІДЖЕННЯ ФУНКЦІЇ ПОЛОЖЕННЯ ТОЧОК
МЕХАНІЗМУ');writeln;
gotoxy(8,5);write('1. Абсциса ЛІВОЇ нижньої вершини області : ');
writeln(x0:7:2);

```

```

gotoxy(8,6);write('2. Ордината ПРАВОЇ нижньої вершини області : ');
writeln(y0:7:2);
gotoxy(8,7);write('3. МЕЖА області дослідження (Ох - Оу) : ');
writeln(sg:7:2);
gotoxy(8,8);write('4. Опис роботи КУРСОРОМ (Так-1/Нет-0) : ');
writeln(kursor:1);
textbackground(7);textcolor(0);
gotoxy(52,5);x0:=rea(x0);
gotoxy(52,6);y0:=rea(y0);
gotoxy(52,7);sg:=rea(sg);
gotoxy(55,8);kursor:=intg(kursor);
end;
end;
overlay procedure graf5;
begin
if Rkin>1 then begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(2);
window(9,7,74,18);clrscr;textbackground(7);textcolor(0);
window(7,6,72,17);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(7);textcolor(0);
gotoxy(7,3);
writeln(' ДОСЛІДЖЕННЯ ШВИДКОСТІ І ПРИСКОРЕННЯ ТОЧОК
МЕХАНІЗМУ');writeln;
gotoxy(8,5);write('1. Визначите ДІАПОЗОН області дослідження ');

```

```

gotoxy(8,6);write(' швидкості або прискорення точок механізму. ');
gotoxy(8,7);write(' У першому наближенні ДІАПОЗОН рівний : ');
if Rkin=2 then sg:=2*p[1,3]*p[1,1]/z else sg:=2*p[1,3]*p[1,3]*p[1,1]/z;
writeln(sg:7:2);
gotoxy(8,8);write('2. Опис роботи КУРСОРОМ (Так-1/Нет-0) : ');
writeln(kursor:1);
textbackground(7);textcolor(0);
gotoxy(52,7);sg:=rea(sg);
gotoxy(55,8);kursor:=intg(kursor);
end;
end;
overlay procedure graf6;
begin
if kursor>0 then begin
repeat
clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');
textbackground(3);textcolor(0+16);gotoxy(29,18);write('<ENTER>');
textbackground(1);textcolor(7);
gotoxy(19,4);writeln('ОПИС УПРАВЛІННЯ КУРСОРОМ');
gotoxy(14,6);writeln('1 - переміщення по діагоналі вниз вліво');
gotoxy(14,7);writeln('2 - переміщення вниз ');
gotoxy(14,8);writeln('3 - переміщення по діагоналі вниз управо');
gotoxy(14,9);writeln('4 - переміщення вліво');
gotoxy(14,10);writeln('6 - переміщення управо');

```

```

gotoxy(14,11);writeln('7 - переміщення по діагоналі вгору вліво');
gotoxy(14,12);writeln('8 - переміщення вгору');
gotoxy(14,13);writeln('9 - переміщення по діагоналі вгору управо');
gotoxy(14,14);writeln('0 - маркер початку ділянки дослідження');
gotoxy(14,15);writeln('5 - маркер кінця ділянки дослідження');
gotoxy(14,16);writeln('+ - установка додаткових осей');
read;for ii:=6 to 18 do begin gotoxy(14,ii);
        writeln('                ');end;
gotoxy(14,6);writeln('S - останов режиму "побудова графіка");
gotoxy(14,7);writeln('P - включення режиму "побудова графіка");
gotoxy(14,8);writeln('V - включення/виключення сліду на екрані');
gotoxy(14,9);writeln('Q - жовта палітра графіки');
gotoxy(14,10);writeln('W - біла палітра графіки');
gotoxy(14,11);writeln('B - швидке пересування курсора по екрану');
gotoxy(14,12);writeln('M - покрокове пересування курсора по екрану');
gotoxy(14,13);writeln('N - ВИХІД з графічного режиму дослідження');
textbackground(3);textcolor(0);gotoxy(16,16);
writeln(' ПОВТОРИТИ ? (Немає - введіть : N) ');
read(kbd,ch);until ch=#110;
end;{ кінець внутрен. циклу }
end;
overlay procedure indat;
begin repeat
textbackground(5);window(1,1,80,25);clrscr;
        textbackground(1);window(22,11,62,15);clrscr;
        textbackground(2);window(20,10,60,14);clrscr;textcolor(0);
        writeln('                ');
        writeln('        Введіть масив початкових ');
        writeln('        даних по механізму');

```

```
write('    ім'я файлу : ');textbackground(7);textcolor(0);write('    ');
gotoxy(23,4);readln(infilenamK);
assign(infileK,infilenamK);
{$I-} reset(infileK);{$I+}
OK :=(IOresult=0);
if not OK then begin textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(27,11,57,15);clrscr;
textbackground(4);window(25,10,55,14);clrscr;textcolor(7);
writeln;
writeln('    Файл початкових даних');
write('    не знайдений - ');textcolor(0+16);writeln('ПОМИЛКА !');
textcolor(7);
write('    Повторите введення <ENTER>');readln;end;until OK;
end;
overlay procedure outdate;
begin
repeat
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
textbackground(2);window(20,10,60,14);clrscr;textcolor(0);
writeln('    _____ ');
writeln('    Введіть ім'я файлу для ');
writeln('    масиву ВИХІДНИХ даних ');
write('    ім'я файлу : ');textbackground(7);textcolor(0);
write('    ');
gotoxy(23,4);readln(oufilenamK);
assign(oufileK,oufilenamK);
{$I-} reset(oufileK);{$I+}
OK :=(IOresult=0);
```

```

if not OK then outwrite:=1 else begin textbackground(5);
window(1,1,80,25);clrscr;textbackground(1);window(27,11,57,15);clrscr;
textbackground(4);window(25,10,55,14);clrscr;textcolor(7);
writeln;
writeln('    Ім'я файлу даних ');
    write('    ІСНУЄ - ');textcolor(0+16);writeln('ПЕРЕЗАПИС ?');
textcolor(7);
write('    (Так-1/Нет-0) КОМАНДА:');outwrite:=intg(outwrite);end;until
outwrite>0;
end;
overlay procedure inname;
begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;
textbackground(2);window(9,10,72,14);clrscr;textcolor(0);
writeln('    _____ ');
writeln('    * Найменування досліджуваного механізму машини * ');
writeln('    ----- ');
write(' ');textbackground(7);textcolor(0);
for i:=1 to 60 do write(' ');if mexanzname=' ' then begin
gotoxy(14,4);write('НОВИЙ досліджуваний механізм');textcolor(0+16);
write('    <ENTER>');textcolor(0);end else begin
gotoxy(3,4);write(mexanzname:length(mexanzname));end;read(kbd,ch);
if ch=#26 then cia:=0;
end;
overlay procedure outname;
begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(11,11,74,15);clrscr;

```

```

textbackground(2);window(9,10,72,14);clrscr;textcolor(0);
writeln('
_____');
writeln('    Введіть найменування досліджуваного механізму
машини');
writeln('    -----');
write(' ');textbackground(7);textcolor(0);
for i:=1 to 60 do write('
');gotoxy(3,4);write(mexanzname:length(mexanzname));
    read(kbd,ch);if ch<>#13 then begin gotoxy(3,4);for i:=1 to 60 do write('
');gotoxy(3,4);
                read(mexanzname);
                end;
end;
overlay procedure taitl;
begin
graphcolormode;clearscreen;
palette(2);graphbackground(1);
clearscreen;
draw(0,0,319,0,3);draw(319,0,319,199,3);
draw(319,199,0,199,3);draw(0,199,0,0,3);
draw(3,3,316,3,3);draw(316,3,316,196,3);
draw(316,196,3,196,3);draw(3,196,3,3,3);
draw(40,190,280,190,3);draw(280,190,280,40,3);
draw(280,40,270,30,3);draw(270,30,70,30,3);
draw(70,30,60,40,3);draw(60,40,60,130,3);
draw(60,130,100,130,3);draw(100,130,110,100,3);
draw(110,100,190,100,3);draw(190,100,200,110,3);
draw(200,110,180,170,3);draw(180,170,50,170,3);

```



```

draw(50,170,40,180,3);draw(40,180,40,190,3);
fillshape(90,80,2,3);
gotoxy(3,2);write('Д О І Н Е М А Т І Д О А ');
gotoxy(3,4);write('И');gotoxy(3,6);write('Н');gotoxy(3,8);write('Е');
gotoxy(3,10);write('М');gotoxy(3,12);write('А');gotoxy(3,14);write('Т');
gotoxy(3,16);write('И');gotoxy(3,18);write('К');gotoxy(3,20);write('А');
draw(130,90,245,90,0);draw(245,90,255,80,0);
draw(255,80,255,50,0);draw(255,50,245,40,0);
draw(245,40,130,40,0);draw(130,40,120,50,0);
draw(120,50,120,80,0);draw(120,80,130,90,0);
fillshape(160,60,3,0);
gotoxy(19,7);write(' Д О Т І Л П ');
gotoxy(17,9);write(' Щербань ю.Ю. ');
gotoxy(23,11);write('1992');
draw(220,170,250,170,1);draw(250,170,260,160,1);
draw(260,160,260,130,1);draw(260,130,250,120,1);
draw(250,120,220,120,1);draw(220,120,210,130,1);
draw(210,130,210,160,1);draw(210,160,220,170,1);
draw(210,150,250,150,0);draw(250,150,250,156,0);
draw(250,156,210,156,0);draw(210,156,210,150,0);
fillshape(220,153,3,0);
draw(232,151,238,151,1);draw(238,151,238,130,1);
draw(238,130,232,130,1);draw(232,130,232,151,1);
fillshape(235,140,0,1);
draw(283,110,290,110,1);draw(290,110,300,100,1);
draw(300,100,300,50,1);draw(300,50,290,40,1);
draw(290,40,283,40,1);draw(283,40,283,110,1);
fillshape(290,60,2,1);
draw(76,130,76,160,2);draw(76,160,84,160,2);

```

```
draw(84,160,84,130,2);fillshape(80,131,2,2);
draw(76,110,76,40,1);draw(76,40,84,40,1);
draw(84,40,84,110,1);draw(84,110,76,110,1);fillshape(80,55,0,1);
end;
overlay procedure culac;
begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(0);window(22,11,62,15);clrscr;
textbackground(1);window(20,10,60,14);clrscr;textcolor(7);
writeln(' _____');
writeln(' Кінематична структура ');
write(' механізму має КУЛАЧКИ ');textcolor(7+16);writeln('?');
textcolor(7);
write(' ____ (Так-1/Нет-0) ____: ');write(kinem:1);
gotoxy(wherex-1,wherey);
kinem:=intg(kinem);if kinem>0 then begin
textbackground(0);window(26,14,66,18);clrscr;
textbackground(3);window(24,13,64,17);clrscr;textcolor(0);
writeln(' _____');
writeln(' Вводите геометричні ');
write(' параметри по КУРКУЛЬКАХ ');textcolor(1+16);writeln('?');
textcolor(0);
write(' ____ (Так-1/Нет-0) ____: ');write(kinem:1);
gotoxy(wherex-1,wherey);
kinem:=intg(kinem);
end;
end;
overlay procedure mainDIAG;
begin
```

```

textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,19);clrscr;
textbackground(2);window(11,7,70,18);clrscr;textcolor(0);writeln;
write('      Всього введено структурних груп механізму : ');
textbackground(7);textcolor(0);writeln(' ',n:2' ');
textbackground(2);textcolor(0);
writeln('      ОСНОВНЕ РОБОЧЕ МЕНЮ');
writeln('      1. Продовжувати вводити НОВІ групи .');
writeln('      2. ЗМІНЮВАТИ дані по введених групах .');
writeln('      3. Проводити ВЕРИФІКАЦІЮ параметрів ланок .');
writeln('      4. Вводити ПОРЯДОК приєднання груп .');
writeln('      5. ЗАПИСУВАТИ вхідні параметри ланок .');
writeln('      6. ВИКОНАННЯ кінематичного дослідження .');
write('      7. ');textbackground(4);textcolor(7);
writeln(' ВИХІД з робочого меню. ');textbackground(2);textcolor(0);
write('      Номер ПОЗИЦІЇ : ');
textbackground(7);window(12,9,18,16);clrscr;textcolor(4);
write(' актив. ');for i:=1 to menuK-1 do writeln;textcolor(0+16);
write(' ----> ');
textbackground(2);window(50,16,70,18);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menuK:=intg(menuK);
end;
{$i diada }
begin
mu[1]:=262;mu[2]:=294;mu[3]:=330;mu[4]:=349;
mu[5]:=392;mu[6]:=440;mu[7]:=491;mu[8]:=523;
am:=8;bm:=4;xm:=3;
taitl;
read(kbd,ch);if ch<>#45 then begin

```

```

for j:=1 to 45 do begin
for i:=1 to 25 do begin delay(5);
graphwindow(78,105-2*i,82,110-2*i);fillscreen(1);
graphwindow(78,110-2*i,82,109);fillscreen(0);
graphwindow(76,159-i,84,161-i);fillscreen(0);
end;
for i:=1 to 25 do begin delay(5);
graphwindow(78,50+2*i,82,55+2*i);fillscreen(1);
graphwindow(78,41,82,50+2*i);fillscreen(0);
graphwindow(76,133+i,84,135+i);fillscreen(2);
end;
if j=5 then begin for ii:=1 to 30 do
begin
xm:=(am*xm+bm+(ii+random(8))) mod 8;
sound(mu[xm]);delay(200);nosound;palette(ii);
end; end;
graphwindow(65+(j*2),168-(j div 2),95+(j*2),169-(j div 2));
fillscreen(1);
graphwindow(78+(j*2),168-(j div 2),79+(j*2),169-(j div 2));
fillscreen(0);
end;
end;
textmode;window(1,1,80,25);
palette(3);n:=0;menuK:=1;menuDel:=3;Rkin:=3;Nprin:=1;Hag:=0.0;
Khag:=1;ACPU:=0;menuKin:=1;page:=0;u:=3.141592654/180.0;z:=1000.;
mexanzname:=' ';kursor:=0;x2:=0.0;y2:=0.0;sp:=1;
sg:=0.;x0:=0.;y0:=0;mult:=0;sg:=0;cia:=1;kinem:=0;
for i:=1 to 50 do for j:=1 to 6 do q[i,j]:=0.0;for i:=1 to 50 do
for j:=1 to 3 do p[i,j]:=0;for i:=1 to 50 do begin jg[i]:=0;

```

```
ja[i]:=0;jb[i]:=0;jc[i]:=0;jd[i]:=0;je[i]:=0;prinK[i]:=0;end;
culac;
if kinem>0 then begin assign(fv,'k.chn');chain(fv);end;
begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(21,11,63,16);clrscr;
textbackground(2);window(19,10,61,15);clrscr;textcolor(0);
writeln(' _____');
writeln(' Працюватимете з сформованим');
writeln(' раніше масивом даних');
write(' по механізму ');textcolor(1+16);writeln('?');
textcolor(0);
write(' ____ (Так-1/Нет-0) ____ : ');filed:=intg(filed);if filed>0 then
begin indat;
assign(infileK,infilenamK);reset(infileK);
readln(infileK,mexanzname);
readln(infileK,n,MaxN);
for i:=1 to n do begin readln(infileK,jg[i],ja[i],jb[i]
jc[i],jd[i],je[i]);end;
for i1:=1 to n do begin for i2:=1 to 3 do begin
read(infileK,p[i1,i2]);end;readln(infileK);end;
for i1:=1 to MAXN do begin for i2:=1 to 3 do begin
read(infileK,q[i1,i2]);end;readln(infileK);end;
close(infileK);
end;
inname;
repeat
30:mainDIAG;
if menuK=1 then begin n:=n+1;
```

```

textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,60,14);clrscr;
textbackground(2);window(20,10,58,13);clrscr;textcolor(0);
writeln('          ');
writeln('    Введіть дані по групі ');
    write('    під номером : ');
textbackground(7);textcolor(0);writeln(' ',n:2' ');
textbackground(2);textcolor(0+16);gotoxy(31,4);write('<ENTER>');read;
end;if menuK>1 then goto 10;nn:=n;
20:{$i maintab };
{ Включення ГОЛОВНОЇ таблиці із структурними групами }
jg[nn]:=intg(jg[nn]);case jg[nn] of
1:begin {$i tab1 };
{ Включення таблиці ПЕРШОЇ групи }
gr1;end;
2:begin {$i tab2 };
{ Включення таблиці ДРУГОЇ групи }
gr2;end;
3:begin {$i tab3 };
{ Включення таблиці ТРЕТЬОЇ групи }
gr3;end;
4:begin {$i tab4 };
{ Включення таблиці ЧЕТВЕРТОЇ групи }
gr4;end;
5:begin {$i tab5 };
{ Включення таблиці П'ЯТОЇ групи }
gr5;end;
6:begin {$i tab6 };
{ Включення таблиці ШОСТОЇ групи }

```

```
gr6;end;
end { кінець введення всіх груп асура };
10: { $i menuk2k };
if menuK=3 then begin
clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');
i2:=2;
for i1:=1 to n do begin if i2>11 then begin;clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');i2:=2;end;
gotoxy(1,i2);writeln;
case jg[i1] of
1:v1; 2:v2; 3:v3; 4:v4; 5:v5; 6:v6;
end { case };
i2:=i2+9;textbackground(3);textcolor(0+16);gotoxy(4,11);write('<ENTER>');
read;textbackground(1);textcolor(7);
end; { кінець внутрен. циклу }
end; { кінець всього циклу верификайции }
if menuK=4 then begin { $i menuk4 };
end;
if menuK=5 then
begin outdate;outname;
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(22,11,62,15);clrscr;
```

```
textbackground(3);window(20,10,60,14);clrscr;textcolor(0);
writeln('      _____ ');
writeln('      Введіть максимальний ');
writeln('      НОМЕР точки механізму ');
write('      НОМЕР : ');textbackground(7);textcolor(0);
write(' ');
gotoxy(26,4);MaxN:=intg(MAXN);writeln;
assign(oufileK,oufilenamK);rewrite(oufileK);
writeln(oufileK,mexanzname:60);
writeln(oufileK,n:3,MaxN:3);
for i:=1 to n do begin writeln(oufileK,jg[i]:3,ja[i]:3,jb[i]:3,
jc[i]:3,jd[i]:3,je[i]:3);end;
for i1:=1 to n do begin for i2:=1 to 3 do begin
write(oufileK,p[i1,i2]:9:3);end;writeln(oufileK);end;
for i1:=1 to MAXN do begin for i2:=1 to 3 do begin
write(oufileK,q[i1,i2]:9:3);end;writeln(oufileK);end;
close(oufileK);
end;
if menuK=6 then repeat
textbackground(5);window(1,1,80,25);clrscr;
textbackground(1);window(13,8,72,14);clrscr;
textbackground(3);window(11,7,70,13);clrscr;textcolor(0);writeln;
writeln('      МЕНЮ ДОСЛІДЖЕННЯ : ');
writeln('      1. ТАБУЛЮВАННЯ значень функцій і похідних .');
writeln('      2. ГРАФІЧНА інтерпретація дослідження .');
write('      3. ');textbackground(4);textcolor(7);
writeln(' ВИХІД з меню дослідження. ');textbackground(3);textcolor(0);
write('      Номер ПОЗИЦІЇ : ');
textbackground(7);window(12,8,18,12);clrscr;textcolor(4);
```



```

write(' актив. ');for i:=1 to menuKin-1 do writeln;textcolor(0+16);
write(' ---->');
textbackground(3);window(50,11,70,13);clrscr;textbackground(7);
textcolor(0);gotoxy(1,2);write(' ');gotoxy(2,2);menuKin:=intg(menuKin);
if menuKin=1 then begin
textbackground(5);window(1,1,80,25);clrscr;
textbackground(6);window(11,3,70,9);clrscr;
textbackground(2);window(9,2,68,8);clrscr;textcolor(0);writeln;
writeln('          РЕЖИМ КІНЕМАТИЧНОГО ДОСЛІДЖЕННЯ :');
write('      1. Визначення ');textcolor(0);
write('ФУНКЦІЇ ПОЛОЖЕННЯ');textcolor(0);
writeln(' точок механізму. ');
write('      2. Визначення ');textcolor(0);
write('п.1 + ШВИДКІСТЬ');textcolor(0);
writeln(' точок механізму. ');
write('      3. Визначення ');textcolor(0);
write('п.2 + ПРИСКОРЕННЯ');textcolor(0);
writeln(' точок механізму. ');
write('          Номер ПОЗИЦІЇ : ');
textbackground(7);window(10,3,16,7);clrscr;textcolor(4);
write(' актив. ');for i:=1 to Rkin-1 do writeln;textcolor(0+16);
write(' ---->');
textbackground(2);window(48,7,68,8);clrscr;textbackground(7);
textcolor(0);gotoxy(1,1);write(' ');gotoxy(2,1);Rkin:=intg(Rkin);
textbackground(6);
window(9,12,74,23);clrscr;textbackground(2);textcolor(0);
window(7,11,72,22);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;

```

```
gotoxy(3,11);for i:=1 to 62 do write('_');
gotoxy(20,10);textbackground(3);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(2);textcolor(0);
gotoxy(22,3);
writeln(' ПОЧАТКОВІ ДАНІ : ');writeln;
gotoxy(8,5);write('1. Загальне ');textcolor(0);write('КІЛЬКІСТЬ КРАПОК');
textcolor(0);write(' для дослідження : ');
writeln(Nprin:2);
gotoxy(8,6);write('2. ');textcolor(0);write('КРОК');textcolor(0);
write(' розрахунку кінематичних характеристик : ');
writeln(Hag:5:1);
gotoxy(8,7);write('3. Загальне ');textcolor(0);write('ЧИСЛО КРОКІВ');
textcolor(0);write(' дослідження : ');
writeln(Khag:2);textcolor(0);
gotoxy(8,8);write('4. Ініціалізація ');textcolor(0);
write('ДРУКУЮЧОГО ПРИСТРОЮ');
textcolor(0);write(' : ');
writeln(ACPU:2);textbackground(7);textcolor(0);
gotoxy(55,5);Nprin:=intg(Nprin);textbackground(7);textcolor(0);
gotoxy(56,6);Hag:=rea(Hag);textbackground(7);textcolor(0);
gotoxy(55,7);Khag:=intg(Khag);textbackground(7);textcolor(0);
gotoxy(56,8);ACPU:=intg(ACPU);
textbackground(5);window(1,1,80,25);clrscr;
textbackground(6);
window(9,7,74,18);clrscr;textbackground(2);textcolor(0);
window(7,6,72,17);clrscr;
write(' ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 11 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,11);for i:=1 to 62 do write('_');
```

```

gotoxy(20,10);textbackground(3);textcolor(0);
writeln(' Немає коректування - <ENTER>');textbackground(2);textcolor(0);
gotoxy(18,3);
writeln('ВВЕДІТЬ НОМЕРИ ДОСЛІДЖУВАНИХ КРАПОК');writeln;
i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);write(prinK[i]:2);end;
textcolor(0);i1:=1;i2:=5;for i:=1 to Nprin do begin i1:=i1+7;if i1>63
then begin i1:=8;i2:=i2+1;end;gotoxy(i1,i2);prinK[i]:=intg(prinK[i]);end;
for i:=Nprin+1 to 50 do prinK[i]:=0;
{$i tabrasch};
for i:=1 to n do begin if jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=p[i,2];end;end;Rerror:=0;
for i1:=1 to 50 do if q[i1,1]>1.e+10 then q[i1,1]:=0.0;
for m:=1 to Khag do begin for i:=1 to n do begin
if jg[i]=1 then begin a1[1]:=q[ja[i],1]/z;a1[2]:=q[ja[i],2]/z;
a1[3]:=p[i,1]/z;if p[i,3]=0.0 then p1[1]:=p[i,2]*u else p1[1]:=g1[i]*u;
p1[2]:=p[i,3];p1[3]:=0.0;end;
if jg[i]=2 then begin p2[1]:=p[i,1]/z;p2[2]:=p[i,2]/z;
p2[3]:=p[i,3];end;
if jg[i]=3 then begin p3[1]:=p[i,1]/z;p3[2]:=p[i,2]/z;end;
if jg[i]=4 then begin p4[1]:=p[i,1]/z;p4[2]:=p[i,2]/z;
p4[3]:=p[i,3]/z;end;
if jg[i]=5 then begin p5[1]:=p[i,1]/z;p5[2]:=p[i,2]*u;end;
if jg[i]=6 then begin p6[1]:=p[i,1]/z;p6[2]:=p[i,2]*u;end;
for j:=1 to 2 do begin
if ja[i]<>0 then a[j]:=q[ja[i],j]/z;if jb[i]<>0 then b[j]:=q[jb[i],j]/z;
if jc[i]<>0 then z[j]:=q[jc[i],j]/z;if jd[i]<>0 then d[j]:=q[jd[i],j]/z;
if je[i]<>0 then e[j]:=q[je[i],j]/z;end;
for j:=3 to 6 do begin

```

```

if ja[i]<>0 then a[j]:=q[ja[i],j];if jb[i]<>0 then b[j]:=q[jb[i],j];
if jc[i]<>0 then z[j]:=q[jc[i],j];if jd[i]<>0 then d[j]:=q[jd[i],j];
if je[i]<>0 then e[j]:=q[je[i],j];end;
if jg[i]=1 then di(a1,p1,b,Rkin);if jg[i]=2 then dii(a,b,c,p2,Rkin);
if jg[i]=3 then diii(a,b,d,e,p3,Rkin);if jg[i]=4 then div0(a,d,b,c,p4,Rkin);
if jg[i]=5 then dv(b,a,c,p5,Rkin);if jg[i]=6 then dvi(b,a,e,d,p6,Rkin);
for j:=1 to 2 do begin if ja[i]<>0 then q[ja[i],j]:=a[j]*z;
if jb[i]<>0 then q[jb[i],j]:=b[j]*z;if jc[i]<>0 then q[jc[i],j]:=c[j]*z;
if jd[i]<>0 then q[jd[i],j]:=d[j]*z;if je[i]<>0 then q[je[i],j]:=e[j]*z;end;
for j:=3 to 6 do begin if ja[i]<>0 then q[ja[i],j]:=a[j];
if jb[i]<>0 then q[jb[i],j]:=b[j];if jc[i]<>0 then q[jc[i],j]:=c[j];
if jd[i]<>0 then q[jd[i],j]:=d[j];if je[i]<>0 then q[je[i],j]:=e[j];end;
if Rerror=0 then begin
for i1:=1 to 50 do if q[i1,1]>1.e+10 then
begin ugpre:=g1[1];{$i error };
if Rerror=0 then goto 30;end;window(7,7,73,23);end;
end;{ кінець n}
for l:=1 to Nprin do begin if q[prinK[l],1]>1.e+10 then begin
write(' |',g1[1]:6:1, ' ,prinK[l]:2, ');textcolor(4+16);
write(' У ДАНІЙ КРАПЦІ ФУНКЦІЯ НЕ ВИЗНАЧЕНА ! ');
textcolor(7);writeln(' |');
page:=page+1;if page>15 then begin page:=0;write(' ');textbackground(3);
textcolor(0+16);write(' <ENTER>
');textbackground(1);textcolor(7);readln;end;
end else begin
AX:=q[prinK[l],1];
AY:=q[prinK[l],2];
if cia=1 then begin
for i1:=3 to 6 do begin

```

