

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**«Експертна система декомпозиції
складного контура»**

Завідувач

випускаючої кафедри

А. С. Довбиш

Керівник роботи

І. В. Шелехов

Студентка групи ІН.мз-92с

О. А. Шовкопляс

СУМИ 2021

Сумський державний університет

(назва вузу)

Факультет ЦЗДВФН Кафедра Комп'ютерних наук
Спеціальність «Інформатика»

Затверджую _____

Зав. кафедри Довбиш А. С.

« ____ » _____ 20 ____ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТЦІ

Шовкопляс Оксані Анатоліївні

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Експертна система декомпозиції складного контура
затверджую наказом по університету від « ____ » _____ 20 ____ р.
№ _____

2. Термін здачі студентом закінченого проєкту (роботи) _____

3. Вхідні дані до проєкту (роботи) _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми. Постановка задачі дослідження. 2) Математичний аналіз оптичних спектрів поглинання. Розроблення методики і алгоритму декомпозиції складного контура 3) Розроблення інформаційного та програмного забезпечення декомпозиції складного контура

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Термін виконання проєкту (роботи)	Примітка
1.	Аналіз проблеми. Постановка задачі дослідження	19.10–25.10	
2.	Математичний аналіз оптичних спектрів поглинання	26.10–06.11	
3.	Розроблення методики і алгоритму декомпозиції складного контура	07.11–17.11	
4.	Розроблення інформаційного та програмного забезпечення декомпозиції складного контура	18.11–25.11	
5.	Оформлення пояснювальної записки до дипломної роботи	28.11–03.12	

Студентка–дипломниця

(підпис)

Керівник проєкту

(підпис)

РЕФЕРАТ

Записка: 78 стор., 10 рис., 1 додаток, 44 джерела.

Об'єкт дослідження – комп'ютерні методи обробки результатів вимірювань.

Мета роботи – на основі чисельних методів розробити методикау і алгоритм аналізу експериментальних спектрів, створити експертну систему декомпозиції складних контурів.

Методи дослідження – чисельні методи, зокрема, методи оптимізації та метод найменших квадратів.

Результати – розроблений обчислювальний метод, алгоритм та програмне забезпечення системи декомпозиції складного контура. У Delphi-середовищі створена експертна система, яка реалізує розроблений алгоритм, та має зручні можливості для наочного спостереження результатів декомпозиції, збереження їх для подальших досліджень. Метод декомпозиції побудований на припущенні, що отриманий апаратурний спектр є дискретним і являє собою суперпозицію окремих характеристичних смуг. Була обрана така послідовність етапів для вирішення поставленої задачі: інтерполяція функціональних залежностей, заданих таблично; диференціювання експериментальних даних, декомпозиція спектрів, лінійний МНК; уточнення параметрів індивідуальних смуг.

АПАРАТУРНИЙ СПЕКТР, АПРОКСИМАЦІЯ, ДЕКОМПОЗИЦІЯ, ПАРАМЕТРИЗОВАНІ ФУНКЦІЇ, ФУНКЦІЯ ГАУССОВОГО ТИПУ, МЕТОД НАЙМЕНШИХ КВАДРАТІВ, BORLAND DELPHI

Зміст

ВСТУП	6
1 АНАЛІЗ ПРОБЛЕМИ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	7
1.1 Задача аналізу експериментального спектра.....	7
1.2 Математична модель апаратного спектра	13
1.3 Визначення параметрів характеристичних смуг шляхом статистичної обробки спектрального контура	14
1.4 Гауссова форма кривої в різних системах координат	17
1.5 Постановка задачі.....	20
2 МАТЕМАТИЧНИЙ АНАЛІЗ ОПТИЧНИХ СПЕКТРІВ ПОГЛИНАННЯ	21
2.1 Вихідні положення.....	21
2.2 Обробка експериментальних даних	22
2.3 Інтерполяція залежностей	24
2.4 Диференціювання експериментальних даних.....	26
2.5 Декомпозиція спектрів.....	27
2.6 Оцінка параметрів апроксимуючих функцій	30
2.7 Уточнення параметрів індивідуальних смуг	33
2.8 Умови однозначності декомпозиції складного спектрального контура	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	35
3.1 Інтерфейс та основні можливості програмного забезпечення	35
3.2 Деякі аспекти програмування	38
3.3 Тестування експертної системи	40
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК А ЛІСТИНГ МОДУЛІВ ЕКСПЕРТНОЇ СИСТЕМИ.....	50

ВСТУП

Серед сучасних методів вивчення явищ природи і проблем техніки різні фізичні методи займають важливе місце. Математична обробка результатів спостережень і вимірювань є невід'ємним етапом будь-якої дослідницької діяльності.

Фізичні методи є основними і при визначенні структури молекул. Для більш точного пояснення фізичних процесів в досліджуваних об'єктах необхідно складний спектр представити у вигляді окремих смуг. Найважливіша задача спектрометрії – аналіз дискретного спектра випромінювання або поглинання фізичних об'єктів на тому чи іншому рівні структури матерії – не розв'язується стандартно. У спектроскопічних дослідженнях часто зустрічаються контури, які складаються з окремих спектральних смуг, що перекриваються.

У теперішній час відомі і продовжують розроблятися методи декомпозиції складного контура. Ап'юрірна інформація визначає клас можливих рішень і робить задачу достатньо визначеною для математичного аналізу. Переважна більшість авторів (не тільки математиків) намагається утиснути задачу аналізу апаратного спектра у вузькі рамки певного розділу математичного аналізу або статистики. В дійсності ця задача є важливим прикладом нетрадиційного математичного підходу і вимагає створення адекватного їй формального апарату, тобто задача аналізу дискретного спектра є задачею загального факторного аналізу, і, таким чином, значення цієї задачі виходить за межі не тільки спектрометрії, але й фізики взагалі.

Існуючі професійні програмні продукти, бази даних спектрів, таких розробників як Aston Scientific (Birmingham, UK), Advanced Chemistry Development Inc., DigiLab Software GmbH (Germany), як правило, недоступні через їх високу вартість, тому актуальною є розробка власного програмного забезпечення, яке було б дієздатним для певного кола задач.

1 АНАЛІЗ ПРОБЛЕМИ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Задача аналізу експериментального спектра

Для широкого кола задач в області природничих наук характерний зв'язок з аналізом експериментальних даних, одержуваних у результаті проведення різноманітних вимірів. При цьому в процесі обробки спостережень експериментатор зазвичай будує деяку модель явища, що вивчається, намагаючись описати його обмеженим числом параметрів, подаючи зареєстрований складний сигнал композицією простих, більш зручних для аналізу. Тому часто в дослідженнях різної природи виникає питання про представлення експериментальних даних сукупністю функцій заданого виду (базисні, еталонні функції або функції порівняння). Звичайно, різноманітність об'єктів вимірів приводить до розмаїття видів базисних функцій, проте майже у всіх природничо-наукових дисциплінах позначається спільність наступного порядку. Справа в тому, що під час аналізу даних спостережень приходить часто зіштовхуватися з тим, що можливе нетривіальне подання сукупності експериментальних даних у вигляді суперпозиції еталонних функцій, що значно ускладнює розв'язання задачі. Цінним є залучення додаткової інформації, обумовленої специфікою конкретної галузі досліджень, це може значно допомогти у розв'язанні задачі розкладання спектра.

Для якісного і кількісного дослідження речовин широкого поширення набув спектральний аналіз [1]. Він заснований на вивченні спектрів випромінювання, поглинання, відбивання, люмінесценції, комбінаційного розсіювання світла. Серед галузей науки, в яких використовується спектральний аналіз – фізика, астрофізика, металургія, хімія, біологія, біохімія, геофізика, медицина тощо.

Велику зацікавленість і прикладне застосування представляють собою обернені задачі спектрального аналізу, які полягають в необхідності представлення вимірюного спектра сумою базисних (апаратурних) функцій заданого вигляду. До обернених некоректно поставлених задач приводять проблеми обробки і інтерпретації експериментальних спостережень, коли незначні зміни вихідних даних можуть спричинити істотні зміни результату. Такі задачі, як правило, не мають, традиційних рішень і потребують спеціальних методів побудови стійких наближених розв'язків [2].

Розробленням різних підходів до розв'язання задачі декомпозиції складного контура займаються науковці протягом багатьох років [3–18]. Відомі методи, що базуються на використанні для наближення експериментальних даних функцій типу періодичних, експонент, експоненціально згасаючих коливань, функцій гауссового, лоренцевого типів, типу Коші тощо [19]. При отриманні першого наближення практично завжди необхідне подальше уточнення отриманої теоретичної кривої з метою її якнайкращого узгодження з даними експерименту. Для цього можливим є використання певних стратегій, переважно заснованих на методі найменших квадратів з обмеженнями типу нерівностей, рівнянь, градієнтних методів, випадкового пошуку та ін.

У [20] поданий один з методів декомпозиції спектрів шляхом штучного виділення областей за кількістю спостережуваних смуг поглинання, побудови теоретичної кривої як суми гауссових або лоренцевих компонент, підрахунку суми квадратів відхилень, але потрібно зауважити, що не вказано, яким саме способом відбувалось подальше уточнення параметрів смуг. Це можна спостерігати в роботах багатьох дослідників.

У роботах [21–23] дослідження проводяться з використанням функцій гауссового типу, часто порівняльний аналіз результатів вказує на адекватність вибору саме такої форми спектральних характеристичних ліній. Особливо примітним є використання зваженої суми функцій лоренцевого і гауссового

типів [24]. У деяких працях подаються результати декомпозиції спектрів, але не згадується про використані методи і підходи до вирішення задачі [23–25]. Оптимізація розв'язку розглянута в задачах кусочно-поліноміальної апроксимації [26]. У роботі [27] охоплені різні математичні аспекти, пов'язані з проблемою аналізу основних типів даних, отриманих в експериментальній фізиці, – спектроподібних розподілів.

Найбільше методів розроблено для спектрів, контур яких описують функцією від однієї незалежної змінної Їх обробка полягає в дослідженні форми контура, вилучення зі спектра неінформаційної фонові складові, підбір параметризованих функцій, якими будуть змодельовані складові компоненти спектра і визначення їх інтенсивності [28].

Апаратурний спектр апроксимується теоретичною функцією з використанням чисельних методів. Залежно від апріорної інформації та поставленої задачі, це може бути інтерполяція поліномами, апроксимація сплайнами, раціональними функціями тощо. Коли отримана аналітична функція, яка описує вихідні дані з заданою точністю, можна проводити її математичну обробку і моделювати експериментальний спектр сумою параметризованих функцій. [29].

Однак, перш ніж розпочати аналіз конкретного спектрального контура, необхідно встановити, чи достатньо інформації для забезпечення умов єдиності розв'язку [30]. В деяких випадках можна отримати додаткову інформацію, яка цим вимогам задовольнить, або проаналізувати всю сукупність розв'язків, які можуть бути отримані, щоб відібрати з них найбільш адекватні, обґрунтовані певними фізичними або хімічними міркуваннями, а інколи і зовсім відмовитись від розв'язку задачі через її велику невизначеність.

Аналіз складного спектра, як правило, полягає у визначенні числа елементарних смуг, форми їх спектральних контурів і положень по шкалі частот (довжин хвиль). Згідно з принципом адитивності фізичних властивостей, фізична величина, яка характеризує молекулу в цілому, може

бути представлена простою сумою окремих значень величин тієї самої розмірності, що належать окремим структурним фрагментам: атомам, хімічним зв'язкам, групам атомів (цей принцип є абсолютно точним лише коли мова йдеться про молекулярну масу).

Експериментальний спектр часто складається з окремих смуг, що перекриваються. Причини перекривання можуть бути такими: невисока роздільна здатність спектральної апаратури, різного роду недоліки методики вимірювань, які не можуть бути усунені без втрати відношення сигналу до шуму. Навіть вважаючи цю некоректну задачу, за визначенням А. Н. Тихонова, вирішеною, залишається велика кількість випадків, коли перекриття обумовлене дійсною півшириною окремих компонент, і тоді отримання відомостей про їх число, положення та форму вимагає розв'язання задачі декомпозиції складного контура на його складові.

Однозначне представлення спектрального контура деяким числом елементарних смуг може бути визначено різними способами, з яких основними є два. Перший – знаходження максимуму інтенсивності, ширини на половині висоти і так далі, і другий – варіювання параметрів до досягнення мінімізації нев'язки між експериментальним та аналітичним спектром. Групи рішень відрізняються апріорною інформацією. У головній групі таких методів як початкову інформацію використовують припущення про форму окремих компонент спектра. Відомо, що у багатьох випадках виправданою є гауссова і лоренцева форма ліній [28].

Функції гауссового типу (1.1) – (1.3):

$$f(x, x_0, \delta, y_0) = \frac{y_0}{\sqrt{2\pi}\delta} \exp\left(-\frac{(x-x_0)^2}{2\delta^2}\right), \quad (1.1)$$

де x_0 – положення піка,

δ^2 – його дисперсія,

y_0 – інтенсивність;

$$f(x, x_0, h, y_0) = y_0 \exp\left(-\ln 2 \frac{(x - x_0)^2}{h^2}\right), \quad (1.2)$$

де x_0 – положення піка,
 h – півширина піка на половині його висоти,
 y_0 – висота піка;

$$f(x, x_0, H, y_0) = y_0 \exp\left(-\ln 2 \frac{(x - x_0)^2}{H^2}\right), \quad (1.3)$$

де $H = \begin{cases} h' \text{ при } x < x_0, \\ h'' \text{ при } x \geq x_0, \end{cases}$

x_0 – положення піка,
 h' , h'' – ліва і права півширини піка на половині його висоти відповідно,
 y_0 – висота піка.

Функція Лоренца (1.4):

$$f(x, x_0, a, H, y_0) = y_0 \frac{H}{2} \frac{0,5H + 2a(x - x_0)}{(0,5H^2) + (x - x_0)^2}, \quad (1.4)$$

параметри якої:

x_0 – положення піка,
 a – його асиметрія,
 H – ширина піка на половині його висоти,
 y_0 – значення функції в точці $x = x_0$.

Для знаходження параметрів, які характеризують форму і положення окремих компонент, найчастіше застосовують різні модифікації методу найменших квадратів і знаходять такі їх значення, за яких досягається найменша сума квадратів ухилень обчислених і експериментально отриманих значень функції, що описує спектральний контур. У деяких випадках для обчислення параметрів компонент складають систему алгебраїчних рівнянь,

праві частини яких знаходять або із значень інтеграла Фур'є контура, або з послідовних моментів функції контура.

При застосуванні методів математичної декомпозиції головну проблему становить питання про число окремих смуг, яке заздалегідь не відомо і само є істотною частиною всієї спектроскопічної інформації, яку потрібно отримати з експерименту. Дослідження і досвід використання програм, які реалізують метод найменших квадратів (МНК), свідчать, що дані про число компонент можна отримати тільки при достатньо великій роздільності компонент в спектрі та невеликому їх числі. У іншому випадку дані виявляються неоднозначними й залежать від характеру вихідного наближення. Ще одним недоліком існуючих методів декомпозиції є велика втрата точності визначення параметрів компонент через значний вплив випадкового шуму вихідних експериментальних даних.

Тому сучасними задачами спектрометрії є:

- 1) потрібна практична методика високоточного аналізу спектрів;
- 2) потрібна практична методика автоматичного (в математичному, не технічному розумінні) аналізу складних контурів, тобто з мінімальною участю людини в управлінні алгоритмом.

Кожне з цих вимог, а особливо у поєднанні, роблять задачу аналізу експериментального спектра нетривіальним розділом аналізу даних як науки.

Під визначенням будови хімічних сполук розуміють встановлення структурної формули, тобто природи і послідовності зв'язків між атомами в молекулі. Речовина, через яку пропускають суцільне випромінювання, поглинає саме ті промені, які воно ж випускає, якщо його нагріти до потрібної температури. Кожна речовина має власний набір спектральних ліній, за сукупністю яких можна виявити присутність цієї речовини. Сигнал, який несе інформацію про природу атомів або молекул, з яких складається речовина, з'являється в той момент, коли на об'єкт впливають відповідним чином, викликаючи той чи інший недвозначний результат.

Маючи задану структуру молекули і силові сталі можна за теорією коливань розрахувати контур. Але обернена задача, тобто розрахунок структури за даним спектром, вирішується не завжди. Щоб зробити висновок про склад і будову речовини за його спектром, доводиться залучати емпіричні та напівемпіричні закономірності. Певні сполуки з подібними елементами структури мають близькі частоти деяких нормальних коливань. Смути поглинання, які відповідають таких нормальним коливанням, називаються характеристичними смугами, а частоти їх максимумів поглинання – характеристичними частотами. Таким чином, певні особливості структури молекул речовини, яка поглинає світло, обумовлюють наявність характеристичних смуг в спектрі.

