Міністерство освіти і науки України

Київський національний університет технологій та дизайну

Кафедра комп’ютерних наук

КОМП’ЮТЕРНА РЕАЛІЗАЦІЯ АЛГОРИТМІЧНИХ ТА ПРОГРАМНИХ КОМПОНЕНТІВ ПРИКЛАДНИХ ЗАДАЧ СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

ЩЕРБАНЬ В.Ю.
КОРОГОД Г.О.
ЧУПРИНКА Н.В.
ВОЛІВАЧ А.П.
КИРИЧЕНКО А.М.

КОМП’ЮТЕРНА РЕАЛІЗАЦІЯ АЛГОРИТМІЧНИХ ТА ПРОГРАМНИХ КОМПОНЕНТІВ ПРИКЛАДНИХ ЗАДАЧ СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЩЕРБАНЬ В.Ю., КОРОГОД Г.О., ЧУПРИНКА Н.В., ВОЛІВАЧ А.П., КИРИЧЕНКО А.М.

КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ АЛГОРИТМІЧНИХ ТА ПРОГРАМНИХ КОМПОНЕНТІВ ПРИКЛАДНИХ ЗАДАЧ СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

Київ – 2021

УДК 004.42
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 № 3 від 27 жовтня 2021року)

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 Київського національного університету технологій та дизайну;

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

ЧУПРИНКА Н.В. - кандидат технічних наук, доцент кафедри інформаційних та комп'ютерних технологій Київського національного університету технологій та дизайну;

ВОЛІВАЧ А.П. - кандидат технічних наук, старший викладач кафедри комп'ютерних наук Київського національного університету технологій та дизайну;

КИРИЧЕНКО А.М. - доктор філософії, асистент кафедри комп'ютерних наук Київського національного університету технологій та дизайну;

Рецензенти:

ОПАНАСЕНКО В.М. – д.т.н., професор, провідний науковий співробітник, Інститут кібернетики НАН України, лауреат Державної премії України в галузі науки і техніки;

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

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

Щ 610 Щербань В.Ю. Комп'ютерна реалізація алгоритмічних та програмних компонентів прикладних задач систем автоматизованого проектування / В.Ю. Щербань, Г.О. Корогод, Н.В. Чупринка, А.П. Волівач, А.М. Кириченко. – К.: Освіта України, 2021. – 645 с.

ISBN 978-617-8077-02-0

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

УДК 004.42
ББК 65.9(4Укр)306.4-6

ISBN 978-617-8077-02-0

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

ЗМІСТ

Вступ ……. 6

1. Алгоритми для розв'язання математичних моделей …… 15

1.1. Системи лінійних рівнянь …….. 15

1.2. Трансцендентні і алгебраїчні рівняння ……. 27

1.3. Диференціальні рівняння …… 43

1.4. Чисельне інтегрування …… 55

1.5. Інтерполяція, екстраполяція і апроксимація …….. 62

1.6. Математичні моделі, отримані шляхом обробки експериментальних даних …… 73

1.7. Питання оптимізації ……. 95

Висновки по розділу 1 ……. 115

Список використаних джерел до розділу 1 …… 118

2. Аналіз сучасних підходів по підвищенню точності вимірювання ……. 129

2.1.1. Аналіз літературних даних та постановка проблеми ……. 129

2.1.2. Суть методів надлишкових вимірювань …… 134

2.1.3. Узагальнені математичні моделі надлишкових методів вимірювань фізичних величин …….. 137

2.1.4. Узагальнені математичні моделі надлишкових методів вимірювань при багатократних вимірюваннях …… 151

2.2. розробка блок-схем алгоритмів роботи методу надлишкових вимірювань при різних видах функції перетворення сенсора …. 156

2.2.1. Побудова алгоритмів роботи методів надлишкових вимірювань ……. 156

2.2.2. Побудова алгоритмів роботи методів надлишкових вимірювань при лінійній функції перетворення сенсора …….. 158

2.2.3. Побудова алгоритмів роботи методів надлишкових вимірювань при функції перетворення сенсора, що описується поліномом 3-го ступеня …… 164

2.2.4. Побудова алгоритмів роботи методів надлишкових вимірювань при логарифмічній функції перетворення сенсора … 170

2.2.5. Побудова алгоритмів роботи методів надлишкових вимірювань при квадратичній функції перетворення сенсора …. 181
2.2.6. Побудова алгоритмів роботи методів надлишкових вимірювань при кубічній функції перетворення сенсора ……. 187
2.2.7. Побудова алгоритмів роботи багатократних надлишкових вимірювань ……. 189
2.3. Особливості застосування запропонованих підходів обробки результатів надлишкових вимірювань …… 197
2.3.1. Порівняльний аналіз похибок вимірювань МНВ і прямого метода при лінійній і поліноміальній функції перетворення ……. 197
2.3.2. Моделювання нелінійної функції перетворення при багатократних вимірюваннях …… 206
2.3.3. Дослідження логарифмічної функції перетворення по розширенню діапазону високоточних вимірювань …… 215
2.3.4. Порівняльний аналіз отриманих результатів …… 227

Висновки по розділу 2 ……. 230

Список використаних джерел до розділу 2 …… 235

3. Автоматизоване проектування деталей галантерейних виробів 241

3.1. Класифікація форм деталей галантерейних виробів …….. 241

3.2. Автоматизоване проектування декоративних елементів для галантерейних виробів ……. 253

3.3. Підготовка інформації про зовнішні контури деталей …… 286

3.4. Коригування зовнішніх контурів деталей …… 298

3.5. Програмне забезпечення для інтерактивного проектування деталей галантерейних виробів …… 322

Висновки по розділу 3 …….. 349

Список використаних джерел до розділу 3 …… 349

4. Застосування інформаційних технологій для оцінювання ризиків освітнього процесу закладів вищої освіти …… 359

4.1. Алгоритм оцінювання ризиків освітнього процесу …… 359

4.2. Застосування комп'ютерної програми «експерти зво» для формування експертних груп …….. 372
4.3. Регресійна модель оцінювання рівня якості освітньої програми ……. 386

4.4. Дослідження оцінювання рівня якості освітньої програми методом нечіткого кластерного аналізу ……. 407

Висновки до розділу 4 ……. 413

Список використаних джерел до розділу 4 …….. 414

5. Математичне моделювання процесу взаємодії нитки з направляючими круглов'язальних машин з урахуванням їх форми та конструктивних параметрів …….. 418

5.1. Геометричні, кінематичні та динамічні компоненти точок нитки при її взаємодії з напрямною великої кривизни …… 419

5.2. Взаємодія ниток з направляючими структурними елементами великої кривизни системи подачі нитки …….. 427

5.3. Визначення приведеного коефіцієнту тертя для структурних елементів системи подачі нитки круглов'язальних машин у формі тора ……. 433

5.4. Алгоритмічне та комп'ютерне програмне забезпечення для визначення натягу нитки після напрямної великої кривизни ….. 438

5.5. Алгоритмічне та комп'ютерне програмне забезпечення для визначення приведеного коефіцієнту тертя для структурних елементів системи подачі нитки ……. 445

Висновки по розділу 5 ……. 451

Список використаних джерел до розділу 5 …… 452

Додатки ……. 454

ВСТУП

За останні десятиріччя інформаційні технології зазнали такого глобального поширення, що зараз уже важко уявити життя сучасної людини без них. На сучасному етапі можна без особливих труднощів навести приклади використання інформаційних технологій у всі галузях: від освіти і до менеджменту. Сьогодні успіх буде мати та фірма, той заклад, який володіє найсучаснішими комп’ютерними технологіями. Значного прогресу можна досягти і в галузі освіти з впровадженням відповідних інформаційних компю’ютерних технологій, які зможуть зробити процес здобуття освіти більш гнучким, індивідуалізованим і одночасно нададуть змогу студентам використовувати глобальні ресурси для навчання, спілкуватись та обмінюватись досвідом.

Відповідно до визначення, прийнятого ЮНЕСКО, інформаційна технологія - це комплекс взаємозалежних, наукових, технологічних, інженерних дисциплін, що вивчають методи ефективної організації праці людей, зайнятих опрацюванням і збереженням інформації; обчислювальну техніку і методи організації і взаємодії з людьми і виробничим устаткуванням, практичні додатки, а також пов'язані з усім цим соціальні, економічні і культурні проблеми. Самі інформаційні технології вимагають складної підготовки, великих початкових витрат і наукомісткої техніки. Їхнє введення повинно починатися зі створення математичного забезпечення, формування інформаційних потоків у системах підготовки спеціалістів.

Такі технологічні поняття, що використовуються у виробничій сфері, як норма, норматив, технологічний процес, технологічна операція і т.п., можуть застосовуватися і в інформаційній технології. Перед тим, як розробляти ці поняття в будь-якій технології, у тому числі й в інформаційній, завжди варто починати з визначення мети. Потім варто

спробувати провести структурування всіх дій, що призводять до наміченої мети, і вибрати необхідний програмний інструментарій.

Необхідно розуміти, що освоєння інформаційної технології і подальше її використання повинні бути зведені до того, щоб було потрібно спочатку добре оволодіти набором елементарних операцій, кількість яких обмежена. З цієї обмеженої кількості елементарних операцій у різних комбінаціях складається дія, а з дій, також у різних комбінаціях, складаються операції, що визначають той або інший технологічний етап. Сукупність технологічних етапів утворить технологічний процес (технологію). Він може починатися з будь-якого рівня і не включати, наприклад, етапи або операції, а складатися тільки з дій. Для реалізації етапів технологічного процесу можуть використовуватися різні програмні ні програмні середовища.

Технічними засобами виробництва інформації будуть апаратне, програмне і математичне забезпечення цього процесу. З їхньою допомогою відбувається переробка первинної інформації в інформацію нової якості. Виділимо окремо з цих засобів програмні продукти і назвемо їх інструментарієм, а для більшої чіткості можна його конкретизувати, назвавши програмним інструментарієм інформаційної технології.

Інформаційна технологія опрацювання даних використовується для розв'язання добре структурованих задач, стосовно яких є необхідні вхідні дані і відомі алгоритми та інші стандартні процедури їх опрацювання. Ця технологія застосовується на рівні операційної (виконавчої) діяльності персоналу невисокої кваліфікації з метою автоматизації деяких рутинних постійно повторюваних операцій управлінської праці. Тому впровадження інформаційних технологій і систем на цьому рівні істотно підвищить продуктивність праці персоналу, звільнить його від рутинних операцій, можливо, навіть призведе до необхідності скорочення чисельності працівників. Багато даних на рівні операційної діяльності необхідно

зберігати для наступного використання або на цьому ж рівні, або на іншому. Для їхнього збереження створюються бази даних.

Метою інформаційної технології керування є задоволення інформаційних потреб усіх без винятку співробітників фірми, що мають справу з прийняттям рішень. Вона може бути корисна на будь-якому рівні керування. Ця технологія орієнтована на роботу в середовищі інформаційної системи керування і використовується при більш поганій структурованості розв'язуваних задач, якщо їх порівнювати з задачами, які розв'язуються за допомогою інформаційної технології опрацювання даних. Інформаційна технологія керування ідеально підходять для задоволення подібних інформаційних потреб працівників різноманітних функціональних підсистем (підрозділів) або рівнів керування фірмою. Інформація, що поставляється нею, містить відомості про минуле, дійсне і ймовірне майбутнє фірми. Ця інформація має вигляд регулярних або спеціальних управлінських звітів.

Ефективність і гнучкість інформаційної технології багато в чому залежать від характеристик інтерфейсу системи підтримки прийняття рішень. Інтерфейс визначають: мова користувача; мова повідомлень комп'ютера, що організує діалог на екрані дисплея; знання користувача. Мова користувача - це ті дії, які користувач робить по відношенню до системи шляхом використання можливостей клавіатури; електронних олівців, що пишуть на екрані; джойстика; "миші"; голосових команд, тощо. "миші"; голосових команд, тощо. Найбільш простою формою мови користувача є створення форм вхідних і вихідних документів. Отримавши вхідну форму (документ), користувач заповнює його необхідними даними і вводить у комп'ютер. Система підтримки прийняття рішень робить необхідний аналіз і видає результати у вигляді вихідного документа заданої форми. Мова повідомлень - це те, що користувач бачить на екрані дисплея (символи, графіка, колір), дані, надруковані принтером, звукові

вихідні сигнали і т.п. Важливим показником ефективності інтерфейсу, який використовується, є обрана форма діалогу між користувачем і системою. В даний час найбільш поширеним є такі форми діалогу: режим “запитання-відповідь”, командний режим, режим меню, режим заповнення пропусків у виразах, запропонованих комп'ютером. Кожна форма в залежності від типу задачі, індивідуальності користувача і рішення, яке приймається, може мати свої переваги і недоліки. Довгий час єдиною реалізацією мови повідомлень був надрукований або виведений на екран дисплея звіт або повідомлення. Тепер з'явилася нова можливість представлення вихідних даних - машинна графіка. Вона дає можливість створювати на екрані і папері кольорові графічні зображення в тривимірному вигляді. Використання машинної графіки, яка значно підвищує наочність і інтерпретованість вихідних даних, стає усе більш популярним в інформаційній технології підтримки прийняття рішень.

Для інформаційних технологій є цілком природним те, що вони застарівають і заміняються новими. Так, наприклад, на зміну технології пакетного опрацювання програм на великий ЕОМ в обчислювальному центрі прийшла технологія роботи на персональному комп'ютері на робочому місці користувача. Телеграф передав усі свої функції телефону. Телефон поступово витісняється службою експрес-доставки. Телекс передав більшість своїх функцій факсу й електронній пошті.

При впровадженні нової інформаційної технології в організації необхідно оцінити ризик відставання від конкурентів у результаті її неминучого старіння, тому що інформаційні продукти, як ніякі інші види матеріальних товарів, мають надзвичайно високу швидкість змінюваності новими видами або версіями. Періоди змінюваності коливаються від декількох місяців до одного року. Якщо в процесі впровадження нової інформаційної технології цьому фактору не приділяти належної уваги, цілком можливо, що до моменту завершення переходу фірми на нову

інформаційну технологію вона вже застаріє і прийдеться вживати заходів щодо її модернізації. Такі невдачі з впровадженням інформаційних технологій звичайно пов'язані з недосконалістю технічних засобів, в той час як основною причиною невдач є відсутність або слабка пропрацьованість методології використання інформаційної технології.

Централізоване опрацювання інформації на ЕОМ обчислювальних центрів були першою історично сформованою технологією. Створювалися великі обчислювальні центри колективного користування, оснащені великими ЕОМ. Застосування таких ЕОМ дозволяло опрацьовувати великі масиви вхідної інформації й одержати на цій основі різноманітні види інформаційної анітні види інформаційної продукції, яка потім передавалася користувачам. Такий технологічний процес був обумовлений недостатнім оснащенням обчислювальною технікою підприємств і організацій у 60 - 70-і рр.

Складне програмне забезпечення має недоліки, якими можуть скористатися сторонні особи (хакери) і використати їх на свою користь. Для попередження несанкціонованого доступу використовуються дуже дорогі системи захисту, а також вдосконалюється програмне забезпечення. При використанні програмного забезпечення існує можливість втрати інформації, спричинена дією вірусів, які використовують його недоліки. У зв’язку з тим, що вартість інформації росте, втрати можуть бути суттєвими. Для захисту доводиться використовувати спеціальні програми – антивіруси. Беручи до уваги те, що зараз відбувається концентрація у сфері інформаційних технологій, перед користувачем постає дилема вибору платформи інформаційної технології, так як в майбутньому він буде залежати від свого постачальника програмного забезпечення. Легкість тиражування інформаційних продуктів надає змогу з легкістю порушувати авторські права. Це стосується, в першу чергу, програмного забезпечення.

Одним із засобів керування розвитком інтелекту і підвищення його організованості на сучасному етапі є інформатизація суспільства, що ґрунтується насамперед на розвитку інформаційних комп’ютерних технологій. Значення інформаційної технології величезне - вона формує передній край науково-технічного прогресу, створює інформаційний фундамент розвитку науки і всіх інших технологій. Головними, визначальними стимулами розвитку інформаційної технології, є соціально-економічні потреби суспільства, і саме зараз суспільство як ніколи зацікавлене в якомога швидшій інформатизації та комп'ютеризації всіх без винятку сфер діяльності. Дуже важливою властивістю інформаційної технології є те, що для неї інформація є не тільки продуктом, але і вихідною сировиною. Особлива роль приділяється всьому комплексу інформаційної технології і техніки в структурній перебудові економіки убік наукоємності. Більш того, інформаційна технологія є свого роду перетворювачем всіх інших галузей господарства, як виробничих, так і невиробничих, основним засобом їхньої автоматизації, якісної зміни продукції і, як наслідок, їх переходу частково або цілком у категорію наукомістких. Пов'язаний з цим і працезаощаджувальний характер інформаційної технології, що реалізується, зокрема, у керуванні багатьма видами робіт і технологічних операцій. Безсумнівною перевагою інформаційної технології є те, що вона сама створює засоби для своєї еволюції. Формування системи, що само розвивається - найважливіший підсумок, досягнутий у сфері інформаційної технології. Таким чином, усі вищевикладені риси інформаційної технології вказують на те, що вона й у майбутньому залишиться самим перспективним видом технології, що допомагає людині впевнено крокувати шляхом прогресу.

У сучасних умовах конкурентоспроможності продукції вітчизняна промисловість потребує підвищення якості товарів при одночасному зниженні витрат на виробництво. Це стимулює впровадження нових

технологій і підходів, що забезпечують підвищення точності вимірювання та/або отримання достовірної інформації в цілому і його параметрів зокрема. Крім того, для успішного проведення будь-якого технологічного процесу, як відомо, необхідно точне дотримання умов його проведення. Навіть незначне відхилення від встановлених правил і норм може призвести до недостовірності отриманого результату вимірювання, до браку продукції, а в деяких випадках навіть до аварії. Таким чином, точність інформації, яка отримана в результаті вимірювань, визначає ефективність всієї роботи технологічного процесу.

При вирішенні питання точності вимірювання спочатку треба встановити причини, що призводять до її зниження. У вирішенні питання щодо підвищення точності вимірювань особливу роль відіграє точність сенсора і вдосконалення відповідних методів вимірювання. Це обумовлено самою фізичною суттю первинного датчика (сенсора), який перетворює вхідну контрольовану фізичну величину, що характеризує технологічний процес або властивість досліджуваного об’єкта, в пропорційний електричний вихідний сигнал. Тож, від того, з якою точністю буде проведено це перетворення і залежить точність всього подальшого вимірювального процесу. Це обумовлено тим, що навіть незначне відхилення, що вноситься сенсором, буде лише підсилене при подальшій обробці в вимірюваному каналі. У зв’язку з тим, що сенсор є первинним перетворювачем вхідного сигналу, то до них (сенсорів) висуваються високі вимоги по точності, по чутливості, по ширині робочого діапазону, по стабільності вхідних характеристик в часі (при зміні навколишнього середовища), по надійність тощо. Слід також зауважити, що нелінійність функції перетворення вимірювального каналу з сенсором також є перепоною до підвищення точності. Це пов’язане з тим, що при нелінійній функції необхідно проводити її лінеаризацію з відповідною похибкою або

звужувати робочий діапазон і працювати на кусочно-лінійних ділянках вхідної характеристики.

Особливу увагу слід також приділити і промисловим вимірюванням, де потрібно проводити багатократні вимірювання протягом довгого часу. Це обумовлено тим, що при багатократних вимірювань крім систематичної складової похибки на кінцевий результат має вплив також і випадкова складова похибки вимірювань. Крім того, через вплив середовища та старіння конструктивних елементів, параметри функції перетворення змінюються. Це призводить до потреби частої повірки засобу вимірювання для забезпечення заданих метрологічних характеристик, що спричиняє матеріальні та часові втрати.

Автоматизована підготовка виробництва забезпечує прискорення створення нових моделей, скорочення трудовитрат на рутинну роботу, максимальна відповідність потребам людини. САПР стає невід'ємною складовою у легкій промисловості.

Інтегрування систем і створення єдиної бази даних, яка дає можливість автоматизації тих чи інших виробничих процесів з системами управління підприємствами дає можливість домогтися відчутного зниження вартості продукції, що забезпечує успішну діяльність на ринках збуту.

Отже застосування сучасних САПР на українських підприємствах дозволяє автоматизувати виготовлення виробів легкої промисловості і виготовляти конкурентоспроможну продукцію яка буде задовольняти сучасним вітчизняним та світовим вимогам.

З автоматизацією проектування пов’язані принципові можливості створення найскладніших технічних об’єктів у різних галузях промисловості, зокрема в легкій, адже проектування будь-яких об’єктів потребує великих затрат людських і часових ресурсів. Від термінів та

якості проектування значно залежать строки впровадження та якість готової продукції

В даний час накопичений значний досвід щодо автоматизації підготовки виробництва взуття та одягу, але задача автоматизованого проектування виробництва шкіргалантерейних виробів залишається ще невирішеною.

З метою розширення асортименту і поліпшення якості готової продукції, до виробництва виробів шкіргалантереї необхідно висувати усе більш високі вимоги. Конструкції моделей шкіргалантерейних виробів і технологічний процес їх складання повинні сприяти максимальній механізації та автоматизації процесів, росту продуктивності праці та зниженню собівартості продукції. Тому удосконалення конструкторської підготовки виробництва і, зокрема, створення високоефективного методу автоматизованого проектування деталей виробів шкіргалантереї різних конструкцій є актуальною задачею.

Створення методу автоматизованого проектування деталей галантерейних виробів з використанням комп’ютерних технологій дозволить істотно зменшити терміни виконання конструкторських робіт і підвищити продуктивність праці, поліпшити якість проектних рішень і, у кінцевому підсумку, підвищити ефективність роботи підприємств.

1. АЛГОРИТМИ ДЛЯ РОЗВ’ЯЗАННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ

1.1. 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]

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n = b_1, \\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k + \ldots + a_{2n}x_n = b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mk}x_k + \ldots + a_{mn}x_n = b_m, \end{aligned} \qquad (1.1.1)$$

де a_{11} … a_{nn} – коефіцієнти при невідомих величинах; x_1 … x_n – невідомі (при вирішенні завдань статики і динаміки це можуть бути реакції зв'язків, узагальнені координати, швидкості і прискорення); b_1 … b_m – постійні величини(необхідно відзначити, що враховуючи фізичну суть досліджуваних статичних і динамічних процесів розмірність парних творів в лівій частині рівності системи (1.1.1) повинна співпадати з розмірністю постійних b_1 … b_m); n – число невідомих, які входять в систему; m – число рівнянь.

Систему рівнянь (1.1.1) зручно представити у векторній формі

$$A\bar{x} = \bar{b}, \qquad (1.1.2)$$

де A – матриця коефіцієнтів m х 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 (1.1.1) можна представити у вигляді (матриця коефіцієнтів перетвориться до квадратного вигляду)

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k + \ldots + a_{2n}x_n &= b_2, \\ \ldots\ldots\ldots\

системи рівнянь. Для визначення невідомих ***x1 ..... xn*** необхідно виконати ***n*** перестановок

$$x_1 = \frac{\begin{Vmatrix} b_1 & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ b_n & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{Vmatrix}}{\begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{Vmatrix}},$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$x_k = \frac{\begin{Vmatrix} a_{11} & \dots & a_{1k-1} & b_1 & a_{1k+1} & \dots & a_{1n} \\ a_{21} & \dots & a_{2k-1} & b_2 & a_{2k+1} & \dots & a_{2n} \\ a_{n1} & \dots & a_{nk-1} & b_n & a_{nk+1} & \dots & a_{nn} \end{Vmatrix}}{\begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{Vmatrix}}, \qquad (1.1.5)$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$x_n = \frac{\begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & b_1 \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & b_2 \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & b_n \end{Vmatrix}}{\begin{Vmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2n} \\ a_{n1} & a_{n2} & \dots & a_{nk} & \dots & a_{nn} \end{Vmatrix}}.$$

Даний метод з обчислювальної точки зору є не ефективним. Для визначення всього коріння ***x1 ... xn*** необхідно виконати з використанням системи (1.1.5) $(n^2-1)\cdot(n!)$ множень і ***n*** ділень. Для системи складається з 50 рівнянь необхідно буде виконати $\boldsymbol{7{,}610^{67}}$ 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 (1.1.3) 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.

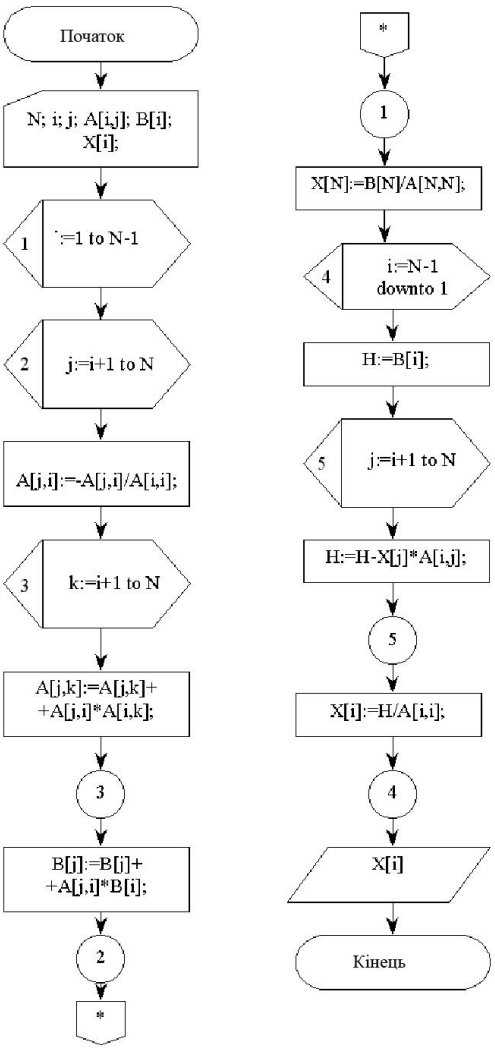
Вибір того або іншого методу залежить від характеристик ЕОМ, числа рівнянь системи, характеристик матриці коефіцієнтів ***A***. Наприклад, при складанні рівнянь рівноваги твердого тіла деякі коефіцієнти ***aij*** дорівнюють нулю. У таких випадках використовувати метод Гауса послідовного виключення невідомих і метод Гауса з вибором головного елементу 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 (1.1.3) за рахунок виключення невідомих ***x1 ... xn-1*** в лінійних рівняннях. Це приводить до отримання матриці коефіцієнтів ***aij*** трикутного вигляду. Ця послідовність процедур характеризується як прямий хід виключення змінних. Для його здійснення необхідно послідовно віднімати перше рівняння з подальших, умножаючи його ліву і праву частини на постійні коефіцієнти, що представляють приватне від ділення постійних коефіцієнтів $m_{ji}=a_{ji}/a_{ii}$, де $i = 1,2,\ldots n-1; j= i + 1, i +2,\ldots n$. Кінцева система рівнянь на прямому ході має вигляд

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \ldots + a_{2k}^{(1)}x_k + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ \ldots\ldots\ldots\

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

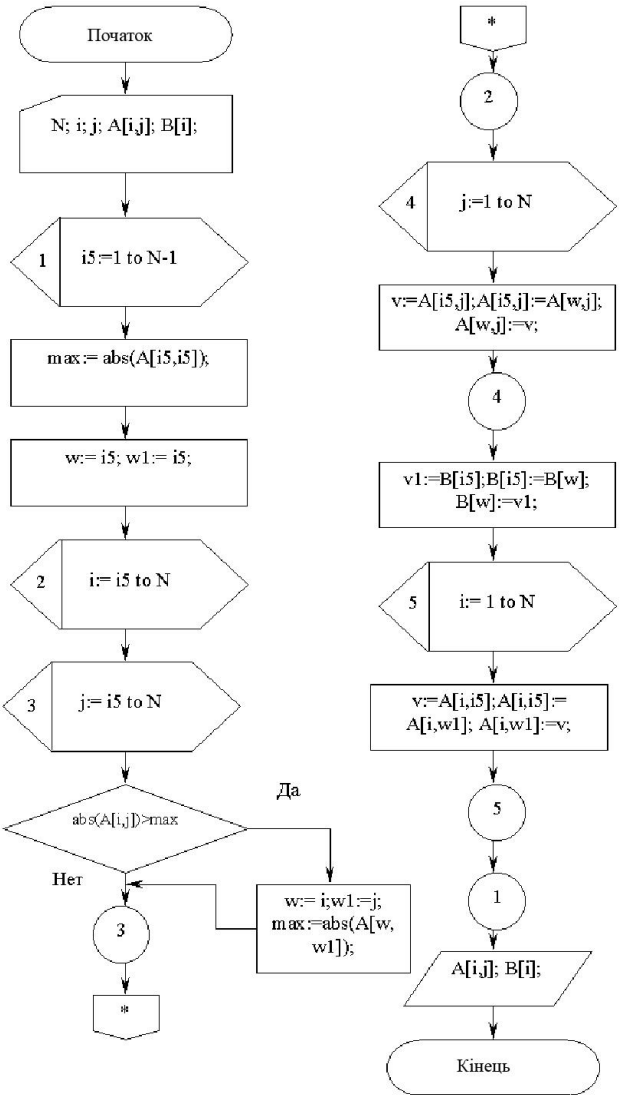
Якщо ми при обчисленнях обмежимося не 4, а 11 значущими цифрами після коми, то тоді останній результат матиме вигляд

$x_1 = 0{,}00000000030$; $x_2 = -2{,}99999999880$; $x_3 = -3{,}99999999820$.

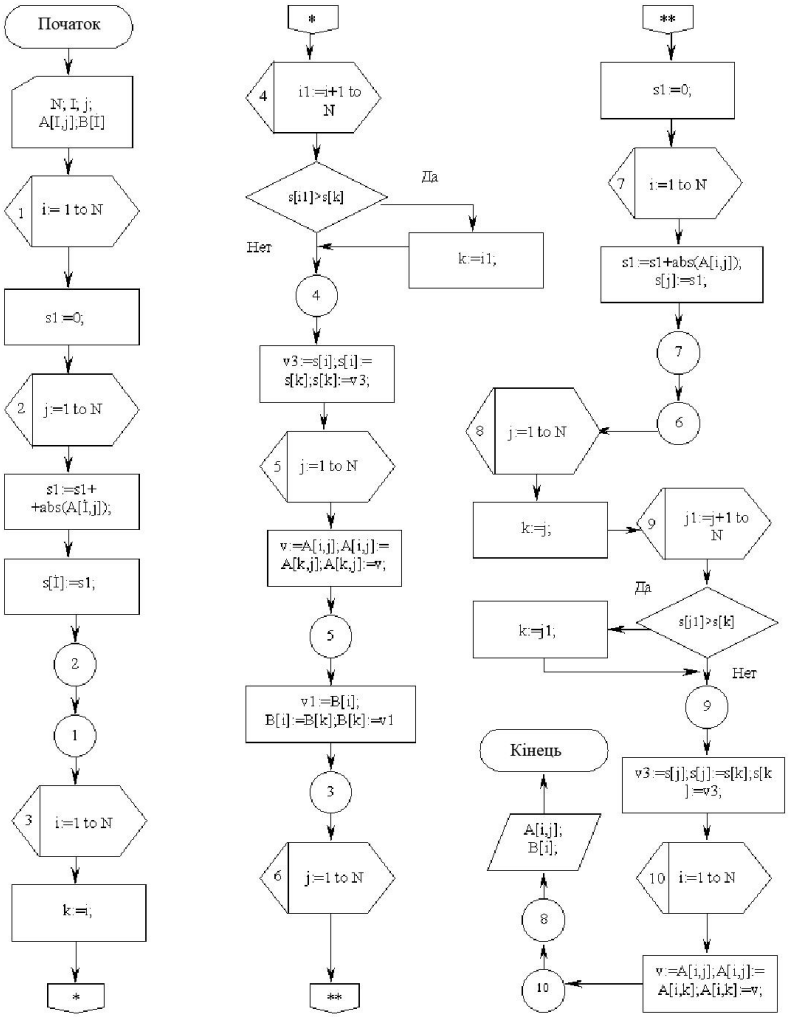
При такому результаті після підстановки в (1.1.7) всі три рівняння звертаються в тотожність.

Така втрата точності пояснюється дуже маленьким значенням коефіцієнта a_{11} , що приводить до різкого зростання коефіцієнтів $a^{(1)}{}_{22} \ldots a^{(1)}{}_{2n}$; $b^{(1)}{}_2$; $\ldots a^{(1)}{}_{n2} \ldots a^{(1)}{}_{nn}$; $b^{(1)}{}_n$ на першому кроці прямого ходу. Для виключення цього використовуються різні схеми перетворення матриць коефіцієнтів на початковому етапі перед визначенням коріння. На рис.1.1.2 і 1.1.3 представлені блок-схеми програм, що реалізовують два методи подібного перетворення. Перший з них полягає в перетворенні матриці коефіцієнтів на основі вибору головного (ведучого) елементу, 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 $max = abs\ (a_{ij})$, 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 (1.1.3) 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 елементу. Програма ***PROGRAM S_G_VGE*** визначення коріння системи лінійних

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



Блок-схема PROGRAM PRE_MATR_VGE



Блок-схема PROGRAM PRE_MAT_MAX_SUM

Інший метод полягає в наступному (см.рис.1.1.3): 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 убування 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 убування 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 (1.1.3) до системи лінійних рівнянь з правою трикутною матрицею на прямому ходу по наступних формулах

$$M_{ki}\left(\sum_{j=1}^{n} a_{ij}x_j - b_i\right) - L_{ki}\left(\sum_{j=1}^{n} a_{kj}x_j - b_k\right) = 0,$$
$$L_{ki}\left(\sum_{j=1}^{n} a_{ij}x_j - b_i\right) - M_{ki}\left(\sum_{j=1}^{n} a_{kj}x_j - b_k\right) = 0, \qquad (1.1.8)$$
$$M_{ki} = \frac{a_{ii}}{\sqrt{a_{ii}^2 + a_{ki}^2}}, \quad L_{ki} = -\frac{a_{ki}}{\sqrt{a_{ii}^2 + a_{ki}^2}}.$$

де ***i=1,2,…n-1; k=i+1,… 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

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}}\left(\sum_{k=1}^{n} a_{ik}x_{k(j)} - b_i\right). \qquad (1.1.9)$$

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

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

$$C = \max_{1\le i\le n}(C_i) < 1, \tag{1.1.10}$$

де $C_i = \left(\sum_{j=1}^{n}\left|\frac{a_{ij}}{a_{ii}}\right| - 1\right)$, $i=1,2,\ldots n$, то система лінійних рівнянь (1.1.3) має єдине рішення і ітераційний процес сходиться до рішення із швидкістю геометричної прогресії. Умову (1.1.10) можна інтерпретувати і так, ітераційний процес сходиться, якщо величина модуля кожного діагонального елементу 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

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}}\left(\sum_{k=1}^{i-1} a_{ik}x_{k(j+1)} + \sum_{k=i}^{n} a_{ik}x_{k(j)} - b_i\right). \tag{1.1.11}$$

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

Достатньою умовою збіжності методу Зейделя є наступний вираз

$$\sum_{\substack{j=1\\ j\ne i}}^{n}\left|\frac{a_{ij}}{a_{ii}}\right| \le q, \quad i = 1\ldots n, \quad q < 1. \tag{1.1.12}$$

Зазвичай метод Зейделя сходиться швидше, ніж метод простих ітерацій.

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

$$\mu = \frac{sup\left[\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2}\right]}{inf\left[\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2}\right]} \geq 1. \qquad (1.1.13)$$

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

Розглянемо простій приклад. Хай дана система двох рівнянь

$$2x_1 + 0x_2 = 2,$$
$$5x_1 + 0{,}05x_2 = 5.$$

Вирішення даної системи рівнянь має вид $x1=1,\ x2=0.$ Трохи змінимо праву частину другого рівняння, тоді

$$2x_1 + 0x_2 = 2,$$
$$5x_1 + 0{,}05x_2 = 5{,}05.$$

Вирішення нової системи рівнянь матиме вид $x1=1,\ x2=1.$ Незначна зміна правій частині другого рівняння привела до істотної зміни рішення. Визначимо по (1.1.13) обумовленість матриці коефіцієнтів останньої системи

$$\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2} = \sqrt{29x_1^2 + 0{,}5x_1x_2 + 2{,}5\cdot 10^{-3}x_2^2}. \qquad (1.1.14)$$

Знайдемо найбільше і найменше значення функції (1.1.14) на одиничному колі із заміною $x1=\cos\alpha,\ x2=\sin\alpha$. Тоді

$$\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2} = \sqrt{29\cos^2\alpha + 0{,}5\cos\alpha\sin\alpha + 2{,}5\cdot 10^{-3}\sin^2\alpha}. \qquad (1.1.15)$$

Пошук безумовного екстремуму функції (1.1.15) однієї змінної дозволив визначити

$$sup\left[\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n}a_{ij}x_j\right)^2}\right]=5,385,$$
$$inf\left[\sqrt{\sum_{i=1}^{n}\left(\sum_{j=1}^{n}a_{ij}x_j\right)^2}\right]=0,0186\ . \qquad (1.1.16)$$

Підставляємо результати (1.1.16) в (1.1.13) 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.***

1.2. 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)

$$ax^2+bx+c=0, \qquad (1.2.1)$$

де ***a, b, c*** – коефіцієнти рівняння (1.2.1); ***x*** – невідоме; визначаються по формулах

$$x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}\ ,\ x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}. \qquad (1.2.2)$$

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

$$x_1+x_2=-\frac{b}{a},\ x_1x_2=\frac{c}{a}, \qquad (1.2.3)$$

зв'язують між собою коріння і коефіцієнти квадратного рівняння (1.2.1). Теорема Вієта, для позитивного коріння, встановлює справедливість співвідношень між ними і коефіцієнтами многочлена [14]

$$P(x) = x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0, \tag{1.2.4}$$

у вигляді

$$\begin{gathered} a_0 = (-1)^n x_1 x_2 ... x_n, \\ a_1 = (-1)^{n-1}(x_1 x_2 ... x_{n-1} + x_1 ... x_{n-2} x_n + ... + x_2 x_3 ... x_n), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n-2} = x_1 x_2 + x_1 x_3 + ... + x_{n-1} x_n, \\ a_{n-1} = -(x_1 + x_2 + ... + x_n). \end{gathered} \tag{1.2.5}$$

Перші вирішення квадратних рівнянь (1.2.1) з позитивними коефіцієнтами (для позитивного коріння) виявлені ще в працях Мухаммеда аль Хорезмі і Діофанту. У *XVI* столітті М.Штіфель запропонував узагальнений метод рішення (1.2.1) включаючи і негативне коріння. Остаточне рішення, з урахуванням випадку уявного коріння, запропоноване Р.Бомбеллі і С.Стевіном [14].

У рівнянні (1.1.2) n називають ступенем рівняння. Вирішення рівняння $a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0 = 0$ 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

$$ax^3 + bx^2 + cx + d = 0, \quad a \neq 0, \tag{1.2.6}$$

приводимо до канонічного вигляду

$$y^3 + py + q = 0, \tag{1.2.7}$$

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 (1.2.7) визначається по формулах Дж. Кардано

$$y_1 = A + B,\quad y_{2,3} = -\frac{A+B}{2} \pm i\frac{A-B}{2}\sqrt{3},$$
$$A = \sqrt[3]{-\frac{q}{2} + \sqrt{Q}},\quad B = \sqrt[3]{-\frac{q}{2} - \sqrt{Q}}, \qquad (1.2.8)$$
$$Q = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2.$$

У якості A і $У$ 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

$$ax^4 + bx^3 + cx^2 + dx + e = 0, \qquad (1.2.9)$$

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

$$y^4 + py^2 + qy + r = 0. \qquad (1.2.10)$$

Коріння рівняння (1.2.10) визначається по формулах

$$y_1 = \frac{1}{2}\left(\sqrt{z_1} + \sqrt{z_2} + \sqrt{z_3}\right),\ y_2 = \frac{1}{2}\left(\sqrt{z_1} - \sqrt{z_2} - \sqrt{z_3}\right),$$
$$y_3 = \frac{1}{2}\left(-\sqrt{z_1} + \sqrt{z_2} - \sqrt{z_3}\right),\ y_4 = \frac{1}{2}\left(-\sqrt{z_1} - \sqrt{z_2} + \sqrt{z_3}\right). \qquad (1.2.11)$$

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 (1.2.11) 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 відрізання 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.1.2.1а представлені графічні залежності двох функцій

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

трансцендентного рівняння

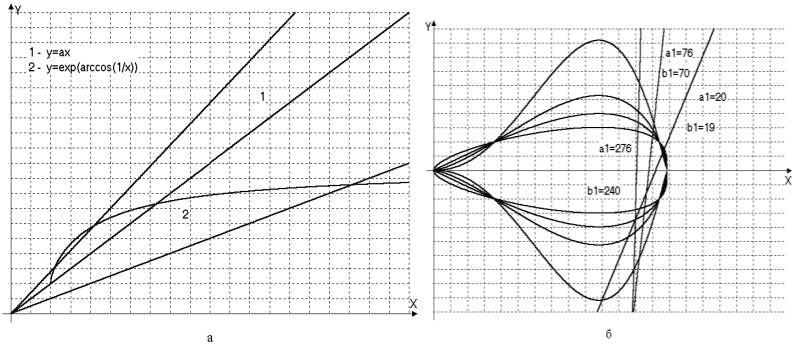
$$ax = e^{\arccos\left(\frac{1}{x}\right)}. \qquad (1.2.12)$$

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 (1.2.12). Причому, на вибраному інтервалі зміни x, для різних значень коефіцієнтів a, може бути від 1 до 2 коріння.

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

$$\begin{aligned} bx^2 - x^4 - y^a &= 0, \\ a_1 x - b_1 - y &= 0. \end{aligned} \qquad (1.2.13)$$

На рис.1.2.1б представлено графічне (наближене) визначення коріння системи (1.2.13) для різних значень a, b, a_1, b_1. Їх величина залежить від ряду технологічних параметрів і конструктивних особливостей ткацького верстата [17].



Ріс.1.2.1. 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 відрізку $[a,b]$ 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

$$f(x)=0, \qquad (1.2.14)$$

представимо до вигляду

$$x = f_1(x), \tag{1.2.15}$$

де $f_1(x) = f(x) + x.$

Скориставшись графічним способом, визначимо наближене значення кореня x_O з області визначення функції $f_1(x)$ і підставимо його в праву частину рівняння (1.2.15). Будуватимемо послідовність чисел $\{x_n\}$, визначених за допомогою ітераційної формули

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

Послідовність чисел $\{x_n\}$ носить назву ітераційної послідовності. Якщо існує $\lim\limits_{n\to\infty} x_n = \xi$, то переходячи в рівності (1.2.15) до межі і рахуючи функцію $f_1(x)$ безперервною, матимемо

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

або

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

З останньої рівності виходить, що ξ буде коренем трансцендентного рівняння (1.2.14). Ітераційний процес продовжується до тих пір, поки дотримується умова

$$|x_{n+1} - x_n| \geq \varepsilon, \tag{1.2.16}$$

де ε - задана погрішність обчислення кореня ξ.

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

$$\left|\frac{f(x_1) - f(x_2)}{x_1 - x_2}\right| \leq q. \tag{1.2.17}$$

Якщо функція (1.2.14) задовольняє умові (1.2.17), то вона є безперервною на відрізку $[a,b]$. Повідомимо аргументу x приріст Δx. Для початкового наближення визначимо приріст функції (1.2.14)

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

звідки, з використанням умови Ліпшиця, отримаємо підтвердження безперервності функції (1.2.14) на відрізку $[a,b]$

$$|\Delta f| \le \alpha |\Delta x|,$$
$$\lim_{\Delta x \to 0} \Delta f = 0.$$

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

Доведемо це твердження. Візьмемо початкове наближення вирішення трансцендентного рівняння (1.2.14) x_O на відрізку $[\xi - \delta, \xi + \delta]$, яке розташоване від точки ξ на відстані не більше ніж δ ($|\xi - x_O| \le \delta$). Виконаємо ітераційний процес з використанням умови Ліпшиця і з обліком (1.2.15)

$$\begin{aligned} x_1 = f_1(x_O), \;\; x_1 - \xi = f_1(x_O) - f_1(\xi), \\ |x_1 - \xi| = |f_1(x_O) - f_1(\xi)| \le q|x_O - \xi| \le q\delta, \\ |x_2 - \xi| = |f_1(x_2) - f_1(\xi)| \le q|x_1 - \xi| \le q^2|x_O - \xi| \le q^2\delta, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ |x_n - \xi| \le q^n |x_O - \xi| \le q^n \delta. \end{aligned} \qquad (1.2.18)$$

З (1.2.18) витікає, що

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

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

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

Коли $q\langle\frac{1}{2}$ то приходимо до (1.2.16). Остаточно

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

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

Перейдемо до визначення коріння трансцендентного рівняння (1.2.14) з використанням методу Ньютона (або метод дотичних). На рис.1.2.2 показана графічна схема, що реалізовує метод Ньютона.

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

$$f_K = f(x_O) + f'(x_O)(x - x_O)\ , \qquad (1.2.19)$$

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

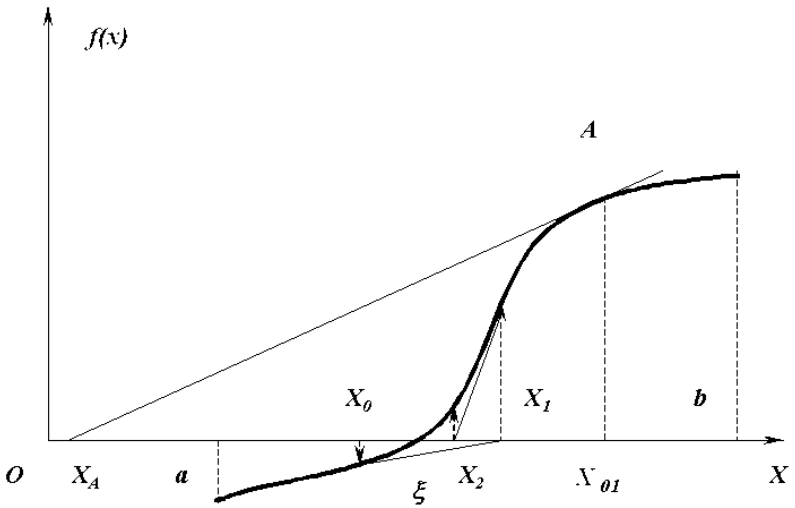


Рис.1.2.2. Вирішення трансцендентного рівняння методом Ньютона (дотичних)

При $x = x_1$, $f_K = 0$. Тоді з (1.2.19) отримаємо

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

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

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0,1,2... \tag{1.2.20}$$

Метод Ньютона, що реалізовується при допомозі (1.2.20), володіє високою швидкістю збіжності, проте він дуже чутливий до вибору початкового наближення x_0. На рис.1.2.2 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 відрізку $[a,b]$. Це обриває процес побудови ітераційної послідовності $\{x_n\}$. Даний метод забезпечує швидку (квадратичну) збіжність якщо

$$f(x_0)f''(x_0)\rangle 0.$$

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

Коли $|f'(x)| \geq m\rangle 0,\ |f''(x)| \leq M,\ x \in [a,b]$ (m – найменше значення похідної $f'(x)$ в $[a,b]$; M – найбільше значення похідної $f''(x)$ в $[a,b]$), то знайдеться таке $\delta : 0\langle \delta \leq min(\xi - a, b - \xi)$, що при будь-якому виборі початкового наближення на відрізку $[\xi - \delta, \xi + \xi] \in [a,b]$ існує нескінченна ітераційна послідовність (1.2.20) і ця послідовність сходиться до кореню ξ трансцендентного рівняння (1.2.14). Дане твердження формалізує теорему про збіжність методу Ньютона. Її доказ побудований з використанням основного рекурентного співвідношення методу простих ітерацій (1.2.15)

$$x = f_1(x), \quad f_1(x) = x - \frac{f(x)}{f'(x)},$$
$$f_1'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}, \quad \lim_{x\to\xi} f(x) = f(\xi) = 0, \tag{1.2.21}$$
$$|f_1'(x)| \le \frac{M}{m^2}|f(x)|, |f(x) - f(\xi)| = |f(x)| \le \frac{m^2}{2M},$$
$$|f_1'(x)| \le q = 0{,}5.$$

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

Для оцінки погрішності n – го наближення x_n можна використовувати формулу

$$|\xi - x_n| \le \frac{|f(x_n)|}{m_1}, \tag{1.2.22}$$

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

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

$$\frac{df(x)}{dx} \cong \frac{f(x_n + \Delta x) - f(x_n)}{\Delta x} = \frac{\Delta f(x_n)}{\Delta x}, \tag{1.2.23}$$
$$\Delta x = \varepsilon.$$

Тоді з обліком (1.2.23) рівняння (1.2.20) прийме вигляд

$$x_{n+1} = x_n - \frac{\Delta x f(x_n)}{f(x_n + \Delta x) - f(x_n)}. \tag{1.2.24}$$

Рекурентну формулу (1.2.24) 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 відрізку $[a,b]$ 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

$$a_n \le \xi, \quad b_n \ge \xi, \quad n = 1,2...$$

де ξ - корінь трансцендентного рівняння (1.2.14) на відрізку $[a,b]$.

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

Вважатимемо, що $f(a)f(b)\langle 0$ и $f(a)\langle 0,\ f(b)\rangle 0$. Тоді візьмемо середню точку відрізку $[a,b]$

$$\xi_1 = \frac{a+b}{2}. \qquad (1.2.25)$$

Обчислюємо в даній крапці значення функції $f(x)$. Припустимо, що $f(\xi_1)=0$. В цьому випадку процес визначення кореня трансцендентного рівняння закінчується. Якщо $f(\xi_1)\ne 0$, то розглядаємо два відрізка $[a,\xi_1]$ и $[\xi_1,b]$. Вибираємо той з них, де виконується умова $F9a) f(\xi_1)\langle 0$ або $f(\xi_1)f(b)\langle 0$. Вибраний відрізок знову поділяємо навпіл приймаючи

$a_1 = a, \ b_1 = \xi_1$ або $a_1 = \xi_1, \ b_1 = b$

$$\xi_2 = \frac{a_1+b_1}{2}. \qquad (1.2.26)$$

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

$$a_n \le a_{n+1}\langle b_{n+1} \le b_n. \qquad (1.2.27)$$

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

$$\lim_{n\to\infty} a_n = z_1, \qquad (1.2.28)$$

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

$$\lim_{n\to\infty} b_n = z_2. \tag{1.2.29}$$

Очевидно, що

$$\begin{aligned} &a_n \le z_1 \le z_2 \le b_n, \\ &z_2 - z_1 \le b_n - a_n = \frac{b-a}{2^n}. \end{aligned} \tag{1.2.30}$$

З останнього рівняння системи (1.2.30) отримуємо $z_2 - z_1 = 0$, т.ч. дана різниця менше будь-якого наперед заданого позитивного числа. Звідки матимемо $z_2 = z_1 = \xi$. Отже

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

З обліком (1.2.30)

$$a_n \le \xi \le b_n.$$

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.

Представляє інтерес метод порозрядного наближення для обчислення всього існуючого коріння трансцендентного рівняння на відрізку $[a,b]$ із заданою погрішністю ε. На рис.1.2.3 представлена блок-схема алгоритму для реалізації вказаного методу.

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

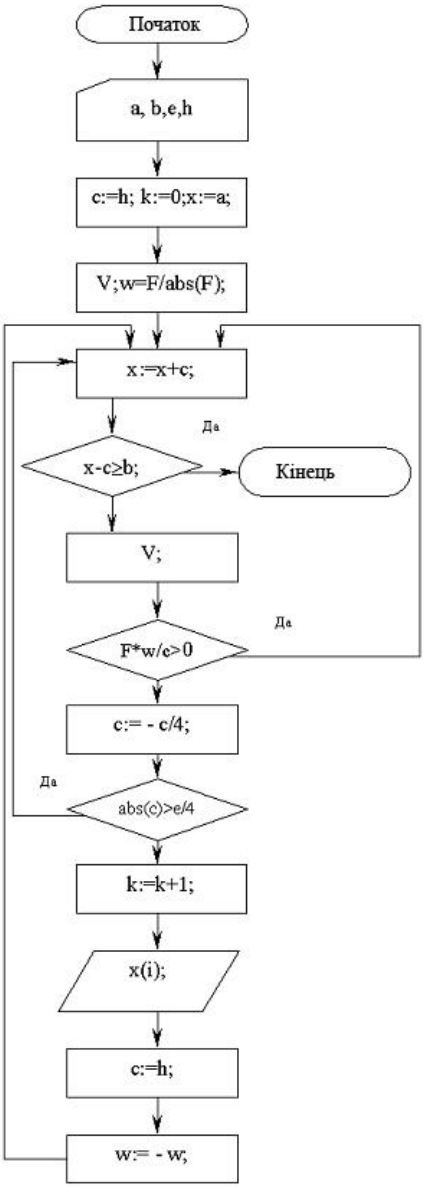


Рис.1.2.3. Блок-схема алгоритму для реалізації методу порозрядного наближення

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 (1.2.14) і перевіряємо умову

$$\frac{f(x)sign[f(x)]}{C} \rangle 0. \qquad (1.2.31)$$

На наступному етапі зменшуємо крок $C = -\frac{C}{R}$ і перевіряємо виконання умови $|C| \rangle \frac{\varepsilon}{R}$, 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

$$f_i(x_1, \ldots x_i) = 0, \quad i = 1,2 \ldots n, \qquad (1.2.32)$$

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-різницевими значеннями

$$\frac{\partial f_i}{\partial x_i} \cong \frac{f_i(x_i + \varepsilon|x_i|) - f_i(x_i)}{\varepsilon|x_i|}, \qquad (1.2.33)$$

де ε - відносна погрішність обчислення.

Матриця Якобі в розгорненому вигляді, з обліком (1.2.33), може бути представлена так

$$\left\|\begin{matrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \cdots & \cdots & \cdots & \cdots\\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n}\end{matrix}\right\| \cong$$

$$\cong \left\|\begin{matrix}\frac{f_1(x_1+\varepsilon|x_1|)-f_1(x_1)}{\varepsilon|x_1|} & \cdots & \frac{f_1(x_n+\varepsilon|x_n|)-f_1(x_n)}{\varepsilon|x_n|}\\ \cdots & \cdots & \cdots\\ \frac{f_n(x_1+\varepsilon|x_1|)-f_n(x_1)}{\varepsilon|x_1|} & \cdots & \frac{f_n(x_n+\varepsilon|x_n|)-f_n(x_n)}{\varepsilon|x_n|}\end{matrix}\right\|. \qquad (1.2.34)$$

Далі складаємо і вирішуємо систему лінійних рівнянь для малих приростів x_i

$$\left\|\begin{matrix}\frac{f_1(x_1+\varepsilon|x_1|)-f_1(x_1)}{\varepsilon|x_1|} & \cdots & \frac{f_1(x_n+\varepsilon|x_n|)-f_1(x_n)}{\varepsilon|x_n|}\\ \cdots & \cdots & \cdots\\ \frac{f_n(x_1+\varepsilon|x_1|)-f_n(x_1)}{\varepsilon|x_1|} & \cdots & \frac{f_n(x_n+\varepsilon|x_n|)-f_n(x_n)}{\varepsilon|x_n|}\end{matrix}\right\| \cdot \left\|\begin{matrix}\Delta x_1\\ \cdots\\ \Delta x_n\end{matrix}\right\| =$$

$$= \left\|\begin{matrix}-f_1(x_1 \ldots x_n)\\ \cdots\\ -f_n(x_1 \ldots x_n)\end{matrix}\right\|. \qquad (1.2.35)$$

Визначені з системи (1.2.35) 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

$$f(x) = x^n + a_{n+1} + \sum_{i=2}^{n} a_i x^{n-i+1}, \qquad (1.2.36)$$

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

$$x^2 + px + q, \tag{1.2.37}$$

де p, q – коефіцієнти квадратного трьохчлена.

Якщо обчислення коріння полінома (1.2.36) починається з найбільших по абсолютній величині значень, то беремо трьох перших членів і

$$p = a_2, \quad q = a_3, \tag{1.2.38}$$

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

$$p = \frac{a_n}{a_{n-1}}, \quad q = \frac{a_{n+1}}{a_{n-1}}. \tag{1.2.39}$$

Схема Горнера полягає в тому, що якщо коріння (1.2.37) є корінням полінома (1.2.36), то останній згідно теоремі Безу ділиться на (1.2.37) 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 (1.2.36) на (1.2.37) отриманий поліном матиме ступінь $n-2$ і з ним виконуються операції описані вище. Таким чином, вираз (1.2.36) буде приведений до вигляду: для випадку коли n - парне

$$f(x) = \prod_{i=1}^{\left[\frac{n}{2}\right]} \left(x^2 + c_{2i} x + c_{2i+1} \right), \tag{1.1.20}$$

для випадку коли n - непарне

$$f(x) = \prod_{i=1}^{\left[\frac{n}{2}\right]} \left(x^2 + c_{2i} x + c_{2i+1} \right) \left(x + c_{n+1} \right), \tag{1.1.21}$$

де c_{ij} - дійсні коефіцієнти.

Ітераційний процес визначення коріння на кожному подальшому етапі пониження ступеня полінома (1.2.36) проводиться по наступному алгоритму

$$
\begin{aligned}
&f(x)=V(x)(x^2+px+q)=0,\\
&V(x)=x^{n-2}+b_2x^{n-3}+\ldots+b_{n-2}x+b_{n-1},\\
&b_1=1,\ \ b_2=a_2-p,\ \ b_i=a_i-pb_{i-1}-qb_{i-2},\ \ i=3\ldots n+1,\\
&c_1=1,\ \ c_2=b_2-p,\ \ c_j=b_j-pc_{j-1}-qc_{j-2},\ \ j=3\ldots n-1,\\
&c_n=-pc_{n-1}-qc_{n-2},\ \ d=c_{n-1}^2-c_nc_{n-2},\\
&\Delta p=\frac{b_nc_{n-1}-b_{n+1}c_{n-2}}{d},\ \ \Delta q=\frac{b_{n+1}c_{n-1}-b_nc_n}{d},\\
&p=p+\Delta p,\ \ q=q+\Delta q,\ \ |\Delta p|\le\varepsilon,\ \ |\Delta q|\le\varepsilon,
\end{aligned}
\qquad (1.1.22)
$$

де ε - погрішність визначення коріння.

Обчислення по (1.1.22) проводяться поки $n\rangle 2$. Програма ***KOR_POLIN***, для обчислення всього коріння полінома з дійсними коефіцієнтами, приведена в Додатку.

1.3. 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 виробництв, 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 $[x_O, x]$, яка задовольняє диференціальному рівнянню $\frac{dy}{dx} = f(x, y)$), ми розглядатимемо чисельну інтеграцію системи звичайних диференціальних рівнянь першого порядку

$$\begin{aligned}
&\frac{dy_1}{dx} = f_1(x, y_1, y_2, \ldots y_j, \ldots y_n),\\
&\frac{dy_2}{dx} = f_2(x, y_1, y_2, \ldots y_j, \ldots y_n),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\frac{dy_j}{dx} = f_j(x, y_1, y_2, \ldots y_j, \ldots y_n),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\frac{dy_n}{dx} = f_n(x, y_1, y_2, \ldots y_j, \ldots y_n),
\end{aligned} \qquad (1.3.1)$$

де x - незалежна змінна (час, переміщення, температура і ін.); y_j - залежна змінна; $j = 1 \ldots n$; n - число диференціальних рівнянь, що входять в систему (1.3.1).

Рішення задачі Коші зводиться до чисельної інтеграції системи диференціальних рівнянь (1.3.1), тобто до знаходження інтегральних залежностей

$$y_1 = y_1(x),\ \ y_2 = y_2(x),...y_j = y_j(x),...y_n = y_n(x), \tag{1.3.2}$$

за наступних початкових (граничних) умов

$$x = x_O,\ \ y_{10} = y_1(x_O),\ \ y_{2O} = y_2(x_O),...y_{jO} = y_j(x_O),...y_{nO} = y_n(x_O). \tag{1.3.3}$$

При цьому мається на увазі, що інтегральні криві проходять через крапки, задані початковими (граничними) умовами

$$M_1(x_O, y_{1O}),\ \ M_2(x_O, y_{2O}),...M_j(x_O, y_{jO}),...M_n(x_O, y_{nO}). \tag{1.3.4}$$

До системи (1.3.1) зводиться і будь-яке диференціальне рівняння вищого порядку вигляду

$$\frac{d^n y}{dx^n} = f(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, ... \frac{d^{n-1} y}{dx^{n-1}}). \tag{1.3.5}$$

Для цього використовують метод пониження порядку диференціального рівняння (1.3.5) шляхом введення нових функцій

$$\begin{gathered}
\frac{dy}{dx} = y_1, \\
\frac{dy_1}{dx} = \frac{d^2 y}{dx^2} = y_2, \\
\text{..... ....} \\
\frac{dy_{n-2}}{dx} = \frac{d^{n-1} y}{dx^{n-1}} = y_{n-1}, \\
\frac{dy_{n-1}}{dx} = \frac{d^n y}{dx^n} = f(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, ... \frac{d^{n-1} y}{dx^{n-1}}).
\end{gathered} \tag{1.3.6}$$

Методи чисельної інтеграції одного звичайного диференціального рівняння можна використовувати і для систем (1.3.1) (1.3.6). Порядок погрішності формул для чисельної інтеграції диференціальних рівнянь залежить від вибору методу побудови рекурентної формули.

Методи вирішення диференціальних рівнянь (1.3.1) можна розділити на: методи, що не містять в собі ніяких способів оцінки погрішності; кінцево-різницеві методи, в яких на основі раніше набутих декількох

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.1.3.1 показана розрахункова схема.

Хай задано диференціальне рівняння

$$\frac{dy}{dx} = f(x, y). \qquad (1.3.7)$$

Необхідно знайти наближене рішення (1.3.7) в крапках з координатами $x_1 = x_O + h,\ x_2 = x_O + 2h, \ldots x_n = x_O + nh$, де h - постійний крок; x_O - координата початку відрізку.

Початкова умова $x = x_O,\ y_O = y(x_O)$. Наближене значення першої похідної має вигляд

$$\frac{dy_k}{dx_k} \cong \frac{\Delta y_k}{\Delta x_k} = \frac{y_{k+1} - y_k}{h}, \qquad (1.3.8)$$

де $k = 0,1 \ldots n-1$.

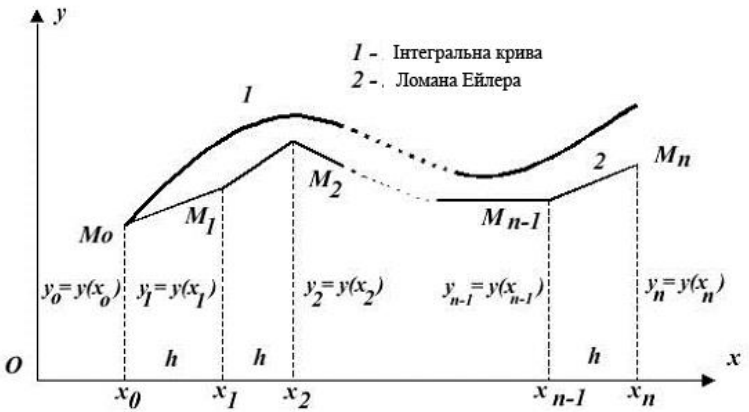


Рис.1.3.1. Розрахункова схема методу Ейлера-Коші

Прирівнюючи (1.3.7) і (1.3.8), отримаємо

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

звідки

$$y_{k+1} = y_k + hf(x_k, y_k). \qquad (1.3.9)$$

Використовуючи рекурентну формулу (1.3.9) для точок $k = 0,1...n-1$ будуємо ламану Ейлера 2, яка приблизно замінює інтегральну криву 1 (см.рис.1.3.1). Суть методу Ейлера-Коші полягає в тому, що через початок кожного відрізку $[x_k, x_{k+1}]$ 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

$$y_{k+1} = y_k + \frac{h}{2}\left\{ f(x_k, y_k) + f\left[x_k + h\ , y_k + hf(x_k, y_k) \right] \right\}. \qquad (1.3.10)$$

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

$$y_{k+1} = y_k + hf\left[x_k + \frac{h}{2}\ , y_k + \frac{h}{2} f(x_k, y_k) \right]\ . \qquad (1.3.11)$$

Графічна інтерпретація даного методу показана на рис.1.3.2. 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.

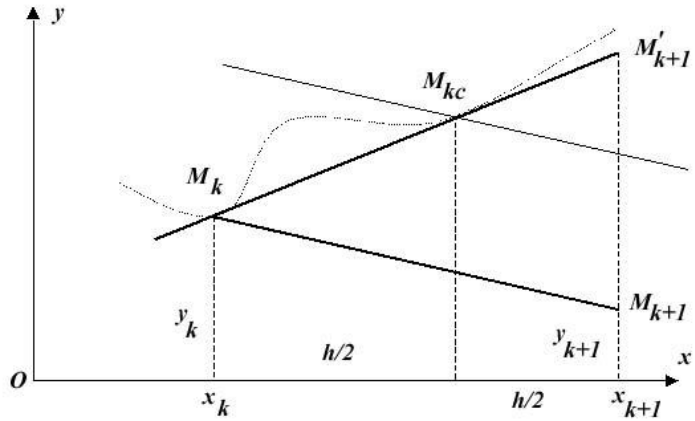


Рис.1.3.2. Графічна інтерпретація методу Рунге-Кутта

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 напряму 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

$$a_k = f\left[x_k + \frac{h}{2},\ y_k + \frac{h}{2} f(x_k, y_k)\right],$$
$$b_k = f\left(x_k + \frac{h}{2},\ y_k + \frac{h}{2} a_k\right),$$
$$c_k = f(x_k + h,\ y_k + b_k h), \qquad (1.3.12)$$
$$y_{k+1} = y_k + \frac{h}{6}\left[f(x_k, y_k) + 2a_k + 2b_k + c_k\right].$$

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

$$
\begin{aligned}
y_{k+1} &= y_k + \frac{h}{6}\left[f(x_k, y_k) + 4V_{3k} + V_{4k}\right], \\
V_{1k} &= hf\left[x_k + \frac{1}{3}h,\ \ y_k + \frac{1}{3}hf(x_k, y_k)\right], \\
V_{2k} &= hf\left[x_k + \frac{1}{3}h,\ \ y_k + \frac{h}{6}f(x_k, y_k) + \frac{1}{6}V_{1k}\right], \\
V_{3k} &= hf\left[x_k + \frac{1}{2}h,\ \ y_k + \frac{h}{8}f(x_k, y_k) + \frac{3}{8}V_{2k}\right], \\
V_{4k} &= hf\left[x_k + h,\ \ y_k + \frac{h}{2}f(x_k, y_k) - \frac{3}{2}V_{2k} + 2V_{3k}\right].
\end{aligned}
\qquad (1.3.13)
$$

Погрішність обчислення y_{k+1}, з обліком (1.3.13), визначається по формулі

$$
R_{k+1} = \frac{-2hf(x_k, y_k) + 9V_{2k} - 8V_{3k} + V_{4k}}{30}. \qquad (1.3.14)
$$

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

$$
|R_{k+1}| \le E,\ \ |R_{k+1}| \ge \frac{E}{30}. \qquad (1.3.15)
$$

Якщо перша умова (1.3.15) 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 (1.3.15). 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 апріорному виборі першого наближення поблизу граничної точки (x_O, y_O)

$$y = d_1(x). \tag{1.3.16}$$

Для (1.3.7), з обліком (1.3.16), отримаємо

$$\begin{aligned} y &= y_O + \int_{x_O}^{x} f[x, d_1(x)]dx = d_2(x), \\ y &= y_O + \int_{x_O}^{x} f[x, d_2(x)]dx = d_3(x), \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ y &= y_O + \int_{x_O}^{x} f[x, d_{n-1}(x)]dx = d_n(x). \end{aligned} \tag{1.3.17}$$

Метод Пікара поблизу граничної крапки при збіжності дає необхідну точність рішення, проте область його застосування обмежена. Це пояснюється необхідністю інтеграції (1.3.7) в кінцевому вигляді.

Другий метод відшукання наближеного рішення поблизу граничної крапки (x_O, y_O) пов'язаний з розкладанням в ряд Тейлора

$$\begin{aligned} y = y_O + \frac{(x-x_O)}{1!}d'(x_O) + \frac{(x-x_O)^2}{2!}d''(x_O) + \ldots + \\ + \frac{(x-x_O)^n}{n!}d^n(x_O) + \ldots, \\ y = d(x). \end{aligned} \tag{1.3.18}$$

Вважаючи, що

$$d'(x) = f(x, y) = \frac{dy}{dx}, \tag{1.3.19}$$

продиференціюємо останню рівність з урахуванням значень в граничній точці

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

або, з обліком (1.3.19), отримаємо

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

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

$$y_{i+1} = y_i + q_i + \frac{1}{2}\Delta q_{i-1} + \frac{5}{12}\Delta^2 q_{i-2} + \frac{3}{8}\Delta^3 q_{i-3}, \quad i = 3,4...,$$
$$q_k = hf(x_k, y_k), \quad \Delta^m q_k = h\Delta^m y'_k, \quad k = 0,1,2... \tag{1.3.20}$$

Для реалізації (1.3.20) при рішенні завдання Коші необхідно визначати на початковому етапі (з використанням методу Пікара або методу розкладання) y_i, де $i = 1...3$. 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.1.3.3) ***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 тяжкість, переносна і кориолісові сили інерції. В рамках справжнього завдання вважатимемо відносну лінійну швидкість і переносну кутову швидкість постійною.

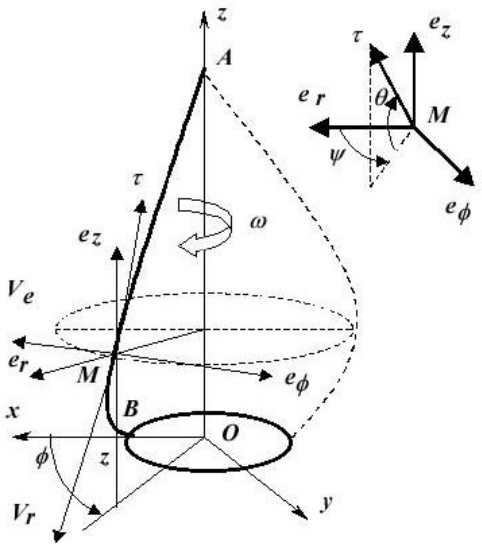


Рис.1.3.3. Розрахункова схема для визначення форми балонуючої нитки

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

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

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

$$r\frac{d\varphi}{ds} = \cos\theta\sin\psi, \qquad (1.3.21)$$

$$\frac{dz}{ds} = \sin\theta,$$

де s - дугова координата.

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

$$\begin{aligned}\frac{d}{ds}\left(P\frac{dr}{ds}\right)-Pr\left(\frac{d\varphi}{ds}\right)^2+F_r=0,\\ \frac{d}{ds}\left(Pr^2\frac{d\varphi}{ds}\right)+rF_\varphi=0,\\ \frac{d}{ds}\left(P\frac{dz}{ds}\right)+F_z=0,\end{aligned} \tag{1.3.22}$$

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

З другого рівняння системи (1.3.21) отримаємо диференціального оператора

$$\frac{d}{ds}=\cos\theta\cos\psi\frac{d}{dr}. \tag{1.3.23}$$

Використовуючи його, перетворимо третє і четверте диференціальні рівняння системи (1.3.21)

$$\begin{aligned}\frac{d\varphi}{dr}=\frac{tg\psi}{r},\ \frac{dz}{dr}=\frac{tg\theta}{\cos\psi},\\ \frac{ds}{dr}=\frac{1}{\cos\theta\cos\psi}.\end{aligned} \tag{1.3.24}$$

Використовуючи (1.3.23) перетворимо систему диференціальних рівнянь (1.3.21)

$$\begin{aligned}\cos\theta\cos\psi\left[\cos\theta\cos\psi\frac{dP}{dr}+P\frac{d}{dr}\left(\cos\theta\cos\psi\right)\right]-\frac{P}{r}\cos^2\theta\sin^2\psi+F_r=0,\\ \cos\theta\cos\psi\left[r\cos\theta\sin\psi\frac{dP}{dr}+P\frac{d}{dr}\left(r\cos\theta\sin\psi\right)\right]+rF_\varphi=0,\\ \cos\theta\cos\psi\left[\sin\theta\frac{dP}{dr}+P\frac{d}{dr}\left(\sin\theta\right)\right]+F_Z=0,\end{aligned} \tag{1.3.25}$$

Помножимо перше рівняння (1.3.25) на $\cos\theta\cos\psi$, друге на $\frac{\cos\theta\sin\psi}{r}$, а третє на $\sin\theta$ і складемо їх [9]

$$\frac{dP}{dr}=-\left(F_r+F_\varphi tg\psi+F_Z\frac{tg\theta}{\cos\psi}\right). \tag{1.3.26}$$

Перетворимо перші два диференціальні рівняння системи (1.3.25) до вигляду [20]

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

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

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

$$\frac{d\psi}{dr}=\frac{F_r\sin\psi-F_\varphi\cos\psi}{P\cos^2\theta\cos\psi}-\frac{tg\psi}{r}. \qquad (1.3.27)$$

Перетворимо третє рівняння системи (1.3.25) з обліком (1.3.26) до вигляду

$$\frac{d\theta}{dr}=\frac{tg\theta}{P}\left(F_r+F_\varphi tg\psi-\frac{F_Z}{tg\theta\cos\psi}\right). \qquad (1.3.28)$$

Рівняння (1.3.24) (1.3.26) (1.3.27).(1.3.28) утворюють систему п'яти диференціальних рівнянь першого порядку. Залишається визначити проекції F_r, F_φ, F_Z активно заданих сил на осі циліндрової координатної системи. Головний вектор зовнішніх сил буде рівний [9,20]

$$\begin{gathered}\vec{F}=F_r\vec{e}_r+F_\varphi\vec{e}_\varphi+F_Z\vec{e}_Z,\\ F_r=\mu\omega^2r-2\mu\omega u\cos\theta\sin\psi-\lambda_2r^2\Phi_n(\chi)ctg\chi\cos\theta\cos\psi,\\ F_\varphi=2\mu\omega u\cos\theta\cos\psi+\lambda_2r^2\Phi_n(\chi)\frac{1}{\sin\chi}-\lambda_2r^2\Phi_n(\chi)ctg\chi\cos\theta\sin\psi,\\ F_Z=-q-\lambda_2r^2\Phi_n(\chi)ctg\chi\sin\theta,\\ \Phi_n(\chi)=a\sin^2\chi,\\ \lambda_2=C\rho\frac{\omega^2}{2}d,\end{gathered} \qquad (1.3.29)$$

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 елементу нитки; C, a- деякі постійні; q - питома вага нитки; ρ - щільність повітря; d - діаметр нитки.

Підставляючи отримані результати з (1.3.29) в (1.3.24) (1.3.26) (1.3.27).(1.3.28), отримаємо наступну систему диференціальних рівнянь

$$\frac{dP}{dr} = q\frac{dz}{dr} - \mu\omega^2 r,$$
$$\frac{d\theta}{dr} = \frac{1}{P}\left[tg\theta\left(\mu\omega^2 r + \lambda_2 r^2 a\sin\chi tg\psi\right) + \frac{q}{\cos\psi}\right],$$
$$\frac{d\psi}{dr} = \frac{\mu\omega^2 r\sin\psi - \lambda_2 r^2 a\sin\chi - 2\mu\omega u\cos\theta}{P\cos^2\theta\cos\psi} - \frac{tg\psi}{r}, \qquad (1.3.30)$$
$$\frac{d\varphi}{dr} = \frac{tg\psi}{r},$$
$$\frac{dz}{dr} = \frac{tg\theta}{\cos\psi}.$$

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 $r = r_B = 50мм,\ \varphi = \varphi_O = 0,\ z = z_O = 0$.

При інтегруванні системи диференціальних рівнянь (1.3.30) використовувався метод Рунге-Кутта четвертого порядку (див. Додаток). Розрахунки проводилися для бавовняної пряжі 25 текс при наступних значеннях:

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

Інтегрування виконували за наступних початкових умов: $P = P_O = 0,2H,\ \theta = \theta_O = 1,2рад,\ \psi = \psi_O = 0,7рад$.

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

1.4. 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 відрізань $[x_{n-1}, x_n]$ 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 $x_0 = a,\ x_1, x_2, ... x_n = b$, причому $y_i = f(x_i)$, $(i = 0,1,2,...n)$. Необхідно визначити функцію, що наближає $g(x)$ в точках $x_0, x_1, x_2, ... x_n$, щоб виконувалася наближена рівність

$$\int_a^b f(x)dx \cong \int_a^b g(x)dx. \qquad (1.4.1)$$

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

$$L_n(x) = \sum_{i=0}^{n} f(x_i) \prod_{j \neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)} == \sum_{i=0}^{n} f(x_i) \frac{\prod_{j=0}^{n} (x - x_j)}{(x - x_i) \prod_{j \neq i}^{n} (x_i - x_j)}. \qquad (1.4.2)$$

Інтерполяційний поліном Ньютона з розділеними різницями матиме вигляд

$$L_n(x) = f(x_0) + f(x_0, x_1)(x - x_0) + ... + f(x_0, x_1, x_2, ... x_n) \prod_{j-0}^{n-1} (x - x_j), \qquad (1.4.3)$$

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

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

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

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

Використовуючи вирази для інтерполяційних поліномів (1.4.2).(1.4.3), отримаємо з (4.4.1)

$$\int_a^b f(x)=\int_a^b L_n(x)dx+R_n, \qquad (1.4.4)$$

де R_n - погрішність квадратурної формули.

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

$$R_n(F)=\sup_{f\in F}\left|\sum_{k=0}^{n}C_k f(x_k)-\int_{\Omega} f(x)\omega(x)dx.\right| \qquad (1.4.5)$$

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 (1.4.4) по формулі

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

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

$$H_i=\frac{(-1)^{n-i}}{ni!(n-i)!}\int_O^n\frac{u(u-1)(u-2)...(u-n)}{u-i}du, \qquad (1.4.6)$$

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

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

$$H_0 = \frac{1}{4}\int_0^2 (u-1)(u-2)du = \frac{1}{6},$$
$$H_1 = -\frac{1}{2}\int_0^2 u(u-2)du = \frac{2}{3}, \qquad (1.4.7)$$
$$H_2 = \frac{1}{4}\int_0^2 u(u-1)du = \frac{1}{6}.$$

Підставляємо (4.4.7) в (4.4.6), тоді

$$\int_{x_0}^{x_2} f(x)dx \cong \frac{(x_2 - x_0)}{6}\left[f(x_0) + 4f(x_1) + f(x_2)\right]. \qquad (1.4.8)$$

Розіб'ємо відрізок $[a,b]$ на парне число n ділянок $[x_0,x_2],[x_2,x_4],\ldots[x_{n-2},x_n]$ довжини $(b-a)/n = h$. Використовуючи (1.4.8) і (1.4.6) отримаємо

$$\int_a^b f(x)dx \cong \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] + \frac{h}{3}\left[f(x_2) + 4f(x_3) + f(x_4)\right] +$$
$$+ \ldots + \frac{h}{3}\left[f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right].$$

Згрупуємо члени і отримаємо відому формулу Сімпсона [14]

$$\int_a^b f(x)dx \cong \frac{b-a}{3n}\left\{f(a) + f(b) + 4\left[f(x_1) + f(x_3) + \ldots + f(x_{n-1})\right] + 2\left[f(x_2) + f(x_4) + \ldots + f(x_{n-2})\right]\right\}. \qquad (1.4.9)$$

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.

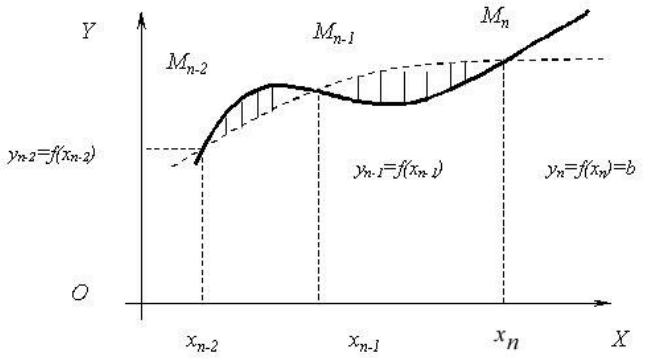


Рис.1.4.1. Розрахункова схема інтерполяції підінтегральної функції по формулі Сімпсона

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

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

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

При $n=1$ (функція, що наближає, є лінійною) з (1.4.6) набудемо наступних значень коефіцієнтів Лагранжа

$$H_O = -\int_0^1 (u-1)du = \frac{1}{2}$$
$$H_1 = \int_0^1 udu = \frac{1}{2}. \qquad (1.4.10)$$

Підставляємо (1.4.10) в (1.4.6), тоді

$$\int_{x_o}^{x_1} f(x)dx \cong \frac{(x_1 - x_O)}{2}\left[f(x_O) + f(x_1)\right]. \qquad (1.4.11)$$

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

$$x_1 - x_O = \frac{b-a}{n} = h. \qquad (1.4.12)$$

Використовуючи (1.4.11).(1.4.12) і (1.4.6) отримаємо з урахуванням суміжних точок відрізань

$$\int_a^b f(x)dx \cong \frac{b-a}{n}\left[\frac{1}{2}f(a)+f(x_1)+f(x_2)+\ldots+f(x_{n-1})+\frac{1}{2}f(b)\right]. \qquad (1.4.13)$$

Формула (1.4.13) є формулою трапецій для наближеного обчислення певного інтеграла (1.4.1).

Геометричний сенс (1.4.13) (см.рис.1.4.2) полягає в тому, що подинтегральної функція $f(x)$ замінюється на кожному інтервалі деякою лінійною функцією, що наближає. На кожному інтервалі $[a,x_1],[x_1,x_2],[x_2,x_3],\ldots[x_{n-1},b]$ визначається площа трапеції, як твір напів суми підстав на висоту. Висота для всіх трапецій, за умови постійності кроку розбиття, дорівнюватиме $h=(b-a)/n$.

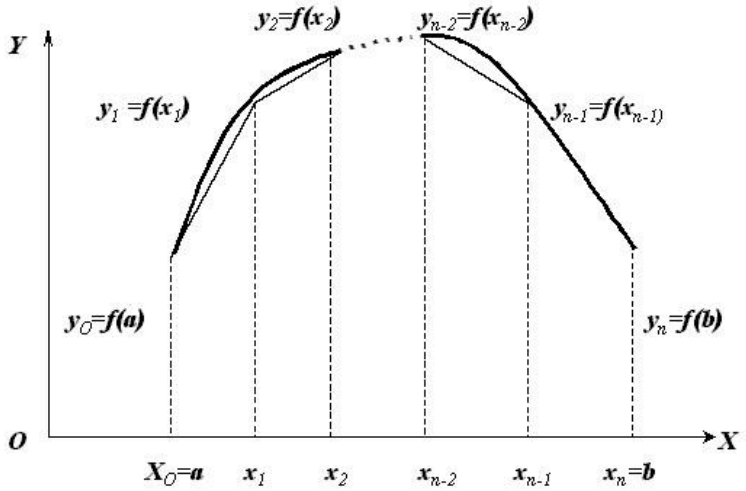


Рис.1.4.2. Геометрична інтерпретація формули трапеції

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

$$R_n = \frac{(b-a)^3}{12n^2}\left|f^{II}(x_*)\right|. \qquad (1.4.14)$$

При $n=0$ (функція, що наближає, є лінійній, паралельній осі x) з (1.4.6) набудемо значення коефіцієнта Лагранжа

$$H_O = 1. \qquad (1.4.15)$$

Підставляємо (1.4.15) в (1.4.6), тоді для відрізання $[x_O, x_1]$ отримаємо

$$\int_{x_o}^{x_1} f(x)dx \cong (x_1 - x_O) f(x_O). \qquad (1.4.16)$$

Розіб'ємо відрізок $[a,b]$ на n рівних частин $[x_O = a, x_1], [x_1, x_2], [x_2, x_3], ... [x_{n-2}, x_{n-1}], [x_{n-1}, x_n = b]$. Середини даних відрізань визначаться з виразів $\xi_1 = a + \frac{h}{2}$, $\xi_2 = x_1 + \frac{h}{2}, ... \xi_{n-1} = x_{n-2} + \frac{h}{2}$, $\xi_n = x_{n-1} + \frac{h}{2}$. Використовуючи (1.4.12), (1.4.16) і (1.4.6) отримаємо

$$\int_a^b f(x)dx \cong \left[f(\xi_1) + f(\xi_2) + f(\xi_3) + ... + f(\xi_{n-1}) + f(\xi_n)\right]\frac{b-a}{n}. \qquad (1.4.17)$$

Вираз (1.4.17) є формулою прямокутників для наближеного обчислення певного інтеграла (1.4.1).

Геометричний сенс (1.4.17) (см.рис.1.4.3) 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 $\xi_1, \xi_2, ... \xi_{n-1}, \xi_n$ відрізань $[x_O = a, x_1], [x_1, x_2], [x_2, x_3] ... [x_{n-2}, x_{n-1}], [x_{n-1}, x_n = b]$.

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

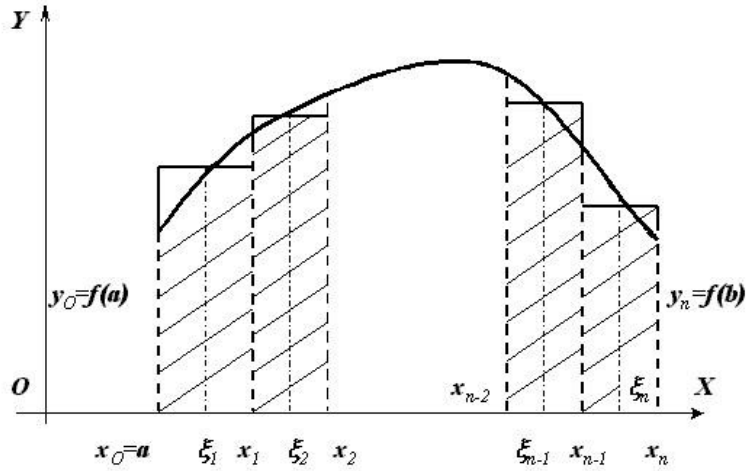


Рис.1.4.3. Геометрична інтерпретація формули прямокутників

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

$$R_n = \frac{(b-a)^3}{24n^2}\left|f^{II}(x_*)\right|. \qquad (1.4.18)$$

Аналіз залежностей (1.4.18) і (1.4.14) дозволяє стверджувати, що формули прямокутників і трапецій володіють приблизно однаковою точністю. Вибір тієї або іншої формули залежить від виду подинтегральної формули, обчислювальних можливостей ЕОМ.

1.5. 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 $x_1, x_2, x_3, ..., x_n$. Інтерполяція має на увазі виконання наступної рівності

$$f(x_i) = g(x_i), \quad i = 1,2,...n. \tag{1.5.1}$$

Метод інтерполяції іноді називають точним методом. При екстраполюванні 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

$$v_i = |f_1(x_i) - f(x_i)| = min, \quad i = 1,2,...n, \tag{1.5.2}$$

де v_1 - 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 вид функції, що наближає, визначають з вимоги мінімізації суми

$$S = min\left\{\sum_{k=1}^{n}\alpha_k\left[f(x_k)-f_1(x_k)\right]^2\right\}, \qquad (1.5.3)$$

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 (1.5.1), що наближає, представляється інтерполяційним многочленом

$$g(x)=\sum_{i=1}^{n}a_i\varphi_i(x), \qquad (1.5.4)$$

де a_i - коефіцієнти многочлена (1.5.4), визначувані з умови збіги заданої функції $f(x)$ і функції, що наближає $g(x)$ у крапках $x_1, x_2, x_3, ...x_n$; $\varphi_i(x)$ – деякі фіксовані функції.

Обчислення коефіцієнтів многочлена (1.5.4) можна проводити по методу Ейткена шляхом послідовного застосування формул лінійної інтерполяції при довільному розташуванні вузлів без явної побудови інтерполяційного полінома по формулах

$$P_{i,i+1}(x)=\frac{1}{x_{i+1}-x_i}\begin{vmatrix} x-x_i & f(x_i) \\ x-x_{i+1} & f(x_{i+1}) \end{vmatrix}, \quad i=1,2...n-1,$$
$$P_{j-1,j,j+1}(x)=\frac{1}{x_{j+1}-x_{j-1}}\begin{vmatrix} x-x_{j-1} & P_{j-1,j}(x) \\ x-x_{j+1} & P_{j,j+1}(x) \end{vmatrix}, \quad j=2,...n-1. \qquad (1.5.5)$$

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 (1.5.5) по методу Ейткена за допомогою функціональних визначників другого порядку побудуємо інтерполяційний поліном

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

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

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

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

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

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

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

Як наголошувалося в підрозділі 1.4, 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]

$$LT_n(x)=A+\sum_{k=1}^{n}(a_k\cos kx+b_k\sin kx), \qquad (1.5.6)$$

де A, a_k, b_k - коефіцієнти тригонометричного полінома (1.5.6). 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 якого-небудь 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 багато інтервальної 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 $\Delta n: a = x_0 < x_1 < ... < x_n = b$ 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-функции має вигляд [14]

$$S_m(\Delta_n; x) = P_{m-1}(x) + \sum_{k=0}^{n-1} c_k (x - x_k)_+^m, \qquad (1.5.7)$$

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

$$\begin{aligned} &S_2'(a) = a_n,\ S_2'(b) = b_n, \\ &S_2''(a) = A_n, S_2''(b) = B_n. \end{aligned} \qquad (1.5.8)$$

Якщо функція $f(x)$ має першу і другу похідні на відрізку $[a,b]$то значення системи (1.5.8) можуть бути визначені з наступної системи рівнянь

$$\begin{aligned} &a_n = f'(a),\ b_n = f'(b), \\ &A_n = f''(a),\ B_n = f''(b). \end{aligned} \qquad (1.5.9)$$

Для параболічного сплайна отримаємо

$$\begin{gathered} S_2(x) = f(x_i) + m_i(x - x_i) + c_i(x - x_i)^2, \\ c_i = \frac{m_{i+1} - m_i}{2h_i} - \frac{f(x_{i+1}) - f(x_i)}{h_i(\bar{h}_i - h_i)} + \frac{m_{i+1} + m_i}{2(\bar{h}_i - h_i)}, \\ h_i = x_{i+1} - x_i,\ \bar{h}_i = x_{i+1} - \bar{x}_{i+1},\ m_i = S_2'(x_i), \\ i = 0,...n, \end{gathered} \qquad (1.5.10)$$

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

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

Для кубічного сплайна на часткових відрізках $[x_i, x_{i+1}]$ загальна формула, з обліком (1.5.7), має вигляд

$$S_3(x) = \frac{1}{6h_i}\left[m_i(x_{i+1} - x)^3 + m_{i+1}(x - x_i)^3\right] +$$
$$+ \frac{1}{h_i}\left\{\left[f(x_i) - \frac{m_i h_i^2}{6}\right](x_{i+1} - x) + \left[f(x_{i+1}) - \frac{m_{i+1} h_i^2}{6}\right](x - x_i)\right\}, \qquad (1.5.11)$$
$$h_i = x_{i+1} - x_i, \ m_i = f''(x_i), \ i = 1,2,...n,$$

де n - число вузлів.

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

$$h_i m_i + 2(h_i + h_{i+1})m_{i+1} + h_{i+1}m_{i+2} = 6\left[\frac{f(x_{i+2}) - f(x_{i+1})}{h_{i+1}} - \frac{f(x_{i+1}) - f(x_i)}{h_i}\right], \qquad (1.5.12)$$
$$i = 1,2,...n.$$

Систему лінійних рівнянь (1.5.12) необхідно доповнити граничними умовами

$$S_3'(a) = a_n, \ S_3'(b) = b_n,$$
$$S_3''(a) = A_n, S_3''(b) = B_n. \qquad (1.5.13)$$

Якщо функція $f(x)$ має відповідні похідні, то система (1.5.13) визначається з (1.5.9). Якщо значення другої похідної рівні m_1=0, m_n=0, то отримуємо нормальні сплайн-функции. У разі, коли $m_1 = m_n$ маємо періодичні сплайн-функции.

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 біль 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. 1.5.1 показаний розподіл пар значень x_i, y_i і графік лінійної апроксимуючої залежності $y = y(x)$.

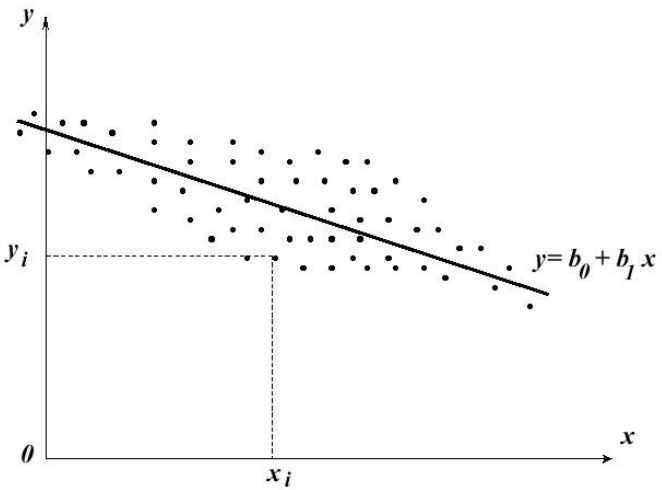


Рис.1.5.1. Лінійна апроксимація

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, \\ &\ldots\ldots\ldots

Вирішуючи систему двох лінійних рівнянь (1.5.18) щодо постійних коефіцієнтів лінійного рівняння апроксимації, отримаємо

$$b_0 = \frac{1}{N}\left(\sum_{i=1}^{N} y_i - b_1 \sum_{i=1}^{N} x_i\right),$$
$$b_1 = \frac{\sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i - N\sum_{i=1}^{N} x_i y_i}{\left(\sum_{i=1}^{N} x_i\right)^2 - N\sum_{i=1}^{N} x_i^2}. \qquad (1.5.19)$$

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

$$y(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m. \qquad (1.5.20)$$

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 (1.5.20), у виконанні наступної умови

$$U = \sum_{i=1}^{N} U_i = \min,$$
$$U_1 = \left[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \ldots - a_m x_1^m\right]^2,$$
$$U_2 = \left[y_2 - a_0 - a_1 x_2 - a_2 x_2^2 - \ldots - a_m x_2^m\right]^2, \qquad (1.5.21)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$U_N = \left[y_N - a_0 - a_1 x_N - a_2 x_N^2 - \ldots - a_m x_N^m\right]^2.$$

Для мінімізації величини U з обліком (1.5.21), визначимо значення приватних похідних по відповідних коефіцієнтах полінома (1.5.20)

$$\begin{aligned}
&\frac{\partial U_i}{\partial a_0} = \frac{\partial U_i}{\partial a_1} = \frac{\partial U_i}{\partial a_2} = \ldots = \frac{\partial U_i}{\partial a_m} = 0, \quad i = 1 \ldots N, \\
&\frac{\partial U_1}{\partial a_0} = -2\left[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \ldots - a_m x_1^m\right], \\
&\frac{\partial U_1}{\partial a_1} = -2\left[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \ldots - a_m x_1^m\right] x_1, \\
&\frac{\partial U_i}{\partial a_2} = -2\left[y_1 - a_0 - a_1 x_1 - a_2 x_1^2 - \ldots - a_m x_1^m\right] x_1^2, \\
&\ldots\ldots\ldots

$$\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 + \ldots + 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 + \ldots + 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 + \ldots + a_m\sum_{i=1}^{N} x_i^{m+2},\\
&\ldots\ldots\ldots

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

$$P(S) = \frac{SB}{SO}. \qquad (1.6.1)$$

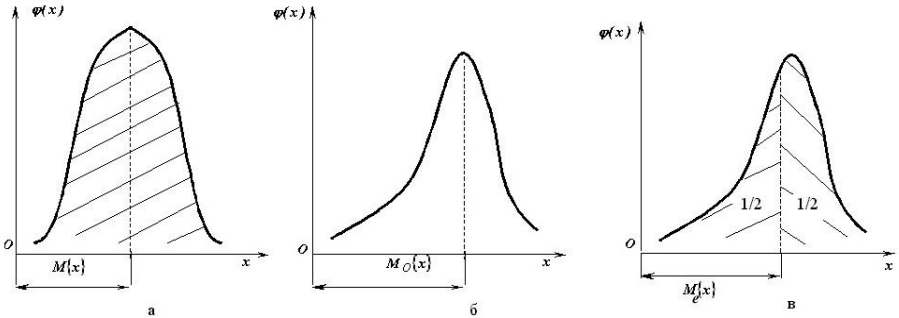
Велике значення при проведенні експериментальних досліджень необхідно приділяти точність вимірювань. Отриманий експериментальним шляхом результат завжди містить деяку помилку. Розрізняють систематичні, випадкові і грубі помилки. Систематичні помилки з'являються унаслідок несправності вимірювальних приладів, 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 неуважністю 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]

$$F(x) = \int_{-\infty}^{x} \varphi(x)dx. \qquad (1.6.2)$$

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.1.6.1 а), мода випадкової величини $M_O\{x\}$ (абсциса відповідає максимальному значенню щільності вірогідності – рис.1.6.1 би)медіана випадкової величини $M_e\{x\}$ (абсциса точок прямої, яка ділить площу, обмежену кривій щільності вірогідності, навпіл – рис.1.6.1 в) [31].



Ріс.1.6.1. Графік щільності вірогідності з вказівкою: математичного очікування, моди і медіани

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

$$D\{x\}=\sigma^2=\int_{-\infty}^{\infty}(x-a)^2\varphi(x)dx. \qquad (1.6.3)$$

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

$$\sigma=\sqrt{\sigma^2}. \qquad (1.6.4)$$

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

$$\gamma = \frac{\sqrt{\int_{-\infty}^{\infty}\left[x - \int_{-\infty}^{\infty} x\varphi(x)dx\right]^2 \varphi(x)dx}}{\int_{-\infty}^{\infty} x\varphi(x)dx} = \frac{\sigma}{a}. \qquad (1.6.5)$$

Вираз (1.6.5) показує наскільки велике розсіяння в порівнянні з середнім значенням випадкової величини.

Показник S_k асиметрії розподілу визначається по формулі

$$S_k = \frac{\mu_3}{\sigma^3}, \quad \mu_3 = \int_{-\infty}^{\infty} (x-a)^3 \varphi(x)dx. \qquad (1.6.6)$$

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

$$E_k = \frac{\mu_4}{\sigma^4} - 3, \quad \mu_4 = \int_{-\infty}^{\infty} (x-a)^4 \varphi(x)dx. \qquad (1.6.7)$$

Визначені по (1.6.6).(1.6.7) 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

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

експоненціальне

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

показово-статечне

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

Вейбулла - Гнеденко

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

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

Бета - розподіл

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

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

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

$$\bar{x}=\frac{1}{n}\sum_{k=1}^{n}x_k, \qquad (1.6.8)$$

де n - число вимірювань.

Достатньо точною оцінкою помилки можна вважати вибіркову 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 вибірковим середнім значенням і вибірковою дисперсією.

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

$$S_n = \sqrt{\frac{\sum_{k=1}^{n} (\bar{x} - x_k)^2}{n-1}}. \qquad (1.6.9)$$

Середня квадратична погрішність ряду вимірювань визначається по формулі

$$S_{\bar{x}} = \frac{S_n}{\sqrt{n}}. \qquad (1.6.10)$$

Аналіз залежності (1.6.10) 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 1.6.1.Значение критерію Стьюдента (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 $S^2_{(x_k)}$ є експериментальною оцінкою дисперсії

$$S^2_{(x_k)} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})^2. \qquad (1.6.11)$$

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 (1.6.8) для розрахунку середньоарифметичного значення $\bar{x}$).