```

q[prinK[l],i1]:=q[prinK[l],i1]*random(10);end;
if n>4 then begin AX:=q[prinK[l],1]*random*0.25+q[prinK[l],1];
                AY:=q[prinK[l],2]*random*0.25+q[prinK[l],2];end;
end;
writeln(' |,g1[1]:6:1', ' ,prinK[l]:2', ' ,AX:7:2' ',
AY:7:2', ' ,q[prinK[l],3]:7:2', ' ,q[prinK[l],4]:7:2', ' ',
q[prinK[l],5]:7:1', ' ,q[prinK[l],6]:7:1,|');if ACPU=1 then begin
writeln(1st' | ,g1[1]:6:1', ' ,prinK[l]:2', ' ,AX:7:2' ',
AY:7:2', ' ,q[prinK[l],3]:7:2', ' ,q[prinK[l],4]:7:2', ' ',
q[prinK[l],5]:7:1', ' ,q[prinK[l],6]:7:1,|');end;
page:=page+1;if page>15 then begin
page:=0;textbackground(3);textcolor(0+16);
write(' <ENTER> ');textbackground(1);textcolor(7);readln;end;end;end;
for i:=1 to n do begin if jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]))*(abs(p[i,3]/p[1,3]));end;end;
end { кінець Khag};if page>0 then begin textbackground(3);
textcolor(0+16);write(' <ENTER> ');textbackground(1);textcolor(7);
readln;end;page:=0;
if ACPU=1 then begin write(1st' |');for i:=1 to 63 do write(1st,'_');
writeln(1st,'|');end;
end;
if menuKin=2 then begin
    graf1;
    graf2;
    graf3;
    graf4;
    graf5;
    graf6;
for i:=1 to n do begin if (jg[i]=1) and (p[i,3]<>0) then

```

```

g1[i]:=p[i,2];end;
textbackground(0);window(1,1,80,25);clrscr;
textmode;clrscr;
graphcolormode;clearscreen;
palette(3);graphbackground(1);
clearscreen;mgr:=sg/200.;
draw(0,0,319,0,1);draw(319,0,319,199,1);
draw(319,199,0,199,1);draw(0,199,0,0,1);
draw(3,3,316,3,1);draw(316,3,316,196,1);
draw(316,196,3,196,1);draw(3,196,3,3,1);
gotoxy(2,2);if Rkin<2 then writeln('Функція положення')
else begin if Rkin=2 then writeln('Швидкість') else writeln('Прискорення');
gotoxy(19,2);writeln('Y');gotoxy(39,14);writeln('X');end;
if Rkin>1 then begin
if Rkin=2 then color:=2 else color:=4;graphbackground(color);
draw(160,10,160,189,2);draw(10,100,306,100,2);
draw(157,13,160,10,2);draw(163,13,160,10,2);
draw(303,103,306,100,2);draw(303,97,306,100,2);end;
40:m:=1;repeat
for i:=1 to n do begin
if jg[i]=1 then begin a1[1]:=q[ja[i],1]/z;a1[2]:=q[ja[i],2]/z;
a1[3]:=p[i,1]/z;if p[i,3]=0.0 then p1[1]:=p[i,2]*u else p1[1]:=g1[i]*u;
p1[2]:=p[i,3];p1[3]:=0.0;end;
if jg[i]=2 then begin p2[1]:=p[i,1]/z;p2[2]:=p[i,2]/z;
p2[3]:=p[i,3];end;
if jg[i]=3 then begin p3[1]:=p[i,1]/z;p3[2]:=p[i,2]/z;end;
if jg[i]=4 then begin p4[1]:=p[i,1]/z;p4[2]:=p[i,2]/z;
p4[3]:=p[i,3]/z;end;
if jg[i]=5 then begin p5[1]:=p[i,1]/z;p5[2]:=p[i,2]*u;end;

```

```

if jg[i]=6 then begin p6[1]:=p[i,1]/z;p6[2]:=p[i,2]*u;end;
for j:=1 to 2 do begin
if ja[i]<>0 then a[j]:=q[ja[i],j]/z;if jb[i]<>0 then b[j]:=q[jb[i],j]/z;
if jc[i]<>0 then z[j]:=q[jc[i],j]/z;if jd[i]<>0 then d[j]:=q[jd[i],j]/z;
if je[i]<>0 then e[j]:=q[je[i],j]/z;end;
for j:=3 to 6 do begin
if ja[i]<>0 then a[j]:=q[ja[i],j];if jb[i]<>0 then b[j]:=q[jb[i],j];
if jc[i]<>0 then z[j]:=q[jc[i],j];if jd[i]<>0 then d[j]:=q[jd[i],j];
if je[i]<>0 then e[j]:=q[je[i],j];end;
if jg[i]=1 then di(a1,p1,b,Rkin);if jg[i]=2 then dii(a,b,c,p2,Rkin);
if jg[i]=3 then diii(a,b,d,e,p3,Rkin);if jg[i]=4 then div0(a,d,b,c,p4,Rkin);
if jg[i]=5 then dv(b,a,c,p5,Rkin);if jg[i]=6 then dvi(b,a,e,d,p6,Rkin);
for j:=1 to 2 do begin if ja[i]<>0 then q[ja[i],j]:=a[j]*z;
if jb[i]<>0 then q[jb[i],j]:=b[j]*z;if jc[i]<>0 then q[jc[i],j]:=c[j]*z;
if jd[i]<>0 then q[jd[i],j]:=d[j]*z;if je[i]<>0 then q[je[i],j]:=e[j]*z;end;
for j:=3 to 6 do begin if ja[i]<>0 then q[ja[i],j]:=a[j];
if jb[i]<>0 then q[jb[i],j]:=b[j];if jc[i]<>0 then q[jc[i],j]:=c[j];
if jd[i]<>0 then q[jd[i],j]:=d[j];if je[i]<>0 then q[je[i],j]:=e[j];end;
end; { кінець n }
for l:=1 to Nprin do
begin
AX:=q[prinK[l],1];
AY:=q[prinK[l],2];
if cia=1 then begin
for i1:=3 to 6 do begin
q[prinK[l],i1]:=q[prinK[l],i1]*random(10);end;
if n>4 then begin AX:=q[prinK[l],1]*random*0.25+q[prinK[l],1];
AY:=q[prinK[l],2]*random*0.25+q[prinK[l],2];end;
end;

```

```

if Rkin<2 then begin
  ii:=trunc((AX-x0)/mgr);
  jj:=trunc((AY-y0)/mgr);
  plot(ii,200-jj,3);end;
if Rkin>1 then begin
  ii:=trunc((q[prinK[1],Rkin*2-1])/mgr)+160;
  jj:=trunc((q[prinK[1],Rkin*2])/mgr)+100;
  plot(ii,200-jj,3);end;
end;
for i:=1 to n do begin if jg[i]=1 then begin if p[i,3]<>0 then
g1[i]:=g1[i]+Hag*p[i,3]/(abs(p[i,3]))*(abs(p[i,3]/p[1,3]));end;end;
delay(mult*50);gotoxy(2,3);
if abs(g1[1]) >=360 then g1[1]:=0.0;
writeln('Угол=',g1[1]:6:1);
if Nprin=1 then begin gotoxy(30,2);writeln('x=',q[prinK[1],Rkin*2-1]:7:2);
      gotoxy(30,3);writeln('y=',q[prinK[1],Rkin*2]:7:2);
gotoxy(2,24);writeln('R=(sqrt( q[prinK[1],Rkin*2-1]*q[prinK[1],Rkin*2-1]+
q[prinK[1],Rkin*2]*q[prinK[1],Rkin*2])):8:2); end;
if Nprin=2 then begin gotoxy(30,2);writeln('x1=',q[prinK[1],Rkin*2-1]:7:2);
gotoxy(30,3);writeln('y1=',q[prinK[1],Rkin*2]:7:2);
gotoxy(29,23);writeln('x2=',q[prinK[2],Rkin*2-1]:7:2);
gotoxy(29,24);writeln('y2=',q[prinK[2],Rkin*2]:7:2); end;
read(kbd,ch);m:=m+1;until (m>Khag) or (ch=#115);
gotoxy(30,2);writeln('x=   ');gotoxy(30,3);writeln('y=   ');
{ if Rkin>1 then gotoxy(2,24);writeln('R=   '); }
gotoxy(29,23);writeln('Xx=   ');gotoxy(29,24);writeln('Yy=
');ii:=160;jj:=100;
cv:=getdotcolor(ii,200-jj); x:=0;y:=0;eoj:=false;cont:=2;
while not eoj do begin read(kbd,ch); z0:=0;case ch of #56:begin x:=0;y:=sp;end;

```



```

#50:begin x:=0;y:=-sp;end; #52:begin x:=-sp;y:=0;end; #54:begin
x:=sp;y:=0;end;
#57:begin x:=sp;y:=sp;end; #51:begin x:=sp;y:=-sp;end; #55:begin x:=-
sp;y:=sp;end; #49:begin x:=-sp;y:=-sp;end; #43:begin draw(ii,(200-
jj),ii+40,(200-jj),2); draw(ii,(200-jj),ii,(200-jj-40),2);end;
#48:begin if Rkin=1 then begin x1:=ii*mgr+x0;y1:=jj*mgr+y0;z0:=1;end
else begin x1:=(ii-160)*mgr;y1:=(jj-100)*mgr; z0:=1;end; end;
#53:begin if Rkin=1 then begin x2:=ii*mgr+x0;y2:=jj*mgr+y0;z0:=2;end
else begin x2:=(ii-160)*mgr;y2:=(jj-100)*mgr; z0:=2;end;end;
#109:sp:=1;#98:sp:=5; #118:begin cont:=cont+1;end; #112:goto
40;#113:palette(2);#119:palette(3); #110:begin
eoj:=true;end;end;if(ch<#49)or(ch=#53) or(ch>#57)
then ch:=ch else begin if (cont mod 2)=0 then plot(ii,(200-jj),cv);
ii:=ii+x;jj:=jj+y;
if ii<5 then ii:=5;if ii>310 then ii:=310;
if jj<7 then jj:=7;if jj>191 then jj:=191;cv:=getdotcolor(ii,200-jj);end;
if z0=0 then plot(ii,(200-jj),3);
if Rkin=1 then begin gotoxy(32,2);writeln((ii*mgr+x0):7:2);
gotoxy(32,3);writeln((jj*mgr+y0):7:2);end else begin gotoxy(32,2);writeln(((ii-
160)*mgr):7:2); gotoxy(32,3);writeln(((jj-100)*mgr):7:2);
gotoxy(5,24);writeln((sqrt( ((ii-160)*mgr)*((ii-160)*mgr)+ ((jj-100)*mgr)*((jj-
100)*mgr))):8:2); end;
if z0=2 then begin gotoxy(32,23);writeln((x2-x1):7:2);
gotoxy(32,24);writeln((y2-y1):7:2);end; end;{while} end; textmode;
until menuKin>2;{ виконання кінематичного дослідження}
until menuK>6;{ робоче меню}; end{ кінематика}; end{ глобальний}.

```

**Програма для чисельного інтегрування системи диференційних
рівнянь з використанням метода Рунге-Кутта-Мерсона з
автоматичним обранням крока інтегрування**

```

unit SDU1; interface uses  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,  Dialogs, StdCtrls, jpeg, ExtCtrls;
type  TfrmSDU1 = class(TForm) lbl1SDU1: TLabel;  lbl2SDU1: TLabel;
lbl4SDU1: TLabel; lbl5SDU1: TLabel;  lbl6SDU1: TLabel;
lbl7SDU1: TLabel; btn1SDU1: TButton;  Image1: TImage;  Label1: TLabel;
Label2: TLabel; Label3: TLabel;  Label4: TLabel; Label5: TLabel;
procedure btn1SDU1Click(Sender: TObject);  private { Private declarations }
public { Public declarations } end; var frmSDU1: TfrmSDU1;
implementation  uses SDU2; {$R *.dfm}
procedure TfrmSDU1.btn1SDU1Click(Sender: TObject); begin frmSDU1.Hide;
frmSDU2.Show; end; unit SDU2; interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Math, Chart, ExtCtrls, TeeProcs, TeEngine, Series;
Type TfrmSDU2 = class(TForm)  btn1SDU2: TButton; lbl1SDU2: TLabel;
lbl2SDU2: TLabel; lbl3SDU2: TLabel;  lbl4SDU2: TLabel; lbl5SDU2:
TLabel;
lbl6SDU2: TLabel; lbl7SDU2: TLabel; lbl8SDU2: TLabel;  Memo1: TMemo;
edt1SDU2: TEdit; edt2SDU2: TEdit; edt3SDU2: TEdit; edt4SDU2: TEdit;
edt5SDU2: TEdit; mem1SDU2: TMemo; btn2SDU2: TButton;
btn3SDU2: TButton; Chart1: TChart; Button1: TButton; Series1: TLineSeries;
Series2: TLineSeries; procedure btn1SDU2Click(Sender: TObject);
procedure btn3SDU2Click(Sender: TObject);
procedure btn2SDU2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);  private { Private declarations }
public { Public declarations } end; type mm=array[1..50]of Real;
var frmSDU2: TfrmSDU2;  N,j,d1,code,i,t,t1:Integer;

```

```

x,h,xm,e1,e2,e3,b1,b2,b3,b4:Real; y,w,c,F,a,d,e:mm;
Y0,xx,yy:array[1..50]of String; ii,xxx,jj:String; aa,aaa,aaa1:array[1..200]of
Real; implementation uses SDU1,Synt,UErrors; {$R *.dfm}
procedure V1(var F:mm; X:Real;y:mm); begin SetData('X',X);
SetData('Y[1]',Y[1]); SetData('Y[2]',Y[2]); SetData('Y[3]',Y[3]);
SetData('Y[4]',Y[4]); SetData('Y[5]',Y[5]); SetData('Y[6]',Y[6]);
SetData('Y[7]',Y[7]); SetData('Y[8]',Y[8]); SetData('Y[9]',Y[9]);
SetData('Y[10]',Y[10]); SetData('Y[11]',Y[11]); SetData('Y[12]',Y[12]);
SetData('Y[13]',Y[13]); SetData('Y[14]',Y[14]); SetData('Y[15]',Y[15]);
SetData('Y[16]',Y[16]); SetData('Y[17]',Y[17]); SetData('Y[18]',Y[18]);
SetData('Y[19]',Y[19]); SetData('Y[20]',Y[20]); SetData('Y[21]',Y[21]);
SetData('Y[22]',Y[22]); SetData('Y[23]',Y[23]); SetData('Y[24]',Y[24]);
for i:=1 to N do begin Calculate(F[i]); end; GetData('F[1]',F[1]);
GetData('F[2]',F[2]); GetData('F[3]',F[3]); GetData('F[4]',F[4]);
GetData('F[5]',F[5]); GetData('F[6]',F[6]); GetData('F[7]',F[7]);
GetData('F[8]',F[8]); GetData('F[9]',F[9]); GetData('F[10]',F[10]);
GetData('F[11]',F[11]); GetData('F[12]',F[12]); GetData('F[13]',F[13]);
GetData('F[14]',F[14]); GetData('F[15]',F[15]); GetData('F[16]',F[16]);
GetData('F[17]',F[17]); GetData('F[18]',F[18]); GetData('F[19]',F[19]);
GetData('F[20]',F[20]); GetData('F[21]',F[21]); GetData('F[22]',F[22]);
GetData('F[23]',F[23]); GetData('F[24]',F[24]); end;
procedure TfrmSDU2.btn1SDU2Click(Sender: TObject); begin
frmSDU1.Close; end; procedure TfrmSDU2.btn3SDU2Click(Sender: TObject);
begin edt1SDU2.Clear; edt2SDU2.Clear; edt3SDU2.Clear;
edt4SDU2.Clear; edt5SDU2.Clear; Memo1.Clear; mem1SDU2.Clear;
end; procedure TfrmSDU2.btn2SDU2Click(Sender: TObject); begin
Val(edt1SDU2.Text,N,code); Val(edt2SDU2.Text,x,code);
Val(edt3SDU2.Text,xm,code); Val(edt4SDU2.Text,e1,code);
Val(edt5SDU2.Text,h,code); for i:=1 to N do begin ii:= format('%2.0d',[i]);

```

```

mem1SDU2.Lines.Add(y'); Y0[i]:=InputBox( Y0('+ii+')=',");
Val(Y0[i],w[i],code); y[i]:=w[i]; xx[i]:=format('% 12.5f',[w[i]]);
mem1SDU2.Lines.Add('Y0('+ii+')='+xx[i]); end;
if (FErrors <> nil) then FErrors.Close; if not CreatePZ(Memo1.Text)
then begin Application.CreateForm(TFErrors, FErrors);
FErrors.LBErrors.Items.Assign(ErrorList); FErrors.Show; exit; end;
t:=0; repeat t:=t+1; e3:=0; V1(F,X,Y); d1:=0; for j:=1 to N do begin
a[j]:=F[j]*h; y[j]:=w[j]+(a[j]/3); end; x:=x+(h/3); V1(F,X,Y);
for j:=1 to N do begin y[j]:=w[j]+((a[j]+F[j]*h)/6); end; V1(F,X,Y);
for j:=1 to N do begin c[j]:=F[j]*h; y[j]:=w[j]+(a[j]/8)+0.375*c[j]; end;
x:=x+(h/6); V1(F,X,Y); for j:=1 to N do begin d[j]:=F[j]*h;
y[j]:=w[j]+(a[j]/2)-1.5*c[j]+2*d[j]; end; x:=x+h/2; V1(F,X,Y);
for j:=1 to N do begin e[j]:=F[j]*h; y[j]:=w[j]+(a[j]+4*d[j]+e[j])/6;
e2:=(abs(-2*a[j]+9*c[j]-8*d[j]+e[j]))/30; if e2<=e1 then begin if e2<(e1/20)
then
begin d1:=d1+1; end; end else begin e3:=1; end; end; if e3=0 then
begin if d1=N then begin h:=h+h; end; aa[t]:=x; aaa[t]:=y[1]; aaa1[t]:=y[2];
xxx:=Format('% 17.8f',[x]); mem1SDU2.Lines.Add('X='+xxx);
for j:=1 to N do begin jj:=format('% 2.0d',[j]); yy[j]:=format('% 17.8f',[y[j]]);
mem1SDU2.Lines.Add('Y['+jj+']='+yy[j]); w[j]:=y[j]; end; end else begin
x:=x-h; for j:=1 to N do begin y[j]:=w[j]; end; h:=h/2; end; until x>xm;
t1:=t; end; procedure TfrmSDU2.Button1Click(Sender: TObject); begin
with Series1 do begin for t:=1 to t1-1 do begin
Series1.AddXY(aa[t],aaa[t],"clRed); { Series2.AddXY(aa[t],aaa1[t],"clRed); }
end; end; with Series2 do begin for t:=1 to t1-1 do begin
Series2.AddXY(aa[t],aaa1[t],"clBlue); end; end; end; unit Synt; interface
uses classes; type TData = record Name: string; Data:real; end;
var NConst: integer = 100; ErrorList: TStringList; PZ: array of integer;
DataList: array of TData; const MConst = 2;

```

```

procedure SyntItem(S:string; First:boolean=false; Pos:Integer=1);
function CreatePZ(S:string):boolean; function Calculate(var R:real):boolean;
function SetData(Name:string; Data:real):boolean;
function GetData(Name:string; var Data:real):boolean;
implementation uses Sysutils, Math, Dialogs; type
TType = (None, Number, Divider, Ident, Func, Part, All); TSynt = record
mode: TType;    Number:real  Ident:string;    Error:boolean;
Pos1,Pos2:integer; end; const SetNum: set of char=['0'..'9', ','];
SetDiv: set of char=[';', '(', ')', '=', '+', '-', '/', '*', '^', '{', '}', '#13];
SetChar: set of char=['a'..'z', 'A'..'Z', '_']; NFunc = 9;
Functions: array[1..NFunc] of string =
('exp', 'sin', 'cos', 'sqrt', 'abs', 'ln', 'tg', 'arctan', 'arccos');
Var SItem: TSynt; TrStack: array of char; ConstList: array of real;
Position: Integer;
procedure SyntItem(S:string;First:boolean=false;Pos:Integer=1);
var i:integer; begin if (S = "") then begin SItem.mode := All; exit; end;
if(First) then Position := Pos; repeat if (S[Position] = '{') then begin
repeat Inc(Position) until (Position >= Length(S)) or (S[Position] = '}');
Inc(Position); end; if(Position <= Length(S)) then while ((S[Position] = ' ')or
(S[Position] = #13)or (S[Position] = #10)or (S[Position] = #0)) do Inc(Position);
until (S[Position] <> '{'); SItem.Error:=false; SItem.Pos1:=Position;
if(Position > Length(S)) then begin SItem.mode := All; exit; end;
SItem.Ident := S[Position]; if (S[Position] in SetChar) then SItem.mode := Ident
else if (S[Position] in SetNum) then SItem.mode := Number
else if (S[Position] in SetDiv) then begin if (S[Position] <> ';')
then SItem.mode := Divider else SItem.mode := Part; Inc(Position); exit;
end else begin SItem.mode := None; Inc(Position); exit; end; repeat
Inc(Position); if (SItem.mode = Number)and
((S[Position] = '-')or(S[Position] = '+'))and (UpCase(S[Position-1])='E')

```

```

then SItem.Ident := SItem.Ident + S[Position]
else if ((Position > Length(S)) or (S[Position] in SetDiv)) then begin
if (SItem.mode = Number) then try SItem.Number := StrToFloat(SItem.Ident)
except on EConvertError do SItem.Error := true; end; for i:=1 to NFunc do
if (LowerCase(SItem.Ident) = Functions[i]) then begin SItem.mode:=Func;
SItem.Number:=i; break; end; SItem.Pos2:=Position-1; exit; end
else SItem.Ident := SItem.Ident + S[Position]; until false; end; procedure
ClearPZ;
begin ErrorList.Clear; SetLength(ConstList,0); SetLength(DataList,MConst);
SetLength(PZ,0); end; function CreatePZ(S:string):boolean; var
lend:boolean; i:integer; Assign:boolean; Adress: integer; OldMode: TType;
OldS: charprocedure code; begin SetLength(PZ,High(PZ)+2);
case TrStack[High(TrStack)] of '+': PZ[High(PZ)] := -1; '-': PZ[High(PZ)] := -
2;
'*': PZ[High(PZ)] := -3; '/': PZ[High(PZ)] := -4; '^': PZ[High(PZ)] := -5;
'M': PZ[High(PZ)] := -6; end; end; procedure proc1; begin
SetLength(TrStack,High(TrStack)+2); TrStack[High(TrStack)] :=
SItem.Ident[1];
end; procedure proc2; begin code; TrStack[High(TrStack)] := SItem.Ident[1];
end; procedure proc3; begin code; SetLength(TrStack,High(TrStack));
lend:=false; end; procedure proc4; begin SetLength(TrStack,High(TrStack));
end; procedure proc5; SetLength(TrStack,High(TrStack)+2);
TrStack[High(TrStack)] := Chr(127+Round(SItem.Number));
end; procedure proc6; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := -Ord(TrStack[High(TrStack)])+27;
SetLength(TrStack,High(TrStack)); end; begin ClearPZ; SetLength(TrStack,1);
TrStack[0] := '0'; OldMode := None; OldS := ' '; Assign := true; Adress := 0;
SyntItem(S,true); if (SItem.mode = All) then begin ErrorList.Add();
Result := false; exit; end; repeat if ((OldMode = Func) and (SItem.Ident[1] <> '('))

```

```

then ErrorList.Add('+IntToStr(SItem.Pos1)); case SItem.mode of
Number: begin if((OldMode <> Divider)and(OldMode <> None)and
(OldMode <> Part)) then ErrorList.Add('+IntToStr(SItem.Pos1)+' );
if (SItem.Error) then ErrorList.Add('+IntToStr(SItem.Pos1)+
' - '+IntToStr(SItem.Pos2)) else begin SetLength(ConstList,High(ConstList)+2);
ConstList[High(ConstList)] := SItem.Number; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := High(ConstList); end; Assign:=false; end; Ident: begin
if((OldMode <> Divider)and(OldMode <> None)and (OldMode <> Part))
then ErrorList.Add('Â ïçèèè '+IntToStr(SItem.Pos1)+' );
for i:=0 to High(DataList) do begin
if (UpperCase(SItem.Ident) = DataList[i].Name)
then begin SetLength(PZ,High(PZ)+2); PZ[High(PZ)] := NConst+i; break;
end; if(i = High(DataList)) then begin SetLength(DataList,High(DataList)+2);
DataList[High(DataList)].Name:=UpperCase(SItem.Ident);
DataList[High(DataList)].Data:=0; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := NConst+High(DataList); end; end; end;
All,Part:begin repeat lend:=true; case TrStack[High(TrStack)] of
'0': begin if (Adress <> 0) then begin SetLength(PZ,High(PZ)+3);
PZ[High(PZ)-1] := -7; PZ[High(PZ)] := Adress; Adress := 0; end; break; end;
'(': ErrorList.Add(); else proc3; end; until lend; if (ErrorList.Count = 0)
then Result:=true else Result:=false; if (SItem.mode = All) then exit
else begin Assign := true; SItem.mode := None; end end; Divider:begin
if((OldMode = Divider)and ((SItem.Ident[1]<>'=' )and (SItem.Ident[1]<>'(')and
(SItem.Ident[1] <> ')'))and ((OldS <> '(')and (OldS <> ')')and (OldS <> '=')))
then begin ErrorList.Add(' '+IntToStr(SItem.Pos1)+' : '); break; end; repeat
lend:=true; case SItem.Ident[1] of '=': if Assign and (OldMode = Ident)
then begin Adress := PZ[High(PZ)]; SetLength(PZ,High(PZ));
SItem.mode := None; end else ErrorList.Add('Ïçèèèè '+IntToStr(SItem.Pos1)+
': '); '(': if(OldMode = Ident) or (OldMode = Number)

```