Структурний аналіз по спектрам наразі зводиться до виявлення характеристичних смуг і їх віднесення до відповідних структурних елементів з урахуванням числових значень частот максимумів поглинання, контура і інтенсивності смуг. За допомогою спектрального аналізу можна визначити практично усі елементи і сполуки в різноманітних речовинах різного агрегатного складу.

1.2 Математична модель апаратного спектра

У результаті вимірювання експериментатор отримує спостережуваний, або апаратний спектр $f(x)$. Аргумент x (довжина хвилі, частота, енергія, час, кут тощо) може бути залежно від типу реєстрації або неперервною, або дискретною величиною. В даний час дискретне зображення є універсальним, тому в даній роботі спираємось на нього.

Найпростіша модель спектра може бути побудована із залученням невеликої кількості фізичних знань. Отриманий апаратний спектр є сумою корисних, зайвих і шумових сигналів. Зайві (фонові) сигнали не цікавлять експериментатора в даний момент і мають детерміноване походження.

Шумові (неінформаційні) сигнали мають випадкове походження. І зайві, і шумові сигнали спотворюють корисні (інформаційні) сигнали.

$$f(x) = p(x) + b(x) + c(x) \quad (1.5)$$

де $p(x)$ – корисний сигнал,
 $b(x)$ – зайвий сигнал,
 $c(x)$ – шумовий сигнал [27].

У дискретній спектрометрії інформація міститься в функціях, заданих на невеликих проміжках значень аргументу. Просту, і разом з тим, всебічну фізичну інтерпретацію можна отримати з їх центральних моментів: положення, інтенсивність, півширина тощо. Тому корисна компонента $p(x)$ є об'єднанням елементарних смуг

$$p(x) = \sum_{i=1}^n I_i(x). \quad (1.6)$$

Форма окремих смуг, кількість, параметри, а також способи їх аналізу є основою для розробки різних методів математичного розкладання складних контурів на елементарні смуги.

1.3 Визначення параметрів характеристичних смуг шляхом статистичної обробки спектрального контура

Для багатьох задач спектрального аналізу потрібно точно знати положення і інтенсивність спектральних смуг, які належать центрам (молекулам) кожного сорту. В усіх випадках, коли вони перекриваються між собою, на практиці мають справу лише з деяким сумарним спектром. Зазвичай при розділенні такого контура на індивідуальні смуги припускають, що ці смуги мають гауссову форму, і постає задача тільки про найбільш простий спосіб знаходження їх параметрів. В сумарному спектральному контурі положення максимумів порівняно з положенням індивідуальних компонент, як правило, зміщується. Величина зміщення залежить від інтенсивності й

півширини сусідніх смуг, найчастіше відбувається повне зникнення максимумів [31].

Спектри, отримані на практиці, їх статистична обробка, мають скінчену точність, тому часто розкладання спектрального контура є неоднозначним [32]. Для аналізу складних контурів застосовують комп'ютерну обробку даних. Окремі смуги моделюють характеристичними функціями (гауссового типу або функцією Лоренца, або їх сумою). Параметри компонент варіюють за допомогою МНК з метою отримання найкращого співпадіння суми компонент с вихідним контуром. Існуючі методи такого роду для декомпозиції спектрів відрізняються між собою, в основному, прийомами, за допомогою яких автори досягають якнайшвидшої збіжності процесів ітерації. Перший з цих методів запропонований Стоуном [33], завжди забезпечує збіжність процесу, але в деяких випадках швидкість збіжності дуже низька. У роботі представлені два методи розкладання апаратного спектра на окремі смуги, кожна з яких має однакову відому форму. Перший описаний спосіб розкладання призначений для смуг однакової півширини і базується на властивостях перетворення Фур'є парних функцій. У другому методі розглянута ітераційна процедура МНК для смуг різної півширини за допомогою комп'ютерної програми.

Детальний огляд методів розкладання перекритих спектральних смуг зроблений Пітом і Джонсоном [34]. Проведено порівняння семи чисельних методів підгонки контурів інфрачервоних смуг з аналітичними функціями з використанням нелінійних наближень МНК. Як модельні функції використані функції Гауса та Коші (Лоренца), а також їх комбінації. Методи порівнювали як за ступенем конвергенції, так і за часом обчислення, необхідному для досягнення прийнятної точності. З більш складними спектрами можна вирішити модифікацію «рухомого підпростору», в якій одночасно регулюються лише параметри групи сусідніх смуг. Для розглянутих методів написані автоматичні комп'ютерні програми.

Існують методи статистичної обробки спектрального контура, які використовують виявлення прихованих періодичностей [35], тобто використання гармонічного аналізу. Наприклад, метод математичного розкладання перекритих гауссових або лоренцевих смуг однієї й тієї самої півширини, базується на перетворенні Фур'є апаратного складного контура. Нехай спектральна функція є сумою гауссових компонент вигляду

$$f(x) = \sum_{i=1}^n A \exp\left(-\frac{(x-x_0)^2}{\delta^2}\right). \quad (1.7)$$

Перетворення Фур'є цієї функції буде:

$$F(p) = \int_{-\infty}^{\infty} f(x) \cos px = \exp\left(-\frac{p\delta^2}{4}\right) \sum_{i=1}^n r_i \cos px, \quad (1.8)$$

де $r_i = \sqrt{\pi} \delta A_i$.

Таким чином, косинус-трансформація Фур'є контура, який є сумою гауссових компонент, це добуток гауссової функції, центрованої на початок координат, і суми гармонік, частоти яких визначаються положенням максимумів компонент в спектрі. У просторі перетворень Фур'є задача декомпозиції контура, що складається з гауссових компонент, еквівалентна задачі про знаходження амплітуди й частоти гармонічних компонент. Задача виявлення прихованих періодичностей має давню історію і для її розв'язання запропоновано багато різних методів [36].

Обчислення параметрів індивідуальних смуг пропонується і з використанням ряду Тейлора [37]. Спектр розкладається в ряд, сконструйований так, що коефіцієнти розкладання одночасно є характеристиками індивідуальних смуг. Такі характеристики A_{mn} , де n – номер характеристики, m – номер смуги, шукають у вигляді коефіцієнтів розкладання в ряд Тейлора

$$A_{nm}^{[j+1]} = \frac{1}{n!} \frac{d^n}{d\nu^n} f_m^{[j]}(\nu) \Big|_{\nu = \nu_{0m}}, \quad (1.9)$$

де $[j]$ – номер ітерації; ν_{0m} визначається з умови $A_{1m}^{[j]} = 0$, а функція $f(\nu)$ є експериментальним спектром $F(\nu)$, з якого відняли вклади «крил» усіх сусідніх з m -лінією й зробили інверсію, тобто,

$$f_m^{[j]}(\nu) = \left[F(\nu) - \sum_{l \neq m} \left(\sum_{n=0}^N A_{nl}^{[j]} (\nu - \nu_{0l})^n \right)^{-1} \right]^{-1}. \quad (1.10)$$

Тоді спектр є сумою інверсних поліномів

$$F(\nu) = \sum_{m=1}^M \left(\sum_{n=0}^N A_{nm} (\nu - \nu_{0m})^n \right)^{-1}. \quad (1.11)$$

Отже, для визначення параметрів спектральних смуг, можна реалізувати сукупність таких операцій: віднімання «крил» сусідніх смуг, інверсія інтенсивності, розкладання в ряд Тейлора.

Для покращання розділення в складному контурі широких окремих смуг, які дуже перекриваються, застосовують лінійні математичні методи, що загострюють ці смуги. Наприклад, метод деконволюції кривих [38].

Інформаційний огляд деяких з великої кількості існуючих методів дає уявлення про їх різноманітність залежно від цілей дослідників, про складнощі, проблеми та неоднозначність розв'язання цієї нетривіальної задачі.

1.4 Гауссова форма кривої в різних системах координат

З теоретичної точки зору залишається незрозумілим питання, чому модельною функцією найчастіше виступає гауссова, а не якась інша колоколоподібна крива. Форма смуги залежить від системи координат, в якій вона зображена. По осі абсцис відкладаються величини, пропорційні довжині

хвилі або частоті, а по осі ординат – величини, пропорційні спектральній густині енергії E_λ або E_ν , або N_ν . (Добуток $E_\lambda d\lambda$ – потужність, яка випускається тілом в інтервалі $d\lambda$; $E_\nu d\nu$ – потужність, яка випускається тілом в інтервалі $d\nu$; $N_\nu d\nu$ – число квантів, яке випускається за одиницю часу в інтервалі $d\nu$). Перехід від E_λ до E_ν потребує множення модельної функції на множник пропорційний λ^2 , а для переходу до N_ν необхідно функцію помножити на множник, пропорційний λ^3 . Очевидно, що такі переходи зміщують максимум кривої і дещо викривляють її форму. Ще більше викривляється при цьому форма складної кривої в масштабі по осі абсцис. На рис. 1.1, 1.2 зображені в двох системах координат три спектральні криві. В координатах (λ, E_λ) вони мають строго гауссову форму і відрізняються лише півшириною. У системі координат (ν, E_ν) їх максимуми виявились розташованими в різних місцях, півширина зменшилась, форма стала несиметричною.



Рис. 1.1. Модельні функції в координатах (λ, E_λ)

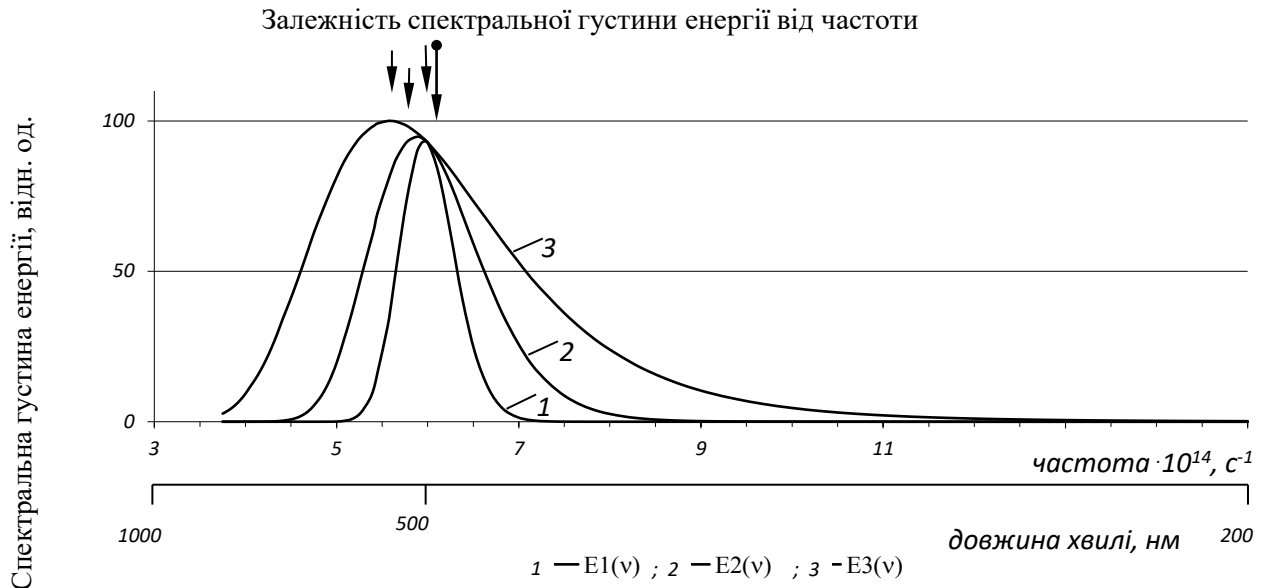


Рис. 1.2. Модельні функції в координатах (ν, E_ν)

Аналогічні зміни відбудуться з кривими, які мають гауссову форму в координатах (λ, E_λ) , якщо їх зобразити в координатах $(h\nu, N_\nu)$. Дещо краща ситуація зі спектрами поглинання. Оскільки величина, яка характеризує коефіцієнт поглинання, не змінюється при переході від одних координат до інших, множник λ^2 або λ^3 тут не потрібен. Тому положення максимуму кривої поглинання однакове в різних системах координат. Форма ж її зазнає майже такі самі зміни, як і форма кривої випромінювання при переході між різними системами координат.

Звідси випливає, що якщо немає жодних додаткових відомостей про форму індивідуальних смуг, то невідомо, в яких координатах необхідно будувати складний контур для розділення його на гауссові складові. Це особливо важливо для широких смуг, оскільки при переході до інших координат їх форма змінюється найбільше. Несиметрія смуги, побудованої в деяких координатах, у сукупності з її великою шириною може стати підставою віднести дану смугу до складної, хоча в інших координатах ця сама смуга може мати строго гауссову форму, тобто бути елементарною. З іншого боку,

теорія спектрів дає гауссову форму смуги лише при виконанні деяких достатньо жорстких вимог (електронні переходи не повинні мати фононних повторень, наявність достатньо низької температури, щоб усі центри, які випромінюють, були в основному коливальному стані тощо). Тому відсутні підстави вважати індивідуальну смугу, на думку деяких авторів [28], завжди гауссовою. Той факт, що отримані дослідні спектральні криві нерідко вдається апроксимувати гауссовими, пояснюється просто невисокою точністю вимірювань, а також тим, що при цьому обмежуються зазвичай лише частиною кривої, яка охоплює ділянку, трохи більшу за її півширину. Явно ж несиметричні смуги вважають неелементарними.

Між тим, для декомпозиції складного контура на індивідуальні складові необхідно враховувати і «крила» смуг, оскільки вони найбільше накладаються на сусідні смуги, а неправильне їх урахування може привести до помилки при знаходженні положення максимуму і ширини сусідньої смуги. В цих «крилах» відхилення від гауссової форми і несиметрія смуг найбільші. Тому не можна задовольнитись гарним співпадінням з гауссовою кривою лише середини індивідуальної смуги.

1.5 Постановка задачі

Використовуючи числові методи розробити методику декомпозиції складних контурів, побудувати відповідний алгоритм та на їх основі створити експертну систему в середовищі Borland Delphi, яка дозволить:

- створювати, зберігати, редагувати бази даних;
- проводити обробку складних контурів;
- візуально спостерігати за процесом декомпозиції;
- зберігати отримані параметри характеристичних компонент розкладеного спектра у текстовий файл подальшого використання;
- експортувати графік у формат .bmp тощо.

2 МАТЕМАТИЧНИЙ АНАЛІЗ ОПТИЧНИХ СПЕКТРІВ ПОГЛИНАННЯ

Для створення експертної системи декомпозиції складного контура розглянемо задачу із залученням мінімуму апріорної інформації. Початкова умова – наявність тільки апаратного експериментального дискретного спектра поглинання. Аналіз спектрів поглинання дозволяє отримати дані про атоми і молекули, які знаходяться в основному незбудженому стані. До переваг методів вимірювання спектрів поглинання належить можливість дослідження речовини в різних агрегатних станах і при різних температурах, зокрема, гелієві. Спектри поглинання успішно використовують для розв'язання різноманітних спектроаналітичних задач.

2.1 Вихідні положення

При поширенні випромінювання в середовищі воно селективно поглинається на власних частотах речовини середовища. В ультрафіолетовій і видимій областях спектра поглинання відбувається на частотах, пов'язаних з електронними переходами в атомах і молекулах, а в інфрачервоній області – на частотах коливальних і обертальних переходів молекул і власних частотах кристалічних решіток. У тих ділянках спектра, де речовина селективно поглинає, величина потоку випромінювання, який проходить, стає менше, утворюючи в спектрі випромінювання, яке падає на речовину, «провали», або смуги поглинання, які і складають спектр поглинання.

Вимірювання спектра поглинання зводиться до вимірювання залежності енергії потоку випромінювання, що пройшов через речовину, від довжини хвилі (або частоти). Аналізовану пробу розміщують між джерелом суцільного випромінювання і спектральним апаратом. Спектр, що характеризує речовину, складають ті довжини хвиль, інтенсивність яких зменшилася.

Фізичні методи визначення структури молекул широко розповсюджені у наукових дослідженнях. У спектроскопічних дослідженнях часто зустрічаються складні контури спектральних смуг, які необхідно розкласти на скінченну кількість спектральних ліній, які у сумі мають дати максимально наблизений до вихідного контур. Структурний аналіз по спектрах у даний час зводиться до відшукування характеристичних смуг і віднесенні їх до відповідних структурних елементів з урахуванням чисельних значень довжин хвиль максимумів поглинання, контуру та інтенсивності смуг.

У рамках даної роботи описана математична модель і розроблене програмне забезпечення у Delphi-середовищі для аналізу апаратурних оптичних спектрів поглинання.

У основі абсорбційних вимірів лежить закон Бугера-Ламберта-Бера:

$$I = I_0 \exp(-D), \quad (1.12)$$

де I – інтенсивність електромагнітного випромінювання досліджуваного зразка на певній довжині хвилі,

I_0 – інтенсивність електромагнітного випромінювання без досліджуваного зразка на тій самій довжині хвилі,

D – оптична густина.