Довірча помилка $\varepsilon_{(x_k)}$ для одиничного результату, з обліком (1.6.11), дорівнює $\varepsilon_{(x_k)} = t \cdot S_{(x_k)}$ а довірча помилка для середнього результату $\varepsilon_{(\bar{x})}$ з обліком (1.6.10), визначається по формулі $\varepsilon_{(\bar{x})} = t \cdot S_{(\bar{x})}$.

Відносну помилку середнього результату визначають по формулі

$$O_{(\bar{x})} = \frac{\varepsilon_{(\bar{x})}}{\bar{x}} 100\%. \qquad (1.6.12)$$

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

$$\Delta = \frac{S_{(\bar{x})}}{\bar{x}} 100\%. \qquad (1.6.13)$$

Як практичне застосування розглянемо випадок визначення необхідної кількості повторних дослідів для отримання результатів із заданою точністю. На попередньому етапі визначуваний по (1.6.9).(1.6.10) 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 $\varepsilon_{(\bar{x})} = (0{,}05...0{,}1)\cdot\bar{x}$). Тоді, з обліком (1.6.10).(1.6.11), отримаємо

$$t_p = \frac{\varepsilon_{(\bar{x})}}{S_{(x_k)}}\sqrt{n}. \qquad (1.6.14)$$

Для заданого значення довірчої вірогідності α і числа мір свободи $f = n-1$ визначаємо по таблиці 1.6.1 значення t- критерію Стьюдента t_T. По формулі (1.6.14) 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 (1.6.14) 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

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

Потім визначаємо розрахункове значення t_p- критерію Стьюдента по формулі

$$t_p = \frac{\left|x_{kS} - \bar{x}_{(n-1)}\right|}{S_{(x_k)}}. \qquad (1.6.15)$$

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 Стьюдента 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

$$x_i = \frac{X_i - X_{iO}}{h_i}, \qquad (1.6.16)$$

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_{iO} - 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 $k \le 5$ вельми ефективним є повний факторний експеримент (ПФЕ) або дробовий факторний експеримент (ДФЕ). Дані плани володіють ортогональністю (сума почленных 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 нім реалізуються всі можливі комбінації даних рівнів чинників). Загальне число дослідів визначається з виразу

$$N = 2^k. \qquad (1.6.17)$$

Графічна інтерпретація ПФЕ для двох чинників показана на рис.1.6.2 а. Матриця ПФЕ включає: стовпці, відповідні кожному чиннику; стовпці, відповідні взаємодіям чинників. Число рядків визначається по (1.6.17). У таблиці 1.6.2 показана матриця ПФЕ для двох чинників.

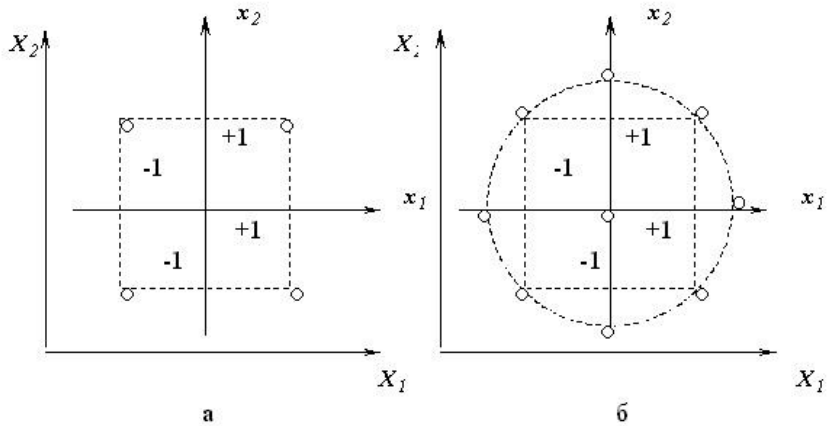


Рис.1.6.2. Графічна інтерпретація ПФЕ і ротатабельного плану для 2 чинників

Таблиця 1.6.2. 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]

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2,$$
$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3. \qquad (1.6.18)$$

Коефіцієнти в рівняннях регресії (1.6.18) визначаються по наступних

формулах

$$b_0 = \frac{\sum_{1}^{N} \bar{y}_u}{N}, \quad b_i = \frac{\sum_{1}^{N} x_{iu}\bar{y}_u}{N}, \quad b_{ij} = \frac{\sum_{1}^{N} x_{iu}x_{ju}\bar{y}_u}{N}, \quad b_{ijk} = \frac{\sum_{1}^{N} x_{iu}x_{ju}x_{ku}\bar{y}_u}{N}, \qquad (1.6.19)$$

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) незначущо відрізняються від нуля. Тоді стає можливим використовувати в плані ПФЕ стовпці, що відносяться до взаємодій вищих порядків, для оцінки додаткових лінійних ефектів.

Оцінка значущості коефіцієнтів регресії (1.6.19) 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

$$\Delta b_i = \frac{t \cdot S_{\{y\}}}{\sqrt{N_1 n}}, \quad S_{\{y\}} = \sqrt{\frac{\sum_{1}^{N}\sum_{1}^{n}(y_{uj} - \bar{y}_u)^2}{N(n-1)}}, \qquad (1.6.20)$$

де n - число спостережень в окремому досвіді; N - число рядків матриці планування експерименту; t - критерій Стьюдента (див. таблицю 1.6.1); N_1 - число дослідів, що враховуються при розрахунку коефіцієнта.

Ротатабільне 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.1.6.2 би показана графічна інтерпретація ротатабельного плану другого порядку для двох чинників. У таблиці 1.6.3 представлені основні співвідношення між числом чинників і загальним числом дослідів для ротатабельного плану другого порядку [32].

Таблиця 1.6.3. Залежність числа дослідів від числа чинників для ротатабельного плану другого порядку

Число чинників k	Число точок «ядра» $n_я$	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 1.6.4 представлена матриця планування експерименту для ротатабельного плану другого порядку для 2 чинників

Таблиця 1.6.1. 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 ротатабельного плану визначаються по наступних формулах [32]

$$b_0 = a_1\sum_{j=1}^{N} y_j - a_2\sum_{i=1}^{k}\sum_{j=1}^{N} x_{ij}^2 y_j,\quad b_i = a_3\sum_{j=1}^{N} x_{ij} y_j,$$
$$b_{ij} = a_4\sum_{j=1}^{N} x_{ij}x_{jj} y_j,\quad b_{ii} = a_5\sum_{j=1}^{N} x_{ij}^2 y_j + a_6\sum_{i=1}^{k}\sum_{j=1}^{N} x_{ij}^2 y_j - a_7\sum_{j=1}^{N} y_j, \qquad (1.6.21)$$

де $a_1 \ldots a_7$ - постійні коефіцієнти, визначувані з таблиці 1.6.5.

Таблиця 1.6.5. Значення коефіцієнтів $a_1 \ldots a_7$ у виразі (1.6.21)

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 ротатабельном плануванні другого порядку визначається по формулах

$$\Delta b_0 = \pm 2S_{\{b_0\}},\quad \Delta b_i = \pm 2S_{\{b_i\}}, \Delta b_{ii} = \pm 2S_{\{b_{ii}\}},\quad \Delta b_{ij} = \pm 2S_{\{b_{ij}\}},$$
$$S_{\{b_0\}} = \sqrt{a_8}\,S_{\{\bar{y}\}},\quad S_{\{b_i\}} = \sqrt{a_9}\,S_{\{\bar{y}\}},\quad S_{\{b_{ii}\}} = \sqrt{a_{10}}\,S_{\{\bar{y}\}},\quad S_{\{b_{ij}\}} = \sqrt{a_{11}}\,S_{\{\bar{y}\}}, \qquad (1.6.22)$$
$$S_{\{\bar{y}\}} = \frac{S_{\{y\}}}{\sqrt{n}},\quad S_{\{y\}} = \sqrt{\frac{\sum_{1}^{N}\sum_{1}^{n}(y_{uj} - \bar{y}_u)^2}{N(n-1)}}.$$

Значення коефіцієнтів $a_8 \ldots a_{11}$ приведені в таблиці 1.6.6 [32].

Таблиця 1.6.6. Значення коефіцієнтів $a_8 ... a_{11}$ у виразі (1.6.22)

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 1.6.7 приведені основні характеристики ортогонального плану для різного числа чинників.

Таблиця 1.6.7. Ортогональні плани другого порядку

Число чинників k	Число точок «ядра» $n_я$	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

$$b_i = \frac{\sum_1^{N_1} x_{iu} y_u}{N_1}, \quad b_{ij} = \frac{\sum_1^{N_2} x_{iu} x_{ju} y_u}{N_2}, \quad b_{ii} = \frac{\sum_1^{N} (x'_{iu})^2 y_u}{\sum_1^{N} (x'_{iu})^2},$$

$$b_0 = \frac{\sum_1^{N} y_u}{N} - \frac{\sum_1^{N} x_{iu}^2}{N} \sum_1^{k} b_{ii}, \quad (x'_{iu})^2 = x_{iu}^2 - \frac{\sum_1^{N} x_{iu}^2}{N}, \tag{1.6.23}$$

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

$$\Delta b_i = \pm 2 S_{\{b_i\}}, \quad S_{\{b_i\}} = \frac{S_{\{\bar{y}\}}}{\sqrt{A}}, \quad S_{\{\bar{y}\}} = \frac{S_{\{y\}}}{\sqrt{n}}, \quad S_{\{y\}} = \sqrt{\frac{\sum_{1}^{N}\sum_{1}^{n}(y_{uj} - \bar{y}_u)^2}{N(n-1)}}. \qquad (1.6.24)$$

Значення коефіцієнта A визначаємо з таблиці 1.6.8 [32].

Таблиця 1.6.8. Значення коефіцієнта A у виразі (1.6.24)

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

$$F_p = \frac{S^2_{ad}}{S^2_{\{y\}}}, \qquad (1.6.25)$$

де S^2_{ad} - дисперсія адекватності рівняння регресії; $S^2_{\{y\}}$ - дисперсія відтворюваності.

Для ПФЕ

$$S^2_{ad} = \frac{\sum_{1}^{N} n(\bar{y}_u - \hat{y}_u)^2}{N - k - 1}, \quad S^2_{\{y\}} = \frac{\sum_{1}^{N}\sum_{1}^{n}(y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

Для ортогонального планування другого порядку

$$S^2_{ad} = \frac{\sum_{1}^{N} n(\bar{y}_u - \hat{y}_u)^2}{N - \frac{(k+2)(k+1)}{2}}, \quad S^2_{\{y\}} = \frac{\sum_{1}^{N}\sum_{1}^{n}(y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

Для ротатабельного планування другого порядку

$$S^2_{ad} = \frac{\sum_1^N n(\bar{y}_u - \hat{y}_u)^2}{N - \frac{(k+2)(k+1)}{2} - (n_0 - 1)}, \quad S^2_{\{y\}} = \frac{\sum_1^N \sum_1^n (y_{uj} - \bar{y}_u)^2}{N(n-1)}.$$

де n_0 - число повторностей нульового досвіду.

На наступному етапі, після визначення по (1.6.25) 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

$$f_e = N(n-1).$$

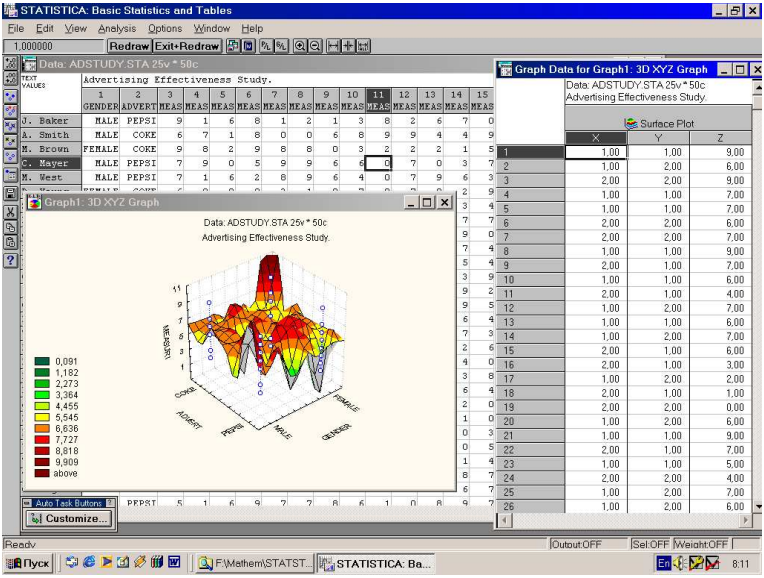
Для того, щоб регресійна модель, при вибраній довірчій вірогідності, адекватно описувала досліджуваний процес необхідне виконання умови

$$F_p < F_T. \qquad (1.6.26)$$

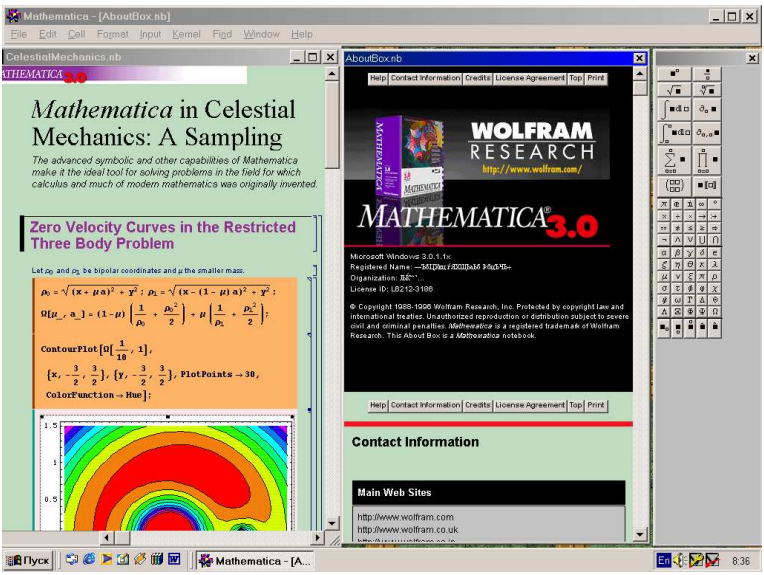
З розвитком інформаційних технологій велике значення придбали різні прикладні програмні пакети для статистичної обробки даних і реалізації активного планування експерименту. На рис.1.6.3а,б показані робочі вікна пакетів ***STATISTICA 5.0*** і ***Mathematica3*** [33, 34]. На рис.1.6.4а,б показані робочі вікна пакетів ***SPSS*** і ***STATGRAPHI CS*** [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.

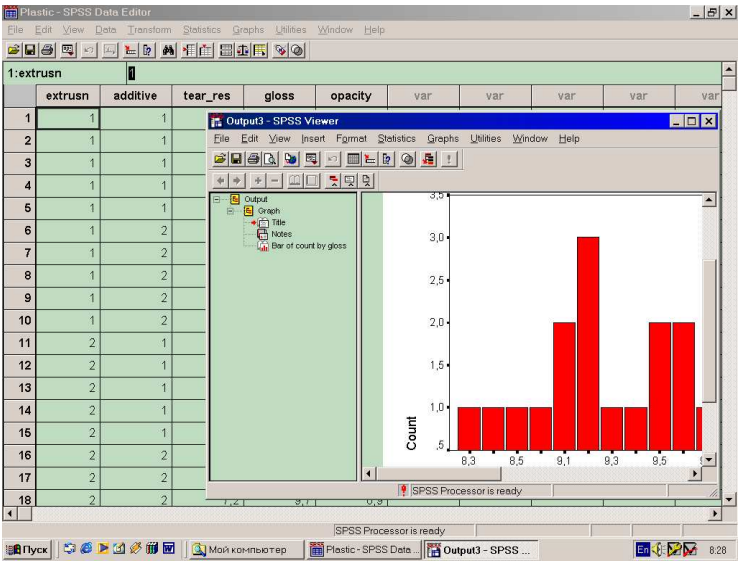


а

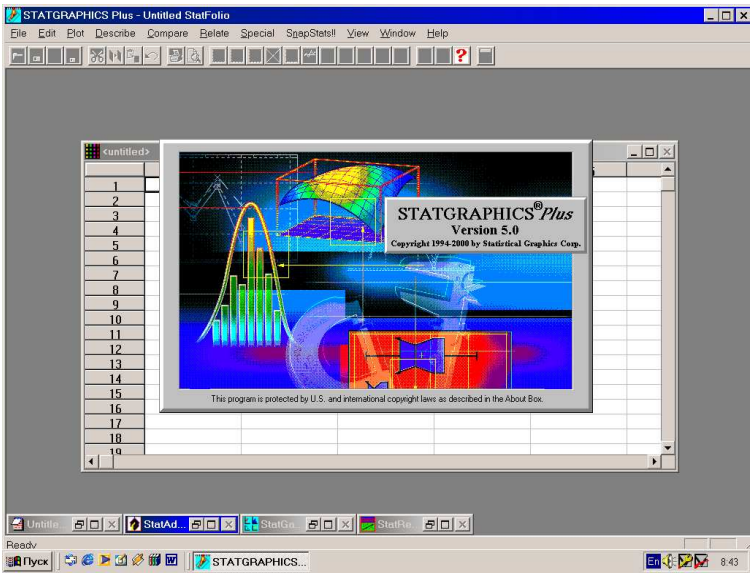


би

Рис.1.6.3. Робочі вікна пакетів ***STATISTICA 5.0*** і ***Mathematica3***



а



би

Рис.1.6.1. Робочі вікна пакетів ***SPSS*** і ***STATGRAPHI CS***

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 символьної 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 $\{x_1, x_2, ... x_n\}$; ***Median***- повертає медіану даних $\{x_1, x_2, ... x_n\}$; ***StandardDeviation*** - повертає стандартне середньо квадратичне відхилення для даних $\{x_1, x_2, ... x_n\}$; ***Variance*** - повертає варіацію для даних $\{x_1, x_2, ... x_n\}$.

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 багатофакторного дисперсійного аналізу, методи порівняння декількох вибірок даних; ***Relate*** - 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 центроідні 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

$$a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + 2a_{23}yz + 2a_{14}x + 2a_{24}y + \\ + 2a_{34}z + a_{44} = 0, \qquad (1.6.42)$$

визначають основні інваріанти ***S, T, ф, U*** і напив інваріантні (семі інваріанти) ***Uг, UU***. Потім по таблиці 1.6.15 визначають, який геометричний образ має поверхню відгуку для тієї або іншої регресійної залежності (1.6.27.1.6.41).

Основні інваріанти і напив інваріантні визначаються як визначники і їх суми відповідних матриць коефіцієнтів многочлена алгебри. Формули для розрахунку приведені у вигляді системи (1.6.43). Перетворимо регресійні залежності (1.6.27).(1.6.41) до вигляду (1.6.42), для чого підставимо в

останні значення кута обхвату і швидкості руху нитки, які відповідатимуть своїм мінімальним значенням (див.

$$
\begin{aligned}
&S = a_{11} + a_{22} + a_{33};\\
&\delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix};\\
&T = \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} \\ a_{13} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{23} & a_{33} \end{vmatrix};\\
&\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{vmatrix};\\
&\Delta' = \begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{23} & a_{33} & a_{34} \\ a_{24} & a_{34} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} & a_{14} \\ a_{13} & a_{33} & a_{34} \\ a_{14} & a_{34} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{12} & a_{22} & a_{24} \\ a_{14} & a_{24} & a_{44} \end{vmatrix};\\
&\Delta'' = \begin{vmatrix} a_{11} & a_{14} \\ a_{14} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{24} \\ a_{24} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{33} & a_{34} \\ a_{34} & a_{44} \end{vmatrix}.
\end{aligned}
\qquad (1.6.43)
$$

план проведення експерименту табл. 1.6.11).

При підстановці ***ц=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.

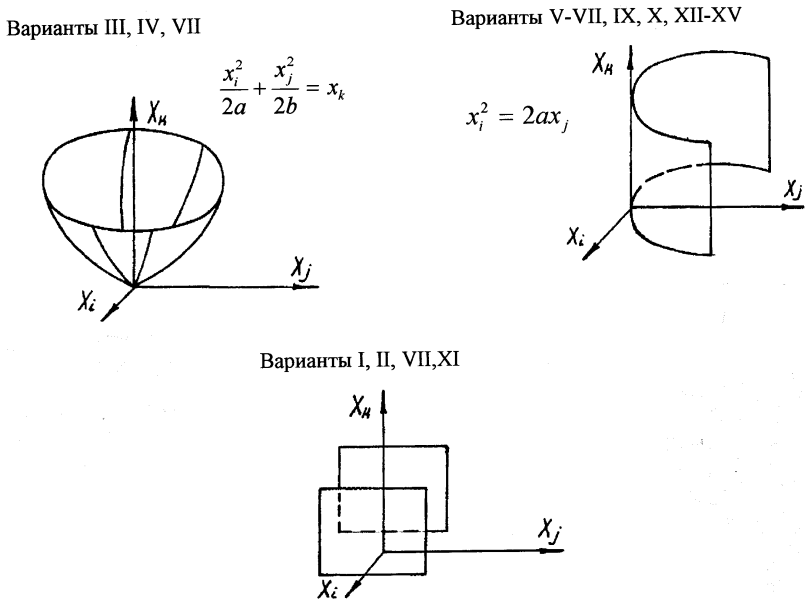
Виконуючи перетворення залежностей (1.6.27).(1.6.41) до (1.6.42), вважали, що ***x=R, y=P0, z=P***. Відповідним чином були розподілені і коефіцієнти при конкретних величинах.

Результати розрахунку по формулах (1.6.43) інваріантів і напив інваріантних також приведені в таблиці 1.6.15.

Аналіз отриманих результатів дозволив встановити, що всі геометричні образи поверхонь відгуку зводяться до еліптичного параболоїда, параболічного циліндра і пари паралельних площин (див. мал. 1.6.9).

З цього можна зробити вивід - при перетині поверхонь відгуку площинами у нас виходитимуть параболи і прямі лінії. Ця обставина враховувалася нижче при дослідженні впливу радіусу кривизни такою, що направляє,

швидкості руху нитки, натягнення веденої гілки на



Рiс.1.6.9. Геометричні образи поверхонь відгуку

натягнення ведучої гілки.

Такий диференційований підхід до вивчення регресійних залежностей (1.6.27) -(1.6.41) необхідний, оскільки велике число чинників утрудняє процес аналізу.

1.7. 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. Відшукання 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 ($x_1 \in [a,b], x_2 \in [a,b]$)

$$f(x_1) \le f(x) \le f(x_2). \qquad (1.7.1)$$

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 відрізання $[a,b]$ для яких $|x-x_1|<\varepsilon$ (або $|x-x_2|<\varepsilon$), виконується нерівність $f(x) \ge f(x_1)$ (або $f(x) \le f(x_2)$).

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 $\lim\limits_{x\to\infty} f(x) = \lim\limits_{x\to-\infty} f(x) = \infty$ те рішення задачі пошуку мінімуму (максимуму) цільової функції існує. Дане слідство можна використовувати і у разі, коли цільова функція безперервна на промені вигляду $a \le x < \infty$ або $a < x < \infty$.

Теорема Ферма [14] формулюється таким чином: якщо цільова функція диференцюєма 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 диференцюєма у внутрішніх крапках $a < x < b$ те рішення знаходиться серед критичних крапок.

Вирішуючи рівняння

$$f'(x)=0, \qquad (1.7.2)$$

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 $f(x_1,x_2,...x_n)$ декілька змінних $x_1,x_2,...x_n$ вважаючи, що вони незалежні між собою. Необхідною умовою існування екстремуму функції $f(x_1,x_2,...x_n)$ є наступне: якщо функція $f(x_1,x_2,...x_n)$ досягає екстремуму при $x_1=x_{10},x_2=x_{20},...x_n=x_{n0}$ то кожна приватна похідна першого порядку функції $f(x_1,x_2,...x_n)$ по відповідній незалежній змінній $x_1,x_2,...x_n$ звертається в нуль при цих значеннях аргументів або не існує [37]

$$\left[\frac{\partial f(x_1,x_2,...x_n)}{\partial x_1}\right]_{\substack{x_1=x_{10}\\x_2=x_{20}\\.....\\x_n=x_{n0}}}=0,$$

$$\left[\frac{\partial f(x_1,x_2,...x_n)}{\partial x_2}\right]_{\substack{x_1=x_{10}\\x_2=x_{20}\\.....\\x_n=x_{n0}}}=0, \qquad (1.7.3)$$

$$..... ....$$

$$\left[\frac{\partial f(x_1,x_2,...x_n)}{\partial x_n}\right]_{\substack{x_1=x_{10}\\x_2=x_{20}\\.....\\x_n=x_{n0}}}=0.$$

Значення $x_{10},x_{20},...x_{n0}$ називаються критичними точками функції $f(x_1,x_2,...x_n)$. 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 ($n=0,1,2,..$) і залишкового члена [14]. Хай в деякій області, що містить крапку $M_0(x_{10},x_{20},...x_{n0})$ функція $f(x_1,x_2,...x_n)$ має безперервні приватні похідні до третього порядку включно. Хай крапка $M_0(x_{10},x_{20},...x_{n0})$ є критичною крапкою для функції $f(x_1,x_2,...x_n)$ тоді

$$\left[\frac{\partial f(x_{10},x_{20},...x_{n0})}{\partial x_1}\right]=0,$$

$$\left[\frac{\partial f(x_{10},x_{20},...x_{n0})}{\partial x_2}\right]=0, \qquad (1.7.4)$$

$$..... ....$$

$$\left[\frac{\partial f(x_{10},x_{20},...x_{n0})}{\partial x_n}\right]=0.$$

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

$$\frac{\partial^2 f(x_{10},x_{20})}{\partial x_1^2}\cdot\frac{\partial^2 f(x_{10},x_{20})}{\partial x_2^2}-\left[\frac{\partial^2 f(x_{10},x_{20})}{\partial x_1\partial x_2}\right]^2>0,\ \frac{\partial^2 f(x_{10},x_{20})}{\partial x_1^2}<0,$$

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 $x_1,x_2,...x_n$ є незалежними можна говорити про відшукання 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 $x_1,x_2,...x_n$ зв'язані m $(m<n)$ рівняннями

$$\begin{aligned}&\varphi_1(x_1,x_2,...x_n)=0,\\&\varphi_2(x_1,x_2,...x_n)=0,\\&\quad\cdots\cdots\cdots\cdots\\&\varphi_m(x_1,x_2,...x_n)=0.\end{aligned}\qquad(1.7.5)$$

Необхідні умови існування екстремуму виходять шляхом визначення приватних похідних від функції

$$\begin{aligned}\Phi(x_1,x_2,...x_n,\lambda_1,...\lambda_n)&=f(x_1,x_2,...x_n)+\lambda_1\varphi_1(x_1,x_2,...x_n)+\\&+\lambda_2\varphi_2(x_1,x_2,...x_n)+...+\lambda_m\varphi_m(x_1,x_2,...x_n)=0,\end{aligned}\qquad(1.7.6)$$

по відповідних змінних $x_1,x_2,...x_n$. У виразі (1.7.6) $\lambda_1,\lambda_2,...\lambda_m$ називаються множниками Лагранжа. Прирівнюючи нулю виразу для відповідних приватних похідних, отримаємо

$$\frac{\partial f(x_1,x_2,...x_n)}{\partial x_1}+\lambda_1\frac{\partial \varphi_1(x_1,x_2,...x_n)}{\partial x_1}+\lambda_2\frac{\partial \varphi_2(x_1,x_2,...x_n)}{\partial x_1}+...+$$
$$+\lambda_m\frac{\partial \varphi_m(x_1,x_2,...x_n)}{\partial x_1}=0,$$
$$\frac{\partial f(x_1,x_2,...x_n)}{\partial x_2}+\lambda_1\frac{\partial \varphi_1(x_1,x_2,...x_n)}{\partial x_2}+\lambda_2\frac{\partial \varphi_2(x_1,x_2,...x_n)}{\partial x_2}+...+$$
$$+\lambda_m\frac{\partial \varphi_m(x_1,x_2,...x_n)}{\partial x_2}=0, \qquad (1.7.7)$$
$$.....\ .....\ .....\ .....\ .....\ .....\ .....$$
$$\frac{\partial f(x_1,x_2,...x_n)}{\partial x_n}+\lambda_1\frac{\partial \varphi_1(x_1,x_2,...x_n)}{\partial x_n}+\lambda_2\frac{\partial \varphi_2(x_1,x_2,...x_n)}{\partial x_n}+...+$$
$$+\lambda_m\frac{\partial \varphi_m(x_1,x_2,...x_n)}{\partial x_n}=0.$$

Вирішуючи спільно $m+n$ рівнянь (1.7.5), (1.7.7) визначаємо значення $x_{10},x_{20},...x_{n0}$ змінних, при яких функція $f(x_1,x_2,...x_n)$ досягає умовного екстремуму і значення множників Лагранжа $\lambda_1,\lambda_2,...\lambda_m$.

Для визначення достатніх умов існування умовного екстремуму поступають аналогічно, як і у разі незалежних змінних $x_1,x_2,...x_n$. Проте, функцію $f(x_1,x_2,...x_n)$ необхідно замінити функцією (1.7.6). Так для функції два змінних x_1,x_2 отримаємо наступні умови: функція $f(x_1,x_2)$ має відносний максимум в крапці $M_0(x_{10},x_{20})$ якщо

$$\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1^2}\cdot\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_2^2}-\left[\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1\partial x_2}\right]^2>0,\ \frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1^2}<0,$$

функція $f(x_1,x_2)$ має відносний мінімум, якщо

$$\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1^2}\cdot\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_2^2}-\left[\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1\partial x_2}\right]^2>0,\ \frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1^2}>0,$$

функція $f(x_1,x_2)$ не має ні максимуму, ні мінімуму, якщо

$$\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1^2}\cdot\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_2^2}-\left[\frac{\partial^2\Phi(x_{10},x_{20})}{\partial x_1\partial x_2}\right]^2<0,$$

функція $f(x_1,x_2)$ може мати екстремум в крапці $M_0(x_{10},x_{20})$ а може і не мати, якщо

$$\frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_1^2} \cdot \frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_2^2} - \left[\frac{\partial^2 \Phi(x_{10}, x_{20})}{\partial x_1 \partial x_2}\right]^2 = 0.$$

У разі, коли змінні зв'язані додатковими умовами $\sum_{i=1}^{m} \varphi_i(x_1, x_2, \ldots x_n) = 0$ 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 $f(x_1, x_2, \ldots x_n)$ при наступних обмеженнях

$$\sum_{i=1}^{m} \varphi_i(x_1, x_2, \ldots x_n) = 0 \quad \sum_{j=1}^{V} g_j(x_1, x_2, \ldots x_n) \geq 0$$

Достатність умов Куна-Таккера полягає в наступному. Якщо цільова функція $f(x_1, x_2, \ldots x_n)$ опукла, а всі обмеження у вигляді нерівностей $\sum_{j=1}^{V} g_j(x_1, x_2, \ldots x_n) \geq 0$ містять увігнуті функції $g_j(x_1, x_2, \ldots x_n)$ а обмеження у вигляді рівності містять лінійні функції $\varphi_i(x_1, x_2, \ldots x_n)$ то якщо існує рішення $(x_{10}, x_{20}, \ldots x_{n0}, u_1, u_2 \ldots u_V, \lambda_1, \lambda_2, \ldots \lambda_m)$ що задовольняє умовам

$$\frac{\partial f(x_1, x_2, \ldots x_n)}{\partial x_k} - \frac{\partial}{\partial x_k}\left[\sum_{k_1=1}^{V} u_{k_1} g_{k_1}(x_1, x_2, \ldots x_n)\right] - \frac{\partial}{\partial x_k}\left[\sum_{k_2=1}^{m} \lambda_{k_2} \varphi_{k_2}(x_1, x_2, \ldots x_n)\right] = 0,$$

$$k = 1,2,\ldots n, \quad k_1 = 1,2,\ldots V, \quad k_2 = 1,2,\ldots m,$$

$$\sum_{i=1}^{m} \varphi_m(x_1, x_2, \ldots x_n) = 0, \quad \sum_{j=1}^{V} g_j(x_1, x_2, \ldots x_n) \geq 0, \qquad (1.7.8)$$

$$u_{k_1} g_{k_1}(x_1, x_2, \ldots x_n) = 0, \quad u_{k_1} \geq 0,$$

то сукупність $x_{10}, x_{20}, \ldots x_{n0}$ є оптимальним рішенням задачі нелінійного програмування [38].

У разі, коли цільова функція $f(x_1, x_2, \ldots x_n)$ є лінійною функцією змінних $x_1, x_2, \ldots x_n$ а умови $\sum_{i=1}^{m} \varphi_i(x_1, x_2, \ldots x_n) = 0 \quad \sum_{j=1}^{V} g_j(x_1, x_2, \ldots x_n) \geq 0$

$f(x_1, x_2, ...x_n)$.

Класичними завданнями лінійного програмування є: транспортне завдання (необхідно так спланувати перевезення, щоб мінімізувати витрати, пов'язані з транспортуванням вантажів); завдання про використання сировини (при фіксованих запасах сировини необхідно так спланувати випуск декількох видів продукції, щоб прибуток підприємства був максимальним); завдання про використання устаткування (необхідно скласти план використання декількох видів виробничого устаткування з різною продуктивністю, щоб можна було виконати план виробництва по номенклатурі виробів); завдання проектування механізмів, пристроїв і систем механізмів (потрібно спроектувати механізм так, щоб його надійність була найбільшою, а маса, габарити і вартість менше деяких наперед заданих величин).

Суть завдань лінійного програмування полягає у відшуканні ненегативних значень змінних $x_{10}, x_{20}, ...x_{n0}$ які забезпечують досягнення найбільшого(найменшого) значення цільової функції $f(x_1, x_2, ...x_n)$. Завдання лінійного програмування можуть містити обмеження змінних $x_1, x_2, ...x_n$ у вигляді рівності. Тоді отримуємо основне завдання лінійного програмування

$$f(x_1, x_2, ...x_n) = b_{11}x_1 + b_{12}x_2 + ... + b_{1n}x_n + b_0, \qquad (1.7.9)$$

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = c_1, \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = c_2, \\ .... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = c_m, \\ m < n, \end{gathered} \qquad (1.7.10)$$

де $b_0, b_{11}, ...b_{1n}, a_{11}, ...a_{mn}, c_1, ...c_m$ - деякі постійні.

Будь-яким допустимим рішенням основної задачі лінійного програмування можна вважати сукупність змінних $x_1, x_2, ...x_n$ яка задовольняє наступним нерівностям

$$x_1 \geq 0, x_2 \geq 0, \ldots x_n \geq 0. \quad (1.7.11)$$

Крім того, необхідно, щоб вказана сукупність змінних $x_1, x_2, \ldots x_n$ задовольняла системі рівнянь (1.7.10). Оптимальним рішенням, зі всіх допустимих рішень, буде таке, при якому значення цільової функції (1.7.9) прийме мінімальне (максимальне) значення. У разі коли сукупність змінних $x_1, x_2, \ldots x_n$ задовольняє (1.7.10), але не задовольняє (1.7.11) або рівняння системи (1.7.10) суперечать один одному, то говорять, що завдання лінійного програмування не має допустимих рішень. Коли цільова функція $f(x_1, x_2, \ldots x_n)$ 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 $x_1, x_2, \ldots x_n$ задані у вигляді лінійних нерівностей, то говорять про завдання лінійного програмування з обмеженнями-нерівностями

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - c_1 \geq 0, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n - c_2 \geq 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n - c_m \geq 0. \end{aligned} \quad (1.7.12)$$

Використовуючи (1.7.12) необхідно знайти таку сукупність ненегативних значень змінних $x_1, x_2, \ldots x_n$ щоб набути мінімального (максимальне) значення цільової функції (1.7.9). Дану задачу вирішують шляхом переходу до основного завдання лінійного програмування шляхом введення нових базисних (додаткових) змінних

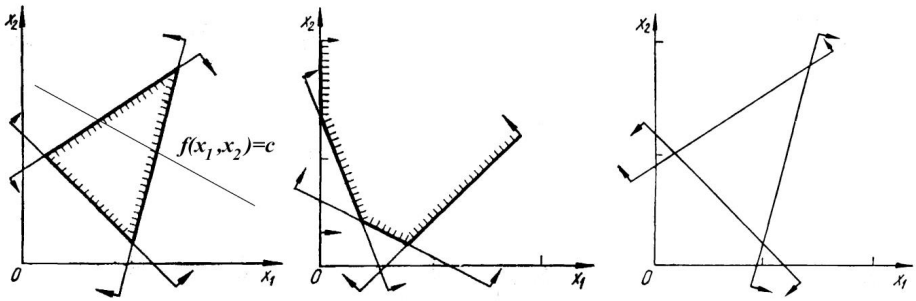
$$\begin{aligned} y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - c_1, \\ y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n - c_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m = a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n - c_m. \end{aligned} \quad (1.7.13)$$

Базисні змінні $y_1, y_2, \ldots y_m$ і початкові $x_1, x_2, \ldots x_n$ повинні задовольняти нерівностям

$$y_j \geq 0, x_k \geq 0, \quad j = 1, \ldots m, k = 1, \ldots n. \quad (1.7.14)$$

Отже, необхідно знайти ненегативні значення $n+m$ змінних $x_1, x_2, \ldots x_n$ $y_1, y_2, \ldots y_m$

Геометрична інтерпретація системи лінійних нерівностей (1.7.12) (залежно від величини коефіцієнтів $a_{11}, \ldots a_{mn}, c_1, \ldots c_m$) може бути представлена трьома варіантами (для наочності обмежимося двома змінними x_1 і x_2), які представлені на рис.1.7.1 а-в.



а б у

Ріс.1.7.1. Області вирішень системи лінійних нерівностей

Ріс.1.7.1 а показує, що областю вирішення системи нерівностей (1.7.12) є обмежене, опукле тіло (трикутник). Стрілками на малюнку вказані на пів площини, які є областями вирішень відповідних нерівностей системи (1.7.12), яка є визначеною. На рис.1.7.1 би показаний випадок, коли область вирішення системи нерівностей (1.7.12) є необмеженим, опуклим тілом. В даному випадку система нерівностей є невизначеною. На рис.1.7.1 в область вирішень системи нерівностей (1.7.12) порожня – не містить жодної крапки. Дана система нерівностей є несумісною.

Для пошуку оптимального рішення в завданні лінійного програмування розглянемо лінії рівня функції (1.7.9) на площині x_1, x_2 (см.рис.1.7.1а)

$$b_{11}x_1 + b_{12}x_2 = c. \qquad (1.7.15)$$

Рівняння (1.7.15) описує сімейство прямих, паралельних прямою

$$b_{11}x_1 + b_{12}x_2 = 0. \qquad (1.7.16)$$

Оптимальне значення (максимальне або мінімальне) цільової функції відповідатиме прямій сімейства (1.7.15), яка міститиме крайню крапку з області вирішення системи нерівностей (1.7.12). 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 $x_1, x_2, \ldots x_n$ 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 $f(x_1, x_2, \ldots x_n)$.

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

$$W = W(u). \tag{1.7.17}$$

Суть завдання оптимального управління полягає у відшуканні такого u^* яке дозволяє переводити систему S з початкового стану в кінцевий так, щоб критерій W приймав максимальне (мінімальне) значення

$$W^* = W(u^*) = \max_u [W(u)] \ . \tag{1.7.18}$$

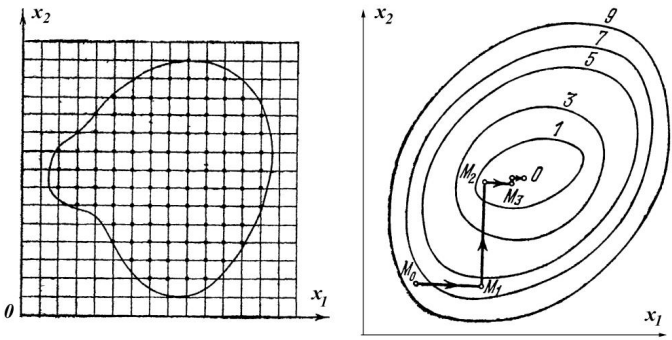
Стан системи S характеризується певною кількістю параметрів $x_1, x_2, ... x_n$ (фазовими координатами системи). Процес переходу системи з початкового положення в кінцеве розіб'ємо на певне число кроків. Після цього, на кожному з кроків, починаючи з останнього, визначимо умовне оптимальне управління. 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 відшуканні максимальної (мінімальною) точки поверхні відгуку цільової функції $f(x_1, x_2, ... x_n)$. 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

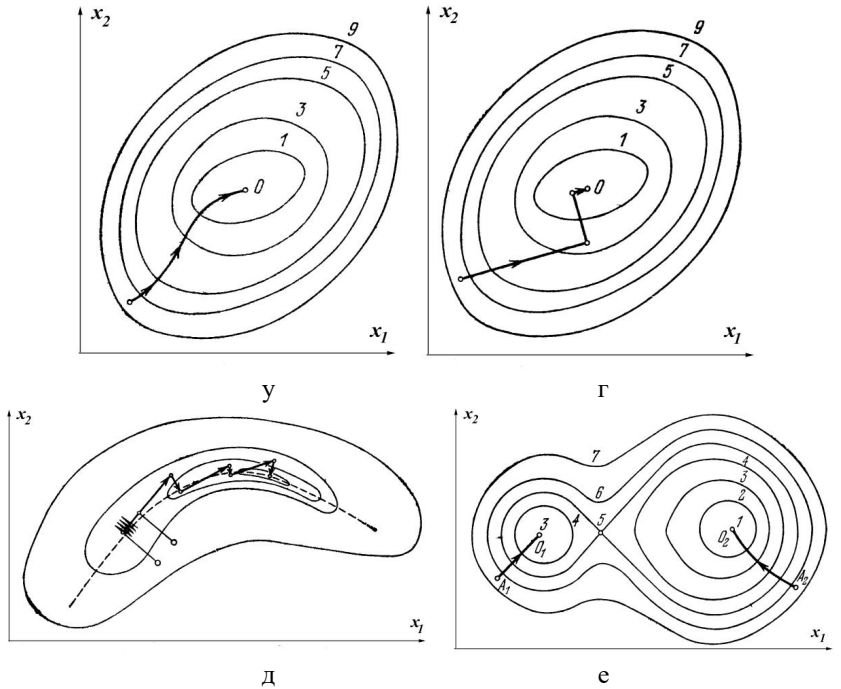
екстремуму цільової функції $f(x_1, x_2, \ldots x_n)$розглядатимемо функції двох параметрів $f(x_1, x_2)$.

На рис.1.7.2 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.1.7.2 би 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 убування (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 убування (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.



–



Рiс.1.7.2. Методи визначення екстремуму цільової функції $f(x_1, x_2)$

На рис.1.7.2 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

$$\left| grad \quad f(x_1, x_2) \right| = \sqrt{\left[\frac{\partial f(x_1, x_2)}{\partial x_1}\right]^2 + \left[\frac{\partial f(x_1, x_2)}{\partial x_2}\right]^2}. \qquad (1.7.19)$$

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 (1.7.19) вирази для градієнта на кожному кроці.

Досконалішою є методика модифікованого градієнтного методу – метод найскорішого спуску. На рис.1.7.2 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.1.7.2 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.1.7.2 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

$$S(r) = 2\pi r^2 + 2\pi rh,$$
$$l(r) = 4\pi r + h, \qquad (1.7.20)$$
$$V = \pi r^2 h,$$

де h - висота судини.

Перетворимо (1.7.20) до вигляду

$$S(r) = 2\pi r^2 + \frac{2V}{r}, \quad 0 < r < \infty,$$
$$l(r) = 4\pi r + \frac{V}{\pi r^2}, \quad 0 < r < \infty. \qquad (1.7.21)$$

Набутих значень цільових функцій показую, що вони залежать від однієї змінної r. Скориставшись (1.7.2) визначимо їх похідні і прирівняємо нулю

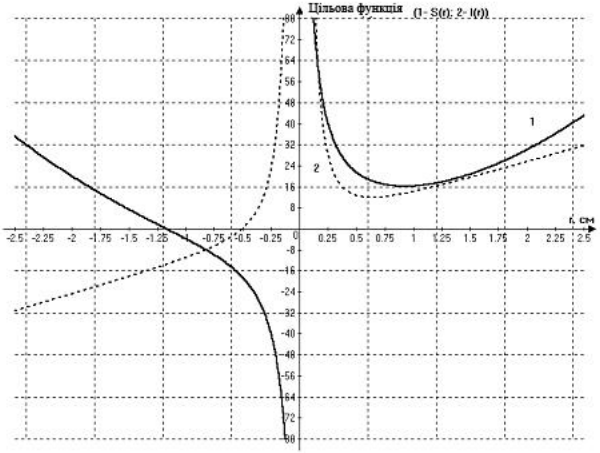
$$\frac{\partial S(r)}{\partial r} = \frac{2}{r^2}(2\pi r^3 - V) = 0,$$
$$\frac{\partial l(r)}{\partial r} = \frac{2}{\pi r^3}(2\pi^2 r^3 - V) = 0.$$

З отриманої системи рівнянь визначаємо значення радіусу циліндрової судини при якому отримаємо мінімальну площу поверхні $S(r_S)$ і мінімальну довжину швів $l(r_l)$

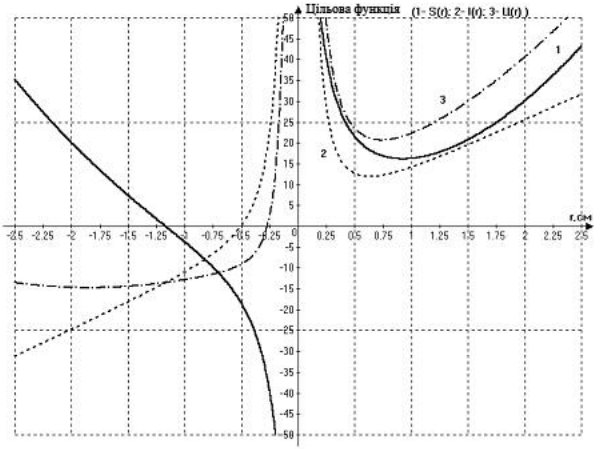
$$r_S = \sqrt[3]{\frac{V}{2\pi}}, \quad h_S = 2r_S,$$
$$r_l = \sqrt[3]{\frac{V}{2\pi^2}}, \quad h_l = 2\pi r_l. \qquad (1.7.22)$$

На рис.1.7.3 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 $Ц_1$ а ціна одиниці довжини зварного шва $Ц_2$. Тоді як цільова функція можна узяти вартість циліндрової судини $Ц$.



а



б

Рис.1.7.3. Графіки зміни цільових функцій $S(r)$, $l(r)$ и $Ц(r)$

З обліком (1.7.20), (1.7.21), отримаємо

$$Ц(r) = 2Ц_1 \pi r^2 + \frac{2Ц_1 V}{r} + 4Ц_2 \pi r + \frac{VЦ_2}{\pi r^2}, \quad 0 < r < \infty. \tag{1.7.23}$$

Визначимо першу похідну і прирівняємо її нулю

$$\frac{\partial Ц(r)}{\partial r} = 4Ц_1\pi r - \frac{2Ц_1 V}{r^2} + 4Ц_2\pi - \frac{2VЦ_2}{\pi r^3} = 0. \qquad (1.7.24)$$

Після перетворень (1.7.24) прийме вигляд

$$4Ц_1\pi^2 r^4 - 2Ц_1 V\pi r + 4Ц_2\pi^2 r^3 - 2VЦ_2 = 0.$$

Вирішуємо отримане рівняння четвертого ступеня чисельним методом (см.1.7). Для V=5см^3, $Ц_1$=0,5 ед. ціни $Ц_2$=1 ед. ціни отримаємо чотири корені: r_1=0,7364 см r_2= - 1,8472 см $r_{3,4}$= - 0,4446 ±0,418 j. Комплексне коріння відкидаємо і внаслідок того, що $0 < r < \infty$ отримуємо $r_Ц$=0,7364 див.

На рис.1.7.3 би показані графічні залежності зміни цільових функцій $S(r)$ $l(r)$

$$r_l < r_Ц < r_S.$$

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

$$Ax + By + Cz + D = 0, \qquad (1.7.25)$$

де A, B, C, D - деякі постійні.

Квадрат відстані від носика спрямовувача нитки до змінної крапки (x, y, z) виражається залежністю

$$r^2 = (x-a)^2 + (y-b)^2 + (z-c)^2. \qquad (1.7.26)$$

Вирішуємо дану задачу, з обмеженнями у вигляді рівності, з використанням множників Лагранжа. Необхідно мінімізувати вираз (1.7.26). Допоміжна функція має вигляд (з обліком (1.7.6))

$$\Phi = (x-a)^2 + (y-b)^2 + (z-c)^2 + \lambda_1(Ax + By + Cz + D).$$

Визначаємо її приватні похідні і прирівнюємо їх нулю. Тоді отримаємо

$$\begin{aligned} x &= a - \frac{1}{2}\lambda_1 A, \\ y &= b - \frac{1}{2}\lambda_1 B, \\ z &= c - \frac{1}{2}\lambda_1 C. \end{aligned} \qquad (1.7.27)$$

Підставляючи значення x, y, z у (1.7.25) визначаємо значення множника Лагранжа

$$\lambda_1 = \frac{2(Aa + Bb + Cc + D)}{A^2 + B^2 + C^2}. \qquad (1.7.28)$$

Підставляючи (1.7.27) в (1.7.26), з обліком (1.7.28), 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

$$V = xyz, \qquad (1.7.29)$$

де x, y, z - довжини сторін паралелепіпеда.

Рівняння, що зв'язує змінні x, y, z має вигляд

$$xy + xz + yz - a = 0,$$
$$0 < x < \infty, \quad 0 < y < \infty, \quad 0 < z < \infty. \qquad (1.7.30)$$

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

$$yz + \lambda(y + z) = 0,$$
$$xz + \lambda(x + z) = 0, \qquad (1.7.31)$$
$$xy + \lambda(x + y) = 0.$$

Вирішуючи спільно (1.7.30) і (1.7.31), отримаємо

$$\lambda = -\frac{3xyz}{2a}.$$

Підставляючи λ у (1.7.31), 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.

Висновки по розділу 1

З позицій розроблених наукових основ взаємодії ниток з направляючими викладені обгрунтовані технічні і технологічні рішення, реалізація яких вносить помітний внесок до науково-технічного прогресу, а саме, при вдосконаленні технологічних процесів і устаткування текстильної і легкої промисловості.

Вирішення наукової проблеми по розробці основ взаємодії ниток з направляючими при формуванні тканини і трикотажу дозволило отримати наступні основні результати:

Виконані комплексні теоретичні дослідження по:

- визначенню головних компонентів кривизни і кручення осі нитки з урахуванням зминання в зоні контакту з направляючою поверхнею довільного профілю;
- визначенню швидкості і прискорення точок осі ниток здатних до деформації в перетині при її русі по направляючій великої кривизни;
- опису динаміки процесу взаємодії здатних до деформації в перетині, жорстких на вигин ниток з направляючою великої кривизни з урахуванням і без урахування її деформації.

Розробка основ механіки здатної до деформації в перетині, жорсткої на вигин нитки дозволила вирішити ряд конкретних завдань механіки нитки:

- отримати вирази для визначення форми осі і натягнення нитки у разі подовжнього і поперечного ковзання по напрямній з урахуванням сили тяжіння, коефіцієнта жорсткості на вигин;
- у полі сили тяжіння визначити форму рівноваги жорстких на вигин ниток з двома точками закріплення;

- описати процес взаємодії здатних до деформації в перетині, жорстких на вигин ниток з направляючими великої кривизни;
- встановити вплив форми направляючої поверхні на натяг мононіток, комплексних ниток і пряжі.
- здійснити вибір критеріїв оптимізації розмірів спрямовувачів нитки різних трикотажних машин;

Для реалізації планів експериментальних досліджень і перевірки правильності і коректності зроблених допущень при теоретичних дослідженнях:

- встановлений зв'язок між умовами взаємодії ниток з направляючою і їх питомим розривним навантаженням і відносним розривним подовженням після проходження останньої;
- запропоновані принципово нові способи визначення натягу ниток при взаємодії з приладами для натягу нитки, що дозволило розробити рекомендації для їх істотної модернізації.

Реалізація комплексних теоретико-експериментальных досліджень дозволила отримати наступні практичні результати:

- виходячи з реальних умов взаємодії ниток з направляючими, оптимізувати геометричні розміри останніх для кожного конкретного виду ниток;
- отримати пакети прикладних програм для визначення технологічних зусиль при переробці ниток на технологічному устаткуванні, регресійні залежності, нові методи визначення фізико-механічних властивостей тканин і ниток з них можуть бути використані на машинобудівних підприємствах, фабриках, НДІ даного профілю для модернізації існуючого устаткування і розробки нового;

До найбільш перспективних питань даного наукового напряму можна віднести наступні:

- визначення впливу будови нитки на умови її взаємодії з направляючими поверхнями великої кривизни довільного профілю;
- вивчення взаємодії ниток з направляючою поверхнею, що деформується, з урахуванням анізотропії физико-механических властивостей як нитки, так і направляючої;
- розробка універсальних вимірювальних комплексів для дослідження взаємодії ниток з спрямовувачами і натягувачами нитки сучасного технологічного устаткування;
- вдосконалення конструкції спрямовувачів і натягувачів нитки трикотажних машин і, як наслідок, вдосконалення системи подачі нитки на основі стабілізації і мінімізації натягу нитки перед зоною в’язання.

На основі аналізу структурних схем лінії заправки нитки круглов’язальних машин 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.

На основі реалізації розробленого алгоритму вперше вирішена плоска та просторова задача синтезу системи подачі нитки на круглов’язальних машинах для випадку перешкод у вигляді вертикальних ліній та кіл, та розроблено програмне забезпечення для пошуку оптимальної форми заправки нитки на круглов’язальній машині.

На основі розв’язання системи диференційних рівнянь, яка описує умови взаємодії жорсткої на згин, зминаємої нитки в зоні контакту з поверхнею фідера та ниткоспрямовувача, вперше отримані залежності вихідного натягу та кута охоплення від фізико-механічних властивостей комплексних ниток і пряжі та конструктивних параметрів конструктивних елементів системи подачі нитки.

В процесі формалізації критерію оптимізації конструктивних параметрів ниткоспрямовувача вперше 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.

Досліджено вплив конструктивних параметрів фідерних та кільцевих спрямовувачів нитки та на її натяг, при цьому отримані відповідні регресійні залежності натягу комплексних ниток та пряжі як функцій від внутрішнього радіусу фідера, радіусу зовнішньої кривизни фідера та спрямовувача, натягу веденої гілки, кута охоплення ниткою фідера, побудовані відповідні поверхні відгуку, проведена оцінка значущості коефіцієнтів в рівняннях регресії з використанням спеціально розробленого програмного забезпечення.

Список використаних джерел до розділу 1

1. 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.

2. 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.

3. Ресурсоощадні технології та обладнання швейної та текстильної промисловості: монографія: в 2 ч. Ч.1/Наукові основи та інженерні методи проектування ресурсоощадних технологій і обладнання швейної та

текстильної промисловості/ В.Ю.Щербань, Г.Б.Параска, Б.В.Орловський та ін. – К.:КНУТД, 2015. – 339 с.

4. Ресурсоощадні технології та обладнання швейної та текстильної промисловості: монографія: в 2 ч. Ч.2/Шляхи підвищення ефективності швейної та текстильної галузей України на базі новітних технологій та управління/ В.Ю.Щербань, Г.Б.Параска, Б.В.Орловський та ін. – К.:КНУТД, 2015. – 270 с.

5. 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.

6. Щербань В. Ю.Інформаційні технології в науці, виробництві та підприємництві/В.Ю.Щербань.-К.:КНУТД, 2016. – 184 с.

7. Щербань В. Ю.Інформаційні технології в науці, виробництві та підприємництві/В.Ю.Щербань.-К.:Освіта України, 2017. – 238 с.

8. 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.

9. 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.

10. Щербань В.Ю. Математичні та програмні компоненти САПР технологічних процесів та обладнання текстильної та взуттєвої галузі/ В.Ю.Щербань, О.З.Колиско, М.І.Шолудько, В.Ю.Калашник. – К.:Бумсервіс, 2016. – 588 с.

11. Слізков А.М., Щербань В.Ю., Кизимчук О.П. Механічна технологія текстильних матеріалів. Частина ІІ. (Ткацьке, трикотажне та неткане виробництво): підручник / А.М.Слізков, В.Ю.Щербань, О.П.Кизимчук. – К.:КНУТД, 2018. – 276 с.

12. Щербань В.Ю. Механіка нитки/В.Ю.Щербань. – К.:Видавництво «Укрбланковидав». – 2018. – 533 с.

13. Щербань В. Ю.Інформаційні технології в науці, виробництві та підприємництві/В.Ю.Щербань.-К.:Освіта України, 2018. – 257 с.

14. Щербань В.Ю. Базове проектуюче забезпечення САПР в індустрії моди/ В.Ю.Щербань, Ю.Ю.Щербань, О.З.Колиско, Г.В.Мельник, М.І.Шолудько, В.Ю.Калашник. – К.:Освіта України, 2018. – 902 с. 15. Щербань В. Ю. Інформаційні технології в науці, виробництві та підприємництві/В.Ю.Щербань.-К.:Освіта України, 2019. – 252 с.

16. Щербань В.Ю. Комп'ютерне проектування систем: програмні та алгоритмічні компоненти / В.Ю.Щербань, О.З.Колиско, Г.В.Мельник, М.І.Шолудько, В.Ю.Калашник. – К.:Освіта України, 2019. – 902 с.

17. Щербань В. Ю. Інформаційні технології в науці, виробництві та підприємництві/В.Ю.Щербань – К.:Освіта України: ФОП Маслаков, 2020. – 236 с.

18. Щербань В.Ю. САПР складних систем: математичні, алгоритмічні та комп'ютерні програмні компоненти /В.Ю.Щербань, О.З.Колиско, Ю.Ю.Щербань, Г.В.Мельник, М.І.Шолудько, В.Ю.Калашник. – К.:Освіта України: ФОП Маслаков, 2020. – 923 с.

19. Shcherban' V. Yu. Mathematical software models for determining technological efforts in the production of technical fabrics and knitwear for military needs / V. Yu. Shcherban', L.E. Halavska, O.Z. Kolysko, Yu.Yu. Shcherban', T.V. Ielina, M.I. Kolysko. – K.: Education of Ukraine, 2021. – 148 p.

20. Щербань В. Ю. Інформаційні технології в науці, виробництві та підприємництві / В.Ю.Щербань – К.:Освіта України, 2021. – 248 с.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. Shcherban' V.Yu. Interaction of stiff yarns with the working parts of knitting and sewing machines/V.Yu.Shcherban' // Textile industry. -1988. - № 10. - pp.53.

31. Vasil'chenko V.N., Shcherban' V.Yu., Apokin Ts.V. Attachment for holding multilayer fabrics in the clamps of a universal tensile tester/ V.N.Vasil'chenko , V.Yu.Shcherban' , Ts.V.Apokin // Textile industry. – 1987. - №8. - pp.62.

32. 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.97-104.

33. Shcherban' V., Melnyk G. , Sholudko M., Kolysko O. and Kalashnyk V. Yarn tension while knitting textile fabric/V.Shcherban' , G. Melnyk , M.Sholudko , O.Kolysko, V.Kalashnyk// Fibres and Textiles. – 2018. - volume 25. - №3. - pp. 74-83.

34. Shcherban' V., Melnyk G. , Sholudko M., Kolysko O. and Kalashnyk V. Improvement of structure and technology of manufacture of multilayer technical fabric/V.Shcherban' , G. Melnyk , M.Sholudko , O.Kolysko, V.Kalashnyk// Fibres and Textiles. – 2019. - volume 26 - № 2 - pp. 54-63.

35. Shcherban' V., Korogod G., Chaban V., Kolysko O., Shcherban' Yu., Shchutska G. Computer simulation methods of redundant measurements with the nonlinear transformation function / V. Shcherban', G. Korogod, V. Chaban, O. Kolysko, Yu. Shcherban', G. Shchutska // Eastern-European Journal of Enterprise Technologies. - 2019. - volume 98. -№2/5. – pp.16-22.

36. Shcherban' V., Makarenko J., Melnyk G., Shcherban' Y., Petko A., Kirichenko A. Effect of the yarn structure on the tension degree when interacting with high-curved guides/ V. Shcherban', J. Makarenko, G. Melnyk, Y. Shcherban', A. Petko, A. Kirichenko // Fibres and Textiles. – 2019. - volume 26 - № 4 - pp. 59-68.

37. Shcherban' V., Makarenko J., Petko A., Melnyk G., Shcherban' Yu., Shchutska G. Computer implementation of a recursion algorithm for

determining the tension of a thread on technological equipment based on the derived mathematical dependences / V.Shcherban', J.Makarenko, A.Petko, G.Melnyk, Yu.Shcherban', G.Shchutska // Eastern-European Journal of Enterprise Technologies. - 2020. - volume 104. -№2/1. – pp.41-50.

38. Shcherban' V., Kolysko O., Melnyk G., Sholudko M., Shcherban' Y. and Shchutska G. Determining tension of yarns when interacting with guides and operative parts of textile machinery having the torus form / V. Shcherban', O. Kolysko, G. Melnyk, M. Sholudko, Y. Shcherban' and G. Shchutska // Fibres and Textiles. – 2020. - volume 27 - № 4 - pp. 87-95.

39. Shcherban' V., Korogod G., Kolysko O., Kolysko M., Shcherban' Yu., Shchutska G. Computer simulation of multiple measurements of logarithmic transformation function by two approaches / V. Shcherban', G. Korogod, O. Kolysko, M. Kolysko, Yu. Shcherban', G. Shchutska // Eastern-European Journal of Enterprise Technologies. - 2020. - volume 6. -№4 (108). – pp. 6-13.

40. Shcherban' V., Korogod G., Kolysko O., Kolysko M., Shcherban' Yu., Shchutska G. Computer simulation of logarithmic transformation function to expand the range of high-precision measurements / V. Shcherban', G. Korogod, O. Kolysko, M. Kolysko, Yu. Shcherban', G. Shchutska // Eastern-European Journal of Enterprise Technologies. - 2021. - volume 2. -№9 (110). – pp. 27-36.

41. Shcherban' V., Kolysko O., Melnyk G., Sholudko M., Shcherban' Yu., Shchutska G. and Kolva N. Determination of tension for polyamide and basalt multifilament yarns while weaving industrial fabrics / V. Shcherban', O. Kolysko, G. Melnyk, M. Sholudko, Yu. Shcherban', G. Shchutska, N. Kolva // Fibres and Textiles. – 2021. - volume 28 - № 1 - pp. 75-85.

42. Shcherban' V., Kolysko O., Melnyk G., Kolysko M., Halavska L., Shcherban' Y. The influence of the curvature radius of the guiding surface on the tension of polyethylene and polyamide complex yarns during processing on weaving and knitting machines / V. Shcherban', O. Kolysko, G. Melnyk, M.

Kolysko, L. Halavska, Y. Shcherban' // Fibres and Textiles. – 2021. - volume 28 - № 3 - pp. 72-81.

43. 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.

44. 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.

45. 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.

46. 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.

47. Shcherban' V. Y., Korogod G. A., Kolysko O. Z., Sholudko M. I., Melnik G. V., Chaban V.V., Shcherban' Y. Y. Computer Implementation of Algorithmic Components of Redundant Measurement Methods/ Vladimir Y. Shcherban', Ganna A. Korogod, Oksana Z. Kolysko, Mariana I. Sholudko, Gennady V. Melnik, Vitaliy.V. Chaban, Yury Y. Shcherban'// Hong Kong. International Journal of Intelligent Systems and Applications (IJISA). – 2020. - Volume 12. - №1 - pp. 23-32.

48. Shcherban V., Petko A., Makarenko Ju., Kolva N., Egorov D. Computer Implementation Of The Dichotomy Algorithm For Transcendent Equations When Determining The Thread Tension// Intellectual Archive, Toronto: Shiny

World Corp., Richmond Hill, Ontario, Canada. – January - March 2020. – Volume 9.- Number 1. – P. 120-125. DOI 10.32370/IA_2020_01_7

49. Shcherban V., Kolva N., Egorov D., Petko A., Makarenko Ju. Algorithmic and Computer Software for Determination of Thread Tension After Guide Large Curvacity // Intellectual Archive, Toronto: Shiny World Corp., Richmond Hill, Ontario, Canada. – January - March 2021. – Volume 10.- Number 1. – P. 15-20.

50. 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.

51. 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.

52. 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.

53. 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.

54. 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.

55. Колиско М.І. Дослідження взаємодії нитки з циліндричною напрямною поверхнею/ М.І.Колиско, В.Ю.Щербань// Вісник КНУТД. – 2013. - №5. – С44-47.

56. Колиско М.І. Визначення впливу зминання 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.

57. 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.

58. 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.

59. Щербань В.Ю. Визначення приведеного коефіцієнту тертя для кільцевих та трубчатих спрямовувачів нитки трикотажних машин/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько//Вісник ХНУ.- 2017.-№6(255). - С.23-27.

60. Щербань В.Ю. Визначення натягу нитки при її взаємодії з трубчастими спрямовувачами/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько//Вісник ХНУ.- 2018.-№1 (257). - С.213-217.

61. Щербань В.Ю. Використання рекурсивного підходу для визначення натягу ниток в робочій зоні технологічного обладнання/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, Г.В. Мельник, М.І.Шолудько//Вісник ХНУ.- 2018.-№ 3(261). - С.7-11.

62. Щербань В.Ю. Взаємодія текстильних ниток з напрямними великої кривини у випадку наявності радіального охоплення/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, Г.В. Мельник, М.І.Шолудько// Вісник Хмельницького національного університету.- 2018.-№ 2 (259). - С.12-16.

63. Щербань В.Ю. Удосконалення системи подачі ниток на основі оптимізації пружної системи заправки круглов'язальних машин/В.Ю.Щербань, Г.В. Мельник, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько // Вісник Хмельницького національного університету.- 2018.-№ 4 (263). - С.11-16.

64. Щербань В.Ю. Структура комп'ютерної програми реалізації алгоритму

рекурсії для визначення технологічних зусиль/В.Ю. Щербань, А.К. Петко, О.З. Колиско, Ю.Ю. Щербань, М.І. Шолудько// Вісник Хмельницького національного університету.- 2020.-№ 1 (281). - С.249-253.

65. Первая Н.В., Андреєва О.А., Щербань В.Ю. Дослідження технологічних параметрів процесу формування верху взуття / Н.В. Первая, О.А. Андреєва, В.Ю. Щербань// Вісник Хмельницького національного університету.- 2020.-№ 1 (281). - С.175-181.

66. Щербань В.Ю. Програмні модулі комп’ютерної програми реалізації алгоритму рекурсії для випадку змінного вхідного натягу/ В.Ю.Щербань, А.К.Петко, О.З.Колиско, Ю.Ю.Щербань, М.І.Шолудько// Вісник Хмельницького національного університету.- 2020.-№ 2 (283). - С.213-218.

67. Щербань В.Ю., Петко А.К., Колиско О.З., Щербань Ю.Ю., Шолудько М.І. Комп’ютерна реалізація алгоритму рекурсії для випадку змінного діаметру сировини / В.Ю.Щербань, А.К.Петко, О.З.Колиско, Ю.Ю.Щербань, М.І.Шолудько // Вісник Хмельницького національного університету.- 2020, № 3(285). – С.263-267.

68. Щербань В.Ю., Колиско О.З., Щербань Ю.Ю., Шолудько М.І., Мельник Г.В. Алгоритмічні та програмні компоненти при комп’ютерному визначенні натягу для шайбового натягувача з використанням рекурсії / В.Ю.Щербань, О.З.Колиско, Ю.Ю.Щербань, М.І.Шолудько, Г.В.Мельник // Вісник Хмельницького національного університету. Том 1.- 2020, № 4(287).– С.252-256.

69. Щербань В.Ю., Колиско О.З., Щербань Ю.Ю., Шолудько М.І., Мельник Г.В. Структура програмних модулів та процедур комп’ютерної програми для основних елементів системи при реалізації алгоритму рекурсії / В.Ю.Щербань, О.З.Колиско, Ю.Ю.Щербань, М.І.Шолудько, Г.В.Мельник// Вісник Хмельницького національного університету. - 2020, № 5(289).– С.302-306.

70. Щербань В.Ю., Петко А.К., Колиско О.З., Щербань Ю.Ю., Колиско М.І. База фрикційних властивостей комп’ютерної програми для визначення натягу нитки при реалізації алгоритму рекурсії / В.Ю.Щербань, А.К.Петко, О.З.Колиско, Ю.Ю.Щербань, М.І.Колиско // Вісник Хмельницького національного університету. - 2021, № 1(293).– С.234-237.

71. Щербань В.Ю., Петко А.К., Колиско О.З., Щербань Ю.Ю., Галавська Л.Є. Програмні модулі та процедури комп’ютерної програми для визначення натягу кевларової нитки при в’язанні з використанням алгоритму рекурсії / В.Ю.Щербань, А.К.Петко, О.З.Колиско, Ю.Ю.Щербань, Л.Є. Галавська // Вісник Хмельницького національного університету. - 2021, № 2(295).– С.271-274.

72. Щербань В.Ю., Макаренко Ю.В., Колиско О.З., Щербань Ю.Ю., Галавська Л.Є. Комп’ютерна реалізація алгоритму рекурсії при визначенні натягу ниток при формуванні багатошарових тканин з поліетиленових ниток / В.Ю. Щербань, Ю.В. Макаренко, О.З. Колиско, Ю.Ю. Щербань, Л.Є.Галавська // Вісник Хмельницького національного університету. – 2021, №3 (297). – С. 204-207.

2. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ПО ПІДВИЩЕННЮ ТОЧНОСТІ ВИМІРЮВАННЯ

2.1.1. АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

На вирішення задач по підвищенню точності вимірювань і по вдосконаленню методів вимірювання спрямовані роботи багатьох вчених і науковців.

Так, в роботі [1] наведені результати дослідження можливості розширення діапазону. Показано, що шляхом використання запрограмованих коефіцієнтів підсилення та синхронних детекторів досягалося розширення смуги пропускання сигналу фотодіоду. Дослідженнями оптимізаційних процесів фотодіодів, описаних в роботі [2], встановлено, що завдяки введенню органічного прошарку досягалася зміна фотоелектричних параметрів фотодіода. Однак слід зазначити, що в даних роботах не були розглянуті питання підвищення точності вимірювань, що може викликати труднощі з отриманням достовірної інформації. Однією з причин цього є недостатня чутливість фотоприймача. Ще одним варіантом підвищення точності, який був представлений в роботі [3], є використання калібрувального коефіцієнта. До того ж, підвищення точності вимірювання може бути досягнуто і за рахунок статистичної обробки багатократних вимірювань, як представлено в роботі [4]. Але невирішеними в цих роботах залишилися питання, що пов’язані з відхиленням параметрів функції перетворення від номінальних значень під впливом дестабілізуючих факторів. Варіант подолання цих труднощів було розглянуто в роботі [5]. Показано, що завдяки обчисленню частотної характеристики і щільності фотоструму стає можливим прогнозування спектрального відгуку. Але, не зважаючи на позитивні результати, залишилися невирішеними питання, що пов’язані з отриманням високоточного результату вимірювання при нелінійній і нестабільній

функції перетворення сенсора з можливістю проведення метрологічного самоконтролю. Причиною цього є те, що в зазначених роботах не наведена можливість досягнення незалежності отриманого результату від відхилень параметрів фотоприймачів. З практичної точки зору це може викликати додаткові похибки, які пов'язані з відхиленням параметрів функції перетворення від номінальних значень. Дана обставина пов'язана з тим, що на сенсор чинить вплив як оточуюче середовище, так і старіння самого матеріалу сенсора. В результаті чого істотно змінюються показання вимірювального приладу і інформація стає недостовірною. Слід зазначити, що проблема достовірності отриманого результату була розглянута в роботі [6], де за рахунок введення надлишковості досягалося підвищення надійності отриманого результату.

Так, на вирішення задач по підвищенню точності вимірювання і покращенню технологічних характеристик направлено багато наукових праць і технічних рішень, наприклад таких, як зазначено в роботі [7]. Дослідженнями, що були проведені в роботі [8], підвищення точності досягалося при застосуванні адитивного і мультиплікативного тестів. Показано, що адитивний тест реалізується за рахунок ведення зразкової міри, а мультиплікативний – за рахунок зразкової зміни чутливості вимірювального каналу. Однак залишилися невирішеними питання, залежності результату вимірювання від нестабільності параметрів функції перетворення, а також не були наведені алгоритми обробки результатів багатократних вимірювань. Причиною виникнення похибок від нестабільності параметрів функції перетворення може слугувати як вплив навколишнього середовища, так і старіння самого матеріалу. В роботі [9] підвищення чутливості і точності вимірювання досягалося завдяки зменшенню впливу темнового струму і компенсації власних шумів вимірювального каналу радіометричного вимірювача. Показано, що, за рахунок комутації сигналів опорного (затемненого) і вимірювального

(відкритого) фотодіодів та відповідній їх обробці, виключаються власні шуми фотоприймача і зменшується вплив їх темнових струмів. Питання підвищення точності вимірювання за рахунок збільшення чутливості фотоприймача також було розглянуто в роботі [10]. Однак слід зазначити, що в даних роботах не враховувався вплив зміни параметрів функції перетворення на результат вимірювань, що, в свою чергу, може привести до появи додаткової складової похибки. Дана обставина пов’язана з впливом навколишнього середовища і впливом випадкової складової похибки на результат вимірювання. Для подолання цієї проблеми в роботах [11, 12] були наведені відповідні алгоритм обчислювань, в результаті яких досягалося покращення вихідного сигналу сенсора при різних умова. Однак в даній роботі не визначено, як саме проводити обчислення при нелінійній функції перетворення з нестабільними параметрами, що спричиняють додаткові похибки вимірювання. Теоретичні аспекти систематичної і випадкової складових похибки вимірювань та деякі шляхи їх зменшення були розглянуті в роботі [13]. У дослідженнях показано, що мінімізацію систематичної складової похибки пропонувалося досягти шляхом розробки і формування відповідних вимог та заходів щодо зменшення інструментальної, методичної та суб'єктивної складової. Незважаючи на значущість таких підходів, не розглянуто в достатній мірі конкретні шляхи для їх реалізації. Основні фактори невизначеності та їх оцінка були розглянути в роботі [14]. Показано, що завдяки оцінці цих факторів стає можливим отримання більш точних результатів вимірювання оптичного випромінювання, або, як заначено в [15], і іншого контрольованого параметру. Однак слід зазначити, що в даних роботах не наведені шляхи зменшення невизначеності при багатократних вимірювань. Для подолання цієї проблеми в роботі [16] використовується об'єднання даних в багатовимірному датчику. Показано, що такий підхід дозволяє підвищити точність та зменшити сенсорну невизначеність. Однак у даній роботі не

були розглянуті питання безпосередньої роботи з нелінійної функцією перетворення. Дана обставина пов’язана з тим, що лінеаризація нелінійної функції перетворення вносить додаткові похибки або потребує роботи на лінійних ділянках вхідної характеристики сенсора, що призводить до звуження діапазону вимірювання. Питання підвищення точності за рахунок зменшення впливу нелінійності був розглянутий в роботі [17]. Тут шляхом калібрування при декількох еталонних температурах досягалося зменшення кривизни поправки і таким чином підвищення точності. Однак тут не розглядається виключення впливу на результат вимірювання систематичної складової похибки, викликаної нестабільністю параметрів функції перетворення під впливом зовнішніх факторів.

Так в роботі [18] були наведені основні рекомендації для зменшення впливу додаткових похибок, що впливають на результат вимірювання, а також встановлені межі їх значень. В роботі [19] підвищення точності відбувалося шляхом введення поправок, які виключають систематичні похибки вимірювального каналу. Однак дана методика передбачає необхідність вхідної інформації у вигляді параметрів каналу, які необхідно визначити рекомендованим чином. В роботі [20] підвищення точності вимірювання температури досягалося за рахунок зменшення методичної похибки від тепловідведення та високої точності градуювальної характеристики. Однак слід зазначити, що в даній роботі значення температури градуювальної характеристики прив’язані до інтерференційних мінімумів двозаломлення чутливого елемента термоперетворювача. Це означає, що у випадку його заміни необхідно підбирати ідентичні термоперетворювачі, що викликає певні труднощі при налагодженні чи проведенні додаткового калібрування. Крім цього, не було зазначено вплив темнового струму. В роботі [21] підвищення точності вимірювання квадратним детектором InGaAs досягалося шляхом поєднання методу нескінченного інтеграла з методом Больцмана, що

мають протилежні характеристики похибки. В роботі [22] для зменшення перехресних завад було запропоновано рекурсивний метод, заснований на отриманні індивідуальних значень опорів резисторів в масиві резистивних датчиків. Однак, в даних роботах не наведені шляхи розширення діапазону при нелінійній функції перетворення. Для розширення діапазону і зменшення шуму в роботі [23] було запропоноване використання запрограмованих коефіцієнтів підсилення в підсилювачі при фотодіоді. В роботі [24] для цього був запропонований покращений алгоритм, який розширює лінійний діапазон вимірювання без зниження точності вимірювання. Питаннями підвищення точності за рахунок підвищення чутливості були розглянуті в роботі [25]. В даній роботі застосування методу фазочутливого виявлення для кремнієвих лавинних фотодіодів дозволило покращити відношення сигнал/шум, а також розширити діапазон вимірювання в сторону його початкових значень. Підвищення чутливості в роботі [26] досягалося завдяки використання аргонового лазера з заданою потужністю, а в роботі [27] – завдяки використання фотодетектору на основі гетеропереходу графен/кремній (G/Si). Однак тут не враховувалися нелінійність функції перетворення фотоперетворювача, що, в свою чергу, спричиняє появу похибки від нелінійності або потребує роботи на лінійних ділянках вхідної характеристики сенсора. В роботі [28] було запропоновано застосування методу адаптивного нейроно-невизначеного припущення (ANIIS), завдяки якому стає можливим підвищити точність вимірювання за рахунок встановлення високої здатності до нелінійної апроксимації. Але залишилися невирішеними питання, пов'язані з нестабільністю параметрів фотодіоду, а також залежність від таблиць відповідності для кожного окремого датчика. В роботі [29] підвищення точності датчика з нелінійною функцією досягалося шляхом лінеаризації кривої за рахунок використання комплементарного металооксидного напівпровідника.

Для комплексного вирішення поставленої задачі щодо зменшення похибки від нелінійності і нестабільності функції перетворення в даній роботі запропоновано використання методів надлишкових вимірювань [30, 31]. Це стає можливим завдяки застосуванню представленій математичній моделі, в якій, завдяки обробці отриманих результатів згідно з рівнянь надлишкових вимірювань, виключається вплив адитивної і мультиплікативної складових похибки вимірювання.

Оскільки методи надлишкових вимірювань (МНВ) зарекомендували себе як високоефективні методи, тому є підстави провести дослідження представлених методів вимірювання (в тому числі і багаторазових) при різних видах функції перетворення.

2.1.2. СУТЬ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ

Ненадлишкові методи вимірювання — методи опосередкованого визначення контрольованої вхідної величини за результатами вимірювань інших величин (або знання їх значень), з якими інформативна вхідна фізична величини x пов'язана функціональною залежністю виду:

$$y_x = f(x, k_x, S'_{л1}, S'_{н1}, \Delta_a), \tag{2.1.1}$$

де $S_{лі}$ і $S_{ні}$ — крутості перетворення лінійної і нелінійної складових функції перетворення сенсора (мультиплікативна складова похибки);

Δ_a — параметр функції перетворення з врахуванням адитивної складової похибки вимірювання.

Тобто, при вимірюванні інформативної вхідної фізичної величини ненадлишковим методом виникає ряд задач з підвищення точності вимірювання через наявність похибок від нелінійності і похибок, що

обумовлені відхиленням параметрів функції перетворення від номінальних значень. Все це впливає на якість вимірювання, яка є однією з проблем вимірювання. Зазначимо, що на даний час існують наступні проблеми вимірювання: забезпечення єдності вимірювань, підвищення якості вимірювань, проблеми метрологічної надійності вимірювального каналу і проблеми створення мір [32].

В даній роботі розглядалися такі характеристики якості вимірювання як точність. Як відомо, точність характеризується правильністю (систематична складова похибки близька до нуля) і прецизійністю (випадкова складова похибки близька до нуля).

На підвищення точності вимірювання направлені методи надлишкових вимірювань. Теоретичні основи надлишковості базуються на таких принципах, як принцип спадкоємності знань, принцип інваріантності (всі вимірювальні перетворення відбуваються за допомогою одного вимірювального каналу послідовно у часі), принцип симетрії, принцип автоматичного виключення систематичної складової похибки, що обумовлені відхиленням параметрів функції перетворення від номінальних значень, тощо.

Методи надлишкових вимірювань — це методи, засновані на виконанні кінцевої сукупності вимірювальних перетворень інформативної вхідної величини (від об’єкта дослідження) і декількох нормованих за значенням величини, при незмінних і дискретно змінених на нормовані значення параметрах функції перетворення сенсора з наступним визначенням значення інформативної вхідної величини за рівнянням надлишкових вимірювань, яке одержують в результаті рішення системи рівнянь величин [33]. Для прикладу, при нелінійній функції перетворення це може бути записано в такий спосіб:

$$\begin{cases} y'_{н1} = f(x_1, S'_н, S'_л) + \Delta y'; \\ y'_{н2} = f(x_2, S'_н, S'_л) + \Delta y'; \\ y'_{н3} = f(x_3, S'_н, S'_л) + \Delta y'; \\ y'_{н4} = f(x, x_2, S'_н, S'_л) + \Delta y'; \\ y'_{н5} = f(x, x_3, S'_н, S'_л) + \Delta y', \end{cases} \quad (2.1.2)$$

$$x = F(y'_{н1}, ..., y'_{н5}, x_2, x_3), \quad (2.1.3)$$

де $y'_{нi}$ — вихідні сигнали сенсора (i=5);

x— інформативна вхідна величина від об'єкта дослідження;

$x_1, ..., x_3$ — нормованих за значенням величини, розміри яких зв'язані між собою за певним законом (арифметичної прогресії) від об'єкта дослідження так і від джерела з нормованими характеристиками;

$S'_н$, $S'_л$ — чутливість (крутість) перетворення нелінійної і лінійної складової функції перетворення, де $S'_н = S_н(1 + \Delta S_н + \Delta_S(t))$ *і* $S'_л = S_л(1 + \Delta S_л + \Delta_S(t))$, в яких $S_н$ *і* $S_л$ — ідеальні значення крутості перетворення нелінійної і лінійної складової функції перетворення, $\Delta S_л, \Delta S_н$ — значення, що обумовлюють систематичну складову похибки, $\Delta_S(t)$ — значення, що обумовлюють випадкову складову похибки;

$\Delta y'$ — параметр (зміщення) функції перетворення з врахуванням адитивної складової похибки ($\Delta U' = \Delta U + \Delta_a + \Delta\gamma_t$, де ΔU — номінальне значення параметру функції перетворення, $\{\Delta_a\}$— адитивна складова похибки, $\{\Delta\gamma_t\}$ — випадкова складова похибки).

Таким чином, на відміну від опосередкованих методів вимірювань, методи надлишкових вимірювань відрізняються складанням системи нелінійних рівнянь величин, кількість яких визначається кількістю

змінних. Причому розв’язання таких систем можливо тільки при відомому виді функції перетворення сенсора та вибору декількох рядів фізичних величин, розміри яких зв’язаних між собою за певними законами. Як буде показано нижче, завдяки обробці отриманих результатів за рівнянням надлишкових вимірювань забезпечується автоматичне виключення систематичних складових похибок результату вимірювання, що обумовлені зміною параметрів функції перетворення під дією дестабілізуючих факторів, а при використанні багаторазових вимірювань і статистичній обробці — зменшити і випадкові складові похибки.

Сутність методів надлишкових вимірювань полягає у вимірювальному перетворенні не одного, а декількох фізичних величин, значення яких пов’язані між собою за законом арифметичної прогресії. Це дає можливість скласти систему з n нелінійних рівнянь величин з подальшим її розв’язком з метою отримання рівняння надлишкових вимірювань.

2.1.3. УЗАГАЛЬНЕНІ МАТЕМАТИЧНІ МОДЕЛІ НАДЛИШКОВИХ МЕТОДІВ ВИМІРЮВАНЬ ФІЗИЧНИХ ВЕЛИЧИН

В основу математичних моделей методів надлишкових вимірювань покладено формування інформативної надлишковості — оптимальної (необхідної і достатньої) кількості рядів фізичних величин (ФВ), розміри яких пов’язані між собою за певними законами, з метою подальшого складання системи нелінійних рівнянь величин для отримання рівняння надлишкових вимірювань.

Згідно з [33], математичні моделі методів надлишкових вимірювань розрізняються як за видом функції перетворення сенсора, так і за вибраними рядами перетворюваних ФВ, які, по-перше, є однорідні з шуканою (інформативною), а, по-друге, розміри цих величин складають арифметичну прогресію:

$$a_n = a_1 + d(n-1) \tag{2.1.4}$$

або геометричну прогресію:

$$b_n = b_1 q^{n-1}, \tag{2.1.5}$$

де a_1 і a_n — відповідно перший і n-й члени арифметичної прогресії;

d — різниця арифметичної прогресії;

b_1 і b_n — відповідно перший і n-й члени геометричної прогресії;

q — знаменник геометричної прогресії.

У методах надлишкових вимірювань замість математичного поняття різниці арифметичної прогресії використовується поняття «нормована фізична величина» ($\pm\Delta x_0$), а замість поняття «знаменник геометричної прогресії» — поняття коефіцієнт $k_л$ локальної лінеаризації, що не дорівнює нулю чи одиниці. Практично цей коефіцієнт вибирають в межах 20 %: $0{,}8 \le k_л \le 0{,}99$ і $1{,}01 \le k_л \le 1{,}2$.

Якщо ФВ мають направлену дію (наприклад, струм), то в цьому випадку інформативну вхідну величину x можна змінювати на нормовану за значенням величину Δx_0: $(x + \Delta x_0)$ чи $(x - \Delta x_0)$. У випадку, коли інформативна вхідна величина x має ненаправлену дію (наприклад, потік випромінювання) і операцію віднімання провести не можливо, то в цьому випадку застосовується додаткове формування фізичних величин x_2 і x_3 нормованого значення, наприклад, $\{x_2\}=\{x_0\}+\{\Delta x_0\}$ і $\{x_3\}=\{x_0\}-\{\Delta x_0\}$. У загальному випадку система нелінійних рівнянь величин має вид:

$$\begin{cases} y'_{н1} = f(0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н2} = f(x_0 - \Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н3} = f(x_0 + \Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н4} = f(x - \Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н5} = f(x + \Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}, \end{cases} \quad (2.1.6)$$

де $\Delta y'_{н}$ — зміщення функції перетворення з урахуванням адитивної складової похибки ($\{\Delta y'_{н}\} = \{\Delta y_{н}\} + \{\Delta_{а}\} + \{\Delta\gamma_t\}$, в якому $\{\Delta y_{Н}\}$ — номінальне значення
зміщення функції перетворення, $\{\Delta_{а}\}$ — адитивна складова похибки, $\{\Delta\gamma_t\}$ — випадкова складова похибки);

$S'_{Н1}$, $S'_{Л1}$ — крутість перетворення нелінійної і лінійної складових функції перетворення, де $S'_{Н1} = S_{Н1}(1 + \Delta S_{Н1} + \Delta_S(t))$ і $S'_{Л1} = S_{Л1}(1 + \Delta S_{Л1} + \Delta_S(t))$, в яких $S_{н1}, S_{л1}$ — ідеальні значення крутості перетворення нелінійної і лінійної складових функції перетворення, $\Delta S_{Н1}$, $\Delta S_{Л1}$ — значення, що обумовлюють систематичну складову похибки, $\Delta_S(t)$ — значення, що обумовлюють випадкову складову похибки.

Якщо параметри функції перетворення (ФП) можуть бути нормовано змінені, тобто є керованими, то, наприклад, при наступних п'яти ФВ: $\{x_1\}$=0, $\{x_2\}$=$\{x\}$, $\{x_3\}$=$k_{л1}\{x\}$, $\{x_4\}$=$\{\Delta x_0\}$ і $\{x_5\}$=$k_{л1}\{\Delta x_0\}$ можна скласти систему нелінійних рівнянь величин, що описує стан сенсора у дискретні моменти часу:

$$\begin{cases} y'_{н1} = f(0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н2} = f(x, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н3} = f(k_{л}x, S'_{н1}, S'_{л1}k_{л1}) + \Delta y'_{н}; \\ y'_{н4} = f(\Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}; \\ y'_{н5} = f(k_{л}\Delta x_0, S'_{н1}, S'_{л1}) + \Delta y'_{н}, \end{cases} \quad (2.1.7)$$

де $k_{\text{л}}$ — коефіцієнт локальної лінеаризації.

Рішення даної системи відносно інформативної ФВ дає змогу отримати рівняння надлишкових вимірювань, яке в загальному випадку має такий вигляд:

$$x = \Delta x_0 F(\Delta x_0, n, k_{\text{л}1}, y'_{\text{н}1}, ..., y'_{\text{н}n}), \tag{2.1.8}$$

де F — функція взаємного зв'язку інформативної ФВ від перетворених та нормованих за значенням ФВ і коефіцієнтів, які приведені в дужках;
n — показник ступеня при ступеневій ФП вимірювального каналу;
$y'_{\text{н}1}, \ldots, y'_{\text{н}n}$ — вихідні величини.

Розглянемо інші варіанти систем математичних моделей, що реалізують методи надлишкових вимірювань, у випадку, коли параметри функції перетворення (ФП) можуть бути нормовано змінені, тобто керованими. Покажемо, що отримані рівняння надлишкових вимірювань забезпечують визначення дійсного значення вхідної фізичної величини x, яке приведене до входу, тобто забезпечують лінійну залежність результату вимірювання. Крім того, додаткове визначення параметрів $(S'_{\text{н}}, S'_{\text{л}})$ нелінійної ФП при кожному вимірюванні забезпечує можливість визначити час виходу цих параметрів за довірчі інтервали та спрогнозувати метрологічну надійність засобу вимірювання.

Покажемо на прикладі дев'яти систем різні варіанти формування рядів вхідних величин, а також виведення рівняння надлишкових вимірювань при кубічній ФП.

Система нелінійних рівнянь величин №1.

$$\begin{cases} y_{н1} = \Delta y_н; \\ y_{н2} = S'_н \Delta x^3 + S'_л \Delta x + \Delta y_н; \\ y_{н3} = S'_н k_{л1}^3 \Delta x^3 + S'_л k_{л1} \Delta x + \Delta y_н; \\ y_{н4} = S'_н x^3 + S'_л k_{л1} x + \Delta y_н; \\ y_{н5} = S'_н x^3 + S'_л \dfrac{x}{k_{л1}} + \Delta y_н. \end{cases} \quad (2.1.9)$$

Виведення рівняння надлишкових вимірювань виконується наступним чином. Спочатку помножимо рівняння для $y_{н2}$ на $(k_{л1})^3$ та знайдемо різницю для $y_{н3}$ і $y_{н2}$:

$$y_{н3} - y_{н2} k_{л1}^3 = S'_л \Delta x (k_{л1} - k_{л1}^3) + \Delta y_н (1 - k_{л1}^3). \quad (2.1.10)$$

З отриманого рівняння виразимо $S'_л$ і підставимо його в різницю для $y_{н5}$ і $y_{н4}$:

$$S'_л = \frac{(y_{н3} - y_{н2} k_{л1}^3) - \Delta y_н (1 - k_{л1}^3)}{\Delta x (k_{л1} - k_{л1}^3)}, \quad (2.1.11)$$

$$y_{н5} - y_{н4} = S'_л x \left(\frac{1 - k_{л1}^2}{k_{л1}}\right). \quad (2.1.12)$$

З останнього рівняння отримуємо рівняння надлишкових вимірювань:

$$x = \frac{k_{л1}(y_{н5} - y_{н4})}{S'_л (1 - k_{л1}^2)} = \frac{(y_{н5} - y_{н4}) k_{л1} \Delta x (k_{л1} - k_{л1}^3)}{(1 - k_{л1}^2)[(y_{н3} - y_{н2} k_{л2}^3) - y_{н8}(1 - k_{л1}^3)]}. \quad (2.1.13)$$

Система нелінійних рівнянь величин №2.

$$\begin{cases} y_{н1} = S'_н x^3 + S'_л x + \Delta y_н; \\ y''_{н4} = S'_н x^3 + S'_л k_{л1} x + \Delta y_н; \\ y_{н5} = S'_н \Delta x^3 + S'_л \Delta x + \Delta y_н; \\ y_{н7} = S'_н k_{л2}^3 \Delta x^3 + S'_л k_{л1} \Delta x + \Delta y_н; \\ y_{н8} = \Delta y_н. \end{cases} \quad (2.1.14)$$

Виведемо рівняння надлишкових вимірювань. Для цього помножимо $y_{н5}$ на $(k_{л2})^3$ та віднімемо його від $y_{н7}$, а з отриманої різниці виразимо $S'_л$:

$$y_{н7} - y_{н5} k_{л2}^3 = S'_л \Delta x (k_{л1} - k_{л2}^3) + \Delta y_н (1 - k_{л2}^3), \quad (2.1.15)$$

$$S'_л = \frac{(y_{н7} - y_{н5} k_{л2}^3) - \Delta y_н (1 - k_{л2}^3)}{\Delta x (k_{л1} - k_{л2}^3)}, \quad (2.1.16)$$

і, підставивши отримане значення $S'_л$ в різницю для $y_{н1}$ і $y''_{н4}$, знайдемо x:

$$y_{н1} - y''_{н4} = S'_л x (1 - k_{л1}), \quad (2.1.17)$$

$$x = \frac{(y_{н1} - y''_{н4})}{S'_л (1 - k_{л1})} = \frac{(y_{н1} - y''_{н4}) \Delta x (k_{л1} - k_{л2}^3)}{(1 - k_{л1})[(y_{н7} - y_{н5} k_{л2}^3) - y_{н8} (1 - k_{л2}^3)]}. \quad (2.1.18)$$

При $k_{л1}$=$k_{л2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x = \frac{(y_{н1} - y''_{н4})\Delta x(k_{л1} + k_{л1}^2)}{(y_{н7} - y_{н5}k_{л1}^3) - y_{н8}(1 - k_{л1}^3)}. \qquad (2.1.19)$$

Система нелінійних рівнянь величин №3.

$$\begin{cases} y_{н1} = S'_{н}x^3 + S'_{л}x + \Delta y_{н}; \\ y''_{н4} = S'_{н}x^3 + S'_{л}k_{л1}x + \Delta y_{н}; \\ y'_{н14} = S'_{н}(k_{л2}x_0)^3 + S'_{л}k_{л1}x_0 + \Delta y_{н}; \\ y'_{н15} = S'_{н}(x_0 / k_{л2})^3 + S'_{л}(x_0 / k_{л1}) + \Delta y_{н}; \\ y_8 = \Delta y_{н}. \end{cases} \qquad (2.1.20)$$

Виведення рівняння надлишкових вимірювань виконується наступним чином. Помножимо рівняння для $y'_{н14}$ на $(1/k_{л1})$, рівняння для $y'_{н15}$ на $k_{л1}$ та знайдемо їх різницю, а з отриманої різниці виразимо $S'_{н}$:

$$\frac{y'_{н14}}{k_{л1}} - y'_{н15}k_{л1} = S'_{н}x_0\left(\frac{k_{л2}^6 - k_{л1}^2}{k_{л1}k_{л2}^3}\right) + \Delta y_{н}\left(\frac{1 - k_{л1}^2}{k_{л1}}\right), \qquad (2.1.21)$$

$$S'_{н} = \frac{\dfrac{(y'_{н14} - y'_{н15}k_{л1}^2) - \Delta y_{н8}(1 - k_{л1}^2)}{k_{л1}}}{\dfrac{x_0(k_{л2}^6 - k_{л1}^2)}{k_{л1}k_{л2}^3}} = \frac{[(y'_{н14} - y'_{н15}k_{л1}^2) - \Delta y_{н}(1 - k_{л1}^2)]k_{л2}^3}{x_0(k_{л2}^6 - k_{л1}^2)}. \qquad (2.1.22)$$

Віднімемо від рівняння для $y_{н1}$, помножене на $k_{л1}$, рівняння для $y''_{н4}$, а з отриманої різниці виразимо x, підставивши отримане значення для $S'_{н}$:

$$k_{л1}y_{н1} - y''_{н4} = S'_{н}x^3(k_{л1} - 1) + \Delta y_{н}(k_{л1} - 1), \qquad (2.1.23)$$

$$x^3 = \frac{(k_{л1}y_{н1} - y''_{н4}) - \Delta y_н(k_{л1} - 1)}{S'_н(k_{л1} - 1)} = \frac{[(k_{л1}y_{н1} - y''_{н4}) - y_{н8}(k_{л1} - 1)]x_0(k_{л2}^6 - k_{л1}^2)}{(k_{л1} - 1)k_{л2}^3[(y'_{14} - y'_{15}k_{л1}^2) - y_8(1 - k_{л1}^2)]}$$
. (2.1.24)

При $k_{л1}$=$k_{л2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x^3 = \frac{[(k_{л1}y_{н1} - y''_{н4}) - y_{н8}(k_{л1} - 1)]x_0(k_{л1}^4 - 1)}{(k_{л1} - 1)k_{л1}[(y'_{14} - y'_{15}k_{л1}^2) - y_8(1 - k_{л1}^2)]}. \qquad (2.1.25)$$

Система нелінійних рівнянь величин №4

$$\begin{cases} y''_{н4} = S'_н x^3 + S'_л k_{л1} x + \Delta y_н; \\ y_8 = \Delta y_н; \\ y_{н13} = S'_н x^3 + S'_л \dfrac{x}{k_{л1}} + \Delta y_н; \\ y_{н7} = S'_н (k_{л2}\Delta x)^3 + S'_л k_{л1}\Delta x + \Delta y_н; \\ y_{н15} = S'_н (\Delta x / k_{л2})^3 + S'_л (\Delta x / k_{л1}) + \Delta y_н. \end{cases} \qquad (2.1.26)$$

Виведемо рівняння надлишкових вимірювань. Для цього помножимо рівняння для $y_{н7}$ на $(1/k_{л2})^3$, рівняння для $y_{н15}$ на $(k_{л2})^3$ та знайдемо їх різницю:

$$\frac{y_{н7}}{k_{л2}^3} - y_{н15}k_{л2}^3 = S'_л \Delta x\left(\frac{k_{л1}^2 - k_{л2}^6}{k_{л2}^3 k_{л1}}\right) + \Delta y_н\left(\frac{1 - k_{л2}^6}{k_{л2}^3}\right). \qquad (2.1.27)$$

З отриманого рівняння виразимо $S'_л$:

$$S'_{\text{л}} = \frac{\dfrac{(y_{\text{н}7} - y_{\text{н}15}k_{\text{л}2}^6) - \Delta y_{\text{н}8}(1-k_{\text{л}2}^6)}{k_{\text{л}2}^3}}{\Delta x(k_{\text{л}1}^2 - k_{\text{л}2}^6)\big/k_{\text{л}2}^3 k_{\text{л}1}} = \frac{[(y_{\text{н}7} - y_{\text{н}15}k_{\text{л}2}^6) - \Delta y_{\text{н}}(1-k_{\text{л}2}^6)]k_{\text{л}1}}{\Delta x(k_{\text{л}1}^2 - k_{\text{л}2}^6)}. \tag{2.1.28}$$

Віднімемо від рівняння для $y_{\text{н}13}$ рівняння для $y''_{\text{н}4}$, а з отриманої різниці виразимо x, підставивши отримане значення для $S'_{\text{л}}$:

$$y_{\text{н}13} - y''_{\text{н}4} = S'_{\text{л}}x((1-k_{\text{л}1}^2)\big/k_{\text{л}1}), \tag{2.1.29}$$

$$x = \frac{(y_{\text{н}13} - y''_{\text{н}4})k_{\text{л}1}}{S'_{л}(1-k_{\text{л}1}^2)} = \frac{(y_{\text{н}13} - y''_{\text{н}4})\Delta x(k_{\text{л}1}^2 - k_{\text{л}2}^6)}{(1-k_{\text{л}1}^2)[(y_{\text{н}7} - y_{\text{н}15}k_{\text{л}2}^6) - y_8(1-k_{\text{л}2}^6)]}. \tag{2.1.30}$$

При $k_{\text{л}1}$=$k_{\text{л}2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x = \frac{(y_{\text{н}13} - y''_{\text{н}4})\Delta x k_{\text{л}1}^2(1+k_{\text{л}1}^2)}{(y_{\text{н}7} - y_{\text{н}15}k_{\text{л}1}^6) - y_8(1-k_{\text{л}1}^6)}. \tag{2.1.31}$$

Система нелінійних рівнянь величин №5

$$\begin{cases} y''_{\text{н}4} = S'_{\text{н}}x^3 + S'_{\text{л}}k_{\text{л}1}x + \Delta y_{\text{н}}; \\ y_8 = \Delta y_{\text{н}}; \\ y_{\text{н}13} = S'_{\text{н}}x^3 + S'_{\text{л}}\dfrac{x}{k_{\text{л}1}} + \Delta y_{\text{н}}; \\ y'_{\text{н}14} = S'_{\text{н}}(k_{\text{л}2}x_0)^3 + S'_{\text{л}}k_{\text{л}1}x_0 + \Delta y_{\text{н}}; \\ y'_{\text{н}15} = S'_{\text{н}}(x_0\big/k_{\text{л}2})^3 + S'_{\text{л}}(x_0\big/k_{\text{л}1}) + \Delta y_{\text{н}}. \end{cases} \tag{2.1.32}$$

Виведення рівняння надлишкових вимірювань виконується наступним чином. Спочатку помножимо рівняння для $y'_{н14}$ на $(1/k_{л2})^3$, рівняння для $y'_{н15}$ на $(k_{л2})^3$ та знайдемо їх різницю:

$$(y'_{н14}/k_{л2}^3) - y'_{н15}k_{л2}^3 = S'_л x_0((k_{л1}^2 - k_{л2}^6)/k_{л2}^3 k_{л1}) + \Delta y_н((1-k_{л2}^6)/k_{л2}^3). \quad (2.1.33)$$

З отриманого рівняння виразимо $S'_л$:

$$S'_л = \frac{\left[(y'_{н14} - y'_{н15}k_{л2}^6) - \Delta y_{н8}(1-k_{л2}^6)\right]/k_{л2}^3}{x_0(k_{л1}^2 - k_{л2}^6)/k_{л2}^3 k_{л1}} = \frac{[(y'_{н14} - y'_{н15}k_{л2}^6) - \Delta y_н(1-k_{л2}^6)]k_{л1}}{x_0(k_{л1}^2 - k_{л2}^6)}. \quad (2.1.34)$$

Віднімемо від рівняння для $y_{н13}$ рівняння для $y''_{н4}$, а з отриманої різниці виразимо x, підставивши отримане значення для $S'_л$:

$$y_{н13} - y''_{н4} = S'_л x((1-k_{л1}^2)/k_{л1}), \quad (2.1.35)$$

$$x = \frac{(y_{н13} - y''_{н4})k_{л1}}{S'_л(1-k_{л1}^2)} = \frac{(y_{н13} - y''_{н4})x_0(k_{л1}^2 - k_{л2}^6)}{(1-k_{л1}^2)[(y'_{н14} - y'_{н15}k_{л2}^6) - y_8(1-k_{л2}^6)]}. \quad (2.1.36)$$

При $k_{л1}$=$k_{л2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x = \frac{(y_{н13} - y''_{н4})x_0 k_{л1}^2(1+k_{л1}^2)}{(y'_{н14} - y'_{н15}k_{л1}^6) - y_8(1-k_{л1}^6)}. \quad (2.1.37)$$

Система нелінійних рівнянь величин №6.

$$\begin{cases} y_{н1} = S'_н x^3 + S'_л x + \Delta y_н; \\ y'_{н4} = S'_н k_{л2} x^3 + S'_л x + \Delta y_н; \\ y'_{н14} = S'_н (k_{л2} x_0)^3 + S'_л k_{л1} x_0 + \Delta y_н; \\ y'_{н15} = S'_н (x_0 / k_{л2})^3 + S'_л (x_0 / k_{л1}) + \Delta y_н; \\ y_8 = \Delta y_н. \end{cases} \quad (2.1.38)$$

Виведемо рівняння надлишкових вимірювань. Для цього помножимо рівняння для $y'_{н14}$ на $(1/k_{л2})^3$, рівняння для $y'_{н15}$ на $(k_{л2})^3$ та знайдемо їх різницю, а з отриманої різниці виразимо $S'_л$:

$$\frac{y'_{н14}}{k^3_{л2}} - y'_{н15} k^3_{л2} = S'_л x_0 \left(\frac{k^2_{л1} - k^6_{л2}}{k^3_{л2} k_{л1}}\right) + \Delta y_н \left(\frac{1 - k^6_{л2}}{k^3_{л2}}\right), \quad (2.1.39)$$

$$S'_л = \frac{\dfrac{(y'_{н14} - y'_{н15} k^6_{л2}) - \Delta y_{н8}(1 - k^6_{л2})}{k^3_{л2}}}{\dfrac{x_0 (k^2_{л1} - k^6_{л2})}{k^3_{л2} k_{л1}}} = \frac{[(y'_{н14} - y'_{н15} k^6_{л2}) - \Delta y_н (1 - k^6_{л2})] k_{л1}}{x_0 (k^2_{л1} - k^6_{л2})}.$$

(2.1.40)

Віднімемо від рівняння для $y'_{н4}$, рівняння для $y_{н1}$, помножене на $k_{л2}$, а з отриманої різниці виразимо рівняння надлишкових вимірювань, підставивши отримане значення для $S'_л$:

$$y'_{н4} - k_{л2} y_{н1} = S'_л x (1 - k_{л2}) + \Delta y_н (1 - k_{л2}), \quad (2.1.41)$$

$$x = \frac{(y'_{н4} - k_{л2} y_{н1}) - \Delta y_н (1 - k_{л2})}{S'_л (1 - k_{л2})} = \frac{[(y'_{н4} - k_{л2} y_{н1}) - \Delta y_н (1 - k_{л2})] x_0 (k^2_{л1} - k^6_{л2})}{(1 - k_{л2}) k_{л1} [(y'_{н14} - y'_{н15} k^6_{л2}) - y_{н8} (1 - k^6_{л2})]}$$

.