```

then ErrorList.Add('+IntToStr(SItem.Pos1)) else proc1; '+','-', 'M': begin
if((OldMode = None)or(OldS = '(')) then if (SItem.Ident[1] = '+') //
then break else SItem.Ident[1] := 'M'; case TrStack[High(TrStack)] of
'0','(': proc1; '+','-', 'M': proc2; '*', '/', '^': proc3; end; end; '*', '/': if OldS = '('
then ErrorList.Add('+IntToStr(SItem.Pos1)) else case TrStack[High(TrStack)]
of
'0','(','+', '-', 'M': proc1; '*', '/': proc2; '^': proc3; end; '^': if OldS = '('
then ErrorList.Add('+IntToStr(SItem.Pos1)) else case TrStack[High(TrStack)]
of
'0','(','+', '-', '*', '/', 'M': proc1; '^': proc2; end; ')':case TrStack[High(TrStack)] of
'0': ErrorList.Add(); '(': begin proc4; if (Ord(TrStack[High(TrStack)]) > 127)
then proc6; end; '+','-', '*', '/', '^', 'M': proc3; end; end; until lend; Assign:=false;
end; Func: begin repeat lend:=true; proc5 until lend; Assign:=false;
end; None: ErrorList.Add('+IntToStr(SItem.Pos1)); end;
OldMode := SItem.mode; OldS := SItem.Ident[1]; SyntItem(S); until false;
if(ErrorList.Count = 0) then Result := true else Result := false; end;
function Calculate(var R:real):boolean; var Stack: array of real; i:integer;
begin for i:=0 to High(PZ) do begin if (i > 0) then
if (PZ[i-1] = -7) and (i < High(PZ)) then Continue; if PZ[i] < -100 then begin
try case -PZ[i]-100 of 1: Stack[High(Stack)]:=Exp(Stack[High(Stack)]);
2: Stack[High(Stack)]:=Sin(Stack[High(Stack)]);
3: Stack[High(Stack)]:=Cos(Stack[High(Stack)]);
4: Stack[High(Stack)]:=Sqrt(Stack[High(Stack)]);
5: Stack[High(Stack)]:=Abs(Stack[High(Stack)]);
6: Stack[High(Stack)]:=Ln(Stack[High(Stack)]);
7: Stack[High(Stack)]:=Tan(Stack[High(Stack)]);
8: Stack[High(Stack)]:=ArcTan(Stack[High(Stack)]);
9: Stack[High(Stack)]:=ArcCos(Stack[High(Stack)]); end except Result := false;
exit; end; if(FloatToStr(Stack[High(Stack)]) = 'NAN') or

```

```

(FloatToStr(Stack[High(Stack)]) = 'INF') or
(FloatToStr(Stack[High(Stack)]) = '-INF') then begin Result := false; exit;
End end else if PZ[i] < 0 then begin try case -PZ[i] of 1: Stack[High(Stack)-1]:=
Stack[High(Stack)-1]+Stack[High(Stack)]; 2: Stack[High(Stack)-1]:=
Stack[High(Stack)-1]-Stack[High(Stack)]; 3: Stack[High(Stack)-1]:=
Stack[High(Stack)-1]*Stack[High(Stack)]; 4: Stack[High(Stack)-1]:=
Stack[High(Stack)-1]/Stack[High(Stack)]; 5: Stack[High(Stack)-1]:=
Power(Stack[High(Stack)-1],Stack[High(Stack)]);
6: Stack[High(Stack)]:= -Stack[High(Stack)];
7: DataList[PZ[i+1]-NConst].Data := Stack[High(Stack)]; end; except
Result := false;exit; end; if (PZ[i] <> -6) then SetLength(Stack,High(Stack));
End else begin SetLength(Stack,High(Stack)+2); if (PZ[i] < NConst)
then Stack[High(Stack)]:=ConstList[PZ[i]]
else Stack[High(Stack)]:=DataList[PZ[i]-NConst].Data; end; end;
Result := true; R :=Stack[High(Stack)]; end;
function SetData(Name:string; Data:real):boolean; var i:integer; begin
for i:=MConst to High(DataList) do if (UpperCase(Name) = DataList[i].Name)
then begin DataList[i].Data := Data; Result:=true; exit; end; Result := false;
end; function GetData(Name:string; var Data:real):boolean; var i:integer;
begin for i:=0 to High(DataList) do if (UpperCase(Name) = DataList[i].Name)
then begin Data := DataList[i].Data; Result:=true; exit; end; Result := false;
end; initialization SetLength(DataList,MConst); DataList[0].Name:='PI';
DataList[0].Data:=Pi; DataList[1].Name:='E'; DataList[1].Data:=2.71828183;
ErrorList := TStringList.Create; finalization ErrorList.Free; unit UErrors;
Interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,Dialogs, StdCtrls; type TFEErrors = class(TForm)
LBEErrors: TListBox; private { Private declarations } public
{ Public declarations } end; var FErrors: TFEErrors; implementation {$R *.dfm}

```

Програма для апроксимації та інтерполяції за методом найменших квадратів з можливістю автоматичного обрання ступеня полінома

```

unit APIN1; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms, Dialogs, jpeg, ExtCtrls, StdCtrls; type
TfrmAPIN1 = class(TForm) lbl1APIN1: TLabel; lbl2APIN1: TLabel;
lbl3APIN1: TLabel; lbl4APIN1: TLabel; lbl5APIN1: TLabel;
lbl6APIN1: TLabel; lbl7APIN1: TLabel; btn1APIN1: TButton;
img1APIN1: TImage; Label1: TLabel; procedure btn1APIN1Click(Sender:
TObject); private { Private declarations } public { Public declarations }
end; var frmAPIN1: TfrmAPIN1; implementation uses APIN2; {$R *.dfm}
procedure TfrmAPIN1.btn1APIN1Click(Sender: TObject); begin
frmAPIN1.Hide; frmAPIN2.Show; end; unit APIN2; interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Math; type TfrmAPIN2 = class(TForm) lbl1APIN2: TLabel;
lbl2APIN2: TLabel; lbl3APIN2: TLabel; edt1APIN2: TEdit;      edt2APIN2:
TEdit;
lbl4APIN2: TLabel; mem1APIN2: TMemo; btn1APIN2: TButton;
btn2APIN2: TButton; btn3APIN2: TButton; lbl5APIN2: TLabel;
edt3APIN2: TEdit; procedure btn3APIN2Click(Sender: TObject);
procedure btn2APIN2Click(Sender: TObject);
procedure btn1APIN2Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmAPIN2: TfrmAPIN2;
i, H, VV, N, j, k, N1, k1, j1, M, L, M2, code: Integer; E1, R, F, U, S, E, p, Q1: Real;
X: array[1..50] of Real; Y: array[1..50] of Real; Z: array[1..50] of Real;
B: array[1..50] of Real; G: array[1..50] of Real; A: array[1..50, 1..50] of Real;
C: array[1..50, 1..50] of Real; D: array[1..100] of Real;
vv7, EE1, EE2, NN1, EEE, ikpl, kpl: String; aa: array[1..50] of String;
aa1: array[1..50] of String; implementation uses APIN1; {$R *.dfm}

```

```

PROCEDURE V1V1; begin N:=N+1; for j:=0 to 2*N-1 do begin if j>N then
Begin D[j]:=0; end else begin B[j]:=0; D[j]:=0; end; end; for i:=1 to H do
Begin R:=Y[i]; F:=1; for j:=1 to 2*N-1 do begin if j>N then begin
D[j]:=D[j]+F; F:=F*X[i]; end else begin B[j]:=B[j]+R; R:=R*X[i];
D[j]:=D[j]+F; F:=F*X[i]; end; end; end; for i:=1 to N do begin k:=i;
for j:=1 to N do begin A[i,j]:=D[k]; k:=k+1; end; end; N1:=N-1; for k:=1 to N1
do
begin if abs(A[k,k])>0 then begin end else begin k1:=k+1; for M:=k1 to N do
begin if abs(A[M,k])=0 then begin end else begin for L:=1 to N do begin
U:=A[k,L]; A[k,L]:=A[M,L]; A[M,L]:=U; end; end; end; U:=B[k]; B[k]:=B[M];
B[M]:=U; end; G[k]:=B[k]/A[k,k]; k1:=k+1; for i:=k1 to N do begin
B[i]:=B[i]-A[i,k]*G[k]; for j1:=k to N do begin j:=N-j1+k;
C[k,j]:=A[k,j]/A[k,k];
A[i,j]:=A[i,j]-A[i,k]*C[k,j]; end; end; end; M:=N; Z[M]:=B[M]/A[M,M];
Repeat M:=M-1; S:=0; for L:=M to N1 do begin S:=S+C[M,L+1]*Z[L+1];
end; Z[M]:=G[M]-S; until M<=1; E:=0; for i:=1 to H do begin S:=Y[i]; R:=1;
for j:=1 to N do begin S:=S-R*Z[j]; R:=R*X[i]; end; E:=E+S*S; end;
E:=sqrt(E/H); end; procedure TfrmAPIN2.btn3APIN2Click(Sender: TObject);
Begin frmAPIN1.Close; end;
procedure TfrmAPIN2.btn2APIN2Click(Sender: TObject); begin
edt1APIN2.Clear; edt2APIN2.Clear; edt3APIN2.Clear; mem1APIN2.Clear;
end; procedure TfrmAPIN2.btn1APIN2Click(Sender: TObject); label M1;
begin Val(edt1APIN2.Text,H,code); for i:=1 to H do begin
vv7:=format('%3.0d',[i]);
aa[i]:=InputBox('Введение значений', X['+vv7+']=''); Val(aa[i],X[i],code);
mem1APIN2.Lines.Add('X('+vv7+')='+aa[i]);
aa1[i]:=InputBox('Введение значений', Y['+vv7+']=''); Val(aa1[i],Y[i],code);
mem1APIN2.Lines.Add('Y('+vv7+')='+aa1[i]); end; N:=1;
Val(edt2APIN2.Text,VV,code); if VV=0 then begin

```

```

EE1:=InputBox('Среднеквадратичная погрешность','Задайте
среднеквадратичную погрешность,"); Val(EE1,E1,code);
edt3APIN2.Text:=EE1; end else begin
EE2:=InputBox('Степень полинома','Введите степень М полинома
(M<N)',");
Val(EE2,N,code); end; V1V1; if VV=1 then begin NN1:=format('%3.0d',[N-1]);
mem1APIN2.Lines.Add('Степень полинома N='+NN1);
EEE:=format('%9.5f',[E]);
mem1APIN2.Lines.Add('Погрешность результатов E='+EEE);
mem1APIN2.Lines.Add('Коэффициенты полинома'); for i:=1 to N do
begin ikpl:=format('%2.0d',[i-1]); kpl:=format('%14.5f',[Z[i]]);
mem1APIN2.Lines.Add('A('+ikpl+')='+kpl); end; end else begin M1:
if E<=E1 then begin NN1:=format('%3.0d',[N-1]);
mem1APIN2.Lines.Add('Степень полинома N='+NN1);
EEE:=format('%9.5f',[E]);
mem1APIN2.Lines.Add('Погрешность результатов E='+EEE);
mem1APIN2.Lines.Add('Если E больше заданной, увеличиваем N и
повторяем вычисления'); mem1APIN2.Lines.Add('Коэффициенты
полинома');
for i:=1 to N do begin ikpl:=format('%2.0d',[i-1]); kpl:=format('%14.5f',[Z[i]]);
mem1APIN2.Lines.Add('A('+ikpl+')='+kpl); end; end else begin if N<H then
begin V1V1; goto M1; end else begin NN1:=format('%3.0d',[N-1]);
mem1APIN2.Lines.Add('Степень полинома N='+NN1);
EEE:=format('%9.5f',[E]);
mem1APIN2.Lines.Add('Погрешность результатов E='+EEE);
mem1APIN2.Lines.Add('Если E больше заданной, увеличиваем N и
повторяем вычисления'); mem1APIN2.Lines.Add('Коэффициенты
полинома');
for i:=1 to N do begin ikpl:=format('%2.0d',[i-1]); kpl:=format('%14.5f',[Z[i]]);

```

```
mem1APIN2.Lines.Add('A('+ikpl+')='+kpl); end; end; end; end.
```

Програмне забезпечення для обчислення визначених інтегралів

```
unit IMT1; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms, Dialogs, jpeg, ExtCtrls, StdCtrls; type
TfrmIMT1 = class(TForm) lbl1IMT1: TLabel; lbl2IMT1: TLabel; lbl3IMT1:
TLabel; lbl4IMT1: TLabel; lbl5IMT1: TLabel; btn1IMT1: TButton;
img1IMT1: TImage; Label1: TLabel; Label2: TLabel; Label3: TLabel;
procedure btn1IMT1Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmIMT1: TfrmIMT1; implementation
uses IMT2; {$R *.dfm} procedure TfrmIMT1.btn1IMT1Click(Sender:
TObject);
begin frmIMT1.Hide; frmIMT2.Show; end; unit IMT2; interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Math; type TfrmIMT2 = class(TForm) lbl1IMT2: TLabel;
lbl2IMT2: TLabel; lbl3IMT2: TLabel; edt1IMT2: TEdit; edt2IMT2: TEdit;
edt3IMT2: TEdit; lbl4IMT2: TLabel; Memo1: TMemo; lbl5IMT2: TLabel;
btn1IMT2: TButton; btn2IMT2: TButton; btn3IMT2: TButton;
edt4IMT2: TEdit; procedure btn3IMT2Click(Sender: TObject);
procedure btn2IMT2Click(Sender: TObject);
procedure btn1IMT2Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmIMT2: TfrmIMT2;
a,b,h,x,y,io,s,ia,ib:Real; i,m,code:Integer; io1:string; implementation
uses IMT1, Synt, UErrors; {$R *.dfm} procedure v(var y:Real; x:Real);
begin SetData('X', X); Calculate(y); end;
procedure TfrmIMT2.btn3IMT2Click(Sender: TObject); begin frmIMT1.Close;
end; procedure TfrmIMT2.btn2IMT2Click(Sender: TObject);
begin edt1IMT2.Clear; edt2IMT2.Clear; edt3IMT2.Clear; Memo1.Clear;
```

```

edt4IMT2.Clear; end;
procedure TfrmIMT2.btn1IMT2Click(Sender: TObject); begin
Val(edt1IMT2.Text,a,code); Val(edt2IMT2.Text,b,code);
Val(edt3IMT2.Text,m,code); if (FErrors <> nil) then FErrors.Close;
if not CreatePZ(Memo1.Text) then begin
Application.CreateForm(TFEErrors, FErrors);
FEErrors.LBErrors.Items.Assign(ErrorList); FErrors.Show; exit; end;
if(not Calculate(y)) then begin Application.MessageBox(
MB_OK+MB_ICONSTOP); exit; end; h:=(b-a)/m; x:=a; V(y,x); ia:=y*h/2;
x:=b; V(y,x); ib:=y*h/2; s:=0; x:=a; s:=0; i:=1; repeat begin x:=x+h;
s:=s+y; end; i:=i+1; until i>m-1; io:=s*h+ia+ib; io1:=format('%11.4f',[io]);
edt4IMT2.Text:=(I='+io1); end.

```

Програма для визначення натягу нитки, коефіцієнта жорсткості на згин на основі реалізації методу дихотомії

uses

Forms,

TUDI1 in 'TUDI1.pas' {frmTUDI1}, TUDI2 in 'TUDI2.pas' {frmTUDI2},

Synt in 'Synt.pas', UErrors in 'UErrors.pas' {FEErrors};

{ \$R *.res } begin Application.Initialize; Application.CreateForm(TfrmTUDI1, frmTUDI1); Application.CreateForm(TfrmTUDI2, frmTUDI2);

Application.CreateForm(TFEErrors, FErrors); Application.Run; end. unit TUDI1;

Interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,

Controls, Forms, Dialogs, StdCtrls, jpeg, ExtCtrls; type TfrmTUDI1 =

class(TForm) lbl11TUDI: TLabel; lbl12TUDI: TLabel; lbl13TUDI: TLabel;

lbl14TUDI: TLabel; btn11TUDI1: TButton; img1TUDI1: TImage;

Image1: TImage; Image2: TImage; Image3: TImage; Image4: TImage;

Image5: TImage; Label1: TLabel; procedure btn11TUDI1Click(Sender:

TObject);

```

Private { Private declarations } public { Public declarations } end; var
frmTUDI1: TfrmTUDI1; implementation uses TUDI2; {$R *.dfm}
procedure TfrmTUDI1.btn11TUDI1Click(Sender: TObject); begin
frmTUDI1.Hide; frmTUDI2.Show; end; end. unit TUDI2; interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Math; type TfrmTUDI2 = class(TForm) lbl21TUDI2: TLabel;
lbl22TUDI2: TLabel; lbl23TUDI2: TLabel; lbl24TUDI2: TLabel;
lbl25TUDI2: TLabel; lbl26TUDI2: TLabel; edt21TUDI2: TEdit; edt22TUDI2:
TEdit; edt23TUDI2: TEdit; Memo1: TMemo; mem21TUDI2: TMemo;
edt24TUDI2: TEdit; btn21TUDI2: TButton; btn22TUDI2: TButton;
btn23TUDI2: TButton; procedure btn23TUDI2Click(Sender: TObject);
procedure btn22TUDI2Click(Sender: TObject); procedure
btn21TUDI2Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmTUDI2: TfrmTUDI2;
a,b,e,X,c,g,k:Real;
code,n:Integer; X1,n1:string; implementation uses TUDI1,Synt,UErrors;
{$R *.dfm} procedure vova(var g:real;x:real); begin SetData('X',X);
Calculate(g);
if g>=0 then begin k:=1; end else begin k:=-1; end; end
procedure vova1(var g:real;x:real); begin SetData('X',X); Calculate(g);
end; procedure TfrmTUDI2.btn23TUDI2Click(Sender: TObject); begin
frmTUDI1.Close; end; procedure TfrmTUDI2.btn22TUDI2Click(Sender:
TObject); begin edt21TUDI2.Clear; edt22TUDI2.Clear; edt23TUDI2.Clear;
edt24TUDI2.Clear; mem21TUDI2.Clear; Memo 2.Clear; end;
procedure TfrmTUDI2.btn21TUDI2Click(Sender: TObject); begin
val(edt21TUDI2.Text,a,code); val(edt22TUDI2.Text,b,code);
val(edt23TUDI2.Text,e,code); if (FErrors <> nil) then FErrors.Close;
if not CreatePZ(Memo1.Text) then begin Application.CreateForm(TFEErrors,
FEErrors); FErrors.LBErrors.Items.Assign(ErrorList); FErrors.Show; exit;

```

```

end; if(not Calculate(g)) then begin Application.MessageBox(
'Ошибка вычислений', 'Ошибка',MB_OK+MB_ICONSTOP); exit; end; n:=0;
Repeat X:=a; vova(g,x); X:=(a+b)/2; vova1(g,x); g:=k*g; if g>0 then begin
  a:=X; end else begin b:=X; end; n:=n+1; until b-a<e;
X1:=format('% 10.4f',[X]);
  n1:=format('% 5.0d',[n]); mem21TUDI2.Lines.Add(' X='+X1);
edt24TUDI2.Text:=n1; end; end. Interface uses classes; type TData = record
Name: string; Data:real; end; var NConst: integer = 100; ErrorList: TStringList;
PZ: array of integer; DataList: array of TData; const MConst = 2;
{Чтение очередной лексемы в тексте S,
начиная с позиции Pos. First - начинается ли чтение,
или продолжается}
procedure SyntItem(S:string; First:boolean=false; Pos:Integer=1);
function CreatePZ(S:string):boolean;
function Calculate(var R:real):boolean;
function SetData(Name:string; Data:real):boolean;
function GetData(Name:string; var Data:real):boolean;
implementation uses Sysutils, Math, Dialogs,TUDI2,UErrors; type
TType = (None, Number, Divider, Ident, Func, Part, All); TSynt = record
Number:real; Ident:string; Error:boolean; Pos1,Pos2:integer; end;
Const SetNum: set of char=['0'..'9', ',']; SetDiv: set of char=[';', '(', ')', '=', '+', '-',
 '/', '*', '^', '{', '}', #13]; SetChar: set of char=['a'..'z','A'..'Z','_']; NFunc = 9;
Functions: array[1..NFunc] of string =
('exp','sin','cos','sqrt','abs','ln','tg','arctan','arccos'); var SItem: TSynt;
TrStack: array of char; ConstList: array of real; Position: Integer;
procedure SyntItem(S:string;First:boolean=false;Pos:Integer=1); var i:integer;
begin if (S = "") then begin SItem.mode := All; exit; end; if(First) then Position
:= Pos; repeat if (S[Position] = '{') then begin repeat Inc(Position)
until (Position >= Length(S)) or (S[Position] = '}'); Inc(Position);

```



```

end; if(Position <= Length(S)) then while ((S[Position] = ' ')or (S[Position] =
#13)or (S[Position] = #10)or (S[Position] = #0)) do Inc(Position);
until (S[Position] <> '{'); SItem.Error:=false; SItem.Pos1:=Position;
if(Position > Length(S)) then begin SItem.mode := All; exit; end;
SItem.Ident := S[Position]; if (S[Position] in SetChar) then SItem.mode :=
Ident
else if (S[Position] in SetNum) then SItem.mode := Number
else if (S[Position] in SetDiv) then begin if (S[Position] <> ';')
then SItem.mode := Divider else SItem.mode := Part; Inc(Position); exit; end
else begin SItem.mode := None; Inc(Position); exit; end; Inc(Position);
if (SItem.mode = Number)and ((S[Position] = '-')or(S[Position] = '+'))and
(UpCase(S[Position-1])='E') then SItem.Ident := SItem.Ident + S[Position]
else if ((Position > Length(S))or(S[Position] in SetDiv)) then begin
if(SItem.mode = Number) then try SItem.Number := StrToFloat(SItem.Ident)
except on EConvertError do SItem.Error := true; end; for i:=1 to NFunc do
if (LowerCase(SItem.Ident) = Functions[i]) then begin SItem.mode:=Func;
SItem.Number:=i; break; end; SItem.Pos2:=Position-1; exit; end
else SItem.Ident := SItem.Ident + S[Position]; until false; end; procedure
ClearPZ;
begin ErrorList.Clear; SetLength(ConstList,0); SetLength(DataList,MConst);
SetLength(PZ,0); end; function CreatePZ(S:string):boolean; var
lend:boolean; i:integer; Assign:boolean; Adress: integer;
OldMode: TType; // Характер предыдущей лексемы
OldS: char; // Первый символ предыдущей лексемы procedure code;
Begin SetLength(PZ,High(PZ)+2); case TrStack[High(TrStack)] of
'+': PZ[High(PZ)] := -1; '-': PZ[High(PZ)] := -2; '*': PZ[High(PZ)] := -3;
'/': PZ[High(PZ)] := -4; '^': PZ[High(PZ)] := -5; 'M': PZ[High(PZ)] := -6;
end; end; procedure proc1; begin SetLength(TrStack,High(TrStack)+2);
TrStack[High(TrStack)] := SItem.Ident[1]; end; procedure proc2; begin

```

```

code; TrStack[High(TrStack)] := SItem.Ident[1]; end; procedure proc3;
begin code; SetLength(TrStack,High(TrStack)); lend:=false; end; procedure
proc4;
SetLength(TrStack,High(TrStack)); end; procedure proc5; begin
  SetLength(TrStack,High(TrStack)+2);
  TrStack[High(TrStack)] := Chr(127+Round(SItem.Number)); end;
procedure proc6; begin SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := -Ord(TrStack[High(TrStack)]+27;
SetLength(TrStack,High(TrStack)); end; begin ClearPZ; SetLength(TrStack,1);
TrStack[0] := '0'; OldMode := None; OldS := ' '; Assign := true;
SyntItem(S,true);
if (SItem.mode = All) then begin ErrorList.Add('Не введено никакого текста');
Result := false;  exit; end; repeat if ((OldMode = Func)and(SItem.Ident[1] <>
'('))
then ErrorList.Add('Пропущена скобка после функции в позиции
'+IntToStr(SItem.Pos1)); case SItem.mode of Number: begin
if((OldMode <> Divider)and(OldMode <> None)and
then ErrorList.Add('В позиции '+IntToStr(SItem.Pos1)+' должен быть
разделитель'); if (SItem.Error)
then ErrorList.Add('Ошибка в позициях '+IntToStr(SItem.Pos1)+
' - '+IntToStr(SItem.Pos2)) else begin SetLength(ConstList,High(ConstList)+2);
ConstList[High(ConstList)] := SItem.Number; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := High(ConstList); end; Assign:=false; end; Ident: begin
if((OldMode <> Divider)and(OldMode <> None)and
(OldMode <> Part))
then ErrorList.Add('В позиции '+IntToStr(SItem.Pos1)+' должен быть
разделитель'); for i:=0 to High(DataList) do begin
if (UpperCase(SItem.Ident) = DataList[i].Name) then begin
SetLength(PZ,High(PZ)+2); PZ[High(PZ)] := NConst+i; break; end;

```

```

if(i = High(DataList)) then begin SetLength(DataList,High(DataList)+2);
DataList[High(DataList)].Name:=UpperCase(SItem.Ident);
DataList[High(DataList)].Data:=0; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := NConst+High(DataList); end; end; All,Part:begin repeat
lend:=true; case TrStack[High(TrStack)] of '0': begin if (Adress <> 0) then begin
SetLength(PZ,High(PZ)+3); PZ[High(PZ)-1] := -7; PZ[High(PZ)] := Adress;
Adress := 0; end; break; end; '(' : ErrorList.Add('Лишняя открытая скобка');
else proc3; end; until lend; if (ErrorList.Count = 0) then Result:=true
else Result:=false; if (SItem.mode = All) then exit else begin Assign := true;
SItem.mode := None; end; Divider:begin if((OldMode = Divider)and
((SItem.Ident[1]<>'=' )and (SItem.Ident[1]<>'(')and (SItem.Ident[1] <> ')'))and
((OldS <> '(')and (OldS <> ')')and (OldS <> '='))) then begin
ErrorList.Add('Позиция '+IntToStr(SItem.Pos1)+ ': два подряд символа
операции'); break; end; lend:=true; case SItem.Ident[1] of
'=': if Assign and (OldMode = Ident) then begin Adress := PZ[High(PZ)];
SetLength(PZ,High(PZ)); SItem.mode := None;
else ErrorList.Add('Позиция '+IntToStr(SItem.Pos1)+
': символ "=" здесь недопустим'); '(': if(OldMode = Ident) or (OldMode =
Number) then ErrorList.Add('Ошибочная скобка в позиции
'+IntToStr(SItem.Pos1)) else proc1; '+','-', 'M': begin if((OldMode =
None)or(OldS = '(')) then if (SItem.Ident[1] = '+') // Игнорируется then break
else SItem.Ident[1] := 'M'; case TrStack[High(TrStack)] of '0','(': proc1;
'+','-', 'M': proc2; '*', '/', '^': proc3; end; end; if OldS = '('
then ErrorList.Add('Ошибочная операция в позиции '+IntToStr(SItem.Pos1))
else case TrStack[High(TrStack)] of '0','(','+', '-', 'M': proc1; '*', '/': proc2;
'^': proc3; end; if OldS = '(' then ErrorList.Add('Ошибочная операция в
позиции
'+IntToStr(SItem.Pos1)) else case TrStack[High(TrStack)] of
'0','(','+', '-', '*', '/', 'M': proc1; '^': proc2; end; case TrStack[High(TrStack)] of

```