Вихідний спектр величини $D = \ln(I_0/I)$, що досліджується, є дискретним. Для скінченного числа рівновіддалених довжин хвиль вимірюються величини I та I_0 , обчислюється логарифм їх відношення і таким чином отримуємо спектр оптичної густини D . За отриманими експериментальними результатами можна визначити, на яких довжинах хвиль досліджувана речовина поглинає більше.

2.2 Обробка експериментальних даних

На основі чисельних методів (п. 2.3–2.7) побудований обчислювальний алгоритм для розв’язання поставленої задачі (рис. 2.1).

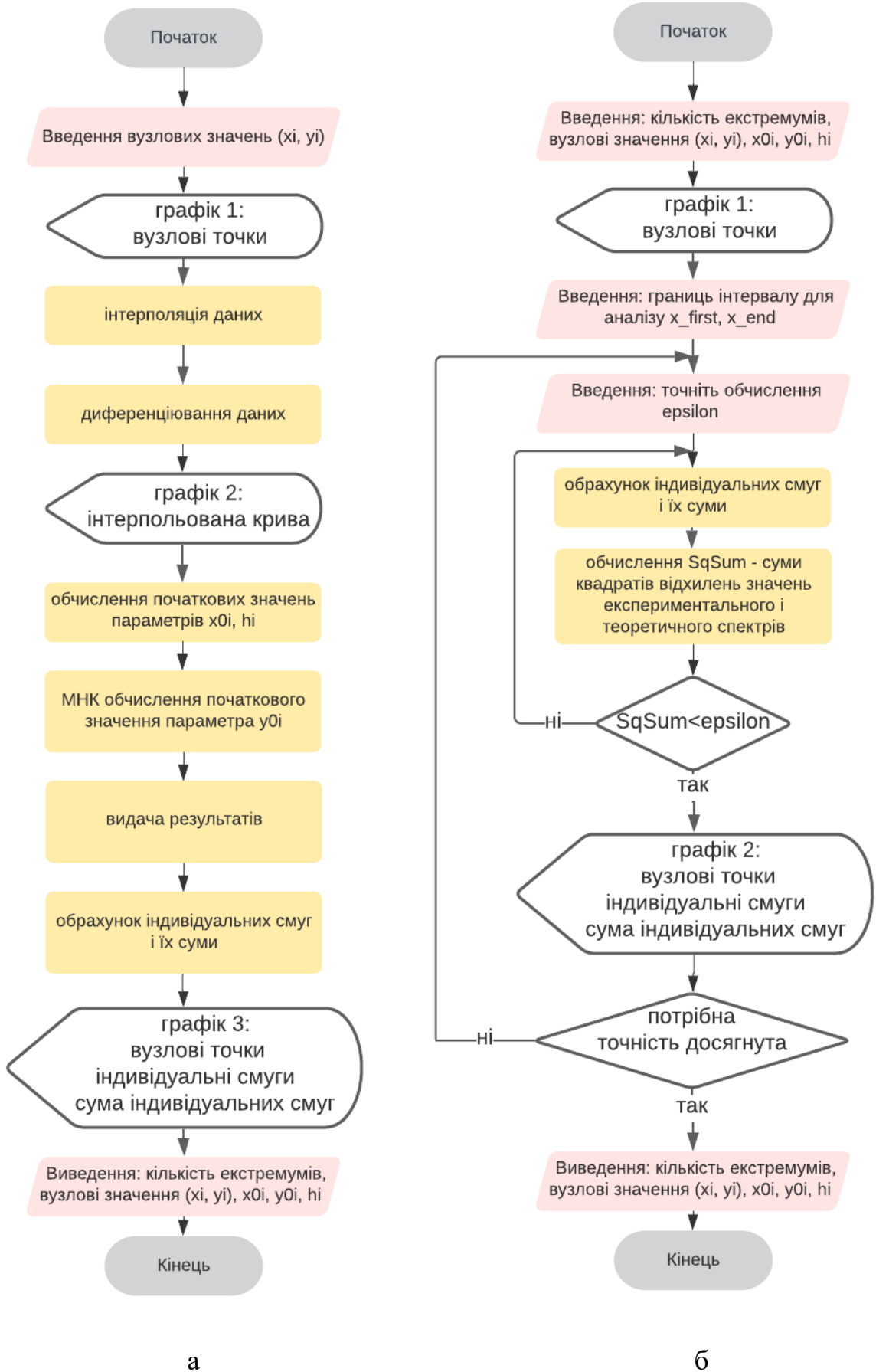


Рис. 2.1. Схема алгоритму декомпозиції контура: SpanLagr (а), Exacting (б)

Створена експертна система декомпозиції складного контура реалізує алгоритм двох логічних схем: SpanLagr (рис. 2.1a) – отримання першого наближення і Exacting (рис. 2.1б) – уточнення параметрів індивідуальних смуг.

Логічна схема аналізу експериментального контура показана на рис. 2.2. Об'єктом дослідження є дискретний експериментальний спектр. Дослідження форми спектральних смуг проводиться методом апроксимації ізольованих піків параметризованими функціями (п. 1.1). Математична модель є сумою функцій гауссового типу і поліному невисокого степеня, який описує фон.

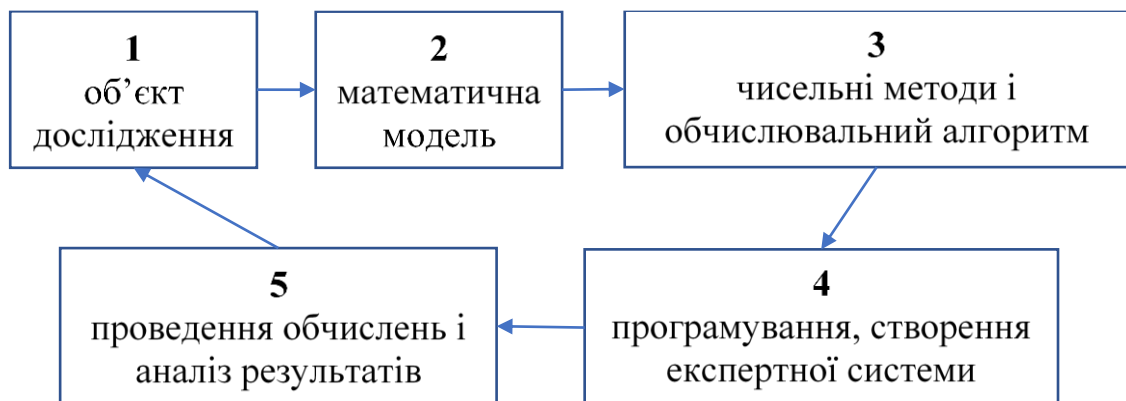


Рис. 2.2. Схема обчислювального експерименту

2.3 Інтерполяція залежностей

Припускаючи апріорі відомою форму індивідуальної смуги, перш, ніж розпочати декомпозицію складного дискретного спектра $f(x)$, його необхідно апроксимувати деякою функцією $p(x)$, яка має аналітичний вираз (лат. *approxīmo* – наближаюся).

Введену функцію використовують:

- для наближеного визначення числових значень $f(x)$;
- для проведення аналітичних викладок при теоретичному дослідженні моделі.

Постановка задачі: відомі результати вимірювання $y_k=f(x)$ деякої фізичної величини $f(x)$ в точках x_k , $k = 0-n$, і потрібно визначити її значення в інших точках.

Алгебраїчні і тригонометричні поліноми, які дають наближений вираз заданої неперервної функції $f(x)$, можна побудувати різноманітними способами. Це поліноми Бернштейна, частинні суми ортогональних розкладань функції $f(x)$, суми Фур'є, Фейєра, Валле–Пуссена тощо [39]. Серед усіх методів апроксимації виділяють інтерполяцію (inter – між, всередині; pole – вузол). Потрібно побудувати такий поліном $p(x)$, який у заздалегідь заданих точках («вузлах інтерполювання») співпадав би з функцією $f(x)$.

Якщо не накладати на функцію $f(x)$ ніяких обмежень, то її значення в вузлах (а тільки вони визначають поліном $p(x)$) не будуть ні в якій мірі пов'язані зі значеннями функції в інших точках. Тому збільшення числа вузлів ніяк не гарантує наближення полінома $p(x)$ до функції $f(x)$ в точках, відмінних від вузлів інтерполяції. Проте, якщо на функцію $f(x)$ накладені додаткові обмеження, то при вдалому виборі закону розширення множини вузлів можна отримати позитивні результати.

Інтерполяційна формула Лагранжа.

Нехай на проміжку $a \leq x \leq b$ задані точки $x_k, k = 0-n$ (вузли інтерполяції), в яких відомі значення функції $f(x)$:

$$x_0, x_1, x_2, x_3, \dots, x_n, \quad (1.13)$$

$$y_0, y_1, y_2, y_3, \dots, y_n, \quad (1.14)$$

причому усі числа (1.13) відрізняються між собою (на числа (1.14) така умова не накладається).

Потрібно побудувати поліном $L(x)$ якомога нижчого степеня так, щоб

$$L(x_i) = y_i \quad (i = 0-n) \quad (1.15)$$

Поліном

$$l_k(x) = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}, \quad (1.16)$$

такий, що $l_k(x_i) = \begin{cases} 0 & \text{при } i \neq k, \\ 1 & \text{при } i = k. \end{cases}$

Тому поліном $L(x) = \sum_{i=0}^n y_k l_k(x)$ задовольняє вимоги (1.15).

Степінь цього полінома не вище $n-1$. З другого боку, ніякого іншого полінома $M(x)$ степеня не вище $n-1$, який би задовольняв вимоги (1.15), існувати не може, бо інакше різниця $L(x)-M(x)$ була б не тотожним нулю поліномом степеня не вище $n-1$, який має n коренів (1.13), що неможливо. Отже, поліном $L(x)$ і є єдиним розв'язком поставленої задачі.

Поліном Лагранжа:

$$L(x) = \sum_{j=0}^n y_j \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i}. \quad (1.17)$$

У конструктивній теорії функцій доведена така теорема:

Якщо $f(x)$ є цілою функцією, заданою на $[a; b]$, то при будь-якому законі введення вузлів, аби їх число необмежено зростало і вони не виходили з $[a; b]$, рівномірно на $[a; b]$

$$\lim_{n \rightarrow \infty} L(x) = f(x) \quad (1.18)$$

Існують також інші форми запису полінома Лагранжа (інтерполяційний поліном Ньютона, інтерполяційна схема Ейткена), які більш зручні в застосування у тому чи іншому випадку [40–42].

2.4 Диференціювання експериментальних даних

Нерідко експериментатора цікавить не тільки апроксимуюча функція, а й її похідна по аргументу. Зокрема, в алгоритмі передбачена ця операція з метою виявлення екстремумів функції (число максимумів на першому етапі буде відповідати числу індивідуальних смуг в спектрі).

Оскільки для наближення експериментальних даних був використаний поліном Лагранжа, то похідну експериментальної функції можна вважати наближено рівною похідній в тій самій точці функції, яка її апроксимує.

У цьому випадку, використовуючи вузлові точки $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, маємо:

$$\begin{aligned} \frac{dL}{dx} &= y_0 \frac{\sum_{j=1}^n \left\{ \frac{1}{x-x_j} \prod_{i=1}^n (x-x_i) \right\}}{\prod_{i=1}^n (x_0-x_i)} + y_1 \frac{\sum_{\substack{j=0 \\ j \neq 1}}^n \left\{ \frac{1}{x-x_j} \prod_{\substack{i=0 \\ i \neq 1}}^n (x-x_i) \right\}}{\prod_{\substack{i=0 \\ i \neq 1}}^n (x_1-x_i)} + \dots + y_n \frac{\sum_{j=0}^{n-1} \left\{ \frac{1}{x-x_j} \prod_{i=0}^{n-1} (x-x_i) \right\}}{\prod_{i=0}^{n-1} (x_n-x_i)} = \\ &= \sum_{k=0}^n y_k \frac{\sum_{\substack{j=0 \\ j \neq k}}^n \left\{ \frac{1}{x-x_j} \prod_{\substack{i=0 \\ i \neq k}}^n (x-x_i) \right\}}{\prod_{\substack{i=0 \\ i \neq k}}^n (x_k-x_i)} \end{aligned} \quad (1.19)$$

Отже, похідну dL/dx використовують як наближене значення похідної експериментальних значень dy/dx .

2.5 Декомпозиція спектрів

Основна задача обробки спектра – це декомпозиція, або розкладання його, а точніше його інформаційної компоненти, на складові субкомпоненти і обчислення характеристик останніх.

Існуюча практика розв'язання задачі декомпозиції дозволяє сформулювати задачу таким чином.

Зіставляють реальному спектру – сумі компонент – його модель, яка є сумою моделей компонент, і підганяють її до реального спектра в надії, що для того, щоб сумарна модель була близька до сумарного спектра в розумінні тієї чи іншої метрики, моделі компонент будуть змушені розташовуватися в сумарній моделі так само, як реальні компоненти в сумарному реальному

спектрі. Однак ситуація тут настільки складна, що завдання декомпозиції в загальному випадку настільки просто не вирішується [27].

Розглянемо задачу для випадку малої перешкоди, накладеної на вимірюваний спектр. Вважаємо форму індивідуальних смуг відомою. У роботі використаємо характеристичну функцію гауссового типу (1.2).

Часто спектри містять фон, який, як правило, змінюється набагато повільніше порівняно з піками спектрів. Для наближення фону використовують прості функції, і найбільш часто, поліноми невисокого степеня. Наявність фону найбільше впливає на визначення початкового значення висоти піків y_i і менше – на всі інші параметри.

Оскільки при обробці спектрів був отриманий незначний вклад полінома, апроксимуючого фон (коефіцієнти порядку $10^{-6} - 10^{-4}$), неінформаційними сигналами $\sum_{i=1}^m c_i x^i$ в подальших викладках нехтуємо.

Наближення експериментальних спектрів $y_i = y(x_i)$ розглядаємо функцією

$$u(x) = \sum_{j=0}^m y_{0j} \exp\left(-\frac{\ln 2(x - x_{0j})^2}{h^2}\right), \quad (1.20)$$

де x_{0j} – положення j -го піка,

h_j – півширина j -го піка на половині його висоти,

y_{0j} – висота піка.

Початкові значення параметрів функції можна визначити, наприклад, з умови її проходження через три точки вершини піка (x_{i-1}, y_{i-1}) , (x_i, y_i) , (x_{i+1}, y_{i+1}) ординати яких задовольняють нерівності $y_{i-1} \leq y_i$, $y_i > y_{i+1}$.

Вважаємо, що спектр рівновіддалений за аргументами, тобто $x_i - x_{i-1} = x_{i+1} - x_i = d = const$.

Тоді із системи рівнянь

$$\begin{cases} y_{i-1} = y_0 \exp\left(-\frac{\ln 2(x_{i-1} - x_0)^2}{h^2}\right) \\ y_i = y_0 \exp\left(-\frac{\ln 2(x_i - x_0)^2}{h^2}\right) \\ y_{i+1} = y_0 \exp\left(-\frac{\ln 2(x_{i+1} - x_0)^2}{h^2}\right) \end{cases} \quad (1.21)$$

знайдемо

$$x_0 = x_i + \frac{d}{2} \frac{\ln\left(\frac{y_{i-1}}{y_{i+1}}\right)}{\ln\left(\frac{y_{i-1}y_{i+1}}{y_i^2}\right)}, \quad (1.22)$$

$$h = \left[\frac{-2d(x_i - x_0 + d/2) \ln 2}{\ln(y_{i+1}/y_i)} \right]^{\frac{1}{2}}, \quad (1.23)$$

$$y_0 = y_i \exp\left(\frac{\ln 2(x_i - x_0)^2}{h^2}\right). \quad (1.24)$$

У формулах x_i – положення максимуму складної експериментальної функції, y_i – значення функції в точці x_i , тобто значення полінома Лагранжа в цій точці. З метою точнішого визначення положення максимумів, в алгоритмі передбачений метод дихотомії. Метод дихотомії, або половинного ділення, полягає в наступному. Нехай в точці a похідна має додатний знак, а в точці b – від’ємний. Отже, в деякій точці відрізка $[a, b]$ похідна стає нулем, – це і є шуканий максимум. Задача – знайти його, або, принаймні, найбільш близьке до нього значення.

Визначаємо середину $\bar{x} = \frac{a+b}{2}$ відрізка $[a, b]$ і обчислюємо похідну в точці \bar{x} . Далі вибираємо, яку з двох частин відрізка узяти для подальшого уточнення максимуму. Оскільки $f(x)$ – неперервна функція аргументу x , то

максимум буде знаходитись в тій половині відрізка, на кінцях якого $f(x)$ має різні знаки. Ітераційний процес продовжуватимемо до тих пір, поки проміжок $[a, b]$ не стане меншим заданої похибки.

Обчисливши максимуми, для кожного обчислюємо початкові параметри смуг – x_{0i} та h_i . Інтенсивність y_{0i} знаходимо за допомогою МНК.