(2.1.42)

Система нелінійних рівнянь величин №7.

$$\begin{cases} y''_{н4} = S'_{н}x^3 + S'_{л}k_{л1}x + \Delta y_{н}; \\ y_{н5} = S'_{н}\Delta x^3 + S'_{л}\Delta x + \Delta y_{н}; \\ y_{н7} = S'_{н}k^3_{л2}\Delta x^3 + S'_{л}k_{л1}\Delta x + \Delta y_{н}; \\ y_{н8} = \Delta y_{н}; \\ y_{н13} = S'_{н}x^3 + S'_{л}(x/k_{л1}) + \Delta y_{н}. \end{cases} \quad (2.1.43)$$

Виведення рівняння надлишкових вимірювань виконується наступним чином. Віднімемо від рівняння $y_{н7}$ рівняння $y_{н5}$, помножимо на $(k_{л2})^3$, а з отриманої різниці виразимо $S'_{л}$:

$$y_{н5}k^3_{л2} - y_{н7} = S'_{л}\Delta x(k^3_{л2} - k_{л1}) + \Delta y_{н}(k^3_{л2} - 1), \quad (2.1.44)$$

$$S'_{л} = \frac{(y_{н5}k^3_{л2} - y_{н7}) - \Delta y_{н}(k^3_{л2} - 1)}{\Delta x(k_{л1} - k^3_{л2})}. \quad (2.1.45)$$

Віднімемо від рівняння для $y''_{н4}$ рівняння для $y_{н13}$, а з отриманої різниці виразимо x, підставивши отримане значення для $S'_{л}$:

$$y''_{н4} - y_{н13} = S'_{л}x((k^2_{л1} - 1)/k_{л1}), \quad (2.1.46)$$

$$x = \frac{(y''_{н4} - y_{н13})k_{л1}}{S'_{л}(k^2_{л1} - 1)} = \frac{(y''_{н4} - y_{н13})\Delta x(k_{л1} - k^3_{л2})k_{л1}}{(k^2_{л1} - 1)[(y_{н7} - y_{н5}k^3_{л2}) - y_{8}(1 - k^3_{л2})]}. \quad (2.1.47)$$

При $k_{л1}$=$k_{л2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x = \frac{(y''_{н4} - y_{н13})\Delta x k_{л1}^2}{(y_{н7} - y_{н5}k_{л1}^3) - y_8(1-k_{л1}^3)}. \qquad (2.1.48)$$

Система нелінійних рівнянь величин №8.

$$\begin{cases} y_{н1} = S'_н x^3 + S'_л x + \Delta y_н; \\ y'_{н4} = S'_н k_{л2} x^3 + S'_л x + \Delta y_н; \\ y_{н5} = S'_н \Delta x^3 + S'_л \Delta x + \Delta y_н; \\ y_{н7} = S'_н k_{л2}^3 \Delta x^3 + S'_л k_{л1} \Delta x + \Delta y_н; \\ y_{н8} = \Delta y_н. \end{cases} \qquad (2.1.49)$$

Виведемо рівняння надлишкових вимірювань. Для цього помножимо $y_{н5}$ на $(k_{л2})^3$ та віднімемо його від $y_{н7}$, а з отриманої різниці виразимо $S'_л$:

$$y_{н7} - y_{н5}k_{л2}^3 = S'_л \Delta x(k_{л1} - k_{л2}^3) + \Delta y_н(1-k_{л2}^3), \qquad (2.1.50)$$

$$S'_л = \frac{(y_{н7} - y_{н5}k_{л2}^3) - \Delta y_н(1-k_{л2}^3)}{\Delta x(k_{л1} - k_{л2}^3)}. \qquad (2.1.51)$$

Віднімемо від рівняння для $y'_{н4}$, рівняння для $y_{н1}$, помножене на $k_{л2}$, а з отриманої різниці виразимо рівняння надлишкових вимірювань, підставивши отримане значення для $S'_л$:

$$y'_{н4} - k_{л2}y_{н1} = S'_л x(1-k_{л2}) + \Delta y_н(1-k_{л2}), \qquad (2.1.52)$$

$$x = \frac{(y'_{н4} - k_{л2} y_{н1}) - \Delta y_н (1 - k_{л2})}{S'_л (1 - k_{л2})} = \frac{[(y'_{н4} - k_{л2} y_{н1}) - \Delta y_н (1 - k_{л2})]\Delta x (k_{л1} - k_{л2}^3)}{(1 - k_{л2}) k_{л1} [(y_{н7} - y_{н5} k_{л2}^3) - y_{н8} (1 - k_{л2}^3)]} \quad . \quad (2.1.53)$$

Система нелінійних рівнянь величин №9.

$$\begin{cases} y_{н1} = S'_н x^3 + S'_л x + \Delta y_н; \\ y''_{н4} = S'_н x^3 + S'_л k_{л1} x + \Delta y_н; \\ y_{н5} = S'_н \Delta x^3 + S'_л \Delta x + \Delta y_н; \\ y_{н7} = S'_н k_{л2}^3 \Delta x^3 + S'_л k_{л1} \Delta x + \Delta y_н; \\ y_{н8} = \Delta y_н. \end{cases} \quad (2.1.54)$$

Виведемо рівняння надлишкових вимірювань. Віднімемо від рівняння для $y_{н5}$, помножене на $k_{л1}$, рівняння для $y_{н7}$, а з отриманої різниці виразимо $S'_н$:

$$y_{н5} k_{л1} - y_{н7} = S'_н \Delta x^3 (k_{л1} - k_{л2}^3) + \Delta y_н (k_{л1} - 1), \quad (2.1.55)$$

$$S'_н = \frac{(y_{н5} k^3 - y_{н7}) - \Delta y_н (k_{л1} - 1)}{\Delta x^3 (k_{л1} - k_{л2}^3)}. \quad (2.1.56)$$

Віднімемо від рівняння для $y''_{н4}$, рівняння для $y_{н1}$, помножене на $k_{л1}$, а з отриманої різниці виразимо x, підставивши отримане значення для $S'_н$:

$$y''_{н4} - k_{л1} y_{н1} = S'_н x^3 (1 - k_{л1}) + \Delta y_н (1 - k_{л1}), \quad (2.1.57)$$

$$x^3 = \frac{(y''_{н4} - k_{л1}y_{н1}) - y_{н8}(1-k_{л1})}{S'_л(1-k_{л1})} = \frac{[(y''_{н4} - k_{л1}y_{н1}) - y_{н8}(1-k_{л1})]\Delta x^3(k_{л1} - k_{л2}^3)}{(1-k_{л1})[(y_{н5}k_{л1} - y_{н7}) - y_{н8}(1-k_{л1})]}. \quad (2.1.58)$$

При $k_{л1}$=$k_{л2}$ рівняння надлишкових вимірювань буде матиме вигляд:

$$x^3 = \frac{[(y''_{н4} - k_{л1}y_{н1}) - y_{н8}(1-k_{л1})]\Delta x^3(k_{л1} + k_{л1}^2)}{(y_{н5}k_{л1} - y_{н7}) - y_{н8}(1-k_{л1})}. \quad (2.1.59)$$

Таким чином, представлена розмаїтість варіантів систем математичних моделей, що реалізують методи надлишкових вимірювань, дає можливість отримати рівняння надлишкових вимірювань дійсного значення вхідної фізичної величини *x*, яке приведене до входу, тобто забезпечують лінійну залежність результату вимірювання. Отриманий результат вимірювання не залежить від параметрів функції перетворення і, відповідно, їх змін. Таким чином, забезпечується виключення систематичних складових похибки, які обумовлені нестабільністю параметрів функції перетворення вимірювального каналу під дією дестабілізуючих факторів. Крім того, додаткове визначення параметрів ($S'_н, S'_л$) нелінійної функції перетворення забезпечує можливість прогнозування метрологічної надійність засобу вимірювання. Це, в свою чергу, також сприяє отриманню достовірної інформації від сенсора.

2.1.4. УЗАГАЛЬНЕНІ МАТЕМАТИЧНІ МОДЕЛІ НАДЛИШКОВИХ МЕТОДІВ ВИМІРЮВАНЬ ПРИ БАГАТОКРАТНИХ ВИМІРЮВАННЯХ

Оскільки МНВ зарекомендували себе як високоефективні методи, тому є підстави провести дослідження представлених методів при багаторазових вимірюваннях з нелінійною функцією перетворення,

оскільки вплив дестабілізуючих факторів на сенсор приводить до появи поряд з систематичними і випадкових складових похибки вимірювання.

Наведені вище варіанти математичних моделей застосовуються у випадку, коли не поставлена задача зменшення випадкової складової похибки або коли вона незначна. Такі математичні моделі називають моделями МНВ І-го роду [34], вони забезпечують виключення систематичних складових похибки результату вимірювання, спричинених впливом на нього абсолютних значень параметрів функції перетворення вимірювального каналу і їх відхилень від номінальних значень. При наявності значних випадкових похибок і для зменшення їх впливу на результат вимірювання використовують МНВ ІІ-го і ІІІ-го роду [34].

Як відомо, у випадку проведення багаторазових вимірювань, коли виникає необхідність враховувати випадкові складові похибки, використовують, як правило, методи статистичної обробки даних. В теорії надлишкових вимірювань ця процедура виконується з використанням особливих підходів, що властиві МНВ.

Якщо при вимірюваннях виникає випадкова складова похибка, яка представляє собою стаціонарний процес, то в цьому випадку використовуються МНВ ІІ-го роду, тобто здійснюється вимірювальне перетворення кожної ФВ ряду m із наступною статистичною обробкою даних. Можливі й інші варіанти багаторазових вимірювань, коли кратність вимірювального перетворення в кожному такті різна. Слід зазначити, що МНВ ІІ-го роду також використовують, використовуючи два підходи. При першому підході в кожному з m циклів відбувається вимір всіх n тактів вимірювань фізичних величин. Далі отримані результати вимірювань усереднюють по кожній з цих фізичних величин і підставляються в рівняння надлишкових вимірювань контрольованої фізичної величини.

Таким чином, здійснюється статистична обробка результатів багаторазових вимірювань за n тактами та m циклами. Відповідна система

нелінійних рівнянь величин та рівняння надлишкових вимірювань будуть такими [34]:

$$\begin{cases} \overline{y}_{\text{н}1} = \dfrac{1}{m}\sum\limits_{j=1}^{m} f(n, x_{1j}, S'_{\text{л}1}, S'_{\text{н}1}) + \Delta y'_{\text{н}1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \overline{y}_{\text{н}n} = \dfrac{1}{m}\sum\limits_{j=1}^{m} f(n, x_{nj}, S'_{\text{л}n}, S'_{\text{н}n}) + \Delta y'_{\text{нn}}, \end{cases} \qquad (2.1.60)$$

та

$$\overline{x}_i = \Delta x_{0(i)}\left(\frac{1}{m_1}\sum_{j=1}^{m_1} F(\Delta x_0, x_i, n, k_{\text{л}1}, \ldots, k_{\text{л}4}, y'_{\text{н}1j}, \ldots, y'_{\text{н}nj})\right). \qquad (2.1.61)$$

Таким чином, при першому підході в кожному з m циклів відбувається вимір всіх *n* тактів вимірювань фізичних величин. Далі отримані результати вимірювань усереднюють по кожній з цих фізичних величин і підставляються в рівняння надлишкових вимірювань контрольованої фізичної величини.

При другому підході проводять *m* раз цикли вимірювань, в кожному з яких по рівнянню надлишкових вимірювань знаходять значення контрольованої фізичної величини, а отримані значення усереднюють за m циклів вимірювань.

Таким чином, здійснюється статистична обробка результатів багаторазових вимірювань за *n* тактами та *m* циклами. Відповідна система нелінійних рівнянь величин та рівняння надлишкових вимірювань будуть такими:

$$
\begin{cases}
\begin{cases} y_{\text{н}1} = \sum\limits_{j=1}^{m} f(n, x_{1j}, S'_{\text{л}1}, S'_{\text{н}1}) + \Delta y'_{\text{н}1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_{\text{н}n} = \sum\limits_{j=1}^{m} f(n, x_{nj}, S'_{\text{л}n}, S'_{\text{н}n}) + \Delta y'_{\text{н}n}, \end{cases} \Rightarrow \quad x_{i1} = \Delta x_{0(i)}\left(F(\Delta x_0, x_i, n, k_{\text{л}1}, \ldots, k_{\text{л}4}, y'_{\text{н}1j}, \ldots, y'_{\text{н}nj})\right) \\
\begin{cases} y_{\text{н}1} = \sum\limits_{j=1}^{m} f(n, x_{1j}, S'_{\text{л}1}, S'_{\text{н}1}) + \Delta y'_{\text{н}1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_{\text{н}n} = \sum\limits_{j=1}^{m} f(n, x_{nj}, S'_{\text{л}n}, S'_{\text{н}n}) + \Delta y'_{\text{н}n}, \end{cases} \Rightarrow \quad x_{i2} = \Delta x_{0(i)}\left(F(\Delta x_0, x_i, n, k_{\text{л}1}, \ldots, k_{\text{л}4}, y'_{\text{н}1j}, \ldots, y'_{\text{н}nj})\right) \\
\ldots\ldots \\
\begin{cases} y_{\text{н}1} = \sum\limits_{j=1}^{m} f(n, x_{1j}, S'_{\text{л}1}, S'_{\text{н}1}) + \Delta y'_{\text{н}1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_{\text{н}n} = \sum\limits_{j=1}^{m} f(n, x_{nj}, S'_{\text{л}n}, S'_{\text{н}n}) + \Delta y'_{\text{н}n}, \end{cases} \Rightarrow \quad x_{im} = \Delta x_{0(i)}\left(F(\Delta x_0, x_i, n, k_{\text{л}1}, \ldots, k_{\text{л}4}, y'_{\text{н}1j}, \ldots, y'_{\text{н}nj})\right)
\end{cases}
$$

(2.1.62)

де

$$\overline{x_i} = \frac{1}{m}\sum_{j=1}^{m} x_{ij}\,. \tag{2.1.63}$$

У випадку, коли випадкова складова похибки вимірювання представляє собою нестаціонарний процес (математичне очікування не дорівнює нулю), то використовують МНВ ІІІ-го роду. Вони відрізняються від МНВ ІІ-го роду тим, що всі дії, що були описані при МНВ ІІ-го роду, повторюються m_2 рази з наступною статистичною обробкою результатів багаторазових вимірювань.

З теорії статистичної обробки даних відомо, що, при багаторазових m_1 раз вимірюваннях ФВ, забезпечується зменшення випадкової складової похибки в $\sqrt{m_1}$ раз.

Таким чином, МНВ ІІ-го та ІІІ-го роду реалізують класичний підхід усереднення результатів багаторазових вимірювань та статистичну обробку отриманих результатів. Практично доцільно використовувати 10-ти кратне вимірювальне перетворення та обробку отриманих даних з урахуванням критерію Стьюдента.

МНВ забезпечують широкі можливості не тільки щодо виключення систематичної похибки (що обумовлені зміною параметрів функції перетворення), а і відносно обробки результатів проміжних вимірювань при наявності випадкових похибок. Як видно з наведених рівнянь надлишкових вимірювань, МНВ ІІ-го і ІІІ-го роду можуть бути використані за різним числом вимірювань у кожному такті, що свідчить про їх гнучкість.

Вибір запропонованих математичних моделей залежить від характеру випадкової складової похибки, складності формування нормованих за значенням ФВ, складання необхідної системи нелінійних рівнянь величин та їх розв’язання.

2.2. РОЗРОБКА БЛОК-СХЕМ АЛГОРИТМІВ РОБОТИ МЕТОДУ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ РІЗНИХ ВИДАХ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА

2.2.1. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ

Як відомо, алгоритмом називають систему правил для вирішення певного класу задач. Тож розглянемо алгоритми роботи МНВ при різних видах функції перетворення сенсора.

Розглянемо основні етапи роботи алгоритму МНВ.

1. Визначається вид ФП і кількість n невідомих параметрів ФП.

2. Складається система з такої кількості рівнянь, щоб вивести n і більше число рівнянь вимірювань. Кількість рівнянь залежить від кількості можливих способів формування надлишковості і від складності виду ФП.

3. Виводиться рівняння вимірювання шуканої фізичної величини (ШФВ).

4. Виводиться рівняння вимірювання для кожного з параметрів ФП.

5. Порівнюється отримані значення ШФВ з заданими.

6. Порівнюється отримані значення параметрів ФП з їх допустимими значеннями.

7. За результатами, отриманих в п.5 і п.6, приймається рішення про регулюванню технологічного процесу або про заміну датчика.

Алгоритм роботи МНВ матиме вид, що представлено на рис.2.2.1.

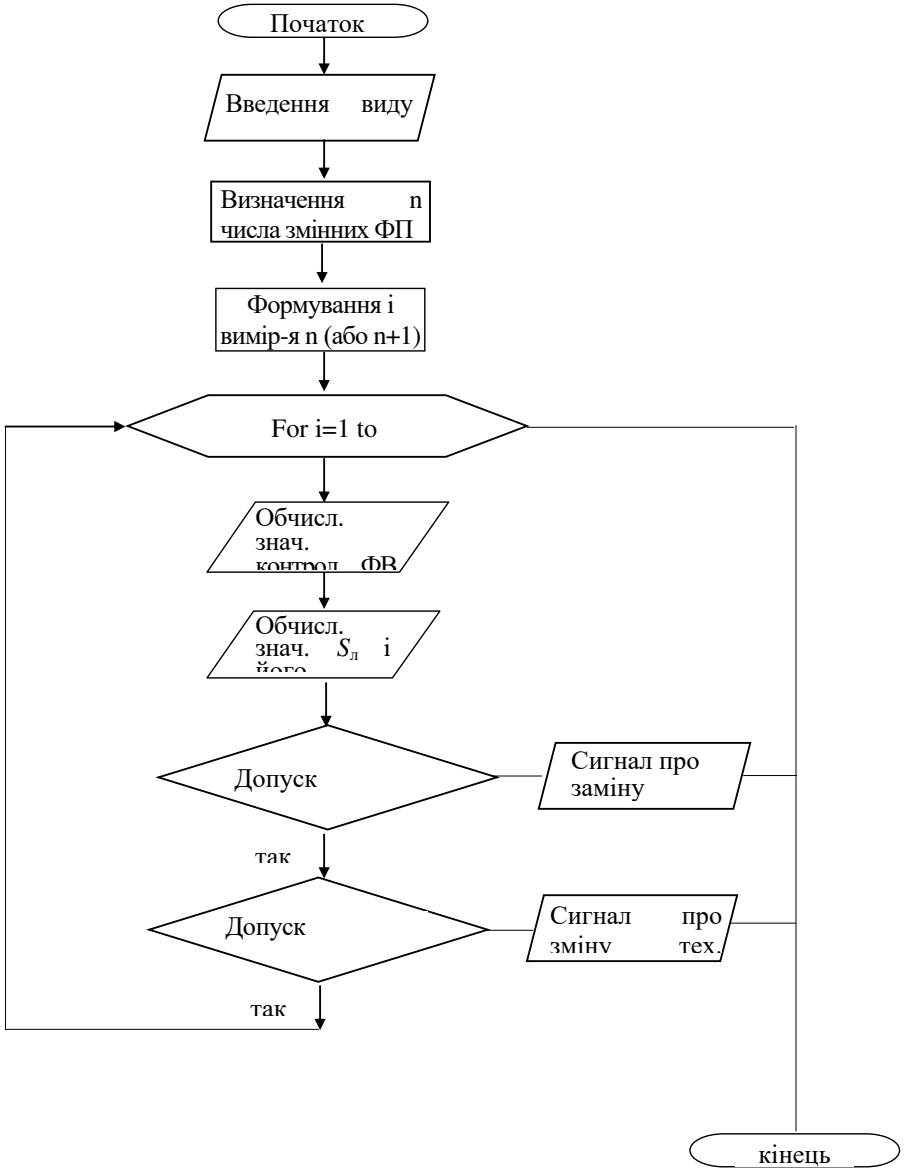


Рис.2.2.1. Алгоритм роботи МНВ

2.2.2. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ ЛІНІЙНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА

Розглянемо більш детально алгоритм роботи МНВ при лінійній функції перетворення сенсора.

Для створення алгоритмів роботи МНВ необхідно описати і дослідити математичну модель об'єкту дослідження, яка б адекватно описувала реальний стан речей. Математична модель будується при наявності апріорної інформації про об'єкт контролю і показує вплив факторів, що чинять дію на об'єкт контролю, а також вид функції перетворення вхідних і вихідних сигналів у сенсорі.

Розглянемо лінійну ФП:

$$y_l' = S_l' x_i + \Delta y', \tag{2.2.1}$$

де y'_l – сигнал на виході сенсора;

x_i – шукана фізична величина;

S'_l – чутливість (крутизна) перетворення лінейної складової ФП;

$\Delta y'$ – зміщення ФП з урахуванням адитивної складової похибки.

У рівнянні (2.2.1) всі величини (y'_l, S'_l $\Delta y'$) вказані з штрихами, що вказує на їх реальні (неідеальні) значення, тобто значення з похибкою. Дотримуючись п.1 алгоритму, визначаємо, кількість невідомих параметрів ФП. Для рівняння (2.2.1) є 3 невідомих (y'_l, S'_l, $\Delta y'$), тому можна скласти систему з 3х рівнянь надлишкових вимірювань. Таким чином, необхідно сформувати ще два рівняння. Для цього використовують нормовану (або нормовані) за значенням фізичну величину x1. Формування нормованої

величини відбувається за допомогою каліброваного джерела. В результаті, система рівнянь матиме вигляд:

$$\begin{cases} y'_{l1} = \Delta y'; \\ y'_{l2} = S'_l x_1 + \Delta y'; \\ y'_{l3} = S'_l x_i + \Delta y'. \end{cases} \qquad (2.2.2)$$

В результаті рішення системи надлишкових рівнянь (2.2.2) отримано рівняння шуканої фізичної величини (ШФВ):

$$x_i = x_1 \frac{\left(y'_{l3} - y'_{l1}\right)}{\left(y'_{l2} - y'_{l1}\right)} \qquad (2.2.3)$$

і рівняння параметрів ФП:

$$S'_l = \frac{\left(y'_{l2} - y'_{l1}\right)}{x_1}, \qquad \Delta y' = y'_{l1}\ . \qquad (2.2.4)$$

Як видно з рівняння (2.2.3), отриманий результат x_i не залежить від параметрів ФП (y'_l, $\Delta y'$). Таким чином, вимірюючи тільки сигнали на виході сенсора і обробляючи їх по запропонованим рівнянням (2.2.3), можна досягти незалежність від зміни параметрів ФП.

Для перевірки правильності виведених рівнянь (2.2.3 і 2.2.4) досить підставити замість величин y'_{l1}, y'_{l2} і y'_{l2} їх вирази з системи (2).

На основі запропонованої математичної моделі на рис. Р.2.2.2 продемонструємо блок-схема алгоритму вимірювання шуканої фізичної величини при лінійній функції перетворення.

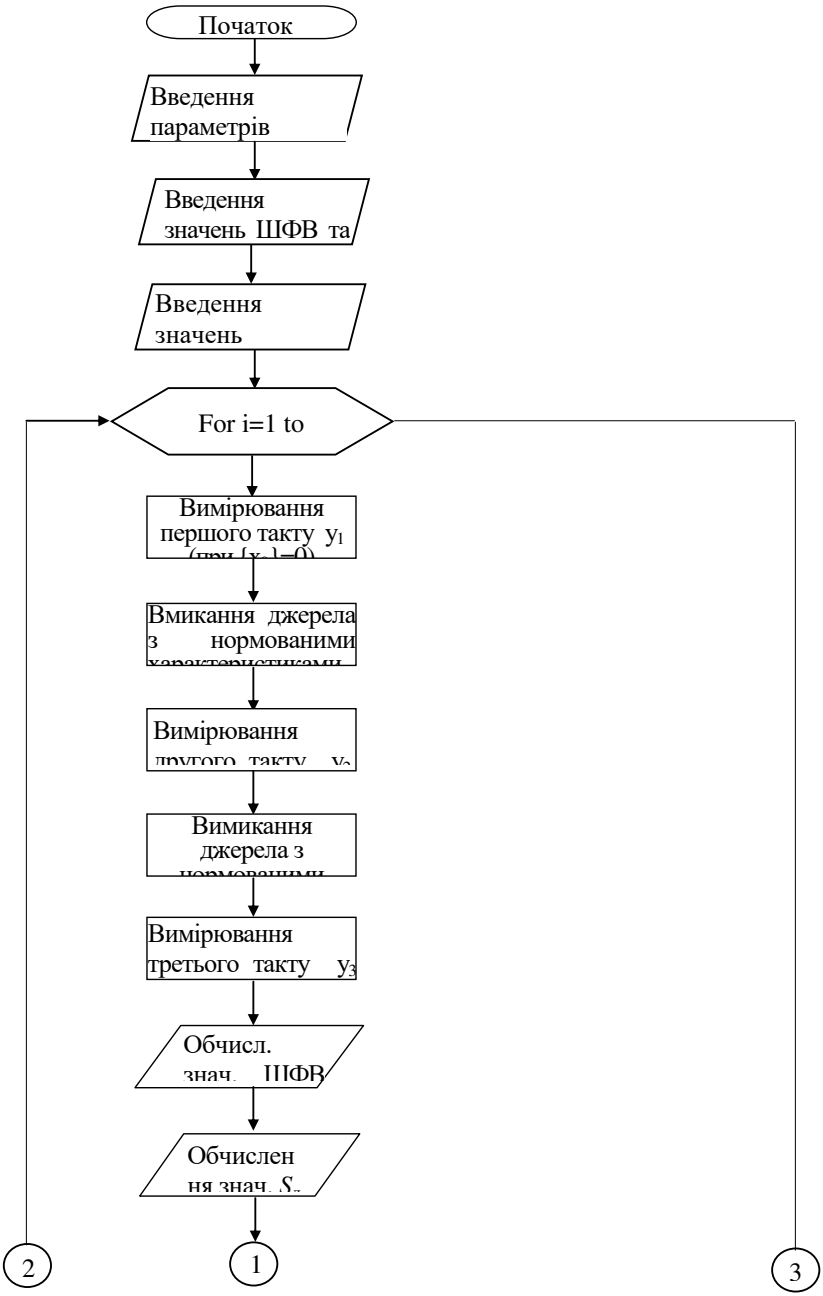


Рис.2.2.2 – Алгоритмів вимірювання ШФВ при лінійній функції

перетворення

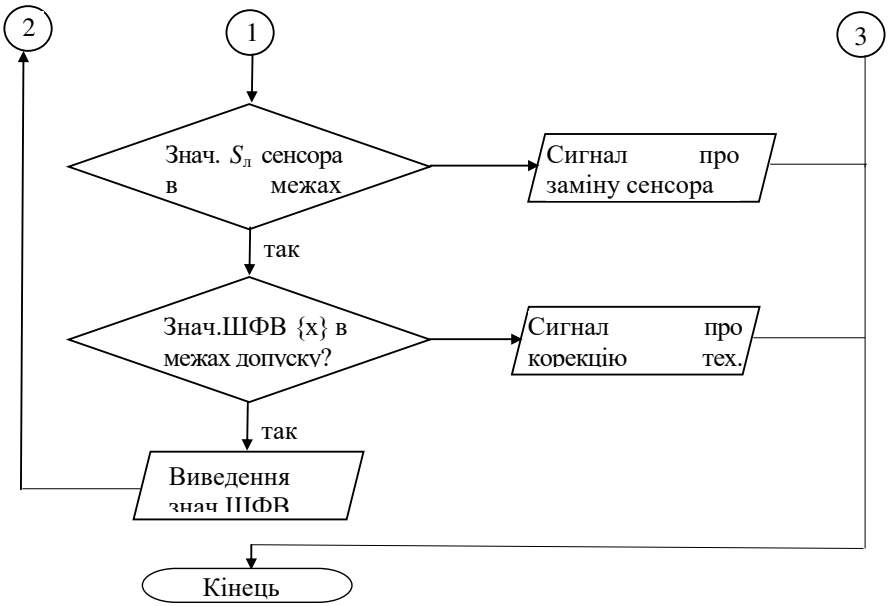


Рис.2.2.2 (продовження) – Алгоритмів вимірювання ШФВ при лінійній функції перетворення

Запропонований алгоритм є підставою для розробки відповідних технічних рішень, що реалізує метод надлишкових вимірювань. В таких технічних рішеннях використовують сенсор з лінійною ФП, а також мікроковертор. У пам'яті мікроконвертора апріорі записується код числа N_0, що відповідає каліброваному за значенням фізичній величині $\{x_1\}$. Крім того, у пам'яті мікроконвертора відповідно до запропонованому алгоритму записується програма виконання заданої послідовності операцій. Вона забезпечує функціонування технічних рішень як єдиного програмно-технічного засобу вимірювання, а також обробку результатів поточних вимірювань по заданому рівнянню числових значень. Крім того,

мікроконвертор приймає аналогові сигнали за допомогою вбудованого АЦП, перетворює їх в коди відповідно до програми, що записана в ПЗП мікроконвертора, обробляє ці коди, отримані результати передає на рідиннокристалічний індикатор, а коди керування за допомогою ЦАП знову перетворює в аналогові сигнали. Статична ОЗП виконує функції накопичення і збереження інформації. Пристрій виводить отримані дані на екран рідино-кристалічного індикатора, на екран ПК або на друк за допомогою цифрового пристрою друку.

Завдяки впровадження запропонованого алгоритму, реалізація технічного рішення при лінійній функції перетворення забезпечує автоматичне виключення адитивної і мультиплікативної складових систематичної похибки результату визначення, що обумовлена зміною параметрів функції перетворення під дією дестабілізуючих факторів. Це забезпечується за рахунок реалізації методу надлишкових вимірювань, який забезпечує виключення впливу на результат вимірювання абсолютних значень параметрів лінійної функції перетворення вимірювального каналу ($S'_{\text{л}}$ і $\Delta U'_{\text{зм}}$) та їх відхилень від номінальних значень.

Якщо технічний пристрій працює в умовах експлуатації, які відрізняються від нормальних, і має місце вплив випадкової складової похибки, то кожен такт або цикл вимірювального перетворення повторюється (10 ÷ 34) разів (за Стьюдентом) з подальшою статистичною обробкою.

Завдяки введенню мікроконвертора з розрахунковим ядром для обробки даних, стало можливим виконання великого об’єму розрахунків.

Так на рис. 2.2.3 продемонстрована блок-схема алгоритму вимірювання шуканої фізичної величини при лінійній функції перетворення з використанням технічного рішення.

Рис. 2.2.3 – Алгоритм роботи технічного пристрою при лінійній ФП

Рис. 2.2.3 (продовження) – Алгоритм роботи технічного пристрою при лінійній ФП

2.2.3. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА, ЩО ОПИСУЄТЬСЯ ПОЛІНОМОМ 3-ГО СТУПЕНЯ

Розглядається використання методів надлишкових вимірювань для випадку, коли сенсор має функцію перетворення, що описується поліномом

3-го ступеня. Покажемо можливість роботи даного сенсора при використанні всієї ділянки вхідної характеристики.

Дотримуючись запропонованого алгоритму (див. рис.2.2.1) складемо математичну модель для ФП, що має вигляд полінома 3й ступеня:

$$y'_n = S'_{n2}x^3{}_i + S'_{n1}x^2{}_i + S'_l x_i + \Delta y', \qquad (2.2.5)$$

де y'_n – сигнал на виході сенсора;

x_i – шукана фізична величина;

S'_l – лінейна складової ФП;

S'_{n1} і S'_{n2} – нелінейні складові ФП;

$\Delta y'$ – зміщення ФП з урахуванням адитивної складової похибки.

Оскільки рівняння (2.2.5) має складну структуру і п'ять невідових, то сформуємо надлишковість за допомогою нормованих за значенням величин x_0 і Δx. В результаті отримуємо систему виду:

$$\begin{cases} y'_{n1} = S'_{n2}(x_1)^3 + S'_{n1}(x_1)^2 + S'_l(x_1) + \Delta y'; \\ y'_{n2} = S'_{n2}(x_2)^3 + S'_{n1}(x_2)^2 + S'_l(x_2) + \Delta y'; \\ y'_{n3} = S'_{n2}(x_3)^3 + S'_{n1}(x_3)^2 + S'_l(x_3) + \Delta y'; \\ y'_{n4} = S'_{n2}(x_4)^3 + S'_{n1}(x_4)^2 + S'_l(x_4) + \Delta y'; \\ y'_{n5} = S'_{n2}(x_5)^3 + S'_{n1}(x_5)^2 + S'_l(x_5) + \Delta y'; \\ y'_{n6} = S'_{n2}(x_6)^3 + S'_{n1}(x_6)^2 + S'_l(x_6) + \Delta y'; \\ y'_{n7} = S'_{n2}(x_7)^3 + S'_{n1}(x_7)^2 + S'_l(x_7) + \Delta y'; \\ y'_{n8} = S'_{n2}(x_8)^3 + S'_{n1}(x_8)^2 + S'_l(x_8) + \Delta y', \end{cases} \qquad (2.2.6)$$

де $\{x_1\}=\{x0\}-\{\Delta x\}$, $\{x_2\}=\{x0\}-2\{\Delta x\}$, $\{x_3\}=\{x0\}+\{\Delta x\}$, $\{x_4\}=\{x0\}+2\{\Delta x\}$, $\{x_5\}=\{x_i\}+\{x_1\}=\{x_i\}+\{x0\}-\{\Delta x\}$, $\{x_6\}=\{x_i\}+\{x_2\}=\{x_i\}+\{x0\}-2\{\Delta x\}$, $\{x_7\}=\{x_i\}+\{x_3\}=\{x_i\}+\{x0\}+\{\Delta x\}$, $\{x_8\}=\{x_i\}+\{x_4\}=\{x_i\}+\{x0\}+2\{\Delta x\}$.

В результаті рішення системи надлишкових рівнянь (2.2.6) отримуємо рівняння шуканої фізичної величини (2.2.7) і відповідних параметрів (2.2.8–2.2.10):

$$x_i = \frac{(y'_{n7} - y'_{n5}) - (y'_{n3} - y'_{n1}) - 4((y'_{n8} - y'_{n4}) - (y'_{n7} - y'_{n3}))}{-3(2(y'_{n7} - y'_{n5}) - (y'_{n8} - y'_{n6}))}, \quad (2.2.7)$$

$$S'_{n2} = \left[\left((y'_{n8} - y'_{n6}) - 2(y'_{n7} - y'_{n5}) \right) \Big/ 12 \cdot \Delta x^3 \right], \quad (2.2.8)$$

$$S'_{n1} = \frac{2\Delta x \left[(y'_{n4} + y'_{n2}) - (y'_{n3} + y'_{n1}) \right]}{12\Delta x^3} - \frac{3x_0 \left[(y'_{n8} - y'_{n6}) - 2(y'_{n7} - y'_{n5}) \right]}{12\Delta x^3} \quad (2.2.9)$$

$$S'_l = \frac{6\Delta x^2 (y'_{n3} - y'_{n1}) - 4\Delta x \cdot x_0 \left[(y'_{n4} + y'_{n2}) - (y'_{n3} + y'_{n1}) \right]}{12\Delta x^3} - \frac{(3x_0^2 - \Delta x) \left[(y'_{n8} - y'_{n6}) - 2(y'_{n7} - y'_{n5}) \right]}{12\Delta x^3}, \quad (2.2.10)$$

$$\Delta y' = y'_{n1} - \frac{(x_0 - \Delta x)}{12\Delta x^3} \Big[6\Delta x^2 (y'_{n3} - y'_{n1}) - 2\Delta x (x_0 + \Delta x) \left[(y'_{n4} + y'_{n2}) - (y'_{n3} + y'_{n1}) \right] + x_0 (x_0 + \Delta x) \left[(y'_{n8} - y'_{n6}) - 2(y'_{n7} - y'_{n5}) \right] \Big]. \quad (2.2.11)$$

На основі запропонованої математичної моделі на рис. Р.2.2.4 продемонструємо блок-схема алгоритму вимірювання шуканої фізичної величини при лінійній функції перетворення.

Блок-схема алгоритму роботи технічного приладу, що реалізує МНВ при поліноміальній функції перетворення сенсора, аналогічна попередній (при лінійній функції перетворення) і відрізняється кількістю занесених до пам'яті кодів чисел, що відповідають каліброваним за значенням фізичній величині ($\{x_1\}$–$\{x_4\}$), а також кількістю вимірювальних перетворень нормованих за значенням фізичних величин та діями по їх відтворенню.

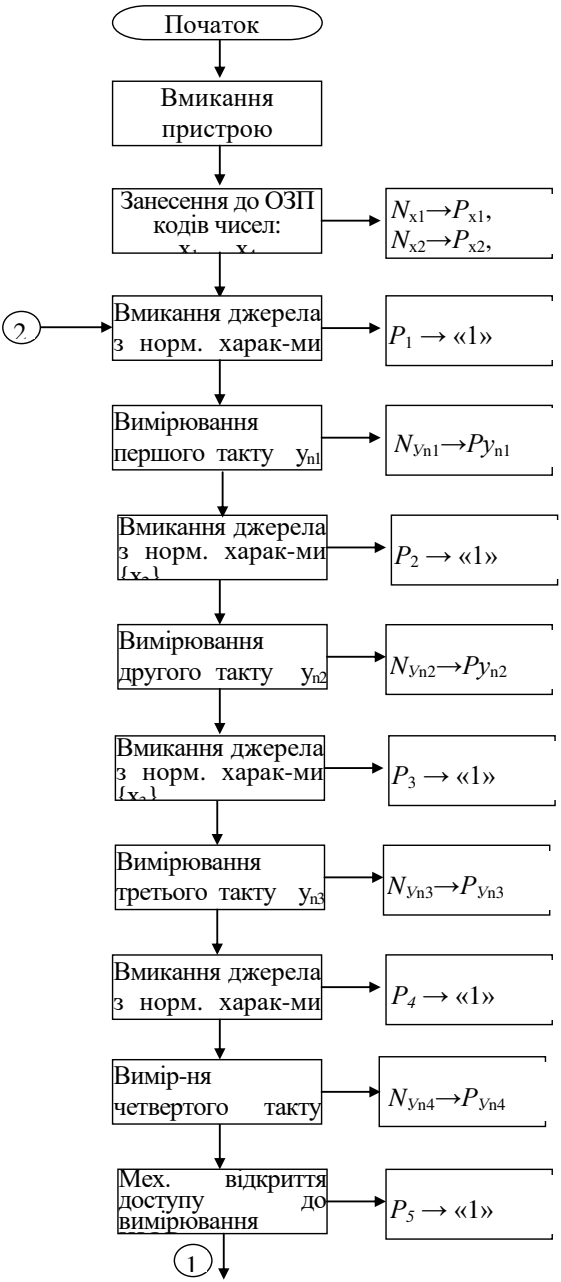


Рис. 2.2.4 – Алгоритм роботи технічного пристрою при поліноміальній ФП

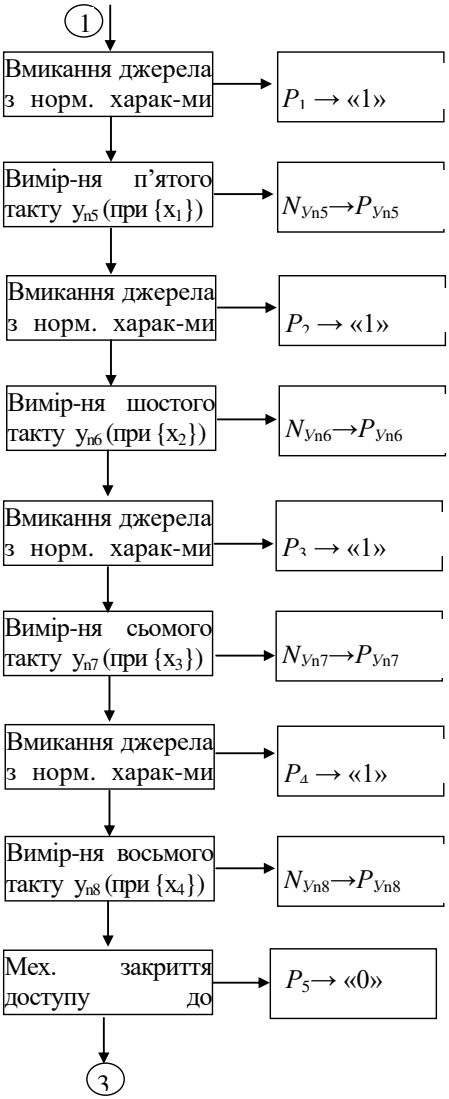


Рис. 2.2.4 (продовження) – Алгоритм роботи технічного пристрою при поліноміальній ФП

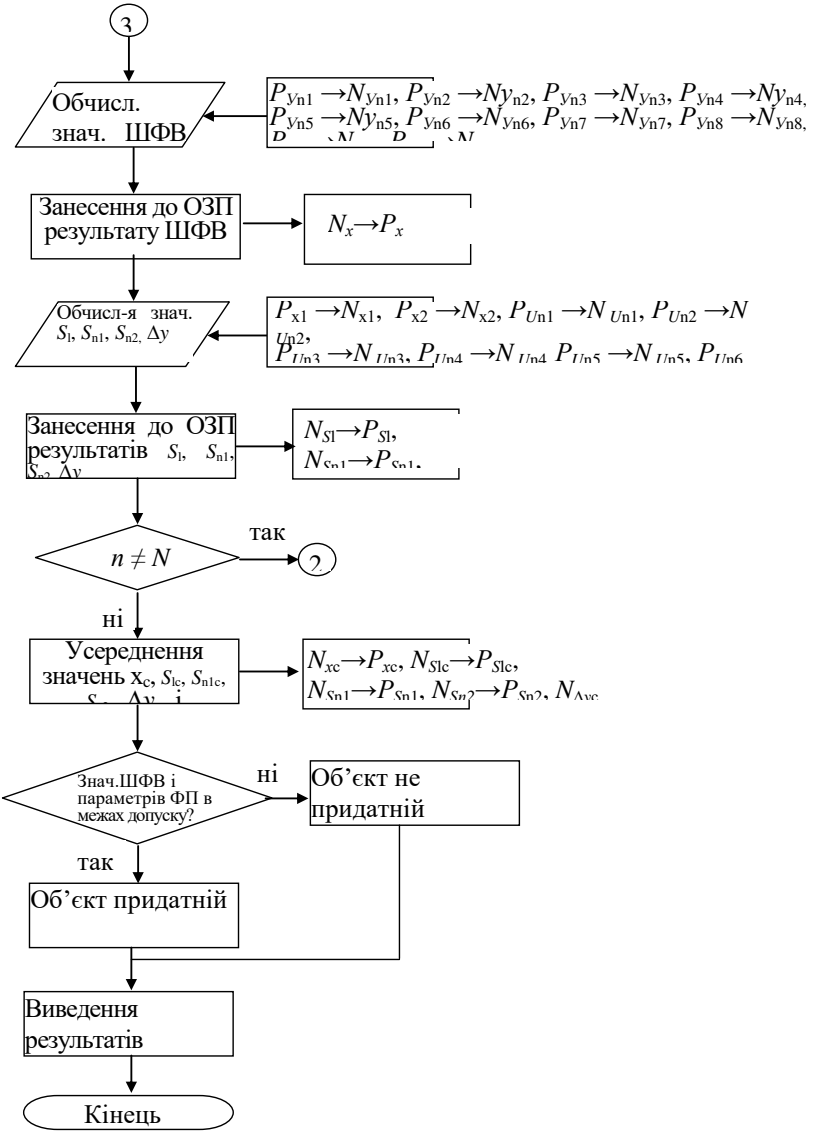


Рис. 2.2.4 (закінчення) – Алгоритм роботи технічного пристрою при поліноміальній ФП

Слід зазначити, що при наявності випадкових похибок, що являють собою стаціонарний процес, рівняння надлишкових вимірювань (2.2.7) прийме вид:

$$\overline{x}_i = \frac{(\overline{y}'_{n7} - \overline{y}'_{n5}) - (\overline{y}'_{n3} - \overline{y}'_{n1}) - 4((\overline{y}'_{n8} - \overline{y}'_{n4}) - (\overline{y}'_{n7} - \overline{y}'_{n3}))}{-3(2(\overline{y}'_{n7} - \overline{y}'_{n5}) - (\overline{y}'_{n8} - \overline{y}'_{n6}))}, \qquad (2.2.12)$$

де $\overline{y}'_{ni}$ – усереднені значення вихідних сигналів сенсора.

2.2.4. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ ЛОГАРИФМІЧНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА

Розглядається використання методів надлишкових вимірювань для випадку, коли в якості фотоприймача використовується фотодіод з логарифмічною функцією перетворення. Покажемо можливість роботи даного фотоприймача при використанні всієї ділянки логарифмічної характеристики.

ФП, що описується логарифмічною функцією, має вид:

$$y'_n = S'_n \ln\left(\frac{1}{x_k} \cdot x_i + 1\right) - \Delta y', \qquad (2.2.13)$$

де y'_n – сигнал на виході сенсора;

x_i – шукана фізична величина;

x_k – фізична величина, що описується через паспортні параметри сенсора;

S'_n – нелінейна складова ФП;

$\Delta y'$ – зміщення ФП з урахуванням адитивної складової похибки.

Оскільки в рівнянні (2.2.13) шуканою (контрольованою) величиною є потік x_i, то рівняння (2.2.13) було розв'язане відносно нього. В результаті отримано вираз наступного виду:

$$x_i = x_k \left(e^{\frac{y'_n + \Delta y'}{S'_n}} - 1 \right). \qquad (2.2.14)$$

В рівняннях величин (2.2.13) і (2.2.14) параметри $\Delta y'$ і S'_n вказані зі штрихами, що вказує на їх реальні (не ідеальні) значення, тобто значення з похибкою. Отже, при застосуванні ненадлишкових методів результат вимірювання шуканої величини x_i буде залежати від точного знання цих величин. Крім того, логарифмічну функцію перетворення необхідно лінеаризувати або працювати на лінійній її ділянці. Зазначені задачі по зменшенню похибки від нелінійності і нестабільності функції перетворення успішно вирішують методи надлишкових вимірювань.

Для реалізації надлишкових вимірювань необхідно скласти математичну модель у вигляді системи рівнянь, що описує такти вимірювання. При цьому, кількість тактів визначається кількістю змінних в представленій ФП. Оскільки ФП, що описується рівнянням (2.2.13), має 4 змінні, то система рівнянь буде складатися з (n+1), тобто 5 рівнянь величин. Для цього, крім такту вимірювання шуканої величини x_i проводять такти вимірювання нормованих за значенням величин x_0 і Δx: $\{x_1\}=\{\Delta x\}$, $\{x_2\}=\{x0\}$, $\{x_3\}=\{x_1\}+\{x_2\}=\{x0\}+\{\Delta x\}$, $\{x_4\}=\{x_i\}$, $\{x_5\}=\{x_i\}+\{x_1\}=\{x_i\}+\{\Delta x\}$. Формування нормованої величини відбувається за допомогою каліброваного джерела. Таким чином на сенсор послідовно (в різні такти вимірювання) надходять 5 потоків випромінювання. В результаті отримуємо систему нелінійних рівнянь величин виду:

$$\begin{cases} y'_{n1} = S'_n \ln\left(\left(\Delta x / x_k\right)+1\right)+\Delta y'; \\ y'_{n2} = S'_n \ln\left(\left(x_0 / x_k\right)+1\right)+\Delta y'; \\ y'_{n3} = S'_{\text{н}} \ln\left(\left(\left(x_0+\Delta x\right)/x_k\right)+1\right)+\Delta y'; \\ y'_{n4} = S'_{\text{н}} \ln\left(\left(x_i / x_k\right)+1\right)+\Delta y'; \\ y'_{n5} = S'_{\text{н}} \ln\left(\left(\left(x_i+\Delta x\right)/x_k\right)+1\right)+\Delta y'. \end{cases} \quad (2.2.15)$$

де y'_{ni} – сигнал на виході сенсора в кожному i-му (i=(1÷5)) такті вимірювання.

Слід зазначити, що всі i такти вимірювання (для логарифмічної функції перетворення i=(1÷5)) складають 1 цикл вимірювання.

В результаті рішення цієї системи відносно шуканої величини x_i, отримуємо вираз:

$$x_i = \frac{\Delta x}{\left[\left(\Delta x/\left(\left(x_0+x_k\right)+1\right)\right)^{(y'_{n5}-y'_{n4})/(y'_{n3}-y'_{n2})}\right]-1} - x_k. \quad (2.2.16)$$

За наявності випадкових похибок, що являють собою стаціонарний процес, рівняння надлишкових вимірювань (2.2.16) прийме вид:

$$\overline{x}_i = \frac{\Delta x}{\left[\left(\Delta x/\left(\left(x_0+x_k\right)+1\right)\right)^{(\overline{y}'_{n5}-\overline{y}'_{n4})/(\overline{y}'_{n3}-\overline{y}'_{n2})}\right]-1} - x_k. \quad (2.2.17)$$

де $\overline{y}'_{ni}$ – усереднені значення вихідних сигналів сенсора в кожному i-му (i=(1÷5)) такті вимірювання.

Слід зазначити, що в отримане рівняння (2.2.16), на відміну від рівняння (2.2.14), не входять параметри $\Delta y'$ і S'_n, тобто результат вимірювання шуканої величини x_i не залежить від параметрів ФП та їх змін. Таким чином, МНВ сприяють виключенню систематичної складової

похибки, що обумовлена нестабільністю параметрів функції перетворення. Слід зазначити, що такий результат досягається за умови, якщо за час проведення циклу вимірювань зміни параметрів залишаються сталими.

Для перевірки правильності отриманого рівняння (2.2.16) достатньо замість вихідних сигналів сенсора y'_{ni} підставити їх вирази з системи рівнянь (2.2.15). В результаті цього отримаємо рівняння надлишкових вимірювань:

$$x_i = \frac{\Delta x}{\left[\left(\Delta x \Big/ \left(\left(x_0 + x_k\right)+1\right)\right)^{\frac{\left(S'_n \ln\left((x_i+\Delta x)/x_k+1\right)+\Delta y' - S'_n \ln\left((x_0+\Delta x)/x_k+1\right)-\Delta y'\right)}{\left(S'_n \ln\left((x_i+\Delta x)/x_k+1\right)+\Delta y' - S'_n \ln\left((x_0/x_k)+1\right)-\Delta y'\right)}}\right]-1} - x_k = x_i. \quad (2.2.18)$$

Як видно з рівняння надлишкових вимірювань (2.2.18), значення шуканої величини x_i приведено ко входу. Це означає, що при використанні МНВ не потрібно проводити додаткову лінеаризацію логарифмічною функції перетворення.

Як видно з виду логарифмічної ФП сенсора (2.2.13) невідомими величинами є фізична величина, що описується через паспортні параметри сенсора (x_k) і зміщення ФП з урахуванням адитивної складової похибки ($\Delta y'$). Спочатку виводимо рівняння надлишкових вимірювань чутливості S'_n. Для цього з системи рівнянь (2.2.15) спочатку знайдемо різницю правих і лівих частин рівностей для вихідних сигналів сенсора y'_{n5} і y'_{n4}, а отриманий результат підставимо в результат різниці вихідних сигналів y'_{n3} і y'_{n2}. В результаті чого отримаємо рівняння надлишкових вимірювань чутливості у вигляді:

$$S'_n = \frac{y'_{n3} - y'_{n2}}{\ln\left[\left(\Delta x / \left(x_0 + x_k\right)\right) + 1\right]}. \qquad (2.2.19)$$

Для визначення параметру x_k спочатку знайдемо різницю правих і лівих частин рівностей для вихідних сигналів сенсора y'_{n1} і y'_{n2}, а потім підставимо в нього отримане рівняння (2.2.19). В результаті чого маємо рівняння надлишкових вимірювань фізичної величини, що описується через паспортні параметри сенсора, x_k в неявному вигляді:

$$\frac{y'_{n2} - y'_{n1}}{y'_{n3} - y'_{n2}} = \frac{\ln\left(\left(x_0 + x_k\right)/\left(\Delta x + x_k\right)\right)}{\ln\left[\left(\Delta x / \left(x_0 + x_k\right)\right) + 1\right]}. \qquad (2.2.20)$$

Оскільки рівняння (2.2.20) також отримане в неявному вигляді, то для його розв'язку застосовується метод ітерацій. У цьому випадку здійснюють перебір із заданим кроком всіх можливих дискретних значень шуканої фізичної величини x_i і фізичної величини x_k, що описується через паспортні параметри сенсора.

МНВ дають можливість визначити також зміщення логарифмічної ФП. Для цього розв'яжемо перше або друге нелінійне рівняння величин системи (2.2.15) відносно $\Delta y'$. В результаті маємо:

$$\Delta y' = y'_{n1} - S'_н \ln\left(\frac{\Delta x}{x_k} + 1\right) = y'_{n1} - (y'_{n3} - y'_{n2}) \frac{\ln\left(\frac{\Delta x}{x_k} + 1\right)}{\ln\left[\left(\Delta x / \left(x_0 + x_k\right)\right) + 1\right]} \qquad (2.2.21)$$

або

$$\Delta y' = y'_{n2} - S'_{н} \ln\left(\frac{x_0}{x_k}+1\right) = y'_{n2} - (y'_{n3} - y'_{n2}) \frac{\ln\left(\frac{x_0}{x_k}+1\right)}{\ln\left[\left(\Delta x/\left(x_0 + x_k\right)\right)+1\right]}. \quad (2.2.22)$$

Таким чином, визначення параметрів логарифмічної функції перетворення і їх відхилень від номінальних значень за визначений час, дають можливість спрогнозувати метрологічну надійність засобів вимірювання.

На основі запропонованої математичної моделі на рис. Р.2.2.5 продемонструємо блок-схема алгоритму вимірювання шуканої фізичної величини при логарифмічній функції перетворення з використанням технічного рішення. Блок-схема алгоритму застосування МНВ при логарифмічній функції перетворення сенсора аналогічна попереднім і відрізняється кількістю занесених до пам’яті кодів чисел, кількістю вимірювальних перетворень нормованих за значенням фізичних величин та діями по їх відтворенню.

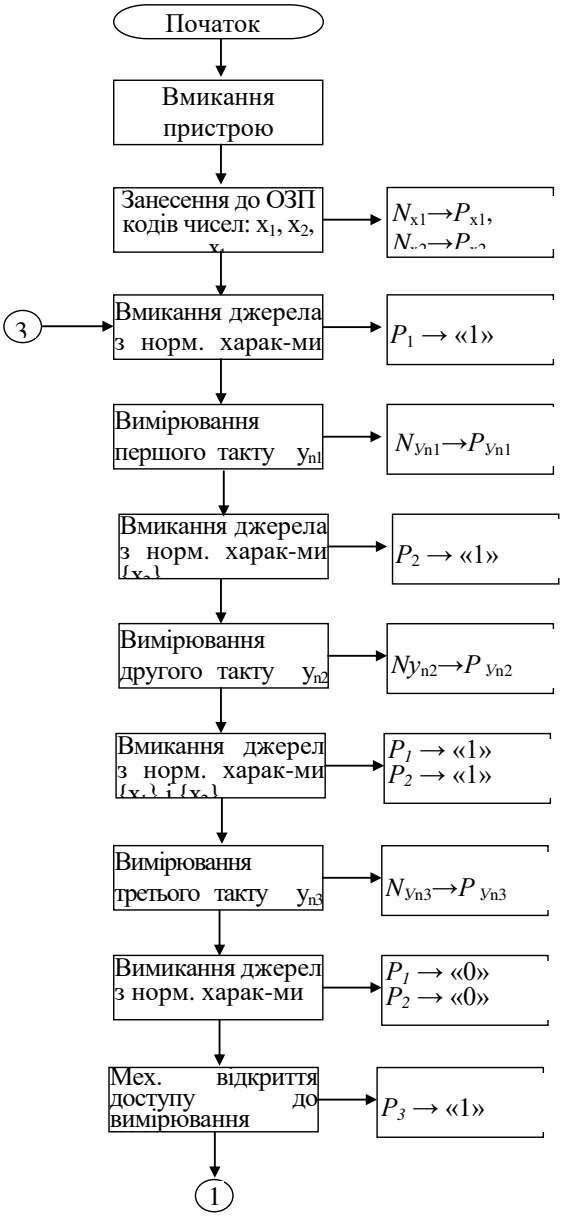


Рис. 2.2.5 – Алгоритм роботи технічного пристрою при логарифмічній ФП

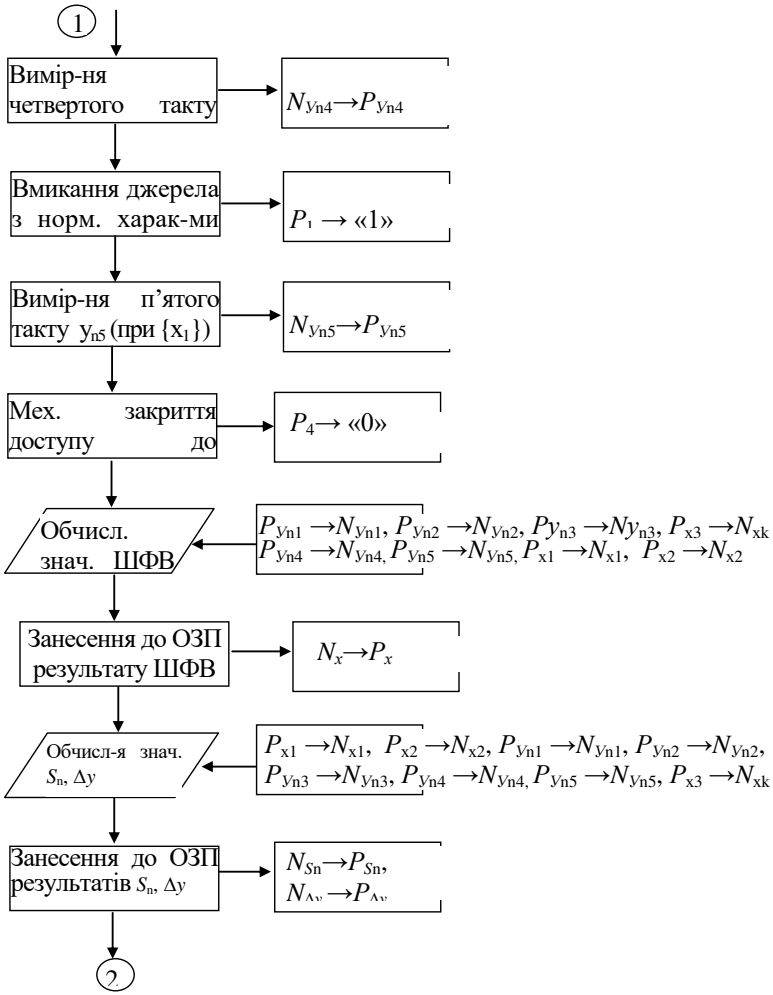


Рис. 2.2.5 (продовження) – Алгоритм роботи технічного пристрою при логарифмічній ФП

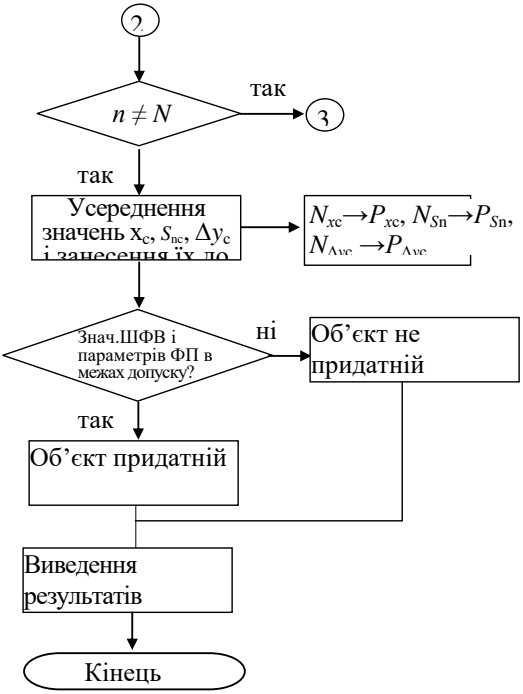


Рис. 2.2.5 (закінчення) – Алгоритм роботи технічного пристрою при логарифмічній ФП

Розглянемо ще один варіант системи нелінійних рівнянь величин, яка реалізує метод надлишкових вимірювань для логарифмічної ФП:

$$\begin{cases} y'_1 = \Delta y; \\ y'_2 = S'_{\text{н}} \ln(\dfrac{x_0}{x_k} + 1) + \Delta y = S'_{\text{н}} \ln(\dfrac{x_0 + x_k}{x_k}) + \Delta y = S'_{\text{н}}(\ln(x_k + x_0) - \ln x_k) + \Delta y; \\ y'_3 = S'_{\text{н}}(\ln(k_{\text{нl}} x_k + x_0) - \ln x_k) + \Delta y; \\ y'_4 = S'_{\text{н}}(\ln(x_k + x_i) - \ln x_k) + \Delta y, \end{cases} \qquad (2.2.23)$$

$$\begin{cases} y'_1 = \Delta y; \\ y'_2 = S'_{\text{н}} \ln(\frac{x_0}{x_k} + 1) + \Delta y = S'_{\text{н}} \ln(\frac{x_0 + x_k}{x_k}) + \Delta y = S'_{\text{н}} (\ln(x_k + x_0) - \ln x_k) + \Delta y; \\ y'_{31} = S'_{\text{н}} (\ln(k_{\text{н}2} x_k + x_0) - \ln x_k) + \Delta y; \\ y'_4 = S'_{\text{н}} (\ln(x_k + x_i) - \ln x_k) + \Delta y, \end{cases} \tag{2.2.24}$$

де $k_{\text{н}2}$ — коефіцієнт локальної лінеаризації.

Рішення системи (2.2.23) дає змогу отримати рівняння надлишкових вимірювань у вигляді:

$$x_i = x_k (e^{\frac{(y'_4 - y'_1)(\ln(k_{\text{н}1} x_0 + x_k) - \ln(x_0 + x_k)}{(y'_3 - y'_2)}} - k_1) = x'_i \tag{2.2.25}$$

та визначити ще й параметр $S'_{\text{н}}$

$$S'_{\text{н}} = \frac{y'_3 - y'_2}{\ln(k_{\text{н}1} x_0 + x_k) - \ln(x_0 + x_k)} . \tag{2.2.26}$$

В результаті рішення системи (2.2.24) можна вивести рівняння надлишкових вимірювань шуканої фізичної величини у вигляді:

$$x_i = x_k (e^{\frac{(y'_4 - y'_1)(\ln(k_{\text{н}2} x_0 + x_k) - \ln(x_0 + x_k)}{(y'_{31} - y'_2)}} - k_1) = x''_i . \tag{2.2.27}$$

Обробка результатів проміжних вимірювань згідно з рівняннями числових значень (2.2.25) і (2.2.27) забезпечує виключення впливу на результат вимірювання, абсолютних значень параметрів нелінійної функції перетворення сенсора та їх відхилень від номінальних значень і забезпечує

лінійну залежність результатів вимірювання від шуканої контрольованої величини.

Як видно з наведених математичних моделей при логарифмічній ФП в них присутня фізична величина x_k, що описується через паспортні параметри сенсора. Тож для її знаходження, було виведено рівняння надлишкових вимірювань. Оскільки рівняння надлишкових вимірювань (2.2.25) і (2.2.27) виведені для ідеального випадку, тобто у відсутності похибок вимірювання, то результат вимірювань за своїм значенням дорівнює значенню вхідної ФВ. Оскільки вимірювання проводилися для однієї і тієї ж шуканої величини x_i, той ділення x_i'' на x_i' у результаті дає одиницю:

$$\frac{x_i''}{x_i'} = 1 = \frac{e^{\frac{(y_4'-y_1')(\ln(k_{н2}x_0+x_k)-\ln(x_0+x_k)}{(y_3'-y_2')}} - k_1}{e^{\frac{(y_4'-y_1')(\ln(k_{н1}x_0+x_k)-\ln(x_0+x_k)}{(y_{31}'-y')}} - k_1}. \qquad (2.2.28)$$

Приведемо отриману рівність до виду:

$$\frac{(y_4'-y_1')}{(y_3'-y_2')}\ln(k_{н1}x_0 + x_k) - \left(\frac{(y_4'-y_1')}{(y_3'-y_2')} - \frac{(y_4'-y_1')}{(y_{31}'-y_2')}\right)\ln(x_0 + x_k) - \\ - \frac{(y_4'-y_1')}{(y_3'-y_2')}\ln(k_{н2}x_0 + x_k) = 0. \qquad (2.2.29)$$

Отриманий аналітичний вираз в неявному виді зв'язує результат вимірювань фізичної величини x_0 з фізичною величиною x_k. Визначення значення x_k здійснюється методом ітерацій до одержання нульового значення в лівій частині аналітичного виразу (2.2.29). Для знаходження

шуканої величини x_i знайдене значення величини x_k підставляють в одне з рівнянь надлишкових вимірювань (2.2.25) чи (2.2.27).

2.2.5. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ КВАДРАТИЧНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА

Представимо розробку математичної моделі методів надлишкових вимірювань при квадратичній функції перетворення.

При квадратичній ФП сенсора за умови можливості формування нормованих за значенням фізичних величин x_1 $\{x_1\}=(\{x0\}-\{\Delta x\})$ і x_2 $(\{x_2\}=\{x0\}+\{\Delta x\})$ таким чином, щоб $\{x_1\}+\{x_2\}=k_2\{x0\}$, а $\{x_2\}-\{x_1\}=k_2\{\Delta x\}$ (k_2 – коефіцієнт пропорційності, $k_2=2$). Система нелінійних рівнянь величин прийме вид:

$$\begin{cases} y'_{н1} = \Delta y'; \\ y'_{н2} = S'_н x_1^2 + S'_л x_1 + \Delta y'; \\ y'_{н3} = S'_н x_2^2 + S'_л x_2 + \Delta y'; \\ y'_{н4} = S'_н (x_i + x_1)^2 + S'_л (x_i + x_1) + \Delta y'; \\ y'_{н5} = S'_н (x_i + x_2)^2 + S'_л (x_i + x_2) + \Delta y', \end{cases} \qquad (2.2.30)$$

де $y'_{ні}$ — вихідний сигнал сенсора в кожному i-му такті вимірювань ($i=1 \div 5$).

В результаті рішення системи когерентних нелінійних рівнянь величин (2.2.30), отримуємо відповідні рівняння надлишкових вимірювань:

$$x_i = (x_1\, x_2)\frac{(y'_{н5} - y'_{н4}) - (y'_{н3} - y'_{н2})}{(x_1 + x_2)(y'_{н3} - y'_{н2}) - (x_2 - x_1)(y'_{н2} + y'_{н3} - k_2 y'_{н1})}. \qquad (2.2.31)$$

Рівняння надлишкових вимірювань (2.2.31) дає лінійну залежність результату вимірювань від значення шуканої фізичної величини. Завдяки операції віднімання значень вихідних сигналів сенсора у чисельнику та знаменнику виключається адитивна складова похибки вимірювання, а завдяки операції ділення виключається мультиплікативна складова систематичної похибки.

У випадку, коли наявні випадкові похибки, що являють собою стаціонарний процес, рівняння надлишкових вимірювань (2.2.31) прийме вид:

$$\overline{x}_i = (\mathrm{x}_1\, x_2)\frac{(\overline{y}'_{\mathrm{н}5} - \overline{y}'_{\mathrm{н}4}) - (\overline{y}'_{\mathrm{н}3} - \overline{y}'_{\mathrm{н}2})}{(\mathrm{x}_1 + \mathrm{x}_2)\left(\overline{y}'_{\mathrm{н}3} - \overline{y}'_{\mathrm{н}2}\right) - (\mathrm{x}_2 - \mathrm{x}_1)\left(\overline{y}'_{\mathrm{н}2} + \overline{y}'_{\mathrm{н}3} - k_2\overline{y}'_{\mathrm{н}1}\right)}, \quad (2.2.32)$$

де $\overline{y}'_{ni}$ – усереднені значення вихідних сигналів сенсора в кожному *i*-му (i=(1÷5)) такті вимірювання.

Крім того, результат вимірювання не залежить від значень параметрів $S'_{\mathrm{н}}$, $S'_{\mathrm{л}}$ нелінійної функції перетворення та їх відхилень від номінальних значень.

Значення параметрів крутостей перетворення $S'_{\mathrm{н}}$, $S'_{\mathrm{л}}$ нелінійної функції перетворення для системи (2.2.28) визначають відповідно по наступним виразам:

$$S'_{\mathrm{н}} = \frac{x_2(y'_{\mathrm{н}2} - y'_{\mathrm{н}1}) - x_1(y'_{\mathrm{н}3} - y'_{\mathrm{н}1})}{x_1x_2(x_1 - x_2)}, \quad (2.2.33)$$

$$S'_{\mathrm{л}} = \frac{(y'_{\mathrm{н}1} - y'_{\mathrm{н}2})\mathrm{x}_2^2 + (y'_{\mathrm{н}3} - y'_{\mathrm{н}1})\mathrm{x}_1^2}{x_2x_1(x_1 - x_2)}. \quad (2.2.34)$$

Визначення параметрів $S'_{\text{н}}$, $S'_{\text{л}}$ нелінійної функції перетворення при кожному вимірюванні забезпечує встановленню характеру їх зміни у часі, що дає можливість визначити час виходу цих параметрів за довірчі інтервали.

На основі запропонованої математичної моделі на рис. Р.2.2.6 продемонструємо блок-схема алгоритму вимірювання шуканої фізичної величини при квадратичній функції перетворення з використанням технічного рішення.

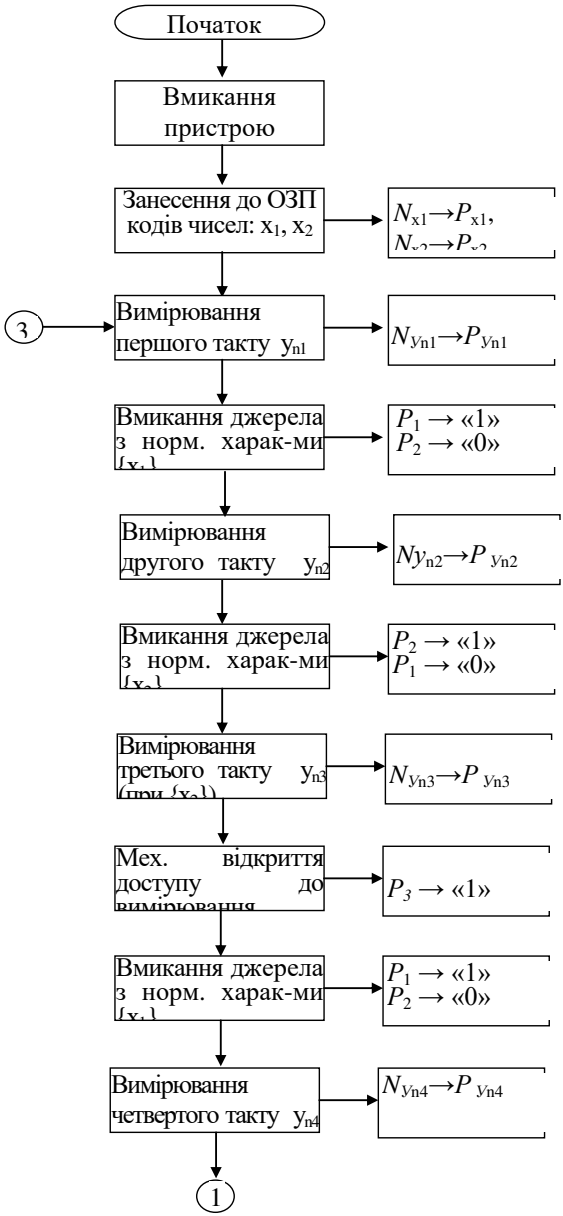


Рис. 2.2.6 – Алгоритм роботи технічного пристрою при квадратичній ФП

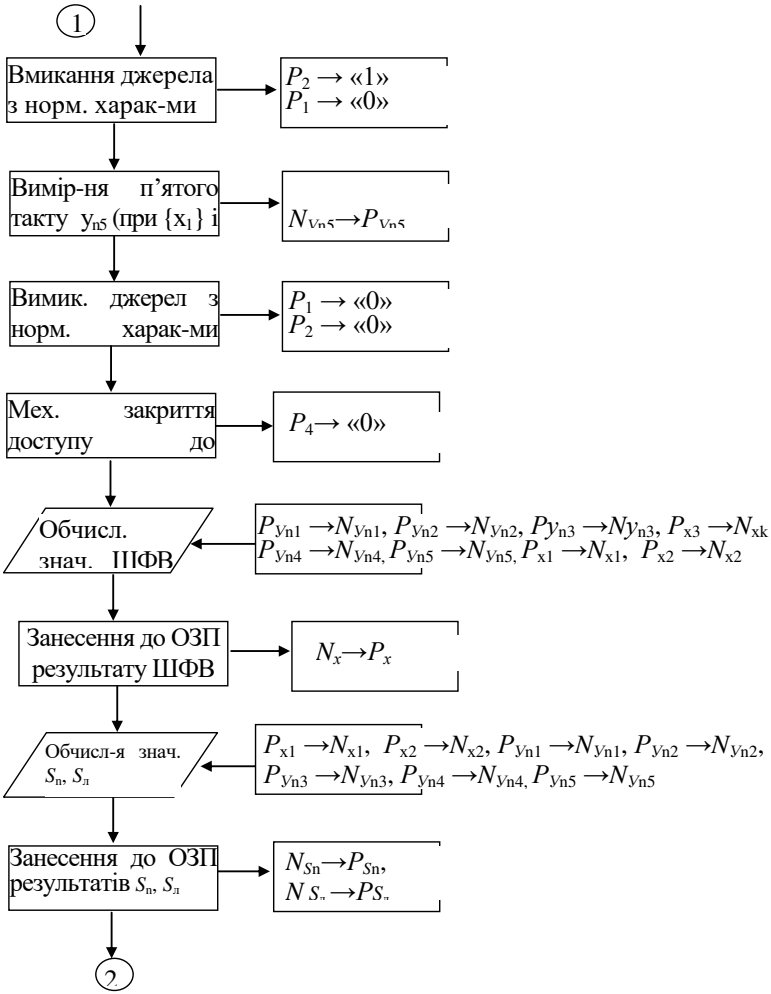


Рис. 2.2.6 (продовження) – Алгоритм роботи технічного пристрою при квадратичній ФП

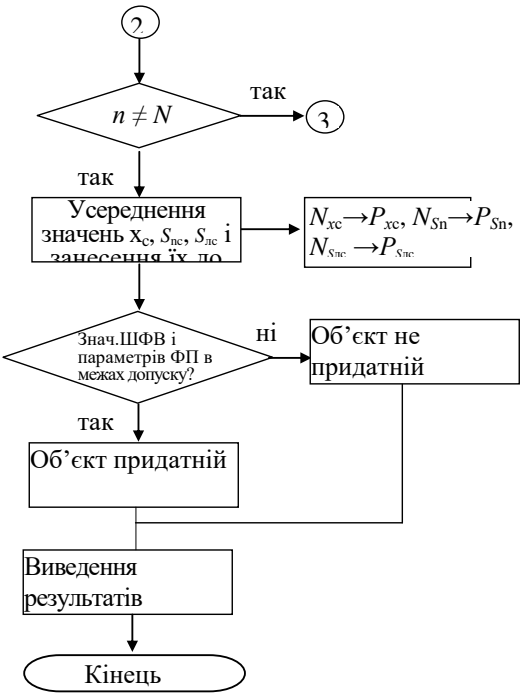


Рис. 2.2.6 (закінчення) – Алгоритм роботи технічного пристрою при квадратичній ФП

Таким чином, блок-схема алгоритму застосування МНВ при квадратичній функції перетворення сенсора аналогічна попереднім і відрізняється кількістю занесених до пам'яті кодів чисел та діями по їх відтворенню. Базуючись на принципах спадкоємності принципу роботи алгоритмів МНВ, розглянемо математичну при кубічній функції перетворення.

2.2.6. ПОБУДОВА АЛГОРИТМІВ РОБОТИ МЕТОДІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ ПРИ КУБІЧНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ СЕНСОРА

Розглянемо математичну модель методів надлишкових вимірювань при кубічній функції перетворення за умови можливості формування нормованих за значенням фізичних величин x_1 $\{x_1\}=(\{x0\}-\{\Delta x\})$ і x_2 $(\{x_2\}=\{x0\}+\{\Delta x\})$ таким чином, щоб $\{x_1\}+\{x_2\}=k_2\{x0\}$, а $\{x_2\}-\{x_1\}=k_2\{\Delta x\}$ (k_2 – коефіцієнт пропорційності, $k_2=2$). Слід зазначити, що ряди потоків випромінення, які використовувались при кубічній функції перетворення, ідентичні тим, що були використані при квадратичній функції перетворення.

Система нелінійних рівнянь величин прийме вид:

$$\begin{cases} y'_1 = \Delta y'; \\ y'_2 = S'_{\text{н}} x_1^3 + S'_{\text{л}} x_1 + \Delta y'; \\ y'_3 = S'_{\text{н}} x_2^3 + S'_{\text{л}} x_2 + \Delta y'; \\ y'_4 = S'_{\text{н}} \left(x_i + x_1 \right)^3 + S'_{\text{л}} \left(x_i + x_1 \right) + \Delta y'; \\ y'_5 = S'_{\text{н}} \left(x_i + x_2 \right)^3 + S'_{\text{л}} \left(x_i + x_2 \right) + \Delta y', \end{cases} \qquad (2.2.35)$$

де $y'_{ні}$ — вихідний сигнал сенсора в кожному i-му такті вимірювань $i=(1 \div 5)$.

Рішення системи нелінійних рівнянь величин (2.2.35) дає можливість отримані відповідне рівняння надлишкових вимірювань шуканої фізичної величини в явному вигляді ($k_3=3$ — коефіцієнт пропорційності):

$$x_i = \frac{(x_2 + x_1)}{k_2} \left[\frac{k_3 (x_2 - x_1)(y'_5 + y'_4 - k_2 y'_1)}{(x_2 + x_1)((y'_5 - y'_4) - (y'_3 - y'_2)) + k_3 (x_2 - x_1)(y'_2 + y'_3 - k_2 y'_1)} - 1 \right]. \qquad (2.2.36)$$

Рівняння надлишкових вимірювань (2.2.36) дає лінійну залежність результату вимірювань від шуканої фізичної величини. Крім того, результат вимірювання не залежить від значень параметрів $S'_{\text{н}}, S'_{\text{л}}$ нелінійної функції перетворення та їх відхилень від номінальних значень.

Крім того, рішення системи (2.2.35) дає змогу визначити значення крутостей перетворення по наступним виразам:

$$S'_{\text{н}} = \frac{x_2(\text{y}'_2 - \text{y}'_1) - x_1(\text{y}'_3 - \text{y}'_1)}{x_2 x_1(\text{x}_1^2 - x_2^2)} \tag{2.2.37}$$

$$S'_{\text{л}} = \frac{(\text{y}'_1 - \text{y}'_2)\,\text{x}_2^3 + (\text{y}'_3 - \text{y}'_1)\,\text{x}_1^3}{x_1 x_2(\text{x}_1^2 - x_2^2)}. \tag{2.2.38}$$

Визначення параметрів $S'_{\text{н}}, S'_{\text{л}}$ нелінійної функції перетворення при кожному вимірюванні забезпечує встановленню характеру їх зміни у часі, що дає можливість визначити час виходу цих параметрів за довірчі інтервали та спрогнозувати метрологічну надійність засобу вимірювання, які реалізує зазначену математичну модель методу надлишкових вимірювань.

У випадку, коли наявні випадкові похибки, що являють собою стаціонарний процес, рівняння надлишкових вимірювань (2.2.36) прийме вид:

$$\bar{x}_i = \frac{(\text{x}_2 + x_1)}{k_2}\left[\frac{k_3(\text{x}_2 - x_1)(\bar{\text{y}}'_5 + \bar{\text{y}}'_4 - k_2\,\bar{\text{y}}'_1)}{(\text{x}_2 + x_1)((\bar{\text{y}}'_5 - \bar{\text{y}}'_4) - (\bar{\text{y}}'_3 - \bar{\text{y}}'_2)) + k_3(\text{x}_2 - x_1)(\bar{\text{y}}'_2 + \bar{\text{y}}'_3 - k_2\,\bar{\text{y}}'_1)} - 1\right]. \tag{2.2.39}$$

Блок-схема алгоритму застосування МНВ при кубічній функції перетворення сенсора аналогічна квадратичній.

В представлених математичних моделей і відповідних блок-схемах алгоритмів наявні можливості проведення багатократних вимірювань. Розглянемо більш детальніше особливості застосування МНВ при багатократних вимірюваннях.

2.2.7. ПОБУДОВА АЛГОРИТМІВ РОБОТИ БАГАТОКРАТНИХ НАДЛИШКОВИХ ВИМІРЮВАНЬ

Вплив дестабілізуючих факторів на сенсор приводить до появи систематичних і випадкових складових похибки вимірювання. Як було зазначено вище (див. розділ 1.4), МНВ І-го роду забезпечують виключення систематичних складових похибки результату вимірювання, спричинених впливом на нього абсолютних значень параметрів функції перетворення і їх відхилень від номінальних значень. МНВ І-го роду доцільно використовувати при наявності вимірювальних каналів із фільтрами верхніх чи нижніх частот, смуговими фільтрами, інтеграторами тощо, які забезпечують зменшення випадкової складової похибки вимірювального перетворення.

Для зменшення випадкової складової похибки вимірювання застосовуються надлишкові вимірювання ІІ-го і ІІІ-го роду. МНВ ІІ-го роду застосовуються у випадках, коли випадкова складова похибки представляє собою стаціонарний ергодичний процес. МНВ ІІІ-го роду використовуються, як правило, у випадках, коли випадкова складова похибки представляє собою нестаціонарний процес і її математичне очікування змінюється за періодичним законом через (таким наприкладом може слугувати мережевий вплив 50 Гц, 100 Гц).

До сьогодні найбільш ефективними є методи статистичної обробки результатів багаторазових вимірювань. У багатьох випадках перевагу відають методам обробки обмежених за кількістю даних з використанням критерію Стьюдента.

Розглянемо рівняння надлишкових вимірювань при лінійній функції перетворення сенсора (2.2.3). У цьому випадку на основі рівняння надлишкових вимірювань II-го роду (2.1.61) при лінійній ФП буде мати вигляд:

$$x_i = x_1 \frac{\left(\frac{k_1}{n} \sum_{i=1}^{n} y'_{l3i} - \frac{k_1}{n} \sum_{i=1}^{n} y'_{l1i} \right)}{\left(\frac{k_1}{n} \sum_{i=1}^{n} y'_{l2i} - \frac{k_1}{n} \sum_{i=1}^{n} y'_{l1i} \right)}, \qquad (2.2.40)$$

де k_1 – пропорційний коефіцієнт (k_1=1).

Відомо [35], що усереднення результатів проміжних вимірювань може бути здійснено двома шляхами. При першому підході кожен так вимірюється n разів ($9 \le n \le 34$), а обробка результатів проміжних вимірювань здійснюється згідно з рівнянням надлишкових вимірювань (2.2.40). Як видно з рівняння надлишкових вимірювань (2.2.40), кількість результатів багаторазових вимірювань є однаковою. Можливі випадки, коли окремі такти вимірюються більшою або меншою кількістю разів, наприклад, визначення вихідних напруг сенсора $y_{л1}$ може здійснюватися n_1 раз, $y_{л2}$ — n_3 рази, а $y_{л3}$ — n_2 рази. У цьому випадку рівняння надлишкових вимірювань температури буде таким:

$$x_i = x_1 \frac{\left(\frac{k_1}{n3} \sum_{i=1}^{n3} y'_{l3i} - \frac{k_1}{n1} \sum_{i=1}^{n1} y'_{l1i} \right)}{\left(\frac{k_1}{n2} \sum_{i=1}^{n2} y'_{l2i} - \frac{k_1}{n1} \sum_{i=1}^{n1} y'_{l1i} \right)}. \qquad (2.2.41)$$

При застосуванні другого підходу методу надлишкових вимірювань ІІ-го роду здійснюють n циклів вимірювань з одноразовими вимірюваннями в кожному такті. У цьому випадку на основі рівняння надлишкових вимірювань ІІ-го роду (2.1.63) при лінійній ФП буде мати вигляд:

$$x_i = x_1 \frac{k1}{n} \sum_{i=1}^{n} \frac{\left(y'_{l3\text{i}} - y'_{l1\text{i}}\right)}{\left(y'_{l2\text{i}} - y'_{l1\text{i}}\right)}. \qquad (2.2.42)$$

Перевагу слід віддати другому підходу, оскільки виключається систематична складова похибки результату вимірювання, що обумовлені зміною параметрів функції перетворення.

МНВ ІІІ-го роду використовують, коли випадкова складова похибки являє собою випадковий нестаціонарний процес. Тоді обробка результатів багаторазових вимірювань здійснюється за методом ІІ-го роду ще m разів. Рівняння надлишкових вимірювань температури приймуть такий вигляд:

$$x_i = \frac{k1}{m} \sum_{j=1}^{m} \left(x_1 \frac{\left(\frac{k_1}{n} \sum_{i=1}^{n} y'_{l3\text{i}} - \frac{k_1}{n} \sum_{i=1}^{n} y'_{l1i} \right)}{\left(\frac{k_1}{n} \sum_{i=1}^{n} y'_{l2i} - \frac{k_1}{n} \sum_{i=1}^{n} y'_{l1i} \right)} \right)_j, \qquad (2.2.43)$$

$$x_i = \frac{k1}{m} \sum_{j=1}^{m} \left(x_1 \frac{k1}{n} \sum_{i=1}^{n} \frac{\left(y'_{l3\text{i}} - y'_{l1\text{i}}\right)}{\left(y'_{l2\text{i}} - y'_{l1\text{i}}\right)} \right)_j, \qquad (2.2.44)$$

де вираз (2.2.43) — рівняння надлишкових вимірювань для МНВ ІІІ-го роду з використанням МНВ ІІ-го роду при першому підході;

вираз (2.2.44) — рівняння надлишкових вимірювань для МНВ ІІІ-го роду з використанням МНВ ІІ-го роду при другому підході.

МНВ забезпечують широкі можливості не тільки щодо виключення систематичної похибки (що обумовлені зміною параметрів функції перетворення), а і відносно обробки результатів проміжних вимірювань при наявності випадкових похибок. Як видно з наведених рівнянь надлишкових вимірювань, МНВ ІІ-го і ІІІ-го роду можуть бути використані за різним числом вимірювань у кожному такті, що свідчить про їх гнучкість. На сьогодні достатньо актуальною є задача коректного застосування методів обробки результатів проміжних вимірювань за Стьюдентом при невідомих законах розподілення випадкової похибки.

Слід зазначити, що МНВ можна використовувати при будь-якому виді ФП, що не можуть забезпечити існуючі методи, для яких обов’язкова лінеаризація ФП. Продемонструємо це на прикладі логарифмічної ФП. При наявності випадкових похибок, що представляють собою стаціонарний процес, рівняння надлишкових вимірювань буде мати вид (2.2.17).

У випадку використання МНВ ІІ-го роду, коли проводять n циклів вимірювань при одноразових вимірюваннях в кожному такті, рівняння надлишкових вимірювань матиме такий вигляд:

$$x_i = \frac{k1}{n}\sum_{i=1}^{n}\left(\frac{\Delta x}{\left[\left(\Delta x/\left(\left(x_0+x_k\right)+1\right)\right)^{(y'_{n5i}-y'_{n4i})/(y'_{n3i}-y'_{n2i})}\right]-1}-x_k\right). \quad (2.2.45)$$

Вище зазначалося, що якщо відомо, що випадкова складова похибки представляє собою випадковий нестаціонарний процес, то використовуються МНВ ІІІ-го роду. У цьому випадку рівняння

надлишкових вимірювань температури при логарифмічній функції перетворення будуть такими:

$$x_i = \frac{k1}{m}\sum_{j=1}^{m}\left(\frac{\Delta x}{\left[\left(\Delta x \Big/ \left(\left(x_0 + x_k\right)+1\right)\right)^{\left(\frac{k1}{n}\sum_{i=1}^{n} y_{n5i} - \frac{k1}{n}\sum_{i=1}^{n} y_{n4i}\right) \Big/ \left(\frac{k1}{n}\sum_{i=1}^{n} y_{n3i} - \frac{k1}{n}\sum_{i=1}^{n} y_{n2i}\right)}\right] - 1} - x_k\right)_j \quad (2.2.46)$$

та

$$x_i = \frac{k1}{m}\sum_{j=1}^{m}\left(\frac{k1}{n}\sum_{i=1}^{n}\left(\frac{\Delta x}{\left[\left(\Delta x \Big/ \left(\left(x_0 + x_k\right)+1\right)\right)^{(y'_{n5i} - y'_{n4i})/(y'_{n3i} - y'_{n2i})}\right] - 1} - x_k\right)\right)_j . \quad (2.2.47)$$

Вираз (2.2.46) представляє собою рівняння надлишкових вимірювань для МНВ ІІІ-го роду з використанням МНВ ІІ-го роду при першому підході, а вираз 2.2.47) – за при другому підході. Надлишкові вимірювання ІІІ-го роду в цілому забезпечують автоматичне виключення систематичної складової, що обумовлені зміною параметрів функції перетворення та зменшення випадкової складової похибки результату вимірювання, що є нестаціонарним процесом.

Таким чином, при наявності випадкової похибки використовуються МНВ ІІ-го і ІІІ-го родів. Використання операцій визначення середніх значень вихідних сигналів сенсора в МНВ ІІ-го та ІІІ-го родів забезпечує зменшення в (3 ÷ 3,8) раз випадкову складову похибки вимірювання.

Блок-схема алгоритму роботи засобу вимірювання при лінійній функції перетворення МНВ ІІ-го роду по другому підході вже була представлена рис. 2.2.3. Тож нижче на рис. 2.2.7 представлено блок-схему

алгоритму роботи засобу вимірювання шуканої фізичної величини МНВ ІІ-го роду по першому підході при лінійній функції перетворення. Зазначимо, що перший підхід відрізняється послідовністю і обробкою вимірювань. При вимірювальному контролі спочатку проводять багаторазові вимірювання контрольованого параметру, а потім отриманий результат порівнюють з нормою (допуском). Згідно з запропонованим алгоритмом роботи пристрою, після його вмикання проводиться занесення в пам'ять кодів чисел: нормованого за значенням фізичної величини x_1 і допуску. Далі проводиться N вимірювань з наступним усередненням результатів по першому підході.

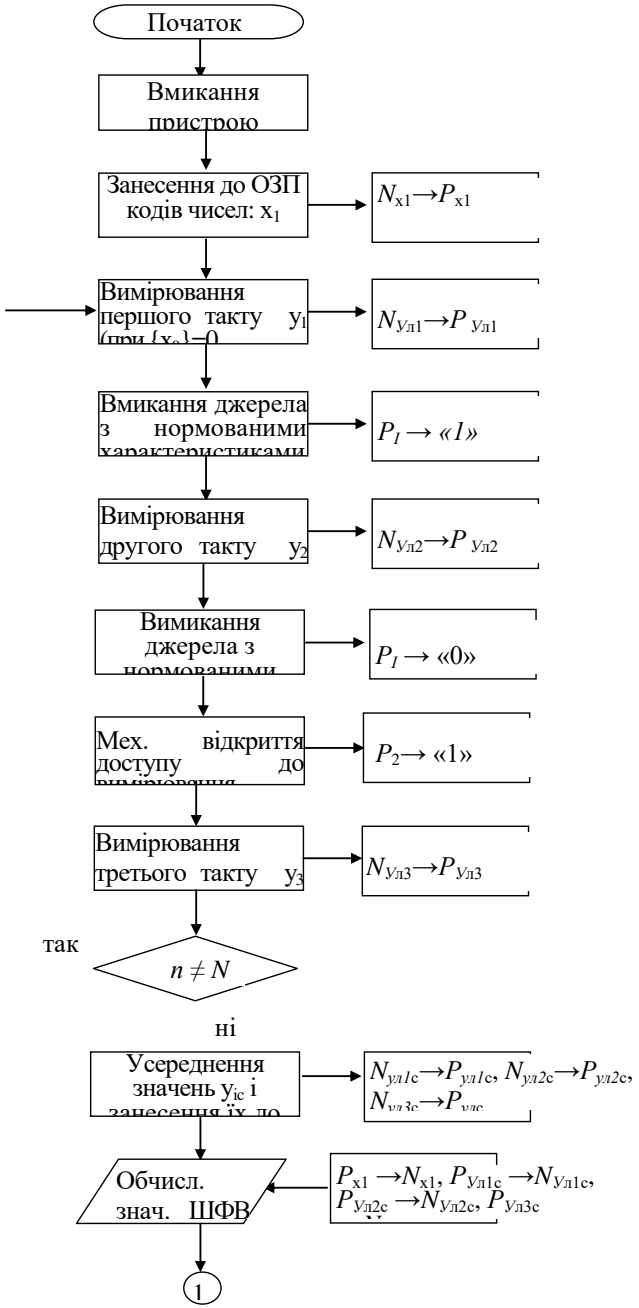


Рис. 2.2.7 – Алгоритм роботи технічного пристрою при лінійній ФП

Рис. 2.2.7 – Алгоритм роботи технічного пристрою при лінійній ФП

Показано, що структура технічних пристроїв, що реалізують МНВ, суттєво залежить від виду функції перетворення, системи математичних моделей, рівня інтелектуалізації і інших факторів. Оптимальними являються структури з мікроконверторами.