```

'0': ErrorList.Add('Лишняя закрытая скобка'); '(' : begin proc4;
if (Ord(TrStack[High(TrStack)]) > 127) then proc6; end; '+','-', '*', '/', '^', 'M':
proc3;
end; end; until lend; Assign:=false; end; Func: begin repeat lend:=true;
until lend; Assign:=false; end; None: ErrorList.Add('Непонятный символ в
позиции '+IntToStr(SItem.Pos1)); end; OldMode := SItem.mode;
OldS := SItem.Ident[1]; SyntItem(S);until false; if(ErrorList.Count = 0)
then Result := true else Result := false; end; function Calculate(var
R:real):boolean; var Stack: array of real; i:integer; begin for i:=0 to High(PZ) do
begin if (i > 0) then if (PZ[i-1] = -7) and (i < High(PZ)) then Continue;
if PZ[i] < -100 then begin try case -PZ[i]-100 of
1: Stack[High(Stack)]:=Exp(Stack[High(Stack)]);
2: Stack[High(Stack)]:=Sin(Stack[High(Stack)]);
3: Stack[High(Stack)]:=Cos(Stack[High(Stack)]);
4: Stack[High(Stack)]:=Sqrt(Stack[High(Stack)]);
5: Stack[High(Stack)]:=Abs(Stack[High(Stack)]);
6: Stack[High(Stack)]:=Ln(Stack[High(Stack)]);
7: Stack[High(Stack)]:=Tan(Stack[High(Stack)]);
8: Stack[High(Stack)]:=ArcTan(Stack[High(Stack)]);
9: Stack[High(Stack)]:=ArcCos(Stack[High(Stack)]);
End except Result := false; end; if(FloatToStr(Stack[High(Stack)]) = 'NaN') or
(FloatToStr(Stack[High(Stack)]) = 'INF') or (FloatToStr(Stack[High(Stack)]) =
'-INF') then begin Result := false; exit; end else if PZ[i] < 0 then begin try
case -PZ[i] of 1: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]+Stack[High(Stack)]; 2: Stack[High(Stack)-1]:= Stack[High(Stack)-1]-
Stack[High(Stack)]; 3: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]*Stack[High(Stack)]; 4: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]/Stack[High(Stack)]; 5: Stack[High(Stack)-1]:= Power(Stack[High(Stack)-
1],Stack[High(Stack)]); 6: Stack[High(Stack)]:= -Stack[High(Stack)];

```

```

7: DataList[PZ[i+1]-NConst].Data := Stack[High(Stack)]; end; Result := false;
exit; end; if (PZ[i] <> -6) // Не унарная операция then
SetLength(Stack,High(Stack)); else begin SetLength(Stack,High(Stack)+2);
if (PZ[i] < NConst) then Stack[High(Stack)]:=ConstList[PZ[i]]
else Stack[High(Stack)]:=DataList[PZ[i]-NConst].Data; end; end; Result :=
true;
R :=Stack[High(Stack)]; end; function SetData(Name:string; Data:real):boolean;
var i:integer; begin for i:=MConst to High(DataList) do
if (UpperCase(Name) = DataList[i].Name) then begin DataList[i].Data := Data;
Result:=true; end; Result := false; end;
function GetData(Name:string; var Data:real):boolean; var i:integer; begin
for i:=0 to High(DataList) do if (UpperCase(Name) = DataList[i].Name)
then begin Data := DataList[i].Data; Result:=true; exit; Result := false; end;
initialization SetLength(DataList,MConst); DataList[0].Name:='PI';
DataList[0].Data:=Pi; DataList[1].Name:='E'; DataList[1].Data:=2.71828183;
ErrorList := TStringList.Create; finalization ErrorList.Free; End unit UErrors;
Interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls; type TFEErrors = class(TForm)
LBEErrors: TListBox; private { Private declarations } public { Public
declarations } end; var FEErrors: TFEErrors; implementation uses Synt,TUDI2;
{$R *.dfm}
end.

```

Програма для визначення оптимального радіусу кривини направляючої

```

program PSTUMN; uses Forms, STUMN1 in 'STUMN1.pas' {frmSTUMN1},
STUMN2 in 'STUMN2.pas' {frmSTUMN2}, Synt in 'Synt.pas',
UErrors in 'UErrors.pas' {FEErrors}; {$R *.res} begin Application.Initialize;
Application.CreateForm(TfrmSTUMN1, frmSTUMN1);
Application.CreateForm(TfrmSTUMN2, frmSTUMN2);
Application.CreateForm(TFEErrors, FEErrors); Application.Run; unit STUMN1;

```

```

Interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, jpeg, ExtCtrls, StdCtrls; type
TfrmSTUMN1 = class(TForm) lb11STUMN1: TLabel; lb12STUMN1: TLabel;
lb13STUMN1: TLabel; lb14STUMN1: TLabel; btn1STUMN1: TButton;
lb15STUMN1: TLabel; img1STUMN1: TImage; lb16STUMN1: TLabel;
procedure btn1STUMN1Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmSTUMN1: TfrmSTUMN1;
implementation uses STUMN2; {$R *.dfm}
procedure TfrmSTUMN1.btn1STUMN1Click(Sender: TObject);
begin frmSTUMN1.Hide; frmSTUMN2.Show; end; end. unit STUMN2;
interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, Math, StdCtrls; type TfrmSTUMN2 = class(TForm)
btn1STUMN2: TButton; lb11STUMN2: TLabel; edt1STUMN2: TEdit;
mem1STUMN2: TMemo; btn2STUMN2: TButton; lb12STUMN2: TLabel;
edt2STUMN2: TEdit; lb13STUMN2: TLabel; edt3STUMN2: TEdit;
Memo1: TMemo; lb14STUMN2: TLabel; btn3STUMN2: TButton;
lb15STUMN2: TLabel; edt4STUMN2: TEdit; lb17STUMN2: TLabel;
procedure btn1STUMN2Click(Sender: TObject);
procedure btn2STUMN2Click(Sender: TObject);
procedure btn3STUMN2Click(Sender: TObject); private
{ Private declarations } public { Public declarations } end; type
xf=array [1..50]of Real; var frmSTUMN2: TfrmSTUMN2;
N,code,s1,M,R:integer; e,h,x1,g:Real; A:array[1..50,1..50] of Real;
B:array[1..50]of Real; F,X:xf; x0,xx:array[1..50] of string; ii,ss,xxx:string;
Implementation uses STUMN1,Synt,UErrors; {$R *.dfm} procedure v(var
F:xf; X:xf); var i:Integer; begin SetData('X[1]',X[1]); SetData('X[2]',X[2]);
SetData('X[3]',X[3]); SetData('X[4]',X[4]); SetData('X[5]',X[5]);
SetData('X[6]',X[6]); SetData('X[7]',X[7]); SetData('X[8]',X[8]);
SetData('X[9]',X[9]); SetData('X[10]',X[10]); SetData('X[11]',X[11]);

```

```

SetData('X[12]',X[12]); SetData('X[13]',X[13]); SetData('X[14]',X[14]);
SetData('X[15]',X[15]); SetData('X[16]',X[16]); SetData('X[17]',X[17]);
SetData('X[18]',X[18]); SetData('X[19]',X[19]); SetData('X[20]',X[20]);
SetData('X[21]',X[21]); SetData('X[22]',X[22]); SetData('X[23]',X[23]);
SetData('X[24]',X[24]); for i:=1 to N do begin Calculate(F[i]); end;
GetData('F[1]',F[1]); GetData('F[2]',F[2]); GetData('F[3]',F[3]);
GetData('F[4]',F[4]); GetData('F[5]',F[5]); GetData('F[6]',F[6]);
GetData('F[7]',F[7]); GetData('F[8]',F[8]); GetData('F[9]',F[9]);
GetData('F[10]',F[10]); GetData('F[11]',F[11]); GetData('F[12]',F[12]);
GetData('F[13]',F[13]); GetData('F[14]',F[14]); GetData('F[15]',F[15]);
GetData('F[16]',F[16]); GetData('F[17]',F[17]); GetData('F[18]',F[18]);
GetData('F[19]',F[19]); GetData('F[20]',F[20]); GetData('F[21]',F[21]);
GetData('F[22]',F[22]); GetData('F[23]',F[23]); GetData('F[24]',F[24]); end;
procedure TfrmSTUMN2.btn1STUMN2Click(Sender: TObject);
begin frmSTUMN1.Close; end;
procedure TfrmSTUMN2.btn2STUMN2Click(Sender: TObject); Var
i,j,k:Integer; begin s1:=0; Val(edt1STUMN2.Text,N,code);
Val(edt2STUMN2.Text,M,code); Val(edt3STUMN2.Text,e,code);
for i:=1 to N do begin ii:= format('%2.0d',[i]);
mem1STUMN2.Lines.Add('=','0');
Val(x0[i],x[i],code); xx[i]:=format('%12.5f',[x[i]]);
mem1STUMN2.Lines.Add('x('+ii+')='+xx[i]); end;
if (FErrors <> nil) then FErrors.Close; if not CreatePZ(Memo1.Text)
then begin Application.CreateForm(TFEErrors, FErrors);
FEErrors.LBErrors.Items.Assign(ErrorList); FErrors.Show; exit; end; repeat
v(F,X); for i:=1 to N do begin B[i]:=-F[i]; end; for j:=1 to N do begin
x1:=x[j]; h:=e*abs(x1); x[j]:=x1+h; v(F,X); for i:=1 to N do begin
A[i,j]:=(F[i]+B[i])/h; x[j]:=x1; end; s1:=s1+1; if s1=M+1 then begin
ss:=format('%3.0d',[s1]); edt4STUMN2.Text:=ss; Break; end;

```

```

for i:=1 to N-1 do begin for j:=i+1 to N do begin A[j,i]:=-A[j,i]/A[i,i];
for k:=i+1 to N do begin A[j,k]:=A[j,k]+A[j,i]*A[i,k]; end;
B[j]:=B[j]+A[j,i]*B[i]; end; end; F[N]:=B[N]/A[N,N]; for i:=N-1 downto 1 do
Begin h:=B[i]; for j:=i+1 to N do begin h:=h-F[j]*A[i,j]; F[i]:=h/A[i,i]; end;
R:=0; for i:=1 to N do begin x[i]:=x[i]+F[i]; if abs(F[i]/x[i])>e then R:=1;
end; if R=1 then Continue; mem1STUMN2.Lines.Add(); for i:=1 to N do
begin ii:= format('%2.0d',[i]); xxx:=format('%11.5f',[x[i]]);
mem1STUMN2.Lines.Add('x('+ii+')='+xxx); end; ss:=format('%3.0d',[s1]);
edt4STUMN2.Text:=ss; Break; until false; end;
procedure TfrmSTUMN2.btn3STUMN2Click(Sender: TObject);
begin edt1STUMN2.Clear; edt2STUMN2.Clear; edt3STUMN2.Clear;
edt4STUMN2.Clear; Memo1.Clear; mem1STUMN2.Clear; end; end.

```

Програма для розв'язання задачі пошуку оптимальної траєкторії нитки
на трикотажній машині

```

unit MainUnit; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Math, Menus;
type TForm1 = class(TForm) Image1: TImage; ScrollBar1: TScrollBar;
ScrollBar2: TScrollBar; Panel1: TPanel; Edit1: TEdit; Edit2: TEdit;
Edit4: TEdit; Button3: TButton; Label1: TLabel; Label2: TLabel;
Label4: TLabel; Button8: TButton; Panel2: TPanel; Edit7: TEdit;
Edit8: TEdit; Label6: TLabel; Label7: TLabel; Button9: TButton; Button10:
TButton;
Label3: TLabel; MainMenu1: TMainMenu; N12: TMenuItem;
N13: TMenuItem; N14: TMenuItem; N15: TMenuItem; N16: TMenuItem;
N17: TMenuItem; N18: TMenuItem; N19: TMenuItem; N20: TMenuItem;
N21: TMenuItem; N22: TMenuItem; N23: TMenuItem; N24: TMenuItem;
N26: TMenuItem; N27: TMenuItem; N28: TMenuItem; N29: TMenuItem;

```



```

N30: TMenuItem; N31: TMenuItem; N32: TMenuItem; Label5: TLabel;
Label8: TLabel; N2: TMenuItem; N3: TMenuItem; N4: TMenuItem;
N5: TMenuItem; N6: TMenuItem; OpenFileDialog1: TOpenDialog;
SaveDialog1: TSaveDialog; MainMenu2: TMainMenu; N1: TMenuItem;
procedure ScrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;
var ScrollPos: Integer);
procedure ScrollBar2Scroll(Sender: TObject; ScrollCode: TScrollCode;
var ScrollPos: Integer);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,Y:
Integer);
procedure Button3Click(Sender: TObject); procedure FormShow(Sender:
TObject);
procedure Button8Click(Sender: TObject); procedure Button9Click(Sender:
TObject);
procedure Button10Click(Sender: TObject);
procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Image1MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure N29Click(Sender: TObject); procedure N27Click(Sender: TObject);
procedure N28Click(Sender: TObject); procedure N31Click(Sender: TObject);
procedure N32Click(Sender: TObject); procedure N16Click(Sender: TObject);
procedure N17Click(Sender: TObject); procedure N18Click(Sender: TObject);
procedure N19Click(Sender: TObject); procedure N13Click(Sender: TObject);
procedure N14Click(Sender: TObject); procedure N21Click(Sender: TObject);
procedure N20Click(Sender: TObject); procedure N23Click(Sender: TObject);
procedure N24Click(Sender: TObject); procedure N2Click(Sender: TObject);
procedure N3Click(Sender: TObject); procedure N4Click(Sender: TObject);
procedure N5Click(Sender: TObject); procedure N6Click(Sender: TObject);

```

```

private { Private declarations } public { Public declarations } end;
  type PMyListE = ^EElList; //список окружностей  EElList = record
x_c,y_c,R:real; elnumber:string{char}; end; type
PMyListM = ^MdList; //список рёбер графа  MdList = record
numrebra:integer; x1,y1,x2,y2:real; el1,el2:string; dest_in:char; naklon:real;
end; type PListL = ^LList; //список L  LList = record NumEl:string;
x1,y1,x2,y2:real; numrebra:integer; sumang:real; path:string;
vn:char; end; type PListM = ^MList; //список M  MList = record
NumEl:string; x1,y1,x2,y2:real; numrebra:integer; sumang:real; path:string;
vn:char; end; type PListSm = ^SmList; //список смежных рёбер
SmList = record numreb:integer; numsmreb:array[1..50]of integer;
kolsmreber:integer; end; type MyD = record //текущее описание
NumEl:string; x1,y1,x2,y2:real; numrebra:integer; sumang:real;
path:string; vn:char; end; var Form1: TForm1;
ksi,eta,x0,y0,math_x,math_y:real; //для расчёта координат
mx,my:integer; //масштаб
MyListE: TList; //список окружностей
ElRecord1: PMyListE; //запись в списке окружностей
MyListM: TList; //список рёбер модели
MdRecord: PMyListM; //запись в списке рёбер модели
ListL: TList; //список L
ListM: TList; //список M
MRecord: PListM; //запись в списке M
ListSm: TList; //список смежных рёбер
ListReb: TList;
RebRecord: PMyListM;
scr_pos1,scr_pos2:integer; //позиции скроллов
xmax,ymax:integer; //максимальные значения координатной сетки
num_el,num_rebra:integer; //нумерация окружностей, нумерация рёбер

```

```

is_source,is_target:boolean; //для определения заданы ли источник и цель
source,target:boolean; //для определения заданы ли источник и цель
clear:boolean; //для удаления лишних рёбер после нахождения пути
editing:boolean; //редактирование
moving:boolean; //перемещение окружности
num:integer; //номер окружности в списке
CopyElRecord:PMyListE;
x_start,y_start:real; //для возвращения окружности к исходному
положению...
elnum:string; //...при пересечении другой окружности
finding_rebro:boolean; //используется при поиске мышью рёбер
implementation {$R *.dfm} procedure Koord(u,v:real);
//перевод математических координат в координаты прибора
Begin if (u<320)and(v>240)then begin ksi:=x0+u*mx; eta:=y0+v*my; end;
if (u<320)and(v<240)then begin ksi:=x0+u*mx; eta:=y0-v*my; end;
if (u>320)and(v<240)then begin ksi:=x0-u*mx; eta:=y0-v*my;
end; if (u>320)and(v>240)then begin ksi:=x0-u*mx; eta:=y0+v*my;
end;
end; procedure MathKoord(ks,et:real);
//перевод координат прибора в математические координаты begin
math_x:=(ks-x0)/mx; math_y:=(et-y0)/(-my); end;
function FindAngle(x1_1,y1_1,x1_2,y1_2,x2_1,y2_1,x2_2,y2_2:real):real;
//поиск угла между двумя прямыми (касательными)
var a,b:real; begin if (x1_1=x1_2)and(y1_1=y1_2) then FindAngle:=0
end; a:=arctan((y1_2-y1_1)/(x1_2-x1_1))*180/pi;
b:=arctan((y2_2-y2_1)/(x2_2-x2_1))*180/pi; if (a>=0)and(b>=0)and(b<a) then
FindAngle:=a-b else if (a>=0)and(b>=0)and(b>a) then
FindAngle:=b-a else if (a>=0)and(b<=0) then
FindAngle:=a+abs(b) else if (a<=0)and(b<=0)and(b<a) then

```

```

FindAngle:=abs(b)-abs(a)  else if (a<=0)and(b<=0)and(b>a) then
FindAngle:=abs(a)-abs(b)  else {if (a<0)and(b>0) then }
FindAngle:=abs(a)+b; end; end; function FindNaklon(x1,y1,x2,y2:real):real;
//пошук кута нахилу дотичної begin  if x1=x2 then  begin
  FindNaklon:=0;  exit;  end else begin FindNaklon:=arctan((y2-y1)/(x2-
x1))*180/pi;
end; end; procedure DrawAxis(sx,fx,sy,fy:integer);
//рисование координатных осей и сетки
Var  i,k:integer; begin form1.Image1.Canvas.Pen.Color:=clBlack;
  k:=0; for i:=sx to fx do begin Koord(k,0);  k:=k+1;
  form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta));
form1.Image1.Canvas.Pen.Width:=1;
form1.Image1.Canvas.LineTo(trunc(ksi),0);  if i<>0 then
form1.Image1.Canvas.TextOut(trunc(ksi),trunc(eta),inttostr(i));
form1.Image1.Canvas.Pen.Width:=4;  end; k:=0; for i:=sy to fy do
begin Koord(0,k); k:=k+1;
form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta));
form1.Image1.Canvas.Pen.Width:=1;
form1.Image1.Canvas.LineTo(640,trunc(eta));  if i<>0 then
form1.Image1.Canvas.TextOut(trunc(ksi),trunc(eta),inttostr(i));
form1.Image1.Canvas.Pen.Width:=4;  end;
form1.Image1.Canvas.Pen.Width:=1;
form1.Image1.Canvas.Pen.Color:=clBlack;
end; Procedure DrawMyEllipse(xc,yc,R:real;mytype:string);
//рисование окружности
var R_p,xc_p,yc_p:integer; //характеристики окружности begin
koord(xc,yc); xc_p:=trunc(ksi); yc_p:=trunc(eta); koord(R,0);

```

```

R_p:=trunc(ksi); form1.Image1.Canvas.Pen.Color:=clRed;
form1.Image1.Canvas.Ellipse(xc_p-R_p,yc_p+R_p,xc_p+R_p,yc_p-R_p);
//подпись окружностей
Form1.Image1.Canvas.TextOut(xc_p-2,yc_p-5,mytype);
form1.Image1.Canvas.Pen.Color:=clBlack; end;
Procedure ReDrawEllipse(shift_x,shift_y:integer);
//перерисовка окружностей, рёбер графа(модели) и оптимального
пути(если он найден)
var i:integer; xc,yc,R:real;//характеристики окружности
x1,x2,y1,y2:real; path:string; //найденный оптимальный путь begin
if MyListE.Count<>0 then //перерисовка окружностей
begin for i:=0 to MyListE.Count-1 do begin ElRecord1:=MyListE.Items[i];
xc:=ElRecord1.x_c-shift_x; yc:=ElRecord1.y_c-shift_y; R:=ElRecord1.R;
DrawMyEllipse(xc,yc,R,ElRecord1.elnumber) end; end; if clear<>true then
Begin if MyListM.Count<>0 then //перерисовка рёбер
begin form1.image1.Canvas.Pen.Color:=clBlue; for i:=0 to MyListM.Count-
1 do
begin MdRecord:=MyListM.Items[i]; x1:=MdRecord.x1-shift_x;
y1:=MdRecord.y1-shift_y;Koord(x1,y1);
form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
x2:=MdRecord.x2-shift_x; y2:=MdRecord.y2-shift_y; Koord(x2,y2);
form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta)); end;
form1.image1.Canvas.Pen.Color:=clBlack; end; end;
if ListM.Count<>0 then //рисование найденного пути begin
MRecord:=ListM.Items[ListM.Count-1]; Form1.Image1.Canvas.Pen.Width:=2;
Form1.Image1.Canvas.Pen.Color:=clGreen; path:=MRecord.path;
delete(path,length(path),1); while path<>" do begin
x1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
y1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));

```

```

x2:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
if pos('*',path)=0 then begin y2:=strtofloat(path); delete(path,1,length(path));
end else begin y2:=strtofloat(copy(path,1,pos('*',path)-1));
delete(path,1,pos('*',path)); end; Koord(x1-shift_x,y1-shift_y);
Form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
Koord(x2-shift_x,y2-shift_y);
Form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta)); end;
Form1.Image1.Canvas.Pen.Width:=1;
Form1.Image1.Canvas.Pen.Color:=clBlack;
end; end; procedure Sort;
//перенумерація
j:=1; for i:=0 to MyListE.Count-1 do begin ElRecord1:=MyListE.Items[i];
if (ElRecord1.elnumber<>'s')and(ElRecord1.elnumber<>'t')then begin
ElRecord1.elnumber:=inttostr(j); MyListE.Items[i]:=ElRecord1; j:=j+1;
end; end; end;
procedure BuildTangent(el1,el2:string;xc1,yc1,R1,xc2,yc2,R2:real);
//построение касательных
Var d:real; //расстояние между центрами окружностей
beta:real; //угол наклона прямой, соединяющей центры окружностей
sn_beta,cs_beta:real; //sin и cos угла наклона прямой, соединяющей центры
окружностей
sn_alpha,cs_alpha:real; //sin и cos угла наклона касательной
x1,y1,x2,y2:real; //координаты построенной касательной
begin if xc1=xc2 then d:=abs(yc1-yc2) else
d:=sqrt((yc2-yc1)*(yc2-yc1)+(xc2-xc1)*(xc2-xc1));
if xc2=xc1 then beta:=90*pi/180 else beta:=arctan((yc2-yc1)/(xc2-xc1));
sn_beta:=sin(beta); cs_beta:=cos(beta); //внешние касательные
sn_alpha:=(R2-R1)/d; cs_alpha:=sqrt(1-sn_alpha*sn_alpha);
//внешняя верхняя касательная

```

```

x1:=((xc1*cs_beta+yc1*sn_beta)-R1*sn_alpha)*cs_beta-((-
xc1*sn_beta+yc1*cs_beta)+R1*cs_alpha)*sn_beta;
y1:=((xc1*cs_beta+yc1*sn_beta)-R1*sn_alpha)*sn_beta+((-
xc1*sn_beta+yc1*cs_beta)+R1*cs_alpha)*cs_beta;
x2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*cs_beta-((-
xc2*sn_beta+yc2*cs_beta)+R2*cs_alpha)*sn_beta;
y2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*sn_beta+((-
xc2*sn_beta+yc2*cs_beta)+R2*cs_alpha)*cs_beta;
num_rebra:=num_rebra+1;
New(MdRecord); //добавление одного ребра графа(новой записи в список)
MdRecord.el1:=e11; MdRecord.el2:=e12;
MdRecord.numrebra:=num_rebra; MdRecord.x1:=x1;
MdRecord.y1:=y1; MdRecord.x2:=x2;
MdRecord.y2:=y2; MdRecord.naklon:=FindNaklon(x1,y1,x2,y2);
MyListM.Add(MdRecord);
//внешняя нижняя касательная
x1:=((xc1*cs_beta+yc1*sn_beta)-R1*sn_alpha)*cs_beta-((-
xc1*sn_beta+yc1*cs_beta)-R1*cs_alpha)*sn_beta;
y1:=((xc1*cs_beta+yc1*sn_beta)-R1*sn_alpha)*sn_beta+((-
xc1*sn_beta+yc1*cs_beta)-R1*cs_alpha)*cs_beta;
x2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*cs_beta-((-
xc2*sn_beta+yc2*cs_beta)-R2*cs_alpha)*sn_beta;
y2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*sn_beta+((-
xc2*sn_beta+yc2*cs_beta)-R2*cs_alpha)*cs_beta;
num_rebra:=num_rebra+1;
New(MdRecord); //добавление одного ребра графа(новой записи в список)
MdRecord.el1:=e11; MdRecord.el2:=e12;
MdRecord.numrebra:=num_rebra; MdRecord.x1:=x1;
MdRecord.y1:=y1; MdRecord.x2:=x2; MdRecord.y2:=y2;

```

```

MdRecord.naklon:=FindNaklon(x1,y1,x2,y2); MyListM.Add(MdRecord);
//-----
//внутренние касательные
sn_alpha:=(R1+R2)/d; cs_alpha:=sqrt(1-sn_alpha*sn_alpha);
//внутренняя верхняя касательная
x1:=((xc1*cs_beta+yc1*sn_beta)+R1*sn_alpha)*cs_beta-((-
xc1*sn_beta+yc1*cs_beta)+R1*cs_alpha)*sn_beta;
y1:=((xc1*cs_beta+yc1*sn_beta)+R1*sn_alpha)*sn_beta+((-
xc1*sn_beta+yc1*cs_beta)+R1*cs_alpha)*cs_beta;
x2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*cs_beta-((-
xc2*sn_beta+yc2*cs_beta)-R2*cs_alpha)*sn_beta;
y2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*sn_beta+((-
xc2*sn_beta+yc2*cs_beta)-R2*cs_alpha)*cs_beta;
num_rebra:=num_rebra+1;
New(MdRecord); //добавление одного ребра графа(новой записи в список)
MdRecord.el1:=e11; MdRecord.el2:=e12; MdRecord.numrebra:=num_rebra;
MdRecord.x1:=x1; MdRecord.y1:=y1; MdRecord.x2:=x2;
MdRecord.y2:=y2; MdRecord.naklon:=FindNaklon(x1,y1,x2,y2);
MyListM.Add(MdRecord);
//внутренняя нижняя касательная
x1:=((xc1*cs_beta+yc1*sn_beta)+R1*sn_alpha)*cs_beta-((-
xc1*sn_beta+yc1*cs_beta)-R1*cs_alpha)*sn_beta;
y1:=((xc1*cs_beta+yc1*sn_beta)+R1*sn_alpha)*sn_beta+((-
xc1*sn_beta+yc1*cs_beta)-R1*cs_alpha)*cs_beta;
x2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*cs_beta-((-
xc2*sn_beta+yc2*cs_beta)+R2*cs_alpha)*sn_beta;
y2:=((xc2*cs_beta+yc2*sn_beta)-R2*sn_alpha)*sn_beta+((-
xc2*sn_beta+yc2*cs_beta)+R2*cs_alpha)*cs_beta;
num_rebra:=num_rebra+1;

```