2.6 Оцінка параметрів апроксимуючих функцій

Попередня обробка складного контура була спрямована на те, щоб отримати початкові значення x_{0i} і h_i для знаходження y_{0i} і подальшого уточнення усіх цих параметрів. Сформулюємо тепер задачу так: провести апроксимуючу криву, яка не обов'язково проходить через експериментальні точки, але водночас відображає досліджувану залежність, згладжує можливі викиди за рахунок похибки експерименту.

Перепозначимо експериментальні вузли як $x_j, j = \overline{1, n}$, а відповідні їм значення вимірної оптичної густини як $f(x_j) = y_j, j = \overline{1, n}$. Введемо неперервну функцію $\varphi(x)$ для апроксимації дискретної залежності $f(x_j)$. У вузлах функції $\varphi(x)$ та $f(x_j)$ будуть відрізнятися на величину $\varepsilon_j = \varphi(x_j) - f(x_j)$. Взявши суму квадратів таких відхилень, отримаємо наступну величину:

$$Q = \sum_{j=0}^n \varepsilon_j^2 = \sum_{j=0}^n [\varphi(x_j) - f(x_j)]^2. \quad (1.25)$$

МНК – метод побудови апроксимуючої функції $\varphi(x)$, який полягає в мінімізації величини Q . Це один з найбільш поширених методів обробки експериментальних даних, особливо ефективний при застосуванні його для апроксимації експериментальних даних, які лінійно залежать від параметрів. У цьому випадку дисперсії оцінок параметрів мають мінімальні значення.

Обираємо функцію $\varphi(x)$ у вигляді лінійної комбінації:

$$\varphi(x) = C_0\varphi_0(x) + C_1\varphi_1(x) + \dots + C_m\varphi_m(x), \quad (1.26)$$

де $\varphi_0(x), \varphi_1(x), \dots, \varphi_m(x)$ – базисні функції; $m \leq n$; C_0, C_1, \dots, C_m – коефіцієнти, які визначаються при мінімізації величини Q .

Саме тому при фіксованих значеннях x_{0i}, h_i , базисна гауссова функція буде залежати тільки від x , коефіцієнтами будуть y_{0i} :

$$U(x) = \sum_{i=0}^m y_{0i} \underbrace{\exp\left(-\frac{\ln 2(x-x_{0i})^2}{h_i^2}\right)}_{\phi_i(x)}, \text{ тобто МНК у цьому випадку буде лінійним.}$$

Математично умови мінімуму суми квадратів відхилень Q запишемо, прирівнюючи до нуля частинні похідні від Q по коефіцієнтам $C_i, 0 \leq i \leq m$:

$$\begin{cases} \frac{dQ}{dC_0} = 2 \sum_{j=0}^n [C_0\phi_0(x_j) + C_1\phi_1(x_j) + \dots + C_m\phi_m(x_j) - f(x_j)]\phi_0(x_j) = 0, \\ \frac{dQ}{dC_1} = 2 \sum_{j=0}^n [C_0\phi_0(x_j) + C_1\phi_1(x_j) + \dots + C_m\phi_m(x_j) - f(x_j)]\phi_1(x_j) = 0, \\ \dots \\ \frac{dQ}{dC_m} = 2 \sum_{j=0}^n [C_0\phi_0(x_j) + C_1\phi_1(x_j) + \dots + C_m\phi_m(x_j) - f(x_j)]\phi_m(x_j) = 0. \end{cases}, \quad (1.27)$$

Усі коефіцієнти C_i визначають із системи лінійних алгебраїчних рівнянь. Ця система називається системою нормальних рівнянь. Матриця цієї системи має такий вигляд:

$$\begin{vmatrix} (\phi_0, \phi_0) & (\phi_0, \phi_1) & \dots & (\phi_0, \phi_m) \\ (\phi_0, \phi_1) & (\phi_1, \phi_1) & \dots & (\phi_1, \phi_m) \\ \dots & \dots & \dots & \dots \\ (\phi_0, \phi_m) & (\phi_1, \phi_m) & \dots & (\phi_m, \phi_m) \end{vmatrix}, \quad (1.28)$$

і називається матрицею Грама [43].

Елементи матриці Грама є скалярними добутками базисних функцій

$$(\phi_l, \phi_k) = \sum_{j=0}^n \phi_l(x_j) \phi_k(x_j) \quad (1.29)$$

Розширена матриця системи рівнянь вийде додаванням справа до матриці Грама вільних членів

$$\begin{pmatrix} (\phi_0, f) \\ (\phi_1, f) \\ \dots \\ (\phi_m, f) \end{pmatrix}, \quad (1.30)$$

де скалярні добутки, які є елементами стовпця, визначаються аналогічно:

$$(\phi_l, f) = \sum_{j=0}^n \phi_l(x_j) f(x_j), \quad (1.31)$$

Під час програмної реалізації алгоритму МНК корисні основні властивості матриці Грама:

- 1) матриця є симетричною, бо $a_{lk} = a_{kl}$, що дозволяє скоротити обсяг обчислень при заповненні матриці;
- 2) матриця є додатно визначеною, отже, при розв'язанні системи нормальних рівнянь методом виключення Гаусса можна відмовитись від процедури вибору головного елемента;
- 3) визначник матриці буде відмінним від нуля, якщо як базис вибрані лінійно незалежні функції $\phi_i(x)$, при цьому система рівнянь має єдиний розв'язок.

Алгоритм передбачає знаходження y_{0i} розв'язанням отриманої системи методом Гаусса.

I етап – прямий хід методу, полягає в послідовному виключенні невідомих з рівнянь, починаючи з першого (приведення матриці до трикутного вигляду).

II етап – обернений хід методу Гаусса, полягає в послідовному визначенні невідомих, починаючи з останнього і завершуючи першим.

Отриманими розв'язками і подальшим винесенням необхідних значень ($x_0[i]$, $y_0[i]$, $h[i]$, X_UZL , Y_UZL , kol_extr) завершується перша частина програми – Program Span (рис. 2.1 а).

2.7 Уточнення параметрів індивідуальних смуг

Отримане розкладання експериментального дискретного спектра в результаті обробки його першою частиною програми дає досить наближений розв'язок поставленої задачі. Уточнення цього розв'язку виконує друга частина експертної системи – Program Exacting. Основне її призначення – подальше застосування МНК. Змінюючи в певних межах значення положення, інтенсивності і півширини, кожен раз обчислюється SqSum (сума квадратів відхилень за всіма вузлами), порівнюється з заданою похибкою epsilon. За умови $SqSum < epsilon$, програма завершує свою роботу (рис. 2.1 б).

2.8 Умови однозначності декомпозиції складного спектрального контура

Для оцінки використовуваних методів дуже важливим є питання про однозначність одержуваних результатів [44, 27, 32]. Загальний вигляд аналізованої кривої і початкові дані істотно впливають на одержувані результати. Автори зазначають, що застосування будь-якого методу розкладання перекритих спектральних смуг дає в багатьох випадках неоднозначні результати.

Враховуючи похибки експерименту, певні наближення математичної обробки тощо можна сказати, що створена в роботі експертна система також потребує удосконалення.

Неоднозначність розшифрування складного контура може полягати не тільки у можливості різних варіантів розкладання з одним і тим самим числом смуг, а й у можливості представлення деяких контурів різним їх числом 2.3.

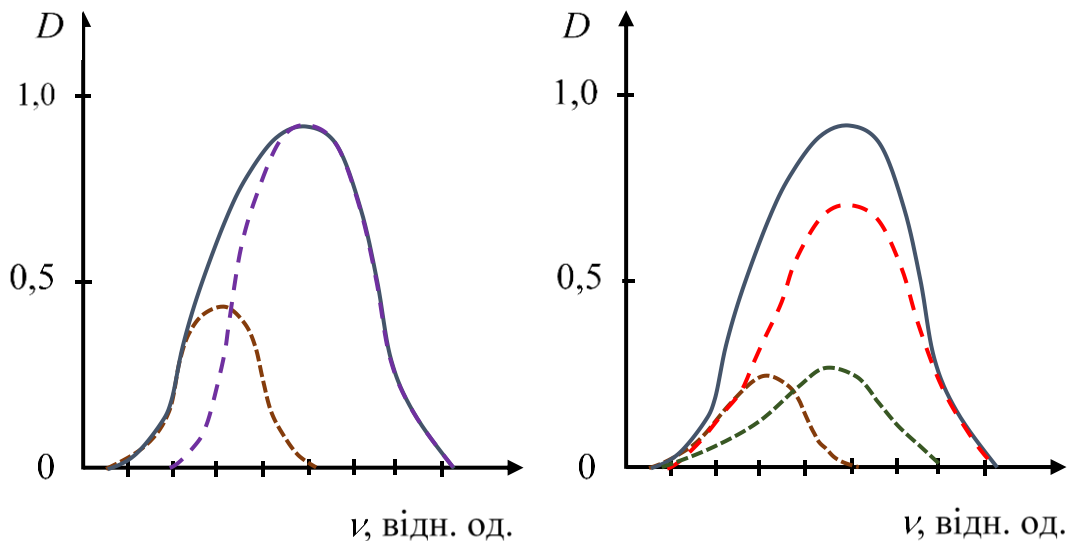


Рис. 2.3. Розкладання на дві (а) та три (б) симетричні смуги [44]

Для з'ясування умов однозначності розкладання складного контура на елементарні смуги використовувались найбільш загальні відомості про спектральні характеристики цих смуг. Залучення додаткової інформації, якщо це можливо, істотно впливає на характер розв'язання, а у ряді випадків може навіть забезпечити його єдиність. Певна річ, найбажанішою інформацією є відомості щодо числа індивідуальних складових спектрів.

Експериментальне визначення сумарного спектрального контура, як правило, здійснюється з деяким рівнем шуму, що істотно ускладнює процедуру розкладання його на елементарні смуги. У такій ситуації навіть виконання принципів умов однозначності не знімає неоднозначності, пов'язаної з нестійкістю розв'язків. Щоправда, у цьому випадку можна вибрати або розробити методи і алгоритми, які дозволяють подолати або хоча б зменшити нові труднощі отримання правильного і стійкого до похибок вимірювань розв'язку.

Зазначені зауваження враховані при створенні експертної системи декомпозиції складного контура, апробація програми і аналіз отриманих результатів свідчить про задовільну якість роботи системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Для програмної реалізації використовувалось середовище Borland Delphi 7, яке є високоефективним інструментом для реалізації прикладних програм.

Програмна реалізація містить в собі:

- а) вирішення задач наочності і зручності для користувача;
- б) повноцінне функціонування побудованої математичної моделі.

Результатом проведеної роботи є програмний продукт “SpecDec”, опис якого подано у наступних пунктах.

3.1 Інтерфейс та основні можливості програмного забезпечення

Загальний вигляд вікна системи “SpecDec” представлений на рис. 3.1.

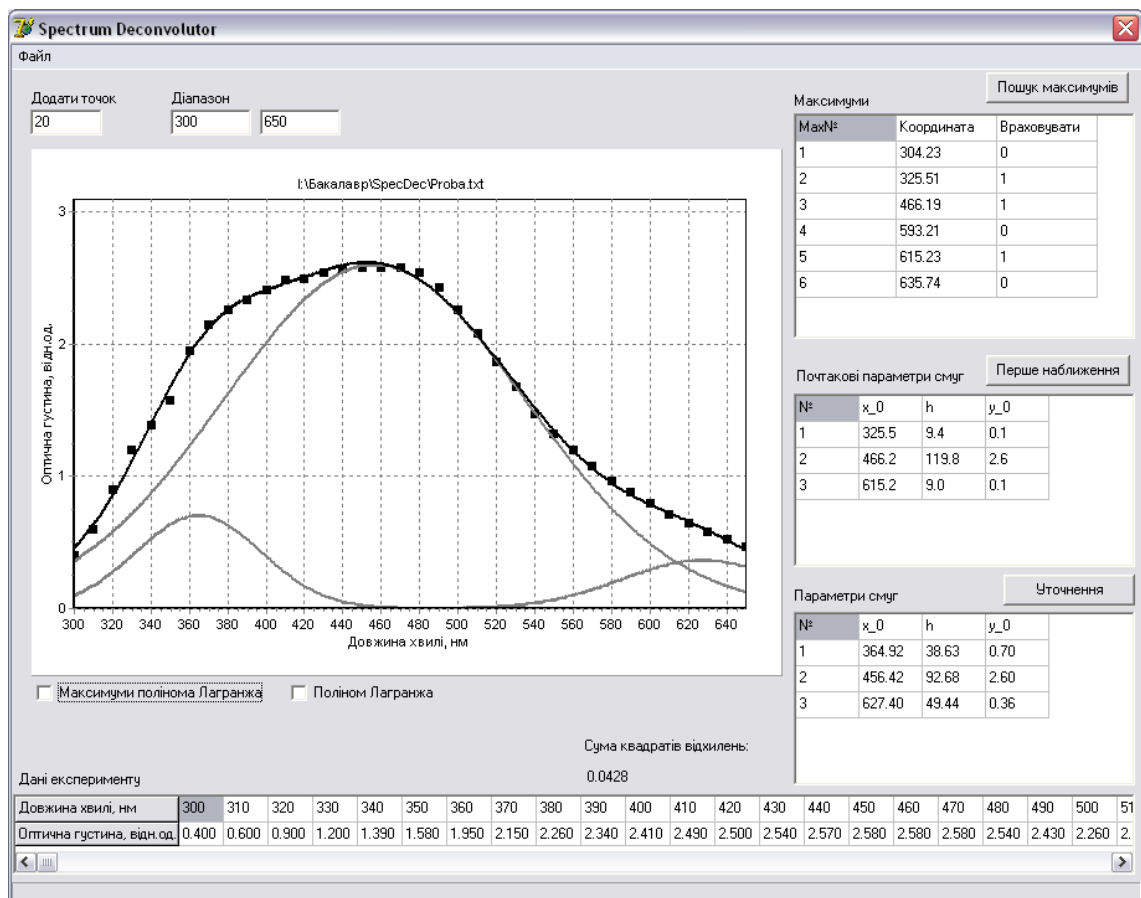


Рис. 3.1. Загальний вигляд вікна програми

Основні елементи головної форми програми – графік і таблиці, де відображаються результати.

Меню *Файл* дозволяє відкрити існуючу базу даних – текстовий файл, в якому містяться дані одного сеансу вимірювання, створити і зберегти нову базу даних, виконати редагування даних, роздрукувати графік, на якому зображено результати декомпозиції експериментального спектра, експортувати графік у формат .bmp, зберегти параметри характеристичних компонент розкладеного спектра у текстовий файл.

Для початку роботи з експертною системою необхідно відкрити існуючу базу даних або створити нову – меню *Файл – Відкрити базу* або *Створити базу*. У полях *Діапазон* ввести значення довжин хвиль, що утворюють діапазон спектра, який буде оброблений. Виконати пошук максимумів натисканням відповідної кнопки *Пошук максимумів*. Відомим фактом є нестійкість поліному Лагранжа між вузлами інтерполяції на кінцях діапазону значень, причому степінь нестійкості зростає з ростом числа вузлів інтерполяції. Для запобігання утворенню в «крилах» неіснуючих смуг, передбачено згладжування полінома Лагранжа: на кінцях експериментальних даних рекомендується додати до вихідних даних справа і зліва деяку кількість точок (10–20), яку можна задати у полі *Додати точок* на головній формі. Значенням оптичної густини у цих точках присвоюються значення оптичної густини крайніх точок, але ці додані точки не використовуються при побудові графіків. Таким чином в роботі була вирішена проблема надмірної кількості знайдених неінформативних максимумів полінома Лагранжа.

Після цього натисканням кнопки *Пошук масимумів* будуть знову розраховані максимуми полінома Лагранжа. Поліном можна візуалізувати активуючи поле *Поліном Лагранжа* під графіком. Після знаходження максимумів у таблиці *Максимуми* потрібно помітити цифрою «1» у стовпчику *Враховувати* ті з них, які на думку експериментатора є адекватними для подальшого уточнення. Інші максимуми, які не треба враховувати,

відмічаються будь-яким символом відмінним від «1». При натисканні кнопки *Перше наближення*, за формулами (1.22) та (1.23) обчислюються початкові значення параметрів смуг x_{0i} , h_i потім застосовується лінійний МНК, мінімізується величина суми квадратів відхилень (1.25). Далі користувач має натиснути кнопку *Уточнення* і за алгоритмом п. 2.7 відбувається пошук оптимальних параметрів смуг з відображенням результатів на графіку. Якщо швидкість зменшення суми квадратів відхилень низька і користувача вже задовольняють отримані результати, то можна натиснути клавішу *ESC* і процес підрахунку зупиниться, поточні результати відображаються на графіку у вигляді окремих спектральних смуг і їх суми, а також числові значення параметрів характеристичних смуг – у відповідній таблиці *Параметри смуг*.