2.3. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЗАПРОПОНОВАНИХ ПІДХОДІВ ОБРОБКИ РЕЗУЛЬТАТІВ НАДЛИШКОВИХ ВИМІРЮВАНЬ

Для вирішення задачі по підвищенню точності вимірювання при нестабільній функції перетворення були запропоновані методи надлишкових вимірювань [36 – 39].

Використовуючи отримані у другому розділі рівняння, проведемо комп'ютерне моделювання МНВ і методу прямих вимірювань при лінійної і полиномиальной ФП. На підставі отриманих результатів, проведемо порівняльний аналіз цих методів в області точності вимірювань і можливості здійснення самоконтролю параметрів ФП.

2.3.1. ПОРІВНЯЛЬНИЙ АНАЛІЗ ПОХИБОК ВИМІРЮВАНЬ МНВ І ПРЯМОГО МЕТОДА ПРИ ЛІНІЙНІЙ І ПОЛІНОМІАЛЬНІЙ ФУНКЦІЇ ПЕРЕТВОРЕННЯ

Проведемо комп'ютерне моделювання при лінійної і полиномиальной ФП, що дозволить оцінити і зробити порівняльний аналіз похибок визначення ІФВ за допомогою МИИ і прямого методів. В якості програмного середовища, як засобу для математичного моделювання і аналізу поведінки сенсора при різних вхідних даних, було обрано Mathcad15.0.

1) Дослідження лінійної ФП

При застосуванні МНВ необхідно сформувати і провести як мінімум 1 цикл вимірювань, який складається, відповідно до системи (2.2.2), з 3 проміжних вимірювань вихідного сигналу сенсора. Для

проведення комп'ютерного моделювання задамо наступні вихідні дані для порівнюваних підходів: $S_l = 2{,}0$; $\Delta y = 0{,}1$; $x_1 = 1{,}8$; $x_i = 2{,}2$.

Слід зазначити, що нормована за значенням фізична величина x_1 задається такою, щоб вона була одного порядку з шуканою фізичною величиною x. Крім того, задамо похибку відтворення параметра x1, яка буде становити ±0,1%. Така похибка обумовлена тим, що нормована величина x_1, як правило, формується за допомогою каліброваного джерела високої точності. Також поставимо межі змін для кожного з параметрів ФП (S'_l, $\Delta y'$): 1) при зміні параметів в межах ± 1,0%, 2) при зміні параметів в межах ± 10,0%. При цьому, слід зазначити, що за час проведення проміжних тактів вимірювань (1 циклу вимірювань) значення параметрів ФП і їх відхилень залишалися незмінними.

В результаті отримаємо такі відносні похибки (δ, %) вимірювань шуканої фізичної величини, які наведені в табл.2.3.1:

Табл. 2.3.1
Відносні похибки вимірювання

	δ, % при зміні параметрів (S'_l, $\Delta y'$) в межах ± 1,0%	δ, % при зміні параметрів (S'_l, $\Delta y'$) в межах ± 10,0%
Прямий метод	(1,00÷1,02)	(10,00÷10,23)
МНВ	0,10	0,10

Порівняльний аналіз отриманих результатів показує перевагу МНВ перед прямим методом в області підвищення точності вимірювання за рахунок виключення впливу відхилень параметрів ФП та їх відхилень від номінальних значень на результат вимірювання. Показано, що, на відміну від прямих методів, збільшення відхилень параметрів ФП в межах ±10% не призводит до змін результатів, отриманих при використанні МНВ. Така особливість досягається завдяки обробці проміжних вимірювань за рівнянням надлишкових вимірювань (2.2.3). Слід зазначити, що значення відносної похибки вимірювання шуканої фізичної величини, отриманої при

застосуванні МНВ, відповідає значенню похибки відтворення нормованої за значенням фізичної величини x_1. Так, збільшення похибки відтворення до 0,5% призведе до збільшення відносної похибки вимірювання шуканої фізичної величини також до 0,5%. Таким чином, методична похибка МИИ обумовлена похибкою відтворення нормованих величин, що, в свою чергу, висуває підвищені вимоги до джерела нормованих за значенням фізичних величин.

Розглянемо можливість визначення параметрів ФП. Для цього використовуємо рівняння параметрів ФП (2.2.4). Даний підхід використовується, якщо зовнішні умови нестабільні або, коли необхідно перевірити значення параметрів на відхилення від їх номінальних значень. В цьому випадку потрібно проводити кілька циклів вимірювань, оскільки може відбуватися зміна параметрів ФП в різних межах.

Розглянемо наступні випадки:

1. Зміна параметрів S'_l та $\Delta y'$ відбувається в межах ± 1,0%.
2. Зміна параметрів S'_l та $\Delta y'$ відбувається в межах ± 10,0%.
3. Зміна параметра S'_l відбувається в межах ± 1,0%, а параметра $\Delta y'$ відбувається в межах ± 10,0%.
4. Зміна параметра S'_l відбувається в межах ± 10,0%, параметра $\Delta y'$ в межах ± 1,0%.

Для визначення параметрів ФП при нестабільній ФП необхідно провести розрахунки за рівняннями (2.2.4) в кожному циклі вимірювання, усереднити їх і, визначивши різницю між розрахованим значенням і номінальним, порівняти з дозволеними нормами кожного з параметрів.

При розрахунках вважалося, що за час проведення проміжних тактів вимірювань значення параметрів ФП і їх відхилень залишалися незмінними. Так, при раніше заданих даних, отримуємо наступне значення зміни параметра S'_l:

1. При зміні параметрів S'_l та $\Delta y'$ в межах ± 1,0% отримують відхилення параметра S'_l, що дорівнює 0,00002 вимірювальних одиниць (або 0,001% від номінального значення).

2. При зміні параметрів S'_l та $\Delta y'$ в межах ± 10,0% отримують відхилення параметра S'_l, що дорівнює 0,0002 вимірювальних одиниць (або 0,01% від номінального значення).

3. При зміні параметра S'_l в межах ± 1,0%, а $\Delta y'$ в межах ± 10,0%, отримують відхилення параметра S'_l, що дорівнює 0,05 вимірювальних одиниць (або 2,5% від свого номінального значення), відносна похибка прямого метода будет складати (0,77÷1,23)%, а похибка МНВ – 0,10%.

4. При зміні параметра S'_l в межах ± 10,0%, а $\Delta y'$ в межах ± 1,0%, отримують відхилення параметра S'_l, що дорівнює 0,12 вимірювальних одиниць (або 6% від свого номінального значення), відносна похибка прямого метода будет складати (9,98÷10,02)%, а похибка МНВ – 0,10%.

Оскільки рівняння для параметра $\Delta y'$ має досить простий вигляд, то в цьому випадку досить просто провести усереднення отриманих результатів проміжних вимірювань. В результаті отримують, що відхилення параметра $\Delta y'$ при заданих даних може досягати 3 % від свого номінального значення.

З отриманих даних видно, що відхилення параметра S'_l вагоміше позначається на результаті прямих вимірювань, ніж відхилення параметра $\Delta y'$. На відміну від прямих методів, МНВ дозволяють працювати в такому режимі з високою точністю (за умови, якщо за час проведення циклу зміни параметрів залишаються незмінними). Крім того, МНВ дозволяють визначати значення параметрів ФП. Так, якщо значення параметра виходить

за допустимі межі, то можна зробити висновок про зміщення ФП і заміні датчика.

2) Дослідження поліноміальної ФП

При класичному підході вимірювань при нелінійній ФП, як правило, необхідно виразити складну залежність у(хі) вихідного сигналу від вхідного простої лінійної формулою. Для цього вхідний діапазон розбивають на лінійні ділянки, що призводить до появи додаткових похибок від нелінійності, або ж обмежуються роботою на обраній лінійній ділянці, що призводить до звуження вхідного діапазону. На відміну від класичного методу, представлений МНВ дозволяє виключити ці недоліки за рахунок рівняння надлишкових вимірювань (2.2.7). В даному рівнянні за рахунок виключення впливу на результат вимірювань параметрів нелінійної ФП надається можливість працювати на всьому діапазоні вхідного сигналу з високою точністю без розбивки діапазону на лінійні ділянки.

Для проведення комп'ютерного моделювання були обрані ті ж вихідні дані, що і при лінійної ФП, але з наступними додатковими параметрами: S_l = 1,5; S_{n2} = 2; S_{n1} = 3. При застосуванні прямого методу для поліноміальної ФП був увзятий піддіапазон вхідної величини x_i=(1,98÷2,42), похибка від лінеаризації якого становить 1%. Таким чином, така лінеаризація призводить до звуження діапазону вимірювання або його розбиття на лінійні ділянки, що збільшує розрахунково-часові витрати. Для порівняльного аналізу прямого методу і МНВ також розглядалося 2 випадки: при зміні параметрів ФП в межах ± 1,0% і в межах ± 10,0%. При цьому, слід заначити, що за час проведення проміжних тактів вимірювань (1 циклу вимірювань) значення параметрів ФП і їх відхилення залишалися

незмінними. В результаті отримаємо наступні значення відносної похибки (δ,%) вимірювання шуканої фізичної величини, які наведені в табл. 2.3.2:

Табл. 2.3.2
Відносні похибки вимірювання

	δ, % при зміні параметрів (S'_l, $\Delta y'$) в межах ± 1,0%	δ, % при зміні параметрів (S'_l, $\Delta y'$) в межах ± 10,0%
Прямий метод	1,02	(1,02÷10,23)
МНВ	0,10	0,10

Як і при лінійної ФП надлишкові методи показали хороші результати щодо зменшення похибки, викликаної нестабільністю і нелінійністю ФП датчика. Показано, що застосування МНВ дозволяє на порядок підвищити точність вимірювання в порівнянні з прямими методами. Крім того, МНВ не вимагають роботи на лінійній ділянці або розбиття на піддіапазони вхідного сигналу, оскільки, завдяки рівняння надлишкових вимірювань (2.2.7) на виході отримують сигнал такий, як і на вході. Методична похибка МИИ, як і при лінійної ФП, визначається похибкою відтворення нормованих за значенням величин x_1 x_2, x_3, x_4.

Розглянемо можливість визначення параметрів ФП згідно рівнянь параметрів ФП (2.2.8) - (2.2.11). Розглянемо наступні випадки:

1. Зміна параметрів S_{n2}, S_{n1}, S'_l та $\Delta y'$ відбувається в межах ± 1,0%.
2. Зміна параметрів S_{n2}, S_{n1}, S'_l та $\Delta y'$ відбувається в межах ± 10,0%.
3. Зміна параметра S_{n2} відбувається в межах ± 10,0%, параметрів S_{n1}, S'_l та $\Delta y'$ в межах ± 1,0%.
4. Зміна параметрів S_{n2} та S_{n1} відбувається в межах ± 10,0%, а параметрів S'_l та $\Delta y'$ в межах ± 1,0%.

5. Зміна параметрів S_{n2}, S_{n1} та S'_l відбувається в межах ± 10,0%, а параметра $\Delta y'$ в межах ± 1,0%.

6. Зміна параметрів S_{n1}, S'_l та $\Delta y'$ відбувається в межах ± 10,0%, а параметра S_{n2} в межах ± 1,0%.

У цьому випадку, як і при лінійної ФП, потрібно також проводити кілька циклів вимірювань, в кожному з яких необхідно провести розрахунки за рівняннями (2.2.8) - (2.2.11), усереднити отримані значення і за отриманими даними визначити величину відхилення кожного з параметрів.

При розрахунках також вважалося, що за час проведення тактів вимірювань значення параметрів ФП і їх відхилень залишалися незмінними. В результаті були отримані наступне значення:

1. При зміні параметрів S_{n2}, S_{n1}, S'_l та $\Delta y'$ в межах ± 1,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,0003 вимірюваних одиниць (або 0,01% від номінального значення);

– відхилення параметра S_{n1} склало 0,0001 вимірюваних одиниць (або 0,005% від номінального значення);

– відхилення параметра S'_l склало 0,00002 вимірюваних одиниць (або 0,001% від номінального значення);

– відхилення параметра $\Delta y'$ склало величини малого порядку.

2. При зміні параметрів S_{n2}, S_{n1}, S'_l та $\Delta y'$ в межах ± 10,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,003 (або 0,1% від номінального значення);

– відхилення параметра S_{n1} склало 0,001 (або 0,05% від номінального значення);

– відхилення параметра S'_l склало 0,0002 (або 0,01% від номінального значення);

– відхилення параметра $\Delta y'$ склало величину малого порядку.

3. При зміні параметра S_{n2} в межах ± 10,0%, параметрів S_{n1}, S'_l та $\Delta y'$ в межах ± 1,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,01 (або 0,33% від номінального значення);

– відхилення параметра S_{n1} склало 0,004 (або 0,2% від номінального значення);

– відхилення параметра S'_l склало 0,001 (або 0,07% від номінального значення);

– відхилення параметра $\Delta y'$ склало величину малого порядку.

4. При зміні параметрів S_{n2} та S_{n1} в межах ± 10,0%, параметрів S'_l та $\Delta y'$ в межах ± 1,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,01 (або 0,33% від номінального значення);

– відхилення параметра S_{n1} склало 0,004 (або 0,2% від номінального значення);

– відхилення параметра S'_l склало 0,02 (або 1,33% від номінального значення);

– відхилення параметра $\Delta y'$ склало 0,001 (або 1,0% від номінального значення).

5. При зміні параметрів S_{n2}, S_{n1} та S'_l в межах ± 10,0%, параметра $\Delta y'$ в межах ± 1,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,01 (або0,33% від номінального значення);

– відхилення параметра S_{n1} склало 0,004 (або 0,2% від номінального значення);

– відхилення параметра S'_l склало 0,001 (або 0,07% від номінального значення);

– відхилення параметра $\Delta y'$ склало 0,001 (або 1,0% від номінального значення).

6. При зміні параметрів S_{n1}, S'_l та $\Delta y'$ в межах ± 10,0%, параметра S_{n2} в межах ± 1,0%, отримують наступні відхилення параметрів:

– відхилення параметра S_{n2} склало 0,01 (або 0,33% від номінального значення);

– відхилення параметра S_{n1} склало 0,004 (або 0,2% від номінального значення);

– відхилення параметра S'_l склало 0,001 (або 0,07% від номінального значення);

– відхилення параметра $\Delta y'$ склало 0,005 (або 5,0% від номінального значення).

Таким чином, проведені розрахунки за рівняннями надлишкових вимірювань показали, що, якщо значення відхилень всіх параметрів ФП досить малі (випадок 1 і 2), то це свідчить про симетричному зміні ФП. Якщо ж значення відхилень параметрів збільшуються (більше 0,1%), то зміна ФП відбувається несиметрично і застосування прямих методів призведе до збільшення похибки вимірювання. Застосування МНВ, в цьому випадку, дозволить вчасно визначити такі відхилення параметрів без зниження точності отриманого результату.

Крім того, було встановлено, що найвагомішим параметром при поліноміальній ФП є параметр S_{n2}. Він найбільш чутливий до будь-яких змін параметрів ФП. Розрахунки показали, що якщо при заданих даних відхилення параметр S_{n2} складає більше 0,1%, то це свідчить про те, що відхилення інших параметрів ФП відбувається несиметрично і необхідно приймати рішення про заміну датчика.

2.3.2. МОДЕЛЮВАННЯ НЕЛІНІЙНОЇ ФУНКЦІЇ ПЕРЕТВОРЕННЯ ПРИ БАГАТОКРАТНИХ ВИМІРЮВАННЯХ

Оскільки МНВ зарекомендували себе як високоточні методі і такі, що не потребують додаткової лінеаризації нелінійної функції перетворення, то особливий інтерес представляє дослідження логарифмічної функції при багатократних вимірюваннях.

Дослідження проводилися з використанням кремнієвого фотодіода ФД307, який має логарифмічну функцію перетворення (ФП) та наступні параметри: темновий струм фотодіода Is=0,003 мкА, струмова (монохромова) чутливість $S_{I\lambda}$=0,27 А/Вт (при λ=0,55 мкм).

Як відомо, функція перетворення фотодіода, що працює в фотогальванічному режимі (режим з навантаженням), має вигляд:

$$U_R = \frac{kT}{q}\ln\left(\frac{S_I\Phi_x}{I_S}+1\right) - U_{RM}, \qquad (2.3.1)$$

де U_R – напруга на навантаженні;

k – стала Больцмана (k=1,38·10^{-23} Дж/К);

T–температура фотодіода (температура навколишнього середовища або блоку стабілізації температури);

q – заряд електрона (q=1,6·10^{-19} Кл);

S_I – струмова (монохроматична) чутливість фотодіода;

Φ_x – потік оптичного випромінювання, що падає на фотодіод;

Is – темновий струм фотодіода;

U_{RM} – падіння напруги на омічних елементах діода.

Після введення наступних змін: $kT/q=S'_{\text{Н}}$ (де $S'_{\text{Н}}$ – крутість перетворення), а також $\frac{S_I}{I_S}=\frac{1}{\Phi_T}$ (де $\Phi_{\text{т}}$ – темновий потік), вираз (2.3.1) прийме спрощений вид:

$$U'_R = S'_H \ln\left(\frac{\Phi_x}{\Phi_T}+1\right) - U'_{RM}. \qquad (2.3.2)$$

Оскільки ФП (2.3.2) має 4 змінні, то необхідно скласти систему з 4 рівнянь величин, кожна з яких описує 4 такту вимірювань. Для цього крім такту вимірювання контрольованої величини відбуваються такти вимірювання нормованих за значенням потоків випромінювання Φ_0 і $\Delta\Phi_0$. Отже, необхідно сформувати 4 такти вимірювання, що будуть описуватися 4 рівняннями величин.

Представимо математичну модель у вигляді наступної системи рівнянь:

$$\begin{cases} U'_{R1} = S'_H \ln\left(\Phi_0/\Phi_T+1\right) - U'_{RM}, \\ U'_{R2} = S'_H \ln\left((\Phi_0+\Delta\Phi_0)/\Phi_T+1\right) - U'_{RM}, \\ U'_{R3} = S'_H \ln\left((\Phi_x)/\Phi_T+1\right) - U'_{RM}, \\ U'_{R4} = S'_H \ln\left((\Phi_x+\Delta\Phi_0)/\Phi_T+1\right) - U'_{RM}, \end{cases} \qquad (2.3.3)$$

де U'_{Ri} – напруга в кажному *i*-му (i=(1÷4)) такті вимірювання.

Вирішуючи цю систему відносно Φ_x, отримуємо вираз виду:

$$\Phi_x = \frac{\Delta\Phi_0}{\left[\left(\Delta\Phi_0/((\Phi_0+\Phi_T)+1)\right)^{(U'_{R4}-U'_{R3})/(U'_{R2}-U'_{R1})}\right]-1} - \Phi_T. \tag{2.3.4}$$

Розглянемо детальніше багатократні вимірювання і їх підходи, математичні моделі і алгоритми яких були розглянуті в попередніх розділах. Так, при проведенні m циклів вимірювань, згідно з алгоритмом першого підходу рівняння надлишкових вимірювань (2.1.61) рівняння надлишкових вимірювань шуканого потоку оптичного випромінювання (2.3.4) набуде виду:

$$\Phi_x = \frac{\Delta\Phi_0}{\left[\left(\Delta\Phi_0/((\Phi_0+\Phi_T)+1)\right)\left(\frac{\sum_{i=1}^{m}U'_{R4i}}{m}-\frac{\sum_{i=1}^{m}U'_{R3i}}{m}\right)\Big/\left(\frac{\sum_{i=1}^{m}U'_{R2i}}{m}-\frac{\sum_{i=1}^{m}U'_{R1i}}{m}\right)\right]-1} - \Phi_T, \tag{2.3.5}$$

а при застосування другого підходу (згідно з алгоритмом першого підходу рівняння надлишкових вимірювань (2.1.63))

$$\Phi_x = \frac{\sum_{i=1}^{m}\Phi_{xi}}{m} = \frac{\sum_{i=1}^{m}\frac{\Delta\Phi_0}{\left[\left(\Delta\Phi_0/((\Phi_0+\Phi_T)+1)\right)^{(U'_{R4i}-U'_{R3i})/(U'_{R2i}-U'_{R1i})}\right]-1} - \Phi_T}{m}. \tag{2.3.6}$$

Проведемо комп'ютерне моделювання цих підходів і на підставі порівняльного аналізу визначимо більш точний підхід. При цьому значення нормованих фізичних величин в сумі повинні бути одного порядку з контрольованою. Тому вибираємо, Φ_0=0,8 мВт, $\Delta\Phi_0$=0,1 мВт.

Також було запропоновано такі значення похибки відтворення нормованих за значенням потоків випромінювання Φ_0 і $\Delta\Phi_0$, які

становитиме ±0,1 %. Така невисока похибка обумовлена тим, що нормовані за значенням величини Φ_0 і $\Delta\Phi_0$ формується за допомогою каліброваного джерела високої точності. Також поставимо межі змін для параметрів U'_{RM} і $S'_{\text{Н}}$ від ±1,0 % до ±10,0 %.

В якості програмного середовища, як засобу для математичного моделювання і аналізу поведінки сенсора при різних вхідних даних, було обрано Mathcad15.0.

В результаті комп'ютерного моделювання багатократних вимірювань при двох підходах були отримані наступні значення, які представлені в табл. 2.3.3. При цьому, слід було, що за час проведення одного циклу вимірів, значення U'_{RM} і $S'_{\text{Н}}$ параметрів ФП і їх відхилень залишалися незмінними.

Таблиця 2.3.3
Результати вимірювань при зміні параметрів U'_{RM} і S'_H

№	Зміна параметрів U'_{RM} і S'_H	U'_{R1}	U'_{R2}	U'_{R3}	U'_{R4}	Φ_x
1	U'_{RM}=(0,01+0,0001) В і $S'_H=S_H\cdot$(1+0,01)	0,2753	0,2783	0,2810	0,2834	$9.998654\cdot10^{-4}$
2	U'_{RM}=(0,01–0,0001) В і $S'_H=S_H\cdot$(1–0,01)	0,2695	0,2724	0,2750	0,2774	$1.000135\cdot10^{-3}$
3	U'_{RM}=(0,01+0,001) В і $S'_H=S_H\cdot$(1+0,1)	0,3016	0,3049	0,3078	0,3105	$9.998654\cdot10^{-4}$
4	U'_{RM}=(0,01–0,0001) В і $S'_H=S_H\cdot$(1–0,1)	0,2432	0,2458	0,2482	0,2504	$1.000135\cdot10^{-3}$
6	U'_{RM}=(0,01+0,001) В і $S'_H=S_H\cdot$(1+0,01)	0,2762	0,2792	0,2819	0,2843	$9.998654\cdot10^{-4}$
7	U'_{RM}=(0,01–0,001) В і $S'_H=S_H\cdot$(1+0,01)	0,2742	0,2772	0,2799	0,2823	$9.998654\cdot10^{-4}$
8	U'_{RM}=(0,01+0,001) В і $S'_H=S_H\cdot$(1–0,01)	0,2706	0,2735	0,2761	0,2785	$1.000135\cdot10^{-3}$
9	U'_{RM}=(0,01–0,001) В і $S'_H=S_H\cdot$(1–0,01)	0,2686	0,2715	0,2741	0,2765	$1.000135\cdot10^{-3}$
10	U'_{RM}=(0,01+0,0001) В і $S'_H=S_H\cdot$(1+0,1)	0,3007	0,3040	0,3069	0,3096	$9.998654\cdot10^{-4}$

11	U'_{RM}=(0,01–0,0001) В і S'_H=S_H·(1+0,1)	0,3005	0,3038	0,3067	0,3094	$9.998654 \cdot 10^{-4}$
12	U'_{RM}=(0,01+0,0001) В і S'_H=S_H·(1–0,1)	0,2443	0,2469	0,2493	0,2515	$1.000135 \cdot 10^{-3}$
13	U'_{RM}=(0,01–0,0001) В і S'_H=S_H·(1–0,1)	0,2441	0,2467	0,2491	0,2513	$1.000135 \cdot 10^{-3}$
Середні значення		0,2724	0,2754	0,2780	0,2804	$10.000002 \cdot 10^{-4}$

Було визначено, що відносна похибка вимірювання контрольованого потоку Φ_x в кожному циклі вимірювань становитиме 0,013 %, що підтверджує високу точність МНВ.

Далі, слідуючи запропонованим алгоритмом першого підходу, знаходять середні значення отриманих значень напруг для кожного з тактів вимірювань: $\bar{U}'_1 = 0,2724$ В, $\bar{U}'_2 = 0,2754$ В, $\bar{U}'_3 = 0,2780$ В, $\bar{U}'_4 = 0,2804$ В. Отримані значення підставлятися в рівняння (2.3.5), в результаті чого отримаємо, що значення контрольованого потоку випромінювання Φ_x=$9,999925 \cdot 10^{-4}$ Вт і відповідне значення відносної похибки δ буде складати $0,75 \cdot 10^{-3}$ %.

При аналізі впливу зміни кожного з параметрів U'_{RM} і S'_H на поточний значення напруг, слід зазначити, що більший вплив має зміна параметра S'_H. Так, при несиметричному зміні параметрів U'_{RM} і S'_H за час проведення циклу вимірювань, було виявлено, що чим більше відбувається відхилення параметра S'_H, тим більше відбувається відхилення значень напруг від його середнього значення.

При другому підході за кожен цикл вимірювань визначалося поточне значення потоку випромінювання Φ_x відповідно до рівняння надлишкових вимірювань (2.3.4). В результаті при усередненому значення контрольованого потоку випромінювання Φ_x=$1,0000002 \cdot 10^{-3}$ Вт, відповідно до рівняння (2.3.6), значення відносної похибки δ становитиме $0.02 \cdot 10^{-3}$ %.

При розгляді впливу зміни кожного з параметрів U'_{RM} і S'_H на поточний

значення напруг, слід зазначити, що другий підхід менш чутливий до їх змін, ніж перший.

Таким чином, аналіз двох підходів по обробці багаторазових вимірювань показав, що обидва підходи забезпечують високу точність вимірювань завдяки виключенню впливу параметрів нелінійної ФП на результат вимірювань. При порівнянні отримані результатів, більш точним виявився другий підхід, який показав на порядок меншу відносну похибку вимірювань. Крім того, при другому підході поточне значення потоку випромінювання Φ_x залежить лише від знака зміни параметра S'_H і не залежить від величини його зміни. Однак це може бути, як перевагою, так і недоліком даного підходу. Перевагою є те, що можна проводити меншу кількість циклів вимірювань. Недоліком є те, що цей підхід менш чутливий до розкиду параметрів ФП. Так, при збільшенні параметрів з 1 % до 10 %, поточне значення потоку випромінювання Φ_x не змінюється. Тому при застосуванні другого підходу необхідно частіше ніж при першому підході проводити метрологічний контроль параметрів ФП

Розглянемо вплив значення нормованої за значенням фізичної величини на результат вимірювань. Для цього досліджуємо випадок, коли різниця між значеннями контрольованої величини і нормованої величиною збільшитися. Зменшимо величину нормованого потоку Φ_0 з 0,8 мВт до 0,4 мВт, при цьому обрані раніше значення інших величин залишимо незмінними. Результати комп'ютерного моделювання представлені в табл. 2.3.4. При цьому, в табл. 2.3.4 наведені результати, отримані при максимальних відхиленнях параметрів U'_{RM} і S'_H.

Таблиця 2.3.4
Результати вимірювань при зміні параметрів U'_{RM} і S'_H

№	Зміна параметрів U'_{RM} і S'_H	U'_1	U'_2	U'_3	U'_4	Φ_x
1	U'_{RM}=(0,01+0,0001) В і	0,2576	0,2633	0,2810	0,2834	$9.997049 \cdot 10^{-4}$

	$S'_H=S_H\cdot(1+0{,}01)$					
2	U'_{RM}=(0,01–0,0001) В і $S'_H=S_H\cdot(1-0{,}01)$	0,2521	0,2577	0,2750	0,2774	$1.000295\cdot10^{-3}$
3	U'_{RM}=(0,01+0,001) В і $S'_H=S_H\cdot(1+0{,}1)$	0,2824	0,2886	0,3078	0,3105	$9.997049\cdot10^{-4}$
4	U'_{RM}=(0,01–0,001) В і $S'_H=S_H\cdot(1-0{,}1)$	0,2274	0,2325	0,2482	0,2504	$1.000295\cdot10^{-3}$
Середні значення		0,2605	0,2652	0,2780	0,2804	$1.0000001\cdot10^{-3}$

В результаті отримаємо, що при першому підході значення контрольованої величини потоку випромінювання становитиме Фх=$9.999836\cdot10^{-4}$ Вт і відповідно значення відносної похибки δ становитиме $1.64\cdot10^{-3}$ %. Отже, при першому підході збільшення різниці між значеннями контрольованої величиною і нормованої за значенням призводить до збільшення відносної похибки майже в 2 рази.

Комп'ютерне моделювання при другому підході показав, що значення контрольованої величини потоку випромінювання становитиме $\Phi_x=1.0000001\cdot10^{-3}$ Вт і значення відносної похибки δ становитиме $0.01\cdot10^{-3}$ %. Отже, при другому підході збільшення різниці між значеннями контрольованої величиною і нормованої за значенням не призводить до істотного збільшення відносної похибки.

Слід зазначити, що відносна похибка вимірювання контрольованого потоку Φ_x в кожному циклі вимірювань становитиме 0,03 %.

Подальше зменшення величини нормованого потоку Φ_0 до 0,2 мВт дало можливість отримати такі результати, які представлені в табл. 2.3.5.

Таблиця 2.3.5
Результати вимірювань при зміні параметрів U'_{RM} і S'_H

№	Зміна параметрів U'_{RM} і S'_H	U'_1	U'_2	U'_3	U'_4	Φ_x
1	U'_{RM}=(0,01+0,0001) В і $S'_H=S_H\cdot(1+0{,}01)$	0,2400	0,2503	0,2810	0,2834	$9.99413\cdot10^{-4}$

2	U'_{RM}=(0,01–0,0001) В і S'_H=S_H·(1–0,01)	0,2348	0,2449	0,2750	0,2774	$1.000569 \cdot 10^{-3}$
3	U'_{RM}=(0,01+0,001) В і S'_H=S_H·(1+0,1)	0,2631	0,2744	0,3078	0,3105	$9.99413 \cdot 10^{-4}$
4	U'_{RM}=(0,01–0,001) В і S'_H=S_H·(1–0,1)	0,2116	0,2209	0,2482	0,2504	$1.000569 \cdot 10^{-3}$
Середні значення		0,2605	0,2652	0,2780	0,2804	$1.0000003 \cdot 10^{-3}$

В результаті отримаємо, що при першому підході значення відносної похибки δ становитиме 0,003 %, а значення відносної похибки δ при другому підході становитиме $0,03 \cdot 10^{-3}$ %.

Отже, другий підхід менш чутливий до збільшення різниці між значеннями контрольованої величиною і нормованої за значенням і не призводить до істотного збільшення відносної похибки.

Слід зазначити, що відносна похибка вимірювання контрольованого потоку Φ_x в кожному циклі вимірювань становитиме 0,057 %. Таким чином, збільшення різниці між значеннями контрольованої величиною і нормованої за значенням призводить до підвищення похибки за цикл вимірювань.

В результаті комп’ютерного моделювання було встановлену високу точність запропонованих підходів при обробці результатів багатократних вимірювань. Це стає можливим завдяки представленим рівнянням надлишкових вимірювань, в яких завдяки операції відніманню і діленню вихідних напруг виключається вплив, відповідно, адитивної і мультиплікативної складової похибки вимірювання. Крім того, статистична обробка результатів багатократних вимірювань згідно запропонованих алгоритмів дозволяє зменшити і випадкову складову похибку.

При проведенні порівняльного аналізу результатів комп'ютерного моделювання багатократних вимірювань було встановлено, що більш точним виявився другий підхід. Так, обробка результатів проміжних

вимірювань відповідно до рівняння (2.3.6), показує на порядок меншу відносну похибку багаторазових вимірювань, ніж обробка отриманих результатів відповідно до рівняння (2.3.5).

Слід зазначити, що ще однією особливістю є те, що другий підхід менш чутливий до збільшення різниці між значеннями, контрольованою і нормованої величинами, і не призводить до істотного збільшення відносної похибки як при першому підході. Це свідчить про можливість вимірювання контрольованого потоку Φ_x великий величини. Крім того, при другому підході поточне значення потоку випромінювання Φ_x залежить лише від зміни знаку параметра S'_H ФП і не залежить від величини зміни, в результаті чого можна проводити меншу кількість циклів вимірювань. Однак недоліком є те, що другий підхід менш чутливий до розкиду параметрів ФП, тому в цьому випадку необхідно частіше, ніж при першому підході, проводити метрологічний контроль параметрів ФП.

Такі висновки можна вважати доцільними з практичної точки зору, тому, що дозволяють обґрунтувати застосування МНВ при багатократних вимірювань за двома підходами. З теоретичної точки зору вони дозволяють стверджувати, що МНВ при багатократних вимірюваннях за двома підходами дозволяють підвищити точність вимірювання при нелінійній ФП, що є одним з їхніх переваг. Однак неможливо не відмітити, що такі високі результати по точності отримуються у випадку, коли за час проведення кожного циклу вимірювань зміни параметрів залишаються сталими. Крім того, МНВ притаманна методична похибка вимірювання, що обумовлена похибкою відтворення нормованих за значенням потоків випромінювання. Неможливість зняти названі обмеження в рамках даного дослідження породжує потенційно цікавий напрям подальших досліджень.

2.3.3. ДОСЛІДЖЕННЯ ЛОГАРИФМІЧНОЇ ФУНКЦІЇ ПЕРЕТВОРЕННЯ ПО РОЗШИРЕННЮ ДІАПАЗОНУ ВИСОКОТОЧНИХ ВИМІРЮВАНЬ

Для дослідження впливу значень нормованих величин на похибку вимірювання, а також встановлення закономірності між вимірювальної величиною і нормованої, в якості датчика було обрано кремнієвий фотодіод ФД307 (Україна).

Робочий діапазон контрольованого потоку випромінювання було задано в межах $\Phi_x=(0{,}90\cdot10^{-5} \div 0{,}97\cdot10^{-3})$ Вт, а діапазон нормованого потоку Φ_0 – в межах $(0{,}80\cdot10^{-5} \div 0{,}96\cdot10^{-3})$ Вт. Для проведення комп'ютерного моделювання кожен з цих обох діапазонів був розділений на 14 кроків. Дослідження також проводилися і при трьох значеннях нормованого потоку $\Delta\Phi_0$, що становлять відповідно $0{,}001\cdot10^{-3}$ Вт, $0{,}01\cdot10^{-3}$ Вт та $0{,}1\cdot10^{-3}$ Вт. Значення похибки відтворення нормованих за значенням потоків випромінювання Φ_0 і $\Delta\Phi_0$ було обрано $\Delta_\phi=0{,}1\cdot10^{-6}$ Вт. Крім того, були задані межі змін для параметрів U'_{RM} і $S'_{\text{Н}}$, що лежать в рамках ±1,0 %.

В результаті проведення комп'ютерного моделювання в середовищі Mathcad15.0 були отримані наступні дані відносної похибки вимірювання при різних значеннях нормованого потоку $\Delta\Phi_0$, які представлено в табл. 2.3.6.

Таблиця 2.3.6
Відносні похибки вимірювань при різних значеннях нормованого потоку $\Delta\Phi_0$, (%)

$\Delta\Phi_0=0{,}001\cdot10^{-3}$ Вт														
потік	Φ_{01}	Φ_{02}	Φ_{03}	Φ_{04}	Φ_{05}	Φ_{06}	Φ_{07}	Φ_{08}	Φ_{09}	Φ_{010}	Φ_{011}	Φ_{012}	Φ_{013}	Φ_{014}
Φ_{x1}	0,34	0,43	0,46	0,48	0,49	0,50	0,50	0,51	0,51	0,51	0,51	0,52	0,52	0,52
Φ_{x2}	0,13	0,04	**0,00**	**0,00**	0,02	0,02	0,03	0,03	0,04	0,04	0,04	0,04	0,04	0,04
Φ_{x3}	0,15	0,07	0,04	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01	0,01	0,02
Φ_{x4}	0,16	0,08	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x5}	0,17	0,08	0,05	0,04	0,03	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x6}	0,17	0,09	0,06	0,04	0,03	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**

Φ_{x7}	0,17	0,09	0,06	0,04	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,18	0,09	0,06	0,04	0,03	0,03	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,18	0,09	0,06	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**
Φ_{x10}	0,18	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**
Φ_{x11}	0,18	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	**0,00**
Φ_{x12}	0,18	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**
Φ_{x13}	0,18	0,10	0,06	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**
Φ_{x14}	0,18	0,10	0,06	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01
$\Delta\Phi_0=0{,}01\cdot10^{-3}$ Вт														
Φ_{x1}	0,17	0,28	0,32	0,35	0,36	0,37	0,38	0,38	0,39	0,39	0,39	0,39	0,40	0,40
Φ_{x2}	0,13	0,05	0,01	**0,00**	0,01	0,02	0,03	0,03	0,03	0,04	0,04	0,04	0,04	0,04
Φ_{x3}	0,15	0,07	0,04	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01	0,02
Φ_{x4}	0,16	0,08	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x5}	0,16	0,08	0,05	0,04	0,03	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x6}	0,16	0,09	0,06	0,04	0,03	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x7}	0,17	0,09	0,06	0,04	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,17	0,09	0,06	0,04	0,03	0,03	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,17	0,09	0,06	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**
Φ_{x10}	0,17	0,09	0,06	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**
Φ_{x11}	0,17	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	**0,00**
Φ_{x12}	0,17	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**
Φ_{x13}	0,17	0,09	0,06	0,05	0,04	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**
Φ_{x14}	0,17	0,09	0,06	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01
$\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт														
Φ_{x1}	0,15	0,03	0,02	0,05	0,07	0,09	0,10	0,11	0,11	0,12	0,12	0,13	0,13	0,13
Φ_{x2}	0,12	0,06	0,04	0,02	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,02	0,02	0,02	0,02	0,02
Φ_{x3}	0,12	0,07	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x4}	0,12	0,07	0,05	0,03	0,02	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x5}	0,12	0,07	0,05	0,04	0,03	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x6}	0,12	0,07	0,05	0,04	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x7}	0,12	0,07	0,05	0,04	0,03	0,02	0,02	0,02	0,01	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**
Φ_{x10}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**	**0,00**
Φ_{x11}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	**0,00**
Φ_{x12}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	**0,00**
Φ_{x13}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	0,01
Φ_{x14}	0,12	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	0,01

Слід зауважити, що табличні значення відносних похибок вимірювання не мають нульового значення, а складають величини малого порядку (тисячні відсотка).

З наведених даних (табл. 2.3.6) видно, що більший вплив на похибку вимірювання справляє величина нормованого потоку Φ_0 ніж $\Delta\Phi_0$. Так зміна на 2 порядки значення нормованого потоку $\Delta\Phi_0$ суттєво не впливає на результат. Однак слід зауважити, що все ж найкращі результати по точності серед представлених 3х значень $\Delta\Phi_0$ були отримані при $\Delta\Phi_0=0{,}01\cdot10^{-3}$ Вт.

Як видно з даних, які були наведені в табл. 2.3.6, між контрольованим потоком Φ_x і нормованим Φ_0 спостерігається певна закономірність. Для її визначення було використано пакет аналізу даних Excel для нелінійної функції. Використовуючи опцію пошуку рішення, було знайдено оптимальне значення Φ_0, при якому виконується близькість до нуля значення похибки вимірювання. Таким чином було знайдено співвідношення між Φ_0 і Φ_x, що складає 3:1. Отже, збільшення в 3 рази значення нормованого потоку Φ_0 відносно контрольованого потоку Φ_x дозволить збільшити діапазон вимірювань, при якому похибка буде мати величину малого порядку (навіть за 1 цикл вимірювання).

Комп'ютерне моделювання проводилося при попередніх даних, але вже при збільшеному в 3 рази потоку Φ_0 відносно потоку Φ_x. Дослідження також проводилося і при різних значеннях нормованого потоку $\Delta\Phi_0$. В результаті були отримані наступні значення відносних похибок, які представлені в табл. 2.3.7.

Таблиця 2.3.7
Відносні похибки вимірювань при різних значеннях нормованого потоку $\Delta\Phi_0$, (%)

	$\Delta\Phi_0=0{,}001\cdot10^{-3}$ Вт													
потік	Φ_{01}	Φ_{02}	Φ_{03}	Φ_{04}	Φ_{05}	Φ_{06}	Φ_{07}	Φ_{08}	Φ_{09}	Φ_{010}	Φ_{011}	Φ_{012}	Φ_{013}	Φ_{014}
Φ_{x1}	0,47	0,50	0,51	0,52	0,52	0,52	0,52	0,53	0,53	0,53	0,53	0,53	0,53	0,53
Φ_{x2}	**0,00**	0,03	0,04	0,04	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05

Φ_{x3}	0,03	**0,00**	**0,00**	0,01	0,02	0,02	0,02	0,02	0,02	0,02	0,03	0,03	0,03	0,03
Φ_{x4}	0,04	0,01	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01	0,01	0,02	0,02	0,02	0,02
Φ_{x5}	0,05	0,02	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01
Φ_{x6}	0,05	0,02	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x7}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,05	0,03	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x10}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x11}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x12}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
$\Delta\Phi_0=0{,}01\cdot10^{-3}$ Вт														
Φ_{x1}	0,33	0,37	0,39	0,39	0,40	0,40	0,41	0,41	0,41	0,41	0,41	0,41	0,41	0,41
Φ_{x2}	**0,00**	0,02	0,03	0,04	0,04	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05
Φ_{x3}	0,03	**0,00**	**0,00**	0,01	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,03	0,03
Φ_{x4}	0,04	0,01	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,02
Φ_{x5}	0,05	0,02	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01
Φ_{x6}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x7}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,05	0,03	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x10}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x11}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x12}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,06	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
$\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт														
Φ_{x1}	0,03	0,09	0,11	0,13	0,14	0,14	0,15	0,15	0,15	0,15	0,15	0,16	0,16	0,16
Φ_{x2}	0,03	**0,00**	0,01	0,02	0,03	0,03	0,03	0,03	0,03	0,03	0,04	0,04	0,04	0,04
Φ_{x3}	0,04	0,01	**0,00**	**0,00**	**0,00**	0,01	0,01	0,02	0,02	0,02	0,02	0,02	0,03	0,03
Φ_{x4}	0,04	0,02	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	0,01	0,01	0,01	0,02	0,02
Φ_{x5}	0,04	0,02	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x6}	0,04	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x7}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x8}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x9}	0,05	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x10}	0,05	0,02	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x11}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**

Φ_{x12}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**

Слід зауважити, що табличні значення відносних похибок вимірювання не мають нульового значення, а складають величини малого порядку (тисячні відсотка).

Як видно з табличних даних (табл. 2.3.7), при співвідношенні значень між потоками Φ_0 і Φ_x, що складає 3:1, стає можливим збільшити діапазон вимірювань, при якому похибка має величину малого порядку. Також слід зауважити, що зміна на 2 порядки значення нормованого потоку $\Delta\Phi_0$ суттєво не впливає на результат вимірювання. Однак, було встановлено, що оптимальним є значення $\Delta\Phi_0=0{,}01\cdot10^{-3}$ Вт. В цьому випадку при значеннях потоку $\Phi_0=1{,}06\cdot10^{-3}$ Вт стає можливим вимірювати з відносною похибкою, що складає тисячні відсотка, контрольований потік в ширшому діапазоні $\Phi_x=(0{,}16\cdot10^{-3}\div0{,}97\cdot10^{-3})$ Вт.

З практичної точки зору представляє інтерес влив на результат вимірювання підвищення похибки відтворення нормованих за значенням потоків випромінювання Φ_0 і $\Delta\Phi_0$. Отже, було проведене комп'ютерне моделювання при збільшенні на один порядок похибки відтворення нормованого потоку з $\Delta_\phi=0{,}1\cdot10^{-6}$ Вт до $\Delta_\phi=0{,}1\cdot10^{-5}$ Вт, а також при умові співвідношенні значень між потоками Φ_0 і Φ_x, що складає 3:1. В результаті були отримані наступні значення відносних похибок, які представлені в табл. 2.3.8.

Таблиця 2.3.8
Відносні похибки вимірювань при різних значеннях нормованого потоку $\Delta\Phi_0$, (%)

$\Delta\Phi_0=0{,}001\cdot10^{-3}$ Вт														
потік	Φ_{01}	Φ_{02}	Φ_{03}	Φ_{04}	Φ_{05}	Φ_{06}	Φ_{07}	Φ_{08}	Φ_{09}	Φ_{010}	Φ_{011}	Φ_{012}	Φ_{013}	Φ_{014}
Φ_{x1}	4,59	4,90	5,02	5,08	5,12	5,15	5,16	5,18	5,19	5,20	5,20	5,21	5,22	5,22
Φ_{x2}	0,03	0,26	0,37	0,42	0,46	0,48	0,50	0,51	0,52	0,53	0,53	0,54	0,54	0,55

Φ_{x3}	0,31	0,02	0,09	0,14	0,17	0,20	0,21	0,23	0,24	0,24	0,25	0,26	0,26	0,27
Φ_{x4}	0,41	0,12	0,02	0,04	0,07	0,10	0,11	0,13	0,14	0,14	0,15	0,16	0,16	0,16
Φ_{x5}	0,46	0,17	0,07	0,01	0,02	0,04	0,06	0,07	0,08	0,09	0,10	0,10	0,11	0,11
Φ_{x6}	0,49	0,21	0,10	0,04	0,01	0,01	0,03	0,04	0,05	0,06	0,07	0,07	0,08	0,08
Φ_{x7}	0,51	0,23	0,12	0,07	0,03	**0,00**	**0,00**	0,02	0,03	0,04	0,04	0,05	0,05	0,06
Φ_{x8}	0,53	0,24	0,14	0,08	0,05	0,02	**0,00**	**0,00**	0,01	0,02	0,03	0,03	0,04	0,04
Φ_{x9}	0,54	0,25	0,15	0,09	0,06	0,04	0,02	**0,00**	**0,00**	0,01	0,02	0,02	0,03	0,03
Φ_{x10}	0,55	0,26	0,16	0,10	0,07	0,05	0,03	**0,02**	**0,00**	**0,00**	**0,00**	0,01	0,02	0,02
Φ_{x11}	0,55	0,27	0,16	0,11	0,08	0,05	0,04	0,02	0,01	**0,00**	**0,00**	**0,00**	0,01	0,01
Φ_{x12}	0,56	0,28	0,17	0,12	0,08	0,06	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,56	0,28	0,18	0,12	0,09	0,06	0,05	0,03	0,02	0,02	0,01	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,57	0,29	0,18	0,12	0,09	0,07	0,05	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**
$\Delta\Phi_0=0{,}01\cdot10^{-3}$ Вт														
Φ_{x1}	3,28	3,69	3,84	3,92	3,97	4,00	4,02	4,04	4,05	4,07	4,08	4,08	4,09	4,10
Φ_{x2}	0,07	0,23	0,34	0,39	0,43	0,45	0,47	0,48	0,49	0,50	0,51	0,51	0,52	0,52
Φ_{x3}	0,32	0,03	0,07	0,13	0,17	0,19	0,21	0,22	0,23	0,24	0,24	0,25	0,25	0,26
Φ_{x4}	0,41	0,13	0,02	0,03	0,07	0,09	0,11	0,12	0,13	0,14	0,15	0,15	0,16	0,16
Φ_{x5}	0,45	0,18	0,07	0,02	0,02	0,04	0,06	0,07	0,08	0,09	0,10	0,10	0,11	0,11
Φ_{x6}	0,48	0,21	0,10	0,05	0,01	0,01	0,03	0,04	0,05	0,06	0,06	0,07	0,07	0,08
Φ_{x7}	0,50	0,23	0,12	0,07	0,03	0,01	**0,00**	0,02	0,03	0,04	0,04	0,05	0,05	0,06
Φ_{x8}	0,52	0,24	0,14	0,08	0,05	0,03	**0,00**	**0,00**	0,01	0,02	0,03	0,03	0,04	0,04
Φ_{x9}	0,53	0,25	0,15	0,09	0,06	0,04	0,02	**0,00**	**0,00**	0,01	0,02	0,02	0,03	0,03
Φ_{x10}	0,54	0,26	0,16	0,10	0,07	0,05	0,03	0,02	**0,00**	**0,00**	**0,00**	0,01	0,02	0,02
Φ_{x11}	0,54	0,27	0,16	0,11	0,08	0,05	0,04	0,02	0,01	**0,00**	**0,00**	**0,00**	0,01	0,01
Φ_{x12}	0,55	0,27	0,17	0,12	0,08	0,06	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,55	0,28	0,17	0,12	0,09	0,06	0,05	0,03	0,02	0,02	0,01	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,56	0,28	0,18	0,12	0,09	0,07	0,05	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**
$\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт														
Φ_{x1}	0,34	0,90	1,14	1,27	1,35	1,41	1,45	1,48	1,51	1,53	1,55	1,55	1,58	1,58
Φ_{x2}	0,29	0,01	0,14	0,21	0,25	0,28	0,30	0,32	0,33	0,34	0,35	0,36	0,36	0,37
Φ_{x3}	0,37	0,11	**0,00**	0,06	0,10	0,12	0,14	0,16	0,17	0,18	0,19	0,19	0,20	0,21
Φ_{x4}	0,41	0,16	0,06	**0,00**	0,03	0,06	0,07	0,09	0,10	0,11	0,11	0,12	0,13	0,14
Φ_{x5}	0,43	0,19	0,10	0,04	**0,00**	0,02	0,04	0,05	0,06	0,07	0,07	0,08	0,09	0,10
Φ_{x6}	0,45	0,21	0,12	0,06	0,03	**0,00**	0,01	0,02	0,03	0,04	0,05	0,05	0,06	0,07
Φ_{x7}	0,45	0,23	0,13	0,08	0,05	0,02	**0,00**	**0,00**	0,02	0,03	0,03	0,04	0,05	0,06
Φ_{x8}	0,46	0,24	0,14	0,09	0,06	0,03	0,02	**0,00**	**0,00**	0,01	0,02	0,03	0,05	0,08
Φ_{x9}	0,47	0,24	0,15	0,01	0,07	0,04	0,03	0,01	**0,00**	**0,00**	**0,00**	0,01	0,02	0,02
Φ_{x10}	0,47	0,25	0,16	0,11	0,07	0,05	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	0,01	0,02
Φ_{x11}	0,47	0,25	0,16	0,11	0,08	0,06	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**	**0,00**

Φ_{x12}	0,48	0,26	0,17	0,12	0,08	0,06	0,05	0,03	0,02	0,02	**0,00**	**0,00**	**0,00**	**0,00**
Φ_{x13}	0,48	0,26	0,17	0,12	0,09	0,07	0,05	0,04	0,03	0,02	0,01	**0,00**	**0,00**	**0,00**
Φ_{x14}	0,48	0,26	0,17	0,12	0,09	0,07	0,05	0,04	0,03	0,02	0,02	**0,00**	**0,00**	**0,00**

Слід зауважити, що табличні значення відносних похибок вимірювання не мають нульового значення, а складають величини малого порядку (тисячні відсотка).

Як видно з табличних даних (табл. 2.3.8), збільшення похибки відтворення нормованих за значенням потоків випромінення призводить до звуження діапазону, в якому забезпечується відносна похибка вимірювання малого порядку. В цьому випадку також, як і в попередніх випадках, оптимальним є значення нормованого потоку$\Delta\Phi_0$, що складає $0{,}1\cdot10^{-3}$ Вт.

Було розглянуто випадок, коли не було проведено збільшення в 3 рази нормованого потоку Φ_0, але похибка відтворення нормованих потоків була збільшена на 1 порядок. Оскільки значення нормованого потоку $\Delta\Phi_0$ чинить незначний вплив на результат, то було обрано значення потоку $\Delta\Phi_0$, що складає 0,01 мВт. Результати такого моделювання представлено в табл. 2.3.9.

Таблиця 2.3.9
Відносні похибки вимірювань при $\Delta\Phi_0$=0,01 мВт, (%)

потік	Φ_{01}	Φ_{02}	Φ_{03}	Φ_{04}	Φ_{05}	Φ_{06}	Φ_{07}	Φ_{08}	Φ_{09}	Φ_{010}	Φ_{011}	Φ_{012}	Φ_{013}	Φ_{014}
Φ_{x1}	1,67	2,77	3,20	3,43	3,57	3,67	3,74	3,79	3,84	3,87	3,90	3,92	3,94	3,96
Φ_{x2}	1,25	0,45	0,13	0,03	0,14	0,21	0,26	0,30	0,33	0,35	0,37	0,39	0,40	0,42
Φ_{x3}	1,46	0,69	0,38	0,22	0,12	0,05	**0,00**	0,04	0,07	0,09	0,11	0,13	0,14	0,15
Φ_{x4}	1,54	0,77	0,47	0,31	0,21	0,15	0,10	0,06	0,03	**0,00**	0,01	0,03	0,04	0,06
Φ_{x5}	1,58	0,82	0,52	0,36	0,26	0,19	0,15	0,11	0,08	0,06	0,04	0,02	**0,00**	**0,00**
Φ_{x6}	1,61	0,85	0,55	0,39	0,29	0,23	0,18	0,14	0,11	0,09	0,07	0,05	0,04	0,03
Φ_{x7}	1,63	0,87	0,57	0,41	0,31	0,25	0,20	0,16	0,13	0,11	0,09	0,07	0,06	0,05
Φ_{x8}	1,64	0,88	0,58	0,43	0,33	0,26	0,21	0,17	0,15	0,12	0,10	0,09	0,07	0,06
Φ_{x9}	1,65	0,89	0,59	0,44	0,34	0,27	0,22	0,19	0,16	0,13	0,11	0,10	0,08	0,07
Φ_{x10}	1,66	0,90	0,60	0,45	0,35	0,28	0,23	0,19	0,17	0,14	0,12	0,11	0,09	0,08
Φ_{x11}	1,66	0,91	0,61	0,45	0,35	0,29	0,24	0,20	0,17	0,15	0,13	0,11	0,10	0,09

Φ_{x12}	1,67	0,91	0,62	0,46	0,36	0,29	0,24	0,21	0,18	0,16	0,14	0,12	0,11	0,09
Φ_{x13}	1,67	0,92	0,62	0,46	0,36	0,30	0,25	0,21	0,18	0,16	0,14	0,13	0,11	0,10
Φ_{x14}	1,67	0,92	0,62	0,47	0,37	0,30	0,25	0,22	0,19	0,16	0,15	0,13	0,12	0,10

Слід зауважити, що табличні значення відносних похибок вимірювання не мають нульового значення, а складають величини малого порядку (тисячні відсотка).

Таким чином, без збільшення нормованого потоку Φ_0 втричі, але при збільшенні похибки відтворення нормованих потоків на 1 порядок, призводить лише до поодиноких випадків зменшення відносної похибки до величин малого порядку (табл.2.3.9). Причому, це виявляється, коли між Φ_0 і Φ_x є співвідношення, як 3 до 1. Однак, в такому випадку необхідне знання поточного (прогнозованого) значення контрольованого потоку, що є не завжди відомим.

Подальше збільшення похибки відтворення нормованих потоків на 2 порядки з $\Delta_\phi=0{,}1\cdot10^{-6}$ Вт до $\Delta_\phi=0{,}1\cdot10^{-4}$ Вт призводить до значного погіршення результатів, а особливо при малих значеннях контрольованої величини. Крім цього, діапазон стає ще вужчим і переміщується в область високих значень контрольованої і нормованої величин. Однак найкращим в цьому випадку є значення нормованого потоку $\Delta\Phi_0$, що складає $0{,}01\cdot10^{-3}$ Вт, що забезпечує відносну похибку вимірювань малого порядку при співвідношенні значень $\Phi_0=(2{,}08\cdot10^{-3}\div2{,}89\cdot10^{-3})$ Вт та відповідних значень $\Phi_x=(0{,}67\cdot10^{-3}\div0{,}97\cdot10^{-3})$ Вт.

При дослідженні було розглянуто випадок при одночасному збільшенні на 1 порядок похибки відтворення нормованих потоків і при збільшенні похибки відтворення темнового потоку $\Phi_т$ (в межах ±10 %). Комп'ютерне моделювання показало незалежність отриманого результату від впливу похибки відтворення темнового потоку, тобто результаті залишились такими, як в табл.2.3.8.

При дослідженні впливу нормованих величин на результат

вимірювання, як витікає з отриманих результатів (табл. 2.3.6) було виявлено, що більший вплив справляє величина нормованого потоку Φ_0 ніж $\Delta\Phi_0$. Це обумовлено рівнянням величин (2.3.4), де величина нормованого потоку Φ_0 стоїть у певному співвідношенні до Φ_x. Так, зміна величини нормованого потоку Φ_0 призводить до суттєвіших змін значень відносних похибки. Натомість зміна на 2 порядки значення нормованого потоку $\Delta\Phi_0$ суттєво не впливає на результат. Вочевидь такий механізм впливу величини нормованого потоку Φ_0 є тим фактором регулювання процесу, завдяки якому можна впливати на відносну похибку вимірювання. Це не розходиться з практичними даними, що наведені в роботах [9, 17, 19], автори яких, теж пов’язують вплив величин певних параметрів каналу (величина темнового струму, введення поправок) на точність вимірювання.

Для доведення ствердження про критерії впливу величини нормованого потоку Φ_0 на діапазон високоточних вимірювань необхідно уважно його дослідити. При зіставленні отриманих значень відносних похибок була виявлена певна закономірність між контрольованою Φ_x і нормованою величиною Φ_0, при якій спостерігалося близькість до нуля даної похибки. Так, при значеннях потоку $\Phi_0=0{,}56\cdot10^{-3}$ Вт і $\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт стає можливим вимірювати з відносною похибкою, що складає тисячні відсотка, контрольований потік в ширшому діапазоні $\Phi_x=(0{,}08\cdot10^{-3}\div0{,}23\cdot10^{-3})$ Вт. При збільшеному значенні нормованого потоку Φ_0 спостерігалось розширення діапазону, в якому відносна похибка складає тисячні відсотка. Тож для виявлення закономірності між потоками Φ_0 і Φ_x, при якій спостерігається збільшення діапазону вимірювань контрольованої величини з високою точністю, був проведений комп’ютерний аналіз. В результаті такого аналізу з використанням опції пошуку рішення було встановлено, що співвідношення між нормованим потоком Φ_0 і контрольованою величиною Φ_x повинно становити як 3 до 1. Таким чином,

дотримання співвідношення між нормованою Φ_0 і контрольованою величиною Φ_x дає змогу отримати результат високої точності в широкому діапазоні значень. Про це свідчать результати досліджень, що наведені в табл. 2.3.7. Наприклад, при значеннях потоку $\Phi_0=1{,}06\cdot10^{-3}$ Вт і $\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт стає можливим вимірювати з відносною похибкою, що складає тисячні відсотка, контрольований потік в ширшому діапазоні $\Phi_x=(0{,}16\cdot10^{-3}\div0{,}97\cdot10^{-3})$ Вт. Тобто, при дотриманні співвідношення між нормованим потоком Φ_0 і контрольованою величиною Φ_x, як 3 до 1, дає змогу більш ніж в 5 разів розширити діапазон вимірювання, в якому відносна похибка має величину малого порядку. Це означає, що врахування даного співвідношення між нормованим і контрольованим потоками відкриває можливість для ефективного регулювання діапазоном високоточного вимірювання. Це також не розходиться з висновками робіт [17, 19], автори яких теж пов'язують підвищення точності вимірювання з дотриманням встановлених поправок до параметрів.

Але, в той самий час, на похибку вимірювання чинить вплив і методична похибка. В цьому сенсі особливий інтерес має вплив запропонованого співвідношення між нормованим і контрольованим потоками на методичну похибку. Це обумовлено тим, що похибка відтворення нормованих потоків є складовою методичної похибки запропонованого методу надлишкових вимірювань. Дослідження проводилися при збільшенні даної похибки на 1 і на 2 порядки. Було встановлено, що збільшення похибки на 1 порядок при збільшенні нормованого потоку Φ_0 втричі відносно Φ_x, призводить до звуження діапазону вимірювань, в якому значення відносної похибки прямую до нуля (табл. 2.3.8). Крім того, при початкових значеннях контрольованої величини відносна похибка приймає неприйнятні значення. Але, в той самий час, якщо ж не збільшувати нормований потік Φ_0 втричі відносно Φ_x, то це призведе лише до поодиноких випадків зменшення відносної

похибки до величин малого порядку (табл. 2.3.9). Слід зазначити, що в цьому випадку буде спостерігатися відношення між Φ_0 і Φ_x, як 3 до 1. Однак це накладає вимоги до знання поточного значення контрольованого потоку, що є не завжди відомим. Таким чином, отримані результати свідчать, що збільшення нормованого потоку Φ_0 втричі відносно Φ_x є доцільним. Подальше збільшення похибки відтворення вже на 2 порядки стає прийнятним лише на дуже вузькому діапазоні малих значень відносної похибки і при великих значеннях контрольованої величини. Це обґрунтовується тим, що похибка відтворення нормованого потоку, в цьому випадку, стає спів розмірною за значенням з контрольованим потоком Φ_x (особливо при малих значеннях потоку Φ_x). Таким чином, отримані результати, на відміну від результатів в [20], свідчать про вплив методичної похибки (похибки відтворення надлишкових потоків) не лише на точність вимірювання, а і на вимірювальний діапазон.

Слід зазначити, що на точність вимірювання чинить вплив і величина темнового струму, як це показано і в роботі [9]. Тож, науковий інтерес має дослідження впливу похибки відтворення темнового потоку. Комп'ютерне моделювання проводилося при похибці відтворення темнового потоку $\Phi_т$ в межах ±10 %. Дослідження показало, що, при знайденому співвідношенні між потоками, а також при збільшенні похибки відтворення нормованих потоків на 1 порядок, похибка відтворення темнового потоку не чинить вплив на результату вимірювання. Це, вочевидь, пояснюється малою величиною самого значення темнового потоку.

Таким чином, при умові виконання співвідношення нормованого потоку Φ_0 до контрольованого Φ_x, як 3 до 1, стає можливим збільшити діапазон вимірювань, при якому відносна похибка складає тисячні відсотка.

Такі висновки можна вважати доцільними з практичної точки зору, оскільки дозволяють обґрунтувати дотримання співвідношення між

нормованою Φ_0 і контрольованою величиною Φ_x, як 3 до1, при якому стає можливим проведення технологічних процесів з високою точністю. При цьому, стає можливим значно розширити діапазон вимірювань, при якому відносна похибка має величину малого порядку (тисячні відсотка), тобто дозволяє проводити весь технологічний процес від його початку до кінця з високою точністю. Але, таке збільшення нормованого потоку Φ_0 накладає вимоги до високої потужності каліброваного джерела випромінювання. Однак, у випадках, коли відоме поточне (або прогнозоване) значення контрольованого потоку, то збільшувати в 3 рази нормований потік Φ_0 не потрібно. В цьому випадку рекомендовано лише дотримуватися співвідношення між Φ_0 і Φ_x, як 3 до 1, навіть при збільшеній на 1 порядок похибці відтворення нормованих потоків. Це дає можливість знизити вимоги до потужності та точності каліброваного джерела випромінювання.

З теоретичної точки зору тримані результати дозволяють стверджувати, що при застосуванні МНВ ваговим параметром між нормованими потоками $\Delta\Phi_0$ і Φ_0 слід вважати потік Φ_0. Так співвідношення між нормованою Φ_0 і контрольованою величиною Φ_x, як 3 до 1, дозволяє підвищити точність вимірювання, а також розширити діапазон вимірювань, при якому відносна похибка складає тисячні відсотка. Це є однією з переваг застосування знайденого співвідношення між потоками. Однак неможливо не відмітити, що такі високі результати по точності отримуються у випадку, коли за час проведення тактів вимірювань зміни параметрів залишаються сталими. Крім того, МНВ, як і будь-яким іншим методам, притаманна методична похибка, що обумовлена похибкою відтворення нормованих потоків. Так, збільшення похибки відтворення нормованих потоків призводить до звуження діапазоні, при якому відносна похибка має величину малого порядку. Однак, слід зазначити, що збільшення похибки відтворення темнового потоку, в цьому випадку, не чинить вплив на результат вимірювання. Тож

прийнятним залишається випадок, коли похибки відтворення нормованих потоків залишається якомога меншою. Неможливість зняти названі обмеження в рамках даного дослідження породжує потенційно цікавий напрям подальших досліджень.

2.3.4. ПОРІВНЯЛЬНИЙ АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

В результаті проведених досліджень було доведено перспективність методів надлишкових вимірювань в сфері підвищення точності вимірювань при нелінійній і нестабільної функції перетворення датчика з можливістю визначення значень параметрів самої функції.

Були наведені компоненти алгоритму методу надлишкових вимірювань, на підставі якого були представлені математичні моделі методу надлишкових вимірювань при лінійній і нелінійних (квадратичній, кубічній, поліноміальній і логарифмічній) функціях перетворення датчика. Показано шляхи формування надмірності за рахунок додаткового введення декількох нормованої величини за допомогою каліброваного джерела. Це дало можливість сформувати систему рівнянь, рішення якої дозволило вивести рівняння надлишкових вимірювань як шуканої фізичної величини, так і параметрів функції перетворення. Показано, що підвищення точності досягається за рахунок обробки проміжних результатів за рівнянням надлишкових вимірювань, в якому виключається вплив абсолютних значень параметрів функції перетворення датчика і їх відхилень від номінальних значень (за умови, що ці параметри залишаються постійними за час проведення тактів вимірювання). Показана можливість визначення значень параметрів функції перетворення по відповідним рівнянням. Завдяки такій самоперевірки параметрів вдається скоротити матеріальні і тимчасові витрати на повірку датчиків і вчасно проводити його заміну.

Також було показано, що, на відміну від прямих методів, застосування методів надлишкових вимірювань при нелінійній функції

перетворення не вимагає безпосередньої її лінеаризації або роботою на лінійній ділянці. Це досягається завдяки обробці результатів проміжних вимірювань за рівнянням надлишкових вимірювань, в якому на виході отримують результат такої як на вході.

В результаті порівняльного аналізу похибок прямого методу і методу надлишкових вимірювань встановлено, що представлений методи забезпечують більш високу точність вимірювання.

Також встановлено особливості застосування двох підходів при обробці результатів багатократних вимірювань. Так, при першому підході відбувається усереднення поточних значень напруг U'_{Ri} за все цикли вимірювань з подальшою їх підстановкою в рівняння надлишкових вимірювань (2.3.4). При другому підході за рівнянням надлишкових вимірювань (2.3.4) визначається поточні значення контрольованої величини Φ_{xi}, з подальшим його усередненням. Ці особливості в обробці даних проявляться в отримані результату високої точності за рахунок зменшення випадкової складової похибки і виключення впливу систематичної складової похибки, викликаної нестабільністю параметрів ФП (під впливом зовнішніх дестабілізуючих факторів). Проведено порівняльний аналіз отриманих результатів за двома підходами. Було встановлено, що перевагу слід віддати другому підходу, при якому:

а) відносна похибка має досить малу величину;

б) збільшення різниці між значеннями контрольованої величиною і нормованої за значенням майже не впливає на результат вимірювань, що дає можливість вимірювання контрольованого потоку Φ_x великий величини без накладення вимог до високих потужностей джерела каліброваного випромінювання;

в) на його результат майже не впливає величина відхилень параметра S'_H від його номінального значення, проте це вимагає більш частого метрологічного контролю.

Крім того, були досліджені математичні моделі при різних значеннях нормованих потоків випромінювання при змінах параметрів функції перетворення в межах ±1,0 %. В результаті порівняльного аналізу було встановлено, що найбільших плив серед нормованих потоків було виявлено потік Φ_0. Це проявляється у тому, шо більший вплив серед нормованих потоків на похибку вимірювання справляє величина потоку Φ_0. Завдяки цьому можна стверджувати, що керуючи величиною нормованого потоку Φ_0 можна розширювати діапазон вимірювань, при якому відносна похибка величину малого порядку (тисячні відсотка).

Знайдене таке співвідношення між значеннями контрольованого потоку та найбільш впливовим з нормованих, яке забезпечує в широкому діапазоні вимірювань найменшу похибку. Завдяки отриманому співвідношення значень між потоками Φ_0 і Φ_x, що складає 3:1, стає можливим збільшити діапазон вимірювань, при якому відносна похибка має величину малого порядку. Наприклад, при значеннях потоку $\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вті $\Phi_0=1{,}06\cdot10^{-3}$ Вт (що відповідає 5 кроку) стає можливим вимірювати з відносною похибкою, що складає тисячні відсотка, контрольований потік в ширшому діапазоні $\Phi_x=(0{,}16\cdot10^{-3}\div0{,}97\cdot10^{-3})$ Вт. У порівнянні з випадком, де не використовується це відношення і при $\Delta\Phi_0=0{,}1\cdot10^{-3}$ Вт та $\Phi_0=0{,}35\cdot10^{-3}$ Вт (що відповідає 5 кроку), значення становитиме лише $\Phi_x=0{,}08\cdot10^{-3}$ Вт. Це свідчить про можливість підвищення точності в широкому діапазоні шляхом застосування заданого співвідношення між потоками Φ_0 і Φ_x. Таким чином, була отримана точність, яку можна порівнянна з точністю зразкових мір.

Встановлено, що збільшення похибки відтворення нормованих за значенням потоків випромінення на 1 порядок призводить до звуження діапазону, в якому забезпечується відносна похибка вимірювання малого порядок. Особливо це звуження проявляється у випадку, коли не

застосовувати співвідношення між потоками $Ф_0$ і $Ф_x$, як 3 до 1 (в цьому випадку висуваються вимоги до знання поточного значення контрольованого потоку). Таким чином, для зменшення впливу похибки відтворення нормованих потоків на 1 порядок, виконання співвідношення між потоками $Ф_0$ і $Ф_x$, як 3 до1, є бажаним. Результати дослідження показали, що вимоги до дотримання похибки відтворення нормованих потоків якомога меншою все ще залишаються високими. Проведено комп'ютерне моделювання впливу похибки відтворення темнового потоку. Було встановлено, що похибка відтворення темнового потоку не чинить вплив на результат вимірювання. Таким чином, є підстави стверджувати про можливість розширення діапазону вимірювання, при якому значення відносної похибки складає тисячні відсотка, навіть за 1 цикл вимірювання.

Висновки по розділу 2

У представленому матеріалі представлені шляхи по розв'язанню задачі підвищення точності вимірювання за рахунок застосуванню методів надлишкових вимірювань.

У результаті проведених досліджень отримані такі наукові та практичні результати:

1. Проаналізовані існуючі підходи щодо підвищення точності вимірювання. Встановлено, що найбільш перспективними є методи надлишкових вимірювань, які працюють як при лінійній, так і при нелінійній функціях перетворення, а також забезпечують автоматичне виключення систематичної складової похибки вимірювання, що обумовлена зміною параметрів функції перетворення вимірювального каналу.

2. Представлені математичні моделі надлишкових вимірювань, за допомогою якої можна отримати рівняння надлишкових вимірювань як шуканої величини, так і параметрів функції перетворення датчика.

Наведені математичні моделі методів надлишкових вимірювань описують стан вимірювального каналу у дискретні моменти часу. Шляхом їх розв’язання отримані рівняння надлишкових вимірювань при різних видах функції перетворення сенсору (при лінійній, при квадратичній, кубічній, логарифмічній функціях перетворення, а також при функції перетворення, що описується поліномом 3-го ступеня). Запропоновані методи надлишкових вимірювань дозволяють визначити як значення потоку випромінення від об’єкта дослідження, так і значення параметрів функції перетворення вимірювального каналу. Це дає можливість додатково спрогнозувати метрологічну надійність вимірювального каналу.

3. На основі запропонованих математичних моделей представлені блок-схеми алгоритмів роботи надлишкових вимірювань при різних видах функції перетворення, а також при наявності випадкової складової похибки. Запропоновані алгоритми є підставою для розробки відповідних технічних рішень, що реалізують метод надлишкових вимірювань. В таких технічних рішеннях використовують сенсор з лінійною ФП, а також мікроковертор. Завдяки введенню мікроконвертора з розрахунковим ядром для обробки даних, стало можливим виконання великого об’єму розрахунків.

Реалізація алгоритмів забезпечує функціонування технічних рішень як єдиного програмно-технічного засобу вимірювання, а також обробку результатів поточних вимірювань по заданому рівнянню числових значень. Завдяки впровадження запропонованого алгоритму, реалізація технічних рішень при лінійній і нелінійних функціях перетворення забезпечує автоматичне виключення адитивної і мультиплікативної складових систематичної похибки результату визначення, що обумовлена зміною параметрів функції перетворення під дією дестабілізуючих факторів. Це забезпечується за рахунок реалізації методу надлишкових вимірювань, який забезпечує виключення впливу на результат вимірювання абсолютних значень параметрів лінійної функції перетворення вимірювального каналу

($S'_{\text{л}}$ і $\Delta U'_{\text{зм}}$) та їх відхилень від номінальних значень.

4. Досліджено та встановлені особливості застосування запропонованих підходів обробки результатів вимірювань в тому числі і багатократних. Проведеними дослідженнями впливу нормованих за значеннями потоків випромінювання на результат вимірювання було виявлено найбільш впливовий. Доведено, що більший вплив на відносну похибку вимірювання справляє величина нормованого потоку Φ_0 ніж $\Delta\Phi_0$. Завдяки цьому стає можливим дослідити співвідношення між контрольованим Φ_x та нормованим потоком Φ_0. Експериментальними дослідженнями підтверджено, що за рахунок збільшення в тричі нормованого за значенням потоку Φ_0 відносно контрольованого потоку Φ_x, стає можливим підвищити точність вимірювання в широкому діапазоні. Зокрема встановлено, що при значенні потоку $\Phi_0=1{,}06\cdot10^{-3}$ Вт стає можливим вимірювання з відносною похибкою, що складає тисячні відсотка, контрольований потік в ширшому діапазоні $\Phi_x=(0{,}16\cdot10^{-3}\div0{,}97\cdot10^{-3})$ Вт. Показано вплив похибки відтворення на результат вимірювання при умові збільшення в тричі нормованого за значенням потоку Φ_0 відносно контрольованого потоку Φ_x. Встановлено, що збільшення похибки відтворення нормованих потоків випромінення на 1 порядок призводить до звуження діапазону, в якому значення відносної похибки прямує до нуля. Показано, що при відсутності збільшення в тричі нормованого потоку Φ_0, збільшення похибки відтворення нормованих потоків на 1 порядок призводить до поодиноких випадків зменшення відносної похибки до величин малого порядку. Останнє, до речі, стосується випадків, коли виконується умова співвідношення між нормованим потокомФ$_0$ і контрольованим Φ_x, як 3 до 1. Показано, що похибка відтворення темнового потоку не чинить вплив на результат вимірювання.

Проведеними дослідженнями можливостей методів надлишкових

вимірювань встановлено високу ефективність представлених методів щодо підвищення точності багатократних вимірювань. Доведено, що рівняння надлишкових вимірювань забезпечує незалежність результату вимірювань від параметрів функції перетворення і їх відхилень від номінальних значень. Експериментальними дослідженнями підтверджено, що точність вимірювання підвищується за рахунок обробки результатів проміжних вимірювань за рівняннями надлишкових вимірювань по двом підходам. Це призводить до підвищення точності багатократних вимірювань. Зокрема встановлено, що при обробці результатів за першим підходом значення відносної похибки буде складати $0{,}75 \cdot 10^{-3}$ %, а за другим – $0{,}02 \cdot 10^{-3}$ %. Це дозволяє стверджувати, що, підвищення точності відбувається завдяки сумарному ефекту, що реалізується в запропонованих підходах: забезпечується виключення систематичної складової похибки, обумовленої зміною параметрів функції перетворення, а також зменшення випадкової складової похибки. Останнє, зокрема, стосується алгоритмів обробки багатократних вимірювань по двом підходам. Крім того, було встановлено, що другий підхід менш чутливий до збільшення різниці між значеннями контрольованої величиною і нормованої за значенням. Це дозволяє стверджувати про можливість вимірювання контрольованого параметру (Φ_x) великого значення без накладення вимог до високих потужностей джерела каліброваного випромінювання.

5. Зроблено порівняльний аналіз отриманих результатів та показати переваги методів надлишкових вимірювань в сфері підвищення точності вимірювань при нестабільних параметрах функції перетворення сенсора. Запропоновані методи забезпечують лінійну залежність результатів вимірювання від потоку випромінювання. На відміну від прямих методів, результат вимірювання є приведеним до входу вимірювального каналу. Безпосередньо прямі методи не забезпечують вирішення завдань

підвищення точності вимірювання при використанні сенсорів з нестабільною та нелінійною функцією перетворення.

У результаті аналізу похибок існуючих та методів надлишкових вимірювань встановлено, що методи надлишкових вимірювань забезпечують високу точність вимірювання завдяки обробці результатів проміжних вимірювань згідно з рівнянням надлишкових вимірювань. Останнє забезпечує виключення впливу на результат вимірювання абсолютних значень параметрів функції перетворення та відхилення їх значень від номінальних. За рахунок виконання операції віднімання виключається адитивна складова систематичної похибки вимірювань, а за рахунок виконання операції ділення – мультиплікативна складова похибки, які обумовлені зміною параметрів функції перетворення сенсору. Проаналізовані похибки методів надлишкових вимірювань, доведена висока ефективність автоматичного виключення систематичної складової похибки, що обумовлена зміною параметрів функції перетворення. Встановлено, що при використанні методів надлишкових вимірювань виключається вплив чутливості сенсора на результат вимірювань. При прямих методах необхідно постійно враховувати похибку, що обумовлена дією дестабілізуючих факторів.

Підтверджено, що усереднення багаторазових вимірювань сприяє підвищенню точності контролю за рахунок зменшення випадкових похибок. Показано, що для зменшення впливу випадкової складових похибки вимірювань на кінцевий результат можуть бути використані методи надлишкових вимірювань II-го роду з обробкою результатів проміжних вимірювань по першому і по другому підходах: багаторазове (10 разів) вимірювання фізичної величини в кожному такті вимірювань і багаторазове (10 разів) проведення циклу вимірювань.

Таким чином, запропоновані методи надлишкових вимірювань забезпечують автоматичне виключення систематичних складових похибок

результату вимірювання, що обумовлені зміною параметрів функції перетворення під дією дестабілізуючих факторів, а при використанні багаторазових вимірювань і статистичній обробці — зменшити і випадкові складові похибки загального характеру.

В цілому, методи надлишкових вимірах сприяють підвищенню точності вимірювання при лінійній і нелінійній функції перетворення на всьому її діапазоні з можливістю визначення параметрів функції перетворення.

Список використаних джерел до розділу 2

1. Luis Orozco**.** Optimizing Precision Photodiode Sensor Circuit Design // Analog devices. 2011.

2. Gencer Imer, Tombak A., Korkut A. Electrical and photoelectrical characteristic investigation of a new generation photodiode based on bromothymol blue dye // Journal of Physics: Conference Series. 2016.

3. Manajit Sengupta. Accuracy of Photodiode Pyranometers for Photovoltaic Applications // Fourth Conference on Weather, Climate, and the New Energy Economy. 2013.

4. Брагинец И. А., Кононенко А. Г., Масюренко Ю. А. Исследование высокочувствительного фотоприемного устройства на основе лавинного фотодиода для оптоэлектронных измерительных систем // Технічна електродинаміка. 2016. № 6. – С. 69–75.

5. Deyasi A., Ganguly A., Rakshit P.C. Calculating frequency response and photocurrent density of p-i-n photodiode // Proceedings of National conference on frontline research in computer, communication and device. 2015. P. 119–123.

6. Пронин А.Н., Сапожникова К.В., Тайманов Р.Е. Достоверность измерительной информации в системах управления. Проблемы и решения // T-Comm: Телекоммуникация и транспорт. 2015. Том 9. №3. С. 32–37.

7. Shcherban' V. Yu., Melnyk G. V., Sholudko M. I., Kolysko O. Z.,

Kalashnyk V. Yu. (2018). Yarn tension while knitting textile fabric / Fibres and Textiles. Issue 3. P. 74–83.

8. Рішан О.Й., Матвієнко Н.В. Структурні методи підвищення точності вимірювань в автоматичних системах дозування сипких матеріалів з використанням магнітопружних первинних вимірювальних перетворювачів зусилля // Наука технології інновації. 2014. №4: 47 – 51.

9. Яненко О. П., Михайленко С. В., Ліснічук А. С. Радіометричний модуляційний вимірювач інтенсивності оптичного випромінювання / Вісник Національного технічного університету України «КПІ» Серія – Радіотехніка. Радіоапаратобудування. 2014. №56: С. 96 – 101.

10. Ana Luz Muñoz Zurita, Joaquín Campos Acosta, Alejandro Ferrero Turrión and Alicia Pons Aglio. (2012). Photodiodes as Optical Radiation Measurement Standards / Intech. Open science | Open minds.

11. Zhen Su, Xiaolong Liang. (2011). Computation and analysis on the Volt-Ampere characteristics of photodiode sensor under the certain conditions / In Proc. of the 4th International Congress on Image and Signal Processing. P.2593 –2597.

12. Shcherban' V., Makarenko J., Petko A., Melnyk G., Shcherban' Yu., Shchutska G.(2020). Computer implementation of a recursion algorithm for determining the tension of a thread on technological equipment based on the derived mathematical dependences / Eastern-European Journal of Enterprise Technologies. Volume 104. – №2/1:41 – 50.

13. Черепанська І.Ю., Безвесільна О.М., Сазонов А.Ю. До питання підвищення точності кутових вимірювань гоніометричними системами // ВІСНИК ЖДТУ. 2016. № 1 (76): 92 – 99. ISSN 1728-4260.

14. Manal A. Haridy and Affia Aslam. (2018). Optical Radiation Metrology and Uncertainty. DOI: 10.5772 / intechopen.75205.

15. Shcherban' V., Makarenko J., Melnyk G., Shcherban' Y., Petko A., Kirichenko A. (2019). Effect of the yarn structure on the tension degree when

interacting with high-curved guides / Fibres and Textiles. Volume 26 – № 4 : 59 – 68.

16. Zhaoyan Fan; Robert X. Gao; Peng Wang; David O. Kazmer. (2016). Multi-sensor data fusion for improved measurement accuracy in injection molding / Международная конференция IEEE по измерительным и измерительным технологиям.

17. Gareth D. Lewis , Patrick Merken, Marijke Vandewal. (2018). Enhanced Accuracy of CMOS Smart Temperature Sensors by Nonlinear Curvature Correction // *Sensors* 2018. *18*(12), 4087.

18. В.М. Годованюк, І.В. Докторович, В.Г. Юр'єв, І.М. Фодчук, Є.О. Шорок (2018). Додаткові похибки, що виникають в освітлювачах. Вісник Хмельницького національного університету, №3, (261), с. 253–257.

19. Танкевич Є. М., Яковлєва І. В., Варський Г. М. (2016). Підвищення точності вимірювальних каналів напруги систем керування електроенергетичних об'єктів. Вісник Вінницького політехнічного інституту. № 1, с. 79–83.

20. М.М. Степаняк, В.Р. Скальський, М.В. Степаняк (2010). Дослідження можливості підвищення точності вимірювання температури обертових об'єктів. Вісник Національного університету "Львівська політехніка", № 686 : Комп'ютерні науки та інформаційні технології. – С. 13–23.

21. Jiabin Wu, Yunshan Chen, Shijie Gao, Yimang Li, and Zhiyong Wu. (2015). Improved measurement accuracy of spot position on an InGaAs quadrant detector. Applied Optics**,**Vol. 54, Issue 27, pp. 8049–8054.

22. José A. Hidalgo-López, Raquel Fernández-Ramos, Jorge Romero-Sánchez, José F. Martín-Canales and Francisco J. Ríos-Gómez. (2018) Improving Accuracy in the Readout of Resistive Sensor Arrays. Journal of Sensors , vol. 2018.

23. Luis Orozco**.** (2011).Optimizing Precision Photodiode Sensor Circuit

Design / Analog devices.

24. Jun Zhang, Weixian Qian, Guohua Gu, Chen Mao, Kan Ren, Chunhong Wu, Xiaoyu Peng, and Qian Chen. (2019). Improved algorithm for expanding the measurement linear range of a four-quadrant detector. Applied Optics,Vol. 58, Issue 28, pp. 7741–7748.

25. Hobbs, M. J., Tan, C. H. and Willmott, J. R. (2013). Evaluation of phase sensitive detection method and Si avalanche photodiode for radiation thermometry, JINST. 8(3), P03016. URL: https://iopscience.iop.org/article/10.1088/1748-0221/8/03/P03016.

26. Петровська Г., Демкович І. (2010). Пристрій для вимірювання поглинання тонкоплівкових покрить фототепловим методом. Теоретична електротехніка, вип. 61. С.128–134.

27. Riazimehr, S., Kataria, S., Bornemann, R., Haring Bolívar, P., Ruiz, F. J. G.; Engström, O., Godoy, A., Lemme, M. C. (2017). High Photocurrent in Gated Graphene–Silicon Hybrid Photodiodes. ACS Photonics, 4, 1506–1514.

28. Jiaqiong Qin, Shaobo Cui and Jun Dai (2020). Noise Analysis and Compensation Strategy of Photoelectric Detection Circuit. Jiaqiong Qin et al 2020 J. Phys.: Conf. Ser. 1601 022047.

29. Chun-Chi Chen , Chao-Lieh Chen, Yi Lin. (2016). All-Digital Time-Domain CMOS Smart Temperature Sensor with On-Chip Linearity Enhancement. Sensors (Basel);16(2):176.

30. V. Shcherban', G. Korogod, V. Chaban, O. Kolysko, Yu.Shcherban' and A.Shchutska. (2019). Computer simulation methods of redundant measurements with the nonlinear transformation function / Eastern-European Journal of Enterprise Technologies. Vol 2, No 5 (98):16 – 22.

31. Кондратов В.Т. Фундаментальная метрология: Теория структурного анализа уравнений избыточных и сверхизбыточных измерений. Сообщение 6. Ааналитические методы / Вимірювальна та обчислювальна техніка в технологічних процесах. 2017. No2. С. 199–215.

32. Кондратов В.Т. Проблемы фундаментальной метрологии XXI-го века и пути их решения / В.Т.Кондратов // Вісник Хмельницького національного університету. – 2011. – №4. – С. 59 – 70. – ISSN 2307-5732.

33. Кондратов В.Т. Основи теорії автоматичної корекції систематичних похибок вимірювання фізичних величин при нестабільній і нелінійній функції перетворення датчика: дис. д-ра техн. наук: 05.11.15 і 05.11.01/ Кондратов Владислав Тимофійович. – К., 2001. – Додаток. Т. 2 – 791 с.

34. Кондратов В.Т. Математические модели избыточных измерений I-го, II-го и III-го родов / В.Т.Кондратов // Фундаментальные и прикладные проблемы приборостроения, информатики и экономики: научные труды X-й Юбилейной Междунар. науч.-техн. конф.– М.: МГУ ПИ, 2007. – С. 134 – 143.

35. Кондратов В.Т. Математические модели избыточных измерений I-го, II-го и III-го родов / В.Т.Кондратов // Фундаментальные и прикладные проблемы приборостроения, информатики и экономики: научные труды X-й Юбилейной Междунар. науч.-техн. конф.– М.: МГУ ПИ, 2007. – С. 134 – 143.

36. Shcherban' V., Korogod G., Kolysko O., Kolysko M., Shcherban' Yu., Shchutska G. (2020). Computer simulation ofmultiple measurementsof logarithmictransformationfunction by twoapproaches. Eastern-European Journal of Enterprise Technologies. 6(4(108), 6–13.

37. Shcherban' V., Korogod G., Kolysko O., Kolysko M., Shcherban' Yu., Shchutska G. (2021). Computer simulation of logarithmic transformation function to expand the range of high-precision measurements. Eastern-European Journal of Enterprise Technologies. 2/9(110), 27–35.

38. Vladimir Y. Shcherban', Ganna A. Korogod, Oksana Z. Kolysko, Mariana I. Sholudko, Gennady V. Melnik, Vitaliy.V. Chaban, Yury Y. Shcherban'. (2020). Computer Implementation of Algorithmic Components of

Redundant Measurement Methods, 23-32.

39. Корогод Г.О. Методи та оптико-електронні засобивимірювального контролю температури розплавів скломас з використанням інформативної надлишковості: дис…канд.техн.наук: 05.11.13/ Корогод Ганна Олександрівна. – К., 2016. – 291 с.

3. АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ДЕТАЛЕЙ ГАЛАНТЕРЕЙНИХ ВИРОБІВ

3.1. КЛАСИФІКАЦІЯ ФОРМ ДЕТАЛЕЙ ГАЛАНТЕРЕЙНИХ ВИРОБІВ

З метою розширення асортименту та поліпшення якості до виробництва галантерейних виробів необхідно висувати все більш високі вимоги. Конструкція моделі сумок і технологічний процес їх складання повинні сприяти максимальній механізації та автоматизації процесів, зростання продуктивності праці та зниження собівартості продукції.

Тому вдосконалення технічної підготовки виробництва і, зокрема, створення високоефективного методу проектування сумок різних конструкцій є актуальним завданням.

В умовах ринкової економіки створення науково обґрунтованих методів проектування виробів шкіргалантереї з використанням обчислювальної техніки дозволить істотно скоротити терміни виконання робіт і підвищити продуктивність праці, поліпшити якість проектних рішень і, в кінцевому підсумку, підвищити ефективність роботи підприємств та автоматизація самого процесу проектування з ціллю отримання об’єктивних результатів за короткий період часу.

Частіше за все деталі шкіргалантерейних виробівмають просту конфігурацію, що представляють наступні прості фігури [1-12]: прямокутник (код 1), трапеція (код 2), трикутник (код 3), еліпс (код 4), пів еліпс (код 5), шестикутник (код 6) та комбінацію з двох або трьох простих фігур. Тоді для однозначного визначення форми деталей галантерейних виробів введемо код форми деталі, який буде складатися із трьох цифр, значення яких визначається в діапазоні від 0 до 6.

Тоді деталь, що має форму простої фігури, матиме наступні коди форми деталі: прямокутник (код 001), трапеція (код 002), трикутник (код 003), еліпс (код 004), пів еліпс (код 005), шестикутник (код 006). Деталь,

що створена в результаті поєднання двох простих фігур, матиме наступні коди форми деталі: прямокутник та пів еліпс (код 051), прямокутник та трапеція (код 021), прямокутник та трикутник (код 031), трикутник та пів еліпс (код 053), прямокутник та многокутник (код 031). Також можуть поєднуватись три простих фігури: прямокутник, трапеція та пів еліпс (код 521), прямокутник, трапеція та трикутник (код 321), прямокутник, шестикутник та трикутник (код 361). Наприклад горт має складну форму і його конфігурація складається з трьох базових: прямокутник, трапеція та трикутник. Тоді код буде складатись з коду простих форм 123 (прямокутник – 1, трапеція – 2, трикутник – 3).

Для однозначного визначення назви деталі введемо код назви деталі, який складається із двох цифр, значення яких визначається в діапазоні від 0 до 9. Приклади кодів назви деталей та форми деталей представлені в таблиці 1 нижче.

Для однозначного визначення назви матеріалу, з якого необхідно виготовити деталі, введемо код назви матеріалу деталі, який складається із двох цифр, значення яких визначається в діапазоні від 0 до 9.

При створенні програмного продукту для проектування деталей галантерейних виробів, першим і найголовнішим є створення вихідної бази даних. Для виконання цієї задачі ми повинні ідентифікувати кожну спроектовану деталь за назвою; за матеріалом, із якого необхідно виготовити деталь; за формою зовнішнього контуру деталі. Для цього було створено код для кожної деталі, який буде складатися із семи цифр. Перші дві цифри нестимуть інформацію про назву деталі, наступні дві цифри – інформацію про назву матеріалу, із якого виготовлена деталь та наступні три цифри – про форму деталі галантерейного виробу.

Проведена класифікація форм зовнішніх контурів деталей галантерейних виробів показала, що в більшості випадків форму зовнішнього контуру деталей галантерейних виробів можна представити

простими фігурами або комбінацією цих простих фігур. В таб. 3.1.1 представлені дані, що описують всі можливі форми деталей галантерейних [13-24].

Таблиця 3.1.1

Класифікація форм деталей галантерейних виробів

	Назва деталі	Код назви деталі	Код форми деталі															
			001	002	003	004	005	006	051	021	031	052	061	521	211	311	321	361
1	Стінка	01	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-
2	Кокетка	02	+	+	+	-	+	-	+	-	+	+	-	-	-	-	-	-
3	Фальда	03	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	Клинчик	04	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-
5	Боковинка	05	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	Ботан	06	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	Полотно	07	+	-	-	+	-	+	+	+	+	-	-	-	-	-	-	-
8	Дно	08	+	-	-	+	-	+	-	-	-	-	-	-	-	-	-	-
9	Цупфер	09	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+
10	Клапан	10	+	+	+	-	+	-	+	+	+	+	-	+	-	+	+	-
11	Горт	11	+	-	-	-	-	-	+	+	+	-	+	+	+	+	+	+
12	Запряжник	12	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	Шлевка	13	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	Петля	14	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	Накладка	15	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16	Кишеня вн.	16	+	+	-	-	+	-	-	-	-	-	-	-	-	-	-	-
17	Кишеня зовн.	17	+	+	+	+	+	-	+	+	+	+	-	-	-	-	-	-
18	Кедер	18	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	Окантовка	19	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	Обплітка	20	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	Перегородка	21	+	+	-	-	+	-	-	-	-	-	-	-	-	-	-	-
22	Середник	22	+	+	-	-	+	-	-	-	-	-	-	-	-	-	-	-
23	Ручка	23	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Наприклад, клапан може мати конфігурацію чотирьох простих форм: прямокутник (код форми деталі 001), трапеція (код форми деталі 002), трикутник (код форми деталі 003), пів еліпс (код форми деталі 005), та їх поєднання: прямокутник(1) та трапеція (2) (код форми деталі 021),

прямокутник (1) та пів еліпс (5) (код форми деталі 051), прямокутник (1) та трикутник (3)(код форми деталі 031), трапеція (3) та пів еліпс (5) (код форми деталі 052), прямокутник(1), трапеція(2) та пів еліпс(5) (код форми деталі 521), прямокутник(1), трапеція(2) та трикутник(3) (код форми деталі 321), прямокутник(1), прямокутник(1) та трикутник(3) (код форми деталі 311), (рис. 3.1.1).

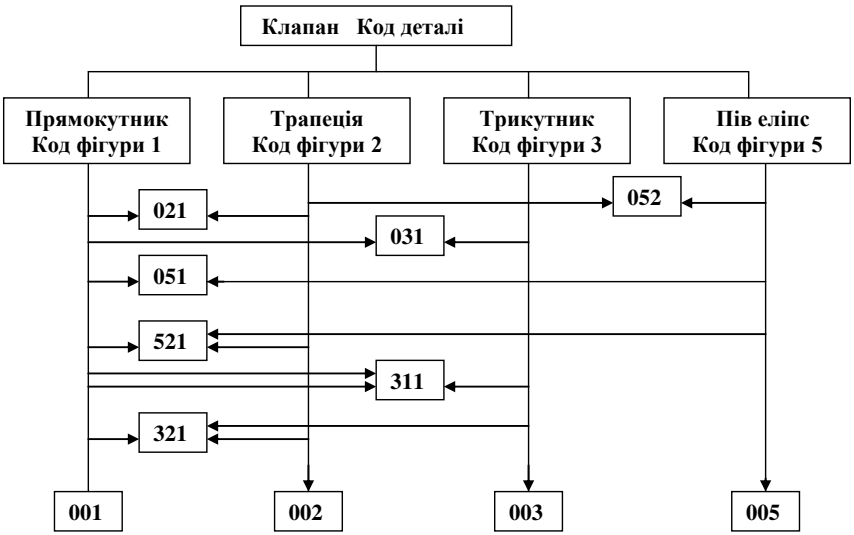


Рис. 3.1.1. **Конфігурація форм клапана**

Стінка може мати конфігурацію шести простих форм : прямокутник (код 001), трапеція (код 002), трикутник (код 003), еліпс (код 004), пів еліпс (код 005), шестикутник (код 006) та їх поєднання: прямокутник (1) та трапеція (2) – код 021, прямокутник (1) та пів еліпс (5) – код 051, прямокутник (1) та трикутник (3) – код 031, трапеція (2) та півколо (5) – код 052 (рис. 3.1.2).

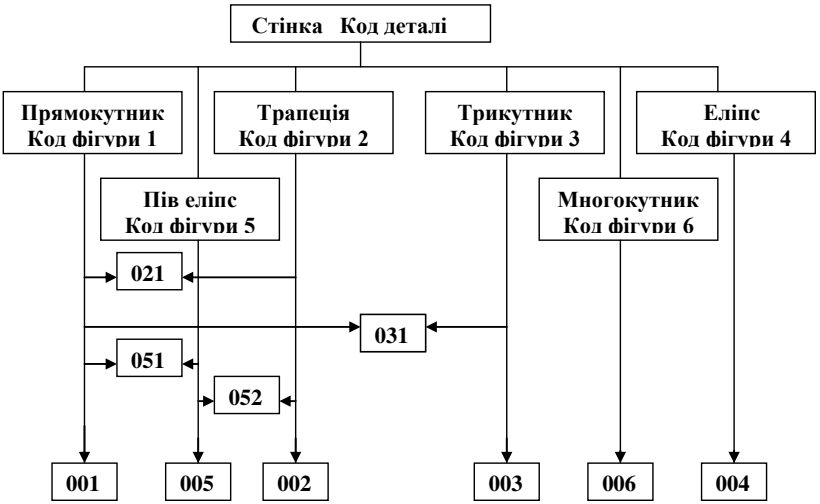


Рис. 3.1.2. **Конфігурація форм стінки галантерейного виробу**

Параметричні моделі деталей, що складаються з одного простого елементу

Щоб однозначно відобразити зовнішні контури деталей галантерейних виробів необхідно мати інформацію про зовнішні контури деталей. Але деталі галантерейних виробів в більшості випадків мають таку форму зовнішнього контуру, що його не можна описати аналітично. Тому в подальшому ми будемо апроксимувати зовнішні контури деталей. Найбільше розповсюдження за своєю універсальністю, можливістю автоматизації, точністю та простотою отримав кусково-лінійний метод апроксимації. Тому ми зупинено ся на цьому методі. При кусково-лінійному методі апроксимації зовнішній контур деталі з необхідною точністю представляє апроксимуючий многокутник. Для однозначного відображення апроксимуючого многокутника необхідно знати координати вершин цього многокутника (рис. 3.1.3) та порядок їх обходу, тобто необхідно мати масив пари чисел $\{X_i, Y_i\}$(де $i=1,2..n$, $X_1=X_n$ та $Y_1=Y_n$), який визначає координати вершин многокутника в порядку їх обходу.