```

New(MdRecord); //добавление одного ребра графа(новой записи в список)
MdRecord.el1:=el1; MdRecord.el2:=el2; MdRecord.numrebra:=num_rebra;
MdRecord.x1:=x1; MdRecord.y1:=y1; MdRecord.x2:=x2; MdRecord.y2:=y2;
MdRecord.naklon:=FindNaklon(x1,y1,x2,y2); MyListM.Add(MdRecord); end;
function FindCross(x1,y1,x2,y2,xc,yc,R:real):boolean;
var D,D2:real; //дискриминант
a,b,c:real; x,y:real; begin
a:=(x1-x2)*(x1-x2)+(y1-y2)*(y1-y2);
b:=2*(y1-y2)*(x1*y2-x2*y1)-2*xc*(x1-x2)*(x1-x2)-2*yc*(x1-x2)*(y1-y2);
c:=xc*xc*(x1-x2)*(x1-x2)+(x1*y2-x2*y1)*(x1*y2-x2*y1)-2*yc*(x1-
x2)*(x1*y2-x2*y1)+yc*yc*(x1-x2)*(x1-x2)-R*R*(x1-x2)*(x1-x2);
if (x1=x2)and(y1=y2)then begin FindCross:=false; exit; end; D:=b*b-4*a*c;
if D>=0 then begin x:=(-b+sqrt(D))/(2*a);
b:=2*(x1-x2)*(x2*y1-x1*y2)-2*yc*(y1-y2)*(y1-y2)-2*xc*(x1-x2)*(y1-y2);
c:=(x2*y1-x1*y2)*(x2*y1-x1*y2)-2*xc*(x2*y1-x1*y2)*(y1-
y2)+(xc*xc+yc*yc-R*R)*(y1-y2)*(y1-y2);
D2:=b*b-4*a*c; if D2>0 then y:=(-b+sqrt(D2))/(2*a) else y:=0;
End else begin x:=0; y:=0; end; if (x1=x2)and(x=x1) then
Begin if (y1<y2)and(y>y1)and(y<y2) then begin FindCross:=true;
exit; end else if (y1>y2)and(y<y1)and(y>y2) then begin FindCross:=true;
exit; end; end;
if (D<0)or((D>-0.0000001)and(D<0.0000001)) then FindCross:=false
else //D>0 begin if (x1<x2)and(x>x1)and(x<x2) then FindCross:=true
else if (x1>x2)and(x>x2)and(x<x1) then FindCross:=true
else FindCross:=false; end; end; function PointtoComma(mystr:string):string;
//замена точек на запятыы
var i:integer; begin for i:=1 to length(mystr) do begin
if copy(mystr,i,1)='.' Then begin delete(mystr,i,1); insert(',',mystr,i);
end; end; PointtoComma:=mystr; end;

```

```

procedure TForm1.ScrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;
  var ScrollPos: Integer);
//прокрутка по осі X
Begin scr_pos1:=ScrollPos;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); end;
procedure TForm1.ScrollBar2Scroll(Sender: TObject; ScrollCode: TScrollCode;
  var ScrollPos: Integer); //прокрутка по осі Y
begin scr_pos2:=ScrollBar2.Max-Scrollpos;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,Scr_Pos2,Scr_Pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); end;
procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
//Вывод в заголовок формы координат курсора (математических)
//и перемещение окружностей в режиме редактирования
var i:integer; begin if (mx<>0)and(my<>0) then begin
MathKoord(X,Y); math_x:=math_x+ScrollBar1.Position;
math_y:=math_y+ScrollBar2.Max-ScrollBar2.Position;
Form1.Caption:='('+FloatToStrf(math_x,ffFixed,3,1)+';'+FloatToStrf(math_y,ff
Fixed,3,1)+''); end;
//находится ли курсор внутри окружности (только в режиме
редактирования) if
(editing=true)and(MyListE.Count<>0)and(moving=false)and(finding_rebro<>tr
ue) then begin for i:=0 to MyListE.Count-1 do begin
ElRecord1:=MyListE.Items[i];
if (math_x>ElRecord1.x_c-
ElRecord1.R)and(math_x<ElRecord1.x_c+ElRecord1.R)and

```

```

(math_y>ElRecord1.y_c-
ElRecord1.R)and(math_y<ElRecord1.y_c+ElRecord1.R)then begin
Image1.Cursor:=crHandPoint; num:=i; break; end else
Image1.Cursor:=crDefault;
end; CopyElRecord:=ElRecord1; end; if (moving=true) then begin
ElRecord1:=CopyElRecord; ElRecord1.x_c:=math_x; ElRecord1.y_c:=math_y;
CopyElRecord:=ElRecord1; MyListE.Items[num]:=ElRecord1;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); end;
if (finding_rebro=true)and(MyListE.Count<>0)and(MyListM.Count<>0)then
Begin   for i:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[i];
if (MdRecord.x1*math_y+MdRecord.x2*MdRecord.y1+math_x*MdRecord.y2-
MdRecord.x2*math_y-MdRecord.x1*MdRecord.y2-math_x*MdRecord.y1>-
0.1)and
(MdRecord.x1*math_y+MdRecord.x2*MdRecord.y1+math_x*MdRecord.y2-
MdRecord.x2*math_y-MdRecord.x1*MdRecord.y2-
math_x*MdRecord.y1<0.1)then
begin if
((MdRecord.x1<MdRecord.x2)and(math_x>=MdRecord.x1)and(math_x<=MdR
ecord.x2))
or((MdRecord.x1>MdRecord.x2)and(math_x<=MdRecord.x1)and(math_x>=M
dRecord.x2)) then begin Image1.Cursor:=crHandPoint;
num:=MdRecord.numrebra;
Form1.Caption:=inttostr(num); break; end; end else Image1.Cursor:=crDefault;
end; end; end; procedure TForm1.Button3Click(Sender: TObject);
//добавление окружности с указанными данными
Var xc,yc,R:real; //характеристики эллипса
d:real; //расстояние между центрами окружностей

```

```

i:integer;
cross:boolean; //для перевірки на пересечение окружностей
begin try xc:=strtofloat(edit1.Text); except ShowMessage('Помилка в значенні
Xc');
exit; end; try yc:=strtofloat(edit2.Text); except
ShowMessage('Помилка в значенні Yc'); exit; end; try
R:=strtofloat(edit4.Text); except ShowMessage('Помилка в значенні R');
exit; end; if (xc>xmax+21)or(yc>ymax+16)or(xc-R<0)or(yc+R<0) then
begin ShowMessage('Коло виходить за межі поля'); exit; end;
//перевірка на пересечение окружностей
cross:=false; if MyListE.Count>0 then begin for i:=0 to MyListE.Count-1 do
begin ElRecord1:=MyListE.Items[i];
if ElRecord1.x_c=xc then d:=abs(ElRecord1.y_c-yc)
else d:=sqrt((yc-ElRecord1.y_c)*(yc-ElRecord1.y_c)+(xc-ElRecord1.x_c)*(xc-
ElRecord1.x_c)); if d<ElRecord1.R+R then begin cross:=true; break;
end; end; end; //окружности не пересекаются
if cross<>true then begin
New(ElRecord1); //добавление новой окружности (новой записи в список)
ElRecord1.x_c:=xc; ElRecord1.y_c:=yc; ElRecord1.R:=R;
if is_source=true then begin ElRecord1.elnumber:='s'; source:=true;
N16.Enabled:=false; end else if is_target=true then begin
ElRecord1.elnumber:='t'; target:=true; N17.Enabled:=false; end
Else begin num_el:=num_el+1; ElRecord1.elnumber:=inttostr(num_el); end;
MyListE.Add(ElRecord1);DrawMyEllipse(xc-
scr_pos1,yc,R,ElRecord1.elnumber);
panel1.Visible:=false; Form1.Menu:=MainMenu1;
is_source:=false;is_target:=false;
N22.Enabled:=true; if (source=true)and(target=true) then
N30.Enabled:=true;end

```

```

else ShowMessage('Коло перетинає інше коло'); end;
procedure TForm1.FormShow(Sender: TObject);
//задание начальных значений для коорд. сетки
Begin finding_rebro:=false; num:=0; xmax:=100;ymax:=100;
  Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height); num_rebra:=0;
num_el:=0; mx:=30; my:=30;  x0:=0; y0:=480; DrawAxis(0,100,0,100);
MyListE := TList.Create; MyListM := TList.Create; ListSm := TList.Create;
ListL := TList.Create; ListM := TList.Create; scr_pos1:=0;scr_pos2:=0;
is_source:=false;is_target:=false; source:=false;target:=false; editing:=false;
end; procedure TForm1.Button8Click(Sender: TObject);
//отмена ввода окружности
Begin panel1.Visible:=false; Form1.Menu:=MainMenu1; end;
procedure TForm1.Button9Click(Sender: TObject);
//задание максимальных значений координатной сетки
var i:integer; tmp_x,tmp_y:integer; begin try tmp_x:=strtoint(edit7.Text);
Except ShowMessage('Помилка в значенні Xmax'); exit; end; try
tmp_y:=strtoint(edit8.Text); except ShowMessage('Помилка в значенні
Ymax');
exit; end; if tmp_x<21 then begin ShowMessage('Xmax не должно быть
менее 21'); exit; end; if tmp_y<16 then begin
  ShowMessage('Ymax не должно быть менее 16');
exit; end; if target=true then begin for i:=0 to MyListE.Count-1 do
begin ElRecord1:=MyListE.Items[i]; if ElRecord1.elnumber='t' then
begin koord(ElRecord1.y_c+ElRecord1.R,ElRecord1.y_c+ElRecord1.R);
if (ksi>tmp_x)or(eta>tmp_y) then begin
  ShowMessage('Последняя окружность выходит за границы поля');
exit; end; end; end; end; xmax:=tmp_x-21; ymax:=tmp_y-16;
  ScrollBar1.Max:=xmax; ScrollBar2.Max:=ymax;
  ScrollBar2.Position:=ScrollBar2.Max; scr_pos1:=0;scr_pos2:=0;

```

```
panel2.Visible:=false; Form1.Menu:=MainMenu1; end;
procedure TForm1.Button10Click(Sender: TObject);
//отмена задания размеров поля
Begin panel2.Visible:=false; Form1.Menu:=MainMenu1; end;
procedure TForm1.Image1MouseDown(Sender: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer); var i:integer;
TmpRecord:PMyListM; //для поиска одинаковой записи в списке
found:boolean; //для поиска одинаковой записи в списке
begin if (editing=true)and(image1.Cursor=crHandPoint)then
begin ElRecord1:=MyListE.Items[num]; x_start:=ElRecord1.x_c;
y_start:=ElRecord1.y_c; elnum:=ElRecord1.elnumber; moving:=true;
end; //выделение ребра с занесением его в список
if (finding_rebro=true)and(Image1.Cursor=crHandPoint) then begin
New(RebRecord); RebRecord:=MyListM.Items[num-1];
//поиск одинаковой записи (если ребро уже было внесено)
found:=false; if ListReb.Count<>0 then for i:=0 to ListReb.Count-1 do
begin TmpRecord:=ListReb.Items[i];if
TmpRecord.numrebra=RebRecord.numrebra
then begin found:=true; break; end; end; if found=false then
ListReb.Add(RebRecord); Form1.Image1.Canvas.Pen.Width:=2;
Form1.Image1.Canvas.Pen.Color:=clMaroon;
Koord(RebRecord.x1-scr_pos1,RebRecord.y1-scr_pos2);
Form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
Koord(RebRecord.x2-scr_pos1,RebRecord.y2-scr_pos2);
Form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta));
Form1.Image1.Canvas.Pen.Width:=1;
Form1.Image1.Canvas.Pen.Color:=clBlack;
end; end;
```

```

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer); var i:integer; d:real; cross:boolean;
Begin cross:=false; if (editing=true)and(moving=true) then begin
moving:=false;
for i:=0 to MyListE.Count-1 do begin ElRecord1:=MyListE.Items[i];
if CopyElRecord.elnumber<>ElRecord1.elnumber then begin
if CopyElRecord.x_c=ElRecord1.x_c then d:=abs(CopyElRecord.y_c-
ElRecord1.y_c)
else d:=sqrt((ElRecord1.y_c-CopyElRecord.y_c)*(ElRecord1.y_c-
CopyElRecord.y_c)+(ElRecord1.x_c-CopyElRecord.x_c)*(ElRecord1.x_c-
CopyElRecord.x_c));
if d<CopyElRecord.R+ElRecord1.R then begin cross:=true; break; end; end;
end; if cross=true then begin for i:=0 to MyListE.Count-1 do begin
ElRecord1:=MyListE.Items[i]; if ElRecord1.elnumber=elnum then begin
ElRecord1.x_c:=x_start; ElRecord1.y_c:=y_start;
MyListE.Items[i]:=ElRecord1;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); break; end; end; end; end; end;
procedure TForm1.N29Click(Sender: TObject);
//очистка екрана (с перерисовкой коорд. сетки и очищением списков)
Var i:integer; begin clear:=false; ScrollBar1.Position:=0;
ScrollBar2.Position:=100;
xmax:=100;ymax:=100; scr_pos1:=0; scr_pos2:=0; mx:=30;my:=30;
label8.Caption:=""; label8.Visible:=false; label5.Visible:=false;
source:=false;target:=false;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(0,100,0,100); num_rebra:=0; num_el:=0; finding_rebro:=false;
if MyListE.Count<>0 then begin

```

```

for i:=0 to MyListE.Count-1 do MyListE.Delete(0); end;
if MyListM.Count<>0 then begin
for i:=0 to MyListM.Count-1 do MyListM.Delete(0); end;
if ListL.Count<>0 then begin for i:=0 to ListL.Count-1 do ListL.Delete(0);
end; if ListM.Count<>0 then begin for i:=0 to ListM.Count-1 do
ListM.Delete(0);
end; if ListSm.Count<>0 then begin for i:=0 to ListSm.Count-1 do
ListSm.Delete(0);
end; N30.Enabled:=false; N31.Enabled:=false; N32.Enabled:=false;
N20.Enabled:=false; N22.Enabled:=false; N23.Enabled:=false;
N24.Enabled:=false; N16.Enabled:=true; N17.Enabled:=true;
N15.Enabled:=true; N13.Enabled:=true; N27.Enabled:=true;
N28.Enabled:=true; N2.Enabled:=false; N3.Enabled:=false;
N4.Enabled:=false; N21.Enabled:=true; end;
procedure TForm1.N27Click(Sender: TObject); //увеличение масштаба
begin mx:=mx*2;my:=my*2;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); if mx>=240 then begin N27.Enabled:=false;
exit; end; N28.Enabled:=true; end; procedure TForm1.N28Click(Sender:
TObject);
//уменьшение масштаба
Begin mx:=trunc(mx/2);my:=trunc(my/2);
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); if mx<=15 then begin N28.Enabled:=false;
exit; end; N27.Enabled:=true; end; procedure TForm1.N31Click(Sender:
TObject);
//построение модели

```



```

var f:textfile; //файл i,j:integer; found:boolean;
//найдена ли заданая запись в списке
ElRecord2,ElRecord3: PMyListE; cross:boolean; tmp_el:string; tmp_kr:real;
Begin Sort; Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2);
//контроль ввода источника
found:=false; for i:=0 to MyListE.Count-1 do begin
ElRecord1:=MyListE.Items[i];
if ElRecord1.elnumber='s' then found:=true; end; if found=false then
begin ShowMessage('Не задан источник'); exit; end; //контроль ввода цели
found:=false; for i:=0 to MyListE.Count-1 do begin
ElRecord1:=MyListE.Items[i];
if ElRecord1.elnumber='t' then found:=true; end; if found=false then begin
ShowMessage('Не задана цель'); exit; end; //построение касательных
for i:=0 to MyListE.Count-2 do begin ElRecord1:=MyListE.Items[i];
j:=i+1; while j<>MyListE.Count do begin ElRecord2:=MyListE.Items[j];
j:=j+1; if (ElRecord1.elnumber<>ElRecord2.elnumber) then
BuildTangent(ElRecord1.elnumber,ElRecord2.elnumber,ElRecord1.x_c,ElReco
rd1.y_c,ElRecord1.R,ElRecord2.x_c,ElRecord2.y_c,ElRecord2.R); end; end;
//проверка на пересечение рёбрами окружностей
for i:=0 to MyListM.Count-2 do begin for j:=0 to MyListM.Count-1 do
begin MdRecord:=MyListM.Items[j]; if MdRecord.numrebra=i+1 then break;
end; cross:=false; for j:=0 to MyListE.Count-2 do begin
ElRecord1:=MyListE.Items[j]; if ElRecord1.elnumber=MdRecord.el1 then
break;
end; for j:=0 to MyListE.Count-1 do begin ElRecord2:=MyListE.Items[j];
if ElRecord2.elnumber=MdRecord.el2 then break; end;
for j:=1 to MyListE.Count-2 do begin ElRecord3:=MyListE.Items[j];

```

```

if (ElRecord1.elnumber<>ElRecord3.elnumber)then begin
cross:=FindCross(MdRecord.x1,MdRecord.y1,MdRecord.x2,MdRecord.y2,
ElRecord3.x_c,ElRecord3.y_c,ElRecord3.R); if cross=true then break; end; end;
//изменение направления ребра
for j:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[j];
if MdRecord.numrebra=i+1 then break; end;
if (MdRecord.x1>MdRecord.x2)and(MdRecord.el2<>'t') then
begin tmp_el:=mdrecord.el1; MdRecord.el1:=MdRecord.el2;
MdRecord.el2:=tmp_el; tmp_kr:=MdRecord.x1; MdRecord.x1:=MdRecord.x2;
MdRecord.x2:=tmp_kr; tmp_kr:=MdRecord.y1; MdRecord.y1:=MdRecord.y2;
MdRecord.y2:=tmp_kr; MyListM.Items[j]:=MdRecord; end;
if cross=true then begin for j:=0 to MyListM.Count-1 do begin
MdRecord:=MyListM.Items[j]; if MdRecord.numrebra=i+1 then break; end;
MyListM.Delete(j); end; end; for i:=0 to MyListM.Count-1 do begin
MdRecord:=MyListM.Items[i]; MdRecord.numrebra:=i+1; end;
assignfile(f,'model1.txt'); rewrite(f); for i:=0 to MyListM.Count-1 do begin
MdRecord:=MyListM.Items[i];
writeln(f,MdRecord.numrebra,' ',MdRecord.el1,' ',MdRecord.el2,'
',floattostrf(MdRecord.x1,ffFixed,5,4),' ', floattostrf(MdRecord.y1,ffFixed,5,4),'
',floattostrf(MdRecord.x2,ffFixed,5,4),' ',floattostrf(MdRecord.y2,ffFixed,5,4));
end; closefile(f); //ВЫВОД на экран касательных
for i:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[i];
koord(MdRecord.x1-scr_pos1,MdRecord.y1-scr_pos2);
image1.Canvas.Pen.Color:=clBlue;
image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
koord(MdRecord.x2-scr_pos1,MdRecord.y2-scr_pos2);
image1.Canvas.LineTo(trunc(ksi),trunc(eta));
image1.Canvas.Pen.Color:=clBlack;
//ShowMessage('OK'); end; N32.Enabled:=true; N15.Enabled:=false;

```

```

N21.Enabled:=false; N23.Enabled:=false; N31.Enabled:=false;
N2.Enabled:=true;
end; procedure TForm1.N32Click(Sender: TObject); var D: MyD; //описание D
ang:real; //угол между рёбрами i,j,k:integer; //счётчики в циклах
min:real; //минимальный суммарный угол
found:boolean; //найдена ли заданая запись в списке
mywhere:string; //имя списка, где найдена запись
path:string; //путь ind:integer; //индекс записи
x1,y1,x2,y2:real; LRecord: PListL; //запись в списке L
SmRecord: PListSm; //запись в списке смежных рёбер
MdRecordTmp: PMyListM; //временная запись при поиске смежных рёбер
f:textfile; tmp:integer; label label1; begin tmp:=1;
assignfile(f,'F:\UNIVER\Diploma\1\MyProject\debug.doc'); rewrite(f);
N31.Enabled:=false; N32.Enabled:=false;
//определение части окружности, в которую направлены рёбра
for i:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[i];
for j:=0 to MyListE.Count-1 do begin ElRecord1:=MyListE.Items[j];
if ElRecord1.elnumber=MdRecord.el2 then break; end;
if MdRecord.y2>ElRecord1.y_c then MdRecord.dest_in:='v'
else MdRecord.dest_in:='n' end; //формирование списка смежных рёбер
New(SmRecord); SmRecord.numreb:=0; SmRecord.kolsmreber:=0; j:=0;
for i:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[i];
if MdRecord.el1='s' then begin j:=j+1;
SmRecord.numsmreb[j]:=MdRecord.numrebra;
SmRecord.kolsmreber:=SmRecord.kolsmreber+1; end; end;
ListSm.Add(SmRecord);
for k:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[k];
New(SmRecord); SmRecord.kolsmreber:=0;
SmRecord.numreb:=MdRecord.numrebra; for i:=0 to MyListE.Count-1 do

```

```

Begin ElRecord1:=MyListE.Items[i];
if ElRecord1.elnumber=MdRecord.el2 then break; end;
if MdRecord.y2>ElRecord1.y_c then begin j:=1; for i:=0 to MyListM.Count-1
do
begin MdRecordTmp:=MyListM.Items[i]; if MdRecordTmp.el1=MdRecord.el2
then
(MdRecordTmp.naklon<MdRecord.naklon)and(MdRecordTmp.y1>ElRecord1.
y_c)
Then begin SmRecord.numsmreb[j]:=MdRecordTmp.numrebra;
SmRecord.kolsmreber:=SmRecord.kolsmreber+1; j:=j+1; end; end;
ListSm.Add(SmRecord); end; if MdRecord.y2<=ElRecord1.y_c then begin
j:=1; for i:=0 to MyListM.Count-1 do begin MdRecordTmp:=MyListM.Items[i];
if MdRecordTmp.el1=MdRecord.el2 then if
(MdRecordTmp.naklon>MdRecord.naklon)and(MdRecordTmp.y1<ElRecord1.
y_c) then begin SmRecord.numsmreb[j]:=MdRecordTmp.numrebra;
SmRecord.kolsmreber:=SmRecord.kolsmreber+1; j:=j+1; end; end;
ListSm.Add(SmRecord); end; end; //формируем первое описание D
MdRecord:=MyListM.Items[0]; D.NumEl:='s'; D.x1:=0; D.y1:=0; D.x2:=0;
D.y2:=0; D.numrebra:=0; D.sumang:=0; D.vn:='s'; D.path:='';
//помещаем первое описание D в L New(LRecord);
LRecord.NumEl:=D.NumEl; LRecord.x1:=D.x1; LRecord.y1:=D.y1;
LRecord.x2:=D.x2; LRecord.y2:=D.y2; LRecord.numrebra:=D.numrebra;
LRecord.sumang:=D.sumang; LRecord.vn:=D.vn; LRecord.path:='';
ListL.Add(LRecord); N20.Enabled:=true; label1: if ListL.Count=0 then
Begin ShowMessage('Пути не существует'); exit; end;
//выбираем из L запись с min суммарным углом ind:=0;
LRecord:=ListL.Items[0];
min:=LRecord.sumang; for i:=1 to ListL.Count-1 do begin
LRecord:=ListL.Items[i]; if min>LRecord.sumang then begin

```

```
min:=LRecord.sumang; ind:=i; end; end; LRecord:=ListL.Items[ind];
//переносим найденную запись в список M New(MRecord);
MRecord.NumEl:=LRecord.NumEl; MRecord.numrebra:=LRecord.numrebra;
MRecord.x1:=LRecord.x1; MRecord.y1:=LRecord.y1;
MRecord.x2:=LRecord.x2;
MRecord.y2:=LRecord.y2; MRecord.sumang:=LRecord.sumang;
MRecord.path:=LRecord.path; MRecord.vn:=LRecord.vn;
ListM.Add(MRecord); ListL.Delete(ind);
if MRecord.NumEl='t' then //последняя окружность begin
MRecord:=ListM.Items[ListM.Count-1];
label5.Visible:=true;label8.Visible:=true;
label8.Caption:=floattostrf(MRecord.sumang,ffFixed,3,5);
//рисование найденного пути Form1.Image1.Canvas.Pen.Width:=2;
Form1.Image1.Canvas.Pen.Color:=clGreen; path:=MRecord.path;
delete(path,length(path),1); while path<>" do begin
x1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
y1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
x2:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
if pos('*',path)=0 then begin y2:=strtofloat(path); delete(path,1,length(path));
end else begin y2:=strtofloat(copy(path,1,pos('*',path)-1));
delete(path,1,pos('*',path)); end; Koord(x1-scr_pos1,y1-scr_pos2);
Form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));
Koord(x2-scr_pos1,y2-scr_pos2);
Form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta)); end;
Form1.Image1.Canvas.Pen.Width:=1;
Form1.Image1.Canvas.Pen.Color:=clBlack; exit; end;
//поиск записи в SmList, которая соответствует найденному ребру
for i:=0 to ListSm.Count-1 do begin SmRecord:=ListSm.Items[i];
if SmRecord.numreb=MRecord.numrebra then break;
```

```

end; i:=1; //перебор всех смежных рёбер while i<=SmRecord.kolsmreber do
begin //поиск в MyListM ребра с номером смежного ребра
for j:=0 to MyListM.Count-1 do begin MdRecord:=MyListM.Items[j];
if MdRecord.numrebra=SmRecord.numsmreb[i] then break; end;
//угол между ребром и смежным ребром
ang:=FindAngle(MRecord.x1,MRecord.y1,MRecord.x2,MRecord.y2,MdRecord
.x1,
MdRecord.y1, MdRecord.x2,MdRecord.y2);
writeln(f,'x1_1=' + floattostrf(mRecord.x1,ffFixed,5,2) + ';'+'y1_1=' +
floattostrf(mRecord.y1,ffFixed,5,2) + ';'+'x1_2=' +
floattostrf(mRecord.x2,ffFixed,5,2) + ';'+'y1_2=' +
floattostrf(mRecord.y2,ffFixed,5,2) + ';'+'x2_1=' +
floattostrf(mdRecord.x1,ffFixed,5,2) + ';'+'y2_1=' +
floattostrf(mdRecord.y1,ffFixed,5,2) + ';'+'x2_2=' +
floattostrf(mdRecord.x2,ffFixed,5,2) + ';'+'y2_2=' +
floattostrf(mdRecord.y2,ffFixed,5,2));
writeln(f, floattostrf(ang,ffFixed,5,2)); tmp:=tmp+1;
if tmp=23 then ShowMessage('24'); //closefile(f);
//формирование описания D' D.NumEl:=MdRecord.el2;
D.x1:=MdRecord.x1; D.y1:=MdRecord.y1; D.x2:=MdRecord.x2;
D.y2:=MdRecord.y2; D.numrebra:=MdRecord.numrebra;
D.sumang:=MRecord.sumang+ang; D.vn:=MdRecord.dest_in;
D.path:=MRecord.path+floattostr(MdRecord.x1)+'*'+floattostr(MdRecord.y1)+'
*'
+floattostr(MdRecord.x2)+'*'+floattostr(MdRecord.y2)+'*';
found:=false;mywhere:="; //поиск в ListL записи, эквивалентной D
for j:=0 to ListL.Count-1 do begin LRecord:=ListL.Items[j]; if
(LRecord.NumEl=D.NumEl)and(LRecord.vn=D.vn)and(LRecord.sumang>D.su
mang

```