Системою передбачено можливість експортувати результати декомпозиції з графіка у формат *.bmp* – меню *Файл – Експорт графіка* – відкриється діалог збереження файлу, де можна задати його розміщення та ім'я. Для друку результатів декомпозиції потрібно вибрати меню *Файл – Роздрукувати*, відкриється стандартний діалог друку, де можна зробити попередній перегляд, налаштувати опції принтера тощо.

Звернемось до опису вікна створення/редагування даних, яке представлено на рис. 3.2.

Для створення бази необхідно заповнити відповідні поля початкового та кінцевого значень довжин хвиль досліджуваного діапазону, кроку зміни λ і перейти у таблицю, де залишиться тільки заповнити рядок *Оптична густина*, а рядок *Довжина хвилі* буде автоматично заповнено.

При створенні або редагуванні бази можливе копіювання даних з буфера обміну Windows, якщо дані розміщені там по рядках у форматі координата довжини хвилі, знак табуляції, оптична густина. Ця можливість дозволяє скопіювати дані з таблиць MSExcel, MSWord та текстового редактора.

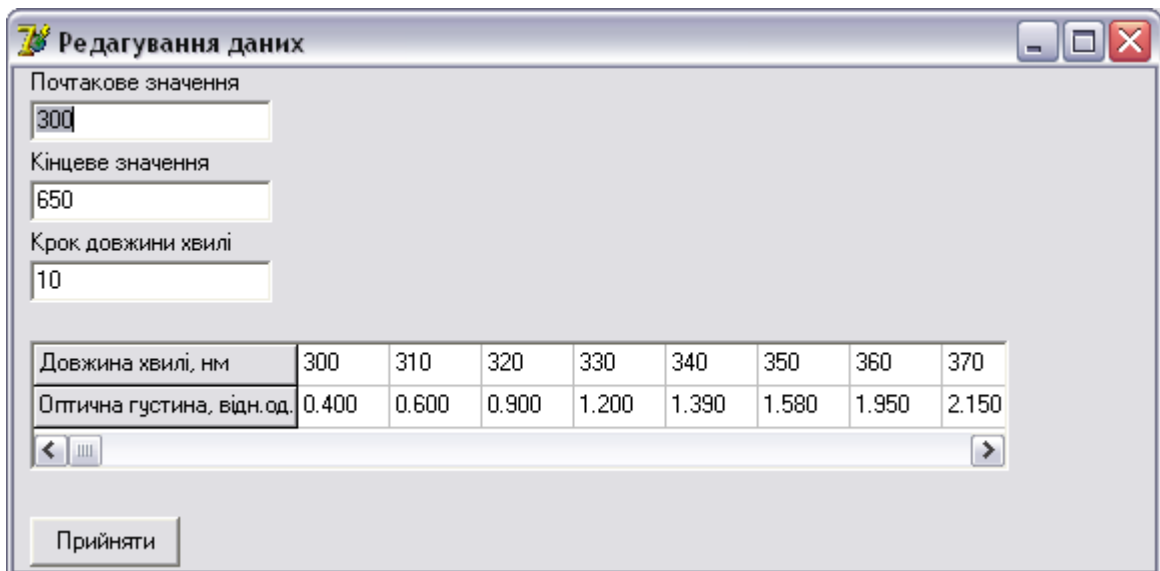


Рис. 3.2. Вікно створення/редагування бази даних експерименту

3.2 Деякі аспекти програмування

Для реалізації розробленого алгоритму використані такі відомі прийоми програмування – робота з масивами, циклами, зчитування даних з текстових файлів, стандартні діалоги відкриття і збереження файлів, обробка подій, зв’язаних з об’єктами форми тощо. Звернемось до деяких моментів реалізації.

Лістинг блоку, що реалізує обчислення значення полінома Лагранжа у точці t , побудованого за даними масиву експериментальних даних mas :

```
function lagr(t:real; mas:data_mas):real;
  function pr(t:real; l:integer):real;
    var tmp, tmp1:real;
    i:integer;
  begin
    tmp:=1;
    tmp1:=1;
    for i:=1 to l-1 do
      tmp:=tmp*(t-mas[i,0]);
    for i:=l+1 to N do tmp:=tmp*(t-mas[i,0]);
    for i:=1 to l-1 do
      tmp1:=tmp1*(mas[l,0]-mas[i,0]);
    for i:=l+1 to N do
      tmp1:=tmp1*(mas[l,0]-mas[i,0]);
    pr:=tmp/tmp1;
  end;
var i:byte;
    tmp:real;
begin
```

```

tmp:=0;
for i:=1 to N do
  begin
    tmp:=tmp+pr(t,i)*mas[i,1];
  end;
end;

```

Метод дихотомії використано для пошуку максимуму полінома

Лагранжа на відрізку $[a,b]$ з точністю ϵ за елементами масиву `matr`:

```

function dihotomia(a,b,eps:real;matr:data_mas):real;
var x1,x2,l,xm:real;
begin
xm:=a+(b-a)/2;
l:=b-a;
repeat
  x1:=a+l/4;
  x2:=b-l/4;
  if lagr(x1, matr)>lagr(xm, matr) then
    begin
      b:=xm;
      xm:=x1;
    end
  else
    if lagr(x2, matr)>lagr(xm, matr)
    then
      begin
        a:=xm;
        xm:=x2;
      end
    else
      begin
        a:=x1;
        b:=x2;
      end;
  l:=b-a;
until abs(l)<eps;
result:=(a+b)/2;
end;

```

Приклад використання рекурсивних процедур – прямий хід методу Гаусса для розв’язання системи лінійних алгебраїчних рівнянь (змінна m – матриця системи):

```

procedure gauss(var m:matr_mas; var r: integer);
var i,j:integer;
    tm:extended;
begin
  if r<n_p then
    begin
      for i:=r+1 to n_p do
        begin

```

```

if m[r,r]<>0 then
begin
tm:=m[r,i]/m[r,r];
for j:=1 to n_p+1 do
if (abs(m[j,i]-m[j,r]*tm) < 1e-17)
then
m[j,i]:=0
else m[j,i]:=m[j,i]-m[j,r]*tm;
end
end;
r:=r+1;
Gauss(m,r);
end;
end;

```

Для реалізації можливості зупинки алгоритму пошуку оптимальних параметрів індивідуальних смуг (п. 2.7) використано метод `Application.ProcessMessages` в тілі циклу, що дозволяє виконати обробку натискання клавіші `ESC`, при натисканні цієї клавіші властивість форми `MainForm.Tag` типу `integer` приймає значення 1, далі в циклі йде перевірка значення цієї властивості, і якщо `MainForm.Tag=1` то цикл завершується.

Обробка ситуації натискання клавіші *ESC*:

```

procedure TMainForm.Button2KeyPress(Sender:
TObject; var Key: Char);
begin
if key=#27 then
mainform.Tag:=1;
end;

```

3.3 Тестування експертної системи

Тестування створеної експертної системи за даними реальних експериментальних спектрів показало її адекватність, але на разі залишається певна низка питань, які потрібно буде розв'язати у подальшому. На рис. 3.3 і рис. 3.4 відображено результати моделювання двох різних апаратурних спектрів. При розкладанні спектра поглинання 0,6% водного розчину сулеми HgCl_2 отримані спектральні смуги з максимумами на довжинах хвиль $\lambda_1 = 209$ нм і $\lambda_2 = 237$ нм, що узгоджується з іншими квантово-механічними розрахунками та відомостями про дану сполуку. На рисунках сірим кольором

зображені характеристичні лінії, чорним – їх сума (теоретична крива), чорними квадратами позначені експериментальні дані.

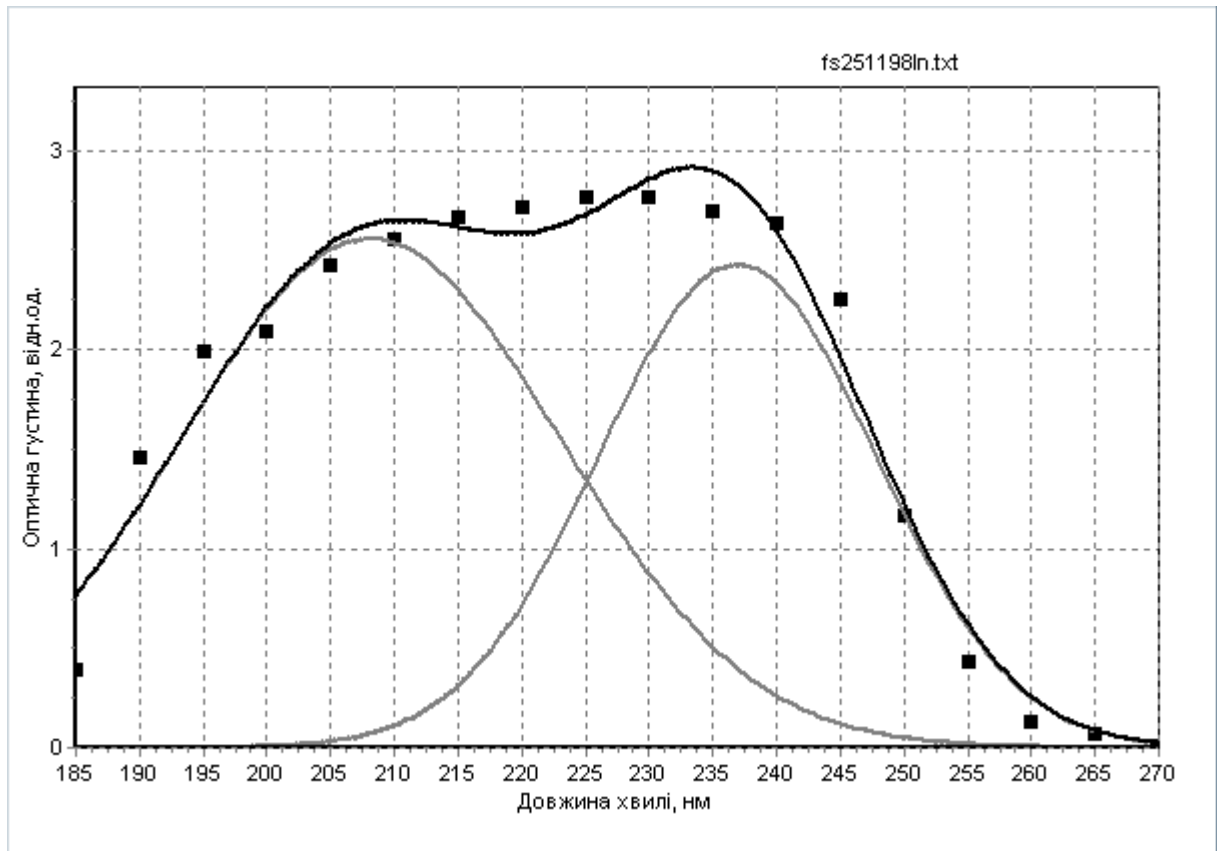


Рис. 3.3. Декомпозиція спектра 0,6% розчину HgCl₂

При розкладанні спектра поглинання нейтрального оптичного скла НС-2 отримані характеристичні смуги з положеннями максимумів окремих компонент на довжинах хвиль $\lambda_1 = 293$ нм, $\lambda_2 = 335$ нм, $\lambda_3 = 275$ нм, $\lambda_4 = 346$ нм та $\lambda_5 = 540$ нм відповідно.

Враховуючи похибки експерименту, неточності математичної обробки не слід ідеалізувати даний обчислювальний метод, і його результати також не можуть претендувати на однозначність. У будь-якому разі дослідник має завжди враховувати додаткову інформацію стосовно досліджуваних об'єктів чи явищ, а отримані поданими у роботі методами результати можуть слугувати лише допоміжним інструментом для тих чи інших висновків, адже

формальність математичних підходів, запропонованих у роботі, може суперечити фізичним відомостям.

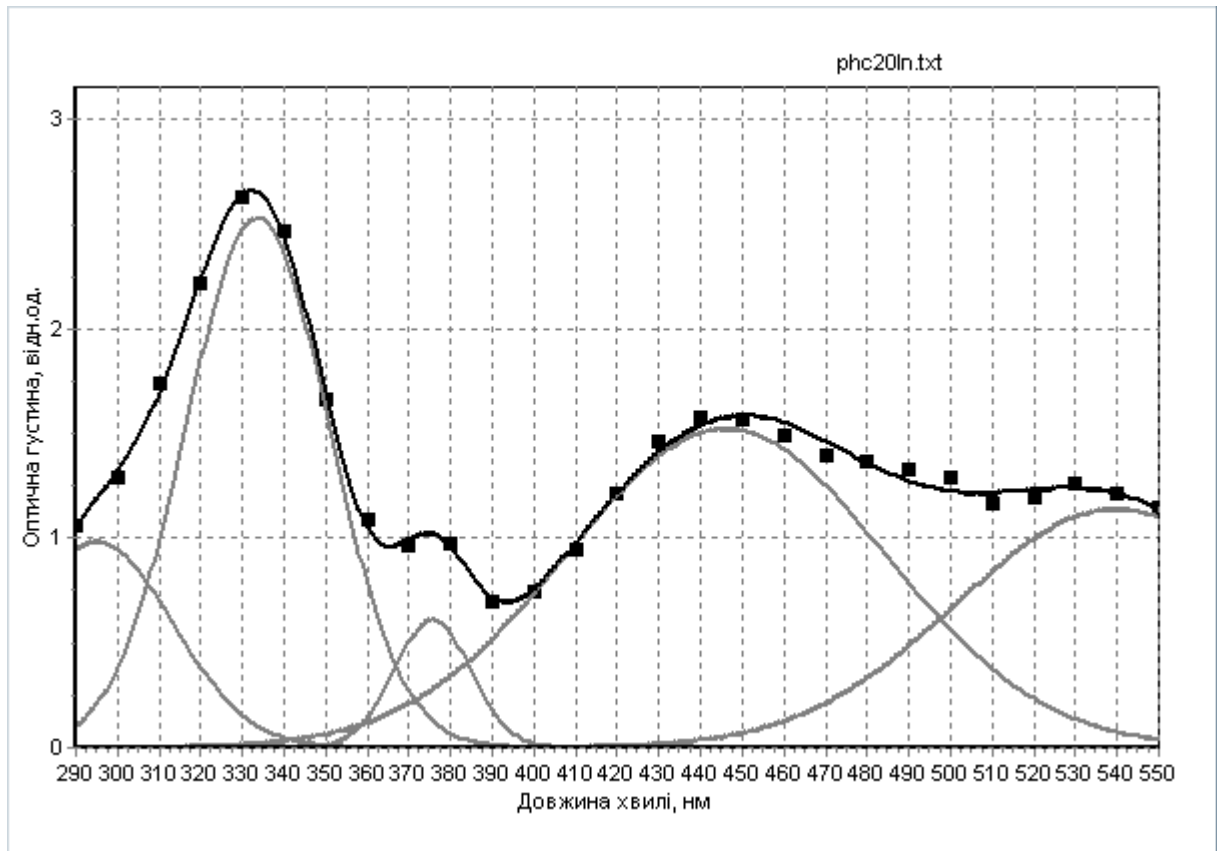


Рис. 3.4. Результат декомпозиції спектра поглинання світлофільтра НС-2

Неоднозначність розкладання складного контура може полягати не тільки у можливості різних варіантів розташування однієї і тієї самої кількості смуг з різними значеннями параметрів x_{0i} , h_i , y_{0i} , а й у можливості представлення експериментального спектра різним числом смуг, що продемонстровано на рис. 3.5: $\lambda_1 = 197$ нм, $\lambda_2 = 221$ нм, $\lambda_3 = 242$ нм.

Аналізуючи рисунки 3.3 і 3.5 можна спостерігати неоднозначність отриманих результатів, хоча суми квадратів відхилень відрізняються не істотно. Але залучення додаткової інформації про досліджуваний зразок вказує на хибність результату на рис. 3.5, бо відомо, що спектр даного зразка у цій області поглинання має лише дві спектральні смуги.