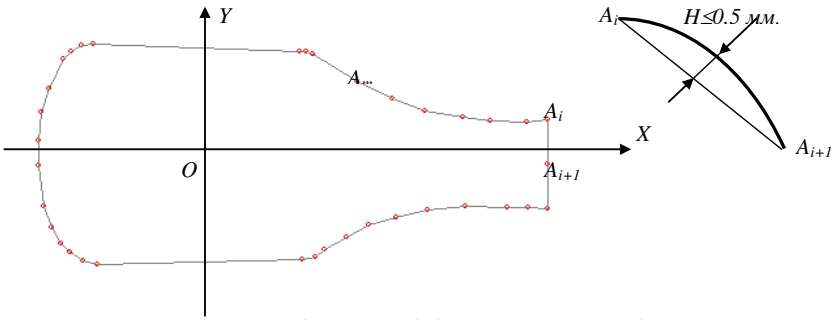


Рис. 3.1.3. **Кусково-лінійна апроксимація**

Для однозначного відображення контурів деталей в нашому випадку необхідно побудувати параметричну модель зовнішнього контуру деталей. Це означає знайти залежність між координат вершин апроксимуючого многокутника від параметрів, які однозначно визначають зовнішній контур деталі, тобто

$$\begin{cases} X_i = f_x^i(t_1, t_2..t_q) \\ Y_i = f_y^i(t_1, t_2..t_q) \end{cases},$$

де t_1, $t_2..t_q$ - відповідні параметри, які однозначно визначають зовнішній контур деталі.

Тоді будь-яку точку на контурі деталі між вершинами A_i A_{i+1} можна визначити наступним чином:

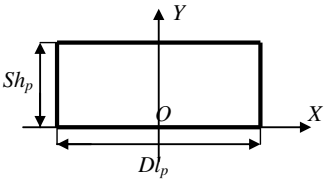
$$\begin{cases} X = X_i + (X_{i+1} - X_i) \cdot t \\ Y = Y_i + (Y_{i+1} - Y_i) \cdot t \end{cases}, \text{ де } 0 \le t \le 1,$$

або

$$\begin{cases} X = f_x^i(t_1, t_2..t_q) + (f_x^{i+1}(t_1, t_2..t_q) - f_x^i(t_1, t_2..t_q)) \cdot t \\ Y = f_y^i(t_1, t_2..t_q) + (f_y^{i+1}(t_1, t_2..t_q) - f_y^i(t_1, t_2..t_q)) \cdot t \end{cases}, \text{ де } 0 \le t \le 1.$$

Розглянемо параметричні моделі наступних зовнішніх форм деталей галантерейних виробів.

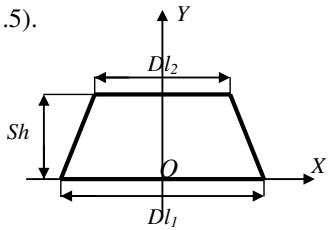
1) Прямокутна форма зовнішнього контуру деталі (код форми деталі 001; формальні параметри: довжина Dl_p та ширина прямокутника Sh_p)(рис. 3.1.4).



$X_1 = Dl_p/2;$ $Y_1 = 0;$
$X_2 = Dl_p/2;$ $Y_2 = Sh_p;$
$X_3 = Dl_p/2;$ $Y_3 = Sh_p;$
$X_4 = -Dl_p/2;$ $Y_4 = 0;$
$X_5 = Dl_p/2;$ $Y_5 = 0;$

Рис. 3.1.4. **Прямокутна форма зовнішнього контуру деталі**

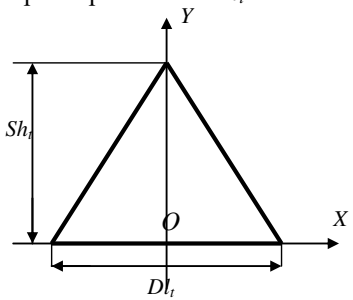
2) Трапецеїдальна форма зовнішнього контуру деталі (код форми деталі 002; формальні параметри: основи Dl_1, Dl_2 та висота трапеції Sh)(рис. 3.1.5).



$X_1 = Dl_1/2;$ $Y_1 = 0;$
$X_2 = Dl_2/2;$ $Y_2 = Sh;$
$X_3 = -Dl_2/2;$ $Y_3 = Sh;$
$X_4 = -Dl_1/2;$ $Y_4 = 0;$
$X_5 = Dl_1/2;$ $Y_5 = 0;$

Рис. 3.1.5. **Трапецеїдальна форма зовнішнього контуру деталі**

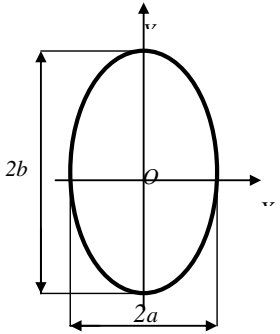
3) Прямокутна форма зовнішнього контуру деталі (код форми деталі 003; формальні параметри: основа Dl_t та висота трикутника Sh_t)(рис. 3.1.6).



$X_1 = Dl_t/2;$ $Y_1 = 0;$
$X_2 = 0;$ $Y_2 = Sh_t;$
$X_3 = -Dl_t/2;$ $Y_2 = 0;$
$X_4 = Dl_t/2;$ $Y_4 = 0;$

Рис. 3.1.6. **Трикутна форма зовнішнього контуру**

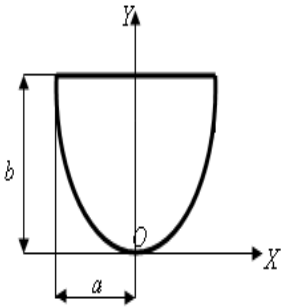
4) Еліпсоподібна форма зовнішнього контуру деталі[13] (код форми деталі 004; формальні параметри: півосі еліпса *a* та *b*)(рис. 3.1.7).



$X_i=a\cdot cos\ \varphi_i;$
$Y_i=b\cdot sin\ \varphi_i;$
де $\varphi_i=2\pi\cdot i/n,\ i=0,1,2..n$

Рис. 3.1. 7. **Еліпсоподібна форма зовнішнього контуру**

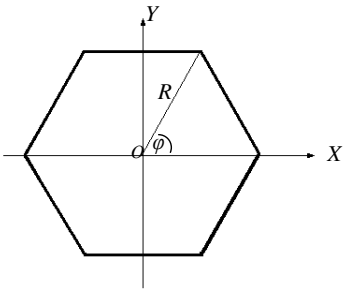
5) Зовнішній контур деталі має форму напівеліпса (код форми деталі 005; формальні параметри: півосі еліпсу *a* та *b*)(рис. 3.1.8).



$X_i=a\cdot cos\ (\varphi_i+\pi);$
$Y_i=b\cdot sin\ (\varphi_i+\pi)+b;$
$X_{n+1}=X_1;\ Y_{n+1}=Y_1;$
де $\varphi_i=\pi\cdot i/n,\ i=0,1,2..n$

Рис. 3.1.8. **Форма зовнішнього контуру деталі подібна напівеліпсу**

6)Зовнішній контур деталі має форму шестикутниках[13] (код форми деталі 006; формальні параметри: радіус *R* описаного кола навколо шестикутника)(рис. 3.1.9).



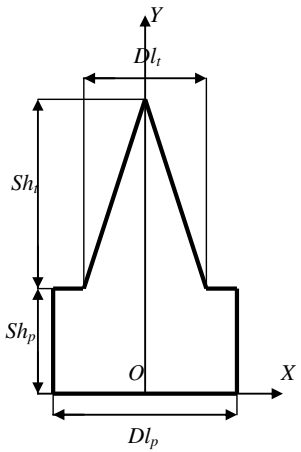
$X_i=R\cdot cos(i\cdot \varphi);$
$Y_i=R\cdot sin(i\cdot \varphi);$
де $i=0,1,2..6;$
$\varphi=2\pi/6.$

Рис. 3.1.9. **Форма зовнішнього контуру деталі подібна**

Параметричні моделі деталей, що складаються з комбінації двох простих елементів

Розглянемо конфігурації деталей, що складаються з комбінації двох простих елементів та їх параметричні моделі.

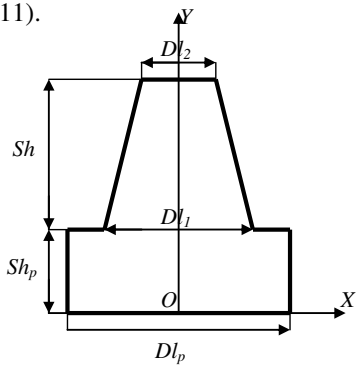
1) Зовнішній контур деталі складається із комбінації прямокутника та трикутника (код форми деталі 031; формальні параметри: довжина Dl_p та ширина Sh_p прямокутника, основа Dl_t та висота Sh_t трикутника)(рис. 3.1.10).



$X_1=-DL_p/2;$ $Y_1=Sh_p;$
$X_2=-DL_p/2;$ $Y_2=0;$
$X_3=DL_p/2;$ $Y_3=0;$
$X_4=DL_p/2;$ $Y_4=Sh_p;$
$X_5=DL_t/2;$ $Y_5=Sh_p;$
$X_6=0;$ $Y_6=Sh_{t+}\ Sh_p;$
$X_7=-DL_t/2;$ $Y_7=Sh_p;$
$X_8=-DL_p/2;$ $Y_8=Sh_p;$

Рис. 3.1.10. **Форма зовнішнього контуру деталі складається із комбінації прямокутника та трикутника**

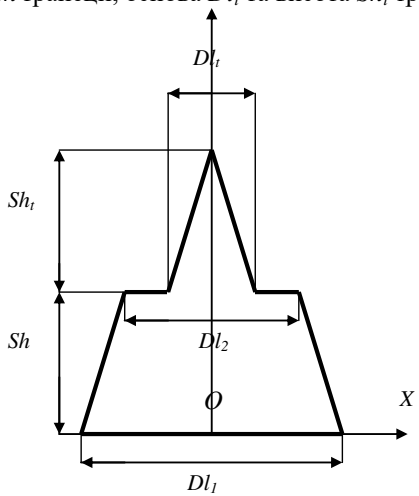
2) Зовнішній контур деталі складається із комбінації прямокутника та трапеції (код форми деталі 021; формальні параметри: довжина Dl_p та ширина Sh_p прямокутника, основи Dl_1, Dl_2 та висота Sh трапеції)(рис. 3.1.11).



$X_1=-DL_p/2;$ $Y_1=Sh_p;$
$X_2=-DL_p/2;$ $Y_2=0;$
$X_3=DL_p/2;$ $Y_3=0;$
$X_4=DL_p/2;$ $Y_4=Sh_p;$
$X_5=DL_1/2;$ $Y_5=Sh_p;$
$X_6=Dl_2/2;$ $Y_6=Sh_+ Sh_p;$
$X_7=-Dl_2/2;$ $Y_7=Sh_+ Sh_p;$
$X_8=-DL_1/2;$ $Y_8=Sh_p;$
$X_9=-DL_p/2;$ $Y_9=Sh_p;$

Рис. 3.1.11. **Форма зовнішнього контуру деталі складається із комбінації прямокутника та трапеції**

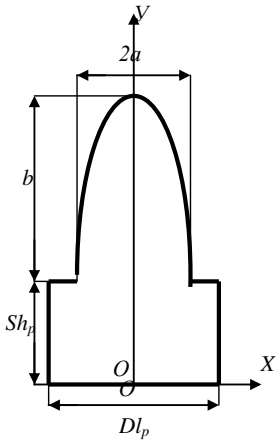
3) Зовнішній контур деталі складається із комбінації прямокутника та трапеції (код форми деталі 032; формальні параметри: основи Dl_1, Dl_2 та висота Sh трапеції, основа Dl_t та висота Sh_t трикутника)(рис. 3.1.12).



$X_1=-DL_2/2;$ $Y_1=Sh;$
$X_2=-DL_1/2;$ $Y_2=0;$
$X_3=DL_1/2;$ $Y_3=0;$
$X_4=DL_2/2;$ $Y_4=Sh;$
$X_5=DL_t/2;$ $Y_5=Sh;$
$X_6=0;$ $Y_6=Sh_+ Sh_t;$
$X_7=-DL_t/2;$ $Y_7=Sh;$
$X_8=-DL_2/2;$ $Y_8=Sh;$

Рис. 3.1.12 **Форма зовнішнього контуру деталі складається із комбінації трапеції та трикутника**

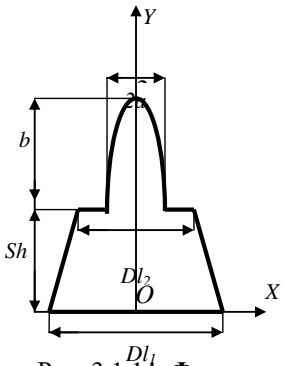
4) Зовнішній контур деталі складається із комбінації прямокутника та напівеліпса(код форми деталі 051; формальні параметри: довжина Dl_p та ширина Sh_p прямокутника, півосі еліпса a та b)(рис. 3.1.13).



$X_i=a\cdot\cos\varphi_i;$
$Y_i=b\cdot\sin\varphi_i+Sh_p;$
$де\ \varphi_i=\pi\cdot i/n,\ i=0,1,2..n$
$X_{n+1}=-Dl_p/2;\ Y_{n+1}=Sh_p;$
$X_{n+2}=-Dl_p/2;\ Y_{n+2}=0;$
$X_{n+3}=Dl_p/2;\ Y_{n+3}=0;$
$X_{n+4}=Dl_p/2;\ Y_{n+4}=Sh_p;$
$X_{n+5}=X_1;\ Y_{n+5}=Y_1;$

Рис. 3.1.13. **Форма зовнішнього контуру деталі складається із комбінації прямокутника та напівеліпса**

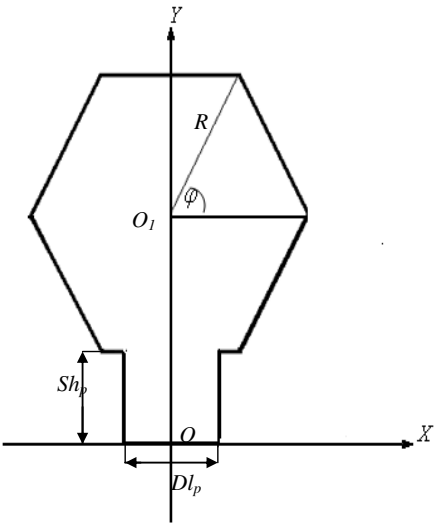
5) Зовнішній контур деталі складається із комбінації трапеції та напівеліпса(код форми деталі 052; формальні параметри: основи Dl_1, Dl_2 та висота Sh трапеції, півосі еліпса a та b)(рис. 3.1.14).



$X_i=a\cdot\cos\varphi_i;$
$Y_i=b\cdot\sin\varphi_i+Sh;$
$де\ \varphi_i=\pi\cdot i/n,\ i=1,2..n$
$X_{n+1}=-Dl_2/2;\ Y_{n+1}=Sh;$
$X_{n+2}=-Dl_1/2;\ Y_{n+2}=0;$
$X_{n+3}=Dl_1/2;\ Y_{n+3}=$

Рис. 3.1.14. **Форма зовнішнього контуру деталі складається із комбінації трапеції та напівеліпса**

6) Зовнішній контур деталі складається із комбінації прямокутника та шестикутника (код форми деталі 061; формальні параметри: довжина Dl_p та ширина Sh_p прямокутника, радіус описаного кола R навколо шестикутника)(рис. 3.1.15).



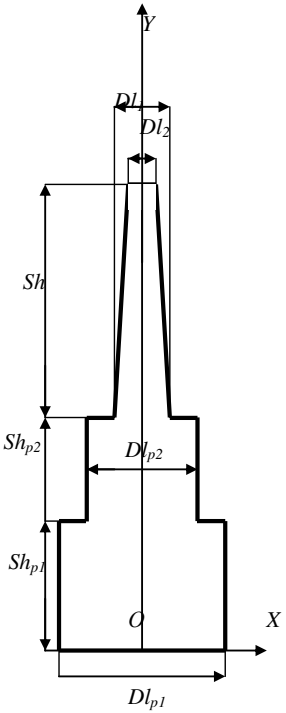
$X_i=R\cdot cos(i\cdot \varphi);$
$Y_i=R\cdot sin(i\cdot \varphi)+R+Sh_p;$
$de\ i=0,1,2..5;$
$\varphi=2\pi/6;$
$X_6=-Dl_p/2;\ Y_6=Sh_p;$
$X_7=-Dl_p/2;\ Y_7=0;$
$X_8=Dl_p/2;\ Y_8=0;$
$X_9=Dl_p/2;\ Y_9=Sh_p;$
$X_{10}=X_0;\ Y_{10}=Y_{0p};$

Рис. 3.1.15. **Форма зовнішнього контуру деталі складається із комбінації прямокутника та шестикутника**

Параметричні моделі деталей, що складаються з комбінації трьох простих елементів

Розглянемо конфігурації деталей, що складаються з комбінації двох простих елементів та їх параметричні моделі [25].

1) Зовнішній контур деталі складається із комбінації прямокутника , прямокутника та трапеції (код форми деталі 211; формальні параметри: довжина Dl_{p1} та ширина Sh_{p1} першого прямокутника, довжина Dl_{p2} та ширина Sh_{p2} другого прямокутника,основи Dl_1, Dl_2 та висота Sh трапеції)(рис. 3.1.16).



$X_1=-Dl_{p1}/2;$ $Y_1=Sh_{p1};$
$X_2=-Dl_{p1}/2;$ $Y_2=0;$
$X_3=Dl_{p1}/2;$ $Y_3=0;$
$X_4=Dl_{p1}/2;$ $Y_4=Sh_{p1};$
$X_5=Dl_{p2}/2;$ $Y_5=Sh_{p1};$
$X_6=Dl_{p2}/2;$
$Y_6=Sh_{p1}+Sh_{p2};$
$X_7=Dl_1/2;$
$Y_7=Sh_{p1}+Sh_{p2};$
$X_8=Dl_2/2;$
$Y_8=Sh_{p1}+Sh_{p2}+Sh;$
$X_9=-Dl_2/2;$
$Y_9=Sh_{p1}+Sh_{p2}+Sh;$
$X_{10}=-Dl_1/2;$
$Y_9=Sh_{p1}+Sh_{p2};$
$X_{11}=Dl_{p2}/2;$
$Y_{11}=Sh_{p1}+Sh_{p2};$
$X_{12}=Dl_{p1}/2;$ $Y_{12}=Sh_{p1};$

Рис. 3.1.16. **Форма зовнішнього контуру деталі складається із комбінації двох прямокутників та трапеції**

3.2. АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ ДЕКОРАТИВНИХ ЕЛЕМЕНТІВ ДЛЯ ГАЛАНТЕРЕЙНИХ ВИРОБІВ

Автоматизоване проектування декоративних елементів, які представляють собою окремі деталі

З метою розширення асортименту та поліпшення якості до виробництва сумок необхідно висувати все більш високі вимоги. Конструкція моделі сумок і технологічний процес їх складання повинні сприяти максимальній механізації та автоматизації процесів, зростання продуктивності праці та зниження собівартості продукції.

Тому вдосконалення технічної підготовки виробництва, зокрема, створення високоефективного методу проектування сумок різних конструкцій є актуальним завданням.

Розробка автоматизованого методу проектування сумок дозволить істотно скоротити терміни виконання робіт і підвищити продуктивність праці, поліпшити якість проектних рішень, зрештою, підвищити ефективність роботи підприємств.

Прикраси галантерейного виробу відносяться до додаткових зовнішнім деталям, що не утворюють корпус виробу, а призначені для допоміжних цілей. Ці деталі надзвичайно різноманітні, до них пред'являють підвищені естетичні вимоги. Часто ті чи інші додаткові деталі, їх вигляд, форма, розміри характеризують стиль оформлення виробів і напрям моди.

Розглянемо такі додаткові деталі як прикраси вироби, а саме: банти, квіти і т.д. [31-37].

Так як деталі прикрас будуть мати складну конфігурацію і їх зовнішній контур в більшості випадках неможливо описати аналітично, то ми будемо апроксимувати їх зовнішні контури з необхідною точністю багатокутниками. Багатокутник може бути однозначно відображений, якщо ми будемо знати координати вершин цього багатокутника і порядок їх проходження. Будь вершина A_i багатокутника однозначно визначається її координатами (X_i, Y_i) в Декартові системі координат.

Бант складатиметься з двох деталей. Для однозначного визначення параметрів кожної з деталей необхідно знати наступні параметри (рис.3.2.1): сторона А - *A*, сторона В - *B*, верхня ширина банта - *ShV*, нижня ширина банта - *ShN*.

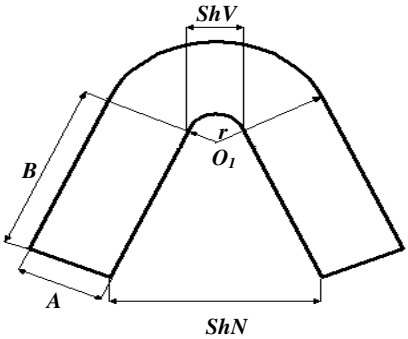


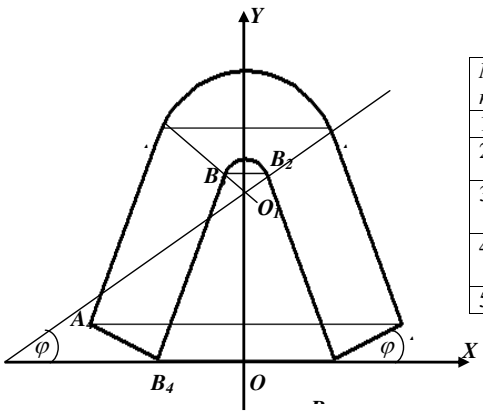
Рис. 3.2.1. **Параметри, які однозначно визначають форму деталі**

Побудова деталі банта по її параметрах здійснюється наступним чином:

- Будуємо трапеції $A_1A_2A_3A_4$ и $B_1B_2B_3B_4$ (рис.3.2.2);

- визначаємо кут φ;

- визначаємо радіуси *r, R* та координати точки $O_1(Xo_1,Yo_1)$;



N n/n	X_{bi}	Y_{bi}
1	*ShN/2*	0
2	*ShV/2*	$\sqrt{B^2-(ShN-ShV)^2}$
3	*- ShV/2*	$\sqrt{B^2-(ShN-ShV)^2}$
4	*- ShN/2*	0
5	*ShN/2*	0

Рис. 3.2.2. **Побудова деталі банту за її параметрами**

Кут $\varphi = \frac{\pi}{2} - \angle B_4B_1B_2$. Так як $tg\angle B_4B_1B_2 = \frac{\sqrt{B^2 - ((ShN - ShV)/2)^2}}{ShN - ShV}$, то

$$\varphi = \frac{\pi}{2} - arctg(\frac{\sqrt{B^2 - ((ShN - ShV)/2)^2}}{ShN - ShV}). \qquad (3.2.1)$$

Тоді координати вершин трапеції $A_1A_2A_3A_4$ будуть визначаться наступним чином:

N n/n	Xa_i	Ya_i
1	$ShN/2 + A\cos\varphi$	$A\sin\varphi$
2	$ShV/2 + A\cos\varphi$	$A\sin\varphi + \sqrt{B^2 - (ShN - ShV)^2}$
3	$-ShV/2 - A\cos\varphi$	$A\sin\varphi + \sqrt{B^2 - (ShN - ShV)^2}$
4	$-ShN/2 - A\cos\varphi$	$A\sin\varphi$
5	$ShN/2$	$A\sin\varphi$

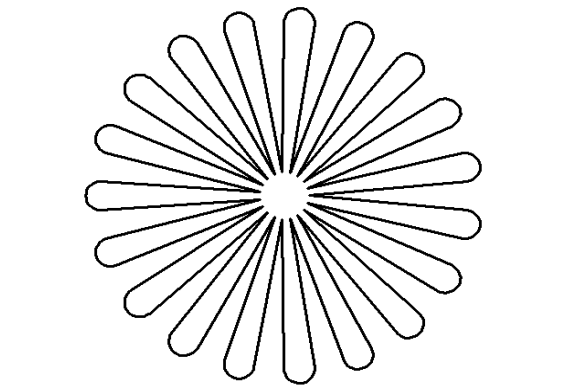
Радіуси *r, R* та координати точки $O_1(Xo_1, Yo_1$ будуть визначаться наступним чином: $r = \mathrm{ShV}/\cos\varphi;$ $R=r+A;$ (3.2.2)

$$Xo_1=0; \quad Yo_1=\sqrt{B^2 - (ShN - ShV)^2} - r\cdot\sin\varphi. \qquad (3.2.3)$$

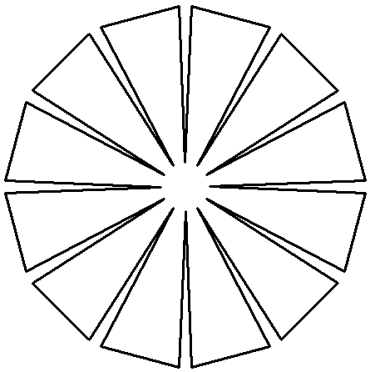
Знаючи значення радіусів *r, R* та координат точки $O_1(Xo_1, Yo_1)$ легко описати дуги A_2A_3 та B_2B_3:

$$\begin{aligned} XdA_i &= R\cdot\cos(\varphi + i\cdot\Delta\varphi) + Xo_1 \\ YdA_i &= R\cdot\sin(\varphi + i\cdot\Delta\varphi) + Yo_1 \\ XdB_i &= r\cdot\cos(\varphi + i\cdot\Delta\varphi) + Xo_1 \\ YdB_i &= r\cdot\sin(\varphi + i\cdot\Delta\varphi) + Yo_1 \end{aligned}, \quad \text{де} \quad \begin{aligned} i &= 0,1...n \\ \Delta\varphi &= 2(\pi - \varphi)/n \end{aligned} \qquad (3.2.4)$$

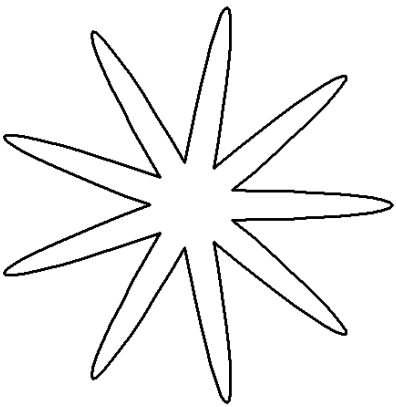
Розглянемо проектування декоративних елементів у вигляді квітів на прикладах генерування деталей ромашки та троянди. Для проектування цих деталей необхідно знати наступні параметри деталі: *Kl*-кількість пелюсток, *R*-зовнішній радіус, *r*-внутрішній радіус, *k* (*k*> *2*) - коефіцієнт. Приклади згенерованих ромашки і троянди представлені на рис 3.2.3



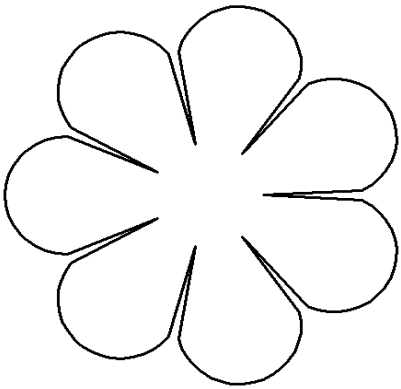
а



б



в



г

Рис. 3.2. 3. **Проектування ромашки та троянди**
а – ромашка_1(параметри: *Kl=19, R=45, r=6, k=4*)
б– ромашка_2(параметри: *Kl=12, R=45, r=6, k=14*)
в– ромашка_3(параметри: *Kl=9, R=45, r=10*)
г – троянда(параметри: *Kl=7, R=45, r=16, k=24*).

Для побудови квітки необхідно (рис.3.2.4):

- побудувати правильні *Kl*-кутники (*Kl*-кількість пелюсток квітки), вписані в кола радіусів *R* і *r*;

- визначити координати точок B_i, $i = 1,2 .. Kl$, які є середні точки відрізків A_iA_{i+1};

- визначити r_0, d;

- визначити кут φ;

- побудувати дуги кіл радіусу r_0, з центром в точці B_i, $i = 1,2 .. Kl$ з початковим кутом φ_i та кінцевим кутом $\varphi_i + \pi$.

Для побудови правильних *Kl*-кутників (*Kl*-кількість пелюсток квітки), вписаних в кола радіусів *R* та *r* достатньо визначити координати вершин цих багатокутників:

$$\begin{aligned} Xa_i &= R \cdot \cos\beta_i \\ Ya_i &= R \cdot \sin\beta_i \\ Xc_i &= r \cdot \cos\beta \\ Yc_i &= r \cdot \sin\beta_i \end{aligned}, \text{ де } \begin{aligned} &i = 1,2..Kl \\ &\beta_i = 2\pi \cdot i / Kl \end{aligned}. \qquad (3.2.5)$$

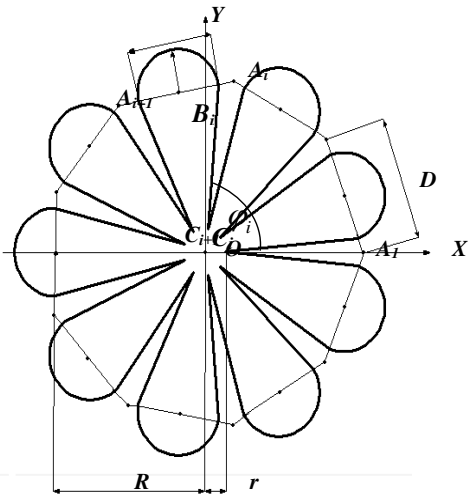


Рис. 3.2.4. **Побудова креслення квітки**

Координати точок B_i, $i=1,3.2..Kl$ визначаються наступним чином:

$$\begin{matrix} Xb_i = (Xa_i + Xa_{i+1})/2 \\ Yb_i = (Ya_i + Ya_{i+1})/2 \end{matrix}, \text{ де } \begin{matrix} i = 1,2..Kl \\ Xa_{kl+1} = Xa_1 \\ Ya_{kl+1} = Ya_1 \end{matrix}. \qquad (3.2.6)$$

Так як за визначенням $k = 2D/(D-d)$, то

$$d = \frac{D \cdot (k-2)}{k}. \qquad (3.2.7)$$

Звідси $$r_0 = d/2 = \frac{D \cdot (k-2)}{2k}. \qquad (3.2.8)$$

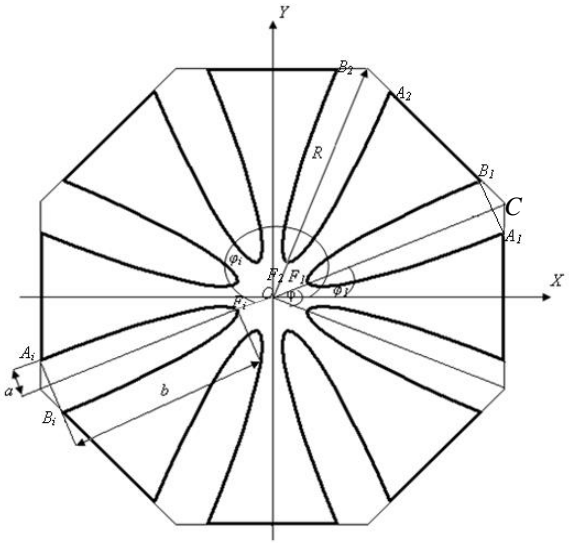


Рис. 3.2.5. **Побудова креслення волошки**

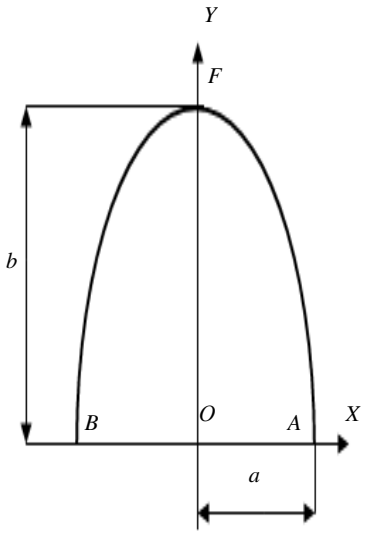
Кут φ_i визначаються наступним чином: $\varphi_i = (i-1) \cdot \alpha + \alpha / k$, де $\alpha = 2\pi / Kl$. Тоді *i*–у дугу квітки можна представити у вигляді многокутника, визначивши координати його вершин (Xd_{ij}, Yd_{ij}), наступним чином: [25-26]:

$$\begin{matrix} Xd_{ij} = r_0 \cdot \cos(\varphi_i + j \cdot \pi / N) + Xb_i \\ Yd_{ij} = r_0 \cdot \cos(\varphi_i + j \cdot \pi / N) + Xb_i \end{matrix}, \quad \text{де} \quad \begin{matrix} i = 1,2..Kl \\ j = 0,1..N \end{matrix}. \qquad (3.2.9)$$

Розглянемо проектування декоративних елементів у вигляді волошки(рис. 3.2.5).

Для проектування цих деталей необхідно знати наступні параметри деталі: *Kl*-кількість пелюсток, *R*-зовнішній радіус волошки.

Координати вершин апроксимуючого многокутника для зовнішнього контуру пів еліпсу, що зображений на рис. 3.6 можна представити наступним чином (3.2.10-3.2.12):



Координати вершин:

$X_i = a \cdot Cos\theta_i$, де $i = 0,1,2..N$ (3.2.10)

$Y_i = b \cdot Sin\theta_i$

$\theta = \pi / N$ та $\theta_i = i \cdot \theta$ (3.2.11)

$$N \geq \frac{2\pi}{arcCos(1-\frac{\varepsilon}{R})} \quad (3.2.12)$$

де ε - точність апроксимації зовнішнього контуру напівеліпсу; N - кількість вершин многокутника, який апроксимує напівеліпс з точністю ε; $R = \max\{a,b\}$.

Рис. 3.2.6. **Представлення інформації про зовнішній контур дуги пів еліпса із заданою точністю** ε

Якщо уважно розглянути рис. 3.2.5, то можна побачити, що зовнішній контур волошки складається із дуг пів еліпсів $B_iF_iA_i$ (де $i=1,3.2..P$ та P-кількість пелюсток у волошки) та прямих B_iA_{i+1}, що з'єднують останню вершину попереднього пів еліпса з першою вершиною наступного пів еліпса.

Для того, щоб отримати дугу пів еліпса $B_iF_iA_i$ необхідно точку O пів еліпса ABC перемістити в точку O_i та повернути пів еліпс ABC на кут $\psi_i=\pi/2+\varphi_i= (\pi+\varphi)/2+\varphi\cdot(i-1)$, де $\varphi=2\pi/P$.

Тоді координати вершин апроксимуючого многокутника для дуги пів еліпса $B_jF_jA_j$, де $j=1,3.2..P$, матимуть наступний вигляд:

$$\begin{cases} X_i^j = (X_i - Xo_i)\cdot\cos\psi_i - (Y_i - Yo_j)\cdot\sin\psi_i + Xf_i \\ Y_i^j = (X_i - Xo_i)\cdot\sin\psi_i + (Y_i - Yo_j)\cdot\cos\psi_i + Yf_i \end{cases} \quad (3.2.13)$$

Визначимо координати точок O_j . Для чого визначимо довжину відрізка OO_j .

$OO_j= OO_1= OC_1- O_1C_1=R-a\cdot tg\omega$, де $\omega=(\pi-\pi(P-2)/P)/2=\pi/P$. (3.2.14)

Тоді $Xo_j=OO_j\cdot cos\ \omega$ та $Yo_j=OO_j\cdot sin\ \omega$. (3.2.15)

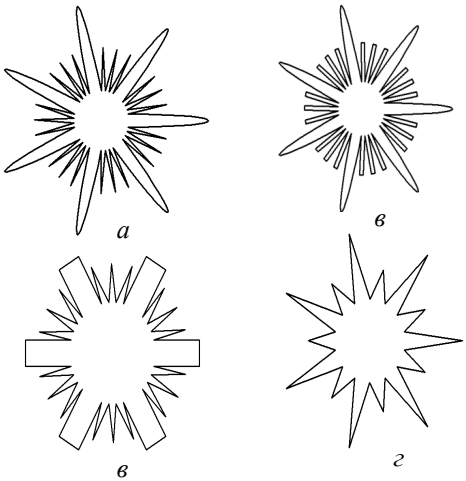


Рис. 3.2. 7. **Проектування зірок**

а – зірка, складовими елементами якої є трикутники та пів еліпси (параметри: Kl=7, R_1=45, R_2=25, a=7, b=30, k_l=3)

б– зірка, складовими елементами якої є чотирикутники та пів еліпси (параметри: Kl=7, R_1=45, R_2=35, a=7, b=14, k_l=3)

в– зірка, складовими елементами якої є трикутники та прямокутники (параметри: Kl=6, R_1=45, R_2=35, r=20, k_l=3)

г – зірка, складовими елементами якої є два види трикутників (параметри: Kl=6, R_1=45, R_2=35, r=20, k_l=1).

Маючи координати вершин апроксимуючого многокутника для дуги пів еліпса $B_jF_jA_j$ (3.2.13) та використавши вирази (3.2.10-3.2.15) легко відтворити координати вершин апроксимуючого многокутника для зовнішнього контуру волошки.

Розглянемо проектування декоративних елементів у вигляді зірок. Для проектування цих деталей необхідно знати наступні параметри деталі: *Kl*-кількість пелюсток, R_1-більший зовнішній радіус зірки, R_2-менший зовнішній радіус зірки, *r*- внутрішній радіус зірки, півосі *а* та *в* (у випадку коли елементами зірки є пів еліпси), *k_l*-кількість ланцугів елементів з радіусом $R_{3.2.}$

Приклад згенерованих зірок представлений на рис 3.2.7.

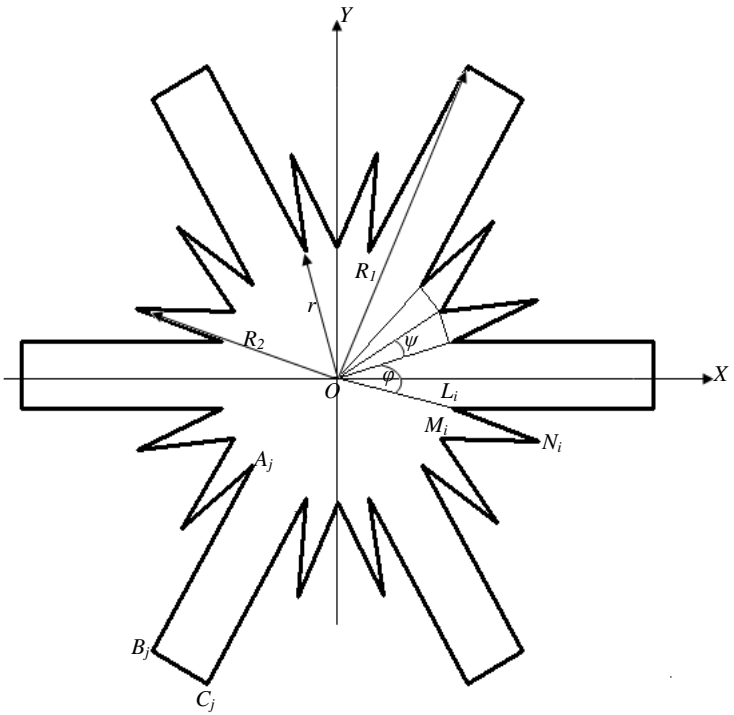
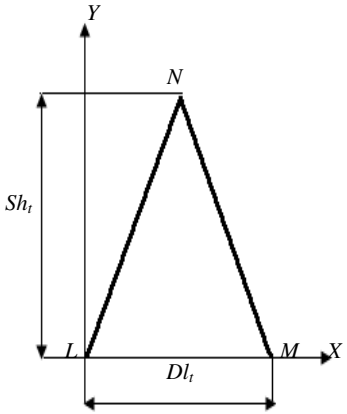


Рис. 3.2.8. **Побудова креслення зірки, зовнішній контур якої представляє комбінацію трикутників та прямокутників**

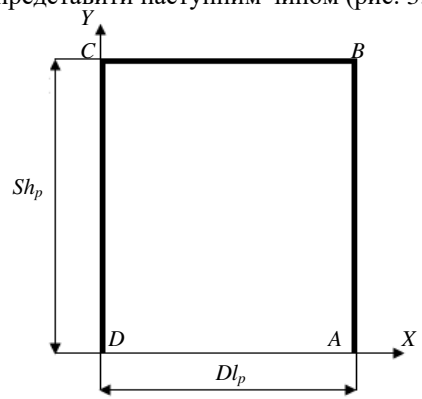
Якщо уважно розглянути рис. 3.2.8, то можна побачити, що зовнішній контур зірки представляє комбінацію трикутників та прямокутників.



N p/p	Xt_k	Yt_k
1	Dl_t	*0*
2	$Dl_t/2$	Sh_t
3	*0*	*0*

Рис. 3.2.9. **Представлення інформації про зовнішній контур трикутника**

Координати вершин трикутника можна представити наступним чином (рис. 3.2.9), а координати вершин прямокутника можна представити наступним чином (рис. 3.2.10).



N p/p	Xp_k	Yp_k
1	Dl_p	*0*
2	Dl_p	Sh_p
3	*0*	Sh_p
4	*0*	*0*

Рис. 3.2.10. **Представлення інформації про зовнішній контур прямокутника**

Для того, щоб отримати елемент зірки $M_iN_iL_i$ необхідно точку M трикутника MNL перемістити в точку M_i та повернути на кут ψ_i:

$$\psi_i = \begin{cases} \varphi/2 + 2\varphi\cdot\left(\left[\frac{i}{k_l}\right]\cdot k_l - 1\right) + (k_l - 1)\cdot\psi - \pi/2 & i = \left[\frac{i}{k_l}\right]\cdot k_l \\ \varphi/2 + 2\varphi\cdot\left(\left[\frac{i}{k_l}\right]\cdot k_l\right) + (i - \left[\frac{i}{k_l}\right]\cdot k_l - 1)\cdot\psi - \pi/2 & i \neq \left[\frac{i}{k_l}\right]\cdot k_l \end{cases} \quad \text{коли}$$

(3.2.16)

де $k=1,2,3$ та $i= 1,3.2..kl\cdot k_l$.

Тобто $$\begin{cases} Xt_k^i = Xm_i + (Xt_k - Xm_i)\cdot\cos\psi_i - (Yt_k - Ym_i)\cdot\sin\psi_i \\ Yt_k^i = Ym_i + (Yt_k - Ym_i)\cdot\sin\psi_i + (Yt_k - Ym_i)\cdot\cos\psi_i \end{cases},$$

(3.2.17)

де $k=1,2,3$, $i= 1,3.2..kl\cdot k_l$, $Xm_i=r\cdot cos(\psi_i+\pi/2)$ та $Ym_i=r\cdot cos(\psi_i+\pi/2)$.

Для того, щоб отримати елемент зірки $A_jB_jC_jD_j$ необхідно точку прямокутника $ABCD$ перемістити в точку A_jта повернути на кут ω_j:

$\omega_j= 2\varphi\cdot(j-1)-\pi/2$, де $j=1.3.2..kl$ та $\varphi=2\pi/kl$.

(3.2.18)

Тобто $$\begin{cases} Xp_k^j = Xa_j + (Xp_k - Xa_j)\cdot\cos\omega_j - (Yp_k - Ya_j)\cdot\sin\omega_j \\ Yp_k^j = Ya_j + (Yp_k - Ya_j)\cdot\sin\omega_j + (Yp_k - Ya_j)\cdot\cos\omega_j \end{cases},$$

(3.2.19)

де $k=1,2,3,4$, $j= 1,3.2..kl$, $Xa_j=r\cdot cos(\varphi/2+ \varphi\cdot(j-1))$ та $Ya_j=r\cdot sin(\varphi/2+ \varphi\cdot(j-1))$.

Маючи координати вершин елементів зірки $M_iN_iL_i$ та $A_jB_jC_jD_j$ легко відтворити координати вершин многокутника, який однозначно відображає зовнішній контур зірки.

Автоматизоване проектування декоративних елементів, які змінюють окремі ділянки зовнішніх контурів деталей

Часто при проектуванні деталей галантерейних виробів виникає необхідність мати можливість змінити конфігурацію країв деталей прикрасивши їх зубцями трикутної, прямокутної або трапецеїдальної форми (рис. 3.2.11-3.2.13).

Розглянемо основні етапи алгоритму генерування зубців на зовнішньому контурі деталі галантерейного виробу:

1. Виділення ділянки на зовнішньому контурі деталі, на якій будуть проектуватися зубці заданої форми;
2. Внесення початкової та кінцевої точок виділеної ділянки на зовнішньому контурі деталі в масив вершин зовнішнього контуру деталі;
3. Циклічне переміщення вершин зовнішнього контуру деталі таким чином, щоб вершини виділеної ділянки перемістились у “хвіст” массиву вершин деталі;
4. Побудова еквідистанти на заданій величині H до виділеної ділянки на зовнішньому контурі деталі таким чином, щоб всі вершини знаходилися всередині деталі;
5. Перетворення зовнішнього контуру еквідистанти таким чином, щоб він представляв ломану лінію із заданою довжиною сторін R;
6. Побудувати ланцюг зубців на ломаній лінії еквідистани ;
7. Замінити в масиві вершин зовнішнього контуру деталі виділену ділянку на ланцюг зубців.

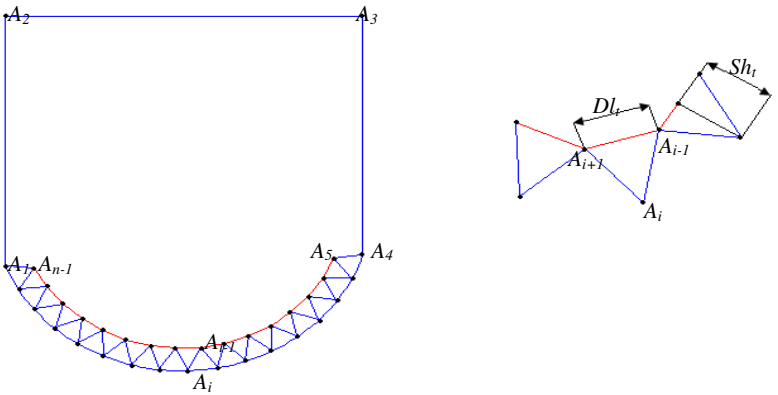


Рис. 3.2.11. **Представлення ділянки на зовнішньому контурі деталі у вигляді зубців трикутної форми**

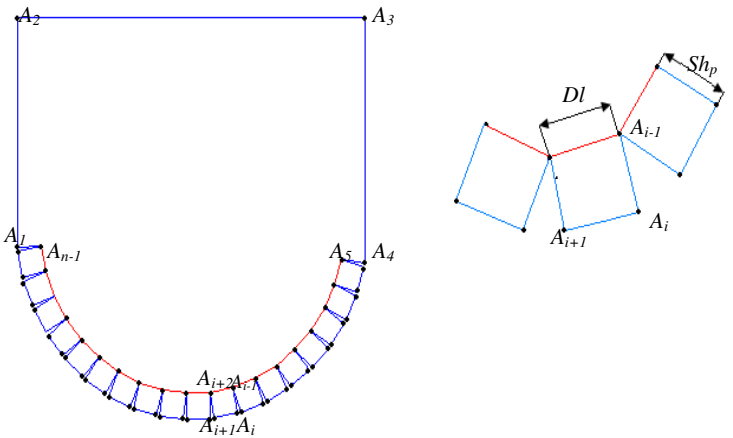


Рис. 3.2.12. **Представлення ділянки на зовнішньому контурі деталі у вигляді зубців прямокутної форми**

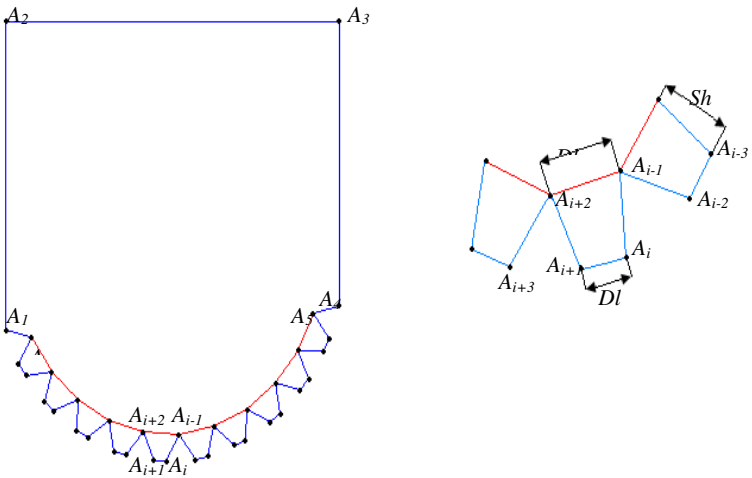


Рис. 3.2.13. **Представлення ділянки на зовнішньому контурі деталі у вигляді зубців трапецеїдальної форми**

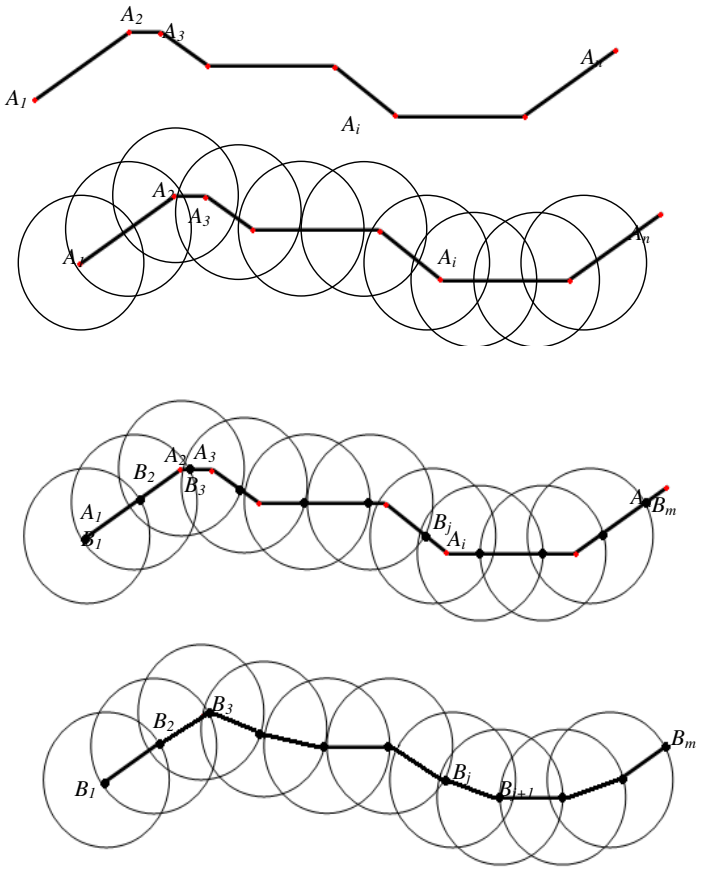


Рис. 3.2.14. **Представлення зовнішнього контуру деталі у вигляді**

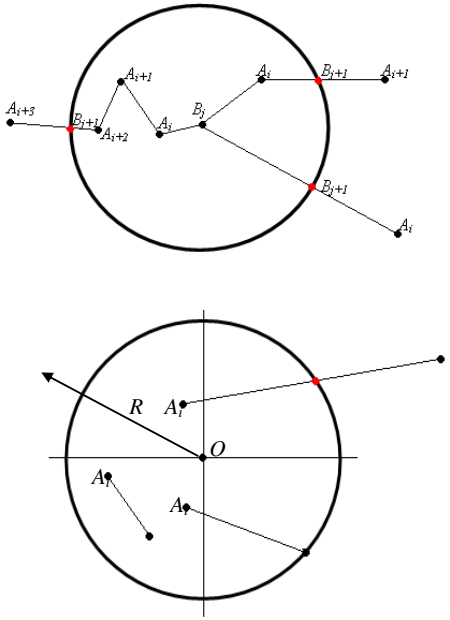


Рис. 3.2.15. **Взаємне розміщення відрізка прямої та кола**

Якщо виконується умова (3.2.20), то координати точки $B_{j+1}(Xb_{j+1},Yb_{j+1})$ можна знайти розв'язавши наступну систему рівнянь [37]:

$$\begin{cases} (X-Xb_j)^2+(Y-Yb_j)^2=R^2 \\ \dfrac{X-Xa_{i+1}}{Xa_i-Xa_{i+1}}=\dfrac{Y-Ya_{i+1}}{Ya_i-Ya_{i+1}} \end{cases} \quad \text{або} \quad \begin{cases} (X-Xb_j)^2+(Y-Yb_j)^2=R^2 \\ Ax+By+C+0 \end{cases},$$

де $A=Ya_i-Ya_{i+1}$, $B=Xa_{i+1}-Xa_i$, $C=Xa_i\cdot Ya_{i+1}-Xa_{i+1}\cdot Ya_i$.

Розглянемо два випадки:

а) $A=0$;

Тоді $y_1=y_2=-\dfrac{C}{B}$; $x_1=-\dfrac{B_1}{2}+\sqrt{\dfrac{B_1^2}{4}-C_1}$, $x_2=-\dfrac{B_1}{2}-\sqrt{\dfrac{B_1^2}{4}-C_1}$,

де $B_1=-2Xb_j$; $C_1=Xb_j{}^2+\dfrac{C^2}{B^2}+\dfrac{2C\cdot Yb_j}{B}+Yb_j-R^2$.

б) $A\neq 0$;

$$y_1 = \frac{-B_2 - \sqrt{B_2{}^2 - 4A_2 \cdot C_2}}{2A_2} \quad \text{та} \quad y_2 = \frac{-B_2 + \sqrt{B_2{}^2 - 4A_2 \cdot C_2}}{2A_2},$$

$$x_1 = \frac{-B \cdot y_1 - C}{A} \quad \text{та} \quad x_2 = \frac{-B \cdot y_2 - C}{A}$$

де

$$A_2 = 1 + \frac{B^2}{A^2},\ B_2 = \frac{2B \cdot C}{A^2} + \frac{2B \cdot Xb_j}{A} - 2Yb_j,\ C_2 = \frac{C^2}{A^2} + \frac{2C \cdot Xb_j}{A} + Xb_j + Yb_j - R^2\ .$$

Визначимо відповідно відстані від точок з координатами (x_1,y_1) та (x_2,y_2) до точки $A_{i+1}(Xa_{i+1},Ya_{i+1})$:

$$D_1 = \sqrt{(x_1 - Xa_{i+1})^2 + (y_1 - Ya_{i+1})^2}$$
$$D_2 = \sqrt{(x_2 - Xa_{i+1})^2 + (y_2 - Ya_{i+1})^2}\ .$$

Якщо $D_1 \le D_2$, то $Xb_{j+1} = x_1,\ Yb_{j+1} = y_1$, інакше $Xb_{j+1} = x_2,\ Yb_{j+1} = y_2$.

Таким чином ми маємо можливість побудувати шукану ломану лінію $B_1B_{3.2.}..B_jB_{j+1}..B_m$ (рис.). При цьому треба пам'ятати, що $B_1B_2 = B_2B_3 = .. = B_jB_{j+1} = .. = B_{m-1}B_m = R$ та $R = Dl_t$ у випадку декоративного елементу у вигляді трикутника, $R = Dl_p$ у випадку декоративного елементу у вигляді прямокутника, $R = Dl_1$ у випадку декоративного елементу у вигляді трапеції. Маючи ломану лінію $B_1B_{3.2.}..B_jB_{j+1}..B_m$ легко відтворити вибрані декоративні елементи (трикутники, прямокутники, трапеції) на ломаній лінії $B_1B_{3.2.}..B_jB_{j+1}..B_m$ (рис.3.2.14). Для цього необхідно визначити кут φ_j, де $j=1,3.2..m-1$, помістити початкову точку цього декоративного елементу в точку B_j та повернути цей декоративний елемент на кут φ_j.

Із рис. 3.2.16 очевидно, що

$$\sin\varphi_j = \frac{Yb_{j+1} - Yb_j}{\left|B_jB_{j+1}\right|} = \frac{Yb_{j+1} - Yb_j}{\sqrt{(Xb_{j+1} - Xb_j)^2 + (Yb_{j+1} - Yb_j)^2}}$$

$$\text{та } \cos\varphi_j = \frac{Xb_{j+1} - Xb_j}{\left|B_jB_{j+1}\right|} = \frac{Xb_{j+1} - Xb_j}{\sqrt{(Xb_{j+1} - Xb_j)^2 + (Yb_{j+1} - Yb_j)^2}}\ .$$

Визначимо кут φ_j, де $j=1,3.2..m-1$.

Якщо $\cos\varphi_j = 0$ та $\sin\varphi_j = 1$, то $\varphi_j = \pi/3.2$.

Якщо $\cos\varphi_j = 0$ та $\sin\varphi_j = -1$, то φ_j=-π/3.2.

Якщо $\cos\varphi_j > 0$, то $\varphi_j = arctg\dfrac{\sin\varphi_j}{\cos\varphi_j} = arctg\dfrac{Yb_{j+1} - Yb_j}{Xb_{j+1} - Xb_j}$.

Якщо $\cos\varphi_j < 0$, то $\varphi_j = arctg\dfrac{\sin\varphi_j}{\cos\varphi_j} + \pi = arctg\dfrac{Yb_{j+1} - Yb_j}{Xb_{j+1} - Xb_j} + \pi$.

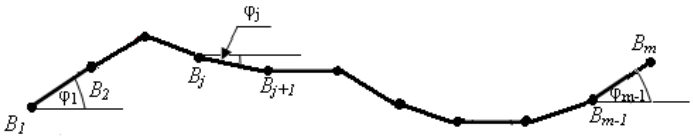


Рис. 3.2.16. **Визначення значення кута φj**

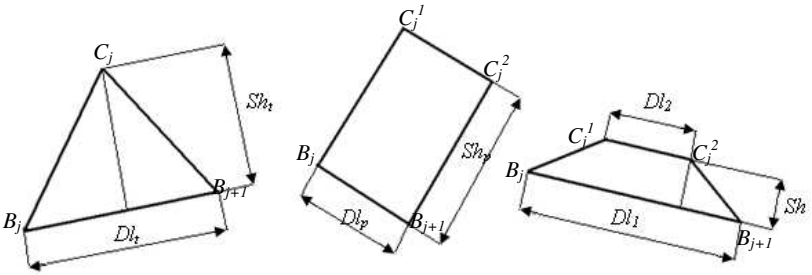


Рис. 3.2.17. **Параметри, що однозначно визначають форму зубців**

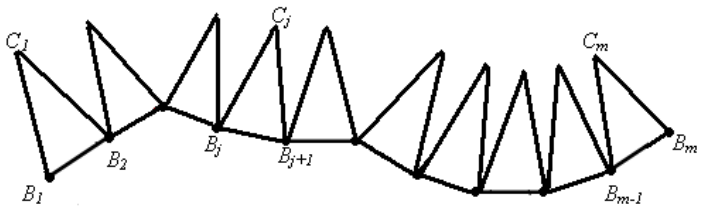


Рис. 3.2.18. **Побудова зубців трикутної форми на зовнішнього контуру деталі**

Автоматизоване проектування декоративних елементів, які представляють декоративні отвори різної форми на деталях сумок

Розглянемо два види групових декоративних елементів у вигляді фігурних отворів на деталях галантерейних виробів: циклічні групові декоративні елементи та системні групові декоративні елементи. Розглянемо більш детально кожен із цих двох видів групових декоративних елементів.

Циклічні групові декоративні елементи

Циклічні групові декоративні елементи – це групові елементи, які складаються з основного одинарного елементу та розміщених навколо нього по колу допоміжних одинарних декоративних елементів(рис.3.2.19).

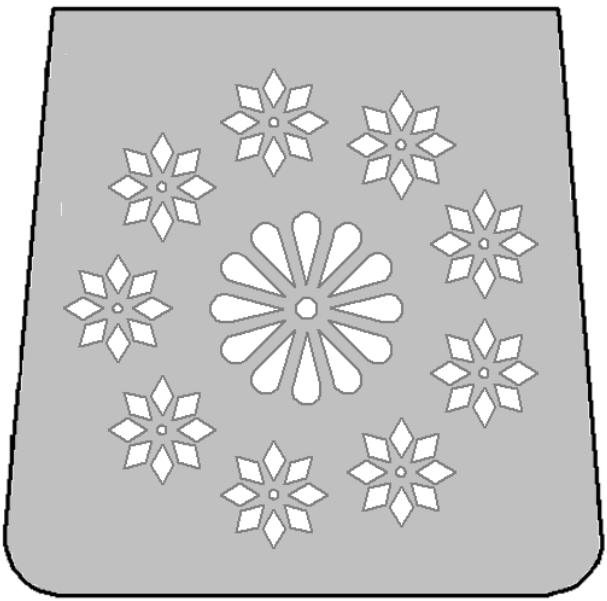


Рис. 3.2.19. **Циклічні групові декоративні елементи у вигляді фігурних отворів на деталях галантерейних виробів**

Для створення параметричної моделі циклічних групових елементів(рис.3.2.20) необхідна наступна інформація [38-40]:

- радіус кола R_1, що описане навколо циклічного групового елементу;
- вид основного групового елементу та радіус кола R_2 ($R_2 < R_1$), що описаний навколо нього;
- кількість N допоміжних групових елементів та їх вид;
- найменша відстань Δ_1 між зовнішніми контурами кіл, що описані навколо основного та допоміжного декоративних елементів;
- найменша відстань Δ_2 між зовнішніми контурами кіл, що описані навколо допоміжними декоративними елементами.

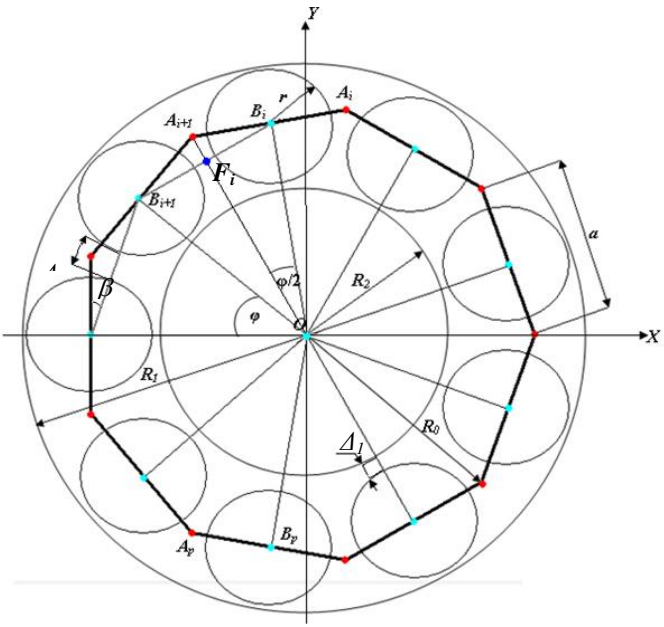


Рис. 3.2.20. **Параметри групових циклічних елементів**

Розглянемо більш детально етапи алгоритму побудови параметричної моделі групового декоративного елементу (див. рис.3.2.20).

1) Знаходимо кут F_iOF_{i+1}: $\angle F_iOF_{i+11}=\varphi=2\pi/N$.

2) Визначаємо радіус r кола, що описаний навколо допоміжного декоративного елементу: розглянемо $\Delta A_{i+1}B_iF_i$. Цей трикутник є прямокутним. Легко показати, що кут $A_{i+1}B_iF_i=\beta=\pi/N$. Тоді

$$r+\Delta_2/2=a/2\cdot\cos\pi/N=R_0\cdot\sin\pi/N\cdot\cos\pi/N=R_0/2\cdot\sin 2\pi/N.$$

Звідси $r=R_0/2\cdot\sin 2\pi/N-\Delta_2/3.2$.

3) Визначаємо радіус R_0 кола, що описаний навколо многокутника $A_0A_1..A_i..A_{N-1}$:

Нехай $H=OB_i$. Тоді $R_1=H+r=R_0\cdot\cos\pi/N+R_0/2\cdot\sin 2\pi/N-\Delta_2/3.2$. Звідси

$$R_0=\frac{R_1+\Delta_2/2}{\cos\pi/N+\sin 2\pi/N}.$$

4) Знаходимо координати вершин A_i, $i=0.1..N-1$: $\begin{matrix}X_{A_i}=R_1\cdot\cos\varphi_i\\ Y_{A_i}=R_1\cdot\sin\varphi_i\end{matrix}$, де $i=0.1..N-1$ та $\varphi_i=\varphi\cdot i$.

5) Знаходимо координати вершин B_i, $i=0.1..N-1$, тобто координати центрів допоміжних декоративних елементів:

$\begin{matrix}X_{B_i}=(X_{A_i}+X_{A_{i+1}})/2\\ Y_{B_i}=(Y_{A_i}+Y_{A_{i+1}})/2\end{matrix}$, де $i=0.1..N-3.2$.

Системні групові декоративні елементи

Системні групові декоративні елементи – це групові елементи, які складаються з однакових одинарних декоративних елементів, що розміщенні у вузлах решітки (рис.3.2.21).

Для створення параметричної моделі системних групових елементів необхідна наступна інформація (рис.3.2.22):

- радіус кола *R*, що описане навколо одного елементу із базового групового елементу;
- вид одинарного групового елементу, що належить базовому груповому елементу;
- найменша відстань Δ між зовнішніми контурами кіл, що описані навколо сусідніх одинарних елементів в базовому груповому елементу;
- інформація про зовнішній контур деталі, що представляє многокутник з координатами вершин $\{Xa_i, Ya_i\}$, $i=0,1,3.2..n-1$.

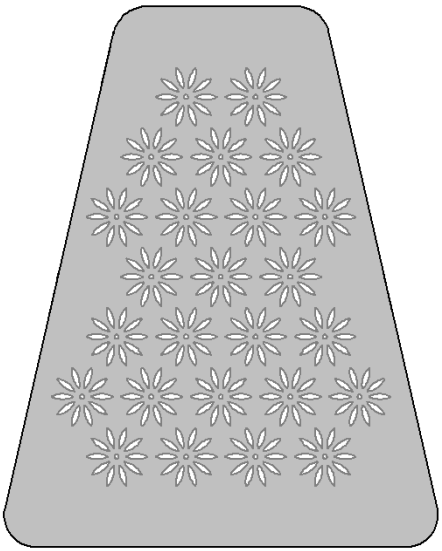


Рис. 3.2. 21. **Системні групові декоративні елементи у вигляді фігурних отворів на деталях галантерейних виробів**

Визначимо параметри, які будуть однозначно визначати положення системних групових декоративних елементів на деталях трикутної, прямокутної та трапецеїдальної форми. Для цього визначимо:

- центр прямокутника (точка *C(Xc,Yc)*), що описаний навколо деталі та зі сторонами, паралельними осям координат;
- радіус R_0;
- вектори одинарної решітки $\lambda = \{\vec{a}, \vec{b}\}$;
- зона допустимих розв’язків (прямокутник $B_0\,B_1\,B_2\,B_3$) (рис.3.2.22, рис.3.2.23.*а-б*).

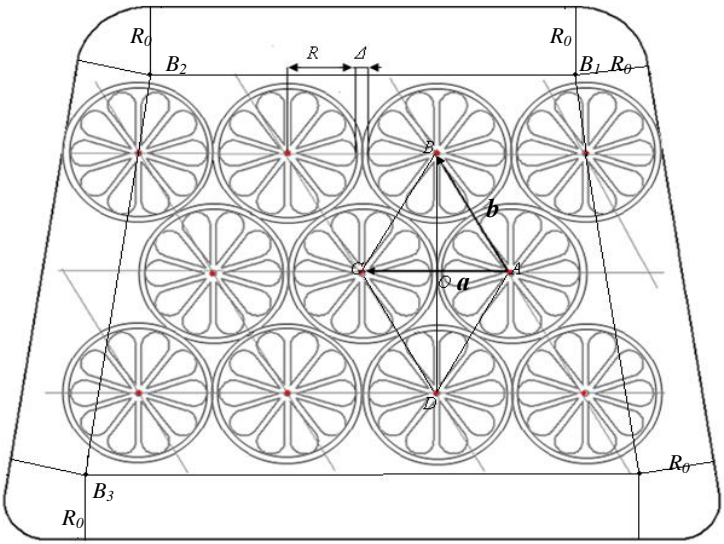


Рис. 3.2.22. **Параметри, що однозначно визначають системні групові декоративні елементи у вигляді фігурних отворів на деталях галантерейних виробів**

Із рис. 3.2.22 очевидно, що

$Xc = (\min X + \max X)/2$
$Yc = (\min Y + \max Y)/2$, де
$R_0 = R + \Delta/2$

$\min X = \min\limits_{i=0,1..n-1}\{Xc_i\}$, $\max X = \max\limits_{i=0,1..n-1}\{Xc_i\}$
$\min Y = \min\limits_{i=0,1..n-1}\{Yc_i\}$, $\max X = \max\limits_{i=0,1..n-1}\{Yc_i\}$.

Із рис.3.2.22 та рис.3.2.23.*в* очевидно, що $|AB|=|BC|=|CD|=|DA|=|AC|=2R_0$. Тоді вектор решітки ***b*** визначається наступним чином:

$$\boldsymbol{b}=\{Xb,Yb\}=\{-2R_0, 0\}.$$

Вектор решітки ***a*** визначається наступним чином:

$$\boldsymbol{a}=\{Xa,Ya\}=\{2R_0\cdot cos\ \beta, 2R_0\cdot sin\ \beta\}=\{2R_0\cdot cos\ (\pi-\alpha), 2R_0\cdot sin\ (\pi-\alpha)\}=$$

$$=\{-2R_0\cdot cos\ \alpha, 2R_0\cdot sin\ \alpha\}.$$

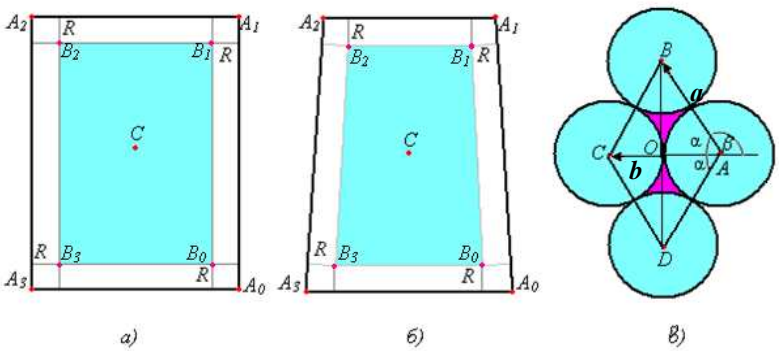


Рис. 3.2.23. **Зона допустимих розв'язків та вектори решітки для системних групових декоративних елементів**

Так як із ΔABO очевидно, що кут $\alpha=\pi/3$, то

$$\boldsymbol{a}=\{Xa,Ya\}=\{-2R_0\cdot cos\ \alpha, 2R_0\cdot sin\ \alpha\}=\left\{-R_0, \sqrt{3}R_0\right\}.$$

Визначимо зону допустимих розв'язків. Вона обмежується прямокутником $B_1B_2B_3B_0$, що побудований всередині деталі(трикутник, прямокутник та трапеція) наступним чином: до кожної із сторін всередині деталі проведемо опорну пряму на відстані R_0. Точки перетину опорних прямих і будуть вершинами прямокутника $B_1B_2B_3B_0$. Очевидно, що

рівняння опорних прямих B_1B_2 та B_3B_0 можна представити наступним чином:

$$\begin{array}{l} B_1B_2:\ Y - Y\max + R_0 = 0 \\ B_3B_0:\ Y - Y\min - R_0 = 0 \end{array}.$$

Визначимо рівняння прямих B_2B_3 та B_1B_0. Для цього визначимо рівняння прямих A_2A_3 та A_1A_0. Вважаємо, точка A_i визначається координатами $A_i(X_i, Y_i)$, $i=0,1..3$. Тоді легко показати, що рівняння прямих A_2A_3 та A_1A_0 мають наступний вигляд [25-26]:

$$\begin{array}{ll} A_2A_3: & A_1x + B_1y + C_1 = 0 \\ A_1A_0: & A_2x + B_2y + C_2 = 0 \end{array},$$

$$\text{де} \quad \begin{array}{l} A_1 = Ya_3 - Ya_2;\ B_1 = Xa_2 - Ya_3;\ C_1 = Xa_3 \cdot Ya_2 - Ya_3 \cdot Xa_1 \\ A_2 = Ya_1 - Ya_0;\ B_2 = Xa_0 - Xa_1;\ C_2 = Xa_1 \cdot Ya_0 - Ya_1 \cdot Xa_0 \end{array}.$$

Так як пряма A_2A_3 паралельна прямій B_2B_3 та пряма A_1A_0 паралельна прямій B_1B_0, то рівняння прямих B_2B_3 та B_1B_0 можна представити наступним чином [25-26]:

$$\begin{array}{ll} B_2B_3: & A_1x + B_1y + C_1^* = 0 \\ B_1B_0: & A_2x + B_2y + C_2^* = 0 \end{array},$$

$$\text{де} \quad \begin{array}{l} C_1^* = -(A_1x_2 + B_1y_2) + R_0\sqrt{A_1^2 + B_1^2} \\ C_2^* = -(A_2x_0 + B_2y_0) + R_0\sqrt{A_2^2 + B_2^2} \end{array}.$$

Полюси одинарних елементів системного декоративного елементу повинні знаходитись у вузлах решітки λ, тобто:

$$\begin{cases} Xp = Xc + i \cdot Xa + j \cdot Xb \\ Yp = Yc + i \cdot Ya + j \cdot Yb \end{cases}, \text{ де } i, j = 0, \pm 1, \pm 2.....$$

Але полюси одинарних елементів системного декоративного елементу повинні знаходитись всередині прямокутника $B_1B_2B_3B_0$, тобто задовольняти наступним умовам:

$$\begin{cases} A_1 x + B_1 y + C_1^* \le 0 \\ A_2 x + B_2 y + C_2^* \ge 0 \\ Y - Y\max + R_0 \le 0 \cdot \\ Y - Y\min - R_0 \ge 0 \end{cases}$$

Генерування одинарних декоративних елементів

Одинарні декоративні елементи, що входять до групового циклічного чи системного елементу мають форму квітки з пелюстками у вигляді: трикутника, трапеції, прямокутника, ромбу та краплі, лівої дуги кола, правої дуги кола, двох дуг кола. Приклади одинарних декоративних елементів у вигляді квітів представлені на рис. 3.2.24.

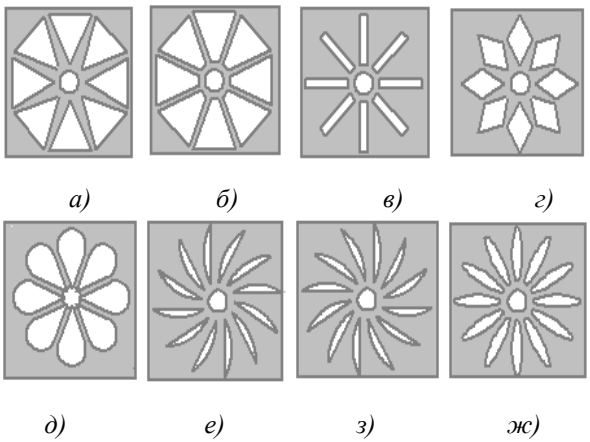
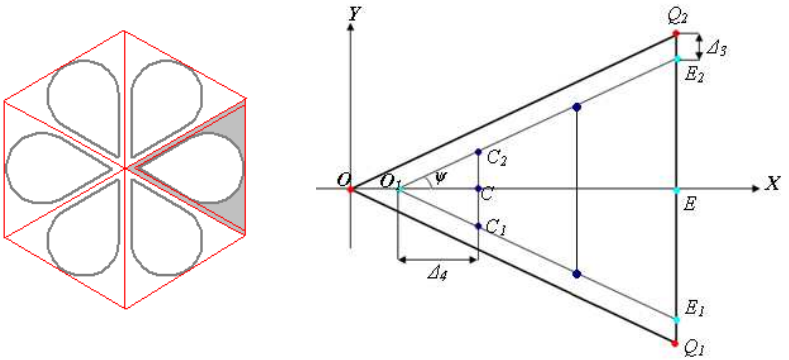


Рис. 3.2.24. **Види одинарних декоративних елементів**
а) квітка з пелюстками у вигляді трикутника; б) квітка з пелюстками у вигляді трапеції; в) квітка з пелюстками у вигляді прямокутника; г) квітка з пелюстками у вигляді ромбу; д) квітка з пелюстками у вигляді краплі; е)квітка з пелюстками у вигляді лівої дуги кола; з) квітка з пелюстками у вигляді правої дуги кола; ж)квітка з пелюстками у вигляді двох дуг кола.

Маючи координати центрів одинарних декоративних елементів та радіуси кіл, що описані навколо них, можна перейти до генерування параметричної моделі для базового декоративного елементу одного із п'яти запропонованих варіантів. Для цього навколо базового декоративного елементу опишемо правильний многокутник. Кількість сторін цього многокутника визначається кількістю пелюстків у базового декоративного елементу (рис.3.2.25.*а*). Більш детально розглянемо зафарбований фрагмент декоративного елементу (рис. 3.2.25.*б*).

Нехай $OQ_1=OQ_2=R$. Тоді кут $\psi=\pi/N$, де N - кількість пелюстків у базового декоративного елементу та $a=|Q_1Q_2|=2\cdot R\cdot \sin\psi$,

$Q_1(Xq_1,Yq_1)=(R\cdot\cos\psi,\ -R\cdot\sin\psi)$ та $Q_2(Xq_2,Yq_2)=(R\cdot\cos\psi,\ R\cdot\sin\psi)$, $E_1(Xe_1,Ye_1)=(R\cdot\cos\psi,\ -R\cdot\sin\psi+\Delta_3)$ та $E_2(Xe_2,Ye_2)=(R\cdot\cos\psi,\ R\cdot\sin\psi-\Delta 3)$, $E(Xe,Ye)=(\ (Xe_1+Xe_2)/2,\ (Ye_1+Ye_2)/2)=(\ R\cdot\cos\psi,0)$, $O_1(Xo_1,Yo_1)=(\Delta_3\cdot tg\ \psi,\ 0)$.



а) *б)*

Рис. 3.2.25. **Вузлові точки, що визначають параметри одинарного декоративного елементу**

З подібності ΔOQ_1Q_2 та $\Delta\ O_1C_1C_2$ легко визначити координати $C_1(Xc_1,Yc_1)$, $C_2(Xc_2,Yc_2)$, $C(Xc,Yc)$, $D_1(Xd_1,Yd_1)$ та $D_2(Xd_2,Yd_2)$:

$$C_1(Xc_1,Yc_1)=((2\cdot\Delta_3\cdot R\cdot\cos\psi+\Delta_4\cdot a)/a,\ -\Delta_4\cdot tg\ \psi);$$

$$C_2(Xc_2,Yc_2)=((2\cdot\Delta_3\cdot R\cdot\cos\psi+\Delta_4\cdot a)/a,\ \Delta_4\cdot tg\ \psi);$$

$C(Xc,Yc)=((Xc_1+ Xc_2)/2, (Yc_1+ Yc_2)/2)=((2\cdot\Delta_3\cdot R\cdot cos\ \psi+\Delta_4\cdot a)/a,0)$;

$D_1(Xd_1,Yd_1)=((Xc_1+ Xe_1)/2, (Yc_1+ Ye_1)/2)=$

$=(((2\cdot\Delta_3\cdot R\cdot cos\ \psi+\Delta_4\cdot a)/a+R\cdot cos\ \psi)/2,\ (-\Delta_4\cdot tg\ \psi-R\cdot sin\ \psi+\Delta_3)/2)$;

$D_2(Xd_2,Yd_2)=((Xc_2+ Xe_2)/2, (Yc_2+ Ye_2)/2)=$

$=(((2\cdot\Delta_3\cdot R\cdot cos\ \psi+\Delta_4\cdot a)/a+R\cdot cos\ \psi)/2,\ (\Delta_4\cdot tg\ \psi+R\cdot sin\ \psi-\Delta_3)/2)$

Як очевидно із рис. 3.2.26 трикутна пелюстка описується $\Delta O_1E_1E_2$, трапецеїдальна пелюстка описується чотирикутником $C_1E_1E_2C_2$, прямокутна пелюстка описується чотирикутником $C1D_1D_2C_2$, ромбовидна пелюстка описується чотирикутником $CD_1ED_{3.2.}$ Так як координати всіх цих вершин відомі, то параметричну модель однієї пелюстки представлених на рис. 3.2.26.*а-г* легко відтворити (таблиці 3.2.1-3.2.4), де $a=2\cdot R\cdot sin\ \psi$/

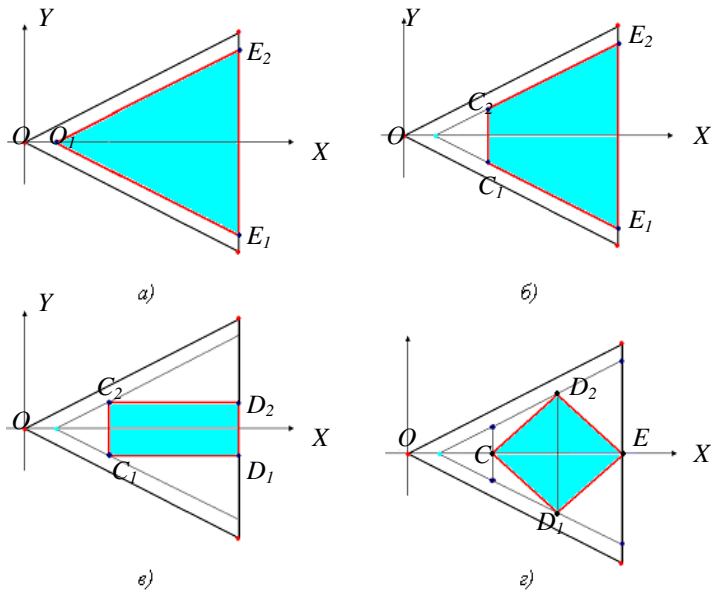


Рис. 3.2.26. **Проектування пелюстків різної форми для декоративного елементу**

Таблиця 3.2.1

Параметрична модель пелюстки трикутної форми

N n/n	*Xr*	*Yr*
1	$\Delta_3 \cdot tg\ \psi$	*0*
2	$R \cdot cos\ \psi$	$-R \cdot sin\ \psi+\Delta_3$
3	$R \cdot cos\ \psi$	$R \cdot sin\ \psi -\Delta 3$
4	$\Delta_3 \cdot tg\ \psi$	*0*

Таблиця 3.2.2

Параметрична модель пелюстки трапецеїдальної форми

N n/n	*Xr*	*Yr*
1	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$-\Delta_4 \cdot tg\ \psi$
2	$R \cdot cos\ \psi,$	$-R \cdot sin\ \psi+\Delta_3$
3	$R \cdot cos\ \psi$	$R \cdot sin\ \psi -\Delta 3$
4	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$\Delta_4 \cdot tg\ \psi$
5	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$-\Delta_4 \cdot tg\ \psi$

Таблиця 3.2.3

Параметрична модель пелюстки прямокутної форми

N n/n	*Xr*	*Yr*
1	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$-\Delta_4 \cdot tg\ \psi$
2	$R \cdot cos\ \psi,$	$-\Delta_4 \cdot tg\ \psi$
3	$R \cdot cos\ \psi$	$\Delta_4 \cdot tg\ \psi$
4	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$\Delta_4 \cdot tg\ \psi$
5	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	$-\Delta_4 \cdot tg\ \psi$

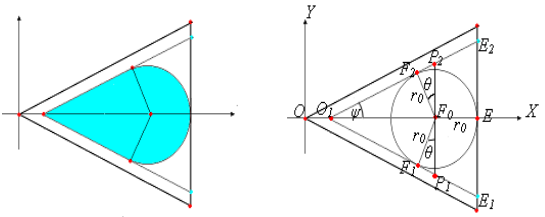
Таблиця 3.2.4

Параметрична модель пелюстки ромбовидної форми

N n/n	*Xr*	*Yr*
1	$(2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a$	*0*
2	$((2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a+R \cdot\ cos\ \psi)/2$	$(-\Delta_4 \cdot tg\ \psi-R \cdot sin\ \psi+\Delta_3)/2$
3	$R \cdot cos\ \psi$	*0*
4	$((2 \cdot \Delta_3 \cdot R \cdot cos\ \psi+\Delta_4 \cdot a)/a+R \cdot\ cos\ \psi)/2$	$(\Delta_4 \cdot tg\ \psi+R \cdot sin\ \psi-\Delta_3)/2$

5	$(2\cdot\Delta_3\cdot R\cdot\cos\psi+\Delta_4\cdot a)/a$	*0*

Окремо зупинимося на генеруванні пелюстки декоративного елементу у вигляді краплі (рис.3.2.27).



а) *б)*

Рис. 3.2.27. **Проектування пелюстків різної форми для декоративного елементу**

Для отримання параметричної моделі пелюстки декоративного елементу у вигляді краплі необхідно мати координати центра $F_0(Xf_0,Yf_0)$ дуги F_1PF_2 , радіус дуги r_0 та початковий ω_1 та кінцевий ω_2 кути дуги .

Так як $\theta=\psi=\pi/N$, то $\omega_1=-\pi/2-\psi$ та $\omega_2=\pi/2+\psi$. Очевидно із рис.3.2.27.*б*, що центр шуканої дуги має наступні координати:

$$F_0(Xf_0,Yf_0)=(Xe-r_0,0)=(R\cdot\cos\psi-r_0,\ 0).$$

Площа трикутника $E_1O_1E_2$ $S=|0_1E|\cdot|E_1E|/3.2$. Так як $|E_1E|=R$ та $|0_1E|=R\cdot\sin 2\psi$, то $S=R^2\cdot\sin 2\psi$. З другої сторони площа трикутника $E_1O_1E_2$ $S=p\cdot r_0$. Так як $p=(R+R+2R\cdot\sin\psi)/2=R(1+\sin\psi)$, то $S=R(1+\sin\psi)\cdot r_0$. Звідси

$$r_0=\frac{R\cdot\sin 2\psi}{2(1+2\sin\psi)}.$$

Так як дугу F_1PF_2 будемо представляти у вигляді ломаної лінії із заданою точністю ε, то для визначення кількості вершин Nr у апроксимуючої кривої скористаємося наступною нерівністю [41-45]:

$$Nr \geq \frac{\omega_2 - \omega_1}{2arc\cos(1 - \frac{\varepsilon}{r_0})}.$$

Тоді параметричну модель базової пелюстки у вигляді краплі можна представити наступним чином:

$$\begin{array}{c} Xr_o = \Delta_3 \cdot tq\psi; \quad Yr_0 = 0; \\ Xr_i = r_0 \cdot \cos\omega_i + Xf_0; \; Yr_i = r_0 \cdot \sin\omega_i + Yf_0; \\ Xr_{Nr+1} = \Delta_3 \cdot tq\psi; \quad Yr_{Nr+1} = 0; \end{array} \quad \text{де} \quad \begin{array}{c} i = 1,2..Nr \\ \omega_i = \omega_1 + (i-1) \cdot \frac{(\omega_2 - \omega_1)}{Nr} \end{array}$$

Для отримання параметричної моделі пелюстки декоративного елементу у вигляді лівої дуги необхідно мати координати центра $D_3(Xd_3, Yd_3)$ дуги CD_6E, радіус дуги r_L та початковий ω_{1L} та кінцевий ω_{2L} кути дуги (рис. 3.2.28). Для отримання параметричної моделі пелюстки декоративного елементу у вигляді правої дуги необхідно мати координати центра $D_4(Xd_3, Yd_3)$ дуги CD_5E, радіус дуги r_R та початковий ω_{1R} та кінцевий ω_{2R} кути дуги (рис. 3.28). Для отримання параметричної моделі пелюстки декоративного елементу у вигляді лівої та правої дуги необхідно мати координати центра $D_3(Xd_3, Y\,d_3)$ лівої дуги CD_6E, радіус лівої дуги r_L та початковий ω_{1L} та кінцевий ω_{2L} кути лівої дуги, координати центра $D_4(Xd_3, Y\,d_3)$ правої дуги CD_5E, радіус правої дуги r_R та початковий ω_{1R} та кінцевий ω_{2R} кути правої дуги (рис. 3.2.28). Введемо позначення: $a=|CE|$, $h=|DD_5|=|DD_6|$. $r_L=|CD_3|=|ED_3|=|D_6D_3|$, $r_R=|CD_4|=|ED_4|=|D_4D_5|$. Із рис. 3.2.28 очевидно, що $r_L= r_K=r_0$, $|DE|=|CD|=a/2$ та $|D_4D|=|D_3D|=b$.

Визначимо довжину відрізка CE: $|CE| = \sqrt{(Xc - Xe)^2 + (Xc - Xe)^2}$. Приймемо, що довжина відрізка D_6D_5 дорівнює *0.1-0.2* довжина відрізка CE.

Із ΔDED_3 за теоремою Піфагора маємо, що $|D_3E|^2=|D_3D|^2+|DE|^2$, або $r_0^2 = b^2 + a^2/4.$

Так як $r_0=b+h$, то отримаємо $(b+h)^2=b^2+a^2/3.2$. Звідси отримаємо:

$$b = \frac{a^2 - 4h}{8h}, \quad r_0 = b + h = \frac{a^2 - 4h}{8h} + h, \quad \varphi = arctg(\frac{4h \cdot a}{a^2 - 4h^2}).$$

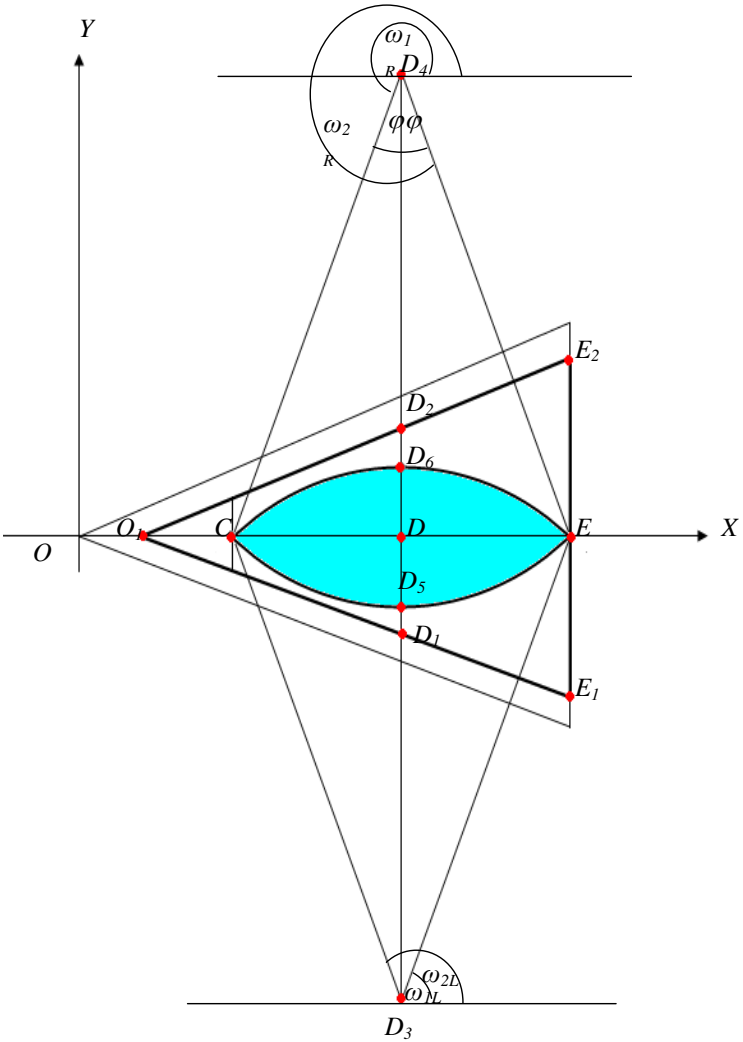


Рис. 3.2.28. **Побудова пелюстки у вигляді дуг для одинарного декоративного елементу**

Визначимо кути ω_{1L}, ω_{2L}, ω_{1R} та ω_{2R} (рис. 3.2.28):

$$\omega_{1L} = \pi/2 - \varphi; \quad \omega_{2L} = \omega_{1L} + 2\varphi;$$
$$\omega_{1R} = 3\pi/2 - \varphi; \quad \omega_{2R} = \omega_{1R} + 2\varphi.$$

Тоді параметричну модель пелюстки декоративного елементу у вигляді лівої дуги можна представити наступним чином:

$$\begin{aligned} Xr_i &= r_0 \cdot \cos(\omega_{1L} + \frac{2\varphi}{n} i) + Xd_3 \\ Yr_i &= r_0 \cdot \sin(\omega_{1L} + \frac{2\varphi}{n} i) + Yd_3 \\ Xr_{n+1} &= Xr_0 \qquad Yr_{n+1} = Yr_0 \end{aligned}, \text{ де } i = 0,1...n.$$

Параметричну модель пелюстки декоративного елементу у вигляді правої дуги можна представити наступним чином:

$$\begin{aligned} Xr_i &= r_0 \cdot \cos(\omega_{1R} + \frac{2\varphi}{n} i) + Xd_4 \\ Yr_i &= r_0 \cdot \sin(\omega_{1R} + \frac{2\varphi}{n} i) + Yd_4 \\ Xr_{n+1} &= Xr_0 \qquad Yr_{n+1} = Yr_0 \end{aligned}, \text{ де } i = 0,1...n.$$

Параметричну модель пелюстки декоративного елементу у вигляді лівої та правої дуги можна представити наступним чином:

$$\begin{aligned} Xr_i &= r_0 \cdot \cos(\omega_{1L} + \frac{2\varphi}{n} i) + Xd_3 \\ Yr_i &= r_0 \cdot \sin(\omega_{1L} + \frac{2\varphi}{n} i) + Yd_3 \\ Xr_{n+i+1} &= r_0 \cdot \cos(\omega_{1R} + \frac{2\varphi}{n} i) + Xd_4 \\ Yr_{n+i+1} &= r_0 \cdot \sin(\omega_{1R} + \frac{2\varphi}{n} i) + Yd_4 \end{aligned}, \text{ де } i = 0,1...n.$$

Маючи параметричну модель базової пелюстки легко отримати параметричну модель j –ої пелюстки одинарного декоративного елементу:

$$\begin{aligned} X_{ji} &= Xr_i \cdot \cos\psi_j - Yr_i \cdot \sin\psi_j \\ Y_{ji} &= Xr_i \cdot \sin\psi_j - Yr_i \cdot \cos\psi_j \end{aligned}, \text{ де } \begin{matrix} i = 0,1..Nr \\ j = 1,2..N \\ \psi_j = 2 \cdot \psi \cdot (j-1) \end{matrix}$$

3.3. ПІДГОТОВКА ІНФОРМАЦІЇ ПРО ЗОВНІШНІ КОНТУРИ ДЕТАЛЕЙ

Оскільки деталі виробів легкої промисловості не адекватні будь-яким геометричним образам, зовнішній контур їх являє собою замкнуту лінію складної конфігурації. Це ускладнює, а в більшості випадків, унеможливлює аналітичний опис зовнішньої границі деталі. Тому їх доводиться апроксимувати, тобто заміняти більш простими кривими, які можна описати аналітично.

Відомо, що інформацію про лінію можна задати аналітично, у вигляді математичних рівнянь, таблицею і графічно. Графічна інформація буде найбільш точною, але вона не придатна для введення в ЕОМ, які не можуть зберігати в своїй пам’яті у натуральному вигляді образи.

Способи апроксимації можна розділити на наступні основні групи:

1. Аналітичний опис зовнішнього контуру деталі:

   а) за допомогою многочленів;

   б) за допомогою інших елементарних математичних функцій.

2. Дискретний опис зовнішнього контуру:

   а) покриття деталі більш простими геометричними фігурами;

   б)апроксимація зовнішнього контуру деталі сукупністю елементарних ліній.

Апроксимацію за допомогою многочленів можна розділити на два види: апроксимація за допомогою сплайнів і за допомогою поліномів. Для апроксимації за допомогою сплайнів пропонується застосовувати кубічні сплайни та сплайни інших порядків.

Заслуговують уваги способи апроксимації за допомогою інших функцій. Серед них можна виділити спосіб апроксимації за допомогою R-функцій.

Проаналізувавши способи аналітичного опису зовнішнього контуру можна відмітити, що алгоритми опису контурів деталей для розглянутих способів дуже громіздкі, потребують значних затрат ручної праці, складних математичних обчислень для задовільної апроксимації деталі, не мають достатньої швидкодії. Крім того, при такій апроксимації ускладнюється розв’язок задач автоматизації технологічної підготовки виробництва, що призводить до значних затрат часу на підготовку висхідної інформації і розв’язок поставлених задач.

Тому зупинимося на способах дискретного опису плоских геометричних об’єктів, тобто способах, при яких інформація про плоский геометричний об’єкт задається у вигляді масивів чисел.

Вирішальне значення при виборі способу апроксимації має точність відтворення форми і розмірів вихідної деталі. При будь-якому способі апроксимації можна одержати необхідну точність, збільшуючи число застосовуваних геометричних елементів, тобто збільшуючи масив чисел. Однак, при збільшенні вихідної інформації збільшується машинний час розв’язку задачі. Всім цим вимогам найбільш відповідає кусково-лінінійна апроксимація зовнішнього контуру плоского геометричного об’єкту.

Кусково-лінійна апроксимація являє собою заміну дійсного контуру деталі відрізками прямих, причому відстань від відрізків до точок контуру не перевищує похибки апроксимації. Найбільше розповсюдження отримав кусково-лінійний спосіб апроксимації[47]. При цьому способі апроксимації зовнішній контур деталі апроксимується багатокутником. Довжина сторін багатокутника залежить від кривизни контуру і вибраної погрішності апроксимації. Сторони апроксимуючого багатокутника можуть бути дотичними до контуру деталі, січними, хордами. Проведені дослідження показали доцільність використання за сторони апроксимуючого багатокутника хорди до зовнішнього контуру деталі. Тому в подальшому

будемо використовувати кусково-лінійну апроксимацію, при якій сторонами апроксимуючого багатокутника є хорди (рис. 3.3.1)

Кусково-лінійна апроксимація зводиться до побудови багатокутника, вершини якого лежать на контурі деталі.

Кусково-лінійна апроксимація використовується, в основному, в двох випадках: при автоматичному зчитуванні графічної інформації і перетворенні її в цифрову і при використанні систем програмного керування для креслення і виготовлення деталей і шаблонів, коли пристрій керування може задавати рух виконавчим органам тільки по відрізках прямих.

Нехай маємо заданий контур деталі. При перетворенні графічної інформації в цифрову будуть отримані координати множини точок $A_i(X_i, Y_i)$, $i = 1, 2, ... n$. Таку заміну контуру множиною точок називають кусково-лінійною апроксимацією. Така апроксимація контуру має істотний недолік, який полягає в необхідності запам'ятовування невиправдано великого числа координат точок A_i. Тому одночасно зі зчитуванням графічної інформації вирішується задача зменшення кількості (ущільнення) інформації. З цією метою розраховують координати нової множини точок A_j, причому відстань від точок A_i до відрізків A_jA_{j+1} $(j = 1, 2, ..., m)$ буде меншою заданої точності ε. Ця задача розв’язується за допомогою ПК.