```

) then begin found:=true;mywhere:=mywhere+'L';ind:=j;break; end end;
//поиск в ListM записи, эквивалентной D for j:=0 to ListM.Count-1 do
Begin MRecord:=ListM.Items[j]; if
(MRecord.NumEl=D.NumEl)and(MRecord.vn=D.vn)and(MRecord.sumang>D.
sumang) then begin found:=true;mywhere:=mywhere+'M';ind:=j;break; end end;
//если ребро нигде не найдено if found=false then begin //помещаем D в L
New(LRecord); LRecord.NumEl:=D.NumEl; LRecord.x1:=D.x1;
LRecord.y1:=D.y1; LRecord.x2:=D.x2; LRecord.y2:=D.y2;
LRecord.numrebra:=D.numrebra; LRecord.sumang:=D.sumang;
LRecord.vn:=D.vn; LRecord.path:=D.path; ListL.Add(LRecord);
i:=i+1; continue; end //если ребро найдено else begin
if mywhere='LM' then begin if MRecord.sumang>D.sumang then
ListM.Delete(ind);
if LRecord.sumang>=D.sumang then begin //помещаем описание D в L
if LRecord.sumang>D.sumang then ListL.Delete(ind); New(LRecord);
LRecord.NumEl:=D.NumEl; LRecord.x1:=D.x1; LRecord.y1:=D.y1;
LRecord.x2:=D.x2; LRecord.y2:=D.y2; LRecord.numrebra:=D.numrebra;
LRecord.sumang:=D.sumang; LRecord.vn:=D.vn; LRecord.path:=D.path;
ListL.Add(LRecord); i:=i+1; continue; end else begin i:=i+1; continue;end;
end; if mywhere='L' then begin if LRecord.sumang>=D.sumang then
begin //помещаем описание D в L
if LRecord.sumang>D.sumang then ListL.Delete(ind); New(LRecord);
LRecord.NumEl:=D.NumEl; LRecord.x1:=D.x1; LRecord.y1:=D.y1;
LRecord.x2:=D.x2; LRecord.y2:=D.y2; LRecord.numrebra:=D.numrebra;
LRecord.sumang:=D.sumang; LRecord.vn:=D.vn; LRecord.path:=D.path;
ListL.Add(LRecord); i:=i+1; continue; end else begin i:=i+1; continue;end;
end; if mywhere='M' then begin if MRecord.sumang>=D.sumang then
begin //помещаем описание D в M
if MRecord.sumang>D.sumang then ListM.Delete(ind); New(LRecord);

```

```

IRecord.NumEl:=D.NumEl; IRecord.x1:=D.x1;IRecord.y1:=D.y1;
IRecord.x2:=D.x2; IRecord.y2:=D.y2; IRecord.numrebra:=D.numrebra;
IRecord.sumang:=D.sumang; IRecord.vn:=D.vn; IRecord.path:=D.path;
List1.Add(MRecord); i:=i+1; continue; end else begin i:=i+1;continue;end;
end; end; end; goto label1; closefile(f); end;
procedure TForm1.N16Click(Sender: TObject); //задание источника
begin panel1.Visible:=true; Form1.Menu:=MainMenu2; is_source:=true;
edit1.SetFocus; if target=true then N31.Enabled:=true; end;
procedure TForm1.N17Click(Sender: TObject); //задание цели
begin panel1.Visible:=true; Form1.Menu:=MainMenu2; is_target:=true;
edit1.SetFocus; if source=true then N31.Enabled:=true; end;
procedure TForm1.N18Click(Sender: TObject);
//ВЫЗОВ ОКНА ДЛЯ ДОБАВЛЕНИЯ НОВОЙ ОКРУЖНОСТИ
Begin panel1.Visible:=true; Form1.Menu:=MainMenu2; edit1.SetFocus;
end; procedure TForm1.N19Click(Sender: TObject); begin
if MyListE.Count<>0 then begin ElRecord1:=MyListE.Items[MyListE.Count-
1];
if ElRecord1.elnumber='s' then begin source:=false; N16.Enabled:=true;
N31.Enabled:=false; N32.Enabled:=false; end else if ElRecord1.elnumber='t'
then
Begin target:=false; N17.Enabled:=true; N31.Enabled:=false;
N32.Enabled:=false;
end; MyListE.Delete(MyListE.Count-1); num_el:=num_el-1;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); end; end;
procedure TForm1.N13Click(Sender: TObject);
//считывание окружностей из файла var f:textfile; //файл с окружностями
str:string; xc,yc,R:real; num:string; //номер окружности

```



```

i:integer;//вспомогательные переменные для разбивки строки
ch:string; //вспомогательная переменная для разбивки строки
begin if opendirialog1.Execute and fileexists(opendirialog1.FileName) then
begin assignfile(f,opendirialog1.FileName); try reset(f); except
showmessage('Невозможно открыть файл'+ opendirialog1.FileName);
exit; end; while not eof(f) do begin readln(f,str); str:=PointtoComma(str);
//разбивка строки num:=""; xc:=0; yc:=0; R:=0; i:=1; repeat ch:=copy(str,i,1);
if (num<>"")and(xc<>0)and(yc<>0)and(R=0) then begin R:=strtofloat(str);
break; end else if ch=' ' then begin if num="" then num:=copy(str,1,i-1) else
if xc=0 then xc:=strtofloat(copy(str,1,i-1)) else
if yc=0 then yc:=strtofloat(copy(str,1,i-1)) else R:=strtofloat(copy(str,1,i-1));
delete(str,1,i); i:=1; end else i:=i+1; until str=""; if num='s' then begin
if source=true then begin ShowMessage('Источник уже задан');
exit; end; DrawMyEllipse(xc-scr_pos1,yc,R,'s'); N16.Enabled:=false;
source:=true; end else if num='t' then begin if target=true then begin
ShowMessage('Цель уже задана'); N31.Enabled:=true; exit; end;
DrawMyEllipse(xc-scr_pos1,yc,R,'t'); N17.Enabled:=false; target:=true; end
else DrawMyEllipse(xc-scr_pos1,yc,R,num);
New(EIRecord1); //добавление новой окружности (новой записи) в список
if (num='s')or(num='t') then EIRecord1.elnumber:=num
else begin num_el:=num_el+1; EIRecord1.elnumber:=inttostr(num_el);
end; EIRecord1.x_c:=xc; EIRecord1.y_c:=yc; EIRecord1.R:=R;
MyListE.Add(EIRecord1); end; closefile(f); end; if MyListE.Count<>0 then
Begin N30.Enabled:=true; N31.Enabled:=true; N13.Enabled:=false;
N22.Enabled:=true; N23.Enabled:=true; end; end;
procedure TForm1.N14Click(Sender: TObject);
//сохранение в файл var f:textfile; i,j:integer; str:string; begin
savedialog1.Execute then begin assignfile(f,savedialog1.FileName);
rewrite(f); if MyListE.Count<>0 then begin for i:=0 to MyListE.Count-1 do if

```

```

begin ElRecord1:=MyListE.Items[i];
str:=ElRecord1.elnumber+' '+floattostr(ElRecord1.x_c)+'
'+floattostr(ElRecord1.y_c)+' '+floattostr(ElRecord1.R);
//замена запятых на точки for j:=1 to length(str) do begin
if copy(str,j,1)=',' then begin delete(str,j,1); insert('.',str,j); end; end;
writeln(f,str); end; end; closefile(f); end; end;
procedure TForm1.N21Click(Sender: TObject);
//вызов окна для задание размеров поля begin Panel2.Visible:=true;
edit7.SetFocus; Form1.Menu:=MainMenu2; end;
procedure TForm1.N20Click(Sender: TObject); var i:integer;
xc,yc,R:real;//характеристики окружности x1,x2,y1,y2:real;
path:string; //найденный оптимальный путь begin N2.Enabled:=false;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
if MyListE.Count<>0 then //перерисовка окружностей
begin for i:=0 to MyListE.Count-1 do begin ElRecord1:=MyListE.Items[i];
xc:=ElRecord1.x_c-scr_pos1; yc:=ElRecord1.y_c-scr_pos2;
R:=ElRecord1.R; DrawMyEllipse(xc,yc,R,ElRecord1.elnumber) end; end;
if ListM.Count<>0 then //рисование найденного пути begin
MRecord:=ListM.Items[ListM.Count-1]; Form1.Image1.Canvas.Pen.Width:=2;
Form1.Image1.Canvas.Pen.Color:=clGreen; path:=MRecord.path;
delete(path,length(path),1); while path<>" do begin
x1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
y1:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
x2:=strtofloat(copy(path,1,pos('*',path)-1)); delete(path,1,pos('*',path));
if pos('*',path)=0 then begin y2:=strtofloat(path); delete(path,1,length(path));
end else begin y2:=strtofloat(copy(path,1,pos('*',path)-1));
delete(path,1,pos('*',path)); end; Koord(x1-scr_pos1,y1-scr_pos2);
Form1.Image1.Canvas.MoveTo(trunc(ksi),trunc(eta));

```

```

Koord(x2-scr_pos1,y2-scr_pos2);
Form1.Image1.Canvas.LineTo(trunc(ksi),trunc(eta)); end;
Form1.Image1.Canvas.Pen.Width:=1;
Form1.Image1.Canvas.Pen.Color:=clBlack;
end; clear:=true; end; procedure TForm1.N23Click(Sender: TObject);
//начать редактирование begin editing:=true; label3.Visible:=true;
N31.Visible:=false; N32.Visible:=false; N23.Enabled:=false;
N24.Enabled:=true; N12.Enabled:=false; N15.Enabled:=false;
N30.Enabled:=false; N20.Enabled:=false; N21.Enabled:=false; end;
procedure TForm1.N24Click(Sender: TObject); //закончить редактирование
begin editing:=false; label3.Visible:=false; N31.Visible:=true;
N32.Visible:=true;
N23.Enabled:=true; N24.Enabled:=false; N12.Enabled:=true;
N15.Enabled:=true;
N30.Enabled:=true; N20.Enabled:=true; N21.Enabled:=true; Sort;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); end;
procedure TForm1.N2Click(Sender: TObject); //определение номера ребра
begin ListReb:=TList.Create; if MyListM.Count>0 then finding_rebro:=true;
N3.Enabled:=true; end; procedure TForm1.N3Click(Sender: TObject);
//поиск пути по выделенным рёбрам var i,j:integer; sum_angle:real;
TmpRecord: PMyListM;
SmRecord: PListSm; //запись в списке смежных рёбер
found:boolean; ind:integer; begin
if (finding_rebro=true)and (ListReb.Count<>0)then begin sum_angle:=0;
//если последнее ребро не входит в цель
RebRecord:=ListReb.Items[ListReb.Count-1]; if RebRecord.e12<>'t' then
Begin ShowMessage('Последнее ребро не входит в цель'); exit; end;

```

```
ind:=0; for i:=0 to ListReb.Count-2 do begin found:=false;
RebRecord:=ListReb.Items[i]; TmpRecord:=ListReb.Items[i+1];
for j:=0 to ListSm.Count-1 do begin SmRecord:=ListSm.Items[j];
if SmRecord.numreb=RebRecord.numrebra then begin found:=true;
ind:=j; break; end; end; if found=true then begin
SmRecord:=ListSm.Items[ind]; for j:=0 to SmRecord.kolsmreber do begin
if TmpRecord.numrebra=SmRecord.numsmreb[j] then break; end; end; end;
for i:=0 to ListReb.Count-2 do begin RebRecord:=ListReb.Items[i];
TmpRecord:=ListReb.Items[i+1];
sum_angle:=sum_angle+FindAngle(RebRecord.x1,RebRecord.y1,RebRecord.x
2,Reb
Record.y2, TmpRecord.x1,TmpRecord.y1,TmpRecord.x2,TmpRecord.y2);
end; Label5.Visible:=true; Label8.Visible:=true;
Label8.Caption:=floattostr(sum_angle); n4.Enabled:=true; end;
if ListReb.Count<>0 then begin for i:=0 to ListReb.Count-1 do
ListReb.Delete(0);
end; end; procedure TForm1.N4Click(Sender: TObject);
//отмена поиска по рёбрам var i:integer; begin
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
DrawAxis(scr_pos1,scr_pos1+45,scr_pos2,scr_pos2+45);
ReDrawEllipse(scr_pos1,scr_pos2); Label5.Visible:=false;
Label8.Visible:=false;
if ListReb.Count<>0 then begin for i:=0 to ListReb.Count-1 do
ListReb.Delete(0);
end; end; procedure TForm1.N5Click(Sender: TObject);
//вывод формы на печать begin
if MessageDlg('Вывести на печать',mtConfirmation, [mbYes, mbNo], 0) =
mrYes then begin Form1.Color:=clWhite; Form1.Print;
Form1.Color:=clBtnFace;
```

end; end; END.

Програма PUG для реалізації модифікованого методу Ньютона для визначення кутів обхвату нитками основи циліндричних направляючих роторного натягувача нитки активної дії.

```
unit UG1; interface uses  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,  Dialogs, StdCtrls, jpeg, ExtCtrls; type  TfrmUG1 =
class(TForm)  lbl1UG1: TLabel;  lbl2UG1: TLabel;  lbl3UG1: TLabel;
btn1UG1: TButton;  img1UG1: TImage;  lbl4UG1: TLabel;  lbl5UG1:
TLabel;  procedure btn1UG1Click(Sender: TObject); private  { Private
declarations } public  { Public declarations } end; var  frmUG1: TfrmUG1;
implementation uses UG2; {$R *.dfm} procedure
TfrmUG1.btn1UG1Click(Sender: TObject); begin frmUG1.Hide;
frmUG2.Show; end; end. unit UG2; interface uses  Windows, Messages,
SysUtils, Variants, Classes, Graphics, Controls, Forms,  Dialogs, Math, Menus,
TeEngine, Series, ExtCtrls, TeeProcs, Chart, StdCtrls,  Printers; type
TfrmUG2 = class(TForm)  MainMenu1: TMainMenu;  N1: TMenuItem;
N2: TMenuItem;  N3: TMenuItem;  N4: TMenuItem;  N5: TMenuItem;
N6: TMenuItem;  N7: TMenuItem;  N8: TMenuItem;  N9: TMenuItem;
N10: TMenuItem;  lbl1UG2: TLabel;  Memo1: TMemo;  mem1UG2:
TMemo;  lbl2UG2: TLabel;  chr1UG2: TChart;  lbl3UG2: TLabel;
lbl4UG2: TLabel;  Series1: TLineSeries;  chr2UG2: TChart;  Series2:
TLineSeries;  Series3: TLineSeries;  Series4: TLineSeries;  Series5:
TLineSeries;  N12: TMenuItem;  N13: TMenuItem;  N11: TMenuItem;
procedure N5Click(Sender: TObject);  procedure N6Click(Sender: TObject);
procedure N7Click(Sender: TObject);  procedure N8Click(Sender: TObject);
procedure N10Click(Sender: TObject);  procedure N12Click(Sender:
TObject);  procedure N13Click(Sender: TObject);  procedure
N11Click(Sender: TObject);  private  { Private declarations } public  {
Public declarations } end; type xf=array[1..50]of Real; var  frmUG2:
```

```

TfrmUG2; s1,RR,i,j,k,ug:integer; h,x1,g,Fi:Real; A:array[1..50,1..50] of
Real; B:array[1..50]of Real; F,X:xf; x0,xx:array[1..50] of string;
ii,ss,xxx,ugi,LL1:string; ugol1,ugol2:array[1..50]of Real;
h1,d1,h2,d2,L1:array[1..50]of Real; implementation uses
UG1,Synt,UErrors,UG3,UG4,UG5; {$R *.dfm} procedure v(var F:xf;
X:xf;r4,r2,r1,R,L,Fi:Real); var i:Integer; begin SetData('r4',r4);
SetData('r2',r2); SetData('r1',r1); SetData('R',R); SetData('L',L);
SetData('Fi',Fi); SetData('X[1]',X[1]); SetData('X[2]',X[2]);
SetData('X[3]',X[3]); SetData('X[4]',X[4]); SetData('X[5]',X[5]);
SetData('X[6]',X[6]); SetData('X[7]',X[7]); SetData('X[8]',X[8]);
SetData('X[9]',X[9]); SetData('X[10]',X[10]); SetData('X[11]',X[11]);
SetData('X[12]',X[12]); SetData('X[13]',X[13]);
SetData('X[14]',X[14]); SetData('X[15]',X[15]);
SetData('X[16]',X[16]); SetData('X[17]',X[17]);
SetData('X[18]',X[18]); SetData('X[19]',X[19]);
SetData('X[20]',X[20]); SetData('X[21]',X[21]);
SetData('X[22]',X[22]); SetData('X[23]',X[23]);
SetData('X[24]',X[24]); for i:=1 to N do begin Calculate(F[i]);
end; GetData('F[1]',F[1]); GetData('F[2]',F[2]);
GetData('F[3]',F[3]); GetData('F[4]',F[4]); GetData('F[5]',F[5]);
GetData('F[6]',F[6]); GetData('F[7]',F[7]); GetData('F[8]',F[8]);
GetData('F[9]',F[9]); GetData('F[10]',F[10]); GetData('F[11]',F[11]);
GetData('F[12]',F[12]); GetData('F[13]',F[13]); GetData('F[14]',F[14]);
GetData('F[15]',F[15]); GetData('F[16]',F[16]); GetData('F[17]',F[17]);
GetData('F[18]',F[18]); GetData('F[19]',F[19]); GetData('F[20]',F[20]);
GetData('F[21]',F[21]); GetData('F[22]',F[22]); GetData('F[23]',F[23]);
GetData('F[24]',F[24]); end; procedure TfrmUG2.N5Click(Sender: TObject);
begin frmUG1.Close; end; procedure TfrmUG2.N6Click(Sender: TObject);
begin frmUG5.Show; frmUG3.show; end; procedure

```

```

TfrmUG2.N7Click(Sender: TObject); begin frmUG4.Show; end; procedure
TfrmUG2.N8Click(Sender: TObject); begin  s1:=0;    for i:=1 to N do
begin  ii:= format('%2.0d',[i]);    mem1UG2.Lines.Add('Начальное
значение'+ii+'-го корня');    x0[i]:=InputBox('Введение начальных
значений корня X0(i)', 'Значение корня X0('+ii+')=', '');
Val(x0[i],x[i],code);    xx[i]:=format('%12.5f',[x[i]]);
mem1UG2.Lines.Add('x('+ii+')='+xx[i]);    end;    for ug:=1 to 19 do
begin  Fi:=10*(ug-1)*Pi/180;    // Удаляется форма сообщений об
ошибках  if (FErrors <> nil) then FErrors.Close;    // Создается польская
запись  if not CreatePZ(Мемо1.Text)  then  begin  // При ошибках
трансляции показывается форма с сообщениями
Application.CreateForm(TFEErrors, FErrors);
FEErrors.LBErrors.Items.Assign(ErrorList);  FErrors.Show;  exit;  end;
repeat  v(F,X,r4,r2,r1,R,L,Fi);    for i:=1 to N do
begin  B[i]:=-F[i];    end;    for j:=1 to N do
begin  x1:=x[j]; h:=e*abs(x1);    x[j]:=x1+h;
v(F,X,r4,r2,r1,R,L,Fi);    for i:=1 to N do    begin
A[i,j]:=(F[i]+B[i])/h;    end;    x[j]:=x1;    end;
s1:=s1+1;    if s1=M+1 then    begin
ss:=format('%3.0d',[s1]);    mem1UG2.Lines.Add('Число
итераций s='+ss);    Break;    end;
for i:=1 to N-1 do    begin    for j:=i+1 to N do
begin  A[j,i]:=-A[j,i]/A[i,i];    for k:=i+1 to N
do  begin  A[j,k]:=A[j,k]+A[j,i]*A[i,k];
end;    B[j]:=B[j]+A[j,i]*B[i];    end;
end;    F[N]:=B[N]/A[N,N];    for i:=N-1 downto 1 do
begin  h:=B[i];    for j:=i+1 to N do
begin  h:=h-F[j]*A[i,j];    end;
F[i]:=h/A[i,i];    end;    RR:=0;    for i:=1

```

```

to N do          begin          x[i]:=x[i]+F[i];
if abs(F[i]/x[i])>e then RR:=1;          end;          if RR=1
then Continue;          mem1UG2.Lines.Add('Корни решения системы
уравнений');          ugi:=format('%4.0d',[(ug-1)*10]);
mem1UG2.Lines.Add('Угол поворота водила='+ugi);          for i:=1 to
N do          begin          ii:= format('%2.0d',[i]);
xxx:=format('%11.5f',[x[i]]);
mem1UG2.Lines.Add('x('+ii+')='+xxx);          end;
ugol1[ug]:=x[1];          ugol2[ug]:=x[2];
h1[ug]:=R*sin(Fi)+r1*cos(ugol1[ug])-r4-r2*(1-cos(ugol1[ug]));
d1[ug]:=L-R*cos(Fi)-sin(ugol1[ug])*(r1+r2);
h2[ug]:=R*sin(Fi)+r1*cos(ugol2[ug])-r4-r2*(1-cos(ugol2[ug]));
d2[ug]:=L+R*cos(Fi)-sin(ugol2[ug])*(r1+r2);
l1[ug]:=(r1+r2)*(ugol1[ug]+ugol2[ug])+sqrt(h1[ug]*h1[ug]+d1[ug]*d1[ug])+sqrt
(h2[ug]*h2[ug]+d2[ug]*d2[ug])-2*L;
LL1:=format('%11.5f',[L1[ug]]);          mem1UG2.Lines.Add('Подача L
равна='+LL1);          ss:=format('%3.0d',[s1]);
mem1UG2.Lines.Add('Число итераций s='+ss);          Break;
until false; end; procedure TfrmUG2.N10Click(Sender: TObject); begin with
Series1 do          begin          for ug:=1 to 19 do          begin          Fi:=10*(ug-
1)*Pi/180;          Series1.AddXY(Fi,L1[ug],"clRed);
Series2.AddXY(Fi,Ugol1[ug],"clBlue);
Series3.AddXY(Fi,Ugol2[ug],"clGreen);          Series4.AddXY(arcsin((r4-
r1)/R),L1[ug],"clBlack);          Series5.AddXY(arcsin((r4-
r1)/R),Ugol1[ug],"clBlack);          end;          end; end; procedure
TfrmUG2.N12Click(Sender: TObject);          Var          Prn:TextFile;          k:Integer;
begin          AssignPrn(Prn);          Rewrite(Prn);
Printer.Canvas.Font:=mem1UG2.Font;          for k:=0 to mem1UG2.Lines.Count-
1 do          WriteLn(Prn,mem1UG2.Lines[k]);          CloseFile(Prn);          end;

```



```

procedure TfrmUG2.N13Click(Sender: TObject); begin  chr1UG2.Print;
chr2UG2.Print; end; procedure TfrmUG2.N11Click(Sender: TObject); begin
mem1UG2.Clear; end; end. unit UG3; interface uses  Windows, Messages,
SysUtils, Variants, Classes, Graphics, Controls, Forms,  Dialogs, StdCtrls; type
TfrmUG3 = class(TForm)  lbl2UG3: TLabel;  lbl3UG3: TLabel;  lbl4UG3:
TLabel;  lbl5UG3: TLabel;  edt1UG3: TEdit;  edt2UG3: TEdit;
edt3UG3: TEdit;  edt4UG3: TEdit;  btn1UG3: TButton;  lbl1UG3: TLabel;
edt5UG3: TEdit;  procedure btn1UG3Click(Sender: TObject); private  {
Private declarations } public  { Public declarations } end; var frmUG3:
TfrmUG3; r4,r2,r1,R,L:Real; code:Integer; implementation  uses UG2; {$R
*.dfm} procedure TfrmUG3.btn1UG3Click(Sender: TObject); begin
Val(edt1UG3.Text,r4,code); Val(edt2UG3.Text,r2,code);
Val(edt3UG3.Text,r1,code); Val(edt4UG3.Text,R,code);
Val(edt5UG3.Text,L,code); frmUG3.Hide; end; end. unit UG4; interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls; type  TfrmUG4 = class(TForm)  lbl1UG4: TLabel;
lbl2UG4: TLabel;  lbl3UG4: TLabel;  btn1UG4: TButton;  procedure
btn1UG4Click(Sender: TObject); private  { Private declarations } public
{ Public declarations } end; var frmUG4: TfrmUG4; implementation  uses
UG2; {$R *.dfm} procedure TfrmUG4.btn1UG4Click(Sender: TObject); begin
frmUG4.Close; end; end. unit UG5; interface uses  Windows, Messages,
SysUtils, Variants, Classes, Graphics, Controls, Forms,  Dialogs, StdCtrls; type
TfrmUG5 = class(TForm)  lbl1UG5: TLabel;  lbl2UG5: TLabel;  lbl3UG5:
TLabel;  edt1UG5: TEdit;  edt2UG5: TEdit;  edt3UG5: TEdit;
btn1UG5: TButton;  procedure btn1UG5Click(Sender: TObject); private  {
Private declarations } public  { Public declarations } end; var frmUG5:
TfrmUG5; N,M,code:Integer; e:Real; implementation  uses UG2; {$R
*.dfm} procedure TfrmUG5.btn1UG5Click(Sender: TObject); begin
Val(edt1UG5.Text,N,code);  Val(edt2UG5.Text,M,code);

```

```

Val(edt3UG5.Text,e,code); frmUG5.Hide; end; end. unit Synt; interface uses
classes; type TData = record Name: string; Data:real; end; var NConst:
integer = 100; ErrorList: TStringList; PZ: array of integer; DataList: array of
TData; const MConst = 2; procedure SyntItem(S:string; First:boolean=false;
Pos:Integer=1); function CreatePZ(S:string):boolean; function Calculate(var
R:real):boolean; function SetData(Name:string; Data:real):boolean; function
GetData(Name:string; var Data:real):boolean; implementation uses Sysutils,
Math, Dialogs,UG2,UErrors; type TType = (None, Ident, Func, Part, All);
TSynt = record mode: TType; Number:real; Ident:string;
Error:boolean; Pos1,Pos2:integer; end; const SetNum: set of char=['0'..'9',
']; SetDiv: set of char=[';', '(', ')', '=', '+', '-', '/', '*', '^',
'{' , '}',
#13]; SetChar: set of char=['a'..'z','A'..'Z','_']; NFunc = 11; Functions:
array[1..NFunc] of string =
('exp','sin','cos','sqrt','abs','ln','tg','arctan','arccos','sqr','arcsin'); var SItem: TSynt;
TrStack: array of char; ConstList: array of real; Position: Integer; procedure
SyntItem(S:string;First:boolean=false;Pos:Integer=1); var i:integer; begin if (S
= "") then begin SItem.mode := All; exit; end; if(First) then Position := Pos;
repeat if (S[Position] = '{') then begin repeat Inc(Position) until
(Position >= Length(S)) or (S[Position] = '}'); Inc(Position); end;
if(Position <= Length(S)) then while ((S[Position] = ' ')or (S[Position] =
#13)or (S[Position] = #10)or (S[Position] = #0)) do Inc(Position);
until (S[Position] <> '{'); SItem.Error:=false; SItem.Pos1:=Position;
if(Position > Length(S)) then begin SItem.mode := All; exit; end;
SItem.Ident := S[Position]; if (S[Position] in SetChar) then SItem.mode :=
Ident else if (S[Position] in SetNum) then SItem := Number else if
(S[Position] in SetDiv) then begin if (S[Position] <> ';') then
SItem.mode := Divider else SItem.mode := Part; Inc(Position); exit;
end else begin SItem.mode := None; Inc(Position); exit; end; repeat
Inc(Position); if (SItem.mode = Number)and ((S[Position] = '-

```