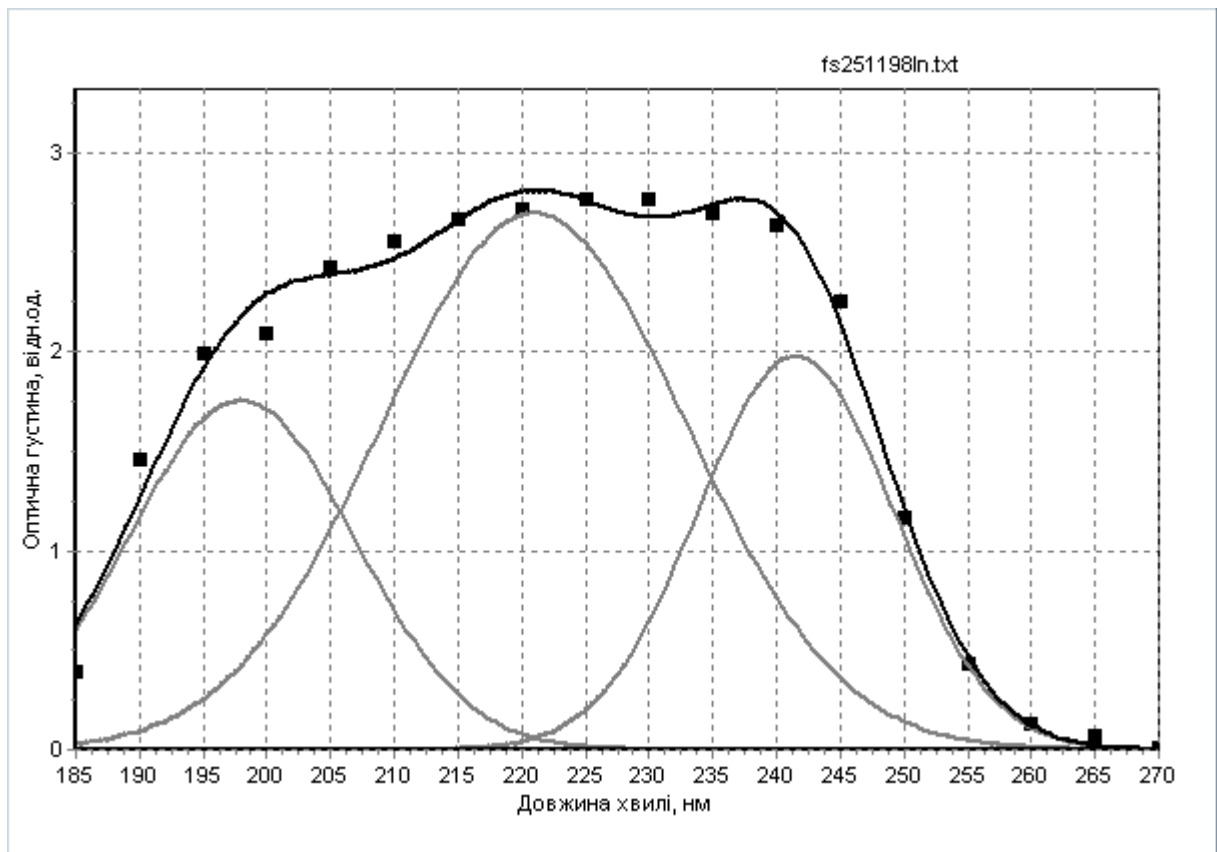


Рис. 3.5. Представлення спектра сулеми без залучення додаткової інформації

Отже, досягнуто певних результатів, але дуже важливим є пошук інших альтернативних шляхів розв'язання описаної проблеми, серед яких можна відмітити використання відмінних від поліноміальної інтерполяції методів початкової апроксимації даних, наприклад, апроксимація сплайнами, раціональними функціями, і найголовніше, – навчання системи більш досконалому алгоритму уточнення параметрів спектральних смуг.

Експертна системи апробована при дослідженні впливу рентгенівського випромінювання на інтенсивність смуг оптичного поглинання рентгеночутливих органічних шарів. Аналіз спектрів показав, що під дією опромінення відбуваються зміни в ультрафіолетовому діапазоні, дозволив виявити діапазон поглинання утворюючих сполук, що можна використовувати для порівняння їх поведінки в різних фізичних процесах.

ВИСНОВКИ

У кваліфікаційній магістерській роботі проаналізовані основні методи та алгоритми декомпозиції складних контурів. На основі чисельних методів розроблена методика і алгоритм аналізу оптичних спектрів. У Delphi-середовищі створена експертна система для обробки експериментальних дискретних спектрів, яка реалізує розроблений алгоритм та має зручні можливості для наочного спостереження результатів декомпозиції, збереження їх для подальших досліджень.

Метод декомпозиції побудований на припущенні, що отриманий апаратний спектр є дискретним і являє собою суперпозицію окремих характеристичних смуг. Обрана така послідовність етапів для вирішення поставленої задачі: інтерполяція залежностей; диференціювання експериментальних даних, декомпозиція спектрів; уточнення параметрів індивідуальних смуг.

Система протестована на прикладі обробки спектра 0,6 % розчину сулеми та нейтрального оптичного скла NC_2 . Апробація експертної системи проведена при обробці спектрів органічних шарів, виявлений вплив рентгенівського випромінювання на інтенсивність смуг оптичного поглинання рентгеночутливих шарів в ультрафіолетовій ділянці спектра. Апробація показала дієздатність системи та можливість застосування на практиці.

Одержані в роботі результати можуть застосовуватись для подальшого пошуку розв'язання даної нетривіальної задачі. Залучення інших математичних методів, квантово-хімічних розрахунків дозволить вдосконалити модель і отримати кращу її відповідність фізичному об'єкту. Прикладний характер результатів кваліфікаційної роботи полягає також у їх використанні у навчальному процесі СумДУ при викладанні дисциплін «Чисельні методи» та «Математичні методи дослідження операцій» для студентів спеціальностей «комп'ютерні науки» та «кібербезпека».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сизиков В.С. Устойчивые методы обработки результатов измерений: навч. посіб. СПб. : СпецЛит, 1999. 240 с.
2. Тихонов А. Н, Арсенин В. Я. Методы решения некорректных задач. М. : Наука, 1986. 288 с.
3. Komada P., Litvinenko V. I., Olszewski S. V. Preprocessing optical spectra superposition for further decomposition // SPIE-Intl Soc Optical Eng, 16 March 19. C. 28.
4. Kalantzis V., Kestyn J., Polizzi E. et al. Domain decomposition approaches for accelerating contour integration eigenvalue solvers for symmetric eigenvalue problems // Numerical Linear Algebra with Applications. 2018. Vol. 25, No. 5.
5. Vacavant A., Kerautret B., Roussillon T. et al. Reconstructions of noisy digital contours with maximal primitives based on multi-scale/irregular geometric representation and generalized linear programming // 20th IAPR International Conference on Discrete Geometry for Computer Imagery, Sep 2017. Vienna, Austria. pp. 291–303.
6. Gao M., Chen H., Zheng S. et al. A factorization based active contour model for texture segmentation // Proceedings - International Conference on Image Processing (ICIP), IEEE Computer Society, 03 August 16.
7. Xi Y., Saad Y. Computing partial spectra with least-squares rational filters // SIAM Journal on Scientific Computing. 2016. Vol. 38, No. 5. pp. A3020–A3045.
8. Shaimukhametova E. R., Galimullin D. Z., Sibgatullin M. E. et al. Influence of the character of noise on the decomposition of complex spectral contours // Bulletin of the Russian Academy of Sciences: Physics, March. 12.
9. Shaimukhametova E. R., Galimullin D. Z., Sibgatullin M. E. et al. Application of a genetic algorithm and wavelet analysis to the interpretation of the infrared Fourier

spectra of branched polymethylmethacrylate // *Bulletin of the Russian Academy of Sciences: Physics*. 2010. Vol. 74, No. 7. pp. 959–962.

10. Karasalo M., Piccolo G., Kragic D. et al. Contour reconstruction using recursive smoothing splines – Algorithms and experimental validation // *Robotics and Autonomous Systems*. 2009. Vol. 57, No. 6–7. pp. 617–628.

11. Pallikari F. A study of the fractal character in electronic noise processes // *Chaos, Solitons & Fractals*. 2001. Vol. 12, No. 8. pp. 1499–1507.

12. Kharintsev S. S., Nigmatullin R. R., Salakhov M. K. Solving inverse problems in applied spectroscopy with random fractal noise // *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2000. Vol. 67, No. 3. pp. 239–252.

13. Coleman T. F., Li Y. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds // *SIAM Journal on Optimization*. 1996. Vol. 6, No. 2. pp. 418–445.

14. Васильев В. Н., Гуров И. П. Компьютерная обработка сигналов в приложении к интерферометрическим системам. СПб.: БХВ, 1998. 240 с.

15. Дьяконов В. П. Вейвлеты. От теории к практике. М.: Солон-Р, 2010. 400 с.

16. Реклейтис Г., Рейвиндран А., Регсдел К. Оптимизация в технике: В 2-х кн. Кн. 1. [пер. з англ.]: М.: Мир, 1986. 349 с.

17. Банди Б. Методы оптимизации. Вводный курс [пер. з англ.]: 1988. 128 с.

18. Бейко И. В., Бублик Б. Н., Зинько П. Н. Методы и алгоритмы решения задач оптимизации. К.: Вища школа. Головное изд-во, 1983. 512 с.

19. Тайбин Б. З. Нелинейные методы обработки данных: дис. д.ф.-м.н.: 05.13.18. СПб: СПбГУ. 350 с.

20. Танасюк Д. А., Горностаева С. И., Ермаков В. И. Обработка оптических спектров поглощения и анализ составляющих полос на форму линии по Гауссу и Лоренцу. Исследовано в России. 2006. С. 2018–2022.

21. Танасюк Д. А., Горностаева С. В., Ревина А. А и др. О форме линий оптических спектров поглощения наночастиц серебра, полученных в

обратных мицеллах радиационно-химическим методом. Исследовано в России. 2006. С. 2012–2017.

22. Покатилов В. С. Параметры сверхтонких взаимодействий в сплаве Лавеса ScFe_2 с гексагональной структурой MgZn_2 // ФТТ. 2001. Vol. 34, No. 11. С. 1922.

23. Витушкина С. В., Стародуб В. А. Синтез и свойства биядерного комплекса бис[(1,3-дителиол-2-тион4,5-дитиолато)-ди-(карбонил)-циклопентадиенил-железа(II)] // Вісник Харківського національного університету. 2005. Vol. 669, No. 13(36). С. 148–150.

24. Прохоров К. А., Николаева Г. Ю., Гордеев С. А. и др. Комбинационное рассеяние света в ориентированном полиэтилене: область валентных колебаний группы CH_2 // Физико-химия полимеров: синтез, свойства и применение. 2003. No. 9. С. 50–54.

25. Егорова Н. И. Исследование фазовых характеристик фототропных материалов с помощью дисперсионных соотношений. СПб: СПбГУ.

26. Дикусар Н. Д. Оптимизация решений в задачах кусочно-полиномиальной аппроксимации // Супервычисления и математическое моделирование : материалы XVI Международной конференции, Саров, 3–7 октября 2016 г.

27. Злоказов В. Б. Математические методы и программное обеспечение для компьютерного анализа спектроподобных распределений: дис. д-ра ф.-м.н.. Дубна, Объединенный институт ядерных исследований. 210 с.

28. Рудчик А. Т. Обробка експериментальної інформації на ЕЦОМ. Київ, 1976. 322 с.

29. Шовкопляс О. А. Вплив умов формування та терморадіаційної обробки на структуру й властивості покриттів систем Ti-W-C і Ti-W-B : дис. к.ф.-м.н. : 01.04.07: Сумський державний університет. 2016. 179 с.

30. Тихонов А. Н., Костомаров Д. П. Вводные лекции по прикладной математике. М. : Наука, 1984. 192 с.

31. Тяжелова В. Г. Определение параметров спектральных составляющих путём математической обработки отдельных участков спектральной кривой // ЖПС. 1969. Т.10, N1. С. 22 – 27.
32. Антипова-Каратаева И. И., Архипова С. Ф., Гречушников Б. Н. О неоднозначности математического разложения перекрытых спектральных полос методом затухающих наименьших квадратов // ЖПС. 1969. Т.10, N3. С. 480 – 487.
33. Stone H. Mathematical Resolution of Overlapping Spectral Lines // JOSA. 1962. V. 52. pp. 998–1003.
34. Pitha J., Jones R. N. A comparison of optimization methods for fitting curves to infrared band envelopes // Canad. J. Chem. 1966. V. 44, N 24. pp. 3031–3050.
35. Гречушников Б. Н., Калинина И. Н. Разложение сложного спектрального контура на составляющие заданной формы методом преобразования Фурье // ЖПС. 1984. Т.41, N2. С. 249 – 251.
36. Серебряников М. Г., Первозванский А. А. Выявление скрытых периодичностей. М.: Наука, 1965. 244 с.
37. Фок М. Ф. Разделение сложных спектров на индивидуальные полосы при помощи обобщённого метода Аленцева // Труды ФИАН. Люминесценция и нелинейная оптика. 1972. Т.59. С. 3 – 29.
38. Буслов Д. К., Никоненко Н. А., Жбанков Р. Г. Исследование формы спектральных полос, получаемых при деконволюции сложных спектров // ЖПС. 1992. Т.57, N3–4. С. 214 – 220.
39. Натансон И. П. Конструктивная теория функций. М. ; Л. : Гос. изд.-во техн.-теорет. лит., 1949. 688 с.
40. Бахвалов Н. С. Численные методы. М.: Наука, 1973. 631 с.
41. Калиткин Н. Н. Численные методы. М.: Наука, 1978. 512 с.
42. Сахарский А. А., Гулин А. В. Численные методы. М.: Наука, 1989. 430 с.

43. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. М.: Наука, 1984. 318 с.
44. Гречушников Б. Н., Михайленко В. И., Калинина И. Н. Условия однозначности разложения сложного спектрального контура на элементарные полосы // ЖПС. 1989. Т.50, N3. С. 412 – 416.

ДОДАТОК А

ЛІСТИНГ МОДУЛІВ ЕКСПЕРТНОЇ СИСТЕМИ

Модуль математичних процедур та функцій Types_Procs.pas

```

unit types_procs;

interface

uses grids, SysUtils, series, UnMain, Math, Dialogs;

type data_mas=array[1..200,0..1] of real;
matr_mas=array[1..50,1..50] of real;
      pol_mas=array[1..50,1..3] of real;

function  dihotomia(a,b,eps:real;matr:data_mas):real;
function  lagr(t:real; mas:data_mas):real;
function  getDatafromfile(name:string; var mas:data_mas; var
n_d:integer):boolean;
function  der(x:real; data:data_mas):real;
procedure dots_add(var mas:data_mas; q:integer);
function  peak(y0,x0,h,x:real):real;
procedure form_matr(var matr: matr_mas; pol:pol_mas);
procedure gauss(var m:matr_mas; var r: integer);
procedure BackStep(var m:matr_mas);
procedure Scale(var m: matr_mas);
procedure pre_add(var mas:data_mas; var q:integer);
function  findmax(mas:data_mas; st,nd:integer):real;

var N:integer; {загальна кількість експериментальних точок,
включаючи додані}
      n_d: integer; {вихідна кількість точок}
      mas: data_mas; {робочий масив, що містить вихідні вузли
та додані вузли}
      matr: matr_mas; {рабоча матриця розрахунку коефіцієнтів
Ci в МНК}
      n_p: integer; {кількість смуг}
      pol: pol_mas; {масив параметрів смуг}
      g_matr: matr_mas;
      m:matr_mas; {масив для розв'язку системи, МНК}
implementation

function findmax(mas:data_mas; st,nd:integer):real;
var i: integer;
      tmp:real;
begin
tmp:=mas[st,1];
for i:=st to nd do
      if mas[i,1]>tmp then

```

```

    tmp:=mas[i,1];
findmax:=tmp;
end;

function peak(y0,x0,h,x:real):real;
begin
result:=y0*exp(-ln(2)*sqr(x-x0)/sqr(h));
end;

{поліном Лагранжа}
function lagr(t:real; mas:data_mas):real;
    {произведение}
    function pr(t:real; l:integer):real;
        var tmp, tmp1:real;
        i:integer;
        begin
            tmp:=1;
            tmp1:=1;
            for i:=1 to l-1 do
                tmp:=tmp*(t-mas[i,0]);
            for i:=l+1 to N do tmp:=tmp*(t-mas[i,0]);
            for i:=1 to l-1 do
                tmp1:=tmp1*(mas[l,0]-mas[i,0]);
            for i:=l+1 to N do
                tmp1:=tmp1*(mas[l,0]-mas[i,0]);
            pr:=tmp/tmp1;
        end;
    var i:byte;
        tmp:real;
    begin
        tmp:=0;
        for i:=1 to N do
            begin
                tmp:=tmp+pr(t,i)*mas[i,1];
            end;
        if tmp>=0
            then lagr:=tmp
            else lagr:=0;
        end;

    {похідна в точці}
function der(x:real; data:data_mas):real;
    var h:real;
    begin
        h:=0.01;
        der:=(lagr(x+h,data)-lagr(x-h,data))/(2*h);
    end;

    {метод дихотомії}
function dihonomia(a,b,eps:real;matr:data_mas):real;
    var x1,x2,l,xm:real;
    begin

```

```

xm:=a+(b-a)/2;
l:=b-a;
repeat
  x1:=a+l/4;
  x2:=b-l/4;
  if lagr(x1, matr)>lagr(xm, matr) then
    begin
      b:=xm;
      xm:=x1;
    end
  else
    if lagr(x2, matr)>lagr(xm, matr)
    then
      begin
        a:=xm;
        xm:=x2;
      end
    else
      begin
        a:=x1;
        b:=x2;
      end;
    end;
  l:=b-a;
until abs(l)<eps;
result:=(a+b)/2;
end;