Кусково-лінійний спосіб апроксимації є універсальним, тобто придатним до будь-якої форми плоских геометричних об’єктів, не потребує великих затрат часу при ручному способі апроксимації, легко автоматизується.

Крім того інформація при кусково-лінійному способі апроксимації піддається ущільненню, тобто відсів зайвих точок без втрат точності апроксимації.

Тоді кусково-лінійний спосіб апроксимації зовнішнього контуру деталі є самим зручним та простим при автоматизованій та ручній підготовці інформації, не накладає обмежень на геометрію деталей.

Інтерактивна підготовка інформації про зовнішні контури

Контури деталей виробів легкої промисловості в більшості випадків мають складну конфігурацію, тому їх апроксимують більш простими кривими. Найбільше розповсюдження одержала кусково-лінійна апроксимація, тобто апроксимація кривими першого порядку (прямими). Це зручно також для автоматизованого вводу такої інформації при використанні спеціальних пристроїв.

Нехай S - фігура з заданою орієнтацією. Зв'яжемо з деталлю S

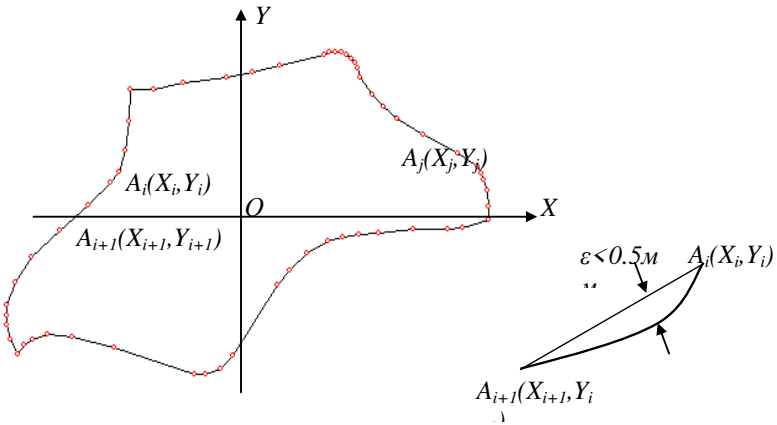


Рис. 3.3.1. **Кусково-лінійна апроксимація**

координатну систему XOY, де O – полюс деталі, обраний в будь-якій її внутрішній точці. Контур взуттєвої деталі S апроксимуємо ламаною лінією з вершинами в послідовно вибраних на контурі деталі точках (рис. 3.3.1.).

Будь-яка деталь S може бути представлена координатами вершин

опукло-ввігнутого многокутника, тобто масивом $\{X_j, Y_j\}$,. $j=1..n$, а координати будь-якої точки на прямій між вершинами A_j та A_{j+1} можна визначити наступним чином:

$$\begin{aligned} X &= X_j + (X_{j+1} - X_j)t \\ Y &= Y_j + (Y_{j+1} - Y_j)t \\ 0 &\le t \le 1; j = 1,2..n \end{aligned} \qquad (3.3.1)$$

де j– порядковий номер відрізка між вершинами A_j та A_{j+1} ;

n– кількість точок апроксимації для деталі S;

t– параметр, що визначає положення точки на прямій між вершинами A_j та A_{j+1}.

Постановка задачі. Нехай маємо файл формату **.bmp* із кресленнями деталей виробів легкої промисловості. Необхідно отримати інформацію про зовнішні контури цих деталей у вигляді координат апроксимуючого многокутника; інформацію про деталі представити у файлі формату **.dgt* системи проектування моделей взуття «Ірис»[48].

В будь-якому файлі формату **.bmp* в перших двох (0-1) байтах зберігаються символи 4D42h, які є буквами 'BM'; в наступних чотирьох (2-5) байтах зберігається розмір файлу в байтах; наступні чотири(6-9) байти – резервні поля; наступні чотири(10-13) байти – зсув, з якого починається саме зображення (растр); наступні чотири (14-17) байти – розмір заголовка BITMAP у байтах (дорівнює 40); наступні чотири (18-21) байти - ширина зображення в пікселах; наступні чотири (22-25) байти - висота зображення в пікселах; наступні два (26-27) байти - кількість площин, повинно бути 1; наступні два (28-29) байти – кількість біт на відображення одного піксела: 1, 4, 8 або 24; наступні чотири (30-33) байти - тип стиску; наступні чотири (34-37) байти - 0 або розмір стиснутого зображення в байтах; наступні чотири (38-41) байти - горизонтальна роздільна здатність, *піксел/м*; наступні чотири (42-47) байти - вертикальна роздільна здатність, *піксел/м*;

наступні чотири (46-49) байти - кількість використовуваних кольорів; наступні чотири(50-53) байти - кількість "важливих" кольорів. Далі йде інформація про піксели самого зображення по рядкам.

Необхідно зауважити, що якщо інформація записана в декількох байтах, то перший байт є наймолодшим. Для виводу креслення на екран монітору нас будуть цікавити 18-21 байти - ширина зображення в пікселах та 22-25 байти - висота зображення в пікселах. Для визначення реальних координат вершин апроксимуючих многокутників для зовнішніх контурів деталей взуття, креслення яких представлені у файлі формату **.bmp*, нас будуть цікавити 38-41 байти - горизонтальна роздільна здатність в *піксел/мм* та 42-47 байти - вертикальна роздільна здатність в *піксел/мм*.

Рис. 3.3.2. **Структурна схема алгоритму інтерактивної підготовки інформації про зовнішні контури деталей взуття**

Нехай нам потрібно визначити ширину зображення в пікселах *ShIm*, що зберігається в 18-21 байтах, та значення 18, 19, 20, 21 байтів, що відповідно позначимо: *A,B,C,D*.

Тоді *ShIm* можна визначити за наступною формулою:

$$ShIm = A + 256 * B + 256^2 * C + 256^3 * D \qquad (3.3.2)$$

Аналогічно визначається висота зображення, горизонтальна та вертикальна роздільна здатність в *піксел/м*.

Нехай координати вершин апроксимуючого многокутника для зовнішнього контуру деталі в пікселах представлено масивом $\{X_j^E, Y_j^E\}$, *j=1..n*. Тоді реальні координати вершин в *мм* можна розрахувати за наступними формулами:

$$\begin{matrix} X_j = mx \cdot X_j^E \\ Y_j = my \cdot Y_j^E \end{matrix}, \quad \text{де} \quad \begin{matrix} mx = 1000/RSh \\ my = 1000/RVs \end{matrix}, \quad j = 1,2,...n \qquad (3.3.3),$$

де *RSh*- горизонтальна роздільна здатність в *піксел/м*,

RVs - вертикальна роздільна здатність в *піксел/м*.

На основі формул (3.3.1-3.3.3) розроблений алгоритм інтерактивної підготовки інформації про зовнішні контури деталей взуття[49-50]. Структурна схема алгоритму представлена на рис. 3.3.2.

Автоматична підготовка інформації про зовнішні контури деталей

При побудові креслень виникає задача отримати інформацію про зовнішні контури деталей із заданою точністю. Ця задача зводиться до класу задач по розпізнаванню образів. Відомі методи розпізнавання образів є досить складними, і не отримують результати у зручному представленні для систем автоматизованого проектування (САПР) легкої промисловості. Існуючі алгоритми автоматизованої підготовки вихідної інформації, розроблені для легкої промисловості, відрізняються великою

складністю підготовки і переробки інформації та містять інші недоліки, і не завжди представляють результати із визначеною точністю. Звідси очевидна актуальність автоматичної підготовки вихідної інформації про зовнішні контури виробів легкої промисловості для побудови раціональних схем розкрою.

За допомогою сканера можна отримати зображення креслення будь-якого плоского об'єкту із заданою точністю (роздільною здатністю, або кількістю пікселів на метр) .

Будь-який плоский геометричний об'єкт можна представити координатами точок вершин апроксимуючого опукло-ввігнутого многокутника, тобто масивом $L_i(x_i, y_i)$, $i=1,...,n$, де x_i, y_i – координати i-ої вершини, а n – кількість вершин апроксимуючого многокутника.

Тоді задача підготовки раціональної схеми розкрою може бути сформульована наступним чином: із растрового зображення креслення деталі отримати векторне зображення зовнішньої границі цієї деталі, тобто із растрового зображення креслення деталі отримати координати вершин апроксимуючого многокутника $L_i(x_i, y_i)$ для зовнішнього контуру деталі. У цій задачі можна виділити такі етапи [51]:

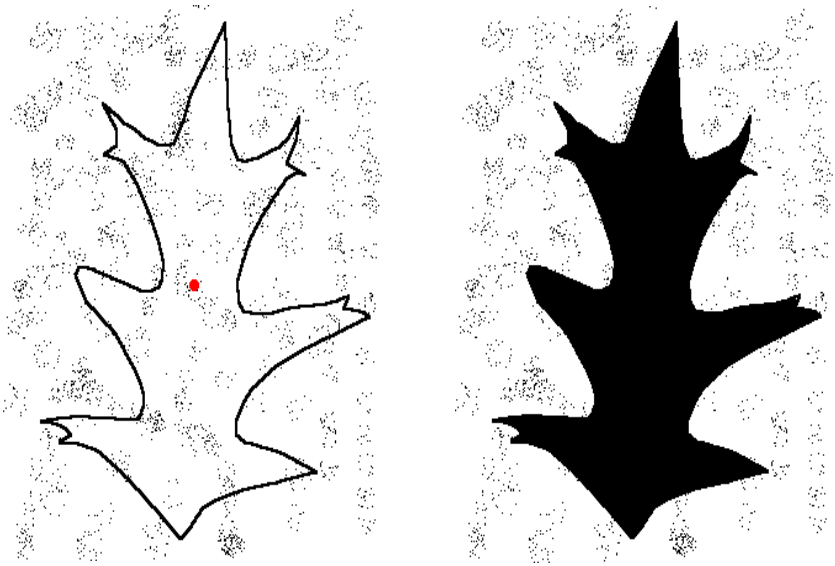
- ввід інформації зі сканера та визначення точності введення зображення креслення деталі ;
- визначення точки всередині деталі та зафарбування деталі одним кольором;
- позбавлення від небажаної випадкової інформації, тобто від шумів;
- визначення першої точки що належить зовнішньому контуру деталі;
- визначення наступних точок на зовнішньому контуру деталі;
- ущільнення інформації про зовнішній контур деталі.

Перші два етапи розв'язку задачі тривіальні і ми на них не зупиняємось. Більш детально розглянемо останні етапи (рис.4.3). Будемо вважати, що після перших двох етапів маємо растрове монохромне

зафарбоване зображення (за допомогою пікселей) креслення деталі із небажаною випадковою інформацію, тобто із шумами (рис.4.3).

Для позбавлення від небажаної випадкової інформації, тобто від шумів запропонований наступний алгоритм:

Масив точок які складають растрове зображення вихідної деталі позначимо як *A [w,h],* де *w –* кількість точок у стовбці масиву, *h* - кількість точок у рядку, *a(i,j)* – відповідно елемент цього масиву Складемо новий масив даних *C[w,h]* того ж розміру, що і зображення креслення деталі із небажаною випадковою інформацію. Значення елементів цього масиву, а саме кольору точки *c(i,j), i=1,...,w , j=1,...,h* визначаються наступним чином:



а) *б)*

Рис. 3.3.3. Креслення деталі із випадковою інформацією

а)Креслення деталі після сканування та вибору точки б) Креслення деталі після зафарбування

Якщо у множині точок $\lambda_{ij} = \bigcup_{p=i-1}^{i+1} \bigcup_{q=j-1}^{j+1} a_{pq} \setminus a_{ij}$ більше точок білого кольору, то точка c_{ij} приймається теж білого кольору, інакше – чорного. Тобто значення кольору точки c_{ij} визначається з наступного виразу[52]:

$$c_{ij} = \text{int}\left(\frac{\sum_{p=i-1}^{i+1} \sum_{q=j-1}^{j+1} a_{pq} - a_{ij}}{8}\right) \qquad (3.3.4)$$

Результати роботи описаного алгоритму представлені на рис. 4.4 .

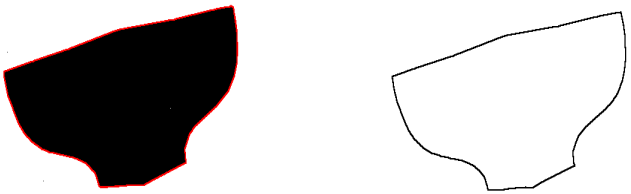
Позначимо ланцюг точок, що визначають координати контуру, як *L*. Точку, що належить цьому ланцюгу, як $L_f(x_f, y_f)$, *f=0,…,m-1,* де *m* – кількість точок у ланцюзі.

а) *б)* *в)*

Рис. 3.3.4. Етапи фільтрації шумів на кресленні сканованої деталі
а) після зафарбування; б) після першого проходу через фільтр; в) після декількох проходів через фільтр

Для пошуку першої точки $L_0(x_0, y_0)$ проскануємо зображення по рядках до знаходження першої точки чорного кольору. Решта точок визначається наступним чином: у множині точок λ_{ij}, де *i=p-1,p,p+1; j=q-1,q,q+1 (i=1,…,m-2 ,j=1,…,m-2)* визначається точка *P(x,y)*, в околі якої найбільша кількість білих точок(це є гарантією, що точка *P(x,y)* належить контуру деталі). Обрана точка заноситься до ланцюгу *L, L=L∪P(x,y).* В

результаті обходу контуру за приведеним вище алгоритмом отримаємо зовнішній контур деталі в пікселах.



а) *б)*

Рис. 3.3.5. Представлення зовнішнього контуру деталі
а) за допомогою пікселів б) за допомогою опукло-ввігнутого многокутника

На рис. 3.3.5.а представлений зовнішній контур деталі, який містить 958 пікселів. Очевидно, що можна ущільнити інформацію про зовнішній контур.

Для розв'язання цієї задачі сформуємо новий ланцюг пікселів $L^{|}$, для формування якого виконаємо наступні дії.

Позначимо точність, із якою буде проводитися апроксимація контуру деталі, як ε.

Проведемо через точки $L_i(X_i,Y_i)$ та $L_k(X_k,Y_k)$, що належать ланцюгу L пряму L_iL_k. Рівняння прямої прийме вигляд [53]:

$$\frac{x-X_i}{X_k-X_i}=\frac{y-Y_i}{Y_k-Y_i} \qquad (4.5)$$

або $Ax+By+C=0$,де $A=Y_k-Y_i; B=X_i-X_k; C=X_kY_i-X_iY_k$. (4.6)

(Початкове значення *i* приймається рівним нулю, *k* – двом : *i=0, k=2.*

Опустимо перпендикуляри із точок L_j, де *j=i+1,...,k-1* на пряму L_iL_k.

Позначимо довжини відрізків L_jB_j (рис. 3.3.6), як $\delta=|f_j|$. Тоді $\delta=|f_j|$ знаходяться за формулою[54]:

$$\delta=|f_j|=\frac{AX_j+BY_j+C}{\sqrt{A^2+B^2}}. \qquad (4.7)$$

На кожному кроці *k* збільшується на одиницю.

Якщо хоча б одне значення $|f_j| \geq \varepsilon$, то точка L_{k-1} заноситься до ланцюгу $L', L' = L' \cup L_{k-1}$ та *i* приймається рівним *k-1 (i=k-1).* Ці дії виконуємо до тих пір, поки *k* не стане рівним *m-1* (поки контур не буде пройдений повністю).

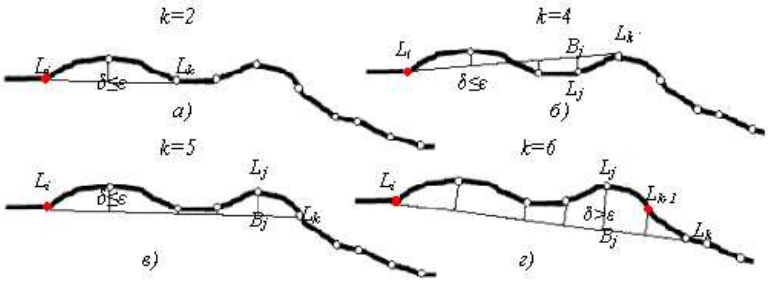


Рис. 3.3.6. **Апроксимація контуру деталі із заданою точністю *ε***

Контур деталі при вилученні цих вершин буде відображений із заданою точністю. Результат ущільнення інформації за допомогою представленого алгоритму відображений на рис. 3.3.5.*б*. При ущільненні інформації (тобто видаленні зайвих вершин) замість 958 пікселів на зовнішньому контурі деталі (рис.3.3.5.*а*) залишається 33 вершини опукло-ввігнутого многокутника. На рис. 34.7 наведені приклади того, як впливає точність апроксимації *ε* на форму зовнішнього контуру деталі.

Очевидно, що зовнішні контури при точності апроксимації *ε=0.25мм* (рис.3.3.7.*д*) та *ε=0.1мм* (рис 3.3.7.*е*) візуально не відрізняються, але при точності апроксимації *ε=0.25мм* кількість вершин апроксимуючого многокутника *N=29*, а точності апроксимації *ε=0.1 мм* - кількість вершин апроксимуючого многокутника *N=372*.

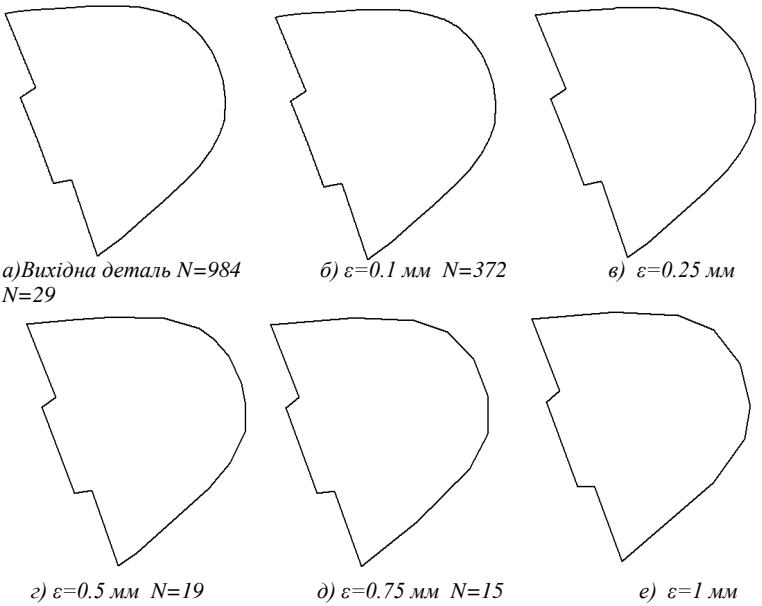


Рис. 3.3.7. Вплив точності апроксимації ε на форму зовнішнього контуру деталі

3.4. КОРИГУВАННЯ ЗОВНІШНІХ КОНТУРІВ ДЕТАЛЕЙ

При автоматизованому проектуванні часто необхідно мати інструмент для коригування зовнішніх контурів деталей. Контур деталі представляє собою многокутник з вершинами $A_j(x_j,y_j)$, $j=1,2...n$. Тоді нам необхідно для коригування зовнішнього контуру деталі, що представляє многокутник, наступні функції:

- виведення креслення деталі в область із заданими розмірами;
- вилучення будь-якої вершини на контурі деталі;
- зміна положення будь-якої вершини на контурі деталі;
- введення додаткової вершини на контурі деталі;
- згладжування за допомогою сплайну заданої ділянки на контурі деталі;
- спряження кутів дугою заданого радіусу.

Виведення креслення деталі в прямокутну область із заданими розмірами

Для візуалізації деталі та коригування її зовнішнього контуру необхідно вивести креслення деталі на прямокутну область заданих розмірів із максимальним масштабом(рис. 3.4.1)

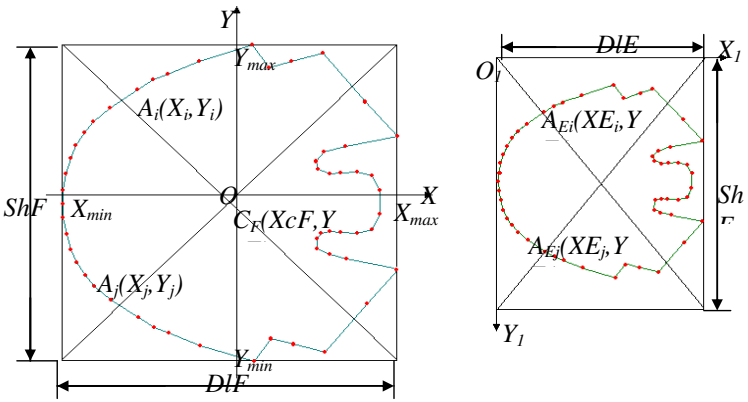


Рис.3.4.1. **Визначення масштабного множника *mxv***

Алгоритм розв’язку цієї задачі можна представити наступним чином [55]:

- визначення прямокутника, описаного навколо деталі, в якого лівий верхній кут має координати (X_{min},Y_{max}), правий нижній кут має координати (X_{max},Y_{min}), де

$$\begin{cases} X_{\min} = \min\limits_{i=1,2..n}\{X_i\} \\ X_{\max} = \max\limits_{i=1,2..n}\{X_i\} \\ Y_{\min} = \min\limits_{i=1,2..n}\{Y_i\} \\ Y_{\max} \max\limits_{i=1,2..n}\{X_i\} \end{cases} \qquad (3.4.1)$$

- визначення координат центра (*XcF,YcF*) та довжини *DlF* і ширини *ShF* прямокутника, описаного навколо деталі(рис.3.4.2):

$$XcF = (X_{\max} + X_{\min})/2 \quad YcF = (Y_{\max} + Y_{\min})/2$$
$$DlF = X_{\max} - X_{\min}; \quad ShF = Y_{\max} - Y_{\min}.$$

(3.4.2)

- визначення максимально допустимого масштабу *mxy* для виведення креслення в прямокутну область із довжиною *DlE* і шириною *ShE:*

$mxy = \min\{mx, my\}$, де $mx = DlE / DlF$ та $my = ShE / ShF$;

- визначення координат вершин креслення деталі (XE_i,YE_i), $i=1,2\ldots n$:

$$\begin{cases} XE_i = (X_i - XcF)\cdot mxy + XcE \\ YE_i = (YcF - Y_i)\cdot mxy + YcE \end{cases} \text{. де } \begin{matrix} XcE = DlE/2 \\ YcE = ShE/2 \end{matrix} \text{ та } i = 1,2,...n. \qquad (3.4.3)$$

- вивід креслення деталі у вигляді многокутника із координатами вершин(XE_i,YE_i), $i=1,2\ldots n$.

Алгоритм вилучення будь-якої вершини на контурі деталі

Нехай контур деталі представляє собою многокутник з вершинами $A_j(X_j,Y_j)$, $j=1..n$. Курсор знаходиться в точці $O(X_0,Y_0)$. Будемо вважати, що нам необхідно вилучити ту вершину, яка найменше віддалена від точки $O(X_0,Y_0)$ (рис.3.4.2).

Для вилучення необхідної вершини потрібно[55-56]:

- ідентифікувати необхідну вершину. Для цього потрібно виконати наступні дії:

а) знайти відстані $D_j = \sqrt{(Xo - X_j)^2 + (Yo - Y_j)^2}$ від точки $O(X_0,Y_0)$ до вершин $A_j(X_j,Y_j)$, $j=1..n$,

$XoF=XcF+(Xo-XcE)/mxy$ та $YoF=YcF-(Yo-YcE)/mxy$;

b) визначити вершину A_i, для якої $D_i = \min\{D_j\}, j = 1..n$;

- вилучити цю вершину із файлу з інформацією про деталі;

- вилучити відрізки A_iA_{i-1} та A_jA_{j+1} на екрані;

- провести новий відрізок $A_{i+1}A_{i-1}$.

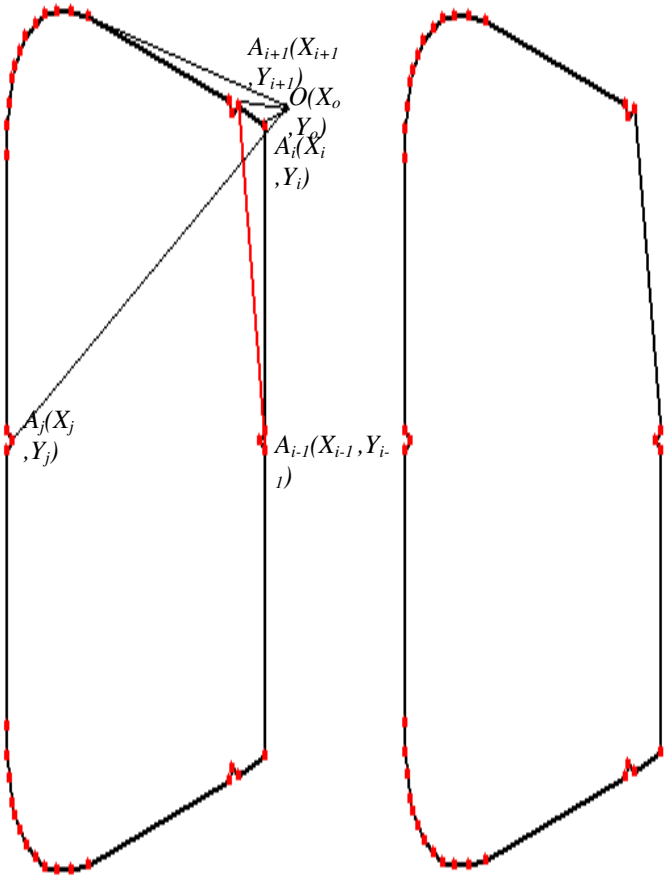


Рис. 3.4.2. **Вилучення вершини на контурі деталі**
а) деталь до вилучення вершини б) деталь після

Алгоритм зміни положення будь-якої вершини на контурі деталі

Нехай контур деталі представляє собою многокутник з вершинами $A_j(X_j,Y_j)$, $j=1..n$. Курсор знаходиться в точці $O(X_0,Y_0)$. Будемо вважати, що

нам необхідно вершину A_i, яка найменше віддалена від точки $O(X_0,Y_0)$(рис.3.4.3), перемістити в точку $A_i^{/}$.

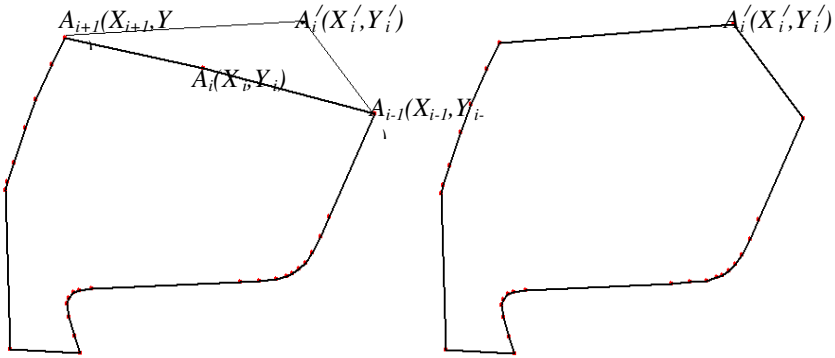


Рис. 3.4.3. **Зміна положення вершини на контурі деталі**
а) деталь до зміни положення вершини на контурі деталі б) деталь після зміни положення вершини на контурі деталі

Для зміни значення координат будь-якої вершини на контурі деталі потрібно[55, 57]:

- ідентифікувати необхідну вершину. Для цього потрібно виконати наступні дії:

  а) знайти відстані $D_j = \sqrt{(XoF - X_j)^2 + (YoF - Y_j)^2}$ від точки $O(X_0,Y_0)$ до вершин $A_j(X_j,Y_j)$, $j=1..n$,

  $XoF=XcF+(Xo-XcE)/mxy$ та $YoF=YcF-(Yo-YcE)/mxy$;

  b) визначити вершину A_i, для якої $D_i = \min\{D_j\}, j = 1..n$;

- знайти нове значення координат цієї вершини, тобто координати точки $A_i^{/}(X_i^{/},Y_i^{/})$:

$$\begin{aligned} X_i^{/} &= XcF + (Xo^{/} - XcE)/mxy \\ Y_i^{/} &= YcF - (Yo^{/} - YcE)/mxy \end{aligned},$$

де $O^{/}(X_0^{/},Y_0^{/})$- нове положення курсору (координати точки $A_i^{/}$ в системі координат, яка пов'язана з монітором;

- замінити у файлі з інформацією про деталі старі координати цієї вершини $A_i\,(X_i\,,Y_i\,)$ на нові $A_i^{/}(X_i^{/},Y_i^{/})$;

3.4. вилучити відрізки A_iA_{i-1} та A_jA_{j+1} на екрані;

5. провести нові відрізки $A_{i-1}A_i^{/}$ та $A_{i+1}A_i^{/}$.

Алгоритм введення додаткової вершини на контурі деталі

Нехай контур деталі представляє собою многокутник з вершинами $A_j(x_j,y_j)$, $j=1..n$. Курсор знаходиться в точці $O(X_0,Y_0)$. Будемо вважати, що нам необхідно між вершинами A_i та A_{i+1} ввести нову вершину $A^*(x^*,y^*)$, якщо точка $O(X_0,Y_0)$ (рис.3.4.4) найменше віддалена від сторони A_iA_{i+1} та проекція точки $O(X_0,Y_0)$ на пряму, яка проходить через точки A_i та A_{i+1}, буде належати цій стороні.

Для введення додаткової вершини на контурі деталі потрібно[55, 57]:

- ідентифікувати необхідну сторону $A_i\ A_{i+1}$ на зовнішньому контуру деталі.

- Для цього потрібно виконати наступні дії:

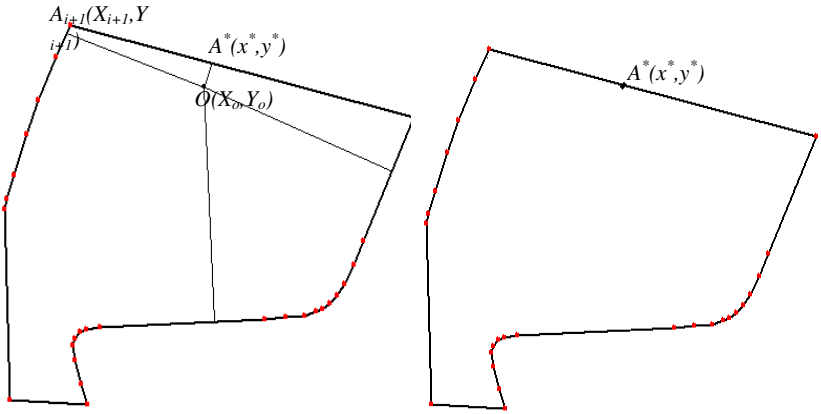
а) знайти відстані D_j точки $O(X_0,Y_0)$ до сторін A_jA_{j+1}, $j=1..n-1$,

де $D_j = A_j^*YoF + B_j^*XoF + C_j^*$, $XoF=XcF+(Xo-XcE)/mxy$ та

$YoF=YcF-(Yo-YcE)/mxy$; $A_j^* = (Y_{j+1} - Y_j)/Q_j$, $B_j^* = (Y_j - Y_{j+1})/Q_j$,

$C_j^* = ((X_{j+1} - X_j)Y_j - (Y_{j+1} - Y_j)Y_j)/Q_j$ та $Q_j = \sqrt{(X_{j+1} - X_j)^2 + (X_{j+1} - Y_j)^2}$.

При пошуку D_j ми розглядаємо тільки ті сторони $A_j\ A_{j+1}$, для яких проекція точки $O(X_0,Y_0)$ на пряму, що проходить через точки A_j та A_{j+1} буде належати даному відрізку.



а) *б)*

Рис. 3.4.4. **Введення додаткової вершини на контурі деталі**
а*) деталь до введення додаткової вершини*
б) деталь після введення додаткової вершини

Більш детально зупинимося на знаходженні відстані D від точки C до відрізка AB, якщо перпендикуляр, проведений із точки C на відрізок AB перетинає його. Нехай точки A, B, C мають координати: $A(X_a,Y_a)$, $B(X_b,Y_b)$, $C(X_c,Y_c)$. Тоді рівняння прямої, що проходить через точки A, B, має вигляд:

$$\frac{x-X_a}{X_b-X_a}=\frac{y-Y_a}{Y_b-Y_a}, \text{ або } Ax+By+C=0, \qquad (3.4.4)$$

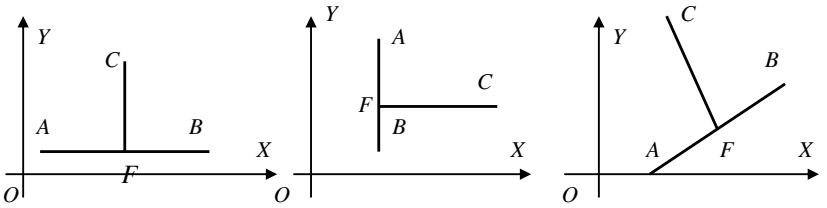
де $A=Y_b-Y_a;\ B=X_a-X_b;\ C=X_bY_a-X_aY_b.$

Відстань від точки C до прямої, що проходить через точки A, B визначається наступним чином [53]:

$$D=\frac{AX_c+BY_c+C}{\sqrt{A^2+B^2}} \qquad (3.4.5)$$

Знайдемо точку перетину перпендикуляра $F(X_f,Y_f)$, проведеного із точки C до прямої, що проходить через точки A, B з цією прямою. Можливі три випадки:

$A=0$ (пряма, що проходить через точки A, B, паралельна вісі OX (рис. 3.4.3.4.*а*). Тоді $X_f=X_c$, $Y_f=Y_a$.



а) *б)* *в)*

Рис. 3.4.5

- $B=0$ (пряма, що проходить через точки A, B, паралельна вісі OY (рис. 3.4.5.*б*).

Тоді $X_f=X_a$, $Y_f=Y_c$

- $A \neq 0$ та $B \neq 0$ (рис. 3.4.5.*в*). Тоді точку перетину перпендикуляра $F(X_f,Y_f)$, проведеного із точки C до прямої, що проходить через точки A, B з цією прямою знаходимо із наступної системи рівнянь:

$$\begin{cases} Ax + By + C = 0 \\ y - Y_c = K(x - X_c) \end{cases}, \quad \text{де} \quad K = -\frac{X_b - X_a}{Y_a - Y_b} = \frac{X_a - X_b}{Y_a - Y_b} \qquad (3.4.6)$$

Розв'язавши систему рівнянь (3.4.6) знайдемо координати точки перетину F:

$$X_f = \frac{K \cdot B \cdot X_c - B \cdot Y_c - C}{A + K \cdot B},$$

$$Y_f = Y_c + K \cdot (X_f - X_c)$$

.(3.4.7)

Тепер нам залишилося впевнитись, що точка перетину F лежить всередині відрізка AB. Нехай $T_1 = \min(X_a, X_b)$, $T_2 = \max(X_a, X_b)$, $Q_1 = \min(Y_a, Y_b)$ та $Q_2 = \max(Y_a, Y_b)$.

Тоді, якщо виконується наступна умова:

$$\begin{cases} T_1 \le X_f \le T_2 \\ Q_1 \le Y_f \le Q_2 \end{cases}, \qquad (3.4.8)$$

то точка перетину F лежить всередині відрізка AB.

Описану вище процедуру проробимо для всіх сторін зовнішнього контуру деталі. Серед тих сторін, для яких точка перетину перпендикуляра $F(X_f,Y_f)$, проведеного із точки C лежить на стороні, будемо шукати сторону, що найменше віддалена від точки C.

Згладжування будь-якої ділянки контуру деталі за допомогою параметричного сплайну

Контури деталей виробів легкої промисловості у виключних випадках можуть бути описані аналітично. Тому їх будемо апроксимувати більш простими кривими. Найбільше поширення отримала кусково-лінійна апроксимація, тобто апроксимація кривими першого порядку (прямими). Тоді будь-який плоский геометричний об'єкт можна представити координатами точок вершин апроксимуючого опукло-ввігнутого багатокутника, тобто масивом $\{X_i,Y_i\}$, $i=1\ldots n$, де X_i,Y_i – координати і-ої вершини, а n – кількість вершин апроксимуючого багатокутника. Але апроксимація окремих ділянок контуру не завжди задовольняє потребам. Для одержання нової ділянки контуру застосуємо параметричний сплайн.

Для апроксимації виберемо параметричний B-сплайн [57], оскільки він забезпечує одержання більш плавних кривих, ніж при інших способах згладжування, та інтерполяційний параметричний сплайни. Криві, описані за допомогою B-сплайну, є неперервними та мають також

неперервні перші і другі похідні, а завдяки використанню інтерполяційного параметричного сплайну вони також неперервні та мають неперервну першу похідну, але крім того вони проходять через вузлові точки. Необхідно відмітити, що апроксимація за допомогою параметричного сплайну не накладає обмежень на геометрію деталі (рис.3.4.6).

Нехай будь-яка точка на кривій між двома послідовними опорними точками P_i та P_{i+1} має координати[57]:

$$x(t)=((A_{i3}t+A_{i2})t+A_{i1})t+A_{i0} \quad (3.4.9)$$

$$y(t)=((B_{i3}t+B_{i2})t+B_{i1})t+B_{i0}, \quad (3.4.10)$$

де t збільшується від 0 до 1.

Ці рівняння для параметричного B-сплайну мають такі коефіцієнти[57]:

$$A_{i3}=(-X_{i-1}+3X_i-3X_{i+1}+X_{i+2})/6; \qquad B_{i3}=(-Y_{i-1}+3Y_i-3Y_{i+1}+Y_{i+2})/6;$$

$$A_{i2}=(X_{i-1}-2X_i+X_{i+1})/2; \qquad B_{i2}=(Y_{i-1}-2Y_i+Y_{i+1})/2;$$

$$A_{i1}=(-X_{i-1}+X_{i+1})/2; \qquad B_{i1}=(-Y_{i-1}+Y_{i+1})/2; \quad (3.4.11)$$

$$A_{i0}=(X_{i-1}+4X_i+X_{i+1})/6; \qquad B_{i0}=(Y_{i-1}+4Y_i+Y_{i+1})/6.$$

Властивості отриманої кривої

- проходить біля опорних точок(рис.3.4.6.а);
- має неперервні першу і другу похідні;
- змінюючи координати окремих точок можна здійснювати локальне регулювання форми кривої;
- в точках стику попереднього і наступного сегментів кривих параметричного B сплайну співпадають значення функцій, якими описуються ці сегменти кривих, перші та другі похідні цих функцій.

Ці рівняння для параметричного інтерполяційного сплайну мають такі коефіцієнти [59]:

$$A_{i3}=(-X_{i-1}+3X_i-3X_{i+1}+X_{i+2})/2; \qquad B_{i3}=(-Y_{i-1}+3Y_i-3Y_{i+1}+Y_{i+2})/2;$$
$$A_{i2}=(2X_{i-1}-5X_i+4X_{i+1}-X_{i+2})/2; \qquad B_{i2}=(2Y_{i-1}-5Y_i+4Y_{i+1}-X_{i+2})/2;$$
$$A_{i1}=(-X_{i-1}+X_{i+1})/2; \qquad B_{i1}=(-Y_{i-1}+Y_{i+1})/2; \qquad (3.4.12)$$
$$A_{i0}=X_i; \qquad B_{i0}=Y_i.$$

Властивості отриманої кривої

- проходить точно через опорні точки(рис.3.4.6.*б*);
- має неперервну похідну;
- змінюючи координати окремих точок можна здійснювати локальне регулювання форми кривої;
- в точках стику попереднього і наступного сегментів кривих параметричного інтерполяційного сплайну, співпадають значення функцій, якими описуються ці сегменти кривих, та перші похідні цих функцій.

При побудові отриманої сплайнової кривої для параметричних інтерполяційного та *B* сплайнів кожний сегмент розраховується на дільниці між парою внутрішніх точок чергової четвірки з набору базових точок. Тому для побудови складеної 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.

Для того, щоб провести згладжування відповідної ділянки зовнішнього контуру деталі потрібно [60-63]:

- вказати крайні точки ділянки зовнішнього контуру деталі, які необхідно

згладити;

- продублювати ці точки;
- ідентифікувати цю ділянку зовнішнього контуру деталі;
- вибрати метод згладжування:
- за допомогою параметричного В-сплайну
- за допомогою параметричного інтерполяційного сплайну;
- перенумерувати вершини апроксимуючого багатокутника таким чином, щоб ділянка контуру, що необхідно згладити, знаходилась в кінці;
- провести згладжування;
- замінити ділянку контуру, що необхідно згладити, новим контуром.

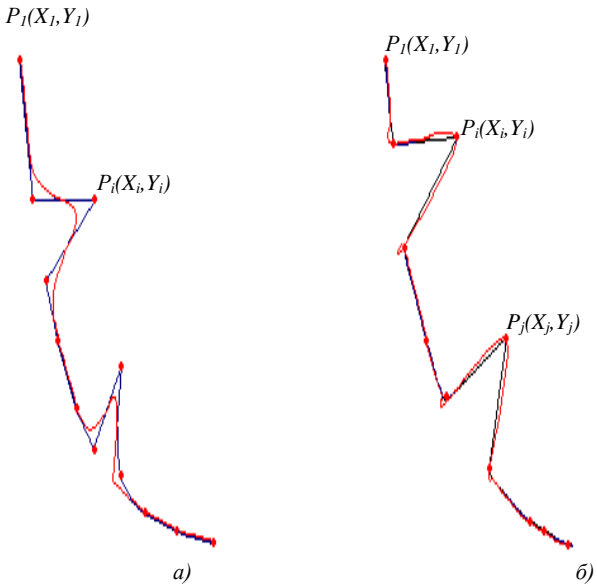


Рис. 3.4.6. **Згладжування ділянки контуру деталі**
а) за допомогою параметричного В сплайну б) за допомогою параметричного інтерполяційного сплайну

Спряження кута дугою кола

Серед завдань моделювання найбільшу складність представляє спряження кута, утвореного двома суміжними сторонами деталі, дугою кола, що передбачає визначення параметрів дуги, та апроксимація дуги ломаною лінією із заданою точністю (визначення кількості та координат апроксимуючої ломаної лінії) [63].

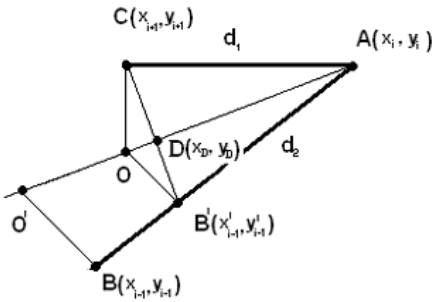


Рис. 3.4.7. **Спряження кута дугою кола**

Спряження кута можна виконати двома способами:

- задається відстань від вершини до точки дотику дуги на стороні деталі;
- за відстань від вершини до точки дотику приймається довжина меншої зі сторін кута.

Для виконання спряження кута необхідно задати відстань від вершини до точки дотику дуги зі стороною. Нехай за відстань від вершини до точки дотику дуги зі стороною приймається довжина меншої з сторін (другий випадок).

Знаючи координати трьох точок, що утворять кут *ABC,* визначаємо довжину сторін *AC* і *AB* (рис.3.4.7) і знаходимо мінімальну з них. Тоді знайдемо точку *B′* таким чином, що *AB′ = AC:*

З подоби трикутників *ABO′* і *AB′O* випливає:

$$\begin{cases} \dfrac{x_{i-1}-x_i}{AB}=\dfrac{x'_{i-1}-x_i}{AB'}, \\ \dfrac{y_{i-1}-y_i}{AB}=\dfrac{y'_{i-1}-y_i}{AB'} \end{cases}, \quad \text{звідки випливає:} \quad \begin{cases} x'_{i-1}-x_i=\dfrac{AB'}{AB}(x_{i-1}-x_i), \\ y'_{i-1}-y_i=\dfrac{AB'}{AB}(y_{i-1}-y_i) \end{cases}$$

Координати точки B' рівні:

$$\begin{cases} x'_{i-1}=\dfrac{AB'}{AB}(x_{i-1}-x_i)+x_i, \\ y'_{i-1}=\dfrac{AB'}{AB}(y_{i-1}-y_i)+y_i \end{cases}. \qquad (3.4.13)$$

Якщо задана відстань d від вершини до точки дотику дуги зі стороною, то знаходимо точки $C'(x'_{i+1},y'_{i+1})$ і $B'(x'_{i-1},y'_{i-1})$:

$$\begin{cases} x'_{i-1}=\dfrac{d_0}{d_2}(x_{i-1}-x_i)+x_i, \\ y'_{i-1}=\dfrac{d_0}{d_2}(y_{i-1}-y_i)+y_i, \end{cases}, \qquad (3.4.14)$$

$$\begin{cases} x'_{i+1}=\dfrac{d_0}{d_1}(x_{i+1}-x_i)+x_i, \\ y'_{i+1}=\dfrac{d_0}{d_1}(y_{i+1}-y_i)+y_i \end{cases}. ,$$

де $d_1=AB$, $d_2=AC$.

Тоді координати точки D: $\begin{cases} x_D=\dfrac{x_{i+1}+x_{i-1}}{2} \\ y_D=\dfrac{y_{i+1}+y_{i-1}}{2} \end{cases}$ (3.4.15)

З подоби трикутників AOC і OCD випливає, що $\dfrac{AC}{AO}=\dfrac{CD}{CO}$, тобто радіус дуги дорівнює $R=CO=\dfrac{AO\cdot CD}{AC}$.

Знайдемо довжину відрізків AO і AD: $AD=\sqrt{(x_D-x_i)^2+(y_D-y_i)^2}$.

З подоби трикутників AOC і ACD випливає, що

$$\frac{AC}{AD}=\frac{AO}{AC}. \qquad (3.4.16)$$

З виразу (3.4.16) одержимо $AO=\dfrac{AC^2}{AD}$

Тепер знайдемо координати точки *O*. Оскільки вектор $\overline{AO} = k * \overline{AD}$, то

$$\begin{cases} x_O - x_A = \dfrac{AO}{AD}(x_D - x_A), \\ y_O - y_A = \dfrac{AO}{AD}(y_D - y_A) \end{cases},$$

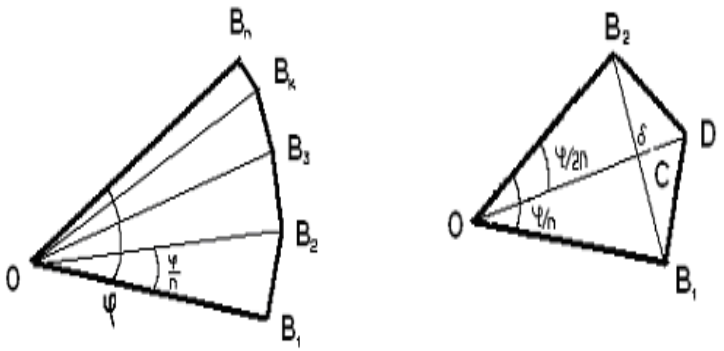
звідки отримаємо координати точки *O*:

$$\begin{cases} x_O = x_A + \dfrac{AO}{AD}(x_D - x_A), \\ y_O = y_A + \dfrac{AO}{AD}(y_D - y_A) \end{cases} \qquad (3.4.17)$$

Дугу будь-якого радіуса можна представити із заданою точністю як ломану лінію, що представляє собою *N* відрізків однакової довжини (рис.3.4.8.*а*). Тоді знайдемо кількість точок в дузі, яка спряжує кут .

Знайдемо допустиму похибку при апроксимації *δ*. Допустима похибка при апроксимації *δ* повинна бути меншою або дорівнювати заданій користувачем точності (рис. 3.4.8.*б*), тобто

$$\delta <= eps \qquad (3.4.18).$$



а) *б)*

Рис. 3.4.8. **Визначення кількості вершин апроксимуючого многокутника**

$$\begin{cases} OC = OB_1 * \cos\dfrac{\varphi}{2N} \\ OD = OB_1 = OB_2 = R \\ \delta = OD - OC = R - OB_1 * \cos\dfrac{\varphi}{2N} = R - R * \cos\dfrac{\varphi}{2N} \end{cases} \quad (3.4.19)$$

З виразів (3.4.18) та (3.4.19) випливає що,

$$eps \geq R - R * \cos\left(\frac{\varphi}{2N}\right) \quad (3.4.20)$$

Перетворивши вираз (3.4.20) отримаємо вираз для знаходження кількості вершин ломаної лінії, що апроксимує шукану дугу:

$$N \geq \frac{\varphi}{2\arccos\left(\dfrac{R - eps}{R}\right)} \quad (3.4.21)$$

Кут між прямими AB та AC визначимо за теоремою косинусів: $a^2 = b^2 + c^2 - 2ab * \cos\varphi$.

Застосувавши цю теорему до нашого випадку, отримаємо такий вираз для знаходження кута заокруглення

$$\varphi = \arccos\left(\frac{CB^2 - CO^2 - OB^2}{-2 * CO * OB}\right) \quad (3.4.22)$$

Знаходження координат точок в спряженій дузі. Для того, щоб знайти координати точок в дузі заокруглення, використаємо лінійне перетворення точки з вихідними координатами (в нашому випадку перетворення точки B_1 для отримання координат точок $B_2, B_3, \ldots, B_N$ (рис.3.4.8.*а*).

Щоб отримати координати точок $B_2, B_3, \ldots, B_N$, необхідно повернути точку B_1 навколо точки O на кут $\frac{K\varphi}{N}$.

Необхідні для повороту формули отримаємо в результаті композиції матриць паралельного переносу, повороту навколо початку координат і оберненої до матриці паралельного переносу:

$$R^{\alpha}_{(x,y)} = T_{(x,y)} \circ R^{\alpha}_{(0,0)} \circ T^{-1}_{(x,y)}. \quad (3.4.23)$$

Тоді отримаємо формулу для знаходження координат *k*-ї вершини деталі (*n* – кількість точок в дузі заокруглення, φ – кут між векторами OB_1 і OB_N):

$$\begin{cases} x_K = \cos\left(\frac{k\varphi}{n}\right) - \sin\left(\frac{k\varphi}{n}\right) - x_O * \cos\left(\frac{k\varphi}{n}\right) + y_O * \cos\left(\frac{k\varphi}{n}\right) + x_O \\ y_K = \sin\left(\frac{k\varphi}{n}\right) + \cos\left(\frac{k\varphi}{n}\right) - x_O * \sin\left(\frac{k\varphi}{n}\right) - y_O * \cos\left(\frac{k\varphi}{n}\right) + y_O \end{cases} \quad (3.4.24)$$

В алгоритмі спряження кута дугою заданого радіуса можна виділити наступні пункти:

1.Ідентифікація кута, який необхідно спряжити дугою заданого радіуса, тобто визначити необхідну вершину на апроксимуючому многокутнику для зовнішнього контуру деталі. Нехай контур деталі представляє собою многокутник з вершинами $A_j(x_j,y_j)$, $j=1..n$. Курсор знаходиться в точці $O(X_0E,Y_0E)$. Будемо вважати, що шуканою вершиною буде та вершина, яка найменше віддалена від точки $O(X_0E,Y_0E)$. Для ідентифікації її потрібно виконати наступні дії:

а) знайти відстані $D_j = \sqrt{(XoF - X_j)^2 + (YoF - Y_j)^2}$ від точки $O(X_0E,Y_0E)$ до вершин $A_j(X_j,Y_j)$, $j=1..n$,

$XoF=XcF+(XoE-XcE)/mxy$ та $YoF=YcF-(YoE-YcE)/mxy$;

b) визначити вершину A_i, для якої $D_i = \min\{D_j\}, j = 1..n$;

2. Ввести відстань d_0 , на яку буде віддалена точка дотику спряжуваної дуги до сторін вибраного кута ($d_0 \le d_1$ та $d_0 \le d_2$).

3. За формулами (3.4.17-3.4.24) обчислюємо параметри спряженої дуги.

3.4. У файлі з інформацією про деталі замінити координати ідентифікованої вершини на координати вершин ломаної лінії, що апроксимує спряжену дугу.

5. Вилучити відрізки $A_iA'_{i-1}$ та $A_jA'_{j+1}$ на екрані.

6. Побудувати ломану лінію, що апроксимує спряжену дугу.

Визначення коефіцієнта точного відображення ζ за точністю ущільнення інформації про зовнішні контури деталей ε

Зовнішній контур кожної із спроектованих деталей задається координатами вершин апроксимуючого опукло-ввігнутого многокутника, тобто масивом $\{X_m, Y_m\}$,. *m=1..n*. Часто кількість вершин буває зайвою, щоб із заданою точністю передати конфігурацію контуру деталі. А зайві вершини приводять до збільшення часу розрахунку параметрів розкрійних схем. Тому актуальна задача відсіву зайвих вершин на зовнішньому контурі апроксимуючого многокутника.

Нехай маємо деталь *Sd* та плоский геометричний об'єкт *Sr,* що апроксимує нашу деталь (рис.3.4.9.б). Під коефіцієнтом точного відображення ζ будемо розуміти $\zeta=1-|Sp|/|Sd|$, де $|Sd|$ – площа деталі *Sd* (рис.3.4.9.а), $|Sp|$ – площа плоского геометричного об'єкта (зафарбованої області на рис.7.1.в), що визначається наступним чином:

$$Sp = (Sd \setminus Sr) \cup (Sr \setminus Sd), \qquad (3.4.25)$$

тобто площа зафарбованої області на рис. 3.4.9.в.

Будь-яка деталь *Sd* може бути представлена координатами вершин апроксимуючого опукло-ввігнутого многокутника, тобто масивом $\{X_m, Y_m\}$,. *m=1..n*. Тоді площу $|Sd|$ деталі *Sd* можна визначити наступним чином:

$$|Sd| = \left| \sum_{m=1}^{n} X_m Y_{m+1} - X_{m+1} Y_m \right| / 2 \qquad (3.4.26)$$

а $|Sp|$ - площа плоского геометричного об'єкта (зафарбованої області на рис. 3.4.9.в визначити наступним чином:

$$|Sp| = \sum_{t=1}^{q} Sp_t , \qquad (3.4.27)$$

де Sp_t – площа t-ої елементарної ділянки об'єкта Sp та q- кількість елементарних ділянок, із яких складається об'єкт Sp.

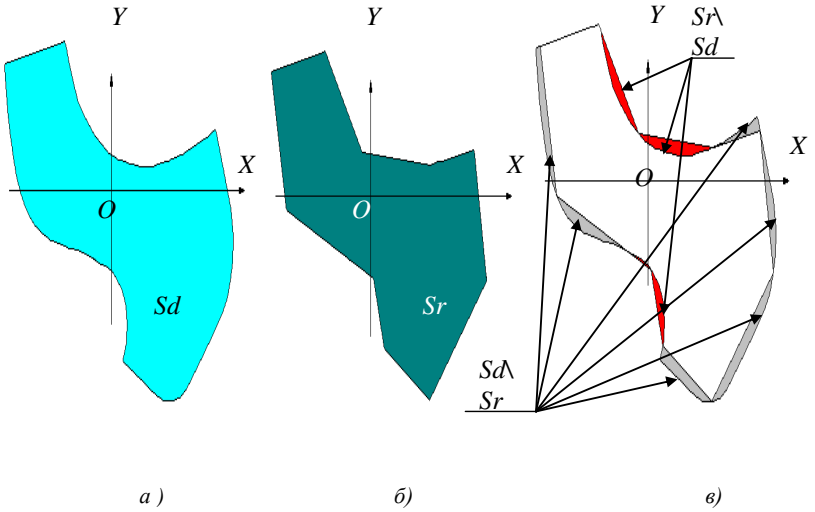


Рис. 3.4.9. **Визначення коефіцієнту точного відображення**

Розглянемо t-у елементарну ділянку об'єкта Sp (рис. 3.4.10), що знаходиться між двома вершинами $A_i(X_i,Y_i)$ та $A_j(X_j,Y_j)$. На цій елементарній ділянці кожна точка на контурі цієї ділянки знаходиться на відстані , що менша ε-точності ущільнення, від опорної прямої A_iA_j, якою буде замінена дуга $A_iA_kBA_{k+1}A_j$ при ущільнені інформації. Нам необхідно знайти всі базові точки на опорної прямої A_iA_j, тобто точки перетину дуги $A_iA_kBA_{k+1}A_j$ та відрізка опорної прямої A_iA_j (див. рис. 3.4.10).

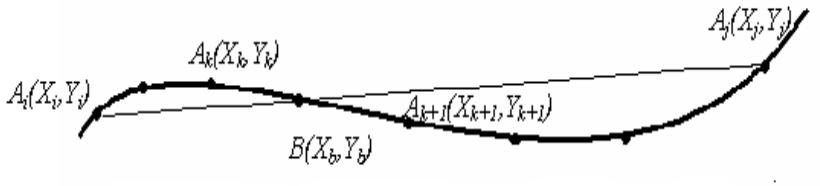


Рис. 3.4.10. Визначення базових точок на елементарній ділянці плоского геометричного об'єкту *Sr*

Рівняння прямої A_iA_j має вигляд:

$$A_1 x + B_1 y + C_1 = 0,$$

де $A_1 = Y_j - Y_i;\quad B_1 = X_i - X_j;\quad C_1 = X_j Y_i - X_i Y_i$

Рівняння прямої A_kA_{k+1} має вигляд:

$$A_2 x + B_2 y + C_2 = 0,$$

де $A_2 = Y_{k+1} - Y_k;\quad B_2 = X_k - X_{k+1};\quad C_2 = X_{k+1} Y_k - X_k Y_{k+1}$

Тоді базові точки ми знайдемо, розв'язавши наступну систему:

$$\begin{cases} A_1 x + B_1 y + C_1 = 0 \\ A_2 x + B_2 y + C_2 = 0 \\ (A_1 \cdot X_k + B_1 \cdot Y_k + C_1)(A_1 \cdot X_{k+1} + B_1 \cdot Y_{k+1} + C_1) \le 0, \\ (A_2 \cdot X_i + B_2 \cdot Y_i + C_2)(A_2 \cdot X_j + B_2 \cdot Y_j + C_2) \le 0 \end{cases} \qquad (3.4.28)$$

де $k = i+1, i+2 \ldots j-2$.

Тобто, якщо виконуються умови:

$$\begin{cases}(A_1 \cdot X_k + B_1 \cdot Y_k + C_1)(A_1 \cdot X_{k+1} + B_1 \cdot Y_{k+1} + C_1) \le 0 \\ (A_2 \cdot X_i + B_2 \cdot Y_i + C_2)(A_2 \cdot X_j + B_2 \cdot Y_j + C_2) \le 0\end{cases} \quad (3.4.29)$$

для $k = i+1, i+2 \ldots j-2$, то координати базових точок визначаються наступними виразами:

$$\begin{cases}X = \Delta_x / \Delta \\ Y = \Delta_y / \Delta\end{cases}, \qquad \text{де } \Delta = \begin{vmatrix}A_1 & B_1 \\ A_2 & B_2\end{vmatrix} = A_1 B_2 - A_2 B_1; \quad (3.4.30)$$

$$\Delta_x = \begin{vmatrix}-C_1 & B_1 \\ -C_2 & B_2\end{vmatrix} = -C_1 B_2 + C_2 B_1; \quad \Delta_y = \begin{vmatrix}A_1 & -C_1 \\ A_2 & -C_2\end{vmatrix} = -A_1 C_2 + A_2 C_1; \quad (3.4.31)$$

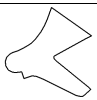
Знайшовши координати опорних точок, використавши вирази (7.3-7.6), легко знайти площу базового елементу (7.2) та площу плоского геометричного об'єкту *Sp*(7.3). В таблиці 7.5 представлено залежність кількості вершин на зовнішньому контурі плоского геометричного об'єкта та коефіцієнта точного відображення ζ від точності апроксимації ε.

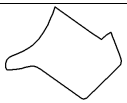
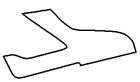
Проаналізувавши дані, що одержані про залежність кількості вершин апроксимуючого многокутника *N* та коефіцієнта точного відображення ζ від ущільнення інформації про зовнішні контури плоского геометричного об'єкту при заданій точності апроксимації ε (див. табл. 3.4.1) можна порекомендувати при автоматичній підготовці інформації про зовнішні контури плоского геометричного об'єкту вибрати точності апроксимації 0.5 *мм*≤ε ≤0.25 *мм.* При такій точності коефіцієнт точного відображення знаходиться в діапазоні 0.985≤ζ≤0.993. При точності ε=0.5 *мм* площа апроксимуючого многокутника буде відрізнятись від площі плоского геометричного об'єкту не більше ніж 1.5%, а при точності ε=0.25 *мм* площа апроксимуючого многокутника буде відрізнятись від площі плоского геометричного об'єкту не більше ніж 0.7%. Кількість вершин апроксимуючого многокутника при 0.5 *мм*≤ε ≤0.25 *мм* буде не більше ніж

60, що повністю задовольняє алгоритмам раціонального розкрою матеріалів.

Таблиця 3.4.1.

Залежність кількості вершин на зовнішньому контурі плоского геометричного об’єкта від точності апроксимації ε

N п.п.	Деталь	К-сть вер-шин	Площа S в $см^2$ та периметр P в *см*.	Кількість вершин після ущільнення для				
				ε =1 *мм*	ε =0.75 *мм*	ε =0.5 *мм*	ε =0.25 *мм*	ε =0.1 *мм*
1	2	3	4	5	6	7	8	9
1		623	S=16.81 P=28.48	14 ζ= 0.936	16 ζ= 0.948	19 ζ= 0.972	86 ζ= 0.995	317 ζ= 0.9999
2		682	S=40.39 P=30.61	14 ζ= 0.966	15 ζ= 0.974	20 ζ= 0.987	29 ζ= 0.996	417 ζ= 0.9999
3		816	S=76.39 P=48.73	24 ζ= 0.977	29 ζ= 0.985	45 ζ= 0.990	59 ζ= 0.998	439 ζ= 0.9999

4		875	*S*=82.12 *P*=44.5	18 ζ= 0.978	23 ζ= 0.986	28 ζ= 0.992	63 ζ= 0.999	506 ζ= 0.9999
5		598	*S*=84.37 *P*=51.27	20 ζ= 0.985	28 ζ= 0.987	57 ζ= 0.997	98 ζ= 0.999	317 ζ= 0.9999
1	2	3	4	5	6	7	8	9
6		659	*S*=93.02 *P*=49.4	17 ζ= 0.976	21 ζ= 0.987	57 ζ= 0.994	96 ζ= 0.999	426 ζ= 0.9999
7		680	*S*=97.03 *P*=44.9	18 ζ= 0.982	22 ζ= 0.988	29 ζ= 0.995	69 ζ= 0.999	501 ζ= 0.9999
8		717	*S*=115.55 *P*=47.1	21 ζ= 0.988	24 ζ= 0.989	39 ζ= 0.995	74 ζ= 0.999	545 ζ= 0.9999
9		803	*S*=117.28 *P*=72.33	27 ζ= 0.98	38 ζ= 0.98	57 ζ= 0.99	83 ζ= 0.99	356 ζ= 0.99

				5	9	5	9	99
10		756	*S*=118.51 *P*=45.82	18 ζ= 0.985	21 ζ= 0.990	31 ζ= 0.996	76 ζ= 0.999	407 ζ= 0.99999
11		614	*S*=179.18 *P*=68.28	34 ζ= 0.986	56 ζ= 0.990	71 ζ= 0.998	96 ζ= 0.999	377 ζ= 0.99999
12		856	*S*=186.14 *P*=77.52	27 ζ= 0.990	35 ζ= 0.992	55 ζ= 0.997	98 ζ= 0.999	398 ζ= 0.999999
1	2	3	4	5	6	7	8	9
13		842	*S*=217.05 *P*=85.4	28 ζ= 0.990	48 ζ= 0.992	63 ζ= 0.998	97 ζ= 0.9997	549 ζ= 0.99999
14		641	*S*=220.18 *P*=71.76	26 ζ= 0.992	43 ζ= 0.994	67 ζ= 0.999	96 ζ= 0.9999	438 ζ= 0.99999

3.5. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНТЕРАКТИВНОГО ПРОЕКТУВАННЯ ДЕТАЛЕЙ ГАЛАНТЕРЕЙНИХ ВИРОБІВ

Структура програмного продукту для інтерактивного проектування деталей галантерейних виробів

Використовуючи параметричні моделі для деталей жіночих сумок, що представлені у розділах 2-3 було створено програмне забезпечення для автоматизованої підготовки інформації про зовнішні контури деталей та декоративних елементів жіночих сумок. Програмний продукт реалізований на мові програмування Object Pascal в середовищі програмування Delphi для операційної системи Windows. Система програмування Delphi розрахована на програмування різних додатків і надає велику кількість компонентів для цього. До того ж роботодавців цікавить, насамперед, швидкість і якість створення програм, а ці характеристики може забезпечити тільки середовище візуального проектування, здатне взяти на себе значні обсяги рутинної роботи з підготовки додатків, а також узгодити діяльність групи постановників, кодувальників, тестерів і технічних письменників. Можливості Delphi повністю відповідають подібним вимогам і підходять для створення систем будь-якої складності.

Для того щоб обгрунтувати, чому наш вибір зупинився на Borland Delphi 7, досить просто перерахувати деякі недоліки мови C ++ порівняно з ObjectPascal [64-69]:

1. Треба робити багато ініціалізації (реєструвати клас вікна, організовувати цикл обробки повідомлень, створювати віконну функцію, піктограму та інше 2. Частково бути системним програмістом. На Delphi - системне програмування вже вбудовано і ініціалізація працює за замовчуванням, тому програміст головний акцент робить на своїх алгоритмах, а не на організації допоміжних робіт.

3. Значно більша, порівняно з Object Pascal, складність мови, навіть, незважаючи на компактність коду, виникають складнощі в його сприйнятті.

4. Одна особливість, на мій погляд, мови C ++ дуже псує цю мову - він чутливий до регістру символів, тобто змінна A і змінна a - це різні змінні.

5. У Delphi класи (об'єкти) можуть розташовуватися тільки в динамічної пам'яті, а в C ++ в будь-якій пам'яті (статична, стек, динамічна). Це додає безпеки програмування в Delphi.

Розроблений програмний продукт має дружній інтерфейс і не вимагає спеціальних знань з комп'ютерних наук при роботі з ним.

Розроблений програмний продукт для проектування деталей та декоративних елементів жіночих сумок може бути використаний на будь-якому шкіргалантерейних підприємстві і в навчальному процесі. При використанні в промисловості він зробить роботу конструктора творчою та забезпечить автоматизовану підготовку інформації про зовнішні контури деталей та декоративних елементів жіночих сумок. А це в свою чергу дозволить впровадити роботизовані комплекси в виробництві, що забезпечить підвищення ефективності виробництва і раціональне використання матеріалів.

Структурна схема програмного продукту представлена на рис. 3.5.1-3.5.3.5. Як видно з рис. 3.5.1 програмний продукт для автоматизованого проектування деталей галантерейних виробів складається із трьох основних модулей: модуль автоматизованого проектування деталей галантерейних виробів (рис. 3.5.2); модуль проектування декоративних елементів галантерейних виробів, що представляють окремі деталі (рис. 3.5.3); модуль проектування декоративних елементів галантерейних виробів, що представляють комбінацію фігурних отворів на деталях галантерейних виробів (рис. 3.5.4); модуль збереження інформації та

виводу на друк креслень деталей жіночих сумок та креслень декоративних елементів (рис. 3.5.5).

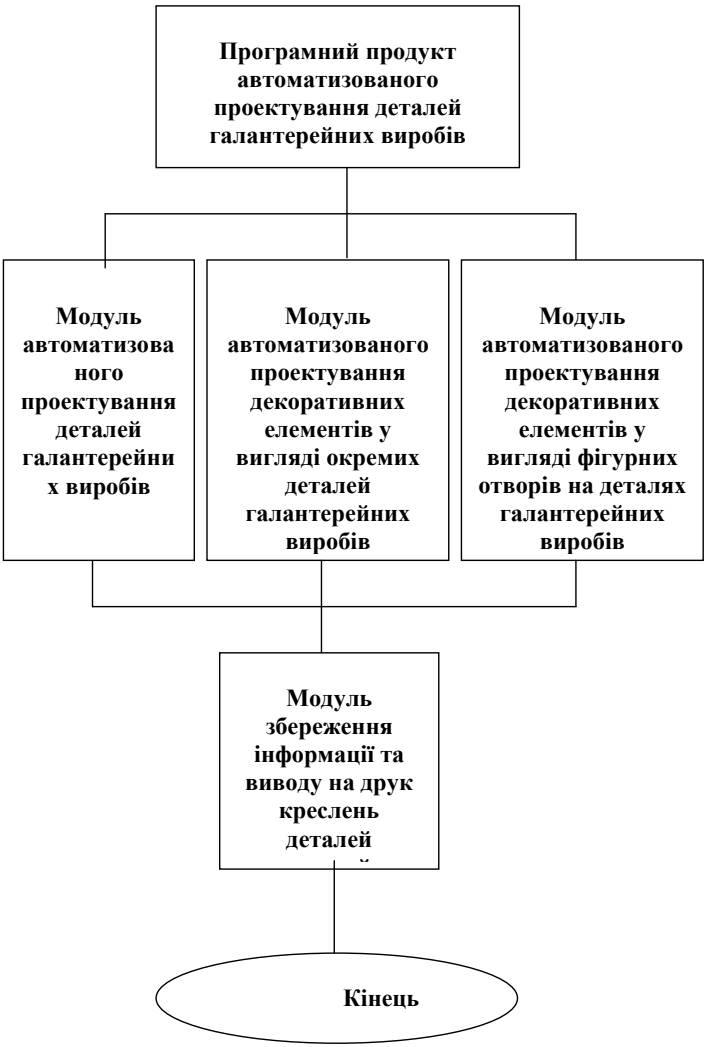


Рис. 3.5.1. **Структурна схема програмного забезпечення автоматизованого проектування галантерейних виробів**

Рис. 3.5.3. **Модуль автоматизованого проектування декоративних елементів, що представляють окремі деталі галантерейних виробів**

Рис. 3.5.2. **Модуль автоматизованого проектування деталей галантерейних виробів**

Рис. 3.5.4. **Структурна схема модулю автоматизованого проектування декоративних елементів у вигляді фігурних елементів на деталях галантерейних виробів**

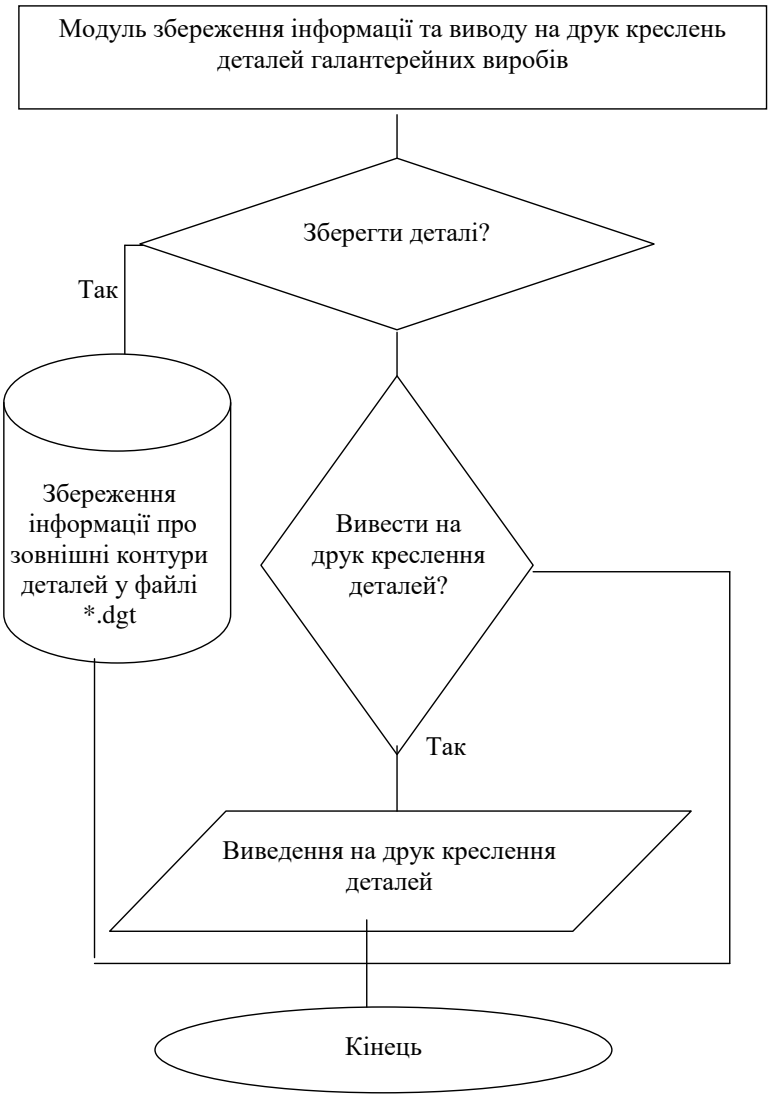


Рис. 3.5.5. **Модуль збереження інформації та виводу на друк креслень деталей галантерейних виробів**

На основі розроблених параметричних моделей деталей та декоративних елементів жіночих сумок було розроблене програмне забезпечення для автоматизованого проектування деталей та декоративних елементів жіночих сумок. Програмне забезпечення складається із двох програм, на які отримане авторські права на комп'ютерні програми (додаток В): для автоматизованого проектування декоративних елементів жіночих сумок «АПДЕЖС» (свідоцтво №6170); для автоматизованого проектування деталей жіночих сумок «АПДЖС» (свідоцтво №6171).

Програмне забезпечення також впроваджене у виробництво та навчальний процес. Копії актів впровадження представлені в додатку Г.

Інтерфейс програмного продукту інтерактивного проектування деталей галантерейних виробів

Модуль інтерактивного проектування деталей галантерейних виробів

Модуль автоматизованого проектування деталей сумок дозволяє із розглянутих у розділі 1 можливих форм для деталей сумок за кодом деталі та її параметрами створити параметричну модель відповідної деталі та однозначно відтворити зовнішній контур деталі. Структурна схема модуля автоматизованого проектування деталей сумок представлена на рис. 3.5.2.

Проектування деталей з простою конфігурацією зовнішніх контурів. Під деталями з простою конфігурацією зовнішніх контурів будемо розуміти деталі, які можна представити простими геометричними фігурами, а саме: прямокутниками, трапеціями, трикутниками, шестикутниками, еліпсами та напівеліпсами.

Приклади проектування деталей з трапецеїдальної формою зовнішніх контурів представлений на рис. 3.5.6.

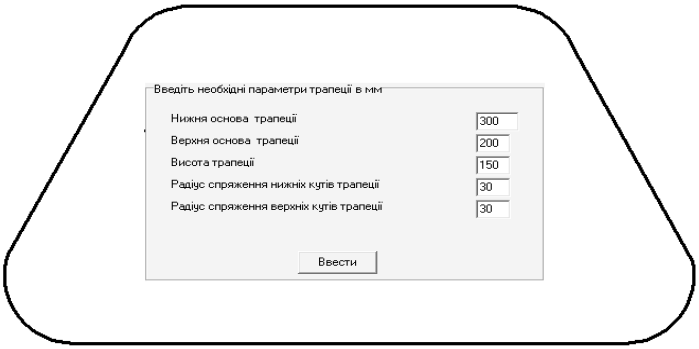


Рис. 3.5.6. **Генерування трапецеїдальної деталі зі спряженими дугою нижніми та верхніми кутами**

Проектування деталей, які можна представити комбінацією двох простих геометричних фігур. Ці деталі генеруються аналогічним чином завданням коду форми деталі та її параметрами, що забезпечують однозначне генерування зовнішнього контуру деталі. Код форми деталі визначає в якій послідовності будуть комбінуватись прості геометричні фігури, а параметри простих геометричних фігур однозначно визначать шуканої деталі (рис. 3.5.7).

Рис. 3.5.7. **Генерування деталі з формою зовнішнього контуру**

Проектування деталей , які можна представити комбінацією трьох простих геометричних фігур. Ці деталі генеруються аналогічним чином завданням коду форми деталі та її параметрами, що забезпечують однозначне генерування зовнішнього контуру деталі.. Код форми деталі визначає в якій послідовності будуть комбінуватись прості геометричні фігури, а параметри простих геометричних фігур однозначно визначать шукану деталь(рис. 3.5.8).

Рис. 3.5.8. **Генерування деталі з формою зовнішнього контуру у вигляді комбінації прямокутника трапеції та напівеліпсу**

Модуль інтерактивного проектування декоративних елементів жіночих сумок у вигляді окремих деталей

Структурна схема модуля автоматизованого проектування декоративних елементів галантерейних виробів у вигляді окремих деталей представлена на рис. 3.5.3. Цей модуль складається із трьох частин: генерування декоративних елементів, що представляють: окремі деталі у вигляді зірок; у вигляді квітів; у вигляді бантів. Приклади згенерованих

декоративних елементів у вигляді окремих деталей представлені на рис. 3.5.9-3.5.11.

Рис. 3.5.9. **Генерування декоративних елементів у вигляді**

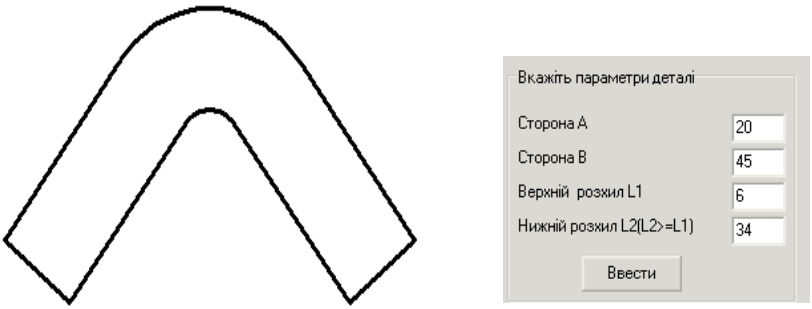


Рис. 3.5.10. **Генерування декоративних елементів у вигляді бантv**

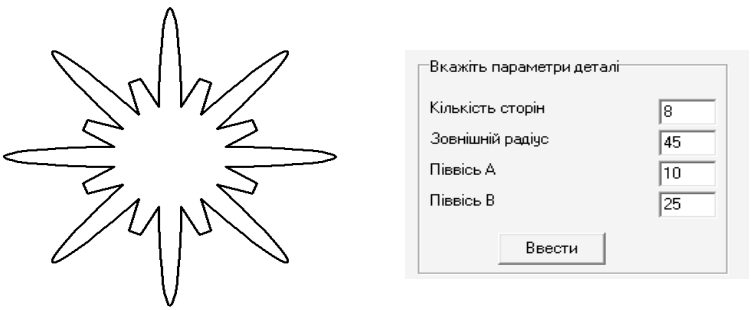


Рис. 3.5.11. **Генерування декоративних елементів у вигляді зірок**

Модуль інтерактивного проектування декоративних елементів галантерейних виробів у вигляді фігурних отворів на окремих деталей

Структурна схема модуля автоматизованого проектування декоративних елементів жіночих сумок у вигляді окремих деталей представлена на рис. 3.5.4. Цей модуль складається із трьох частин: генерування декоративних елементів на контурах деталей; генерування декоративних елементів, що представляють собою циклічні групові декоративні елементи; системні групові декоративні елементи. Приклади згенерованих декоративних елементів на контурах деталей представлені на рис. 3.5.11.

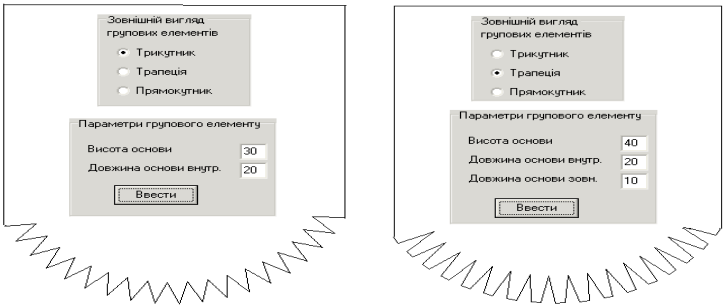


Рис. 3.5.11. **Генерування декоративних елементів у вигляді зубців на контурах деталі**

Циклічні групові декоративні елементи складаються з основного та допоміжних одинарних декоративних елементів. Форма основного та допоміжних декоративних елементів у циклічних групових елементів може бути різною. Розглядаються одинарні групові декоративні елементи у вигляді квітів з різною кількістю та формою пелюстків. Форма пелюсток може бути у вигляді: трикутника, трапеції, прямокутника, ромбу, краплі, лівої дуги кола, правої дуги кола, двох дуг кола. Приклад згенерованих циклічних групових декоративних елементів на деталей жіночих сумок представлений на рис. 3.5.12.

Розглянемо системні групові декоративні елементи на деталях галантерейних виробів. Системні групові декоративні елементи складаються із однакових одинарних групових декоративних елементів у вигляді квітів з різною кількістю та формою пелюстків. Форма пелюсток може бути у вигляді: трикутника, трапеції, прямокутника, ромбу, краплі, лівої дуги кола, правої дуги кола, двох дуг кола. Приклад згенерованих системних групових декоративних елементів на деталей жіночих сумок представлений на рис. 3.5.13.

Основні процедури та функції програмного забезпечення для інтерактивного проектування деталей галантерейних виробів, що можуть бути використані в при розробці САПР в інших галузях легкої промисловості

Процедура *Graph* призначена для виводу креслення деталей на екран. Виклик її здійснюється наступним чином :

Graph(n,col,x,y, XcF,YcF,mxy),

де *n* – кількість вершин в апроксимуючому многокутнику для деталі, що виводиться на креслення;

Col – номер кольору (якщо col – 1, тоді креслення буде зображуватись червонного кольору, 2 – синього, 3 – червоного,4 – зеленого, якщо ж інше значення – то жовтого):

1:Pen.Color:=clRed;

2:Pen.Color:=clBlue;

3:Pen.Color:=clBlack;

4:Pen.Color:=clGreen;

else Pen.Color:=clYellow;

x,y – масив координат вершин апроксимуючого многокутника;

XcF,YcF – координати центру прямокутника, описаного навколо деталі, що будується, при чому сторони прямокутника паралельні осям координат, кожна сторона прямокутника має точку дотику до деталі;

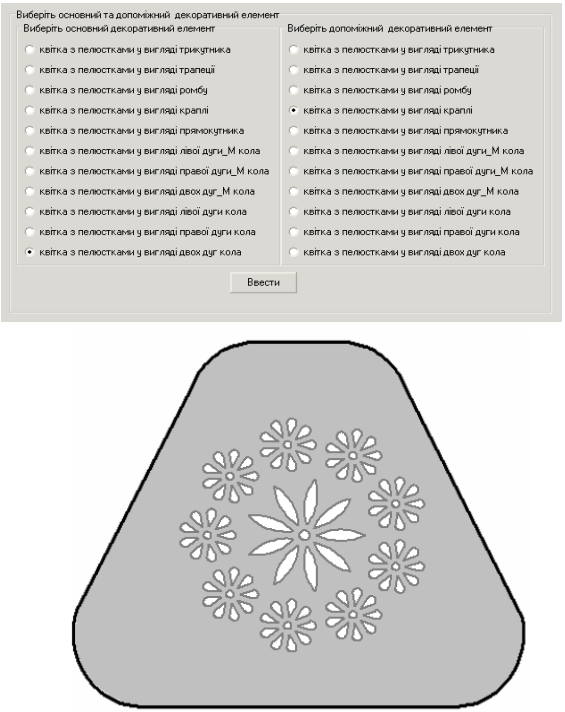


Рис. 3.5.12. **Генерування циклічних декоративних елементів у вигляді фігурних отворів на деталі**

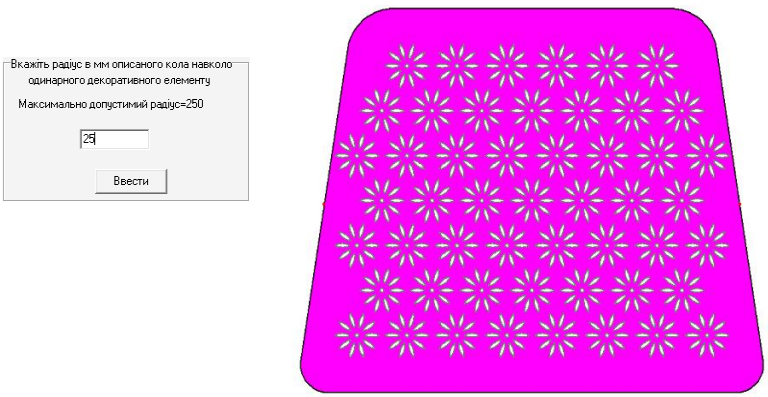


Рис. 3.5.13. **Генерування системних декоративних елементів у вигляді фігурних отворів на деталі**

mxy – масштаб, з яким виводиться зображення на екран.

Типи формальних параметрів

n,col:integer –цілі;

x,y:array of real – масив дійсних чисел (індекс 1-го елемента якого дорівнює нулю);

XcF,YcF:real – дійсне число;

mxy:real – дійсне додатне число;

Текст процедури *Graph* наведений нижче:

```
Procedure Graph(n,col:integer;x,y:array of real; XcF,YcF,mxy:real);
  var Xr,Yr:array [0..200]of integer;
     i,XcIm,YcIm:integer;
   begin
    with Form1.Image2.Canvas do
     begin
       XcIm:=round(Form1.Image2.Width/2);
     YcIm:=round(Form1.Image2.Height/2);
      end;
      For i:=0 to N-1 do
                 begin
                  Xr[i]:=round((X[i]-XcF)*mxy)+XcIm;
                  Yr[i]:=round((-Y[i]+YcF)*mxy)+YcIm;
                 end;
          with Form1.Image2.Canvas do
     begin
          case col of
                    1:Pen.Color:=clRed;
                    2:Pen.Color:=clBlue;
                    3:Pen.Color:=clBlack;
        else Pen.Color:=clYellow;
```

```
    end;
        For i:=0 to N-1 do
        if i=0 then MoveTo(Xr[i],Yr[i])
       else LineTo(Xr[i],Yr[i]);
  end;
 end;
```

Процедура *Splayn_B* призначена для згладжування заданої ділянки зовнішнього контуру вказаної деталі за допомогою параметричного *B*-сплайну. Ділянка контуру задається номерами вершин, між якими відбувається згладжування контуру. Виклик процедури здійснюється наступним чином:

Splayn_B(n1,n2,l, x,y,xr,yr, m);

де *n1* – номер початокової вершини контуру;

n2 – номер кінцевої вершини контуру, при чому *n2-n1>3*;

l – кількість додаткових вершин після згладжування на ділянці між двома сусідніми вершинами апроксимуючого многокутника;

x,y – масив координат вершин апроксимуючого многокутника;

xr,yr – отримані координати вершин ділянки контуру після згладжування *m* – кількість вершин в апроксимуючій кривій *xr,yr.*

Текст процедури *Splayn_B* наведений нижче:

```
Procedure Splayn_B(n1,n2,l:integer; var x,y,xr,yr:array of real; var
m:integer);
        Var i,j,k:integer;
          a0,a1,a2,a3,b0,b1,b2,b3,d,t:real;
          xx1,xx2,yy1,yy2,xn,yn,xk,yk:real;
          xx,yy:array[0..100]of real;
        begin
          For i:=n1 to n2 do
           begin
```

```
        Xx[i]:=x[i];
        yy[i]:=y[i];
    end;
      m:=-1;
      yy1:=y[n1];
      xx1:=x[n1];
      xx2:=x[n2];
      yy2:=y[n2];
      yy[n1-1]:=yy1;
      xx[n1-1]:=xx1;
      yy[n2+1]:=yy2;
      xx[n2+1]:=xx2;
     m:=m+1;
     xr[m]:=xx1;
     yr[m]:=yy1;
for i:=n1 to n2-1 do
 begin
  a3:=(-xx[i-1]+3*(xx[i]-xx[i+1])+xx[i+2])/6;
  a2:=(xx[i-1]-2*xx[i]+xx[i+1])/2;
  a1:=(xx[i+1]-xx[i-1])/2;
  a0:=(xx[i-1]+4*xx[i]+xx[i+1])/6;
   b3:=(-yy[i-1]+3*(yy[i]-yy[i+1])+yy[i+2])/6;
   b2:=(yy[i-1]-2*yy[i]+yy[i+1])/2;
   b1:=(yy[i+1]-yy[i-1])/2;
   b0:=(yy[i-1]+4*yy[i]+yy[i+1])/6;
   k:=1;
      for j:=m+1 to m+l do
        begin
          t:= k/l;
```

```
        k:=k+1;
          xr[j]:=((a3*t+a2)*t+a1)*t+a0;
          yr[j]:=((b3*t+b2)*t+b1)*t+b0;
        end;
    m:=m+L;
  end;
    m:=m+1;
    xr[m]:=xx2;
    yr[m]:=yy2;
end;
```

Процедура *Splayn_Inter* призначена для згладжування заданої ділянки зовнішнього контуру вказаної деталі за допомогою параметричного інтерполяційного сплайну. Ділянка контуру задається номерами вершин, між якими відбувається згладжування контуру. Виклик процедури здійснюється наступним чином:

Splayn_Inter(n1,n2,l, x,y,xr,yr, m);

де *n1* – номер початокової вершини контуру;

n2 – номер кінцевої вершини контуру, при чому *n2-n1>3*;

l – кількість додаткових вершин після згладжування на ділянці між двома сусідніми вершинами апроксимуючого многокутника;

x,y – масив координат вершин апроксимуючого многокутника;

xr,yr – отримані координати вершин ділянки контуру після згладжування *m* – кількість вершин в апроксимуючій кривій *xr,yr.*

Текст процедури *Splayn_B* наведений нижче:

Procedure Splayn_Inter (n1,n2,l:integer; var x,y,xr,yr:array of real; var m:integer);

```
Var i,j,k:integer;
   a0,a1,a2,a3,b0,b1,b2,b3,d,t:real;
   xx1,xx2,yy1,yy2,xn,yn,xk,yk:real;
   xx,yy:array[0..200]of real;
 begin
    For i:=n1 to n2 do
     begin
      Xx[i]:=x[i];
      yy[i]:=y[i];
     end;
    m:=-1;
    yy1:=y[n1];
    xx1:=x[n1];
    xx2:=x[n2];
    yy2:=y[n2];
    yy[n1-1]:=yy1;
    xx[n1-1]:=xx1;
    yy[n2+1]:=yy2;
    xx[n2+1]:=xx2;
   m:=m+1;
   xr[m]:=xx1;
   yr[m]:=yy1;
for i:=n1 to n2-1 do
  begin
   a3:=(-xx[i-1]+3*(xx[i]-xx[i+1])+xx[i+2])/2;
   a2:=(2*xx[i-1]-5*xx[i]+4*xx[i+1]-xx[i+2])/2;
   a1:=(xx[i+1]-xx[i-1])/2;
   a0:=xx[i];
    b3:=(-yy[i-1]+3*(yy[i]-yy[i+1])+yy[i+2])/2;
```

```
b2:=(2*yy[i-1]-5*yy[i]+4*yy[i+1]-yy[i+2])/2;
b1:=(yy[i+1]-yy[i-1])/2;
b0:=yy[i];
k:=1;
  for j:=m+1 to m+l do
    begin
      t:= k/l;
      k:=k+1;
      xr[j]:=((a3*t+a2)*t+a1)*t+a0;
      yr[j]:=((b3*t+b2)*t+b1)*t+b0;
     { writeln(j:5,xr[j]:6:2,yr[j]:6:2);
      readln}
    end;
 m:=m+L
end;
m:=m+1;
 xr[m]:=xx2;
 yr[m]:=yy2;
end;
```

Процедура *DetNewNumPointR* циклічна, переносить ділянку контуру деталі, що знаходиться між вершинами *Nb,Ne* в кінець масиву з координатами вершин *xdet,ydet*. Виклик процедури здійснюється наступним чином:

```
DetNewNumPointR(Nb,Ne,kp,xdet,ydet);
```

де Nb,Ne – номера вершин ділянки контуру апроксимуючого многокутника, яку необхідно перенести;

xdet,ydet – координати зовнішнього контуру деталей.

kp – кількість вершин в апроксимуючому многокутнику

Типи формальних параметрів

Nb,Ne:integer – цілі додатні;

kp:integer – ціле додатнє;

xdet,ydet:array of real – масив дійсних чисел.

```
Procedure DetNewNumPointR(Var Nb,Ne:integer; Var kp:integer;
                          Var xdet,ydet:array of real);
 Var D1,D2:real;
     i,Nr,kol:integer;
     Xr,Yr:array [0..100]of real;
     Xc,Yc:Int64;
 Begin
  If Nb>Ne then
   begin
    Nr:=Nb;
    Nb:=Ne;
    Ne:=Nr
   end;
   D1:=0;
   For i:=Nb to Ne-1 do
    D1:=D1+Sqrt(sqr(Xdet[i+1]-Xdet[i])+sqr(Ydet[i+1]-Ydet[i]));
   D2:=0;
   For i:=0 to Nb-1 do
    D2:=D2+Sqrt(sqr(Xdet[i+1]-Xdet[i])+sqr(Ydet[i+1]-Ydet[i]));
   For i:=Ne to KP-2 do
    D2:=D2+Sqrt(sqr(Xdet[i+1]-Xdet[i])+sqr(Ydet[i+1]-Ydet[i]));
   If D1<D2 then
    begin
     kol:=Ne-Nb+1;
     for i:=Ne to kp-1 do
      begin
```

```
   Xr[i-Ne]:=Xdet[i];
   Yr[i-Ne]:=Ydet[i];
  end;
 for i:=1 to Ne do
  begin
   Xr[kp-Ne-1+i]:=Xdet[i];
   Yr[kp-Ne-1+i]:=Ydet[i];
  end;
 end
else
 begin
  kol:=Kp-Ne+Nb;
 for i:=Nb to Kp-1 do
  begin
   Xr[i-Nb]:=Xdet[i];
   Yr[i-Nb]:=Ydet[i];
  end;
 for i:=1 to Nb do
  begin
   Xr[Kp-Nb-1+i]:=Xdet[i];
   Yr[Kp-Nb-1+i]:=Ydet[i];
  end;
end;
Nb:=Kp-kol;
Ne:=Kp-1;
 for i:=0 to Kp-1 do
  begin
   Xdet[i]:=Xr[i];
   Ydet[i]:=Yr[i];
```

```
    end;
  End;
```

Функція *CrossTwoCircle* приймає значення *true,* якщо два кола, характеристики яких задані, перетинаються та знаходить точки їх перетину. Виклик функції здійснюється наступним чином:

flag:=CrossTwoCircle(Xa,Ya,Ra,Xb,Yb,Rb,X1,Y1,X2,Y2);

де *flag*– допоміжна змінна, яка приймає значення *true,* якщо два задані кола перетинаються та *false,* якщо задані кола не перетинаються.

Xa,Ya – координат центру 1-го кола;

Ra – радіус 1-го кола;

Xb,Yb – координат центру 2-го кола;

Rb – радіус 2-го кола;

X1,Y1,X2,Y2 – координати точок перетину заданих кіл;

Типи формальних параметрів

Xa,Ya,Xb,Yb:real – дійсні;

Ra, Rb –дійсні;

X1,Y1,X2,Y2 – дійсні;

flag-логічне,приймає значення тільки true або false.

```
Function CrossTwoCircle(Xa,Ya,Ra,Xb,Yb,Rb:real;
 Var X1,Y1,X2,Y2:real):boolean;
 Var D,A,B,C,A1,B1,C1,R1,R2:real;
 Begin
  D:=Sqrt(Sqr(Xb-Xa)+Sqr(Yb-Ya));
  If Ra>Rb then
   begin
    R2:=Rb;
    R1:=Ra
   end
  else
```

```
 begin
  R1:=Rb;
  R2:=Ra
 end;
If (D>Ra+Rb)Or(D+R2<R1) then
 begin
  CrossTwoCircle:=False;
  Exit
 end
else
 begin
  CrossTwoCircle:=True;
  A:=2*(Xb-Xa);
  B:=2*(Yb-Ya);
  C:=-(Ra*Ra-Rb*Rb+Yb*Yb-Ya*Ya+Xb*Xb-Xa*Xa);
  if Xa=Xb then
   begin
    B1:=-2*Xa;
    C1:=Xa*Xa+(C/B)*(C/B)+2*Ya*C/B+Ya*Ya-Ra*Ra;
    D:=Sqrt(B1*B1/4-C1);
    X1:=-B1/2+D;
    X2:=-B1/2-D;
    Y1:=C/B;
    Y2:=Y1;
   end
  else
   begin
    A1:=1+(B/A)*(B/A);
    B1:=2*(B/A*C/A+B/A*Xa-Ya);
```

```
    C1:=(C/A)*(C/A)+2*Xa*C/A+Xa*Xa+Ya*Ya-Ra*Ra;
    D:=Sqrt(B1*B1-4*A1*C1);
    Y1:=(-B1+D)/2/A1;
    Y2:=(-B1-D)/2/A1;
    X1:=(B*Y1+C)/A;
    X2:=(B*Y2+C)/A;
   end;
  end;
End;
```

Процедура *SmeshchenieVector* виконує зсув точки вздовж заданої прямої у вказаному напрямку на вказану величину.

Виклик процедури здійснюється наступним чином:

SmeshchenieVector(Xa,Ya,Xb,Yb,delta,Xc,Yc);

де *Xa,Ya* – координати точки початку відрізку, який лежить на прямій;

Xb,Yb – координати точки кінця відрізку, який лежить на прямій,

від якої здійснюється зсув;

delta – величина,на яку необхідно здійснити зсув (якщо *delta* від'ємне, тоді зсув буде відбуватись у зворотному напрямку, тобто в напрямку від кінця до початку відрізку та навпаки);

Xc,Yc – координати точки, отриманої внаслідок зсуву.

Типи формальних параметрів

Xa,Ya,Xb,Yb:real – дійсні;

delta:real – дійсне;

Xc,Yc:real –дійсне.

```
SmeshchenieVector(Xa,Ya,Xb,Yb,delta:real;var Xc,Yc:real);
var Dx,Dy,D,Edx,Edy:real;
begin
```

```
        Dx:=Xb-Xa; Dy:=Yb-Ya;
        D:=sqrt(sqr(Dx)+sqr(Dy));
        Edx:=Dx/D;
        Edy:=Dy/D;
        Xc:=Xb+delta*Edx;
        Yc:=Yb+delta*Edy;
    end;
```

Процедура *Perpend*-будує перпендикуляр заданої величини відносно прямої, що задана відрізком.

Виклик процедури здійснюється наступним чином:

Procedure Perpend(Xa,Ya,Xb,Yb,delta, Xc1,Yc1,Xc2,Yc2);

де *Xa,Ya* – координати точки початку відрізку;

Xb,Yb – координати точки кінця відрізку,від якої будується перпендикуляр;

delta – величина перпендикуляру;

Xc1,Yc1,Xc2,Yc2-координати точок кінця перпендикуляра.

Процедура *ZkryglennjaGorlovunu* – будує дугу еліпса.

X0,Y0 – координати центу еліпса;

a – вликий радіус еліпса*;*

b – малий радіус еліпса;

Fi – кут від якого необхідно відкласти дугу

D_Fi – частина дуги еліпса, яку необхідно побудувати

Xr,Yr – координати після побудови.