```

)or(S[Position] = '+'))and (UpCase(S[Position-1])='E') then SItem.Ident :=
SItem.Ident + S[Position] else if ((Position > Length(S))or(S[Position] in
SetDiv)) then begin if(SItem.mode = Number) then try
SItem.Number := StrToFloat(SItem) except on EConvertError do
SItem.Error := true; end; for i:=1 to NFunc do if
(LowerCase(SItem.Ident) = Functions[i]) then begin
SItem.mode:=Func; SItem.Number:=i; break; end;
SItem.Pos2:=Position-1; exit; end else SItem.Ident := SItem.Ident +
S[Position]; until false; end; procedure ClearPZ; begin ErrorList.Clear;
SetLength(ConstList,0); SetLength(DataList,MConst); SetLength(PZ,0); end;
function CreatePZ(S:string):boolean; var lend:boolean; i:integer;
Assign:boolean; Adress: integer; OldMode: TType; OldS: char;
procedure code; begin SetLength(PZ,High(PZ)+2); case
TrStack[High(TrStack)] of '+': PZ[High(PZ)] := -1; '-': PZ[High(PZ)] := -2;
'*': PZ[High(PZ)] := -3; '/': PZ[High(PZ)] := -4; '^': PZ[High(PZ)] := -5; 'M':
PZ[High(PZ)] := -6; end; end; procedure proc1; begin
SetLength(TrStack,High(TrStack)+2); TrStack[High(TrStack)] :=
SItem.Ident[1]; end; procedure proc2; begin code; TrStack[High(TrStack)] :=
SItem.Ident[1]; end; procedure proc3; begin code;
SetLength(TrStack,High(TrStack)); lend:=false; end; procedure proc4; begin
SetLength(TrStack,High(TrStack)); end; procedure proc5; begin
SetLength(TrStack,High(TrStack)); TrStack[High(TrStack)] :=
Chr(127+Round(SItem.Number)); end; procedure proc6; begin
SetLength(PZ,High(PZ)+2); PZ[High(PZ)] := -Ord(TrStack[High(TrStack)]);
SetLength(TrStack,High(TrStack)); end; begin ClearPZ; SetLength(TrStack,1);
TrStack[0] := '0'; OldMode := None; OldS := ''; Assign := true; Adress := 0;
SyntItem(S,true); if (SItem.mode = All) then begin ErrorList.Add('He
введено никакого текста'); Result := false; exit; end; repeat if
((OldMode = Func)and(SItem.Ident[1] <> '(')) then

```

```

ErrorList.Add('Пропущена скобка после функции в позиции
'+IntToStr(SItem.Pos1)); case SItem.mode of Number: begin
if((OldMode <> Divider)and(OldMode <> None)and (OldMode <>
Part)) then ErrorList.Add('В позиции '+IntToStr(SItem.Pos1)+' должен
быть разделитель'); if (SItem.Error) then
ErrorList.Add('Ошибка в позициях '+IntToStr(SItem.Pos1)+
' - '+IntToStr(SItem.Pos2)) else begin
SetLength(ConstList,High(ConstList)+2); ConstList[High(ConstList)]
:= SItem.Number; SetLength(PZ,High(PZ)+2); PZ[High(PZ)]
:= High(ConstList); end; Assign:=false; end; Ident:
begin if((OldMode <> Divider)and(OldMode <> None)and
(OldMode <> Part)) then ErrorList.Add('В позиции
'+IntToStr(SItem.Pos1)+' должен быть разделитель'); for i:=0 to
High(DataList) do begin if (UpperCase(SItem.Ident) =
DataList[i].Name) then begin SetLength(PZ,High(PZ));
PZ[High(PZ)] := NConst+i; break; end; if(i =
High(DataList)) then begin
SetLength(DataList,High(DataList)+2);
DataList[High(DataList)].Name:=UpperCase(SItem.Ident);
DataList[High(DataList)].Data:=0; SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := NConst+High(DataList); end; end;
end; All,Part:begin repeat lend:=true; case
TrStack[High(TrStack)] of '0': begin if (Adress <> 0) then
begin SetLength(PZ,High(PZ)+3); PZ[High(PZ)-1]
:= -7; PZ[High(PZ)] := Adress; Adress := 0;
end; break; end; '(': ErrorList.Add('Лишняя
открытая скобка'); else proc3; end; until lend; if
(ErrorList.Count = 0) then Result:=true else Result:=false;
if (SItem.mode = All) then exit else begin Assign := true;

```

```

SItem.mode := None;          end          end; Divider:begin
if((OldMode = Divider)and          ((SItem.Ident[1]<>'=')and
(SItem.Ident[1]<>'(')and          (SItem.Ident[1] <> ')'))and          ((OldS <>
')and          (OldS <> ')')and          (OldS <> '=')))          then begin
ErrorList.Add('Позиция '+IntToStr(SItem.Pos1)+          ': два
поряд символа операции');          break;          end;          repeat
lend:=true;          case SItem.Ident[1] of          '=': if Assign and (OldMode =
Ident)          then begin          Adress := PZ[High(PZ)];
SetLength(PZ,High(PZ));          SItem.mode := None;          end
else ErrorList.Add('Позиция '+IntToStr(SItem.Pos1)+          ':
символ "=" здесь недопустим');          '(: if(OldMode = Ident) or (OldMode
= Number)          then ErrorList.Add('Ошибочная скобка в позиции
'+IntToStr(SItem.Pos1))          else proc1;          '+','-', 'M': begin
if((OldMode = None)or(OldS = '('))          then if (SItem.Ident[1] = '+')
then break          else SItem.Ident[1] := 'M';          case
TrStack[High(TrStack)] of          '0','(: proc1;          '+','-', 'M': proc2;
'*','/','^': proc3;          end;          end;          '*','/':          if OldS
= '('          then ErrorList.Add('Ошибочная операция в позиции
'+IntToStr(SItem.Pos1))          else          case
TrStack[High(TrStack)] of          '0','(','+', '-', 'M': proc1;          '*','/':
proc2;          '^': proc3;          end;          '^':          if OldS = '('
then ErrorList.Add('Ошибочная операция в позиции '+IntToStr(SItem.Pos1))
else          case TrStack[High(TrStack)] of          '0','(','+', '-', '*','/','M':
proc1;          '^': proc2;          end;          ')':          case
TrStack[High(TrStack)] of          '0': ErrorList.Add('Лишняя закрытая
скобка');          '(: begin          proc4;          if
(Ord(TrStack[High(TrStack)]) > 127)          then proc6;
end;          '+','-', '*','/','^','M': proc3;          end;          end;
until lend;          Assign:=false;          end; Func: begin          repeat

```

```

lend:=true;      proc5      until lend;      Assign:=false;      end;
None:  ErrorList.Add('Непонятный символ в позиции
'+IntToStr(SItem.Pos1));  end;  OldMode := SItem.mode;  OldS :=
SItem.Ident[1];  SyntItem(S);  until false;  if(ErrorList.Count = 0) then Result
:= true  else Result := false;  end;  function Calculate(var R:real):boolean;  var
Stack: array of real;  i:integer;  begin  for i:=0 to High(PZ) do begin  if (i > 0)
then  if (PZ[i-1] = -7) and (i < High(PZ)) then Continue;  if PZ[i] < -100
then begin  try  case -PZ[i]-100 of  1:
Stack[High(Stack)]:=Exp(Stack[High(Stack)]);  2:
Stack[High(Stack)]:=Sin(Stack[High(Stack)]);  3:
Stack[High(Stack)]:=Cos(Stack[High(Stack)]);  4:
Stack[High(Stack)]:=Sqrt(Stack[High(Stack)]);  5:
Stack[High(Stack)]:=Abs(Stack[High(Stack)]);  6:
Stack[High(Stack)]:=Ln(Stack[High(Stack)]);  7:
Stack[High(Stack)]:=Tan(Stack[High(Stack)]);  8:
Stack[High(Stack)]:=ArcTan(Stack[High(Stack)]);  9:
Stack[High(Stack)]:=ArcCos(Stack[High(Stack)]);  10:
Stack[High(Stack)]:=Sqr(Stack[High(Stack)]);  11:
Stack[High(Stack)]:=ArcSin(Stack[High(Stack)]);  end  except  Result
:= false;  exit;  end;  if(FloatToStr(Stack[High(Stack)]) = 'NA') or
(FloatToStr(Stack[High(Stack)]) = 'INF') or
(FloatToStr(Stack[High(Stack)]) = '-INF')  then begin  Result := false;
exit;  end  end  else if PZ[i] < 0  then begin  try  case -PZ[i] of
1: Stack[High(Stack)-1]:=      Stack[High(Stack)-1]+Stack[High(Stack)];
2: Stack[High(Stack)-1]:=      Stack[High(Stack)-1]-Stack[High(Stack)];
3: Stack[High(Stack)-1]:=      Stack[High(Stack)-1]*Stack[High(Stack)];
4: Stack[High(Stack)-1]:=      Stack[High(Stack)-1]/Stack[High(Stack)];
5: Stack[High(Stack)-1]:=      Power(Stack[High(Stack)-
1],Stack[High(Stack)]);  6: Stack[High(Stack)]:= -Stack[High(Stack)];  7:

```

```

DataList[PZ[i+2]-NConst].Data := Stack[High(Stack)]; end; except
Result := false; exit; end; if (PZ[i] <> -6) then
SetLength(Stack,High(Stack)); end else begin
SetLength(Stack,High(Stack)); if (PZ[i] < NConst) then
Stack[High(Stack)]:=ConstList[PZ[i]] else
Stack[High(Stack)]:=DataList[PZ[i]-NConst].Data; end; end; Result :=
true; R :=Stack[High(Stack)]; end; function SetData(Name:string;
Data:real):boolean; var i:integer; begin for i:=MConst to High(DataList) do if
(UpperCase(Name) = DataList[i].Name) then begin DataList[i].Data :=
Data; Result:=true; exit; end; Result := false; end; function
GetData(Name:string; var Data:real):boolean; var i:integer; begin for i:=0 to
High(DataList) do if (UpperCase(Name) = DataList[i].Name) then begin
Data := DataList[i].Data; Result:=true; exit; end; Result := false; end;
initialization SetLength(DataList,MConst); DataList[0].Name:='PI';
DataList[0].Data:=Pi; DataList[1].Name:='E'; DataList[1].Data:=2.71828183;
ErrorList := TStringList.Create; finalization ErrorList.Free; end. unit UErrors;
interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls; type TFEErrors = class(TForm)
LBEErrors: TListBox; private { Private declarations } public { Public
declarations } end; var FErrors: TFEErrors; implementation {$R *.dfm} end.

```

Програма для визначення приведенного коефіцієнту тертя при реалізації

алгоритму дихотомії для трансцендентних рівнянь

```

program Project1; uses Forms, TUDI1 in 'TUDI1.pas' {frmTUDI1},
TUDI2 in 'TUDI2.pas' {frmTUDI2}, Unit1 in 'Unit1.pas' {Form1}, Unit2 in
'Unit2.pas' {Form2}, Unit3 in 'Unit3.pas' {Form3}; {$R *.res} begin
Application.Initialize; Application.CreateForm(TfrmTUDI1, frmTUDI1);
Application.CreateForm(TfrmTUDI2, frmTUDI2);
Application.CreateForm(TForm1, Form1); Application.CreateForm(TForm2,

```

```

Form2); Application.CreateForm(TForm3, Form3); Application.Run; end.
unit TUDI2; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, ComObj, Forms, Dialogs, StdCtrls, TeEngine, Series,
ExtCtrls, TeeProcs, Chart, Math, Grids, Jpeg, Printers, ShellAPI; type
TfrmTUDI2 = class(TForm) btn21TUDI2: TButton; btn23TUDI2: TButton;
StringGrid1: TStringGrid; Button1: TButton; Button2: TButton; Label1:
TLabel; procedure btn23TUDI2Click(Sender: TObject); procedure
btn21TUDI2Click(Sender: TObject); procedure Button1Click(Sender:
TObject); procedure Button2Click(Sender: TObject); private { Private
declarations } public { Public declarations } end; var frmTUDI2:
TfrmTUDI2; a,b,e,X,g,k,vn,teks,rad,vnp,rH,sn,rn,ktren,ug,ktrp,Ktrenie,R:Real;
m1,m2,m3,m4:array[1..9,1..11] of Real; code,n,j1,j2:Integer;
X1,n1,ug1,ktrp1,ktren1:string; implementation uses TUDI1,Unit1; {$R
*.dfm} procedure TfrmTUDI2.btn21TUDI2Click(Sender: TObject); begin
StringGrid1.Cells[0,0]:='rH'; StringGrid1.Cells[1,0]:='R=1.5';
StringGrid1.Cells[2,0]:='R=2'; StringGrid1.Cells[3,0]:='R=2.5';
StringGrid1.Cells[4,0]:='R=3'; StringGrid1.Cells[5,0]:='R=3.5';
StringGrid1.Cells[6,0]:='R=4'; StringGrid1.Cells[7,0]:='R=4.5';
StringGrid1.Cells[8,0]:='R=5'; for j1:=1 to 8 do begin R:=0.5+j1; for j2:=1 to
10 do begin rH:=j2/10; a:=0.0001; b:=1; e:=0.0001; n:=0; repeat
X:=a; g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-x)*sqrt(2*R*x-sqr(x)); if
g>=0 then begin k:=1; end else begin k:=-1; end; X:=(a+b)/2;
g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-x)*sqrt(2*R*x-sqr(x)); g:=k*g;
if g>0 then begin a:=X; end else begin b:=X; end; n:=n+1; until b-
a<e; m1[j1,j2]:=x; m2[j1,j2]:=2*arccos((R-x)/R); m4[j1,j2]:=rH; end;
end; for j1:=1 to 8 do begin for j2:=1 to 10 do begin
StringGrid1.Cells[j1,j2]:=format('%8.4f',[m1[j1,j2]]); end; end; for j2:=1 to
10 do begin StringGrid1.Cells[0,j2]:=format('%4.1f',[m4[1,j2]]); end; end;
procedure XlsWriteCellLabel(XlsStream: TStream; const ACol, ARow: Word;

```

```

const AValue: string); var L: Word; const {$J+} CXlsLabel: array[0..5]
of Word = ($204, 0, 0, 0, 0, 0); {$J-} begin L := Length(AValue);
CXlsLabel[1] := 8 + L; CXlsLabel[2] := ARow; CXlsLabel[3] := ACol;
CXlsLabel[5] := L; XlsStream.WriteBuffer(CXlsLabel, SizeOf(CXlsLabel));
XlsStream.WriteBuffer(Pointer(AValue), L); end; function
SaveAsExcelFile(AGrid: TStringGrid; AFileName: string): Boolean; const
{$J+} CXlsBof: array[0..5] of Word = ($809, 8, 00, $10, 0, 0); {$J-} CXlsEof:
array[0..1] of Word = ($0A, 00); var FStream: TFileStream; I, J: Integer;
begin Result := False; FStream := TFileStream.Create(PChar(AFileName),
fmCreate or fmOpenWrite); try CXlsBof[4] := 0;
FStream.WriteBuffer(CXlsBof, SizeOf(CXlsBof)); for i := 0 to
AGrid.ColCount - 1 do for j := 0 to AGrid.RowCount - 1 do
XlsWriteCellLabel(FStream, I, J, AGrid.cells[i, j]);
FStream.WriteBuffer(CXlsEof, SizeOf(CXlsEof)); Result := True; finally
FStream.Free; end; end; procedure TfrmTUDI2.Button1Click(Sender:
TObject); begin if SaveAsExcelFile(StringGrid1, extractfilepath(paramstr(0))+
'TableForPrint.xls') then
ShellExecute(Application.Handle, PChar('open'), PChar(extractfilepath(paramstr(
0))+ 'TableForPrint.xls'), Nil, Nil, SW_SHOW); {ShowMessage('StringGrid
saved!')}; end; procedure TfrmTUDI2.btn23TUDI2Click(Sender: TObject);
begin frmTUDI1.Close; end; procedure TfrmTUDI2.Button2Click(Sender:
TObject); begin Form1.Show; frmTUDI2.Hide; end; end. unit Unit1; interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, ComObj, Forms, Dialogs, StdCtrls, TeEngine, Series, ExtCtrls,
TeeProcs, Chart, Math, Grids, Jpeg, Printers, ShellAPI; type TForm1 =
class(TForm) Button1: TButton; StringGrid1: TStringGrid; Button2: TButton;
Button3: TButton; Button4: TButton; Label1: TLabel; procedure
Button1Click(Sender: TObject); procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject); procedure Button4Click(Sender:

```

```

TObject); private { Private declarations } public { Public declarations }
end; var Form1: TForm1;
a,b,e,X,g,k,vn,teks,rad,vnp,rH,sn,rm,ktren,ug,ktrp,Ktrenie,R:Real;
m1,m2,m3,m4:array[1..9,1..11] of Real; code,n,j1,j2:Integer;
X1,n1,ug1,ktrp1,ktren1:string; implementation uses TUDI2,Unit2; {$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject); begin Form1.Hide;
frmTUDI2.Show; end; procedure TForm1.Button2Click(Sender: TObject);
begin StringGrid1.Cells[0,0]:='rH'; StringGrid1.Cells[1,0]:='R=1.5';
StringGrid1.Cells[2,0]:='R=2'; StringGrid1.Cells[3,0]:='R=2.5';
StringGrid1.Cells[4,0]:='R=3'; StringGrid1.Cells[5,0]:='R=3.5';
StringGrid1.Cells[6,0]:='R=4'; StringGrid1.Cells[7,0]:='R=4.5';
StringGrid1.Cells[8,0]:='R=5'; for j1:=1 to 8 do begin R:=0.5+j1; for j2:=1 to
10 do begin rH:=j2/10; a:=0.0001; b:=1; e:=0.0001; n:=0; repeat
X:=a; g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-x)*sqrt(2*R*x-sqr(x)); if
g>=0 then begin k:=1; end else begin k:=-1; end; X:=(a+b)/2;
g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-x)*sqrt(2*R*x-sqr(x)); g:=k*g;
if g>0 then begin a:=X; end else begin b:=X; end; n:=n+1; until b-
a<e; m1[j1,j2]:=x; m2[j1,j2]:=2*arccos((R-x)/R); m4[j1,j2]:=rH; end;
end; for j1:=1 to 8 do begin for j2:=1 to 10 do begin
StringGrid1.Cells[j1,j2]:=format('%8.4f',[m2[j1,j2]]); end; end; for j2:=1 to
10 do begin StringGrid1.Cells[0,j2]:=format('%4.1f',[m4[1,j2]]); end; end;
procedure XlsWriteCellLabel(XlsStream: TStream; const ACol, ARow: Word;
const AValue: string); var L: Word; const {$J+} CXlsLabel: array[0..5]
of Word = ($204, 0, 0, 0, 0, 0); {$J-} begin L := Length(AValue);
CXlsLabel[1] := 8 + L; CXlsLabel[2] := ARow; CXlsLabel[3] := ACol;
CXlsLabel[5] := L; XlsStream.WriteBuffer(CXlsLabel, SizeOf(CXlsLabel));
XlsStream.WriteBuffer(Pointer(AValue), L); end; function
SaveAsExcelFile(AGrid: TStringGrid; AFileName: string): Boolean; const
{$J+} CXlsBof: array[0..5] of Word = ($809, 8, 00, $10, 0, 0); {$J-} CXlsEof:

```

```

array[0..1] of Word = ($0A, 00); var  FStream: TFileStream;  I, J: Integer;
begin  Result := False;  FStream := TFileStream.Create(PChar(AFileName),
fmCreate or fmOpenWrite);  try  CXlsBof[4] := 0;
FStream.WriteBuffer(CXlsBof, SizeOf(CXlsBof));  for i := 0 to
AGrid.ColCount - 1 do  for j := 0 to AGrid.RowCount - 1 do
XlsWriteCellLabel(FStream, I, J, AGrid.cells[i, j]);
FStream.WriteBuffer(CXlsEof, SizeOf(CXlsEof));  Result := True;  finally
FStream.Free;  end; end; procedure TForm1.Button3Click(Sender: TObject);
begin if SaveAsExcelFile(StringGrid1, extractfilepath(paramstr(0))+
'TableForPrint.xls') then
ShellExecute(Application.Handle,PChar('open'),PChar(extractfilepath(paramstr(
0))+ 'TableForPrint.xls'),Nil,Nil,SW_SHOW);  {ShowMessage('StringGrid
saved!')}; end; procedure TForm1.Button4Click(Sender: TObject); begin
Form1.Hide; Form2.Show;  end; end. unit Unit2; interface uses  Windows,
Messages, SysUtils, Variants, Classes, Graphics, Controls,ComObj , Forms,
Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart,Math,
Grids,Jpeg,Printers, ShellAPI; type  TForm2 = class(TForm) Button1:
TButton; Button2: TButton; StringGrid1: TStringGrid; Button3: TButton;
Button4: TButton; Label1: TLabel; Edit1: TEdit; Label2: TLabel; procedure
Button1Click(Sender: TObject); procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject); procedure Button4Click(Sender:
TObject); private { Private declarations } public { Public declarations }
end; var  Form2: TForm2;
a,b,e,X,g,k,vn,teks,rad,vnp,rH,sn,rn,ktren,ug,ktrp,Ktrenie,R:Real;
m1,m2,m3,m4:array[1..9,1..11] of Real;  code,n,j1,j2:Integer;
X1,n1,ug1,ktrp1,ktren1:string;  implementation uses Unit1,Unit3; {$R *.dfm}
procedure TForm2.Button1Click(Sender: TObject); begin Form1.Show;
Form2.Hide; end; procedure TForm2.Button2Click(Sender: TObject); begin
val(Edit1.Text,Ktrenie,code);  StringGrid1.Cells[0,0]:='rH';

```

```

StringGrid1.Cells[1,0]:='R=1.5';   StringGrid1.Cells[2,0]:='R=2';
StringGrid1.Cells[3,0]:='R=2.5';   StringGrid1.Cells[4,0]:='R=3';
StringGrid1.Cells[5,0]:='R=3.5';   StringGrid1.Cells[6,0]:='R=4';
StringGrid1.Cells[7,0]:='R=4.5';   StringGrid1.Cells[8,0]:='R=5';  for j1:=1 to
8 do begin  R:=0.5+j1; for j2:=1 to 10 do begin  rH:=j2/10;  a:=0.0001;  b:=1;
e:=0.0001; n:=0;  repeat  X:=a; g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-
x)*sqrt(2*R*x-sqr(x));  if g>=0 then begin k:=1; end else begin k:=-1;
end; X:=(a+b)/2; g:=3.14*sqr(rH)-sqr(R)*arccos((R-x)/R)+(R-x)*sqrt(2*R*x-
sqr(x)); g:=k*g;  if g>0 then begin a:=X; end else begin b:=X; end;
n:=n+1;  until b-a<e;  m1[j1,j2]:=x;  m2[j1,j2]:=2*arccos((R-x)/R);
m3[j1,j2]:=4*Ktrenie*sin(arccos((R-x)/R))/ 2*arccos((R-
x)/R)+sin(2*arccos((R-x)/R));  m4[j1,j2]:=rH;  end;  end;  for j1:=1 to 8
do begin  for j2:=1 to 10 do begin
StringGrid1.Cells[j1,j2]:=format('%8.4f',[m3[j1,j2]]);  end;  end; for j2:=1 to
10 do begin StringGrid1.Cells[0,j2]:=format('%4.1f',[m4[1,j2]]); end;  end;
procedure XlsWriteCellLabel(XlsStream: TStream; const ACol, ARow: Word;
const AValue: string); var  L: Word; const  {$J+}  CXlsLabel: array[0..5]
of Word = ($204, 0, 0, 0, 0, 0);  {$J-}  begin  L := Length(AValue);
CXlsLabel[1] := 8 + L;  CXlsLabel[2] := ARow;  CXlsLabel[3] := ACol;
CXlsLabel[5] := L;  XlsStream.WriteBuffer(CXlsLabel, SizeOf(CXlsLabel));
XlsStream.WriteBuffer(Pointer(AValue) , L); end;  function
SaveAsExcelFile(AGrid: TStringGrid; AFileName: string): Boolean; const
{$J+} CXlsBof: array[0..5] of Word = ($809, 8, 00, $10, 0, 0); {$J-}  CXlsEof:
array[0..1] of Word = ($0A, 00); var  FStream: TFileStream;  I, J: Integer;
begin  Result := False;  FStream := TFileStream.Create(PChar(AFileName),
fmCreate or fmOpenWrite);  try  CXlsBof[4] := 0;
FStream.WriteBuffer(CXlsBof, SizeOf(CXlsBof));  for i := 0 to
AGrid.ColCount - 1 do  for j := 0 to AGrid.RowCount - 1 do
XlsWriteCellLabel(FStream, I, J, AGrid.cells[i, j]);

```

```

FStream.WriteBuffer(CXlsEof, SizeOf(CXlsEof)); Result := True; finally
FStream.Free; end; end; procedure TForm2.Button3Click(Sender: TObject);
begin if SaveAsExcelFile(StringGrid1, extractfilepath(paramstr(0))+
'TableForPrint.xls') then
ShellExecute(Application.Handle,PChar('open'),PChar(extractfilepath(paramstr(
0))+ 'TableForPrint.xls'),Nil,Nil,SW_SHOW); {ShowMessage('StringGrid
saved!')}; end; procedure TForm2.Button4Click(Sender: TObject); begin
Form2.Hide; Form3.Show; end; end. unit Unit3; interface uses Windows,
Messages, SysUtils, Variants, Classes, Graphics, Controls,ComObj , Forms,
Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart,Math,
Grids,Jpeg,Printers, ShellAPI; type TForm3 = class(TForm) Button1:
TButton; Label1: TLabel; Edit1: TEdit; Label2: TLabel; Edit2: TEdit;
Label3: TLabel; Label4: TLabel; Edit3: TEdit; Label5: TLabel; Edit4: TEdit;
GroupBox1: TGroupBox; Label6: TLabel; Label7: TLabel; Label8: TLabel;
Label9: TLabel; Label10: TLabel; Label11: TLabel; GroupBox2: TGroupBox;
Label12: TLabel; Label13: TLabel; Label14: TLabel; Label15: TLabel; Edit5:
TEdit; Button2: TButton; Memo1: TMemo; Button3: TButton; Chart1:
TChart; Chart2: TChart; Chart3: TChart; Chart4: TChart; Series1:
TBarSeries; Series2: TBarSeries; Series3: TBarSeries; Series4: TBarSeries;
Series5: TBarSeries; Series6: TBarSeries; Series7: TBarSeries; procedure
Button1Click(Sender: TObject); procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var Form3: TForm3;
a,b,e,X,g,k,vn,teks,rad,vnp,rn,sn,ktren,ug,ktrp:Real; code,n:Integer;
X1,n1,ug1,ktrp1,ktren1:string; implementation uses Unit2; {$R *.dfm}
procedure TForm3.Button1Click(Sender: TObject); begin Form2.Show;
Form3.Hide; end; procedure TForm3.Button2Click(Sender: TObject); begin
val(edit1.Text,vn,code); val(edit2.Text,teks,code); val(edit3.Text,rad,code);
val(edit4.Text,vnp,code); a:=0.0001; b:=1; e:=0.0001; if vn=1 then begin

```