```

```

{читання даних из файла}
function getDataFromFile(name:string; var mas:data_mas; var
n_d:integer):boolean;
var f:textfile;
    x,y:real;
    good:boolean;
begin
  good:=true;
  assignfile(f,name);
  reset(f);
  while not eof(f) do
    begin
      try readln(f,x,y);
        except on einouterror do
          begin
            showMessage('Помилка даних! Розділювач цілої і
десятькової частин чисел у файлі має бути "."!');
            good:=false;
          end
        else
          begin
            showMessage('Помилка даних у файлі!');
            good:=false;
          end;
        end;
      end;
    end;
end;

```

```

    end;
  if good then
    begin
      reset(f);
      n_d:=0;
      while not eof(f) do
        begin
          readln(f,x,y);
          if x<>0 then
            begin
              inc(n_d);
              mas[n_d,0]:=x;
              mas[n_d,1]:=y;
            end;
          end;
        end;
      closefile(f);
    getDatafromfile:=good;
  end;

  {обробка ситуації, коли до вихідних точок додано точки}
  procedure pre_add(var mas:data_mas; var q:integer);
  var i:integer;
  begin
    for i:=1 to n_d do
      begin
        mas[i,0]:=mas[i+q,0];
        mas[i,1]:=mas[i+q,1];
      end;
    N:=n_d;
    q:=0;
  end;

  {додавання q точок до вихідних справа та зліва}
  procedure dots_add(var mas:data_mas; q:integer);
  var i:integer;
      s,l:real;
  begin
    s:=mas[2,0]-mas[1,0];
    l:=mas[1,1];
    N:=n_d+q;
    for i:=n_d downto 1 do
      begin
        mas[i+q,0]:=mas[i,0];
        mas[i+q,1]:=mas[i,1];
      end;
    for i:=q downto 1 do
      begin
        mas[i,0]:=mas[i+1,0]-s;
        mas[i,1]:=l;
      end;
    l:=mas[n_d+q,1];
  end;

```

```

N:=n_d+2*q;
for i:=N-q+1 to N do
  begin
    mas[i,0]:=mas[i-1,0]+s;
    mas[i,1]:=1;
  end;
end;

{формування матриці Грама}
procedure form_matr(var matr: matr_mas; pol:pol_mas);

  function skal_proizv(i,j:integer):real;
  var tmp:real;
      a:integer;
  begin
    tmp:=0;
    for a:=1 to n_d do
      tmp:=tmp+peak(1,pol[i,1],pol[i,2],mas[q+a,0])*
        peak(1,pol[j,1],pol[j,2],mas[q+a,0]);
    result:=tmp;
  end;

  var i,j,a:integer;
  begin
    {}
    for i:=1 to n_p do
      for j:=i to n_p do
        begin
          matr[i,j]:=skal_proizv(i,j);
          matr[j,i]:=matr[i,j];
        end;
    {стовбчик правої частини}
    for i:=1 to n_p do matr[i,n_p+1]:=0;
    for i:=1 to n_p do
      for a:=1 to n_d do

matr[i,n_p+1]:=matr[i,n_p+1]+peak(1,pol[i,1],pol[i,2],mas[q+a
,0])*
      mas[q+a,1];;
    end;

{метод Гаусса розв'язання слар, прямий хід}
procedure gauss(var m:matr_mas; var r: integer);
  var i,j:integer;
      tm:extended;
  begin
    if r<n_p then
      begin
        for i:=r+1 to n_p do
          begin
            if m[r,r]<>0 then
              begin

```

```

tm:=m[r,i]/m[r,r];
for j:=1 to n_p+1 do
  if (abs(m[j,i]-m[j,r]*tm) < 1e-17)
  then
    m[j,i]:=0
  else m[j,i]:=m[j,i]-m[j,r]*tm;
end
end;
r:=r+1;
Gauss(m,r);
end;
end;

{зворотній хід методу Гаусса}
procedure BackStep(var m:matr_mas);
var i,j:integer;
begin
for i:=n_p downto 1 do
  begin
m[n_p+1,i]:=m[n_p+1,i]/m[i,i];
m[i,i]:=1;
for j:=i-1 downto 1 do
  begin
m[n_p+1,j]:=m[n_p+1,j]-m[n_p+1,i]*m[i,j];
m[i,j]:=0;
end;
end;
end;
end;
{масштабування елементів матриці}
procedure Scale(var m: matr_mas);
var i,j:integer;
max:extended;
begin
for i:=1 to n_p do
  begin
max:=m[1,i];
for j:=2 to n_p+1 do
  if Abs(m[j,i])>max then
    max:=m[j,i];
  if max<>0 then
    for j:=1 to n_p+1 do
      m[j,i]:=m[j,i]/max;
    end;
  end;
end;
end;
end.

```

Модуль головної форми UnMain.pas

```
unit UnMain;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, Menus, ExtCtrls, TeeProcs, TeEngine, Chart,
  ArrowCha, Series,
  StdCtrls, Grids, ComCtrls, teeprevi, un_endata, math;
```

```
type
```

```
  TMainForm = class(TForm)
    MainMenu: TMainMenu;
    N1: TMenuItem;
    OpenBase: TMenuItem;
    CreateBase: TMenuItem;
    ExitPrj: TMenuItem;
    EditBase: TMenuItem;
    OpenDialog1: TOpenDialog;
    SaveBase: TMenuItem;
    SaveDialog1: TSaveDialog;
    Chart1: TChart;
    enData: TPointSeries;
    lagrData: TArrowSeries;
    data: TStringGrid;
    b_findmaxs: TButton;
    sg_maxs: TStringGrid;
    b_f_eq: TButton;
    sg_elabor: TStringGrid;
    peaks: TArrowSeries;
    LabeledEdit1: TLabelledEdit;
    b_elaborate: TButton;
    Label1: TLabel;
    StatusBar1: TStatusBar;
    Maxs: TArrowSeries;
    cb_maxs: TCheckBox;
    PrintBase: TMenuItem;
    exportresults: TMenuItem;
    pmadd: TPopupMenu;
    N3: TMenuItem;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    cb_lagr: TCheckBox;
    Label5: TLabel;
    e_s: TEdit;
    e_e: TEdit;
    Label6: TLabel;
    sg_start: TStringGrid;
    Label7: TLabel;
    SaveResults: TMenuItem;
    procedure ExitPrjClick(Sender: TObject);
    procedure CreateBaseClick(Sender: TObject);
```



```

    procedure OpenBaseClick(Sender: TObject);
    procedure SaveBaseClick(Sender: TObject);
    procedure b_findmaxsClick(Sender: TObject);
    procedure b_f_eqClick(Sender: TObject);
    procedure b_elaborateClick(Sender: TObject);
    procedure sg_elaborKeyPress(Sender: TObject; var Key:
Char);
    procedure cb_maxsClick(Sender: TObject);
    procedure PrintBaseClick(Sender: TObject);
    procedure exportresultsClick(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure b_elaborateKeyPress(Sender: TObject; var Key:
Char);
    procedure cb_lagrClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure EditBaseClick(Sender: TObject);
    procedure e_eKeyPress(Sender: TObject; var Key: Char);
    procedure SaveResultsClick(Sender: TObject);
    procedure N1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    MainForm: TMainForm;
    q:integer;
    var f:TCheckBox;
implementation

{$R *.dfm}

uses types_procs, unit1;
{знаходження мінімального елемента, для настройки графіка}
function findmin(mas:data_mas; st,nd:integer):real;
var i: integer;
    tmp:real;
begin
    tmp:=mas[st,1];
    for i:=st to nd do
        if mas[i,1]<tmp then
            tmp:=mas[i,1];
    findmin:=tmp;
end;

procedure TMainForm.ExitPrjClick(Sender: TObject);
begin
    close;
end;

```

```

{створення бази - перехід відповідну на форму}
procedure TMainForm.CreateBaseClick(Sender: TObject);
begin
  EnDataForm.Show;
  EnDataForm.Caption:='Створення бази даних';
end;

{очищення сітки, допоміжна процедура}
procedure clear_grid(var gr:tStringgrid);
var i,j:integer;
begin
  for i:=0 to gr.ColCount-1 do
    for j:=1 to gr.RowCount-1 do
      gr.Cells[i,j]:='';
  end;
end;

{відкриття бази}
procedure TMainForm.OpenBaseClick(Sender: TObject);
var i:integer;
begin
  labelededit1.SetFocus;
  enData.Clear;
  lagrData.Clear;
  peaks.Clear;
  maxs.Clear;
  label2.Caption:='';
  cb_maxs.checked:=false;
  cb_lagr.Checked:=false;
  b_f_eq.Enabled:=true;
  b_elaborate.Enabled:=true;
  q:=0;
  OpenDialog1.InitialDir:=ExtractFilePath(ParamStr(0));
  OpenDialog1.Execute;
  if OpenDialog1.FileName<>' ' then
    if
      getDatafromfile(OpenDialog1.FileName,mas,types_procs.n_d)=true
    then
      begin
        data.ColCount:=types_procs.n_d+1;
        enData.Clear;
        for i:=1 to types_procs.n_d do
          begin
            data.Cells[i,0]:=floattostrf(types_procs.mas[i,0],ffFixed,3,0);

            data.Cells[i,1]:=floattostrf(types_procs.mas[i,1],ffFixed,3,3);

            enData.AddXY(mas[i,0],mas[i,1],',',clRed);
          end;
        sg_maxs.RowCount:=1;
      end;
    end;
  end;
end;

```

```

sg_elabor.RowCount:=1;
chart1.LeftAxis.Automatic:=false;
chart1.BottomAxis.Automatic:=false;
Chart1.LeftAxis.Minimum:=0;

Chart1.LeftAxis.Maximum:=1.2*types_procs.findmax(mas,1,n_d);
Chart1.BottomAxis.Minimum:=0;
Chart1.BottomAxis.Maximum:=mas[n_d,0];
Chart1.BottomAxis.Minimum:=mas[1,0];
chart1.Title.Text.Clear;
chart1.Title.Text.Insert(0,OpenDialog1.FileName);
e_s.Text:=floattostr(mas[1,0]);
e_e.Text:=floattostr(mas[n_d,0]);
clear_grid(sg_elabor);
clear_grid(sg_maxs);
end;
end;

{збереження бази}
procedure TMainForm.SaveBaseClick(Sender: TObject);
var f:textfile;
    i:integer;
begin
mainform.SaveDialog1.Execute;
if mainform.SaveDialog1.FileName<>' ' then
begin
assignfile(f,mainform.SaveDialog1.FileName);
rewrite(f);
for i:=1 to mainform.data.ColCount-1 do
begin
if i<>mainform.data.ColCount-1 then
writeln(f,mainform.data.Cells[i,0],#9,
mainform.data.Cells[i,1])
else write(f,mainform.data.Cells[i,0],#9,
mainform.data.Cells[i,1]);
end;
closefile(f);
chart1.LeftAxis.Automatic:=false;
Chart1.LeftAxis.Minimum:=0;
Chart1.LeftAxis.Maximum:=1.2*findmax(mas,1,n_d);
end;
end;

{побудувати поліном Лагранжа на графіку}
procedure showlagr(mas:data_mas;
var series1:TArrowseries;
st,nd:integer);

var x,h:real;
begin
x:=mas[st,0];
h:=(mas[nd,0]-mas[st,0])/((nd-st)*5);
while x<mas[nd,0] do

```

```

begin
  series1.AddArrow(x, lagr(x, mas), x+h, lagr(x+h, mas));
  x:=x+h;
end;
end;

{нанесення експериментальних точок на графік}
procedure showmas(mas:data_mas; st,nd:integer; var
s:TPointSeries);
var i:integer;
begin
s.Clear;
for i:=st to nd do
  s.AddXY(mas[i,0],mas[i,1]);
with mainform.Chart1 do
begin
  LeftAxis.Minimum:=0;
  LeftAxis.Maximum:=1.2*types_procs.findmax(mas,st,nd);
  BottomAxis.Minimum:=0;
  BottomAxis.Maximum:=mas[nd,0];
  BottomAxis.Minimum:=mas[st,0];
end;
end;

{нанесення вертикальних ліній, що відповідають максимумам
таблиці sg_maxs}
procedure maxs_show(source:tstringgrid; var gr:tarrowseries);
var i:integer;
begin
gr.Clear;
for i:=1 to source.RowCount-1 do
begin
  if source.Cells[2,i]='1' then
    gr.AddArrow(strtofloat(source.Cells[1,i]),0,
strtofloat(source.Cells[1,i]),lagr(strtofloat(source.Cells[1,
i]),mas))
  end;
end;

procedure pre_find_maxs;
begin
if q=0 then
begin
  q:=strtoint(mainform.LabeledEdit1.Text);
  dots_add(types_procs.mas,q); {добавляем справа и слева q
точек}
end
else
begin
  //types_procs.pre_add(mas,q);
  q:=strtoint(mainform.LabeledEdit1.Text);

```

```

dots_add(types_procs.mas,q); {добавляем справа и слева q
точек}
end;
end;

```

```

{Пошук максимумів}
procedure TMainForm.b_findmaxsClick(Sender: TObject);
var st,nd,i,k:integer;
    ft,lt, index1, index2:integer;

```

```

{поиск индекса в массиве, соотв. числу z}
function f_index(z:real; var index:integer):boolean;
var i:integer;
    b:boolean;

```

```

begin
b:=false;
for i:=1 to n_d do
if z=mas[i,0] then
begin
index:=i;
b:=true;
break;
end;
f_index:=b;
end;

```

```

{перенесення даних з таблиці в масив}
procedure gridtomas(s:tstringgrid; var mas:data_mas);
var i:integer;
begin
for i:=1 to s.ColCount-1 do
begin
mas[i,0]:=strtofloat(s.Cells[i,0]);
mas[i,1]:=strtofloat(s.Cells[i,1]);
end;
end;

```

```

begin
gridtomas(data,mas);
n_d:=data.ColCount-1;
mainform.peaks.Clear;
if (f_index(strtofloat(e_s.Text),index1)=true) and
(f_index(strtofloat(e_e.Text),index2)=true) and
(strtofloat(e_s.Text)<strtofloat(e_e.Text)) then
begin

if (mas[1,0]<>strtofloat(e_s.Text)) or
(mas[n_d,0]<>strtofloat(e_e.Text)) then
begin
ft:=index1;
lt:=index2;
for i:=ft to lt do

```

```

begin
  mas[i-ft+1,0]:=mas[i,0];
  mas[i-ft+1,1]:=mas[i,1];
end;
n_d:=lt-ft+1;
end;
pre_find_maxs;
showmas(mas,1+q,n_d+q,enData);
maxs.Clear;
sg_maxs.RowCount:=1;
k:=0;
st:=q+1;
nd:=N-q;
for i:=st to nd-1 do
begin
  if ( der(mas[i,0],mas)>0 )
    and ( der(mas[i+1,0],mas)<0 ) then
begin
  inc(k);
  sg_maxs.RowCount:=sg_maxs.RowCount+1;
  sg_maxs.Cells[1,k]:=floattostrf(dihotomia(mas[i,0],
    mas[i+1,0],0.01,mas),ffixed,5,2);
  sg_maxs.Cells[0,k]:=inttostr(k);
  sg_maxs.Cells[2,k]:='1';
end;
end;
if cb_lagr.Checked=true then
begin
  lagrdata.Clear;
  showlagr(mas,lagrdata,1+q,n_d+q);
end;
if k>=1 then
begin
  //maxs_show(sg_maxs,maxs);
  b_f_eq.Enabled:=true;
  b_elaborate.Enabled:=true;
end
else
begin
  showmessage('Максимумів не знайдено!');
  b_elaborate.Enabled:=false;
  b_f_eq.Enabled:=false;
end;
end
else
begin
  showmessage('Невірно ввибрано діапазон');
  e_e.SetFocus;
end;
end;

```

{нанесення на графік теоретичних смуг та їх суми}

```

procedure show_grafik(pol:pol_mas; var peaks:tArrowSeries);

  procedure show_sum_peaks;

    function sum(x:real):real;
    var i:integer;
        tmp:real;
    begin
      tmp:=0;
      for i:=1 to n_p do

tmp:=tmp+types_procs.peak(pol[i,3],pol[i,1],pol[i,2],x);
      sum:=tmp;
    end;

    var x,dx:real;
    begin
      dx:=0.1*(mas[q+2,0]-mas[q+1,0]);
      x:=mas[q+1,0];
      while x<mas[N-q,0] do
        begin
          peaks.AddArrow(x,sum(x), x+dx,sum(x+dx),'',clblack);
          x:=x+dx;
        end;
    end;

    procedure show_peak(y0,x0,h:real; var series:TArrowseries;
color:tcolor);
      var x,dx:real;
      begin
        dx:=0.1*(mas[q+2,0]-mas[q+1,0]);
        x:=mas[q+1,0];
        while x<mas[N-q,0] do
          begin
            series.AddArrow(x,types_procs.peak(y0,x0,h,x),
              x+dx,types_procs.peak(y0,x0,h,x+dx),'',clblack);
            x:=x+dx;
          end;
        end;

    var i:integer;
    const color_mas:array[1..5]of
tcolor=(clred,clblue,clgreen,clmaroon,clyellow);
    begin
      peaks.Clear;
      for i:=1 to n_p do
        show_peak(pol[i,3],pol[i,1],pol[i,2],peaks,color_mas[i]);
      show_sum_peaks;
    end;

    {кнопка Перше наближення}
    procedure TMainForm.b_f_eqClick(Sender: TObject);