X0,Y0,a,b,Fi,D_Fi:real –дійсні;

Q:integer – ціле;

Xr,Yr:array of real – масив дійсних чисел;

```
Procedure Perpend(Xa,Ya,Xb,Yb,delta:real;var Xc1,Yc1,Xc2,Yc2:real);
Var k,Xd1,Yd1,Xd2,Yd2,Dbd:real;
begin
```

```
if Xa=Xb then
begin
 Xc1:=Xb+delta;
 Xc2:=Xb-delta;
 Yc1:=Yb;
 Yc2:=Yb;
end
else
 if Ya=Yb then
    begin
     Yc1:=Yb+delta;
     Yc2:=Yb-delta;
     Xc1:=Xb;
     Xc2:=Xb;
    end
 else
   begin
     k:=(Yb-Ya)/(Xb-Xa);
     Xd1:=Xb+5;
     Xd2:=Xb-5;
     Yd1:=Yb-(Xd1-Xb)/k;
     Yd2:=Yb-(Xd2-Xb)/k;
     SmeshchenieVector(Xd1,Yd1,Xb,Yb,delta,Xc1,Yc1);
     SmeshchenieVector(Xd2,Yd2,Xb,Yb,delta,Xc2,Yc2);
   end;
end;
```

Висновки по розділу 3

1. В умовах ринкової економіки створення науково обґрунтованих методів проектування виробів шкіргалантереї з використанням обчислювальної техніки дозволить істотно скоротити терміни виконання робіт і підвищити продуктивність праці, поліпшити якість проектних рішень і, в кінцевому підсумку, підвищити ефективність роботи підприємств та автоматизація самого процесу проектування з ціллю отримання об'єктивних результатів за короткий період часу.

2. При створенні програмного продукту для проектування деталей галантерейних виробів, першим і найголовнішим є створення вихідної бази даних. Для виконання цієї задачі ми повинні ідентифікувати кожну спроектовану деталь за назвою; за матеріалом, із якого необхідно виготовити деталь; за формою зовнішнього контуру деталі.

3. При кусково-лінійному методі апроксимації зовнішній контур деталі з необхідною точністю представляє апроксимуючий многокутник. Для однозначного відображення апроксимуючого многокутника необхідно знати координати вершин цього многокутника та порядок їх обходу.

Список використаних джерел до розділу 3

1. Храмова Н.С. Разработка технологического процесса проектирования и формования текстильной обуви в AutoCAD : Дис….канд. техн. наук: 05.19.06. – К., 2000. – 182 с.

2. Альберг Дж. Теория сплайнов и ее приложение. – М.: Мир, 1972. – 316 с.

3. Добрышман Е.М., Калита А.Н. О возможности применения сплайнов при проектировании верха обуви // Известия ВУЗов. Технология легкой промышленности. – 1979. - № 4. – С.66-69.

4. Ботезат Л.А. Аппроксимация контуров нормализованных разверток с использование сплайн-функций / Л.А. Ботезат, М.А. Васильев // Известия ВУЗов. Технология легкой промышленности. – 1982. - № 3. – С. 107-109.

5. Каган В.М. К вопросу аппроксимации контура подошвы // Известия ВУЗов. Технология легкой промышленности. – 1974. - № 2. - С. 34-38.

6. Каган В.М. Проектирование контуров обувных деталей с применением ЭВМ // Кожевенно-обувная промышленность. – 1978. - № 1. – с. 37-39

7. Сторожев В.В. Аппроксимация контуров обувных деталей // Известия ВУЗов. Технология легкой промышленности. – 1975. - № 1. – С. 106-109

8. Цимбалюк В.М. Аналіз точності введення графічної інформації в САПР взуття / В.М. Цимбалюк, М.В. Манійчук // Вісник ТУП. – 2003. - № 5. – С. 33-35.

9. Карл де Бор. Практическое руководство по сплайнам / Карл де Бор – М. : Радио и связь, 1985. - 304 с.

10. Адамс А. Математические основы компьютерной графики / Адамс А., Роджерс Д. – М.: Машиностроение, 1981. - 606 с.

11. Гафуров Х. Л. / Х. Л. Гафуров, Т. Х. Гафуров, В. П. Смирнов //Системы автоматизированного проектирования. Судостроение, 2000. – 320 с.

12. De Boor On calculation with B-Spline / Applied Math Sciences, No 27, 1972, p. 50-62 Borm W. Inserting new Knots into B-Spline Curves / Computer Aided Design, No 12, 1980, p. 199-201

13. Шикин Е.В., Боресков А.В. Компьютерная графика. – М.: Диалог-МИФИ, 1995. – 377 с.

14. Зыбин Ю.П. Конструирование изделий из кожи / Ю.П. Зыбин, В.М. Ключникова, Т.С. Кочеткова, В.А. Фукин // – М.: Легкая и пищевая промышленность, 1982. – 263 с.
15. Николаева Ж.Б. Вопросы теории моделирования кожгалантерейных изделий // Кожевенно-обувная промышленность. – 1971. № 11. – С. 11-17.
16. Сумарокова Т.М., Татаров С.В. Конструирование кожгалантерейных изделий : Уч.пос. – СпБ.: СпБГУТД. – 1997. – 103 с.
17. Методика модульного (узлового) конструирования и моделирования дорожных изделий из унифицированных узлов и деталей: Утв. Минлегпромом СССР 14.12.82. – М.: ЦНИИТЭИлегпром. – 1984. – 7с.
18. Николаева Ж.Б. Технология кожгалантерейного и шорного производства. – М.: Легпромбытиздат, 1990. – 367 с.
19. Первышина Л.И., Ахмерова З.Р. Разработка классификатора кожгалантерейных изделий // Кожевенно-обувная промышленность. – 2000. - № 3. - с.22.
20. Фукин В.А. Классификация кожгалантерейных изделий для автоматизированного проектирования / В.А. Фукин, И.Б. Волкодаева // Кожевенно-обувная промышленность. – 1989. – № 4. – С. 16-18.
21. Вуштей О.А. Конструювання шкіргалантерейних виробів із використанням САПР / О.А. Вуштей, Н.В. Чупринка // Вісник КНУТД, – 2012. №2, – С. 13-16.
22. Омельченко Н. Н. Усовершенствонная методика автоматизированного проектирования перчаток / Н. Н. Омельченко, Н.В. Чупринка //Международный сборник научных трудов: Техническое регулирование: Базовая основа качества материалов и услуг, Россия, г. Шахты, 2013, – С. 65-67.
23. Чупринка Н.В. Возможности автоматизированного проектирования деталей женских сумок для улучшения их ассортимента и качества /

Н.В. Чупринка, С.С. Гаркавенко // Международный сборник научных трудов: Техническое регулирование: Базовая основа качества материалов и услуг, Россия, г. Шахты, 2014, – С. 94-97.

24. Чупринка Н.В. Класифікація зовнішніх контурів деталей жіночих сумок для автоматизованого проектування цих деталей з використанням стандартних елементів / Н.В. Чупринка, С.С. Гаркавенко // Тези доповідей Всеукраїнської науково-практичної конференції «Проблеми легкої і текстильної промисловості України», 22-24 жовтня 2014, м. Херсон, – С. 63-65.,
25. Чупринка Н.В. Автоматизованого проектування деталей жіночих сумок з використанням стандартних елементів / Н.В. Чупринка, С.С. Гаркавенко // Вісник КНУТД, – 2014. №6, – С. 42-48.
26. Ильин В.А. Аналитическая геометрия / В.А. Ильин, Э.Г. Позняк –М.: «Наука», 1975, – 243 с.
27. Фильчаков П.Ф. Справочник по высшей математике. – К.: Наукова думка, 1973. – 282 с.
28. Николаева Ж.Б. Моделирование кожгалантерейных изделий / Ж.Б. Николаева, С.М. Темкин, Н.Н. Шаповалова // –М.: легкая индустрия, 1975, – 240 с.
29. Чупринка Н.В. Автоматизована підготовка інформації про деталі шкіргалантерейних виробів / Н.В. Чупринка, С.С. Гаркавенко // Проблеми легкої промисловості України, м. Херсон, – 2012, №2(20), – С. 65-69.
30. Чупринка Н.В. Автоматизированное проектирование деталей кожгалантерейных изделий/ Н.В. Чупринка, С.С. Гаркавенко // Международный сборник научных трудов: Техническое регулирование: Базовая основа качества материалов и услуг, Россия, г. Шахты, 2013, – С. 63-65.

31. Чупринка В.І. Побудова еквідистанти для плоского геометричного обєкта / В. І. Чупринка, К.А. Шлімович // Вісник ДАЛПУ, – 2000. – №1, – С. 83-85.

32. Чупринка В.І. Алгоритм підготовки інформації для побудови розкрійних схем рулонних матеріалів на деталі взуття та шкіргалантерейних виробів/ В. І. Чупринка, О.З. Колиско // Вісник ДАЛПУ, – 2000. – №1, – С. 83-85

33. Дроменко В.І. Метод автоматизованого проектування рукавичкових виробів/ В.І. Дроменко, Н.В. Чупринка// Тези доповідей X Всеукраїнської наукової конференції молодих вчених та студентів, 19-20 квітня 2011 р. „Наукові розробки молоді на сучасному етапі". – Т2 - К.:КНУТД, 2011, с.58.

34. Чупринка Н.В. Програмне забезпечення для інтерактивної побудови схем розкрою натуральних матеріалів / Н.В. Чупринка, О.З. Колиско // Тези доповідей XI Всеукраїнської наукової конференції молодих вчених та студентів, 19-20 квітня „Наукові розробки молоді на сучасному етапі". - Т2, - К.:КНУТД, 2012, с.87-88.

35. Чупринка Н.В. Проектування сумок з використанням автоматизованих систем / Н.В. Чупринка, А.О. Терстуян, С.С. Гаркавенко, О.А. Вуштей // Тези доповідей XII Всеукраїнської наукової конференції молодих вчених та студентів, 25-26 квітня „Наукові розробки молоді на сучасному етапі". – Т1, - К.:КНУТД, 2013, с.142-143.

36. Чупринка Н.В. Автоматизоване проектування деталей шкіргалантерейних виробів / Н.В. Чупринка, С.С. Гаркавенко // Тези доповідей XII Всеукраїнської наукової конференції молодих вчених та студентів, 25-26 квітня „Наукові розробки молоді на сучасному етапі". – Т2 - К.:КНУТД, 2013, с.83-84.

37. Чупринка Н.В. Об особенностях автоматизированного проектирования украшений женских сумок для повышения их востребованости / Н.В. Чупринка, С.С. Гаркавенко // Международный сборник научных трудов: Техническое регулирование: Базовая основа качества материалов и услуг, Россия, г. Шахты, 2014, – С. 97-99.
38. Чупринка Н.В. Автоматизоване проектування декоративних елементів для жіночих сумок. / Н.В. Чупринка, С.С. Гаркавенко // Тези доповідей Всеукраїнської науково-практичної конференції «Проблеми легкої і текстильної промисловості України», 22-24 жовтня 2014, м. Херсон, – С. 127-129.
39. Чупринка Н.В. Автоматизоване проектування декоративних елементів жіночих сумок / Н.В. Чупринка, С.С. Гаркавенко // Легка промисловість, – 2014. – №3-4, – С. 46-47.
40. Чупринка Н.В. Автоматизоване проектування декоративних елементів на зовнішніх контурах деталей жіночих сумок / Н.В. Чупринка, С.С. Гаркавенко // Вісник КНУТД, – 2015. №3, – С. 42-48.
41. Чупринка Н.В. Параметрична модель циклічних групових декоративних елементів на деталях жіночих сумок / Н.В. Чупринка, С.С. Гаркавенко // Вісник КНУТД, – 2015. №3, – С. 237-243.
42. Чупринка Н.В. Програмне забезпечення для генерування декоративних елементів жіночих сумок / Н.В. Чупринка, С.С. Гаркавенко // Тези доповідей XIII Всеукраїнської наукової конференції молодих вчених та студентів, 24-25 квітня „Наукові розробки молоді на сучасному етапі". – Т1 - К.:КНУТД, 2014, с.85.
43. Чупринка Н.В. Програмне забезпечення для генерування декоративних елементів на деталях жіночих сумок / Н.В. Чупринка // Тези доповідей XIV Всеукраїнської наукової конференції молодих

вчених та студентів, 23-24 квітня 2015 р.„Наукові розробки молоді на сучасному етапі". – Т1 - К.:КНУТД, 2015, с.72.

44. Чупринка Н.В. Програмне забезпечення для генерування декоративних елементів у вигляді фігурних отворів на деталях жіночих сумок / Н.В. Чупринка // Тези доповідей XIV Всеукраїнської наукової конференції молодих вчених та студентів, 23-24 квітня 2015 р.„Наукові розробки молоді на сучасному етапі". – Т2 - К.:КНУТД, 2015, с.52.

45. Чупринка В.І. Алгоритм побудови еквідистанти для плоского геометричного об'єкта довільної форми. / В.І. Чупринка, О.О. Хоменко, Л. Т. Свістунова // Вісник КНУТД. – 2009. - №3. – С. 27-32

46. Глинер М.И.Об одном алгоритме аппроксимации контуров швейных лекал / М.И. Глинер, Е.А. Смирнов // Известия ВУЗов. Технология легкой промышленности. – 1976. – № 4. – С. 98-99.

47. Скатерной В.А. Оптимизация раскроя материалов в легкой промышленности / В.А. Скатерной. – М., Легпробытиздат, 1989, – 144 *с.*

48. Рябец О.І. Методика проектування верху взуття на програмному комплексі „Ірис" / О.І. Рябец, О.В. Піляєв // Известия высших учебных заведений. Технологыя легкой промышленности. – 1998. – №3. – с. 32-36.

49. Омельченко П.В. Алгоритм автоматизованої підготовки інформації про деталі шкіргалантерейних виробів / П.В. Омельченко, В.П. Коновал, В.І. Чупринка // Вісник КНУТД. – 2004. – № 1. – С. 85-89.

50. Кременок В.В. Автоматизоване проектування деталей взуття / В.В. Кременок, А.В. Пінчук, В.І. Чупринка. Тези доповідей VII Всеукраїнської наукової конференції молодих вчених та студентів. – Київ, 2008. – С. 123

51. Чупринка В.І., Чебанюк О.В. Алгоритм автоматичної підготовки вихідної інформації для побудови раціональних схем розкрою / В.І. Чупринка, О.В. Чебанюк // Вісник КНУТД. – 2006. – № 5. – С. 18-22.

52. Девид Форсайт. Компътерное зрение. Современный подход / Форсайт Девид, Понс Жан. – М.: Издательский дом «Вильямс», 2004, – 928 с.

53. Ильин В.А., Позняк Э.Г. Аналитическая геометрия / В.А. Ильин, Э.Г. Позняк. – М.:Издательство "Наука", Главная редакция физико-математической литературы., 1975, – 243 с

54. Воднев В.Т., Наумович А.Ф., Наумович Н.Ф.В.Т. Воднев, Основные математические формулы / А.Ф. Наумович, Н.Ф. Наумович. Минск: Вышэйшая школа, – 1988, – 270 с.

55. Омельченко П.В. Алгоритм підготовки та корегування інформації про контури деталей щкіргалантерейних виробів / П.В. Омельченко, В.П. Коновал, В.І. Чупринка // Вісник ТУП. – 2004. – № 2. – С. 217-220.

56. Омельченко П.В. Разработка метода автоматизированного проектирования чехлов для мобильных телефонов / П.В. Омельченко, В.П. Коновал, В.И. Чупринка // Актуальные проблемы науки, техники и экономики производства изделий из кожи. – Витебск: ВГТУ, 2004, – С. 131-132.

57. Адамс А. Математические основы компьютерной графики / Адамс А., Роджерс Д. – М.: Машиностроение, 1981. - 606 с.

58. Аммерал Л. Принципы прогаммирования в машинной графике/ Л. Аммерал – М.: Сол Систем, 1992. –224 с.
59. Щикин Е.В. Компъютерная графика / Е.В. Щикин, А.В. Бересков. –М.: Диалог-МИФИ, 1995. – 282 с.
60. Мелещенко О.А Апроксимація оснастки взуттєвого виробництва В-сплайнами / О.А Мелещенко, П.В. Омельченко, В.І. Чупринка // Зб. наук. пр. – К.: КНУТД, 2002. – С. 23.
61. Мелещенко О.А. Апроксимація деталей чохлів до мобільних телефонів / О.А Мелещенко, П.В. Омельченко, В.І. Чупринка // Зб. наук. пр. – К.: КНУТД, 2003. – С. 66
62. Омельченко П.В. Опис поверхні зовнішніх контурів деталей шкіргалантерейних виробів та колодки за допомогою параметричного В-сплайну / П.В. Омельченко, В.П. Коновал, В.І. Чупринка // Вісник ТУП. – 2003. – № 5. – С.36-38.
63. Колиско О.З. Автоматизована підготовка інформації про зовнішні контури деталей шкіргалантерейних виробів для побудови розкрійних схем деталей дрібної шкіргалантереї / О.З. Колиско, В.І. Чупринка. Тези доповідей V Всеукраїнської наукової конференції молодих вчених та студентів. – Київ, 2006. С. 61.
64. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0: Учеб. пособие. М.: Бином универсал; К.: Юниор, 1997, – 495 с.
65. Сердюченко В.Я. Розробка алгоритмів та програмування мовою Turbo Pascal: Навч. посібник для техн. вузів. – Х.: ВКП "Парітет" ЛТД, 1995. –352 с.
66. Крылов В.И. и др. Вычислительные методы высшей математики. Т.1. – Мн.: Вышэйшая школа, 1972. -584 с.

67. Крылов В.И. и др. Вычислительные методы высшей математики. Т.2. Мн.: Вышэйшая школа, 1975. –672 с.

68. Сердюченко В.Я. Розробка алгоритмів та програмування мовою Turbo Pascal: Навч. посібник для техн. вузів. – Х.: ВКП "Парітет" ЛТД, 1995. – 352 с.

4. ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ОЦІНЮВАННЯ РИЗИКІВ ОСВІТНЬОГО ПРОЦЕСУ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

4.1 АЛГОРИТМ ОЦІНЮВАННЯ РИЗИКІВ ОСВІТНЬОГО ПРОЦЕСУ

На сьогодні в умовах швидкоплинних вимог стейкхолдерів та ринку праці в країні виникає потреба у підготовці висококваліфікованих фахівців, які будуть конкурентоспроможними на ринку праці. Ринок праці все більше залежить від набутих професійних компетентностей фахівців, з однієї сторони та більш жорстких вимог стейкхолдерів до випускників закладів вищої освіти з іншої. За таких умов перед ЗВО постають задачі пошуку нових механізмів та інструментів щодо розроблення та впровадження внутрішніх систем якості, пов’язаних: з освітнім процесом, освітньою програмою, методами контролю, спрямованих на кінцевий результат – випуск конкурентоспроможного випускника.

Оскільки будь-яка діяльність, у тому числі й освітня пов’язані з ризиками то важливим етапом управління ризиками для організацій (підприємств, установ) у тому числі й закладів вищої освіти, є їх оцінка. Згідно зі стандартом ДСТУ ISO 31010 [1] дана процедура узагальнює процес ідентифікації та аналізу ризиків з метою їх подальшого оцінювання.

Для оцінювання ризиків освітнього процесу запропоновано застосовувати покроковий алгоритм [2] побудований за вимогами стандарту ДСТУ ISO 31010 [1], який враховує наступні процедури:

– ідентифікацію ризиків;
– якісний аналіз ризиків;
– кількісне оцінювання ризиків;
– розроблення та впровадження заходів щодо мінімізації впливу ризиків (рис. 4.1).

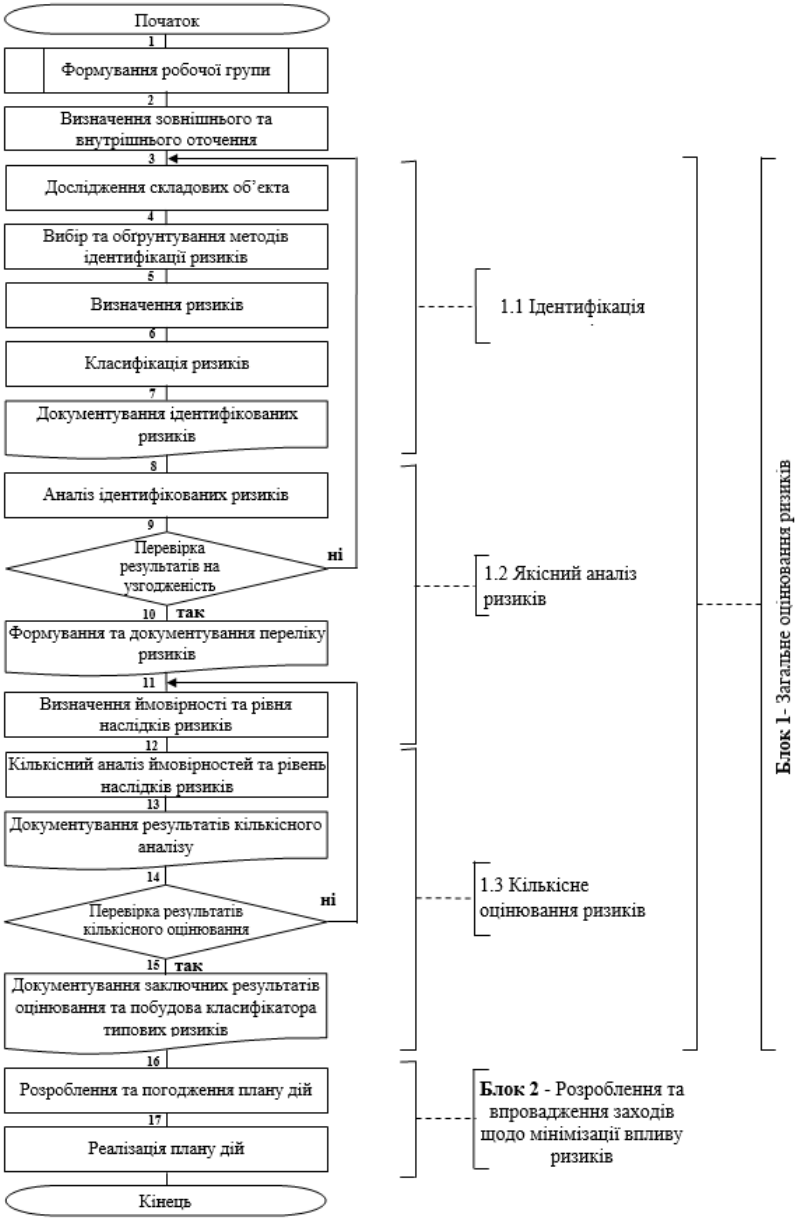


Рис. 4.1 – Алгоритм оцінювання ризиків освітнього процесу [2]

Згідно з наведеним алгоритмом на *першому кроці* оцінювання ризиків освітнього процесу необхідно сформувати робочу експертну групу. Слід зазначити, що від якості роботи експертної групи залежить повнота та вірогідність інформації щодо впливу ризиків на освітній процес за відповідною освітньою програмою [3]. Тому під час формування експертної групи необхідно особливу увагу приділяти компетентності експертів [4]. Оцінювання компетентності експертів варто проводити з урахуванням загальних професійно-орієнтованих критеріїв, які відображають специфіку освітньої діяльності закладів вищої освіти та ліцензійні вимоги МОН України, таких як: відповідна кваліфікація за галуззю знань та спеціальністю (науковий ступінь та вчене звання); стаж роботи; наявність публікацій за напрямом оцінювання (за останні 5-ть років); досвід роботи в експертних групах [5].

Одним із завдань експертної групи є забезпечення консультацій з питань виявлення ризиків та розроблення організаційно-технічних заходів щодо зменшення їх впливу на освітній процес (освітню програму).

На другому кроці алгоритму сформованою експертною групою проводиться визначення зовнішнього та внутрішнього середовища. Слід зазначити, що вони поділені на *два рівні*: макросередовище та мікросередовище.

До макросередовища відносяться такі фактори, як: політичні, економічні та соціальні. Дані фактори визначаються такими інструментами, як PEST та SWOT-аналіз.

Фактори мікросередовища залежать від об’єкта, що підлягає оцінюванню. Наприклад, для освітнього процесу вони обумовлені вимогами споживачів освітніх послуг (здобувачів вищої освіти, батьків, студентського самоврядування, науково-педагогічних та наукових працівників, керівництвом закладу), стейкхолдерів, ринку праці, відповідним їм матеріальним ресурсам тощо.

Починаючи з *третього кроку* алгоритму, процедури розбито на *два* блоки. Де *перший блок* – це *загальне оцінювання ризиків*, що стосується їх ідентифікації, якісного аналізу ризиків та їх кількісного оцінювання. *Другий блок* – це *розроблення та впровадження заходів щодо мінімізації впливу ризиків* (див. рис. 4.1).

До блоку «Загального оцінювання» ризиків відносяться наступні три підблоки: ідентифікація ризиків (1.1), якісний аналіз ризиків (1.2) та кількісне оцінювання ризиків (1.3). Виконання процедур перелічених підблоків проводиться відповідно до пункту 4.3.4 стандарту [1], шляхом застосування методів структурованого аналізу.

Основним підблоком цього блоку є – *ідентифікація* (1.1), що включає проведення наступних процедур:

крок три – дослідження складових об'єкта;

крок чотири – вибір та обґрунтування методів ідентифікації ризиків;

крок п'ять – визначення ризиків;

крок шість – класифікація ризиків;

крок сім – документування ідентифікованих ризиків.

Виконання наведених вище процедур дозволяє виявити джерела ризику, ризикові дії, з'ясувати причини їх виникнення та оцінити наслідки негативного впливу на освітній процес та його складові.

Особлива увага при цьому приділяється кроку чотири – вибору та обґрунтуванню методів, які надалі застосовуються до ідентифікації ризиків. Це пов'язано з тим, що дана процедура безпосередньо впливає на якість їх оцінювання [6, 7].

Наприклад, для освітнього процесу згідно міжнародного стандарту ДСТУ ISO 31010 (Додаток В) найбільш придатними є такі методи ідентифікації, як «Мозковий штурм», «Делфі» та «Аналіз небезпечних чинників і критичні точки контролю». Вони дозволяють досліджувати

складові освітнього процесу закладів вищої освіти, визначати ризики, класифікувати їх за певними ознаками та проводити їх документування.

Основні переваги та недоліки вище перелічених методів наведено в таблиці 4.1

Таблиця 4.1

Переваги й недоліки методів ідентифікації

Методи ідентифікації	Переваги	Недоліки
Мозковий штурм	результативне координування на початковому етапі; прийняття оригінальних рішень робочої групи, за рахунок творчого мислення, шляхом залучення зацікавлених сторін, загального обміну інформацією, швидкого та легкого запровадження.	великі витрати часу координатором на збір інформації (результатів думок експертів); домінування ідей одних учасників групи над іншими, що надалі потребує розроблення окремих процедур для урівноваження результатів досліджень
Делфі	анонімність, відсутність впливу на думку окремого експерта, можливість роботи групи в різних місцях.	досить великі затрати часу на його проведення.
Аналіз небезпечних чинників і критичні точки контролю	можливість отримувати задокументовані докази, які дозволяють надалі ідентифікувати та мінімізувати ризики, шляхом розробки контрольних листів, рекомендацій та організаційно-технічних заходів, тощо.	досить великі кадрові, економічні й часові затрати, у порівнянні з вище наведеними методами.

На п’ятому кроці визначаються ризики, за факторами негативного впливу відповідно до зовнішнього та внутрішнього оточення (визначених на *другому кроці*).

На шостому кроці проводиться класифікація виявлених ризиків освітнього процесу. Для цього необхідно мати ефективні механізми та інструменти побудовані на відповідних класифікаціях, які враховують специфіку освітнього процесу.

Класифікацію ризиків освітнього процесу необхідно проводити за типовою схемою класифікації [8] представленою у вигляді трирівневої структури (рис. 4.2):

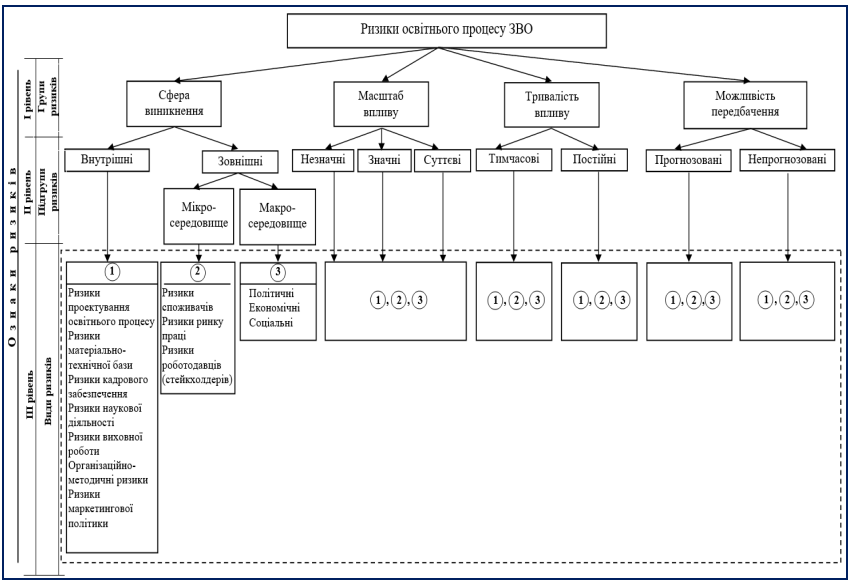


Рис. 4.2 – Типова схема класифікації ризиків освітнього процесу ЗВО [8]

Для побудови даної схеми було використано типову схему освітнього процесу наведену в джерелі [9], метод дерева цілей, ієрархій та вимоги стандарту ДСТУ ISO 31010. Застосування методу ієрархій дозволяє визначити пріоритети структурних складових освітнього процесу та розкриває сутність пов’язаних з ними ризиків. Суть даного методу полягає

в оцінюванні вищого рівня об’єкту, що досліджується (у нашому випадку – освітнього процесу (освітньої програми)) стосовно взаємопов’язаних нижчих рівнів (організаційно-методичних складових освітнього процесу та структурних складових освітньої програми), зв’язки між якими представляюся у вигляді ієрархії – трирівневої структури.

На першому рівні всі ризики поділені на 4-ри групи за ознаками:

– сфера виникнення;
– масштаб впливу;
– тривалість впливу та можливість передбачення.

На другому рівні групові ризики поділено на підгрупи (за допомогою методу структурування) на:

– внутрішні та зовнішні;
– незначні, значні та суттєві;
– тимчасові та постійні;
– прогнозовані та непрогнозовані.

В свою чергу підгрупа зовнішні ризики була поділена на мікросередовище та макросередовище.

На третьому рівні було вибрано й обґрунтовано фактори, що обумовлюють ризики освітнього процесу (освітньої програми), тобто визначено їх види.

Далі для підгрупи внутрішні ризики були виділені фактори, які впливають на якість проектування освітнього процесу:

– матеріально-технічну базу;
– кадрове забезпечення;
– наукову діяльність;
– виховну роботу, організаційно-методичне забезпечення та професійно-орієнтаційну роботу.

Аналогічно для зовнішніх ризиків було визначено фактори мікросередовища, серед яких виділено ризики притаманні споживачам, ринку праці, стейкхолдерам та фактори ризиків макросередовища –

політичні, економічні й соціальні.

Слід зазначити, що для освітнього процесу більш вагомим є мікросередовище яке обумовлюється вимогами: споживачів освітніх послуг, стейкхолдерів, умовами ринку праці, тобто суттєво впливає на конкурентоспроможність ЗВО в цілому. Дане середовище формується шляхом здобувачів вищої освіти, батьків, студентського самоврядування, науково-педагогічних та наукових працівників, вимог: випускових кафедр, деканатів, вченої ради та керівництва ЗВО, оточення без якого не може існувати освітній процес.

Відомо, що для функціонування кожного з середовищ потрібно мати відповідні матеріальні ресурси. Так наприклад, для освітнього процесу за відповідною освітньою програмою матеріальними ресурсами є:

– кадрове забезпечення науково-педагогічних працівників, що враховує професійно-орієнтовані критерії: компетентність, вчене звання (професор, доцент), науковий ступінь (доктор наук, кандидат наук, доктор філософії, магістр), стаж роботи за профілем в зазначених межах (від 1 до 5, від 5 до 10, від 10 до 20, від 20 до 30, від 30 і більше років), наявність публікацій (підручники, навчальні посібники, монографії), статті у наукометричних базах: Scopus, Web of science core Collection, рецензованих фахових виданнях (міжнародних і вітчизняних);

– навчально-методичне забезпечення освітньої програми (навчальні та робочі плани, навчальні та робочі програми дисциплін, конспекти лекцій, методичні розробки для лабораторних та практичних занять, пакет контрольних завдань з дисциплін, контроль знань (залік, екзамен, державний іспит / дипломна робота);

– матеріально-технічне забезпечення (лекційні

аудиторії з інноваційним обладнанням, навчальні професійно-орієнтовані лабораторії з використанням інформаційних технологій і програмного забезпечення, комп'ютерні класи);

– наукова діяльність, яка є інноваційною складовою освітнього процесу і включає: наукові дослідження в рамках держбюджетних робіт, госпдоговірні теми, ініціативні теми, договори про творчу співпрацю, конференції різних рівнів (міжнародні, всеукраїнські, інтернет-конференції, онлайн-конференції), круглі столи, стартапи, бізнес-інкубатори тощо.

При цьому, фактори, які обумовлюють внутрішні та зовнішні ризики стосуються усіх груп першого рівня. Варто зазначити, що їх поділ за видами (на третьому рівні) залежить від специфічних умов освітньої діяльності закладів вищої освіти. Один і той же ризик може по різному впливати на види діяльності, пов'язаної з функціонуванням освітнього процесу за відповідною освітньою програмою.

Ризики першого рівня – за ознакою «Масштаб впливу» було поділено за ступенем впливу на три підгрупи другого рівня: незначні, значні й суттєві, залежно від виду діяльності ЗВО. Кожній з цих підгруп на другому рівні можуть бути притаманні три види ризиків (або їх частина), що входять до ризиків першого рівня за ознакою «Сфера виникнення». Проте, для їх оцінювання потрібно мати спеціальну шкалу вагомості, яка є індивідуальною для кожного освітнього процесу ЗВО, що враховує специфіку його діяльності.

В свою чергу, ризики першого рівня за ознакою «Тривалість впливу» поділені на дві підгрупи: тимчасові й постійні.

До тимчасових віднесено:

– одноразові ризики;
– ризики, що виникають з певною циклічністю;
– ризики, що виникають за певних умов не циклічно.

До постійних – ризики, що мають безперервний вплив.

Остання група першого рівня – «Можливість передбачення» структурована на дві підгрупи: прогнозовані й непрогнозовані.

До прогнозованих віднесено ризики, що викликані діями внутрішнього середовища, тобто ті які можливо передбачити (врахувати), а до непрогнозованих – неочікувані ризики, пов'язані з зовнішнім середовищем (формуванням нової державної політики в освітній галузі; державною політикою фінансування сфери освіти за залишковим принципом; нестабільною політичною ситуацією в країні; інтеграцією в європейський простір вищої освіти; зменшенням кількості місць за державним замовленням, тощо), усі вони виникають під час планування, організації та проведення освітнього процесу [10].

Застосування даної методики дозволяє встановити взаємозв'язки між рівнями ризиків та визначити фактори їх впливу. Такий підхід дозволяє на *сьомому кроці* ідентифікувати ризики і сформувати відповідний документ.

Другий підблок – «Якісний аналіз ризиків» складається з наступних кроків:

крок вісім – аналіз ідентифікованих ризиків;

крок дев'ять – перевірка результатів на узгодженість;

крок десять – формування та документування переліку ризиків;

крок одинадцять – визначення ймовірності та рівня наслідків ризиків.

В свою чергу, на *восьмому кроці* визначаються причини і джерела ризиків, ймовірність їх виникнення та наслідки їх впливу. На *дев'ятому кроці* виконується перевірка результатів ідентифікації на узгодженість і залежність думок експертів. Загальна їх оцінка перевіряється на невизначеність за допомогою перевірки узгодженості оцінювання з використанням коефіцієнта конкордації W (4.1):

$$W = \frac{12}{m^2(n^3 - n)} \sum_{j=i}^{n} \left(\sum_{i=1}^{m} (R_{ij} - \frac{n+1}{2}) \right)^2, \qquad (4.1)$$

де n – кількість об'єктів;

m – кількість експертів;

R_{ij} – ранг j – го експерта, який встановлено і – м експертом.

У разі, якщо думки експертів мають велику розбіжність (невизначеність) потрібно повернутись на *перший крок* та визначити міру впливу кожного експерта на загальну оцінку експертної групи, методом вилучення одного експерта та обчисленням коефіцієнта конкордації W без урахування вилучених оцінок. Якщо під час вилучення оцінок експерта коефіцієнт – W зростає, то це говорить про некомпетентність вилученого експерта. Якщо W – спадає, то навпаки підтверджує компетентність вилученого експерта [11, 12]. В результаті виявлення некомпетентних експертів під час перевірки варто формувати нову експертну групу.

Такий підхід до підбору експертів дозволяє зменшити невизначеність під час проведення оцінювання.

У випадку, якщо результати ідентифікованих ризиків узгоджені, переходять на *крок десять* – «Формування та документування переліку ризиків». Дана процедура дозволяє побудувати взаємозв'язок між факторами, які є потенційними носіями негативного впливу. На *одинадцятому кроці* За сформованими результатами проводиться– визначення ймовірності та рівня наслідків ризиків. Для цього доцільно застосовувати математичні моделі. Наприклад, для оцінювання ймовірності настання ризику втрати компетентності від зниження вимог до якості надання освітніх послуг доцільно застосовувати модель наведену в джерелі [5].

Третій підблок – «Загальне оцінювання ризиків» – це кількісне оцінювання (1.3), що включає в себе:

крок дванадцять – кількісний аналіз ймовірностей та рівень наслідків ризиків;

крок тринадцять – документування результатів кількісного аналізу

На *чотирнадцятому кроці* виконується перевірка результатів кількісного аналізу. У разі, якщо результати є не достовірними відбувається перехід на *крок одинадцять*, у випадку, якщо інформація достовірна переходимо на *крок п'ятнадцять* – документування кінцевих результатів оцінювання та побудова класифікаторів типових ризиків.

Другий блок алгоритму включає процедури розроблення та впровадження заходів щодо мінімізації впливу ризиків [13]:

крок шістнадцять – розроблення та погодження плану дій;

крок сімнадцять – реалізації плану дій. Наприклад, для освітнього процесу план дій повинен включати:

– посилення контролю, шляхом введення додаткових перевірок розроблених чи діючих освітніх програм на відповідність до вимог чинної нормативної бази;

– розроблення та впровадження типових класифікаторів ризиків (для ефективного розроблення організаційно-технічних заходів ЗВО);

– моніторинг навчально-методичного забезпечення;

– проведення додаткових зрізів знань студентів;

– проведення процедур незалежного анкетування учасників освітнього процесу;

– проведення планового підвищення рівня кваліфікації науково-педагогічних працівників, проведення тематичних заходів, щодо обміну досвіду між науково-педагогічними працівниками (НПП), різних напрямів викладання;

– посилення контролю наукової діяльності НПП і студентів;

– заохочення студентів до наукової діяльності;

– введення реєстру відвідувань студентами та НПП занять;

– залучення спонсорів, меценатів для закупівлі нового сучасного обладнання;

– контроль за проведенням виховної роботи кураторами-тьюторами;

– друк дипломів у перевірених типографіях;

– проведення моніторингу набутих студентами компетентностей, відповідно до освітньої програми, тощо.

Завершальними кроками алгоритму *другого блоку* є розроблення і погодження плану дій (*крок шістнадцять*) та реалізація плану дій (*крок сімнадцять*).

Перевагою даного алгоритму є те, що він побудований з урахуванням принципів TQM та спрямований на постійне поліпшення, шляхом моніторингу і перевірки достовірності інформації за результатами оцінювання. Оскільки моніторинг є невід’ємною процедурою процесу управління ризиками та дозволяє проводити критичний аналіз обробки ризиків і перевіряти дієвість заходів щодо їх мінімізації.

Таким чином, застосування даного алгоритму дозволяє формалізувати процедури оцінювання ризиків освітнього процесу (освітньої програми) на всіх етапах його життєвого циклу та за результатами моніторингу розробляти організаційно-технічні заходи щодо їх зменшення з використанням інформаційних систем та технологій.

Для ефективного виконання усіх етапів алгоритму пропонується використовувати сучасне прикладне програмне забезпечення пакету MS Office (MS word, MS Excel, MS Access) та сучасні програмні продукти Mathcad, Matlab.

4.2 ЗАСТОСУВАННЯ КОМП’ЮТЕРНОЇ ПРОГРАМИ «ЕКСПЕРТИ ЗВО» ДЛЯ ФОРМУВАННЯ ЕКСПЕРТНИХ ГРУП

Для формування експертної групи для оцінювання ризиків освітнього процесу було розроблено програмний комплекс «Експерти ЗВО». Дана написана мовою програмування PHP (Hypertext Preprocessor) [14], з використанням Laravel framework. У програмному додатку були використані гіпертекстовий процесор HTML, каскадні таблиці стилів CSS та JavaScript.

За допомогою даної програми спочатку формується база даних експертів. Вона складається із загального (основного) списку та окремих спеціалізованих списків (які залежать від виду роботи, що буде виконуватись експертами) потенційних кандидатів у експерти. Для цього вводяться дані щодо ПІБ, наукового ступеня, вченого звання, галузі знань, спеціальності, стажу роботи, публікацій у різних наукометричних базах (статті у вітчизняних виданнях, Scopus / Web of science Core Collection (WOS), статті в іноземних виданнях), підручники / начальні посібники / монографії, участь в експертних групах. Надалі за даними створених списків визначають загальний коефіцієнт компетентності кожного з експертів. Від останнього залежить допуск фахівця до роботи в конкретній експертній групі. Дані спеціалізованих списків автоматично записуються до загального списку.

Особлива увага приділялась компетентності експертів [4]. Для підрахунку загального коефіцієнта компетентності в програмному коді використано систему професійно-орієнтованих критеріїв, які відображають специфіку освітньої діяльності. Наприклад, $К_1$ – кваліфікація за галуззю знань та спеціальністю, яка поділяється на два підкритерії: науковий ступінь – $К_{1.1}$ і вчене звання – $К_{1.2}$; $К_2$ – стаж роботи; $К_3$ – наявність

публікацій за напрямом оцінювання поділяється на чотири підкритерії: $К_{3.1}$ – статті у вітчизняних виданнях; $К_{3.2}$ – Scopus / Web of science Core Collection (WOS); $К_{3.3}$ – статті в іноземних виданнях; $К_{3.4}$ – підручники / начальні посібники / монографії; $К_4$ – досвід роботи в експертних групах.

Для визначення загального рівня компетентності було використано 10-ти бальну шкалу, яку спочатку було розбито на інтервали значущості кожного критерію. Потім для кожного критерію було проведено інтервальне розбиття на підкритерії, які в сумі становили відведену кількість балів (табл. 4.2).

Загальна компетентність окремого експерта обчислювалась за формулою (4.2):

$$К = \sum_{i=1}^{n} К_i \quad , \qquad (4.2)$$

де i – порядковий номер критерію;
n – кількість критеріїв.

Застосування програми дозволяє сформувати експертні групи за даними загального чи спеціалізованих списків. Формування групи проводиться в залежності від вище наведених професійно-орієнтованих критеріїв, з урахуванням вагових їх коефіцієнтів.

Для розширення функціональних можливостей в програмі передбачено функцію експорту створених груп в MS Excel та виводу їх на друк у вигляді готових таблиць.

Для роботи з програмним комплексом потрібно відкрити доступний браузер на комп'ютері та перейти за спеціальним актуальним посиланням http://volivach.co.ua. В результаті з'явиться веб-сторінка із назвою

«Експерти ЗВО», що містить інформацію щодо назви програми та кнопкою – «УВІЙТИ» (рис. 4.3).

Таблиця 4.2

Критерії для оцінки компетентності експертів

Критерії		Кількість балів	Характеристики відбору	Розподіл балів за шкалою оцінювання
$К_1$ – відповідна кваліфікація	$К_{1.1}$ – науковий ступінь	1,5	без ступеню	0
			кандидат наук / доктор філософії	0,5
			доктор наук	1
	$К_{1.2}$ – вчене звання	1,5	без звання	0
			доцент	0,5
			професор	1
$К_2$ – стаж роботи		1	до 5 років	0,2
			від 5 до 10 років	0,4
			від 10 до 20 років	0,6
			від 20 до 30 років	0,8
			більше 30 років	1
К3 – наявність публікацій за напрямом оцінювання (за останні 5 років)	$К_{3.1}$ – статті у вітчизняних виданнях	1	відсутні	0
			1	0,25
			2	0,5
			3	0,75
			4 і більше	1
	$К_{3.2}$ – Scopus / Web of science Core Collection (WOS)	1	відсутні	0
			1	0,25
			2	0,5
			3	0,75
			4 і більше	1
	$К_{3.3}$ – статті в іноземних виданнях	1	відсутні	0
			1	0,25
			2	0,5
			3	0,75
			4 і більше	1
	$К_{3.4}$ – підручники / начальні посібники / монографії	2	відсутні	0
			1	1
			2 і більше	2
К4 – досвід роботи в експертних групах		1	не приймав участі	0
			участь до 5 разів	0,5
			участь від 6 і більше	1

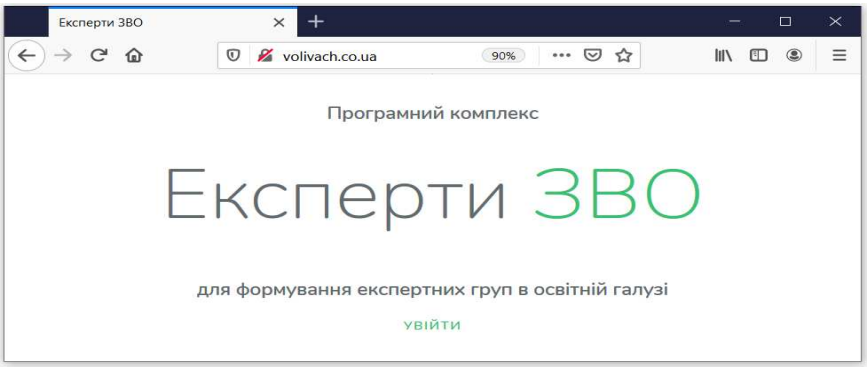


Рис. 4.3 – Вхід до програми

При натисканні на кнопку «УВІЙТИ» з’являється вікно авторизації (рис. 4.4).

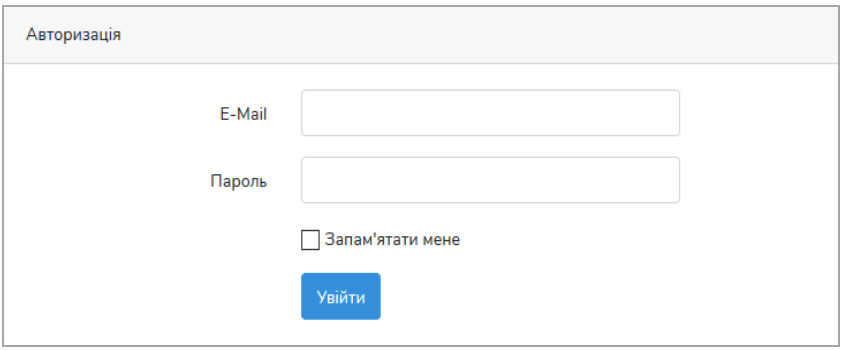


Рис. 4.4 – Вікно авторизації

Після авторизації відкривається веб-сторінка, що містить основні відомості та інструкцію до роботи програми (рис. 4.5).

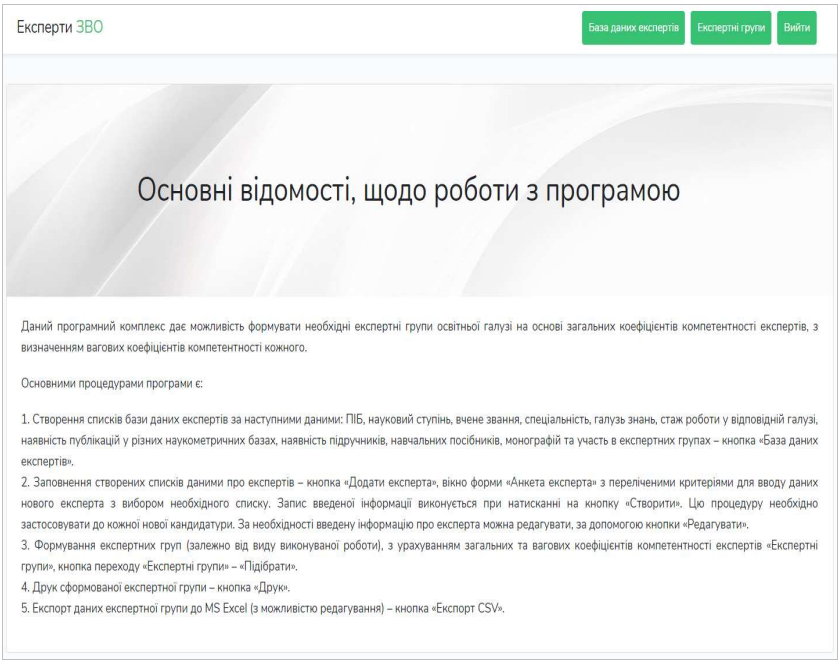


Рис. 4.5 – Сторінка з інструкцією по роботі з програмою

Першим кроком програми є заповнення загального списку бази даних експертів (рис. 4.6) за переліченими вище критеріями. Загальний список є основним джерелом даних про кандидатів до експертних груп.

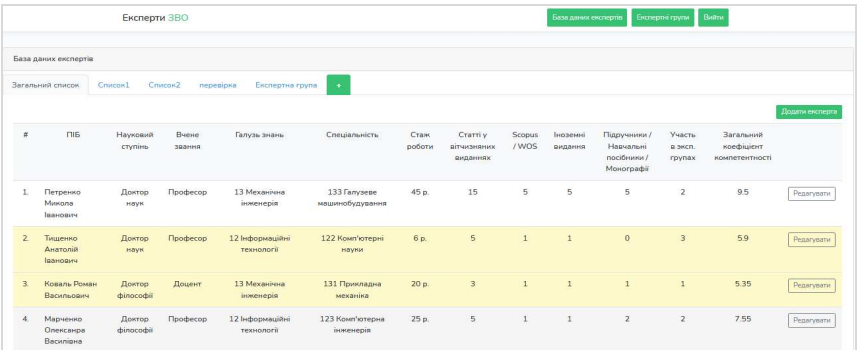


Рис. 4.6 – Вигляд заповненого загального списку кандидатів в експерти

В програмі у розділі «База даних експертів» передбачено створення окремих спеціалізованих списків, які створюються за допомогою кнопки +. При наведенні курсору миші на неї спливає підказка «Створення нового списку» (рис. 4.7). При натисканні на кнопку + в рядку автоматично додається текстове поле «Назва списку», яке є обов’язковим для заповнення, кнопкою «Додати» додається новий список з уведеною назвою. Подвійний «клік» по кнопці + приховує текстове поле вводу.

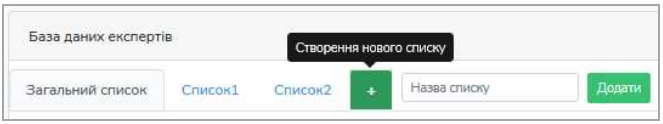


Рис. 4.7 – Створення нового списку

У випадку незаповнення поля назви з’явиться інформація Необхідно заповнити назву нового списку.

Видаляти спеціалізовані списки та корегувати їх назву можна за допомогою кнопки «Редагувати список», яка знаходиться на вкладці відкритого спеціалізованого списку бази даних експертів (рис. 4.8).

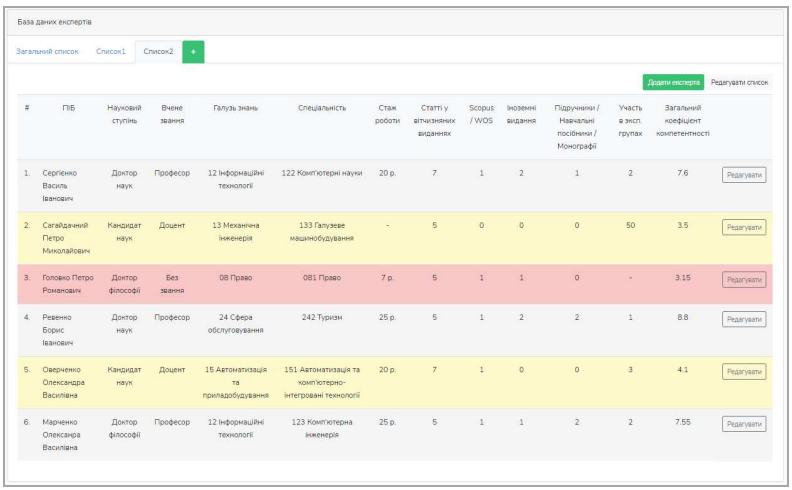


Рис. 4.8 – Вкладка спеціалізованого списку «Список2»

В результаті з’являється діалогове вікно «Редагування спеціалізованого списку» (рис. 4.9).

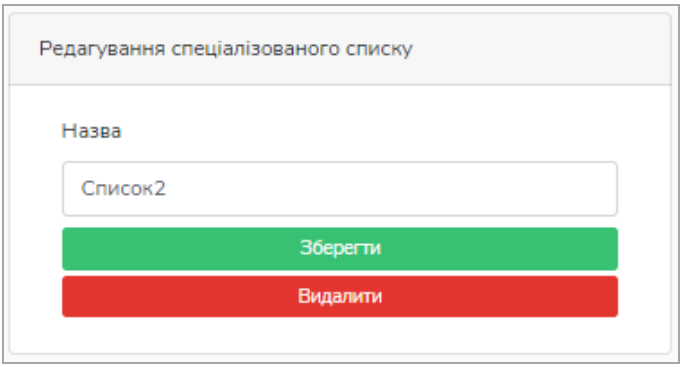


Рис. 4.9 – Редагування назви спеціалізованих списків

При натисканні кнопки «Видалити» з’являється діалогове вікно (рис. 4.10) з інформацією про попередження щодовидалення даного списку.

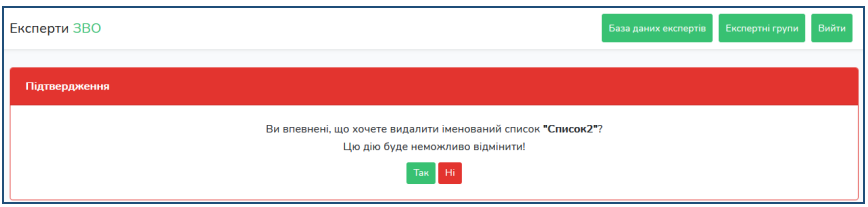
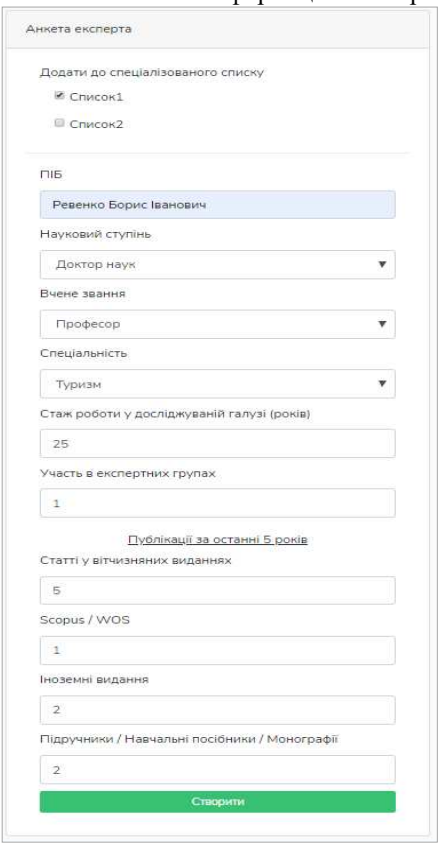


Рис. 4.10 – Підтвердження видалення вибраного списку

У разі підтвердження видалення виділеного списку його буде видалено з бази даних експертів. У разі помилкового вибору списку передбачено кнопку «Ні», яка дозволяє зробити відміну дії.

Заповнення інформацією створених списків бази даних експертів за



виконується при натисканні на кнопку Додати експерта (рис. 4.8). За допомого цієї кнопки відкривається діалогове вікно форми «Анкета експерта» (рис. 4.11) з передбаченою функцією вибору необхідного списку, переліченими критеріями та кнопкою «Створити».

Для заповнення спеціалізованих списків у діалоговому вікні «Анкета експерта» (рис. 4.11) передбачено функцію вибору необхідного списку (Список1, Список2) за потребою їх можна вибирати усі одночасно.

Рис. 4.11 – Діалогове вікно «Анкета експерта»

Спочатку в діалоговому вікні «Анкета експерта» вибирається необхідний спеціалізований список або декілька списків. У випадку, коли список не обирається інформація про експерта автоматично запишеться до загального списку. Далі заповнюється текстове поле ПІБ. Наступним кроком є вибір необхідної інформації щодо наукового ступеню, вченого звання та спеціальності зі списків, що розкриваються. Після вибору спеціальності в програмі передбачена функція, яка дозволяє за вибраною спеціальністю визначити шифр та галузь знань, до якої вона належить.

Далі в текстові поля: стаж роботи, участь в експертних групах та публікації за останні 5-ть років вводиться необхідна числова інформація. Після вводу необхідної інформації у вікні форми «Анкета експерта» (рис. 4.11) натискається кнопка «Створити».

В результаті введені дані про експерта запишуться до обраного списку експертної групи. В програмі передбачено, що інформація, яка вноситься до спеціалізованого списку, автоматично записується до загального списку.

Надалі, при заповнені створеної експертної групи, якщо дані про експерта із загального списку будуть задовольняти необхідні вимоги до створення нової експертної групи то вони автоматично будуть в неї внесені. За результатами відбору формуються експертні групи.

Приклад заповнення вибраного спеціалізованого списку «Список1» наведено на (рис. 4.12).

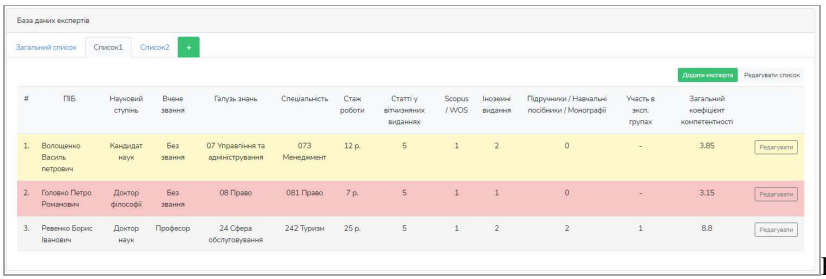


Рис. 4.12 – Результат заповнення спеціалізованого списку «Список1»

Однією з функціональних можливостей є перенесення інформації з одного спеціалізованого списку до іншого. Після заповнення списків інформацією у них автоматично розраховується поле «Загальний коефіцієнт компетентності» – S_{Ki} (рис. 4.6, рис. 4.8, рис. 4.12) від значення величини якого залежить допуск даного експерта до проведення конкретного виду роботи.

Загальний коефіцієнт компетентності експерта S_{Ki} обчислюється за методикою наведеною у джерелах [15] формулою (4.3).

$$S_{Ki} = \sum_{i=1}^{n} K_i, \qquad (4.3)$$

де K_i – професійно-орієнтовані критерії;

n – кількість професійно-орієнтованих критеріїв.

Даний коефіцієнт компетентності потрібен для визначення вагомості кожного експерта, який входить в кінцеву сформовану робочу групу. Ваговий коефіцієнт компетентності експерта залежить від сумарного значення загального коефіцієнта компетентності експертної групи і обчислюється за формулою (4.4):

$$\omega_{kei} = \frac{S_{Ki}}{\sum_{i=1}^{m} S_{Ki}}, \qquad (4.4)$$

де S_{Ki} – загальний коефіцієнт компетентності окремого експерта;

$\sum_{i=1}^{m} S_{Ki}$ – сумарне значення загальних коефіцієнтів (за професійно-орієнтованими критеріями) експертної групи; m – кількість експертів.

Для візуального визначення компетентності кандидата в експерти у списках застосовано кольорову гаму. Червоний колір – низька компетентність ($S_{Ki} < 3{,}5$) експерт не допускається до роботи; світло помаранчевий колір – допустима компетентність ($3{,}5 \le S_{Ki} \le 6{,}9$); світло сірий – достатня компетентність ($7 \le S_{Ki} \le 8{,}9$); білий колір – висока компетентність ($9 \le S_{Ki} \le 10$). Експерти, які мають загальну компетентність в межах $3{,}5 \le S_{Ki} \le 10$ – допускаються до роботи.

Для редагування даних експерта в усіх записах існуючих списків передбачена кнопка «Редагувати» (рис. 4.6, рис. 4.8, рис. 4.11), що відкриває діалогове вікно форми «Редагування анкети експерта» попередньо заповнене інформацією про вибраного експерта. Корегована інформація зберігається за допомого кнопки «Зберегти» (в усіх списках, в

яких вона була). Видалення запису про експерта (з усіх існуючих списків) виконується кнопкою «Видалити».

Наступним кроком є створення та формування конкретних експертних груп, яке відбувається при переході до діалогового вікна «Експертні групи» при натисканні на кнопку Експертні групи на сторінці «Експерти ЗВО» (рис. 4.13).

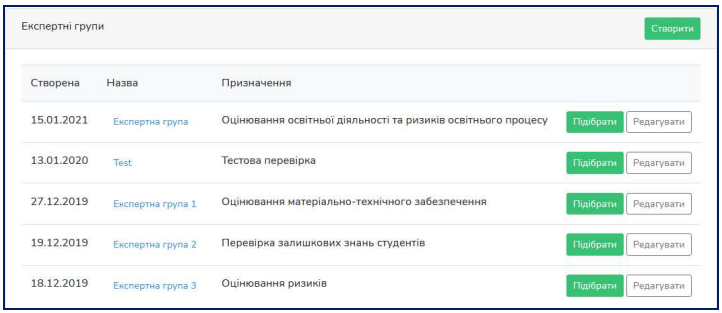


Рис. 4.13 – Створення експертних груп та їх формування

Для створення назви робочої групи та запису її призначення (рис. 4.14) використовується кнопка «Створити» (рис. 4.13). Дана кнопка викликає діалогове вікно (рис. 4.14) «Створення експертної групи», де обов'язковою умовою є заповнення інформацією наступних полів: «Назва експертної групи» та «Призначення». Збереження інформації відбувається за допомогою кнопки «Створити».

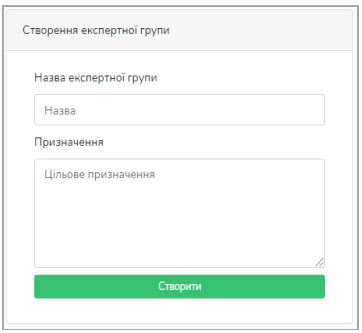


Рис. 4.14 – Діалогове вікно створення експертної групи

У діалоговому вікні «Експертні групи» (рис. 4.13) відкривається перелік дати створення, назви та призначення уже створених груп. При натисканні на назву групи відкривається список відібраних експертів (за певними критеріями) з визначеними ваговими коефіцієнтами компетентності кожного з них. Сумарне значення вагових коефіцієнтів дорівнює одиниці.

Для формування списку експертної групи, з існуючих (рис. 4.13) обирається необхідна, наприклад «Експертна група 1». За допомогою кнопки «Підібрати» (рис. 4.15) викликається вікно форми «Підбір групи експерта» (рис. 4.16).

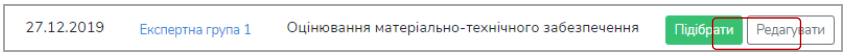


Рис. 4.15 – Приклад підбіру експертів в експертну групу

У діалоговому вікні «Підбір групи експерта» (рис. 4.16) встановлюються необхідні критерії та за допомогою кнопки «Виконати підбір» формується конкретна експертна група.

В наведеному прикладі експерти підбирались за наступними критеріями: із загального списку, кількістю сім чоловік, стаж роботи – від п’яти років, серед спеціальностей було вибрано: 151 Автоматизація та комп’ютерно-інтегровані технології, 131 Прикладна механіка, 133 Галузеве машинобудування, 122 Комп’ютерні науки, 152 Метрологія та інформаційно-вимірювальна техніка; науковий ступінь: кандидат наук, доктор філософії, доктор наук; вчене звання: доцент, професор.

Приклад результату сформованої робочої групи наведено на (рис. 4.17). Як видно з рисунку підбір кандидатів відповідає вказаним вимогам відбору діалогового вікна «Підбір групи експертів» (рис. 4.16). При цьому сформований список експертної групи розташований за принципом зменшення (від більшого до меншого) вагомості компетентності експерта (у даному випадку він знаходится в інтервалі від 9,5 до 4,9).

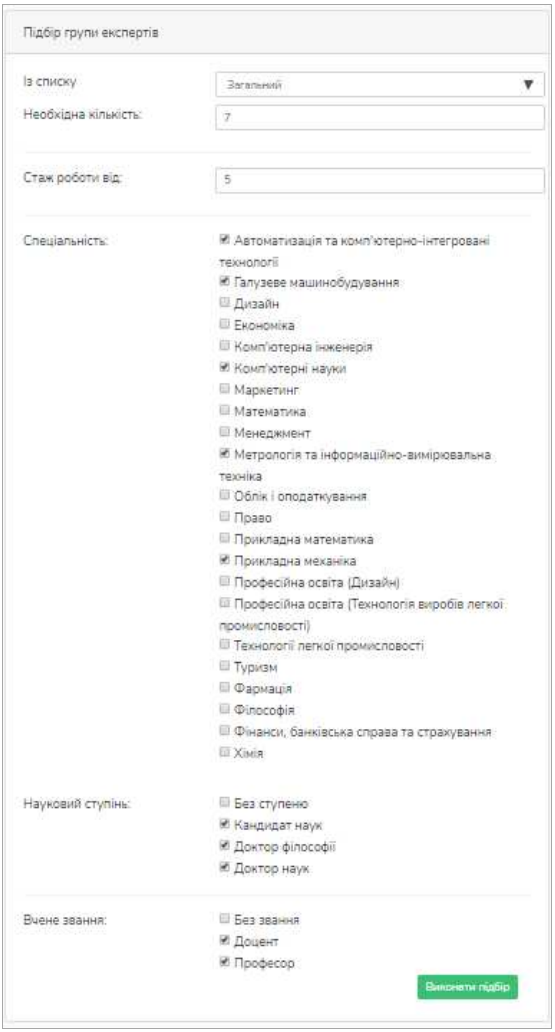
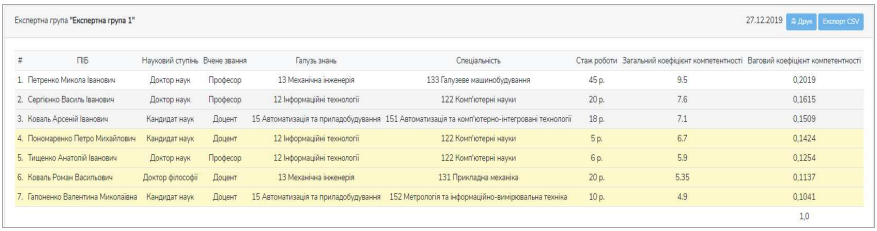


Рис. 4.16 – Вибір необхідних критеріїв для робочої групи



Експертна група "Експертна група 1" 27.12.2019 Друк Експорт CSV

#	ПІБ	Науковий ступінь	Вчене звання	Галузь знань	Спеціальність	Стаж роботи	Загальний коефіцієнт компетентності	Ваговий коефіцієнт компетентності
1.	Петренко Микола Іванович	Доктор наук	Професор	13 Механічна інженерія	133 Галузеве машинобудування	45 р.	9.5	0,2019
2.	Сергієнко Василь Іванович	Доктор наук	Професор	12 Інформаційні технології	122 Комп'ютерні науки	20 р.	7.6	0,1615
3.	Коваль Арсеній Іванович	Кандидат наук	Доцент	15 Автоматизація та приладобудування	151 Автоматизація та комп'ютерно-інтегровані технології	18 р.	7.1	0,1509
4.	Пономаренко Петро Михайлович	Кандидат наук	Доцент	12 Інформаційні технології	122 Комп'ютерні науки	5 р.	6.7	0,1424
5.	Тищенко Анатолій Іванович	Доктор наук	Професор	12 Інформаційні технології	122 Комп'ютерні науки	6 р.	5.9	0,1254
6.	Коваль Роман Васильович	Доктор філософії	Доцент	13 Механічна інженерія	131 Прикладна механіка	20 р.	5.35	0,1137
7.	Гапоненко Валентина Миколаївна	Кандидат наук	Доцент	15 Автоматизація та приладобудування	152 Метрологія та інформаційно-вимірювальна техніка	10 р.	4.9	0,1041
								1,0

Рис. 4.17 – Результат створеної експертної групи

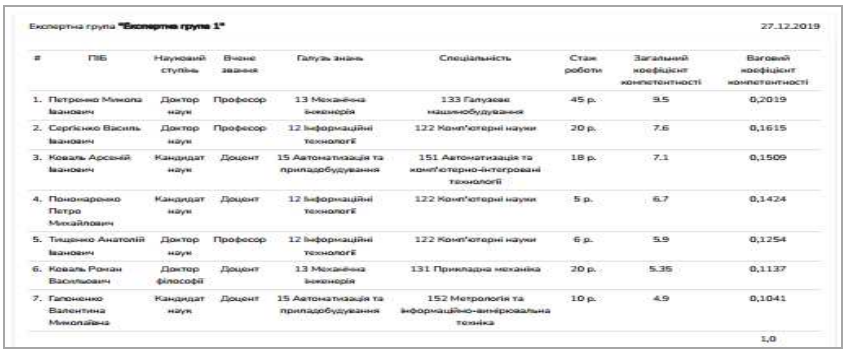
Крім того, це доволяє отримати додаткову інформацію про експерта. Зокрема, науковий ступінь, приналежність експерта до галузі знань та спеціальності, стаж роботи, загальний коефіцієнт компетентності та ваговий коефіцієнт компетентності експерта. Такий підхід дозволяє більш точно виконувати підбір експертів, для конкретного виду роботи.

У випадку, якщо за встановленими критеріями відсутні певні кандидати, то сформована група буде складатися з меншої кількості експертів, що вказує на необхідність добору додаткових кандидатів до бази даних експертів.

В діалоговому вікні створеної експертної групи передбачено кнопки

(рис. 4.17).

Кнопка «Друк» – дає можливість друкувати сформовану таблицю експертної групи у вигляді наведеному на рис. 4.18.



Експертна група **"Експертна група 1"** 27.12.2019

#	ПІБ	Науковий ступінь	Вчене звання	Галузь знань	Спеціальність	Стаж роботи	Загальний коефіцієнт компетентності	Ваговий коефіцієнт компетентності
1.	Петренко Микола Іванович	Доктор наук	Професор	13 Механічна інженерія	133 Галузеве машинобудування	45 р.	9.5	0,2019
2.	Сергієнко Василь Іванович	Доктор наук	Професор	12 Інформаційні технології	122 Комп'ютерні науки	20 р.	7.6	0,1615
3.	Коваль Арсеній Іванович	Кандидат наук	Доцент	15 Автоматизація та приладобудування	151 Автоматизація та комп'ютерно-інтегровані технології	18 р.	7.1	0,1509
4.	Пономаренко Петро Михайлович	Кандидат наук	Доцент	12 Інформаційні технології	122 Комп'ютерні науки	5 р.	6.7	0,1424
5.	Тищенко Анатолій Іванович	Доктор наук	Професор	12 Інформаційні технології	122 Комп'ютерні науки	6 р.	5.9	0,1254
6.	Коваль Роман Васильович	Доктор філософії	Доцент	13 Механічна інженерія	131 Прикладна механіка	20 р.	5.35	0,1137
7.	Гапоненко Валентина Миколаївна	Кандидат наук	Доцент	15 Автоматизація та приладобудування	152 Метрологія та інформаційно-вимірювальна техніка	10 р.	4.9	0,1041
								1,0

Рис. 4.18 – Результат виведення сформованої групи на друк

Для розширення функціональних можливостей кожну створену експертну групу (у форматі CSV) можна експортувати до MS Excel (з можливістю редагування), при натисканні на кнопку «Експорт CSV».

Редагувати назву та призначення експертної групи дозволяє кнопка «Редагувати» (рис. 4.15). Дана функція передбачає як збереження корегованої групи, так і її видалення, якщо вона надалі не буде використовуватись. В процесі видалення експертної групи програма видає запит на підтвердження видалення вибраної групи (рис. 4.19), що є доречним, у разі помилкового натискання на кнопку «Видалити».

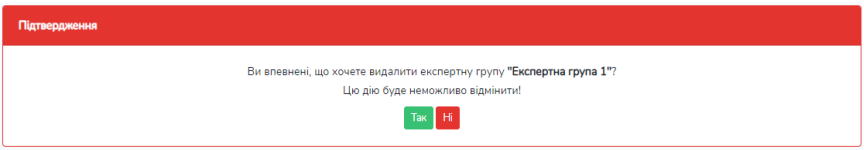


Рис. 4.19 – Запит на підтвердження видалення вибраної групи

Таким чином, наведені вище методичні рекомендації дозволяють використовувати дану програму під час формування експертних груп, для виконання робіт пов’язаних з оцінюванням якості освітньої діяльності закладів вищої освіти. Зокрема для визначення рівня досконалості освітньої програми та визначення ймовірності настання ризику втрати компетентності від зниження вимог до якості надання освітніх послуг.

4.3 РЕГРЕСІЙНА МОДЕЛЬ ОЦІНЮВАННЯ РІВНЯ ЯКОСТІ ОСВІТНЬОЇ ПРОГРАМИ

Освітня програма, за якою здійснюється підготовка здобувачів вищої освіти різних рівнів, спрямована на досягнення заявлених в ній результатів навчання та поєднує між собою усі суб’єкти освітньої діяльності. Від якості її функціонування залежить конкурентоспроможність та затребуваність майбутніх фахівців у стейкхолдерів та ринку праці. Проте, для її оцінювання і модернізації за вимогами стейкхолдерів потрібно мати спеціальний інструментарій.

В ході досліджень, було встановлено, що оцінювання ОП згідно Положення [16] проводиться за десятьма критеріями. При цьому дев'ять критеріїв стосуються першого бакалаврського і другого магістерського рівнів вищої освіти, а десятий передбачений лише для третього науково-освітнього рівня. Всі вище наведені критерії мають описовий характер, оцінюються за чотирирівневою шкалою оцінювання рівня якості (A, B, E та F) і тому не дозволяють отримати кількісну оцінку. Система оцінювання рівня освітньої програми, за вище наведеними критеріями представлена в табл. 4.3.

Таблиця 4.3

Система оцінювання критеріїв

№ п/п	Рівень оцінювання критеріїв	Умови оцінювання критеріїв
1	Рівень A	ОП має інноваційний характер й у повній мірі відповідає якісним характеристикам критерію
2	Рівень B	ОП має недоліки, які не суттєво впливають на кінцеву якість хоча б одного з критеріїв
3	Рівень E	ОП не відповідає якісним характеристикам критерію (дані недоліки будуть усунені протягом 1-го року)
4	Рівень F	ОП має вагомі недоліки, які суттєво знижують її якість і не можуть бути усунені протягом 1-го року

Відповідно до отриманого рівня оцінювання критеріїв (табл. 4.3), Національне агентство, за результатами експертних висновків галузевої експертної ради (ГЕР) приймає рішення щодо терміну дії сертифікату освітньої програми, за якою проводилась акредитація (табл. 4.4).

Таблиця 4.4

Термін дії акредитації ОП

№ п/п	Рішення Національного агентства щодо акредитації ОП	Умови ухвалення акредитації ОП	Термін дії сертифікату
1	«зразкова»	кількість критеріїв за рівнем «А» – від п'яти і більше	5ть років (за умови, якщо сертифікат отриманий під час вперше проведеної акредитації); 10 років (за умови, якщо
2	«акредитована»	кількість критеріїв за рівнем «А» – менше п'яти, решта за рівнем - «В» або усі критерії	

		відповідають рівню «B»	сертифікат отриманий під час другої та наступних акредитацій).
3	«відкладена»	один або два критерії відповідають рівню «E», а інші не мають жодного рівня «F»	1 рік.
4	«відмовлена»	кількість критеріїв за рівнем «E» від трьох і більше або один з критеріїв має рівень «F»	сертифікат не видається але ЗВО має право подати документи на акредитацію цієї ОП наступного навчального року.

Акредитація освітньої програми проходить в три етапи:

– *перший етап* – це оцінювання ОП експертними групами;

– *другий етап* – це формування висновків ГЕР, на основі звіту експертної групи щодо відповідності ОП критеріям оцінювання якості і надання пропозиції Національному агентству про вид акредитації ОП;

– *третій етап* – прийняття кінцевого рішення щодо акредитації ОП та терміну дії сертифікату Національним агентством.

Інформація про термін дії сертифікату щодо ухвалення рішення вноситься до Єдиної державної електронної бази з питань освіти (ЄДЕБО) [17]. Слід зазначити, що у разі подачі закладом вищої освіти недостовірної інформації Національне агентство має право відмовити в акредитації освітньої програми незалежно від отриманого нею рівня.

Для перевірки повноти й достатності якісних оцінок (які отримують освітні програми (за 9-ма критеріями) в ході проведення акредитації) за даними Національного агентства [18, 19, 20] було розроблено по-крокову схему (рис. 4.20). Дана схема складається з 5-ти етапів. Це дозволило перевести якісні характеристики оцінювання в кількісні.

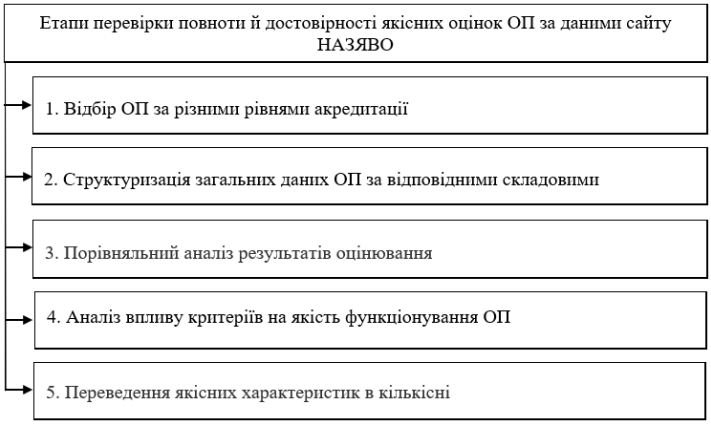


Рис. 4.20 – Схема етапів перевірки повноти і достовірності якісних оцінок

На *першому етапі* згідно даних реєстру акредитаційних справ освітніх програм за даними 2019/2020 навчального року було відібрано 65 освітніх програм другого магістерського рівня вищої освіти, які мали різні рівні акредитації: «зразкова», «акредитована», «відкладена» та «відмовлена» [20].

На *другому етапі* було проведено їх структуризацію за такими складовими, як:

– шифр освітньої програми;
– назва освітньої програми;
– галузь знань;
– спеціальність;
– заклад вищої освіти;
– № акредитаційної справи;
– рішення Національного агентства.

Фрагмент результатів структурного аналізу наведено у табл. 4.5.

Таблиця 4.5

Фрагмент структурного аналізу даних 65 освітніх програм

Шифр	Назва ОП	Галузь знань	Спеціал	ЗВО	№	Рішення

ОП			ьність		акредитаційної справи	Національного агентства
ОП1	Прикладне матеріалознавство	13 Механічна інженерія	132 Матеріалознавство	Державний вищий навчальний заклад «Придніпровська державна академія будівництва та архітектури»	А-19-0178-154	«акредитована»
…	…	…	…	…	…	…
ОП25	Системи, технології і комп`ютерні засоби мультимедіа	17 Електроніка та телекомунікації	171 Електроніка	Харківський національний університет радіоелектроніки	А-19-0021-19	«акредитована»
…	…	…	…	…	…	…
ОП65	Дизайн	02 Культура і мистецтво	022 Дизайн	Харківський національний педагогічний університет ім. Г.С. Сковороди	А-19-0024-229	«відкладена»

Аналіз результатів даних табл. 4.5 показав, що дані освітні програми функціонують в 49 закладах вищої освіти різних регіонів України, мають різні форми власності (державна, приватна) та охоплюють 31 спеціальність в рамках 14-ти діючих галузей знань.

На *третьому етапі* для зручності кожному з 9-ти критеріїв якості ОП [21] було присвоєно наступне скорочене позначення:

– проектування та цілі освітньої програми – ПЦ;

– структура та зміст освітньої програми – СЗ;

– доступ до освітньої програми та визначення результатів навчання – ДВРН;

– навчання і викладання за освітньої програми – НВ;

– контрольні заходи, оцінювання здобувачів освітньої програми та академічна доброчесність – КЗАД;

– людські ресурси – ЛР;

– освітнє середовище та матеріальні ресурси – ОСМР;

– внутрішнє забезпечення якості освітньої програми – ВЗЯ;

– прозорість та публічність – ПП.

Далі було проведено порівняльний аналіз результатів якісних характеристик 9-ти критеріїв якості 65-ти освітніх програм звітів експертних груп із результатами експертних висновків галузевої експертної ради [19].

Фрагмент результатів порівняльного аналізу наведено у табл. 4.6.

Таблиця 4.6

Фрагмент результатів якісного аналізу оцінювання освітніх програм за 9-ма критеріями

Освітня програма	Якісні характеристики	Критерії оцінювання ОП									Рішення Національного агентства
		ПЦ	СЗ	ДВРН	НВ	КЗАД	ЛР	ОСМР	ВЗЯ	ПП	
ОП1	Звіт експертної групи	В	В	А	В	А	В	В	В	А	«акредитована»
	Експертні висновки ГЕР	В	В	А	В	А	В	В	В	А	
…	…	…	…	…	…	…	…	…	…	…	…
ОП25	Звіт експертної групи	В	В	В	В	В	В	В	А	В	«акредитована»
	Експертні висновки ГЕР	В	В	В	В	В	В	В	А	В	
…	…	…	…	…	…	…	…	…	…	…	…
ОП65	Звіт експертної групи	Е	Е	В	В	В	В	В	В	В	«відкладена»
	Експертні висновки ГЕР	Е	Е	В	В	В	В	В	В	В	

Результати аналізу даних (табл. 4.6) показав, що 58% освітніх програм (з 65) мали розходження в результатах експертного оцінювання та висновків ГЕР. Ці розходження стосувались рівнів оцінювання критеріїв, як в сторону їх зменшення так і в сторону їх збільшення. Наприклад, в ОП5 не співпали рівні оцінювання 2-го та 6-го критеріїв; в ОП7 – 3-го, 5-го та 7-го критеріїв; в ОП8 – 5-го та 9-го критеріїв; в ОП13 – 1-го, 5-го, 7-го та 8-го критеріїв; в ОП21 – 1-го, 2-го, 5-го, 8-го та 9-го критеріїв.

Також було встановлено, що серед 65-ти досліджуваних освітніх програм, 4 освітні програми мають рівень акредитації – «зразкова»; 31 освітня програма має рівень акредитації – «акредитована»; 23 освітні програми мають рівень акредитації – «відкладена» та 7 освітніх програм мають рівень акредитації – «відмовлена».

Враховуючи вище наведене, для уточнення достовірності й повноти результатів оцінювання освітніх програм було запропоновано застосовувати додатковий інструментарій.

На *четвертому етапі* було визначено вплив критеріїв на якість функціонування освітньої програми. Для цього за допомогою методу порівняльного аналізу було проведено структурування 65-ти освітніх програм за якісною характеристикою кожного з 9-ти критеріїв (за результатами оцінок експертних груп та висновками ГЕР). Результати аналізу зведено в табл. 4.7.

Таблиця 4.7

Рівні відповідності критеріїв освітніх програм

Повна назва критеріїв	Скорочена назва критеріїв	Всього ОП	Оцінки експертних груп			Висновки ГЕР		
			A (Експ)	B (Експ)	E (Експ)	A (ГЕР)	B (ГЕР)	E (ГЕР)
1. Проектування та цілі ОП	ПЦ	65	10	51	4	10	46	9
2. Структура та зміст ОП	СЗ	65	4	43	18	1	36	28
3. Доступ до ОП та визначення результатів навчання	ДВРН	65	13	52	0	12	53	0
4. Навчання і викладання за освітньою програмою	НВ	65	12	51	2	10	49	6
5. Контрольні заходи, оцінювання здобувачів вищої освіти та академічна доброчесність	КЗ	65	19	43	3	12	50	3
6. Людські ресурси	ЛР	65	8	53	4	4	53	8
7. Освітнє середовище та матеріальні ресурси	ОСМР	65	14	50	1	11	54	0
8. Внутрішнє забезпечення якості ОП	ВЗЯ	65	19	41	5	14	44	7
9. Прозорість та публічність	ПП	65	17	48	0	8	54	3

Як видно з табл. 4.7 однією з «слабких» сторін освітніх програм виявився структурний розділ ОП, який відповідає критерію 2 – «Структура

та зміст ОП». Оскільки даний критерій набрав найбільшу кількість оцінок рівня Е (18), і найменшу кількість оцінок рівня А (4). Також, за результатами аналізу було встановлено, що з 585 критеріїв оцінених експертними групами 510 критеріїв мали повний збіг з оцінками висновків ГЕР, інші 75 критеріїв мали розходження в оцінюванні. Це говорить проте, що 13% якісних оцінок від загальної кількості (585) мали розходження у результатах. Загалом зміни оцінок були внесені ГЕР до 38 освітніх програм (що становить 58% з 65 досліджуваних ОП). Так, наприклад, галузевою експертною радою було переведено результати оцінок експертних груп в 35-ти критеріях з рівня А на рівень В, в 2-х критеріях з рівня А на рівень Е, в 3-х критеріях з рівня В на рівень А, в 30-ти критеріях з рівня В на рівень Е та в 5-ти критеріях з рівня В на рівень Е [22].

За результатами аналізу було встановлено вплив кожного з критеріїв на якість функціонування ОП, тобто визначено її «сильні» та «слабкі» сторони. Проте, результати аналізу показують, що наявність лише якісних характеристик не дозволяє отримати достовірну інформацію щодо оцінювання рівнів відповідності якості освітніх програм. Це в свою чергу потребує розроблення додаткових процедур пов'язаних з кількісним оцінюванням.

Такий підхід дозволяє більш точно і достовірно оцінити будь-який процес, у тому числі й освітній, який забезпечується певною освітньою програмою.

Тому на *п'ятому етапі* для переводу якісних характеристик критеріїв освітніх програм в їх кількісні значення було розроблено описову інтервальну шкалу (табл. 4.8).

Таблиця 4.8

Описова інтервальна шкала

Якісна оцінка	Кількісна оцінка в балах	Визначення	Пояснення
А	3,76 – 5	високий рівень	повна відповідність, що має

		відповідності	інноваційний характер
B	2,51 – 3,75	достатній рівень відповідності	часткова невідповідність, недоліки мають несуттєвий характер
E	1,26 – 2,5	низький рівень відповідності	невідповідність, з можливістю усунення протягом року
F	0 – 1,25	повна невідповідність	невідповідність, без можливості усунення

В основу побудови шкали було покладено принцип рівномірного розподілу бальних значень інтервалу від 0 до 5-ти, з кроком 1,25 (табл. 2.6). Суть оцінювання критеріїв якості освітньої програми за даною шкалою полягає в тому, що якісні (буквені) характеристики ОП за допомогою 5-ти бальної шкали переводяться в кількісні значення. Як видно з табл. 2.6 шкала складається з чотирьох інтервалів: високий рівень відповідності – А (в межах від 3,76 до 5); достатній рівень відповідності – В (в межах 2,51 – 3,75); низький рівень відповідності – Е (в межах 1,26 – 2,5); повна невідповідність – F (в межах від 0 до 1,25).

Згідно Положення [16] заключним висновком щодо рішення НАЗЯВО про визначення рівня акредитації освітніх програм є результати висновків ГЕР. В роботі за допомогою описової інтервальної шкали результати висновків ГЕР було переведено в кількісні оцінки, за кожним з 9 ти критеріїв для усіх 65-ти освітніх програм. Такий підхід дозволив більш точно оцінити кожен з них. Фрагмент зведених результатів переводу якісних оцінок в кількісні наведено в табл. 4.9.

Таблиця 4.9

Фрагмент зведених результатів переводу якісних оцінок в кількісні

Освітня програма	Критерії оцінювання ОП	Рівень відповідності (ГЕР)	Кількісна оцінка	Кількість рівнів відповідності	Рішення Національного агентства
ОП1	ПЦ	В	3,5	3А, 6В	«акредитована»
	СЗ	В	3		

Продовження таблиці 4.9

	ДВРН	А	5		
	НВ	В	3,2		
	КЗАД	А	5		
	ЛР	В	3,5		
	ОСМР	В	3,2		
	ВЗЯ	В	3,5		

	ПП	А	5		
…	…	…	…	…	
ОП25	ПЦ	В	3,65	1А, 8В	«акредитована»
	СЗ	В	3,65		
	ДВРН	В	3,4		
	НВ	В	3,75		
	КЗАД	В	3,5		
	ЛР	В	3,75		
	ОСМР	В	3,7		
	ВЗЯ	А	5		
	ПП	В	3,5		
…	…	…	…	…	…
ОП65	ПЦ	Е	1,3	7В, 2Е	«відкладена»
	СЗ	Е	1,3		
	ДВРН	В	3,2		
	НВ	В	3,35		
	КЗАД	В	3,68		
	ЛР	В	3,6		
	ОСМР	В	3,4		
	ВЗЯ	В	3,7		
	ПП	В	3,6		

Таким чином, проведені дослідження дозволили виявити різницю в акредитації освітньої програми, яка була проведена за якісними та кількісними характеристиками. Як показано вище 58% освітніх програм мають розходження в оцінках критеріїв отриманих за допомогою якісного і кількісного методу. Тому в роботі для підвищення точності й достовірності оцінювання рівня відповідності освітніх програм було запропоновано застосовувати математичний апарат у вигляді багатофакторних моделей на базі застосування методів регресійного аналізу.

Побудова будь-якої моделі являє собою розв'язання складної формалізованої задачі, за допомогою наявних математичних методів і засобів. Проте, слід зазначити, що допущені в процесі формалізації помилки майже не можливо виправити під час обробки результатів досліджень. Це пов'язано з тим, що формалізація передбачає переведення мети досліджень у математичний вигляд та застосування до нього статистичних методів аналізу. Для побудови математичної моделі щодо оцінювання рівня якості освітньої програми була розроблена спеціальна схема (рис. 4.21).

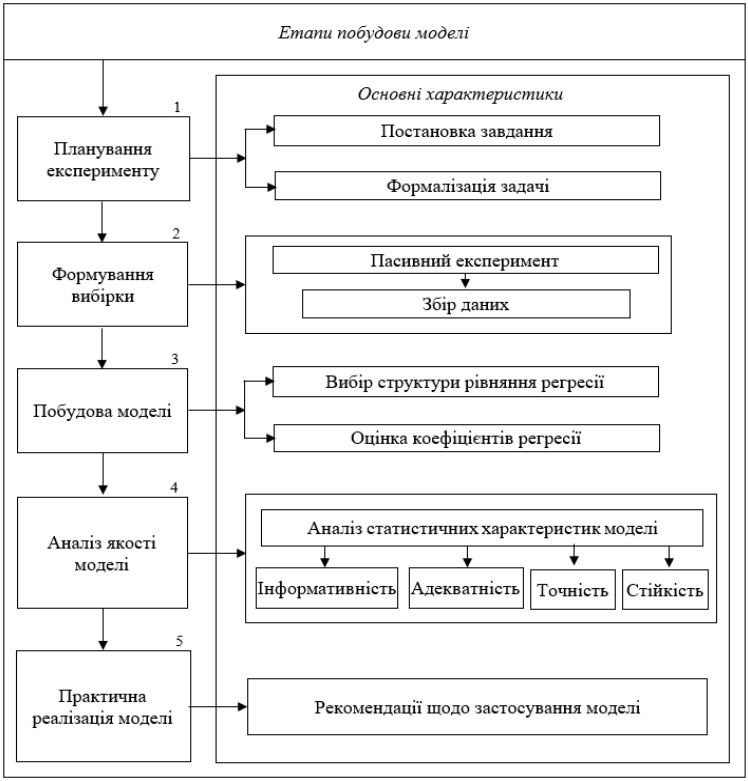


Рис. 4.21 – Схема побудови математичної моделі оцінювання рівня якості освітньої програми

Як видно з рисунку побудова та аналіз її якості моделі складається з 5-ти етапів:

– планування експерименту;
– формування вибірки;
– побудова моделі;
– аналіз якості моделі та практична реалізація моделі.

В свою чергу, кожен з етапів відповідає за окремі складові моделі та дозволяє виявити «слабкі» й «сильні» сторони освітньої програми.

На *першому етапі – «Планування експерименту»* (рис. 4.21) було визначено завдання, які потрібно вирішити в ході досліджень. До них було віднесено:

1. Проведення аналізу висновків галузевої експертної ради (ГЕР) щодо можливості акредитації ОП за встановленою чотирирівневою якісною шкалою (A, B, E та F).

2. Розроблення додаткової шкали для визначення інтервалів рівня акредитації освітніх програм.

3. Формування вихідної матриці для кількісного оцінювання критеріїв якості ОП для побудови математичної моделі.

На *другому етапі – «Формування вибірки»*, за даними звітів ГЕР сайту НАЗЯВО [20] було сформовано робочу матрицю, фрагмент результатів якої наведено вище в табл. 4.9.

Дана матриця отримана в результаті переводу якісних характеристик оцінювання якості освітніх програм в кількісні, за 9-ма критеріями (табл. 4.10).

Таблиця 4.10

Фрагмент результатів якісного аналізу оцінювання ОП за 9-ма критеріями

Освітня програма	Якісні характеристики	Критерії оцінювання ОП									Рішення Національного агентства
		ПЦ	СЗ	ДВРН	НВ	КЗАД	ЛР	ОСМР	ВЗЯ	ПП	
ОП1	Звіт експертної групи	B	B	A	B	A	B	B	B	A	«акредитована»
	Експертні висновки ГЕР	B	B	A	B	A	B	B	B	A	
ОП2	Звіт експертної групи	B	E	B	B	B	A	B	B	B	«відкладена»
	Експертні висновки ГЕР	B	E	B	E	B	B	B	B	B	
..	…	…	…	…	…	…	…	…	…	…	…

Продовження таблиці 4.10

ОП34	Звіт експертної групи	А	А	В	А	А	В	А	А	А	«акредитована»
	Експертні висновки ГЕР	В	В	В	А	А	В	А	А	В	
…	…	…	…	…	…	…	…	…	…	…	…
ОП64	Звіт експертної групи	В	В	В	В	В	В	А	В	В	«відкладена»
	Експертні висновки ГЕР	Е	Е	В	В	В	В	В	В	В	
ОП65	Звіт експертної групи	Е	Е	В	В	В	В	В	В	В	«відкладена»
	Експертні висновки ГЕР	Е	Е	В	В	В	В	В	В	В	

Для побудови математичної моделі запропоновано використовувати метод регресійного аналізу [23]. Для визначення відгуку Y було розроблено додаткову шкалу. Принцип побудови даної шкали полягав в застосуванні максимальної кількісної оцінки рівнів акредитації описової інтервальної шкали (див. табл. 4.8). Додаткова інтервальна шкала наведена в табл. 4.11.

Таблиця 4.11

Додаткова інтервальна шкала для визначення рівня акредитації

Y	«Зразкова»	«Акредитована»	«Відкладена»	«Відмовлена»
	5	3,75	2,5	1,25

Фрагмент загального вигляду матриці для побудови моделі наведено у табл. 4.12.

Таблиця 4.12

Фрагмент матриці для побудови регресійної моделі

Освітні програми	Фактори									Y	Рішення Національного агентства [54]
	Натуральне визначення критеріїв										
	ПЦ	СЗ	ДВРН	НВ	КЗ	ЛР	ОСМР	ВЗЯ	ПП		
	Формалізоване позначення критеріїв (регресорів) та їх чисельні значення										
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9		
ОП1	3,5	3	5	3,2	5	3,5	3,2	3,5	5	3,75	«акредитована»
ОП2	3	2	3,4	1,6	3,5	2,8	3	3,2	3	2,5	«відкладена»
ОП3	3,4	3,7	3,4	3,6	3,5	3,2	5	3,4	3,5	3,75	«акредитована»

Продовження таблиці 4.12

…	…	…	…	…	…	…	…	…	…	…	…
ОП16	5	3,7	5	3,6	5	3,65	5	5	3,65	5	«зразкова»
ОП17	5	3,65	4,8	5	5	5	3,72	3,55	3,45	5	«зразкова»
ОП18	1,7	1,45	3,2	1,35	1,35	1,43	3,5	3,4	3,45	1,25	«відмовлена»
…	…	…	…	…	…	…	…	…	…	…	…
ОП35	3,15	3,3	5	3,69	3,7	3,6	3,55	5	4,85	3,75	«акредитована»
ОП36	3	1,26	3,5	3,5	3,1	3,6	3,6	1,26	3,6	2,5	«відкладена»
ОП37	5	3,2	5	3,4	5	3,6	3,4	5	3,75	3,75	«акредитована»
…	…	…	…	…	…	…	…	…	…	…	…
ОП63	3,65	1,3	3,65	3,55	3,55	3,65	3,5	3,5	3,68	2,5	«відкладена»
ОП64	1,35	1,4	3,5	3	3,68	3,75	3,75	3,6	3,68	2,5	«відкладена»
ОП65	1,3	1,3	3,2	3,35	3,68	3,6	3,4	3,7	3,6	2,5	«відкладена»

Враховуючи, що побудова моделі проводилась в умовах пасивного експерименту, за допомогою програмного продукту ПРІАМ (планування, регресія і аналіз моделі) [24] матрицю було розбито на однорідні підвибірки. Де з кожної однорідної підвибірки спочатку було сформовано навчальну матрицю, яка характеризується найкращими можливими властивостями для побудови моделі, а потім контрольну матрицю, за допомогою якої перевіряється зміщення моделі.

За результатами розрахунку до навчальної матриці увійшло 25 освітніх програм, за результатами їх оцінювання було побудовано модель. При цьому, 40 інших освітніх програм увійшли до контрольної матриці експерименту. Надалі контрольну матрицю було використано для перевірки моделі.

Оскільки, за результатами навчальної матриці будується модель, тому за допомогою програмного продукту ПРІАМ дану матрицю було перевірено на якість. Результати перевірки наведено у табл. 4.13.

Таблиця 4.13

Результати перевірки якості навчальної матриці

Якість навчальної вибірки		Значимість параметра
Розмір вибірки (кількість експериментів (ОП))	25	

Продовження таблиці 4.13

Середнє абсолютне мультиколінеарності	0,646605		0,3
Максимальне абсолютне мультиколінеарності	0,803993		0,25
Частка сильно закорельованих	1	при РХ=0,4	0,25
Середнє абсолютне з відгуком	0,759539		0,1
Частка значимих, закорельованих між собою	0,888889	при РУ=0,05	0,1
Узагальнений показник якості	0,558955		

Як видно з таблиці навчальна матриця повністю перекриває всю область визначення і має мінімально можливу закорельованість між критеріями (регресорами). Таким чином, за результатами перевірки можна зробити висновок, що регресійна модель буде мати задовільні характеристики щодо своєї якості, зокрема адекватність, інформативність та стійкість.