```

rn:=0.0395*sqrt(teks)/2; sn:=3.14*sqr(rn); end else begin if vn=2 then
begin rn:=0.0427*sqrt(teks)/2; sn:=3.14*sqr(rn); end else begin if vn=3
then begin rn:=0.0411*sqrt(teks)/2; sn:=3.14*sqr(rn); end else begin if
vn=4 then begin rn:=0.0411*sqrt(teks)/2; sn:=3.14*sqr(rn); end else begin
if vn=5 then begin rn:=0.0386*sqrt(teks)/2; sn:=3.14*sqr(rn); end
else begin rn:=0.0411*sqrt(teks)/2; sn:=3.14*sqr(rn); end; end;
end; end; end; if vnp=1 then begin if vn=1 then begin ktren:=0.169; end
else begin if vn=2 then begin ktren:=0.143; end else begin if vn=3
then begin ktren:=0.121; end else begin if vn=4 then begin
ktren:=0.152; end else begin if vn=5 then begin ktren:=0.196; end
else begin ktren:=0.159; end; end; end; end; end; end else begin
if vn=1 then begin ktren:=0.27; end else begin if vn=2 then begin
ktren:=0.24; end else begin if vn=3 then begin ktren:=0.2; end else
begin if vn=4 then begin ktren:=0.25; end else begin if vn=5 then
begin ktren:=0.31; end else begin ktren:=0.26; end; end; end;
end; end; end; n:=0; repeat X:=a; g:=sn-sqr(rad)*arccos((rad-
x)/rad)+(rad-x)*sqrt(2*rad*x-sqr(x)); if g>=0 then begin k:=1; end else
begin k:=-1; end; X:=(a+b)/2; g:=sn-sqr(rad)*arccos((rad-x)/rad)+(rad-
x)*sqrt(2*rad*x-sqr(x)); g:=k*g; if g>0 then begin a:=X; end else
begin b:=X; end; n:=n+1; until b-a<e; X1:=format('%10.4f',[X]);
n1:=format('%5.0d',[n]); Memo1.Lines.Add(' h='+X1); Edit5.Text:=n1;
ug:=2*arccos((rad-x)/rad); ktrp:=4*ktren*sin(ug/2)/(ug+sin(ug));
ug1:=format('%10.4f',[ug]); ktrp1:=format('%10.4f',[ktrp]);
ktren1:=format('%10.4f',[ktren]); Memo1.Lines.Add('К трен.='+ktren1);
Memo1.Lines.Add('Кут рад.охоп.='+ug1); Memo1.Lines.Add('К
трен.прив.='+ktrp1); end; procedure TForm3.Button3Click(Sender: TObject);
begin Series1.Add(teks,'T',clRed); Series2.Add(rad,'R',clGreen);
Series3.Add(rn,'Rn',clRed); Series4.Add(sn,'Sn',clBlue);

```

```
Series5.Add(x,'h',clYellow); Series6.Add(ktren,'Kt',clYellow);
Series7.Add(ktrp,'Ktp',clBlue); end; end.
```

Програма для визначення значення натягу при взаємодії нитки з напрямною з використанням зворотних польських записів

```
program PA; uses Forms, NN1 in 'NN1.PAS' {frm1NN1}, NN2 in
'NN2.pas' {frm1NN2}, NN3 in 'NN3.pas' {frm1NN3}, NN4 in 'NN4.pas'
{frm1NN4}, Synt in 'SYNT.PAS', UErrors in 'UERRORS.PAS' {FErrors};
{$R *.res} begin Application.Initialize; Application.CreateForm(Tfrm1NN1,
frm1NN1); Application.CreateForm(Tfrm1NN2, frm1NN2);
Application.CreateForm(Tfrm1NN3, frm1NN3);
Application.CreateForm(Tfrm1NN4, frm1NN4);
Application.CreateForm(TFEErrors, FErrors); Application.Run; end. unit NN1;
interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, Math, StdCtrls, jpeg, ExtCtrls; type Tfrm1NN1 =
class(TForm) btn1NN1: TButton; lbl1NN1: TLabel; lbl2NN1: TLabel;
lbl3NN1: TLabel; lbl4NN1: TLabel; Image1: TImage; procedure
btn1NN1Click(Sender: TObject); private { Private declarations } public {
Public declarations } end; var frm1NN1: Tfrm1NN1; implementation uses
NN2, NN3, NN4; {$R *.dfm} procedure Tfrm1NN1.btn1NN1Click(Sender:
TObject); begin frm1NN1.Hide; frm1NN2.Show; end; end. unit NN2; interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, Math, StdCtrls, Menus, TeEngine, Series, ExtCtrls, TeeProcs,
Chart, Printers; type Tfrm1NN2 = class(TForm) MainMenu1: TMainMenu;
n1NN2: TMenuItem; n2NN2: TMenuItem; n3NN2: TMenuItem; n4NN2:
TMenuItem; n5NN2: TMenuItem; n6NN2: TMenuItem; n7NN2: TMenuItem;
n8NN2: TMenuItem; n9NN2: TMenuItem; n10NN2: TMenuItem; n11NN2:
TMenuItem; n12NN2: TMenuItem; n13NN2: TMenuItem; lbl1NN2: TLabel;
lbl2NN2: TLabel; lbl3NN2: TLabel; lbl4NN2: TLabel; edt1NN2: TEdit;
```

```

edt2NN2: TEdit; edt3NN2: TEdit; edt4NN2: TEdit; lbl5NN2: TLabel; Memo1:
TMemo; lbl6NN2: TLabel; mem1NN2: TMemo; cht1NN2: TChart; Series1:
TLineSeries; lbl7NN2: TLabel; edt5NN2: TEdit; procedure
btn1NN2Click(Sender: TObject); procedure n11NN2Click(Sender: TObject);
procedure n5NN2Click(Sender: TObject); procedure n2NN2Click(Sender:
TObject); procedure n7NN2Click(Sender: TObject); procedure
n8NN2Click(Sender: TObject); procedure n9NN2Click(Sender: TObject);
procedure n3NN2Click(Sender: TObject); procedure n4NN2Click(Sender:
TObject); private { Private declarations } public { Public declarations } end;
var frm1NN2: Tfrm1NN2;
x,F,a,b,e,h,c,R1,kt,d0,d,V1,SS0,SS1,GG0,GG1,USS0,USS1,UGG0,UGG1,L1,L
2,L3,ktp,V1P,L1P,L2P,V11P,L3P:Real; k,code,R11,i:Integer;
aa:array[1..50]of Real; kk,kkk,kkk1:String; w:Integer; implementation uses
NN1,Synt,UErrors,NN3,NN4; {$R *.dfm} procedure v(var
F:Real;X,P0,b1,r,E1,B0,FP,a2,b2,R1,d0,kt,V1:Real); begin
SetData('P0',P0); SetData('b1',b1); SetData('r',r); SetData('E1',E1);
SetData('B0',B0); SetData('FP',FP); SetData('a2',a2); SetData('b2',b2);
SetData('R1',R1); SetData('d0',d0); SetData('kt',kt); SetData('V1',V1);
SetData('X',X); Calculate(F); end; procedure
Tfrm1NN2.btn1NN2Click(Sender: TObject); begin frm1NN1.Close; end;
procedure Tfrm1NN2.n11NN2Click(Sender: TObject); begin frm1NN3.Show;
end; procedure Tfrm1NN2.n5NN2Click(Sender: TObject); begin
frm1NN1.Close; end; procedure Tfrm1NN2.n2NN2Click(Sender: TObject);
begin frm1NN4.Show; end; procedure Tfrm1NN2.n7NN2Click(Sender:
TObject); begin Val(edt1NN2.Text,a,code); Val(edt2NN2.Text,b,code);
Val(edt3NN2.Text,e,code); Val(edt4NN2.Text,h,code); for R11:=2 to 28
do begin R1:=R11/4; i:=R11-1; if (FErrors <> nil) then
FErrors.Close; if not CreatePZ(Memo1.Text) then begin
Application.CreateForm(TFErrors, FErrors);

```



```

FErrors.LBErrors.Items.Assign(ErrorList); FErrors.Show; exit; end;    c:=h;
k:=0; x:=a;    kt:=a2/exp(b2*ln(P0/R1));
d0:=P0*(R1+r)/(r*P0+E1*b1*sqr(R1+r));    if d0>1 then d0:=1;
d:=d0*exp(kt*FP);    if d>1 then d:=1;    SS0:=1-d0*sqr(2*r/R1);    SS1:=1-
d*sqr(2*r/R1);    GG0:=1-(B0/(2*P0*sqr(R1+r)));    GG1:=1-
(B0/(2*X*sqr(R1+r)));    USS0:=1.57-arctan(SS0/sqrt(1-SS0*SS0));
USS1:=1.57-arctan(SS1/sqrt(1-SS1*SS1));    UGG0:=1.57-arctan(GG0/sqrt(1-
GG0*GG0));    UGG1:=1.57-arctan(GG1/sqrt(1-GG1*GG1));
V1:=FP+USS0+USS1-UGG0-UGG1;
V(F,X,P0,b1,r,E1,B0,FP,a2,b2,R1,d0,kt,V1);    w:= Trunc(F/abs(F));    repeat
x:=x+c;    if x-c>=b then Break;
V(F,X,P0,b1,r,E1,B0,FP,a2,b2,R1,d0,kt,V1);    if F*w/c>0 then Continue;
c:=-c/4;    if abs(c)>(e/4) then Continue;    k:=k+1;
kk:=format('%5.2f',[R1]);    kkk:=format('%17.8f',[x]);    aa[i]:=x;
mem1NN2.Lines.Add('P('+kk+')='+'kkk');    c:=h;    w:=-w;    until False;
end; end; procedure Tfrm1NN2.n8NN2Click(Sender: TObject); begin with
Series1 do for i:=1 to 27 do    begin    R1:=(i+1)/4;
series1.AddXY(R1,aa[i],"clRed);    end; end; procedure V2(var
F:Real;X:Real);    begin    kt:=a2/exp(b2*ln(P0/X));
d0:=P0*(X+r)/(r*P0+E1*b1*sqr(X+r));    if d0>1 then d0:=1;
d:=d0*exp(kt*FP);    if d>1 then d:=1;    SS0:=1-d0*sqr(2*r/X);    SS1:=1-
d*sqr(2*r/X);    USS0:=1.57-arctan(SS0/sqrt(1-SS0*SS0));    USS1:=1.57-
arctan(SS1/sqrt(1-SS1*SS1));    V1:=FP+USS0+USS1;    L1:=X+r; L2:=r*d0;
L3:=exp(kt*V1);    ktp:=a2*b2*power(X/P0,b2)/X;    L1P:=1;
L2P:=r*P0*(((r*P0+E1*b1*sqr(X+r))-
2*E1*b1*sqr(X+r))/sqr(r*P0+E1*b1*sqr(X+r)));    V11P:=(4*r/sqr(X))*((L2P-
(L2*2/X))/sqrt(1-sqr(1-4*L2*r/sqr(X)))+exp(kt*FP)*(L2P+L2*(ktp*FP-
(2/X)))/sqrt(1-sqr(1-(4*L2*r*exp(kt*FP)/sqr(X)))));
L3P:=exp(kt*V1)*(ktp*V1+kt*V11P);    F:=L3P*sqr(L1)-L1P*L3*L2-

```

```

L1*L3P*L2+L2P*L1*L3; end; procedure Tfrm1NN2.n9NN2Click(Sender:
TObject); begin a:=0.6; b:=5; e:=0.01; h:=0.1; c:=h; k:=0; x:=a;
V2(F,X); w:= Trunc(F/abs(F)); repeat x:=x+c; if x-c>=b then
Break; V2(F,X); if F*w/c>0 then Continue; c:=-c/4; if abs(c)>(e/4)
then Continue; k:=k+1; kkk1:=format('%17.8f',[x/2]);
edt5NN2.Text:=(R='+kkk1); c:=h; w:=-w; until False; end;
procedure Tfrm1NN2.n3NN2Click(Sender: TObject); var Prn:TextFile;
k:Integer; begin AssignPrn(Prn); Rewrite(Prn);
Printer.Canvas.Font:=mem1NN2.Font; for k:=0 to mem1NN2.Lines.Count-1
do WriteLn(Prn,mem1NN2.Lines[k]); CloseFile(Prn); end; procedure
Tfrm1NN2.n4NN2Click(Sender: TObject); begin cht1NN2.Print; end; end. unit
NN3; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms, Dialogs,Math, StdCtrls; type Tfrm1NN3 =
class(TForm) lbl1NN3: TLabel; lbl2NN3: TLabel; lbl3NN3: TLabel; lbl4NN3:
TLabel; lbl5NN3: TLabel; btn1NN3: TButton; Label1: TLabel; procedure
btn1NN3Click(Sender: TObject); private { Private declarations } public {
Public declarations } end; var frm1NN3: Tfrm1NN3; implementation uses
NN2; {$R *.dfm} procedure Tfrm1NN3.btn1NN3Click(Sender: TObject);
begin frm1NN3.Close; end; end. unit NN4; interface uses Windows,
Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs,Math, StdCtrls; type Tfrm1NN4 = class(TForm) lbl2NN4: TLabel;
lbl3NN4: TLabel; lbl4NN4: TLabel; lbl5NN4: TLabel; lbl6NN4: TLabel;
lbl7NN4: TLabel; lbl8NN4: TLabel; lbl9NN4: TLabel; edt2NN4: TEdit;
edt3NN4: TEdit; edt4NN4: TEdit; edt5NN4: TEdit; edt6NN4: TEdit; edt7NN4:
TEdit; edt8NN4: TEdit; edt9NN4: TEdit; btn1NN4: TButton; procedure
btn1NN4Click(Sender: TObject); private { Private declarations } public {
Public declarations } end; var frm1NN4: Tfrm1NN4; code:Integer;
P0,b1,r,E1,B0,FP,a2,b2:Real; implementation uses NN2,Synt,UErrors; {$R
*.dfm} procedure Tfrm1NN4.btn1NN4Click(Sender: TObject); begin

```

```

val(edt2NN4.Text,P0,code);   val(edt3NN4.Text,b1,code);
val(edt4NN4.Text,r,code);   val(edt5NN4.Text,E1,code);
val(edt6NN4.Text,B0,code);   val(edt7NN4.Text,FP,code);
val(edt8NN4.Text,a2,code);   val(edt9NN4.Text,b2,code); frm1NN4.Hide;
end; end. unit Synt; interface uses classes; type TData = record   Name:
string;   Data:real; end; NConst: integer = 100;   ErrorList: TStringList;   PZ:
array of integer;   DataList: array of TData; const MConst = 2; procedure
SyntItem(S:string; First:boolean=false; Pos:Integer=1); function
CreatePZ(S:string):boolean; {Расчет по польской записи} function
Calculate(var R:real):boolean; function SetData(Name:string;
Data:real):boolean; function GetData(Name:string; var Data:real):boolean;
implementation uses Sysutils, Math, Dialogs,NN2; type TType = (None,
Number, Divider, Ident, Func, Part, All);   TSynt = record mode: TType;
Number:real;   Ident:string;   Error:boolean; Pos1,Pos2:integer; SetNum: set of
char=['0'..'9', ',','];   SetDiv: set of char=[';', '(', ')', '=', '+', '-', '/', '*', '^',   '{', '}',
#13];   SetChar: set of char=['a'..'z','A'..'Z','_'] NFunc = 10 Functions:
array[1..NFunc] of string =
('exp','sin','cos','sqrt','abs','ln','tg','arctan','arccos','sqr'); var   SItem: TSynt;
TrStack: array of char;   ConstList: array of real;   Position: Integer; procedure
SyntItem(S:string;First:boolean=false;Pos:Integer=1); First=true var i:integer;
begin   if (S = "") then begin   SItem.mode := All;   exit; end; Position = if(First)
then Position := Pos;   repeat if (S[Position] = '{')   then begin repeat
Inc(Position) until (Position >= Length(S)) or (S[Position] = '}'); Inc(Position);
end; if(Position <= Length(S)) then   while ((S[Position] = ')')or (S[Position] =
#13)or (S[Position] = #10)or (S[Position] = #0))   do Inc(Position); until
(S[Position] <> '{');   SItem.Error:=false; SItem.Pos1:=Position; Position >
mode = All   if(Position > Length(S)) then begin   SItem.mode := All;   exit;
end; Ident SItem.Ident := S[Position]; if (S[Position] in SetChar) then
SItem.mode := Ident   else if (S[Position] in SetNum) then Item.mode :=

```

```

Number else if (S[Position] in SetDiv) then if (S[Position] <> ';') then
SIItem.mode := Divider else SIItem.mode := Part Inc(Position); exit; end else
begin SIItem.mode := None; Inc(Position); exit; end Inc(Position); if
(SIItem.mode = Number)and ((S[Position] = '-')or(S[Position] = '+'))and
(UpCase(S[Position-1])='E') then SIItem.Ident := SIItem.Ident + S[Position] else
if ((Position > Length(S))or(S[Position] in SetDiv)) then begin if(SIItem.mode =
Number) then try SIItem.Number := StrToFloat(SIItem.Ident) except on
EConvertError do SIItem.Error := true; end; for i:=1 to NFunc do if
(LowerCase(SIItem.Ident) = Functions[i]) then begin SIItem.mode:=Func;
SIItem.Number:=i; break; end; SIItem.Pos2:=Position-1; exit; end else
SIItem.Ident := SIItem.Ident + S[Position]; until false; end; procedure ClearPZ;
begin ErrorList.Clear; SetLength(ConstList,0); SetLength(DataList,MConst);
SetLength(PZ,0); end; function CreatePZ(S:string):boolean; var lend:boolean;
i:integer; Assign:boolean; Adress: integer; OldMode: TType; OldS: char;
procedure code; begin SetLength(PZ,High(PZ)+2); case
TrStack[High(TrStack)] of '+': PZ[High(PZ)] := -1; '-': PZ[High(PZ)] := -2;
'*': PZ[High(PZ)] := -3; '/': PZ[High(PZ)] := -4; '^': PZ[High(PZ)] := -5; 'M':
PZ[High(PZ)] := -6; end; end; procedure proc1; begin
SetLength(TrStack,High(TrStack)+2); TrStack[High(TrStack)] :=
SIItem.Ident[1]; end; procedure proc2; begin code; TrStack[High(TrStack)] :=
SIItem.Ident[1]; end; procedure proc3; begin code;
SetLength(TrStack,High(TrStack)); lend:=false; end; procedure proc4; begin
SetLength(TrStack,High(TrStack)); end; procedure proc5; // Аналог proc1 для
функцій begin SetLength(TrStack,High(TrStack)+2); TrStack[High(TrStack)]
:= Chr(127+Round(SIItem.Number)); end; procedure proc6; // Аналог proc3
для функцій begin SetLength(PZ,High(PZ)+2); PZ[High(PZ)] := -
Ord(TrStack[High(TrStack)])+27; SetLength(TrStack,High(TrStack)); end;
begin ClearPZ; SetLength(TrStack,1); TrStack[0] := '0'; OldMode := None;
OldS := ' '; Assign := true; Adress := 0; // Чтение первой лексемы

```

```

SyntItem(S,true); if (SItem.mode = All) then begin  ErrorList.Add();  Result
:= false;  exit;  end;  repeat  if ((OldMode = Func)and(SItem.Ident[1] <> '('))
then ErrorList.Add('+IntToStr(SItem.Pos1));  case SItem.mode of  Number:
begin  if((OldMode <> Divider)and(OldMode <> None)and  (OldMode <>
Part))  then ErrorList.Add('+IntToStr(SItem.Pos1)+');  if (SItem.Error)
then ErrorList.Add(' '+IntToStr(SItem.Pos1)+      ' - '+IntToStr(SItem.Pos2))
else begin  SetLength(ConstList,High(ConstList)+2);
ConstList[High(ConstList)] := SItem.Number;  SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := High(ConstList);  end;  Assign:=false;  end;  Ident:
begin  if((OldMode <> Divider)and(OldMode <> None)and  (OldMode <>
Part))  then ErrorList.Add('В позиции '+IntToStr(SItem.Pos1)+');  for i:=0
to High(DataList) do  begin  if (UpperCase(SItem.Ident) = DataList[i].Name)
then begin  SetLength(PZ,High(PZ)+2);  PZ[High(PZ)] := NConst+i;
break;  end;  if(i = High(DataList)) then  begin
SetLength(DataList,High(DataList)+2);
DataList[High(DataList)].Name:=UpperCase(SItem.Ident);
DataList[High(DataList)].Data:=0;  SetLength(PZ,High(PZ)+2);
PZ[High(PZ)] := NConst+High(DataList);  end;  end;  end;
All,Part:begin  repeat  lend:=true;  case TrStack[High(TrStack)] of  '0':
begin  if (Adress <> 0) then begin  SetLength(PZ,High(PZ)+3);
PZ[High(PZ)-1] := -7;  PZ[High(PZ)] := Adress;  Adress := 0;  end;
break;  end;  '(': ErrorList.Add(');  else proc3;  end;  until lend;  if
(ErrorList.Count = 0)  then Result:=true  else Result:=false;  if
(SItem.mode = All) then exit  else begin  Assign := true;  SItem.mode :=
None;  end  end;  Divider:begin  if((OldMode = Divider)and
((SItem.Ident[1]<>'=')and  (SItem.Ident[1]<>'(')and  (SItem.Ident[1] <>
'))and  ((OldS <> '(')and  (OldS <> ')')and  (OldS <> '=')))  then begin
ErrorList.Add(' '+IntToStr(SItem.Pos1)+      ');  break;  end;  repeat
lend:=true;  case SItem.Ident[1] of  '=': if Assign and (OldMode = Ident)

```

```

then begin  Adress := PZ[High(PZ)];  SetLength(PZ,High(PZ));
SIItem.mode := None;  end  else ErrorList.Add('+IntToStr(SIItem.Pos1)+
': ');  '(: if(OldMode = Ident) or (OldMode = Number)  then
ErrorList.Add('+IntToStr(SIItem.Pos1))  else proc1;  '+','-',M': begin
if((OldMode = None)or(OldS = '('))  then if (SIItem.Ident[1] = '+' then break
else SIItem.Ident[1] := 'M';  case TrStack[High(TrStack)] of  '0','(: proc1;
'+','-',M': proc2;  '*','/','^': proc3;  end;  end;  '*','/':  if OldS = '('
then ErrorList.Add('+IntToStr(SIItem.Pos1))  else  case
TrStack[High(TrStack)] of  '0','(','+','-','M': proc1;  '*','/': proc2;  '^': proc3;
end;  '^':  if OldS = '('  then ErrorList.Add('+IntToStr(SIItem.Pos1))
else  case TrStack[High(TrStack)] of  '0','(','+','-','*','/','M': proc1;  '^':
proc2;  end;  ')':  case TrStack[High(TrStack)] of  '0': ErrorList.Add();
'(: begin  proc4;  if (Ord(TrStack[High(TrStack)]) > 127)  then proc6;
end;  '+','-',*','/','^','M': proc3;  end;  end;  until lend;  Assign:=false;
end;  Func:  begin  repeat  lend:=true;  proc5  until lend;
Assign:=false;  end;  None:  ErrorList.Add('+IntToStr(SIItem.Pos1));  end;
OldMode := SIItem.mode;  OldS := SIItem.Ident[1];  SyntItem(S);  until false;
if(ErrorList.Count = 0) then Result := true  else Result := false; end;  function
Calculate(var R:real):boolean; var Stack: array of real; i:integer; begin  for i:=0
to High(PZ) do begin  if (i > 0) then if (PZ[i-1] = -7) and (i < High(PZ)) then
Continue;  if PZ[i] < -100 then begin  try  case -PZ[i]-100 of  1:
Stack[High(Stack)]:=Exp(Stack[High(Stack)));  2:
Stack[High(Stack)]:=Sin(Stack[High(Stack)));  3:
Stack[High(Stack)]:=Cos(Stack[High(Stack)));  4:
Stack[High(Stack)]:=Sqrt(Stack[High(Stack)));  5:
Stack[High(Stack)]:=Abs(Stack[High(Stack)));  6:
Stack[High(Stack)]:=Ln(Stack[High(Stack)));  7:
Stack[High(Stack)]:=Tan(Stack[High(Stack)));  8:
Stack[High(Stack)]:=ArcTan(Stack[High(Stack)));  9:

```

```

Stack[High(Stack)]:=ArcCos(Stack[High(Stack)]); 10:
(FloatToStr(Stack[High(Stack)]) = 'INF') or (FloatToStr(Stack[High(Stack)])
= '-INF') then begin Result := false; exit; end end else if PZ[i] < 0 then
begin try case -PZ[i] of 1: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]+Stack[High(Stack)]; 2: Stack[High(Stack)-1]:= Stack[High(Stack)-1]-
Stack[High(Stack)]; 3: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]*Stack[High(Stack)]; 4: Stack[High(Stack)-1]:= Stack[High(Stack)-
1]/Stack[High(Stack)]; 5: Stack[High(Stack)-1]:= Power(Stack[High(Stack)-
1],Stack[High(Stack)]); 6: Stack[High(Stack)]:= -Stack[High(Stack)]; 7:
DataList[PZ[i+1]-NConst].Data := Stack[High(Stack)]; end; except Result :=
false; exit; end; if (PZ[i] <> -6) then SetLength(Stack,High(Stack)); end else
begin SetLength(Stack,High(Stack)+2); if (PZ[i] < NConst) then
Stack[High(Stack)]:=ConstList[PZ[i]] else
Stack[High(Stack)]:=DataList[PZ[i]-NConst].Data; end; end; Result := true; R
:=Stack[High(Stack)]; end; function SetData(Name:string; Data:real):boolean;
var i:integer; begin for i:=MConst to High(DataList) do if (UpperCase(Name)
= DataList[i].Name) then begin DataList[i].Data := Data; Result:=true; exit;
end; Result := false; end; function GetData(Name:string; var
Data:real):boolean; var i:integer; begin for i:=0 to High(DataList) do if
(UpperCase(Name) = DataList[i].Name) then begin Data := DataList[i].Data;
Result:=true; exit; end; Result := false; end; initialization
SetLength(DataList,MConst); DataList[0].Name:='PI'; DataList[0].Data:=Pi;
DataList[1].Name:='E'; DataList[1].Data:=2.71828183; ErrorList :=
TStringList.Create; finalization ErrorList.Free; end. unit UErrors; interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls; type TFEErrors = class(TForm) LBEErrors:
TListBox; private { Private declarations } public { Public declarations } end;
var FErrors: TFEErrors; implementation uses NN2; {$R *.dfm} end.

```