На *третьому етапі – «Побудова моделі»* було вибрано структуру математичної моделі та побудовано регресійну модель за допомогою програмного продукту ПРІАМ (4.5):

$$\hat{Y} = 3{,}21685 + 0{,}865646x_2 + 0{,}647632x_4 + 0{,}00501515x_5 + 0{,}0536686x_7 - 0{,}0113172x_8 + 0{,}348952x_9 + 0{,}396513x_1x_3 - 0{,}568461x_1x_4 - 1{,}62092x_6x_9, \quad (4.5)$$

де: $x_1 = 0{,}523013*(X_1 - 3{,}212)$,

$x_2 = 0{,}673129*(X_2 - 2{,}7456)$,

$x_3 = 0{,}85034*(X_3 - 3{,}824)$,

$x_4 = 0{,}436224*(X_4 - 3{,}5924)$,

$x_5 = 0{,}43163*(X_5 - 3{,}7168)$,

$x_6 = 0{,}516742*(X_6 - 3{,}5352)$,

$x_7 = 0{,}882768*(X_7 - 3{,}8672)$,

$x_8 = 0{,}393329*(X_8 - 3{,}8024)$,

$x_9 = 0{,}429923*(X_9 - 3{,}676)$,

де x_i – поліноми Чебишева першого порядку [25], (4.6):

$$f_i^{(1)} = x_i = a_{11i}(X_i + a_{10i}), \quad (4.6)$$

де і – порядковий номер критерія;

X_i – значення критеріїв (регресорів):

X_1 – проектування та цілі ОП;

X_2 – структура та зміст ОП;

X_3 – доступ до ОП та визначення результатів навчання;

X_4 – навчання і викладання за ОП;

X_5 – контрольні заходи, оцінювання здобувачів ВО та академічна доброчесність;

X_6 – людські ресурси;

X_7 – освітнє середовище та матеріальні ресурси;

X_8 – внутрішнє забезпечення якості ОП;

X_9 – прозорість та публічність.

Значення коефіцієнтів поліномів розраховувались за формулами (4.7, 4.8):

$$\sum_{i=1}^{N} f_i^{(p)} = 0, \quad (4.7)$$

$$\sum_{i=1}^{N} f_i^{(p)} f_i^{(p')} = 0, \quad (4.8)$$

де $f_i^{(p)}$ та $f_i^{(p')}$ – ортогональні контрасти ступенів p для критерія X_i.

Систему лінійних рівнянь $(x_1, \dots x_9)$ було отримано в результаті підстановки виразів ортогональних контрастів у формули (4.7, 4.8). Отримані корені будуть коефіцієнтами ортогональних поліномів Чебишева [57, 58, 59]. Вони розраховувались за формулою (4.9):

$$a_{10i} = \frac{\sum_{i=1}^{N} x_i}{N}, \quad (4.9)$$

Варто заначити, що розраховані a_{11i}, a_{12i} ... a_{1N} змінюються в інтервалі (–1, 1).

В роботі коефіцієнти математичної моделі (b_i) розраховувались з використанням методу найменших квадратів (МНК) Q [25]. Суть МНК полягає у пошуку моделі, для якої виконується умова (4.10) – сума квадратів відхилень (різниці значень відгуку, розрахованих за моделлю та експериментальних даних) наближається до min:

$$Q = \sum_{i=1}^{N} (\bar{Y}_i - \hat{Y}_i)^2 \to \min. \quad (4.10)$$

Далі було взято часткові похідні від вибраного виду моделі, за невідомими

параметрами і записано у вигляді рівнянь: $\frac{\partial Q}{\partial b_i} = 0$. Замість Y_i було підставлено значення Y та отримано систему рівнянь у матричному вигляді: $X^TXB=X^TY$, де B – вектор невідомих коефіцієнтів регресії, Y – вектор значень відгуку, X – значення незалежних змінних, X^T – транспонована матриця [25]. Оскільки матриця ортогональна то коефіцієнти моделі розраховувались за формулою (4.11):

$$b_i = \frac{\sum_{i=1}^{N}(X_i - \bar{X}_i)(Y_i - \bar{Y})}{\sum_{i=1}^{N}(X_i - \bar{X}_i)^2}. \quad (4.11)$$

Індивідуальні довірчі інтервали для коефіцієнтів регресії визначаються за формулою (4.12):

$$b_i = \pm t_{\alpha,\nu} S\sqrt{c_{ii}}, \quad (4.12)$$

де S – середньоквадратичне відхилення (корінь квадратний з дисперсії відтворюваності);

c_{ii} – діагональний елемент матриці дисперсій-коваріацій;

$t_{\alpha,\nu}$ – критичне значення критерію Стьюдента за рівнем значущості α і ν степенем свободи [25].

Відомо, що структура рівняння регресії залежить від мультиколінеарності. Тобто від послідовності включення регресорів у модель. Крім того, для лінійних ефектів знак коефіцієнту регресії вказує на напрямок впливу ефекту натуральної змінної [25].

Перевірка значущості коефіцієнтів регресії проводилась за допомогою:

t – критерію Стьюдента (4.13), частки участі, значення F-критерію для включення ефектів та деяких інших показників.

$$t = \frac{b_i\sqrt{n}}{s_{\text{від}}\sqrt{c_{ii}}}. \quad (4.13)$$

У випадку, якщо $t > t_{\alpha,\nu}$, то коефіцієнт значимий [25].

На *четвертому етапі – «Аналіз якості моделі»,* для прийняття рішення щодо застосування даної моделі було проведено аналіз її статистичних

характеристик за допомогою методу дисперсійного аналізу (табл. 4.14) [26].

Таблиця 4.14

Статистичні характеристики якості моделі

Характеристика	Назва	Умовні позначення	Значення
Інформативність	Множинний коефіцієнт кореляції	R	0,95
	Розрахункове значення критерію Фішера для перевірки значимості R	F_R	16,28
	Критичне значення критерію Фішера для перевірки значимості R	$F_{кр}$	2,59
	Степені свободи	ν_1	9
		ν_2	15
	Частка розсіяння, яка пояснюється моделлю	R^2	0,91
	Критерій Бокса-Веца	γ	2
	Гіпотеза про значимість множинного коефіцієнта кореляції	-	приймається
	Рівень інформативності	-	добрий
Адекватність	Дисперсія залишкова	$S^2_{ад}$	0,11
	Дисперсія відтворюваності	$S^2_{відт}$	0,16
	Розрахункове значення критерію Фішера для перевірки адекватності	$F_{ад}$	7,18
	Критичне значення критерію Фішера для перевірки адекватності	$F_{кр}$	2,59
	Степені свободи (ν_1, ν_2)	ν_1	9
		ν_2	15
	Гіпотеза про адекватність	-	приймається
Точність	Середня точність опису даних в процентах відхилення	%	14,98
	Середня абсолютна похибка апроксимації	Δ	0,34
Стійкість	COND (число обумовленості)		4,73
	висновок	-	задовільна
	Частка пояснення сумнівними регресорами	%	9,69
	висновок	-	задовільна
Рівень значущості	α	-	0,05

В ході регресійного аналізу було використано залишкову дисперсію $s^2_{ад}$ (4.14) та дисперсію відтворюваності $s^2_{відт}$ (4.15).

Залишкова дисперсія розраховувалась за формулою (4.14):

$$s^2_{ад} = \frac{n\sum_{i=1}^{N}(\bar{Y}_i - \hat{Y}_i)^2}{(N-k')}, \qquad (4.14)$$

Дисперсія відтворюваності розраховувалась за формулою (4.15):

$$s^2_{\text{відт}} = \frac{\sum_{i=1}^{N}\sum_{l=1}^{n}(Y_{ij}-\bar{Y}_i)^2}{N(n-1)}, \qquad (4.15)$$

де n – кількість дублюючих дослідів у кожному експерименті;

$n_i = n = \text{const}$;

N – кількість експериментів;

$\bar{Y}_i$ – середнє значення по дублюючим дослідам в і-му експерименті;

$\hat{Y}_i$ – значення відгуку, розраховане для і-го експерименту за моделлю;

Y_{ij} – значення відгуку в і-му експерименті при j-му повторі;

k′ – кількість коефіцієнтів моделі ($s^2_{\text{ад}} = 0{,}11$; $s^2_{\text{відт}} = 0{,}16$)

Інформативність моделі визначалась за формулою (4.16):

$$R^2 = Q_R / Q_{\text{заг}}, \qquad (4.16)$$

За розрахунками $R^2 = 0{,}91$.

Значущість коефіцієнта множинної кореляції визначалась за допомогою критерію Фішера (4.17):

$$F^{\text{експ}}_{v_{k'};v_{\text{ад}}} = (SS\,k'/v_{k'})/(SS_{\text{ад}}/v_{\text{ад}}) = s^2_{k'}/s^2_{\text{зал}}, \qquad (4.17)$$

За розрахунками Fк =2,59. У випадку, якщо $F^{\text{експ}}_{v_{k'};v_{\text{ад}}} > F^{\text{крит}}_{\alpha;v_{k'};v_{\text{ад}}}$ із заданим рівнем значущості, то модель інформативна.

За розрахунками R=0,95 близький до 1 та виконано умови $F_R > F_{\text{кр}}$ (у нашому випадку16,28>2,59).

Адекватність моделі визначалась за умови: ($S^2_{\text{ад}}$ / $S^2_{\text{відт}}$) < $F_{\text{кр}}$ (у нашому випадку 0,7<2,59). Обчислювальна стійкість визначалась за числом обумовленості COND (у нашому випадку COND=4,73) при частці пояснення сумнівним регресором = 9,69 %.

Таким чином, модель є інформативною, адекватною та задовільно стійкою як за структурою, так і за розрахунками.

За результатами статистичних характеристик було отримано аналітичні залежності оцінювання частки впливу критеріїв (регресорів) на якість освітньої програми (рис. 4.22).

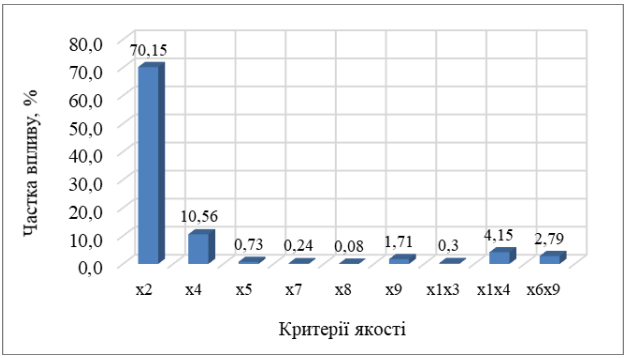


Рис. 4.22 – Частка впливу критеріїв на якість ОП

Як видно з рисунку, найбільш впливовими є два критерії: «Структура та зміст ОП» – СЗ (X_2) й «Навчання і викладання за ОП» – НВ (X_4). Частка їх впливу на якість функціонування ОП відповідно складає 70,15% та 10,56%.

Тому, при розроблені освітньої програми особливу увагу потрібно приділяти саме цим складовим. Якість функціонування ОП також залежить від сумісного впливу критеріїв (вплив критеріїв «Проектування та цілі ОП» – ПЦ (X_1) та «Навчання і викладання за ОП» – НВ (X_4) становить 4,15%, а «Людські ресурси» – ЛР (X_6) та «Прозорість та публічність» – ПП (X_9) – 2,79%). Дані критерії в «явній» або «в неявній формі» присутні в усіх складових освітньої програми.

П'ятий етап – «Практична реалізація моделі» передбачає експериментальні дослідження й розроблення на їх основі рекомендацій щодо застосуванню регресійної моделі.

В ході експериментального дослідження, з використанням програмного продукту ПРІАМ було побудовано маргінальні поверхні сумісного впливу критеріїв на якість освітньої програми:

X_6 – людські ресурси й X_4 – навчання і викладання (рис. 4.23) та X_2 – структура та зміст ОП й X_6 – людські ресурси (рис. 4.24).

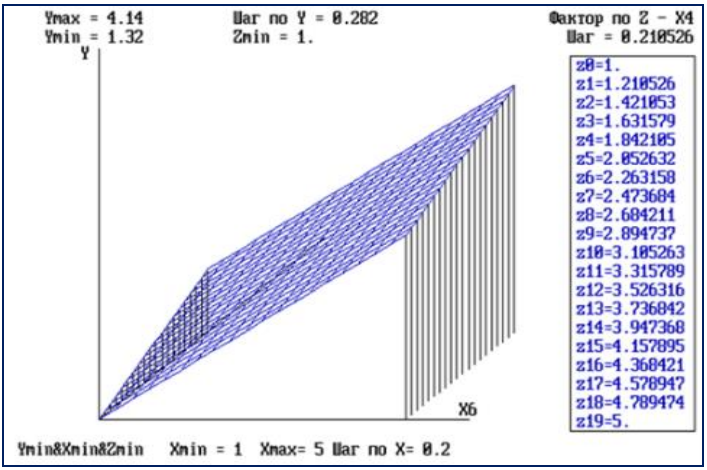


Рис. 4.23 – Сумісний вплив критеріїв X_6 і X_4

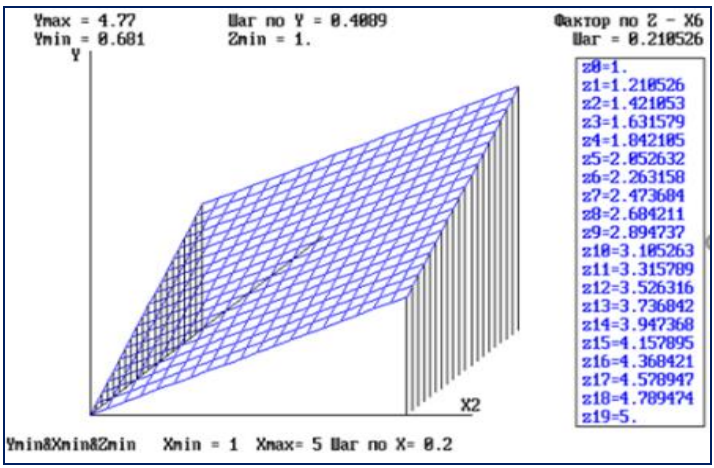


Рис. 4.24 – Сумісний вплив критеріїв X_2 і X_6

В ході досліджень встановлено, що чим більшими будуть сумісні значення таких критеріїв, як: X_6 та X_4; X_2 та X_6 тим кращою буде якість ОП. Це дозволяє визначити для кожної ОП «слабкі» та «сильні» її сторони [5].

Таким чином, проведені дослідження дозволили сформувати матрицю даних і на їх основі побудувати лінійну регресійну модель, яка дозволяє кількісно оцінювати рівні якості освітньої програми. Закладам вищої освіти дану модель доцільно застосовувати, під час проведення самоаналізу, моніторингу та оновлення освітньої програми за вимогами стейкхолдерів.

4.4. ДОСЛІДЖЕННЯ ОЦІНЮВАННЯ РІВНЯ ЯКОСТІ ОСВІТНЬОЇ ПРОГРАМИ МЕТОДОМ НЕЧІТКОГО КЛАСТЕРНОГО АНАЛІЗУ

Для перевірки ефективності та достовірності застосування кількісного багатокритеріального оцінювання рівнів якості освітніх програм було використано метод k-середніх. Дослідження проводилось за методикою наведеною у джерелі [27], яка представлена у вигляді блок-схеми покрокового алгоритму (рис. 4.25). Алгоритм включає в себе сім етапів.

На *першому етапі (крок 1)* за даними досліджень було сформовано матрицю вхідних даних (4.18), яка мала наступний вигляд:

$$X_{Mn} = \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1(n-1)} & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{21} & \cdots & x_{2(n-1)} & x_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{i(n-1)} & x_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{(M-1)1} & x_{(M-1)2} & \cdots & x_{(M-1)j} & \cdots & x_{(M-1)n-1} & x_{(M-1)n} \\ x_{M1} & x_{M2} & \ldots & x_{Mj} & \cdots & x_{M(n-1)} & x_{Mn} \end{vmatrix}, \quad (4.18)$$

де M – кількість досліджуваних об'єктів (у нашому випадку 65 освітніх програм);

n – кількість критеріїв, за якими проводилось оцінювання ОП (у нашому випадку 9-ть критеріїв оцінювання якості ОП);

i – індекси рядків;

j – індекси стовпців.

На *другому етапі (крок 2)* було задано параметри, які впливають на

виконання кластерного аналізу: k – кількість кластерів, l – параметр експоненціальної ваги, ε – критерій зупинки.

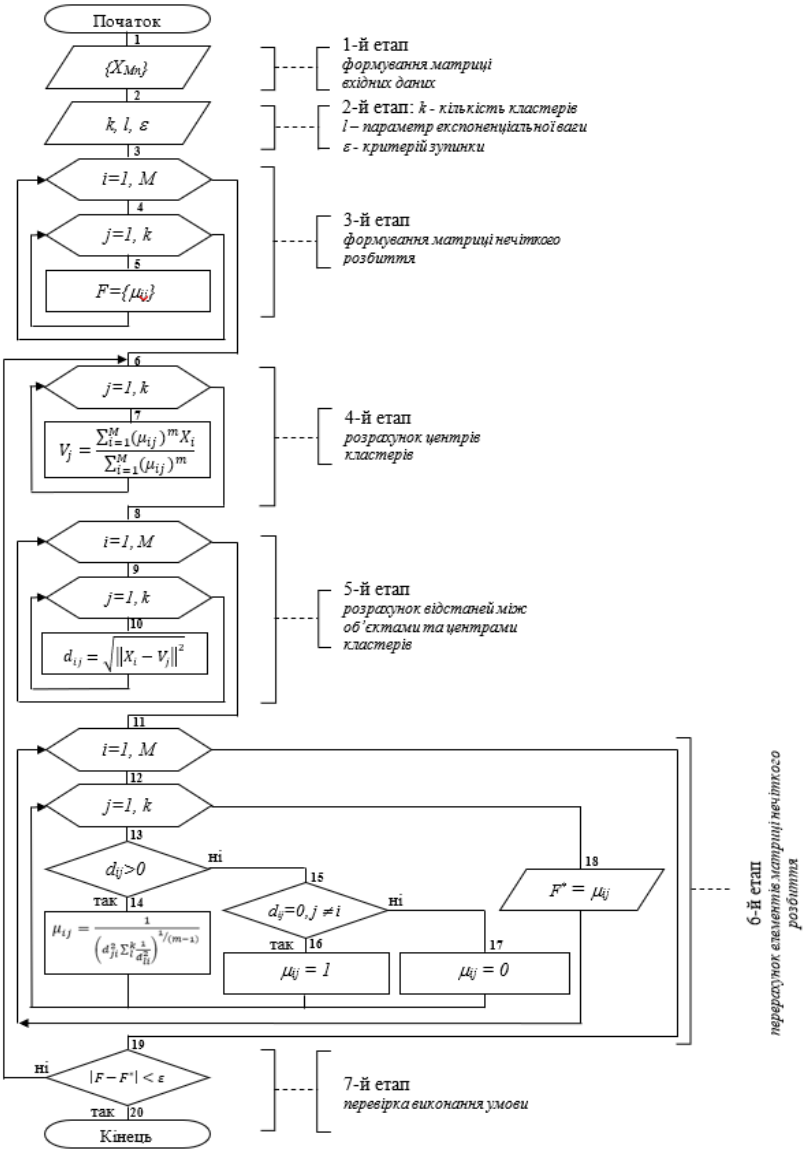


Рис. 4.25 Блок-схема нечіткого кластерного аналізу

На *третьому етапі* (*кроки 3 – 5*) для опису кластерів було використано матрицю нечіткого розбиття $F = [\mu_{ij}]$, яка описує належність елементів вибірки до певного класу розбиття. При цьому $\mu_{ij} \in [0,1]$; $i = \overline{1,M}$; $j = \overline{1,k}$, де вектор рядок і містить степінь належності освітньої програми, що досліджується $X_i=(x_{i1}, x_{i2}, \dots x_{in})$, оціненої за 9-ма критеріями (n=9), відповідним кластерам розбиття – A_1, A_2, …, A_k до кластеру j, а сумарне значення вектор-рядка дорівнює 1, та обчислюється за формулою (4.19) [27]:

$$\sum_{j=1}^{k} \mu_{ij} = 1. \qquad (4.19)$$

Варто зазначити, що сума усіх значень вектор рядків знаходиться в межах умови (4.20):

$$0 < \sum_{i=1}^{M} \mu_{ij} < M. \qquad (4.20)$$

На *четвертому етапі (крок 6 – 7)* виконувався розрахунок центрів кластерів (4.21):

$$V_j = \frac{\sum_{i=1}^{M} (\mu_{ij})^m X_i}{\sum_{i=1}^{M} (\mu_{ij})^m}, \; j = \overline{1,k}. \qquad (4.21)$$

На *п'ятому етапі (крок 8 – 10)* за формулою (4.22) виконувався розрахунок відстані між об'єктами та центрами кластерів:

$$d_{ij} = \sqrt{\|X_i - V_j\|^2}, \; i = \overline{1,M} \;\; j = \overline{1,k}. \qquad (4.22)$$

На *шостому етапі (крок 11 – 18)* за формулою (4.23) було застосовано процедуру перерахунку елементів матриці нечіткого розбиття (матриці приналежності):

$$F^* = \left[\mu_{ij}\right] = \begin{cases} \dfrac{1}{\left(d_{ji}^2 \sum_{l=1}^{k} \frac{1}{d_{li}^2}\right)^{1/(m-1)}}, d_{ij>0} \\ \begin{cases} 1, d_{ij} = 0, & i = j \\ 0 d_{ij} = 0, & i \neq j \end{cases} \end{cases}. \qquad (4.23)$$

На *сьомому етапі (крок 19)* було проведено перевірку виконання умови

(4.23):

$$\|F - F^*\| < \varepsilon, \qquad (4.23)$$

При цьому слід зазначити, що у разі виконання умови (4.23) процедура розбиття на кластери закінчується *(крок 20)*. Якщо умова не виконується то відбувається перехід на *четвертий етап (крок 6)* – розрахунок центрів кластерів за формулою (4.21).

Експериментальні дослідження визначення рівня відповідності ОП проводились за реальними даними якісного оцінювання 65-ти освітніх програм 49 закладів вищої освіти з різних регіонів України. В якості вихідних даних було використано звіти НАЗЯВО з акредитації освітніх програм [54, 55]. Нижче наведено фрагмент отриманої матриці даних (4.24) за результатами кількісного оцінювання 65-ти освітніх програм за 9-ма критеріями якості:

$$X_{mn} = \begin{vmatrix} 3,5 & 3 & 5 & 3,2 & 5 & 3,5 & 3,2 & 3,5 & 5 \\ 3 & 2 & 3,4 & 1,6 & 3,5 & 2,8 & 3 & 3,2 & 3 \\ 3,4 & 3,7 & 3,4 & 3,6 & 3,5 & 3,2 & 5 & 3,4 & 3,5 \\ 2,8 & 2 & 3,5 & 3,5 & 3,2 & 3 & 3,4 & 3,5 & 3,2 \\ 2,8 & 2 & 2,6 & 3 & 3,2 & 1,5 & 2,8 & 2,6 & 2,51 \\ 4,8 & 2,9 & 3,5 & 5 & 5 & 3,5 & 3,2 & 3 & 3,2 \\ 3 & 3,4 & 3,6 & 3,6 & 3,6 & 3,4 & 3,5 & 3,6 & 3,4 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 3,7 & 3,75 & 3,75 & 3,75 & 3,75 & 3,75 & 3,75 & 3,75 & 3,48 \\ 3,75 & 3,6 & 3,55 & 3,75 & 3,75 & 3,75 & 3,75 & 3,75 & 3,75 \\ 2,8 & 2,35 & 2,9 & 2,25 & 3,5 & 1,45 & 3,65 & 3,7 & 3,6 \\ 3,65 & 3,7 & 3,75 & 5 & 3,6 & 5 & 3,6 & 5 & 5 \\ 3,65 & 1,3 & 3,65 & 3,55 & 3,55 & 3,65 & 3,5 & 3,5 & 3,68 \\ 1,35 & 1,4 & 3,5 & 3 & 3,68 & 3,75 & 3,75 & 3,6 & 3,68 \\ 1,3 & 1,3 & 3,2 & 3,35 & 3,68 & 3,6 & 3,4 & 3,7 & 3,6 \end{vmatrix}. \qquad (4.24)$$

Розбиття 65-ти ОП на кластери виконувалось за допомого макросу VBA, в основу якого покладено вище наведений алгоритм, з використанням наступних параметрів:

– кількість кластерів k = 4;
– параметр експоненціальної ваги l = 2;
– критерій зупинки ε = 0,00001.

Далі за допомогою цього ж макросу проводилась перевірка досконалості та достатності кількості розбиття 65-ти ОП на 4-ри кластери. Оцінка якості розбиття на кластери включала в себе внутрішньо групове розсіювання, міжгрупове розсіювання (розсіювання між центрами кластерів) та якість розбиття. Результати визначених їх параметрів наведено у табл. 4.15.

Таблиця 4.15

Параметри якості нечіткого кластерного аналізу

Параметри	Значення
Внутрішньогрупове розсіювання	100,75
Міжгрупове розсіювання	76,86
Якість розбиття	0,43

Отримані результати аналізу є достатніми при розбитті матриці даних 65-ти ОП на чотири кластери.

Оскільки, кластер це множина точок для якої виконується співвідношення $Q_j/M < S/M$ (4.25), а згущення множина точок, яка задовольняє умову: $d^2_{jmax} < S/M$ (4.26),

де Qj – це внутрішньогрупове розсіювання,

S – загальне розсіювання.

Тому під час виконання кластеризації за допомогою вище наведеного макросу було перевірено виконання умов (4.25 та 4.26). Фрагмент зведених результатів сформованої матриці нечіткого розбиття 65-ти ОП на чотири кластери та результати перевірки умов (4.25 та 4.26) наведено в табл. 4.16.

Таблиця 4.16

Розбиття 65-ти ОП на 4 кластери

Назва об`єкту	Номер кластера				Ступінь приналежності ОП до кластеру	Приналеж-ність до кластеру	Вид акредитації
	1	2	3	4			
ОП1	0,33151	0,17831	0,37106	0,11912	0,37106	3	«акредитована»
ОП2	0,10361	0,28688	0,08354	0,52598	0,52598	4	«відкладена»
ОП3	0,45992	0,13732	0,30791	0,09484	0,45992	1	«зразкова»

Продовження таблиці 4.16

ОП4	0,04418	0,70705	0,03219	0,21658	0,70705	2	«відкладена»
…	…	…	…	…	…	…	…
ОП60	0,82640	0,02090	0,14078	0,01191	0,82640	1	«акредитована»
ОП61	0,13156	0,30339	0,11396	0,45108	0,45108	4	«відмовлена»
ОП62	0,35826	0,14269	0,40637	0,09268	0,40637	3	«акредитована»
ОП63	0,09814	0,65476	0,08351	0,16359	0,65476	2	«відкладена»
ОП64	0,11348	0,39504	0,09531	0,39616	0,39616	4	«відкладена»
ОП65	0,11243	0,40234	0,09401	0,39122	0,40234	2	«відкладена»
Qj/M	0,3925	0,3786	0,3950	0,3840	0,3950		
S/M	2,7319	2,7319	2,7319	2,7319	0,2632		
d^2_{jmax}	1,1439	1,2768	1,6749	2,2835			
Кластер: (Q_j/M < S/M)?	Так	Так	Так	Так			
Згущення: (d^2_{jmax} < S/M)?	Так	Так	Так	Так			

Як видно з таблиці таке розбиття освітніх програм за розрахунками задовольняє умови (4.25, 4.26), тобто значення отримані в кожній групі кластерів відповідають як за кластерами так і за згущенням.

Проведений аналіз отриманих результатів (табл. 4.16) показав, що при розбитті 65-ти освітніх програм на чотири кластери:

– до 1-го кластера увійшли 23 ОП, 3 з яких мають рівень акредитації – «зразкова», 1 – «відкладена» та 19 – «акредитована»;

– до 2-го кластера увійшли – 13 ОП, які є «відкладеними»;

– до 3-го кластера увійшли – 13 ОП, серед яких 1 ОП – «зразкова» та 12 ОП – «акредитована»;

– до 4-го кластера увійшли 16 ОП, з них 9 ОП – «відкладена» та 7 ОП – «відмовлена».

Таким чином, групування освітніх програм з використанням нечіткого кластерного аналізу методом k – середніх, доводить, що застосування лише якісного оцінювання рівня освітніх програм не є достатнім і має досить велику суб'єктивну похибку. Тому для отримання достовірної оцінки потрібно застосовувати, як якісне так і кількісне оцінювання. Такий підхід дозволяє більш достовірно оцінювати рівень

відповідності акредитації освітньої програми, тобто зменшує суб’єктивну похибку.

Висновки до розділу 4

1. Розроблено покроковий алгоритм для оцінювання ризиків освітнього процесу за відповідною освітньою програмою, в рамках якого вона функціонує. В основу алгоритму покладено ризик-орієнтований підхід, принципи TQM та вимоги стандартів ДСТУ ISO 31010. Застосування даного алгоритму дозволяє оцінювати ризики освітнього процесу та освітньої програми, в рамках якого вона функціонує на всіх етапах її життєвого циклу. Також за результатами моніторингу розробляти організаційно-технічні заходи щодо їх зменшення.

2. Застосування програмного комплексу «Експерти ЗВО» дозволяє формувати експертну групу (з урахуванням професійно-орієнтованих критеріїв), що скоротить час на обробку інформації щодо їх компетентності.

3. Запропоновано класифікацію ризиків освітнього процесу, за відповідною освітньою програмою, в рамках якого вона функціонує. В основу даної класифікації покладено метод ієрархії, що дозволяє встановити взаємозв’язки між його структурними складовими та систематизувати ризики за певними ознаками.

4. Проаналізовано результати акредитації 65 освітніх програм, які представляють 31 спеціальність за 14 галузями знань і доведено, що якісна оцінка має великий рівень суб’єктивності й тому не завжди вірно відтворює рівень якості освітніх програм та термін дії їх акредитації.

5. Отримано залежності результатів оцінювання рівнів відповідності освітніх програм (за 9-ма критеріями) експертних груп та висновків Галузевої експертної ради. Аналіз даних залежностей показав, що існують розходження в результатах їх оцінювання. Зокрема, 13% від

загальної кількості (585) критеріальних оцінок мали розходження в результатах експертного оцінювання та висновків Галузевої експертної ради. Це доводить, що якісна оцінка є недостатньо достовірною і потребує свого уточнення.

6. Для переведення якісних оцінок в кількісні було побудовано регресійну модель. Дана модель описує якість функціонування освітньої програми незалежно від галузі знань і спеціальності, за якою вона розроблена. Перевірка даної моделі показала, що вона є дієздатною, адекватною, інформативною та стійкою. Застосування моделі дозволяє кількісно оцінити якість освітньої програми, тобто підвищити достовірність визначення рівня її акредитації.

Список використаних джерел до розділу 4

1. ДСТУ ISO 31010:2013 (IEC/ISO 31010:2013, IDT) «Керування ризиком (Методи загального оцінювання ризику)». – Київ: Мінекономрозвитку України. – 2015. – 45 с.

2. Науковий твір «Алгоритм оцінювання ризиків на основі вимог стандарту ДСТУ ISO 31010:2013» : а. с. / А.П. Волівач, Г.І. Хімічева. – № 97089; дата реєстрації 07.04.2020; опубл. 29.05.20. – бюл. № 58.

3. Грабовецький Б.Є. Методи експертних оцінок : теорія, методологія, напрямки використання : монографія / Б.Є. Грабовецький //. – Вінниця : ВНТУ. – 2010. – 171 с.

4. Комп'ютерна програма «Програмний комплекс «Експерти ЗВО» для формування експертних груп в освітній галузі» (скорочена назва «Експерти ЗВО») : а. с. / А.П. Волівач, Г.І. Хімічева. – № 95933; дата реєстрації 10.02.20; опубл. 31.03.20. – бюл. № 57.

5. Волівач А.П. Удосконалення нормативного забезпечення з оцінювання якості освітніх програм закладів вищої освіти. дис. канд. техн. наук: 05.01.02/ Волівач Антоніна Петрівна. – К., 2021. – 299 с.

6. Черненко Н.М. Особливості управління навчальним закладом на

засадах ризик-менеджменту. / Н.М. Черненко Наука і освіта. – 2015. – №9. – С. 196 – 200.

7. Пикфорд Дж. Управление рисками. М. : ООО Вершина. – 2004. – 220 с.

8. Волівач А.П. Класифікація ризиків навчального процесу на основі застосування методу ієрархій / А.П. Волівач, Г.І. Хімічева // Monografia pokonfenrencyjna. Science, research, development. Valletta (Republic of Malta). – 30.07.2019 – 31.07. – 2019. – 19. – С. 96 – 106.

9. Волівач А.П. Визначення факторів ризиків у відповідності до моделі СУЯ ВНЗ за вимогами ДСТУ ISO 9001:2015 / А.П. Волівач, Г.І. Хімічева // збірник наукових праць міжнародної конференції «Інноваційні технології в науці та освіті. Європейський досвід». м. Відень, Австрія. – 2017. — ТОМ І.– С. 48 – 57.

10. Волівач А.П. Застосування методу PEST-аналізу для визначення впливу факторів ризиків на освітню діяльність ЗВО / А.П. Волівач, Г.І. Хімічева // Проблеми інженерно-педагогічної освіти. Зб. наук. пр. – Харків, УІПА. – 2018. – № 59. – С. 74 – 82.

11. Самохвалов Ю.Я., Науменко Е.М. Экспертное оценивание. Методический аспект / Ю.Я. Самохвалов, Е.М. Науменко. – Киев. – 2007. – 264 с.

12. Павлов А.Н. Методы обработки экспертной информации: учебно-метод. Пособие / А.Н. Павлов, Б.В. Соколов // Санкт-Петербург. – 2005. – 42 с.

13. Полякова Г. Концептуальні основи та технологія формування системи внутрішнього забезпечення якості у закладі вищої освіти / Г. Полякова // Електронне наукове фахове видання «Адаптивне управління: теорія і практика». – 2018. Випуск 5 (9). [Електронний ресурс]. – Режим доступу : https://amtp.org.ua/index.php/journal/article/view/73/51.

14. Никсон Р. Создаем динамические веб-сайты с помощью PHP,

MySQL, JavaScript, CSS и HTML5 / Робин Никсон. – Ньютон, Массачусетс, США : O`Reilly. – 2016. – 768 с.

15. Орлов А.И. Экспертные оценки : учеб. пос. / А. И. Орлов. – М. : Экзамен. – 2002. – 31 с.

16. Положення про акредитацію освітніх програм, за якими здійснюється підготовка здобувачів вищої освіти : Наказ МОН України від 11.07.2019 № 977 // [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/z0880-19 (Дата звернення: 08.09.21)

17. Офіційний сайт ЄДЕБО : Реєстр суб'єктів освітньої діяльності // [Електронний ресурс]. – Режим доступу: https://registry.edbo.gov.ua (Дата звернення: 20.09.2021)

18. Офіційний сайт НАЗЯВО : акредитація // [Електронний ресурс]. – Режим доступу: https://naqa.gov.ua/акредитація (Дата звернення: 20.09.2021)

19. Реєстр акредитаційних справ у 2019/2020 навчальному році / [Електронний ресурс]. – Режим доступу: https://docs.google.com/spreadsheets/d/e/2PACX-1vRMzqz6RMhzh58rv5BWQG7ITw7QBSohPckn2IkQqJwJ66L5b2G5dm1vkHZMZBF9FmuyqYtaA5aCv6CO/pubhtml# (Дата звернення: 20.07.2020).

20. Акредитаційні справи у 2019/2020 навчальному році https://naqa.gov.ua/uploads/cases_2019 (Дата звернення: 20.07.2020).

21. Критерії : Додаток до Положення про акредитацію освітніх програм, за якими здійснюється підготовка здобувачів вищої освіти (пункт 6 розділу I) // [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/z0880-19#n182 (Дата звернення: 08.09.21)

22. Хімічева Г.І. Застосування нечіткого кластерного аналізу для оцінювання рівня відповідності освітньої програми / Г.І. Хімічева, А.П. Волівач // Збірник наукових праць I всеукраїнської конференції здобувачів

вищої освіти і молодих учених «Інноватика в освіті, науці та бізнесі: виклики та можливості» (17 листопада 2020 р., м. Київ). – К. : КНУТД. – 2020. – С. 157 – 167.

23. Математичне моделювання та оптимізація технологічних систем : Метод. Вказівки до лаборатор.-комп’ют. практикуму для студ. спец. «Технологія машинобудування» усіх форм навчання / Уклад.: С.М. Лапач, С.Г. Радченко, Р.В. Галайда. – К.: НТУУ «КПІ». – 2007. – 116 с.

24. Лапач С.Н., Радченко С.Г., Бабич П.Н. Планирование, регрессия и анализ моделей PRIAM (ПРИАМ) / Каталог программные продукты Украины. К.: 1993. С. 24 – 27. ДСТУ ISO 9001:2015 «Системи управління якістю (Вимоги)». – Київ: ДП «УкрНДНЦ». – 2016. – 22 с.

25. Лапач С.Н. Статистические методы в медико-биологических исследованиях с использованием Excel / С.Н. Лапач, А.В. Чубенко, П.Н. Бабич // К. : Морион. – 2001. – 408 с.

26. Лапач С.Н. Статистика в науке и бизнесе / С.Н. Лапач, А.В. Чубенко, П.Н. Бабич // К.:Морион. – 2002. – 640 с.

27. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. – 2007 г. – 288 С.

5. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВЗАЄМОДІЇ НИТКИ З НАПРАВЛЯЮЧИМИ КРУГЛОВ’ЯЗАЛЬНИХ 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.

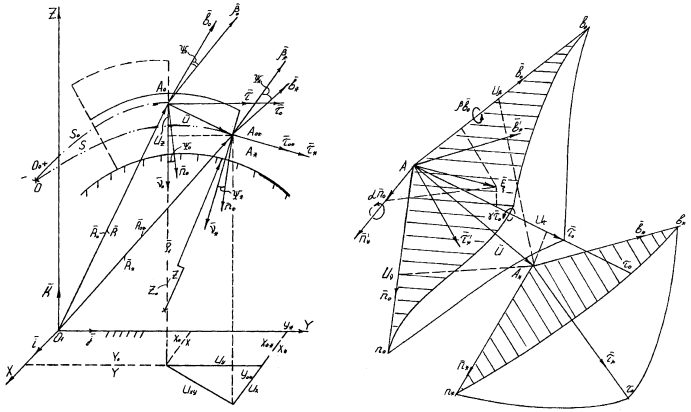
Розв’язання цих питань потребує розробки основ статики, кінематики та динаміки процесу взаємодії нитки з направляючою у формі тора при урахуванні деформації перетину в зоні контакту, жорсткості на згин та зміни коефіцієнту тертя при радіальному охопленні нитки поверхнею напрямної.

5.1. 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 елементу 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.5.1а представлена розрахункова схема деформації у зоні контакту з направляючою у формі тора.



а. б.

Рис.5.1. Схеми для визначення геометричних характеристик форми осі нитки при взаємодії з напрямними великої кривизни у формі тору та

співвідношення між вектором переміщення та вектором оберту натурального тригранника

Для опису форми осі деформованої в перетині нитки з направляючою у формі тора необхідно ввести три координатні системи [1]. Дві системи рухомі: натурального тригранника $\tau_*\nu_*\beta_*$ і головного тригранника $\tau_*\nu_*b_*$. 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 ν_* 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 ν_* натурального тригранника не співпадають.

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

$$\overrightarrow{R_{0*}} = \overrightarrow{R_0} + \overrightarrow{U_\Delta^\beta}\,,\ \overrightarrow{R_*} = \overrightarrow{R_*} + \overrightarrow{U_\Delta^\beta}\,. \tag{5.1}$$

Диференціюючи рівняння (5.1) по відповідних дугових координатах, отримаємо встановити залежність між одиничними ортами в різних системах координатних осей

$$\overrightarrow{\tau_{0*}} = \overrightarrow{\tau_0} + \frac{\partial \overrightarrow{U_\Delta^\beta}}{\partial S_0}\ ,\ \overrightarrow{\tau_*} = \vec{\tau} + \frac{\partial \overrightarrow{U_\Delta^\beta}}{\partial S}\cdot \tag{5.2}$$

Для визначення залежності, що визначає одиничний орт головної нормалі залежно від вибору дугових координат у випадку радіального охоплення, диференціюючи рівняння (5.2) ще раз по відповідних фізичних S_0 і геометричних координатах S, отримаємо

$$\overrightarrow{\nu_*} = (1+\varepsilon)\overrightarrow{\nu_0}\frac{\rho_*}{\rho_{0*}} + (1+\varepsilon)\rho_*\frac{\partial^2 \overrightarrow{U_\Delta^\beta}}{\partial S_0^2} + \frac{\partial \varepsilon}{\partial S_0}\rho_*\overrightarrow{\tau_{0*}}\cdot \tag{5.3}$$

Для визначення одиничного орта бінормалі, у випадку радіального охоплення, скористуємося векторним рівнянням

$$\overrightarrow{\beta_*} = (1+\varepsilon)^2\left[\overrightarrow{\beta_0}\frac{\rho_*}{\rho_{0*}} + \frac{\partial \overrightarrow{U_\Delta^\beta}}{\partial S_0}\times\overrightarrow{\nu_0}\frac{\rho_*}{\rho_{0*}} + \overrightarrow{\tau_0}\times\frac{\partial^2 \overrightarrow{U_\Delta^\beta}}{\partial S_0^2}\rho_* + \frac{\partial \overrightarrow{U_\Delta^\beta}}{\partial S_0}\times\frac{\partial^2 \overrightarrow{U_\Delta^\beta}}{\partial S_0^2}\rho_*\right]\cdot$$

Перейдемо до визначення кривизни осі нитки що деформується для направляючій у формі тора. На рис.5.1б 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

$$\overrightarrow{\tau_*} = \overrightarrow{\tau_0} - \alpha b_0 + \beta\overrightarrow{n_0}\,,\overrightarrow{n_*} = \overrightarrow{n_0} - \beta\overrightarrow{\tau_0} + \gamma\overrightarrow{b_0}\,,\overrightarrow{b_*} = \overrightarrow{b_0} + \alpha\overrightarrow{\tau_0} - \gamma\overrightarrow{n_0}\,, \tag{5.4}$$

де α, β, γ - проекції вектора повороту ξ на осі головного тригранника $A\tau_0 n_0 b_0$ у випадку радіального охоплення.

Перші похідні одиничних ортів (5.4) по дуговій координаті через проекції, для направляючій у формі тора, можна представити у вигляді

$$\frac{\partial U^{\beta}_{\Delta\tau}}{\partial S} - q_0 U^{\beta}_{\Delta\nu} + p_0 U^{\beta}_{\Delta\beta} = 0,\ \frac{\partial U^{\beta}_{\Delta\nu}}{\partial S} + q_0 U^{\beta}_{\Delta\tau} - r_0 U^{\beta}_{\Delta\beta} = \beta,$$
$$\frac{\partial U^{\beta}_{\Delta\beta}}{\partial S} - p_0 U^{\beta}_{\Delta\tau} + p_0 U^{\beta}_{\Delta\nu} = -\alpha,\ q_0 = \frac{\cos\Psi_0}{\rho_0},\ p_0 = \frac{\sin\Psi_0}{\rho_0},\ r_0 = \frac{1}{\rho_{01}} + \frac{\partial\Psi_0}{\partial S}, \quad (5.5)$$

ρ_{01} - радіус кривизни геометричного кручення осі нитки для направляючій у формі тора;

ψ_0 - кут Сен-Венана у випадку радіального охоплення.

Якщо нехтувати проекцією вектора $\vec{U}^{\beta}_{\Delta}$ на дотичну вісь ($U^{\beta}_{\Delta\tau} = 0$) і вважати, що нитка розташовується в дотичній площині ($U^{\beta}_{\Delta\beta} = 0$), то тоді з (5.5) визначимо

$$\frac{1}{\rho_1} = \frac{1}{\rho_0} + \frac{\partial^2 U^{\beta}_{\Delta\nu}}{\partial S^2} - \frac{U^{\beta}_{\Delta\nu}}{\rho_0^2}.$$

Перейдемо до визначення швидкостей і прискорень точок осі нитки яка деформується в зоні контакту у випадку радіального охоплення. На рис. 5.2 представлена розрахункова схема для визначення швидкостей та прискорень нитки в зоні деформації для направляючій 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^{β}_{Δ}.

Диференціюючи векторне рівняння для радіус-вектора за часом, отримаємо

$$\vec{V}_* = \vec{V} + \frac{\partial \vec{U}^{\beta}_{\Delta}}{\partial t}, \quad (5.6)$$

де V - швидкість точки A у випадку радіального охоплення.

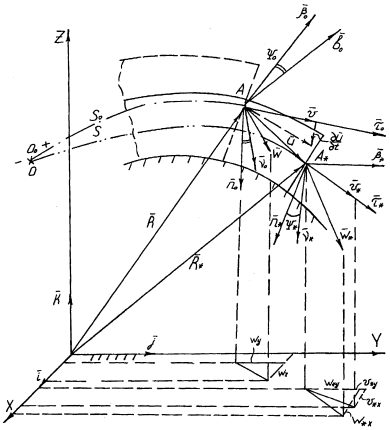


Рис.5.5. Розрахункова схема для визначення швидкостей та прискорень нитки в зоні контакту у випадку радіального охоплення

Система рівнянь, для швидкостей точок осі нитки що деформується в зоні контакту, для направляючій у формі тора у випадку радіального охоплення матиме наступні вирази

$$\left(\frac{\partial \vec{V}_x}{\partial S}\right)_\tau = \frac{\partial V_{x\tau}}{\partial S} + V_{xb}p_1 - V_{xn}q_1, \quad \left(\frac{\partial \vec{V}_x}{\partial t}\right)_\tau = \frac{\partial V_{x\tau}}{\partial t} + \omega_2 V_{*b} - \omega_3 V_{*n},$$
$$\left(\frac{\partial \vec{V}_x}{\partial S}\right)_n = \frac{\partial V_{xn}}{\partial S} - V_{xb}r_1 + V_{x\tau}q_1, \quad \left(\frac{\partial \vec{V}_x}{\partial t}\right)_n = \frac{\partial V_{xn}}{\partial t} - \omega_1 V_{*b} + \omega_3 V_{*\tau}, \qquad (5.7)$$
$$\left(\frac{\partial \vec{V}_x}{\partial S}\right)_b = \frac{\partial V_{xb}}{\partial S} + V_{xn}r_1 - V_{x\tau}p_1 \quad \left(\frac{\partial \vec{V}_x}{\partial t}\right)_b = \frac{\partial V_{xb}}{\partial t} + \omega_1 V_{*n} - \omega_2 V_{*\tau},$$

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 елементу нитки на осі головного тригранника у випадку радіального охоплення (з урахуванням кутової швидкості деформаційного кручення [1]).

Система диференціальних рівнянь (5.7), 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 (5.7) по t, отримаємо

$$\left(\frac{\partial \vec{W}_*}{\partial t}\right)_\tau = \frac{\partial W_{*\tau}}{\partial t} + \frac{\partial \omega_2}{\partial S} V_{*b} + \omega_2 \frac{\partial V_{*b}}{\partial S} - \frac{\partial \omega_3}{\partial S} V_{*n} - \omega_3 \frac{\partial V_{*n}}{\partial S},$$
$$\left(\frac{\partial \vec{W}_*}{\partial t}\right)_n = \frac{\partial W_{*n}}{\partial S} + \frac{\partial \omega_3}{\partial S} V_{*\tau} + \omega_3 \frac{\partial V_{*\tau}}{\partial S} - \frac{\partial \omega_1}{\partial S} V_{*b} - \omega_1 \frac{\partial V_{*b}}{\partial S},$$
$$\left(\frac{\partial \vec{W}_*}{\partial t}\right)_b = \frac{\partial W_{*b}}{\partial S} + \frac{\partial \omega_1}{\partial S} V_{*n} + \omega_1 \frac{\partial V_{*n}}{\partial S} - \frac{\partial \omega_2}{\partial S} V_{*\tau} - \omega_2 \frac{\partial V_{*\tau}}{\partial S}.$$

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 елементу 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

$$\frac{\partial \vec{R}}{\partial S}+\vec{R}_0-\vec{\Phi}_*=0,\ \frac{\partial \vec{M}}{\partial S}+\vec{M}_0+\vec{\tau}_*\times\vec{Q}-\vec{M}_\Phi=0, \tag{5.8}$$

описують рух нескінченно малого елементу нитки у випадку радіального охоплення. Головні вектор R та момент M, для направляючій у формі тора, визначаються наступними векторними рівняннями

$$\vec{R}=P\vec{\tau}_*+Q_2\vec{n}_*+Q_3\vec{b}_*,\ \vec{M}=M_k\vec{\tau}_*+M_{u2}\vec{n}_*+M_{u3}\vec{b}_*. \tag{5.9}$$

Для отримання системи диференціальних рівнянь, для направляючій у формі тора, необхідно векторні рівняння (5.8) спроектувати на осі головного тригранника τ_*, n_*, b_*, з урахуванням (5.9) отримаємо

$$\begin{aligned}
&\frac{\partial P}{\partial S}-Q_2q_1+Q_3p_1-4\frac{\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{aR^b}{P_0^b}N\cos\phi_t=T[\frac{\partial(V_\tau+U_{\Delta\tau}^{\beta\prime})}{\partial t}+\\
&+(\omega_{0n}+\omega_{un})(V_b+U_{\Delta b}^{\beta\prime})-(\omega_{0b}+\omega_{ub})(V_n+U_{\Delta n}^{\beta\prime})],\\
&\frac{\partial Q_2}{\partial S}+P\,q_1+Q_3r_1-N=T[\frac{\partial(V_n+U_{\Delta n}^{\beta\prime})}{\partial t}-\\
&-(\omega_{0\tau}+\omega_{u\tau})(V_b+U_{\Delta b}^{\beta\prime})+(\omega_{0b}+\omega_{ub})(V_\tau+U_{\Delta\tau}^{\beta\prime})],\\
&\frac{\partial Q_3}{\partial S}-P\,p_1+Q_2r_1+4\frac{\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{aR^b}{P_0^b}N\sin\phi_t=T[\frac{\partial(V_b+U_{\Delta b}^{\beta\prime})}{\partial t}+\\
&+(\omega_{0\tau}+\omega_{u\tau})(V_n+U_{\Delta n}^{\beta\prime})-(\omega_{0n}+\omega_{un})(V_\tau+U_{\Delta\tau}^{\beta\prime})],\\
&\frac{\partial M_\kappa}{\partial S}-M_{u2}q_1+Bq_1p_1+M_\tau=\gamma_n\left[J_{\tau_0}\pm\Delta_\tau(\vec{U}_\Delta^{\beta})\right]\varepsilon_1,\\
&\frac{\partial M_{u2}}{\partial S}+M_kq_1+Bq_1r_1-Q_3+M_n=\gamma_n\left[J_{n_0}\pm\Delta_n(\vec{U}_\Delta^{\beta})\right]\varepsilon_2,\\
&\frac{\partial\left(Bq_1\right)}{\partial S}-M_kp_1+M_{u2}r_1+Q_2+r_x4\frac{\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{aR^b}{P_0^b}N=\gamma_n\times\\
&\times\left[J_{b_0}\pm\Delta_b(\vec{U}_\Delta^{\beta})\right]\varepsilon_3,\ N=b_2E_1\delta_1(1-b_3\delta_1^{b_4})+\eta\dot{\delta}_1^{b_5}(1-b_5\dot{\delta}_1^{b_6}),
\end{aligned} \tag{5.10}$$

та рівняння для визначення головних компонентів кривизни у випадку радіального охоплення

$$q_1 = \frac{\cos\Psi_0}{\rho_0} + \frac{\partial\beta}{\partial S} + \frac{\alpha}{\rho_{01}} + \alpha\frac{\partial\Psi_0}{\partial S},\ p_1 = \frac{\sin\Psi_0}{\rho_0} + \frac{\partial\alpha}{\partial S} - \frac{\beta}{\rho_{01}} - \beta\frac{\partial\Psi_0}{\partial S};$$

$$r_1 = \frac{1}{\rho_0} + \frac{\partial\Psi_0}{\partial S} - \alpha\frac{\cos\Psi_0}{\rho_0} + \frac{\sin\Psi_0}{\rho_0}\beta,\ \gamma = \frac{\partial U^{\beta}_{\Delta\tau}}{\partial S} - \frac{\cos\Psi_0}{\rho_0}U^{\beta}_{\Delta\nu} + \frac{\sin\Psi_0}{\rho_0}U^{\beta}_{\Delta\beta}; \qquad (5.11)$$

$$\beta = \frac{\partial U^{\beta}_{\Delta\nu}}{\partial S} + \frac{\cos\Psi_0}{\rho_0}U^{\beta}_{\Delta\tau} - \frac{U^{\beta}_{\Delta\beta}}{\rho_{01}} - U^{\beta}_{\Delta\beta}\frac{\partial\Psi_0}{\partial S},\ \alpha = \frac{\sin\Psi_0}{\rho_0}U^{\beta}_{\Delta\tau} - \frac{\partial U^{\beta}_{\Delta\beta}}{\partial S} - \frac{U^{\beta}_{\Delta\nu}}{\rho_{01\ \nu}} - U^{\beta}_{\Delta\nu}\frac{\partial\Psi_0}{\partial S}.$$

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;

$U^{\beta}_{\Delta\tau}, U^{\beta}_{\Delta\nu}, U^{\beta}_{\Delta\beta}$ – 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].

5.2. ВЗАЄМОДІЯ НИТОК З НАПРАВЛЯЮЧИМИ СТРУКТУРНИМИ ЕЛЕМЕНТАМИ ВЕЛИКОЇ КРИВИЗНИ СИСТЕМИ ПОДАЧІ НИТКИ

В процесі переробки нитки взаємодіють з направляючими поверхнями великої кривини у випадку радіального охоплення. На рис.5.3 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.

а. б. в. г. д.

Рис.5.3. 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.5.4 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

$$\varphi = \varphi_P + \varphi_{см1} + \varphi_{см2} - \varphi_{ижс1} - \varphi_{ижс2}, \qquad (5.12)$$

де φ_P - кут охоплення ниткою напрямної у формі тора без урахування зминання та жорсткості на згин;

$\varphi_{см1}, \varphi_{см2}$- кути, на які збільшується кут φ_P за рахунок деформації зминання в зоні контакту нитки з напрямною у формі тора;

$\varphi_{ижс1}, \varphi_{ижс2}$- кути, на які зменшується кут φ_P за рахунок наявності жорсткості нитки на згин у випадку радіального охоплення.

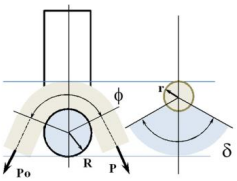


Рис.5.4. Загальна розрахункова схема взаємодії нитки з напрямною у формі тора

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).

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 направляючій у формі тора отримаємо наступні залежності [3-5]

$$\cos\gamma_0 = 1 - \frac{B_0}{2P_0(R+r)^2},\ \cos\gamma = 1 - \frac{B_0}{2P(R+r)^2}. \qquad (5.13)$$

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 нитики на ділянці AB поверхні напрямної у формі тора.

Система диференційних рівнянь у випадку радіального охоплення, яка описує рівновагу нескінченно малого елементу нитки $dS = (R + r)d\varphi$ (S - дугова координата), має вид [4, 5, 7]

$$\frac{dP}{ds} = F_{mp}, \quad \frac{P}{\left[R + r\left(1 - \delta_1\right)\right]} = N, \quad N = bE_1\delta_1, F_{mp} = fN^n, \; f = \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{a}{b\phi^{n_1}}, \tag{5.14}$$

де P - натяг нити для направляючій у формі тора;

F_{mp} - сила тертя, яка діє на нескінченно малого елементу 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 направляючій у формі тора та умов взаємодії між ними;

E_1 - модуль пружності нитки при стисненні на направляючій поверхні у формі тора [2] .

З системи (5.14), розділяючи змінні, отримаємо наступне диференційне рівняння

$$\frac{d}{ds}\left\{\left[R + r(1 - \delta_1)\right]bE_1\delta_1\right\} = \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{a}{b\phi^{n_1}} \left(bE_1\right)^n \left(\delta_1\right)^n.$$

Проінтегруємо останнє диференціальне рівняння. В остаточно вигляді отримаємо формулу у випадку радіального охоплення

$$\left(\delta_1\right)^{1-n} = \left(\delta_{01}\right)^{1-n} + \frac{4\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} \frac{\left(1-n\right)a(bE_1)^{n-1}}{b\left(1-n_1\right)} \phi^{1-n_1}. \tag{5.15}$$

При $n \to 1$, отримаємо остаточно

$$\delta_1 = \delta_{01} e^{\frac{4\sin(\frac{\delta}{2})}{\delta+\sin(\delta)} \frac{a\phi^{1-n_1}}{b(1-n_1)\left[bE_1(\delta_0)\right]^{1-n}}}. \tag{5.16}$$

Для випадку, коли коефіцієнт тертя для направляючій у формі тора є постійною величиною, враховуючи рівняння (5.14), при $n_1 = 0$, отримаємо

$$\delta_1 = \delta_{01} e^{\frac{4\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{a}{b}\frac{\phi}{\left[bE_1(\delta_0)\right]^{1-n}}}. \qquad (5.17)$$

Підставляємо (5.17) в перше рівняння системи (5.14), після інтегрування будемо мати

$$P = P_0\left[1+\frac{(R+r)}{\left[R+r(1-\delta_{01})\right]}\left(e^{\frac{4\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{a}{b}\phi}-1\right)\right]+\left[\frac{B_0}{2\left[R+r(1-\delta_1)\right]^2}\right]- \\ -\left[\frac{B_0}{2\left[R+r(1-\delta_{01})\right]^2}\right]\left[1+\frac{(R+r)}{\left[R+r(1-\delta_{01})\right]}\left(e^{\frac{4\sin(\frac{\delta}{2})}{\delta+\sin(\delta)}\frac{a}{b}\phi}-1\right)\right]. \qquad (5.18)$$

Кут φ, для направляючій у формі тора, можна визначити з використанням формул

$$\varphi_{см1} = \arccos\left[1-\delta_{01}\left(\frac{2r}{R}\right)^2\right], \quad \varphi_{см2} = \arccos\left[1-\delta_1\left(\frac{2r}{R}\right)^2\right], \\ \varphi_{ижс1} = \arccos\left[1-\frac{B_0}{2P_0(R+r)^2}\right], \quad \varphi_{ижс2} = \arccos\left[1-\frac{B_0}{2P(R+r)^2}\right]. \qquad (5.19)$$

Рівняння (5.18) необхідно вирішувати з використанням систем (5.16), (5.19). Отримана залежність для направляючій у формі тора представляє трансцендентне рівняння відносно P. Для його вирішення використовувалися чисельні методи з використанням алгоритму дихотомії (ділення відрізку навпіл). Для цього було розроблене спеціальне програмне забезпечення.

В таблиці 5.1 представлені результати розрахунків по (5.18) залежності натягу ведучої гілки нитки P для: поліамідної комплексної нитки (ПАН) 174 Т (1 крива), B=1,3сН·мм^2, P_0=10 сН, r=0,24 мм, R=0,5 - 10 мм, φ_p=3,14 рад, n=0,82, δ= 0,488рад; вовняної пряжі (ВП) 93,5 Т (2 крива), B=0,3 сН·мм^2, P_0=10 сН, r=0,2 мм, R=0,5 - 10 мм, φ_p=3,14 рад, n=0,85, δ= 0,488 рад; бавовняної пряжі (БП) 100Т(3 крива),

B=2,5сН·мм^2, P_0=10 сН, r=0,21 мм, R=0,5 - 10 мм, φ_p=3,14 рад, n=0,78, δ= 0,488 рад.

По даним таблиці 5.1 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.5.5.

Аналіз графічних залежностей на рис.5.5 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 5.6 мм, для віскозної штапельної пряжі 93,5 Т коли радіус дорівнюватиме 3.8 мм та для бавовняної пряжі 100Т коли радіус дорівнюватиме 5.9 мм.

Таблиця 5.1

Залежність натягу нитки після спрямовувача у формі тора в залежності від вхідного натягу та радіусу кривизни поверхні тора в зоні контакту

R, мм	*Натяг нитки після спрямовувача у формі тора P, сН* $\delta=28^0$								
	Натяг веденої гілки P_O*=5 сН*			*Натяг веденої гілки* P_O*=10 сН*			*Натяг веденої гілки* P_O*=15 сН*		
	ПАН	*ВП*	*БП*	*ПАН*	*ВП*	*БП*	*ПАН*	*ВП*	*БП*
5.01	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	25.59	25.36
4.01	8.87	7.7	8.45	17.27	15.16	16.87	25.59	25.57	25.31
5.21	8.98	7.76	8.52	17.37	15.22	16.91	25.62	25.6	25.27
5.61	9.01	7.78	8.54	17.42	15.24	16.93	25.66	25.62	25.28
6.01	9.05	7.8	8.56	17.46	15.27	16.95	25.7	25.65	25.29
7.21	9.15	7.85	8.62	17.6	15.34	17.02	25.85	25.73	25.35
7.61	9.18	7.86	8.63	17.64	15.36	17.04	25.91	25.75	25.37

8.01	9.21	7.88	8.65	17.69	15.39	17.06	25.96	25.78	25.4
9.61	9.33	7.93	8.71	17.86	15.47	17.15	26.18	25.9	25.49
10.01	9.35	7.95	8.72	17.91	15.49	17.17	26.23	25.92	25.52
11.21	9.43	7.98	8.76	18.03	15.56	17.22	26.38	23	25.59
15.01	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.

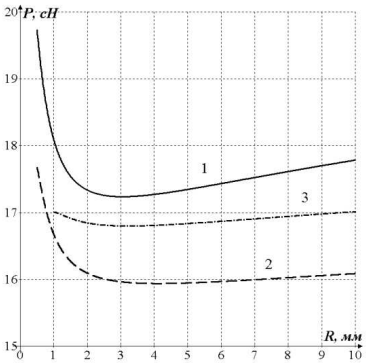


Рис.5.5. Залежності натягу ведучої гілки нитки P в залежності від натягу веденої гілки P_O, радіусу кривизни спрямовувача у формі тора R та кута радіального охоплення δ

5.3. ВИЗНАЧЕННЯ ПРИВЕДЕНОГО КОЕФІЦІЄНТУ ТЕРТЯ ДЛЯ СТРУКТУРНИХ ЕЛЕМЕНТІВ СИСТЕМИ ПОДАЧІ НИТКИ КРУГЛОВ'ЯЗАЛЬНИХ МАШИН У ФОРМІ ТОРА

Недосконаліть 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 елементу 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]

$$f = 4\frac{\sin(\frac{\delta}{2})}{\delta + \sin(\delta)} f_0, \tag{5.20}$$

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.5.6 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.

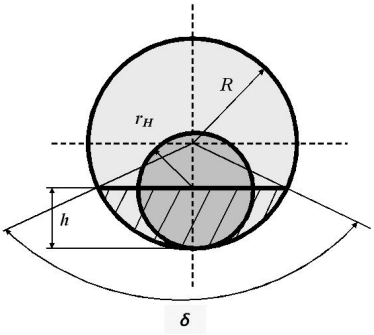


Рис.5.6. Загальна розрахункова схема

Площа сегмента буде визначатися за формулою

$$S_C = R^2 \arccos(\frac{R-h}{R}) - (R-h)\sqrt{2Rh-h^2}, \qquad (5.21)$$

де S_C - площа сегмента;

R - внутрішній радіус кільцевих та трубчастих спрямовувачів;

h - висота сегмента.

Площа сегмента буде визначатися за формулою

$$S_C = \pi r_H^2, \qquad (5.22)$$

де - r_H - радіус поперечного перетину нитки.

З урахуванням рівняння (5.3.3) рівняння (5.3.2) буде мати вигляд

$$\pi r_H^2 = R^2 \arccos(\frac{R-h}{R}) - (R-h)\sqrt{2Rh-h^2} \cdot \qquad (5.23)$$

Рівняння (5.23) представляє собою трансцендентне рівняння відносно h. Для його вирішення необхідно використовувати чисельні методи. Нами був обраний метод дихотомії.

Цільова функція буде мати вигляд

$$g = \pi r_H^2 - R^2 \arccos(\frac{R-h}{R}) - (R-h)\sqrt{2Rh-h^2}, \qquad (5.24)$$

де g - значення цільової функції.

Перейдемо до розгляду методу ділення відрізку $[h_i, h_{i+1}]$ 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

$$x_i = \frac{h_i + h_{i+1}}{2},\ g_i = \pi r_H^2 - R^2 \arccos(\frac{R - x_i}{R}) - (R - x_i)\sqrt{2Rx_i - x_i^2} \rangle 0,$$
$$h_{in} \le \xi,\ h_{i+1n} \ge \xi,\quad n = 1, 2....$$

де ξ - корінь трансцендентного рівняння (5.23) 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

$$\delta = 2\arccos(\frac{R - h}{R}) \cdot \qquad (5.25)$$

Результати розрахунків за формулами (5.20), (5.23), (5.25) представлені в таблицях 5.2, 5.3 та на рис.5.7.

В результаті проведеного дослідження розроблені математичні моделі для визначення приведеного 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.

Таблиця 5.2

Результати визначення висоти h сегмента заповненого матеріалом нитки

r_H	R=1.5	R=2	R=5.5	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

Для розв'язання трансцендентного рівняння (5.24) розроблений алгоритм методу половинного розділення відрізку при розв'язанні трансцендентних рівнянь та програмне забезпечення для визначення висоти сегменту заповненого матеріалом нитки, радіального кута охвату поверхні нитки напрямною поверхнею та приведеного коефіцієнту тертя.

Встановлені графічні залежності приведеного коефіцієнту тертя від співвідношення внутрішнього радіусу кільцевих та трубчастих спрямовувачів та розрахункового радіусі поперечного перетину нитки, що дозволяє оптимізувати геометричні параметри робочих органів системи подачі ниток на трикотажному обладнанні з позиції мінімізації натягу в робочій зоні.

Таблиця 5.3

Значення приведеного коефіцієнту тертя

r_H	R=1.5	R=2	R=5.5	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

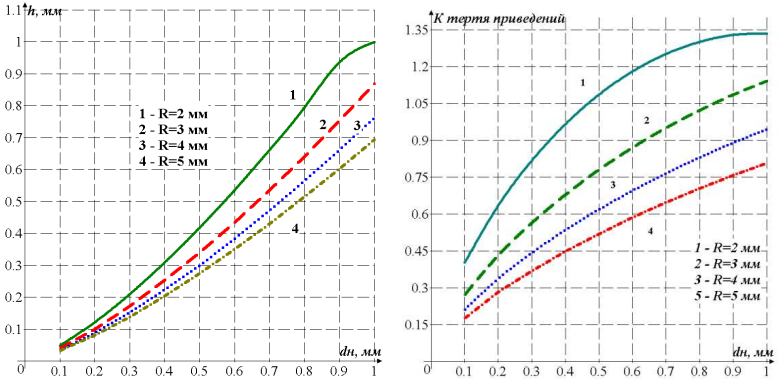
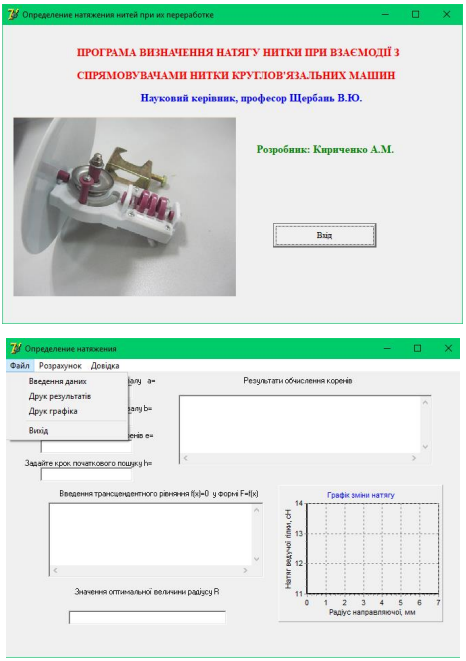


Рис.5.7. Графічні залежності приведеного коефіцієнту тертя та висоти h сегменту

5.4. 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.5.8а) розташовані наступні елементи: btn1NN1: TButton; lbl1NN1: TLabel; lbl2NN1: TLabel; lbl3NN1: TLabel; lbl4NN1: T Label; Image1: TImage.



а. б.

Рис.5.8. Перша форма Tfrm1NN1 = class (TForm) програми та основна форма програми для визначення натягу нитки після напрямної у формі тора

На основній формі програми Tfrm1NN2 = class(TForm) (рис.5.8б) 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: T Edit; 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. 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 (5.18) у формі $F = f(P)$. Багато строкове вікно mem1NN2: TMemo призначене для визначення результатів обчислень коренів трансцендентного рівняння (5.18). 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 Tfrm1NN5.n2NN2Click(Sender: TObject). При цьому активується форма Tfrm1NN4 = class(TForm). Ця форма призначена для введення необхідних даних для обчислення натягу нитки(рис.5.9).

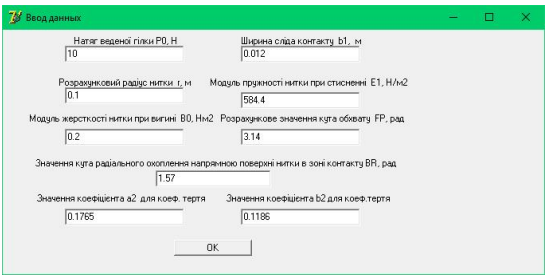


Рис.5.9. Форма Tfrm1NN4 = class(TForm) для введення даних для обчислення натягу нитки

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. 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 Tfrm1NN5.n7NN2Click(Sender: TObject) призначена для виконання розрахунку натягу нитки.

В якості алгоритму для визначення кореня трансцендентного рівняння (5.18) будемо використовувати метод ділення відрізку навпіл (метод дихотомії). На рис.5.10 представлена блок-схема алгоритму ділення відрізку 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

$$a_n \le \xi, b_n \ge \xi,\ n = 1, 2, \ldots.$$

де ξ - корінь трансцендентного рівняння (5.5.28) на відрізку $[a,b]$.

Будемо вважати, що $f(a)f(b)<0$ та $f(a)<0,\ f(b)>0$. Беремо 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 $[a,\xi_1]$ 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 відрізку

$$\xi_2=\frac{a_1+b_1}{2}.$$

Продовження процесу ділення дозволяє отримати ітераційну послідовність вкладених один в одного відрізків$[a_n,b_n]$. Необхідна система нерівностей має вигляд

$$a_n\le a_{n+1}<b_{n+1}\le b_n.$$

Аналіз останньої нерівності показує, що величина a_n визначає корінь ξ трансцендентного рівняння з лівого боку обраного відрізку, а величина b_n з правого боку обраного відрізку. Помилка при цьому не перевищує поточну довжину відрізку 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

$$N\simeq\frac{\ln(\frac{b-a}{\varepsilon})}{\ln 2}.$$

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 Tfrm1NN5.n7NN2Click(Sender: TObject) результати послідовно відображаються в багато строковому вікні mem1NN2: TMemo.

Виконання процедури procedure Tfrm1NN5.n8NN2Click(Sender: TObject) (рис.5.11а) 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 Tfrm1NN5.n9NN2Click(Sender: TObject).

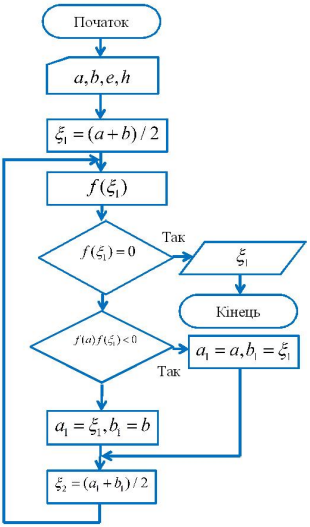


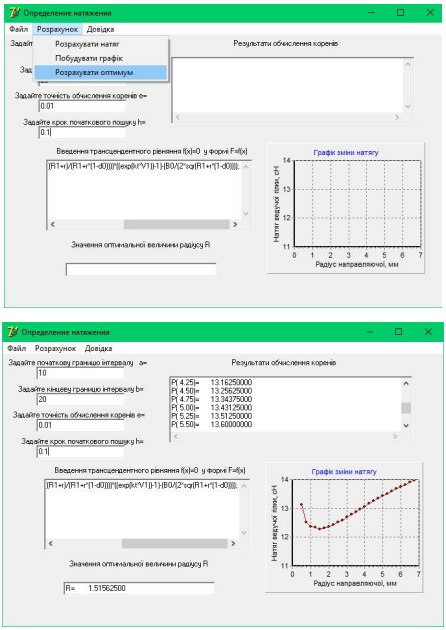
Рис.5.10. Блок-схема алгоритму ділення відрізку навпіл

Введення вхідних даних для реалізації алгоритму ділення відрізку навпіл здійснюється за допомогою наступних програмних операторів: Val (edt1NN5.Text,a,code); Val(edt2NN5.Text,b,code); Val(edt3NN5.Text,e,code); Val(edt4NN5.Text,h,code).

При введенні трансцендентного рівняння в багато строкове вікно Memo1: TMemo трансцендентного рівняння (5.18) 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 TF Errors = class(TForm).

На рис.5.11б представлені результати роботи програми для визначення величини натягу ведучої гілки нитки в залежності від радіусу кривизни напрямної великої кривизни у формі тора.



а. б.

Рис.5.11. Процедура procedure Tfrm1NN5.n8NN2Click(Sender: TObject) та результати роботи програми для визначення величини натягу ведучої гілки нитки

В додатку Б2 наведений лістинг програми для визначення значення натягу при взаємодії нитки з напрямною у формі тора з використанням зворотних польських записів.

5.5. АЛГОРИТМІЧНЕ ТА КОМП’ЮТЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ПРИВЕДЕНОГО КОЕФІЦІЄНТУ ТЕРТЯ ДЛЯ СТРУКТУРНИХ ЕЛЕМЕНТІВ СИСТЕМИ ПОДАЧІ НИТКИ

Програма для визначення приведеного 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.5.12а представлена перша форма TfrmTUDI1 = class(TForm) програми.

На цій формі розташовані наступні компоненти: lbl11TUDI: TLabel; lbl12TUDI: TLabel; lbl13TUDI: TLabel; lbl14TUDI: TLabel; btn11TUDI1: TButton; Label1: TLabel; Label2: TLabel; Label3: TLabel; Label5: TLabel; GroupBox1: TGroupBox; Image1: TImage.

Визначення приведеного коефіцієнту тертя включає три послідовні кроки: визначення висоти сегменту; обчислення радіального кута охоплення нитки поверхнею напрямної; визначення приведеного коефіцієнту тертя.

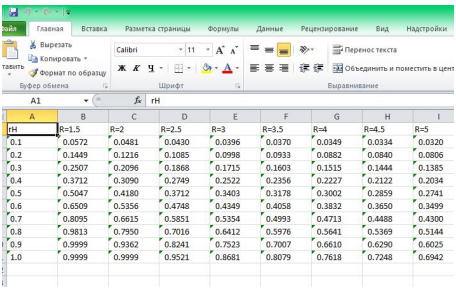
а. б.

Рис.5.15. Перша форма TfrmTUDI1 = class(TForm) програми для визначення приведеного коефіцієнту тертя та друга форма TfrmTUDI2 = class(TForm) для розрахунку висоти сегменту h

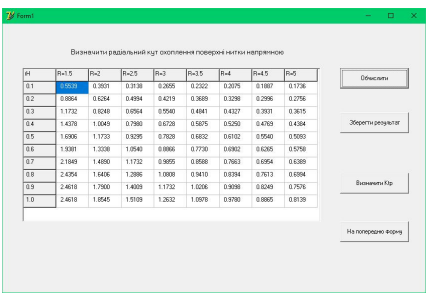
Друга форма TfrmTUDI2 = class(TForm) (рис.5.12б) призначена для розрахунку висоти сегменту h. Для чого необхідно з використанням чисельним методів знайти корені трансцендентного рівняння (5.24). В якості алгоритму для визначення кореня трансцендентного рівняння (5.24) використовували метод ділення відрізку навпіл. Необхідно побудувати ітераційну послідовність для цільової функції

$$g_i = \pi r_H^2 - R^2 \arccos(\frac{R - x_i}{R}) - (R - x_i)\sqrt{2Rx_i - x_i^2}\rangle 0 \cdot$$

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.5.13а).



rH	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



а. б.

Рис.5.13. 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.

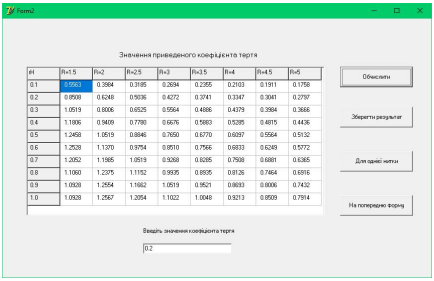
На формі TForm1 = class(TForm) (рис.5.13б) розташовані наступні компоненти: Button1: TButton; StringGrid1: TStringGrid; Button2: TButton; Button3: TButton; Button4: TButton; Label1: TLabel.

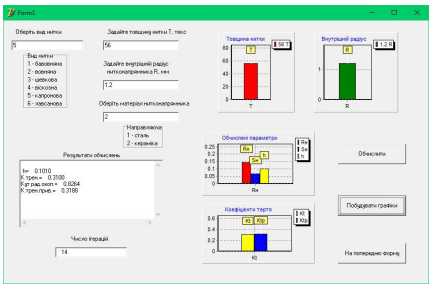
За допомогою компонента Button2: TButton визивається процедура procedure Button2Click(Sender: TObject) обчислення радіального кута охоплення нитки поверхнею напрямної у формі тора. Натиснення на кнопку Button3: TButton призводить до виконання процедури procedure Button3Click(Sender: TObject) збереження 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 приведеного 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.5.14а) розташовані наступні компоненти: Button1: TButton; Button2: TButton; StringGrid1: TStringGrid;

Button3: TButton; Button4: TButton; Label1: TLabel; Edit1: TEdit; Label2: TLabel.





а. б.

Рис.5.14. Четверта форма TForm2 = class(TForm) для визначення приведеного коефіцієнту тертя та п’ята форма TForm3 = class(TForm) для визначення приведеного коефіцієнту тертя для конкретної комплексної нитки чи пряжі

За допомогою компонента Button2: TButton визивається процедура procedure Button2Click(Sender: TObject) визначення приведеного 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 приведеного 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.5.14б) 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 тексах. 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 (5.24).

Багато строкове вікно Memo1: TMemo призначене для виведення результатів обчислень: висоти сегменту; радіального кута охоплення нитки поверхнею напрямної; приведеного 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, приведеного 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 приведеного коефіцієнту тертя при реалізації алгоритму дихотомії для трансцендентних рівнянь.

Висновки по розділу 5

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 диференційних 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 приведеного 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 відрізку при розв'язанні трансцендентних рівнянь та розроблений транслятор з використанням зворотних польських записів.

6. Розроблений алгоритм методу половинного розділення відрізку при розв'язанні трансцендентних рівнянь та програмне забезпечення для визначення висоти сегменту заповненого матеріалом нитки, радіального кута охвату поверхні нитки напрямною поверхнею та приведеного коефіцієнту тертя.

Список використаних джерел до розділу 5

1. 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.

2. 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.

3. Щербань В.Ю. Визначення приведеного коефіцієнту тертя для кільцевих та трубчатих спрямовувачів нитки трикотажних машин/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько//Вісник ХНУ.- 2017.-№6(255). - С.23-27.

4. Щербань В.Ю. Визначення натягу нитки при її взаємодії з трубчастими спрямовувачами/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, М.І.Шолудько//Вісник ХНУ.- 2018.-№1 (257). - С.213-217.

5. Щербань В.Ю. Взаємодія текстильних ниток з напрямними великої кривини у випадку наявності радіального охоплення/В.Ю.Щербань, Н.І.Мурза, А.М. Кириченко, Г.В. Мельник, М.І.Шолудько//Вісник ХНУ.- 2018.-№ 2 (259). - С.12-16.

6. Кириченко А.М. Математичні компоненти САПР спрямовувачів та робочих органів технологічного устаткування легкої та текстильної промисловості/ Інформаційні технології в науці, виробництві та підприємництві: Збірник наукових праць молодих вчених, аспірантів, магістрів кафедри інформаційних технологій проектування/ загал. наук. ред. В.Ю.Щербань – К.:Освіта України. - 2018. - С.22-26.

7. Кириченко А.М., Щербань В.Ю. Математичне моделювання процесу взаємодії ниток з напрямними великої кривини у випадку наявності радіального охоплення/ Тези доповідей XVII Всеукраїнської наукової конференції молодих учених та студентів «Наукові розробки молоді на сучасному етапі», 26-27 квітня 2018 року. Том 2., - К. КНУТД, с. 55-56.

8. Кириченко А.М., Щербань В.Ю. Використання рекурсивного підходу для розрахунку натягу ниток з напрямними великої кривини у випадку наявності радіального охоплення/ Тези доповідей XVIII Всеукраїнської наукової конференції молодих учених та студентів «Наукові розробки молоді на сучасному етапі», 18-19 квітня 2019 року. Том 2, ч. 1. - К. КНУТД, с. 74-75.

ДОДАТКИ

```
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 убуванню 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 убуванню 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[к,j];A[к,j]:=v; end; v1:=B[i];B[i]:=B[k];B[k]:=v1; end;
{ Блок перетворення матриці коефіцієнтів по убуванню 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) елементу 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 убуванню залежно від значення
модуля головного елементу 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 убуванню залежно від значення
модуля головного елементу 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[к,m]) then k:=j;
for z:=1 to N do begin v:=A[i,z];A[i,z]:=A[к,z];A[к,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[к,m])
then k:=j;   for z:=1 to N do   begin   v:=A[i,z];A[i,z]:=A[к,z];A[к,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[к,j]:=A[к,j]/A[до, до];   A[i,j]:=A[i,j]-A[i,k]*C[к,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[к,L];A[к,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[к,j]:=A[к,j]/A[до, до];
A[i,j]:=A[i,j]-A[i,k]*C[к,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[к,m]) then k:=j; END; for z1:=1 to N do begin
v:=A[i,z1];A[i,z1]:=A[к,z1];A[к,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 итераций',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,'-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[к,m]) then k:=j; END; for z1:=1 to N do

```
 begin  v:=A[i,z1];A[i,z1]:=A[к,z1];A[к,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 итераций',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[к,m1]) then k:=j;
END;  for z:=1 to N do  begin  v:=A[i,z];A[i,z]:=A[к,z];A[к,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[к,i] then  begin  if A[к,i]=0 then
begin   m:=1;L:=0;  end else  begin  m:=sqrt((A[i,i]*A[i,i])+(A[к,i]*A[к,i]));
```

```
L:=-A[к,i]/m;m:=A[i,i]/m; end;  end  else  begin
m:=sqrt((A[i,i]*A[i,i])+(A[к,i]*A[к,i]));  L:=-A[к,i]/m;m:=A[i,i]/m;
end;  for j:=1 to N do  begin  R:=(m*A[i,j])-L*A[к,j];
A[к,j]:=L*A[i,j]+(m*A[к,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 An(X^n)+An-1(X^N-
1)+...+A1X+Ao=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  Fi(Xi)=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 fi(Xi)=0 і
підставляємо праві частини отриманих рівнянь у вирази
F(i)=f(Xi)} 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  Fi(Xi)=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 fi(Xi)=0 і підставляємо праві частини отриманих
рівнянь у вирази
F(i)=f(Xi)}    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 dyi/dx=fi(x,yi)
 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 з[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 dyi/dx=fi(x,yi)
  ввиде F(i)=fi(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 dyi/dx=fi(x,yi)
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('Ввedіть 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)=a0+a1*x+a2*x^2+a3*x^3+...+an*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[к,L]; A[к,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[к,j]:=A[к,j]/A[до, до]; A[i,j]:=A[i,j]-A[i,k]*C[к,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(H);  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 стойки   ',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 стійкі <',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 стійкі   ',ja[nn]:2',
: ');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;    write('      7. Ордината стійкі
',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 стійкі
',jc[nn]:2',   : ');q[jc[nn],1]:=rea(q[jc[nn],1]);writeln;    write('      7. Ордината
стійкі   ',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  AB :');  writeln('    - для ABC
и AB1C1 - DE ;  - для AB2C2 и AB3C3 - DE1 .');  writeln(' 2. Выбрать знак
для поводка BC по скалярному произведению');  write('    DE и BC : для CB
и C2B2  { + }; для C1B1 и C3B3  { - }.');  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   0X  : ');
q[jd[nn],1]:=rea(q[jd[nn],1]);writeln;  write('                          <',je[nn]:2,'>   по
оси   0X  : ');  q[je[nn],1]:=rea(q[je[nn],1]);writeln;  write('
<',jd[nn]:2,'>   по оси   0Y  : '); q[jd[nn],2]:=rea(q[jd[nn],2]);writeln;  write('
```

<',je[nn]:2,'> по оси 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(' 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 AE кулисы ',ja[nn]:2,'-',je[nn]:2,' (со знаком) : '); p[nn,1]:=rea(p[nn,1]);writeln; write(' 3. Длина поводка BC ',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 AB и AC');
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 AB 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 (c-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(' |          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.
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. 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(Hag:5:1);  gotoxy(8,7);write('3. Общее
');textcolor(0);write('ЧИСЛО ШАГОВ');  textcolor(0);write(' исследования
: ');  writeln(Khag: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);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(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);wrie('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+}     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 исходных данных');     write('  не найден -
');textcolor(0+16);writeln('ОШИБКА !');     textcolor(7);     write(' Повторите
ввод <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('       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+}      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('      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
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('К И Н Е М А Т И К 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(' 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;
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('    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 indate;
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:{$i menuk2k };  if menuK=3 then
begin  clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
window(7,3,72,22);clrscr;  write('  ');for i:=1 to 62 do write('_');write(' ');  for
i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');  i2:=2;  for i1:=1 to n do begin if
i2>11 then begin;clrscr;  write('  ');for i:=1 to 62 do write('_');write(' ');  for i:=2
to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;  gotoxy(3,19);for
i:=1 to 62 do write('_');i2:=2;end;  gotoxy(1,i2);writeln;  case jg[i1] of  1:v1;
2:v2; 3:v3; 4:v4; 5:v5; 6:v6;  end{case};
i2:=i2+9;textbackground(3);textcolor(0+16);gotoxy(4,11);write('<ENTER>');
read;textbackground(1);textcolor(7);  end;{конец внутрен. цикла }
end;{конец всего цикла верификайции}  if menuK=4 then begin {$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
ДАННЫЕ  : ');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(Hag: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[],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*025+q[prinK[l],1];
AY:=q[prinK[l],2]*random*0.25+q[prinK[l],2];end;  end;  writeln(' |
',g1[]: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[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 {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                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('Функця 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[,1]/z;p2[2]:=p[i2]/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;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;{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,py, 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('|  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]:6:2);  write('|  2. Длина
```

```
звена ', jb[nd]:2,'-',jc[nd]:2);write(' : ');writeln(p[nd,2]:6:2);
write(' | 3. Знак схемы 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(' |            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. 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(Hag:5:1);
gotoxy(8,7);write('3. Общее ');textcolor(0);write('ЧИСЛО ШАГОВ');
textcolor(1);write(' исследования             : ');
writeln(Khag: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('       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+}     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 исходных данных');     write('   не найден -
');textcolor(0+16);writeln('ОШИБКА !');     textcolor(7);     write('
Повторите ввод <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(1);     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+}     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('       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(Hag:5:1);
gotoxy(8,7);write('3. Общее ');textcolor(0);write('ЧИСЛО ШАГОВ');
textcolor(1);write(' исследования          :  ');
writeln(Khag:2);textcolor(1);   gotoxy(8,8);write('4. Инициализация
');textcolor(0);   write('ПЕЧАТАЮЩЕГО УСТРОЙСТВА');
textcolor(1);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);              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('            Введите  УГОЛ
');      writeln('        приложения 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 выводить
графическую  ');        write('      информацию 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,l,292,l,2);draw(40,l+1,292,l+1,2);          for i:=1 to 36 do begin
ii:=trunc(fvn[i]/mgr);                              draw(jj,l,jj,l-ii,2);
draw(jj,l-ii,jj+7,l-ii,2);                          draw(jj+7,l-ii,jj+7,l,2);
if abs(ii)>2 then begin                             if ii>0 then fillshape(jj+3,l-
ii+1,1,2)                                           else fillshape(jj+3,l-ii-1,1,2);
end;                          jj:=jj+7;             end; {----------------------
----------}  read(kbd,ch);  gotoxy(7,24);writeln('Угол=
гр.');gotoxy(21,24);writeln('Нагрузка=       Н');
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,l+10,0);
plot(ii+1,l+10,0);                                  for m:=9 to 14 do
plot(ii,l+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,l+m,1);
plot(ii-1,l+10,1);plot(ii+1,l+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 внутрен. цикла }
end;{menu3<3}  end;{конец всего цикла верификайции}  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 };  {$i vvdiada };  {$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;end;   {   Динамика
табулирование }
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];                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(lst,'|',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,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. Номерів кінематичних');
   write('         пара для  <A> і <B>     : ');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(' Зміните 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 стійкі <',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 стійкі   ',ja[nn]:2',   :
');q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;
   write('      7. Ордината стійкі   ',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 стійкі   ',jc[nn]:2',   : ');q[jc[nn],1]:=rea(q[jc[nn],1]);writeln;
write('      7. Ордината стійкі   ',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  AB :');  writeln('    - для ABC і
AB1C1 - DE ;  - для AB2C2 і AB3C3 - DE1 .');  writeln(' 2. Вибрати знак для
повідця BC по скалярному твору');
 write('    DE і BC : для CB і C2в2  { + }; для C1B1 і C3в3  { - }.');  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 ЕD куліси ',je[nn]:2,'-',jd[nn]:2', (із знаком) : ');
p[nn,3]:=rea(p[nn,3]);writeln;
 writeln(' 5. Позначення стійкі <',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 стійкі   ',ja[nn]:2',   : ');
 q[ja[nn],1]:=rea(q[ja[nn],1]);writeln;  write('              7. Ордината стійкі
',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 стійкі   ',jc[nn]:2',   : ');
q[jc[nn],1]:=rea(q[jc[nn],1]);
 write('            7. Ордината стійкі   ',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 AB і AC');
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 AB 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(' Зміните 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 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;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
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 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 стійкі  ',jd[nd]:2',
: ');
  writeln(q[jd[nd],1]:7:2);  gotoxy(15,7);write('2. Ордината стійкі
',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 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 стійкі      ',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 стійкі      ',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(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 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);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(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 (Ox - Oy) : ');
  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 графика"');
gotoxy(14,7);writeln('P - включення режиму "побудова графика"');
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 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('      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('ДО І Н Е М А Т І ДО 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(' ДО 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('    Вводитимете 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 indate;
 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 ассура};
10:{$i menuk2k };
if menuK=3 then begin
clrscr;textbackground(5);window(1,1,80,25);clrscr;textbackground(2);
window(10,4,74,23);clrscr;textbackground(1);textcolor(7);
 window(7,3,72,22);clrscr;
write('  ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');
i2:=2;
for i1:=1 to n do begin if i2>11 then begin;clrscr;
write('  ');for i:=1 to 62 do write('_');write(' ');
for i:=2 to 19 do begin gotoxy(2,i);write('|');gotoxy(65,i);write('|');end;
gotoxy(3,19);for i:=1 to 62 do write('_');i2:=2;end;
gotoxy(1,i2);writeln;
case jg[i1] of
1:v1; 2:v2; 3:v3; 4:v4; 5:v5; 6:v6;
end{case};
i2:=i2+9;textbackground(3);textcolor(0+16);gotoxy(4,11);write('<ENTER>');
read;textbackground(1);textcolor(7);
end;{ кінець внутрен. циклу }
end;{ кінець всього циклу верификайции}
if menuK=4 then begin {$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 интерпритация 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 з[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 з[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(lst' | ',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(lst' |');for i:=1 to 63 do write(lst,'_');
  writeln(lst,'|');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 з[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 з[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[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>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 числельного 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 ïîçèöèè '+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('Îïçèöèÿ '+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 TFErrors = class(TForm)
LBErrors: TListBox; private { Private declarations } public
{ Public declarations } end; var FErrors: TFErrors; implementation {$R *.dfm}
```

Програма для апроксімації та інтерполяції за методом найменьших 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 M 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(TFErrors, FErrors);
FErrors.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' {FErrors};
{$R *.res} begin  Application.Initialize; Application.CreateForm(TfrmTUDI1,
frmTUDI1); Application.CreateForm(TfrmTUDI2, frmTUDI2);
Application.CreateForm(TFErrors, 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(TFErrors,
FErrors); 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 TFErrors = class(TForm)
LBErrors: TListBox;   private { Private declarations } public { Public
declarations } end; var FErrors: TFErrors; 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' {FErrors}; {$R *.res} begin Application.Initialize;
Application.CreateForm(TfrmSTUMN1, frmSTUMN1);
Application.CreateForm(TfrmSTUMN2, frmSTUMN2);
Application.CreateForm(TFErrors, FErrors); Application.Run; unit STUMN1;
```

```
Interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, jpeg, ExtCtrls, StdCtrls; type
TfrmSTUMN1 = class(TForm) lbl1STUMN1: TLabel; lbl2STUMN1: TLabel;
lbl3STUMN1: TLabel;     lbl4STUMN1: TLabel; btn1STUMN1: TButton;
lbl5STUMN1: TLabel; img1STUMN1: TImage; lbl6STUMN1: TLabel;
procedure btn1STUMN1Click(Sender: TObject); private { Private declarations }
public { Public declarations } end; var frmSTUMN1: TfrmSTUMN1;
implementation uses STUMN2; {$R *.dfm}
procedure TfrmSTUMN1.btn1STUMN1Click(Sender: TObject);
begin frmSTUMN1.Hide; frmSTUMN2.Show; end; end. unit STUMN2;
interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs,Math, StdCtrls; type TfrmSTUMN2 = class(TForm)
btn1STUMN2: TButton; lbl1STUMN2: TLabel; edt1STUMN2: TEdit;
mem1STUMN2: TMemo; btn2STUMN2: TButton; lbl2STUMN2: TLabel;
edt2STUMN2: TEdit; lbl3STUMN2: TLabel; edt3STUMN2: TEdit;
Memo1: TMemo; lbl4STUMN2: TLabel; btn3STUMN2: TButton;
lbl5STUMN2: TLabel; edt4STUMN2: TEdit; lbl7STUMN2: TLabel;
procedure btn1STUMN2Click(Sender: TObject);
procedure btn2STUMN2Click(Sender: TObject);
procedure btn3STUMN2Click(Sender: TObject);   private
{ Private declarations } public { Public declarations } end; type
xf=array [1..50]of Real; var frmSTUMN2: TfrmSTUMN2;
N,code,s1,M,R:integer; e,h,x1,g:Real; A:array[1..50,1..50] of Real;
B:array[1..50]of Real; F,X:xf; x0,xx:array[1..50] of string;   ii,ss,xxx:string;
Implementation uses STUMN1,Synt,UErrors; {$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(TFErrors, FErrors);
FErrors.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 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; OpenDialog1: 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 = ^ElList;    //список окружностей    ElList = 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; //используеться 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;
//перенумерация
 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:=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);
 //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:=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);
//---------------------------------------------------------------------------
//внутренние касательные
 sn_alpha:=(R1+R2)/d; cs_alpha:=sqrt(1-sn_alpha*sn_alpha);
//внутрення 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:=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);
 //внутрення 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 запятые
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 оси 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 оси 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 ценрами окружностей
```

```
 i:integer;
 cross:boolean;   //для проверки 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;
//проверка 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.Image1MouseUp(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 экрана (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);
//конроль 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;  //конроль 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;
//выбиаем 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);
```

```
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(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 opendialog1.Execute and fileexists(opendialog1.FileName) then
begin assignfile(f,opendialog1.FileName); try reset(f); except
showmessage('Невозможно открыть файл'+ opendialog1.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(ElRecord1); //добавление новой окружности (новой записи) в список
if (num='s')or(num='t') then ElRecord1.elnumber:=num
else begin num_el:=num_el+1; ElRecord1.elnumber:=inttostr(num_el);
end; ElRecord1.x_c:=xc; ElRecord1.y_c:=yc; ElRecord1.R:=R;
MyListE.Add(ElRecord1); 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 doif
```

```
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.el2<>'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(Memo1.Text) then begin // При ошибках
трансляции показывается форма с сообщениями
Application.CreateForm(TFErrors, FErrors);
FErrors.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);
1[ug]:=(r1+r2)*(ugol1[ug]+ugol2[ug])+sqrt(h1[ug]*h1[ug]+d1[ug]*d1[ug])+sqr
t(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('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  (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
подряд 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[i1].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  TFErrors = class(TForm)
LBErrors: TListBox;  private  { Private declarations }  public  { Public
declarations }  end; var  FErrors: TFErrors; implementation {$R *.dfm} end.
```

Програма для визначення приведеного 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,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 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('K трен.='+ktren1);
Memo1.Lines.Add('Кут рад.охоп.='+ug1); Memo1.Lines.Add('K
трен.прив.='+ktrp1); end; procedure TForm3.Button3Click(Sender: TObject);
begin Series1.Add(teks,'T',clRed); Series2.Add(rad,'R',clGreen);
Series3.Add(rn,'Rн',clRed); Series4.Add(sn,'Sн',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}; {$R *.res} begin Application.Initialize; Application.CreateForm(Tfrm1NN1, frm1NN1); Application.CreateForm(Tfrm1NN2, frm1NN2); Application.CreateForm(Tfrm1NN3, frm1NN3); Application.CreateForm(Tfrm1NN4, frm1NN4); Application.CreateForm(TFErrors, FErrors); Application.Run; end. unit NN1; interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,Math, StdCtrls, jpeg, ExtCtrls; type Tfrm1NN1 = class(TForm) btn1NN1: TButton; lbl1NN1: TLabel; lbl2NN1: TLabel; lbl3NN1: TLabel; lbl4NN1: TLabel; Image1: TImage; procedure btn1NN1Click(Sender: TObject); private { Private declarations } public { Public declarations } end; var frm1NN1: Tfrm1NN1; implementation uses NN2,NN3,NN4; {$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
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;  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; // Аналог proc1 для
функций begin  SetLength(TrStack,High(TrStack)+2);  TrStack[High(TrStack)]
:= Chr(127+Round(SItem.Number)); end;  procedure proc6; // Аналог proc3
для функций 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));
SItem.mode := None; end else ErrorList.Add(' '+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)]); 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 TFErrors = class(TForm) LBErrors: TListBox; private { Private declarations } public { Public declarations } end; var FErrors: TFErrors; implementation uses NN2; {$R *.dfm} end.

Наукове видання

ЩЕРБАНЬ В.Ю., КОРОГОД Г.О.,
ЧУПРИНКА Н.В., ВОЛІВАЧ А.П.,
КИРИЧЕНКО А.М.

КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ АЛГОРИТМІЧНИХ ТА ПРОГРАМНИХ КОМПОНЕНТІВ ПРИКЛАДНИХ ЗАДАЧ СИСТЕМ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

Підписано до друку 24.10.2021 р.
Формат 60x84/16. Папір офсетний.
Ум. друк. арк. 37,49
Наклад 200 прим.

ФО-П Маслаков Руслан Олексійович
Свідоцтво про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
ДК №4726 від 29.05.2014 р.
Тел. (095) 699-25-20.
E-mail: osvita2005@gmail.com.

ВД «Освіта України»™
Видавничий дім «Освіта України» запрошує авторів до співпраці
з випуску видань, що стосуються питань управління,
модернізації, інноваційних процесів, технологій, методичних
і методологічних аспектів освіти та навчального процесу
у вищих навчальних закладах.
Надаємо всі види видавничих та поліграфічних послуг.