```

```

var r,i,j,s:integer;
    x0i,x,y0i,h,d:real;

begin
MainForm.cb_maxs.OnClick(mainform);
peaks.Clear;
d:=2*0.1;
sg_elabor.RowCount:=1;
s:=sg_elabor.RowCount;
n_p:=0;
for i:=1 to sg_maxs.RowCount-1 do
  if sg_maxs.Cells[2,i]='1' then
    begin
      inc(s);
      inc(n_p);
      sg_elabor.RowCount:=sg_elabor.RowCount+1;
      x:=strtofloat(sg_maxs.Cells[1,i]);
      x0i:=x+d/2*ln( lagr(x-d,mass)/lagr(x+d,mass) )/
        ln( lagr(x-d,mass)*lagr(x+d,mass)/sqr(lagr(x,mass)) );
      if ( -2*d*(x-x0i+d/2)*ln(2) )/ln( lagr(x+d,mass)/lagr(x,mass)
    )<0 then
        begin
          showmessage('!!!! max№'+inttostr(n_p)+#13+
            floattostrf(-2*d*(x-x0i+d/2)*ln(2),ffFixed,4,4)+#13+
            floattostrf(ln( lagr(x+d,mass)/lagr(x,mass)),ffFixed,4,4));
          h:=5;
        end else
          h:=sqrt( ( -2*d*(x-x0i+d/2)*ln(2) )/ln(
            lagr(x+d,mass)/lagr(x,mass) ) );
          y0i:=lagr(x,mass)*exp( ln(2)*sqr((x-x0i)/h) );
          sg_elabor.Cells[1,s-1]:=floattostrf(x0i,ffFixed,4,1);
          sg_elabor.Cells[2,s-1]:=floattostrf(h,ffFixed,4,1);
          sg_elabor.Cells[3,s-1]:=floattostrf(y0i,ffFixed,4,1);
          pol[n_p,1]:=x0i;
          pol[n_p,2]:=h;
          pol[n_p,3]:=y0i;
        end;
    {формування матриці Грама}
    types_procs.form_matr(matr,pol);
    for i:=1 to n_p do
      for j:=1 to n_p+1 do
        m[j,i]:=matr[i,j];
    r:=1;
    scale(m);
    {Розв'язок матриці}
    gauss(m,r);
    types_procs.backstep(m);
    j:=0;
    for i:=1 to n_p do
      {вибір параметрів з додатними коефіцієнтами}
      if m[n_p+1,i]>=0 then

```



```

begin
  inc(j);

sg_elabor.Cells[3,j]:=floattostrf(m[n_p+1,i],ffFixed,4,1);
  sg_elabor.Cells[2,j]:=sg_elabor.Cells[2,i];
  sg_elabor.Cells[1,j]:=sg_elabor.Cells[1,i];
  pol[j,2]:=pol[i,2];
  pol[j,1]:=pol[i,1];
  pol[j,3]:=m[n_p+1,i]
end
  else sg_maxs.Cells[2,i]:='0';
n_p:=j;
sg_elabor.RowCount:=j+1;
sg_start.RowCount:=j+1;
for i:=1 to j do
begin
  sg_elabor.Cells[0,i]:=inttostr(i);
  sg_start.Cells[0,i]:=inttostr(i);
  sg_start.Cells[1,i]:=sg_elabor.Cells[1,i];
  sg_start.Cells[2,i]:=sg_elabor.Cells[2,i];
  sg_start.Cells[3,i]:=sg_elabor.Cells[3,i];
end;
show_grafik(types_procs.pol,mainform.peaks);
maxs_show(sg_maxs,maxs);
cb_maxs.OnClick(mainform);
cb_maxs.Checked:=true;

end;

{уточнення параметрів вибраних смуг}
procedure TMainForm.b_elaborateClick(Sender: TObject);
function sumsqr:real;
var i,j:integer;
    tmp1,tmp2:real;
begin
  tmp2:=0;
  for i:=1 to n_d do
  begin
    tmp1:=0;
    for j:= 1 to n_p do
    begin
      tmp1:=tmp1+peak(pol[j,3],pol[j,1],pol[j,2],mas[q+i,0]);
    end;
    tmp2:=tmp2+sqr(tmp1-mas[q+i,1]);
  end;
  sumsqr:=tmp2;
end;

Var poll: pol_mas;
    SqSum,SqSum1,Y0_Add,h_Add,X0_Add:real;
    j,i,k:integer;

```

```

changed:boolean;
counter:integer;

begin
statusbar1.SimpleText:='Виконуються розрахунки... Для зупинки
натисніть ESC';
label1.Caption:='Сума квадратів відхилень: ';
mainform.SetFocus;
counter:=0;
Y0_Add:=1;
h_Add:=1;
X0_Add:=2;
sqsum:=sumsqr;
for i:=1 to n_p do
begin
pol1[i,1]:=pol[i,1];
pol1[i,2]:=pol[i,2];
pol1[i,3]:=pol[i,3];
end;
k:=0;
repeat
changed:=true;
inc(k);
for j:=1 to n_p do
begin
pol1[j,3]:=pol[j,3];
pol[j,3]:=pol[j,3]+Y0_Add;
SqSum1:=SqSum;
sqsum:=sumsqr;
if SqSum>SqSum1 then
begin
if pol1[j,3]-Y0_Add>=0 then
pol[j,3]:=pol1[j,3]-Y0_Add;
sqsum:=sumsqr;
if SqSum>SqSum1 then
begin
pol[j,3]:=pol1[j,3];
SqSum:= SqSum1;
changed:=false;
end;
end;
end;
end;
for j:=1 to n_p do
begin
pol1[j,2]:=pol[j,2];
pol[j,2]:=pol[j,2]+h_Add; {n}
SqSum1:=SqSum;
sqsum:=sumsqr;
if SqSum>SqSum1 then
begin
if pol1[j,2]-h_Add>=0 then
pol[j,2]:=pol1[j,2]-h_Add;

```

```

    sqsum:=sumsqr;
    if SqSum>SqSum1 then
        begin
            pol[j,2]:=pol1[j,2];
            SqSum:=SqSum1;
            changed:=false;
        end
    end
end;
for j:=1 to n_p do
begin
    pol1[j,1]:=pol[j,1];
    pol[j,1]:=pol[j,1]+X0_Add;
    SqSum1:=SqSum;
    sqsum:=sumsqr;
    if SqSum>SqSum1 then
        begin
            if pol1[j,1]-X0_Add>=0 then
                pol[j,1]:=pol1[j,1]-X0_Add;
            sqsum:=sumsqr;
            if SqSum>SqSum1 then
                begin
                    pol[j,1]:=pol1[j,1];
                    SqSum:=SqSum1;
                    changed:=false;
                end
            end
        end
    end;
If Y0_Add-0.01 > 0 Then Y0_Add:=Y0_Add-0.01;
If h_Add-0.05 > 0 Then h_Add:=h_Add-0.05;
If X0_Add-0.01 > 0 Then X0_Add:=X0_Add-0.01;
label2.Caption:=floattostrf (sumsqr, ffFixed, 5, 4);
Application.ProcessMessages;
if (k mod 10) = 0 then
begin
    show_grafik(types_procs.pol,mainform.peaks);
    for i:=1 to n_p do
        begin
            sg_elabor.cells[0,i]:=inttostr(i);

sg_elabor.cells[3,i]:=floattostrf(pol[i,3], ffFixed, 5, 2);

sg_elabor.cells[1,i]:=floattostrf(pol[i,1], ffFixed, 5, 2);

sg_elabor.cells[2,i]:=floattostrf(pol[i,2], ffFixed, 5, 2);
            end;
        end;
        if (changed=false) then inc(counter);
        until (counter>1000) or (mainform.Tag=1);
    for i:=1 to n_p do
        begin
            sg_elabor.cells[0,i]:=inttostr(i);

```

```

    sg_elabor.cells[3,i]:=floattostrf(pol[i,3],ffFixed,5,2);
    sg_elabor.cells[1,i]:=floattostrf(pol[i,1],ffFixed,5,2);
    sg_elabor.cells[2,i]:=floattostrf(pol[i,2],ffFixed,5,2);
    end;
show_grafik(pol,peaks);
MainForm.Tag:=0;
statusbar1.SimpleText:='';
end;

procedure TMainForm.sg_elaborKeyPress(Sender: TObject; var
Key: Char);
var i:integer;
begin
if key=#13 then
    begin
    for i:=1 to sg_elabor.RowCount-1 do
        begin
            pol[i,1]:=strtofloat(sg_elabor.cells[0,i]);
            pol[i,2]:=strtofloat(sg_elabor.cells[1,i]);
            pol[i,3]:=strtofloat(sg_elabor.cells[2,i]);
        end;
    end;
end;

{відображення максимумів полінома Лагранжа}
procedure TMainForm.cb_maxsClick(Sender: TObject);
begin
with cb_maxs do
if checked=true then
    maxs_show(sg_maxs,maxs)
    else maxs.Clear;
end;

{друк графіка з результатами моделювання}
procedure TMainForm.PrintBaseClick(Sender: TObject);
var i:integer;
begin
//chartPreview(MainForm,Chart1);
form1.peaks.clear;
show_grafik(types_procs.pol,unit1.form1.peaks);
showmas(mas,1+q,n_d+q,unit1.form1.endata);
unit1.Form1.QRRichText1.Lines.Clear;
unit1.Form1.QRRichText1.Lines.Add('x'+#9+'h'+#9+'y');
for i:=1 to sg_elabor.RowCount-1 do
    unit1.Form1.QRRichText1.Lines.Add(sg_elabor.cells[1,i]+#9+
        sg_elabor.cells[2,i]+#9+sg_elabor.cells[3,i]);
unit1.form1.QRLabel2.Caption:=mainform.Chart1.Title.Text[0];
unit1.Form1.QuickRep1.Prepare;
unit1.Form1.QuickRep1.Preview;
end;

{експорт графіка у формат bmp}

```

```

procedure TMainForm.exportresultsClick(Sender: TObject);
begin
savedialog1.FilterIndex:=2;
savedialog1.Execute;
chart1.SaveToBitmapFile(savedialog1.FileName+'.bmp');
end;

{додавання нової смуги вручну}
procedure TMainForm.N3Click(Sender: TObject);
begin
sg_maxs.RowCount:=sg_maxs.RowCount+1;
sg_maxs.Cells[0,sg_maxs.RowCount-1]:=inttostr(sg_maxs.RowCount-1);
sg_maxs.Cells[1,sg_maxs.RowCount-1]:=inputbox('Додавання нової смуги','Координата, нм:', '');
sg_maxs.Cells[2,sg_maxs.RowCount-1]:='1';
end;

{початкова ініціалізація}
procedure TMainForm.FormCreate(Sender: TObject);
begin
data.Cells[0,0]:='Довжина хвилі, нм';
data.Cells[0,1]:='Оптична густина, відн.од.';
WITH data do
  ColWidths[0] := Canvas.TextWidth(Cells[0,1]+' ');
sg_start.Cells[1,0]:='x_0';
sg_start.Cells[2,0]:='h';
sg_start.Cells[3,0]:='y_0';
sg_start.Cells[0,0]:='№';
sg_elabor.Cells[1,0]:='x_0';
sg_elabor.Cells[2,0]:='h';
sg_elabor.Cells[3,0]:='y_0';
sg_elabor.Cells[0,0]:='№';
sg_maxs.Cells[0,0]:='Max№';
sg_maxs.Cells[1,0]:='Координата';
sg_maxs.Cells[2,0]:='Врахувати';
end;

{обробка натискання клавіші Esc при уточненні параметрів смуг}
procedure TMainForm.b_elaborateKeyPress(Sender: TObject; var Key: Char);
begin
if key=#27 then
  mainform.Tag:=1;
end;

{зображення полінома Лагранжа}
procedure TMainForm.cb_lagrClick(Sender: TObject);
begin
with cb_lagr do
if checked=true then

```

```

    showlagr (mas, lagrData, q+1, N-q)
  else
    lagrdata.Clear;
end;

procedure TMainForm.FormShow(Sender: TObject);
begin
  MainForm.Top:=20;
  MainForm.Left:=20;
end;

{редагування даних експерименту}
procedure TMainForm.EditBaseClick(Sender: TObject);
var i:integer;
begin
  if MainForm.data.ColCount=2 then
  begin
    EnDataForm.Tag:=0;
    EnDataForm.Show;
  end
  else
  begin
    EnDataForm.Caption:='Редагування даних';
    EnDataForm.data.ColCount:=MainForm.data.ColCount;
    for i:=1 to MainForm.data.ColCount-1 do
    begin
      EnDataForm.data.Cells[i,0]:=MainForm.data.Cells[i,0];
      EnDataForm.data.Cells[i,1]:=MainForm.data.Cells[i,1];
    end;
  end;

  EnDataForm.le_fvalue.Text:=EnDataForm.data.Cells[EnDataForm.d
ata.ColCount-1,0];
  EnDataForm.le_svalue.Text:=EnDataForm.data.Cells[1,0];
  EnDataForm.le_step.Text:=inttostr(
    strtoint(EnDataForm.data.Cells[2,0]) -
    strtoint(EnDataForm.data.Cells[1,0]));
  EnDataForm.Tag:=1;
  EnDataForm.Show;
end;
end;

procedure TMainForm.e_eKeyPress(Sender: TObject; var Key:
Char);
begin
  if key=#13 then MainForm.b_findmaxs.Click;
end;

{збереження параметрів смуг в файл}
procedure TMainForm.SaveResultsClick(Sender: TObject);
var f:textfile;
    i:integer;
    s:string;

```

```

begin
if sg_elabor.RowCount > 1 then
begin
MainForm.SaveDialog1.Execute;
if SaveDialog1.FileName<>' ' then
begin
assignfile(f, SaveDialog1.FileName+'.txt');
rewrite(f);
writeln(f, chart1.Title.Text[0]);
writeln(f, 'Кількість смуг: ', inttostr(n_p));
writeln(f, 'x', chr(9), 'h', chr(9), 'y');
for i:=1 to sg_elabor.RowCount-1 do

writeln(f, sg_elabor.cells[1,i], chr(9), sg_elabor.cells[2,i], ch
r(9), sg_elabor.cells[3,i]);

//write(f, sg_elabor.cells[1,i+1], chr(9), sg_elabor.cells[2,i+1
], chr(9), sg_elabor.cells[3,i+1]);
closefile(f);
end
end
end;

procedure TMainForm.N1Click(Sender: TObject);
begin
if sg_elabor.RowCount=1 then
begin
printbase.Enabled:=false;
saveresults.Enabled:=false;
exportresults.Enabled:=false;
end
else
begin
printbase.Enabled:=true;
saveresults.Enabled:=true;
exportresults.Enabled:=true;
end;
end;

end.

```

Модуль форми створення/редагування даних Un_EnData.pas

```

unit Un_EnData;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,

```

```

Dialogs, StdCtrls, ExtCtrls, Grids, TeEngine, Series,
TeeProcs, Chart,
Menus, Clipbrd;

```

```

type

```

```

  TEnDataForm = class(TForm)
    le_svalue: TLabelledEdit;
    le_fvalue: TLabelledEdit;
    le_step: TLabelledEdit;
    data: TStringGrid;
    bconfirm: TButton;
    PopupMenu1: TPopupMenu;
    N1: TMenuItem;
    Mem1: TMemo;
    procedure FormCreate(Sender: TObject);
    procedure le_stepKeyPress(Sender: TObject; var Key:
Char);
    procedure le_svalueKeyPress(Sender: TObject; var Key:
Char);
    procedure le_fvalueKeyPress(Sender: TObject; var Key:
Char);
    procedure bconfirmClick(Sender: TObject);
    procedure dataSelectCell(Sender: TObject; ACol, ARow:
Integer;
      var CanSelect: Boolean);
    procedure N1Click(Sender: TObject);
    procedure dataEnter(Sender: TObject);
  private
    PrevHwnd : Hwnd;
    procedure WMChangeCBChain(VAR Msg: TWMChangeCBChain);
    message WM_CHANGECHAIN;
    procedure WMDrawClipboard(VAR Msg: TWMDrawClipboard);
    message WM_DRAWCLIPBOARD;
  public
    { Public declarations }
  end;

```

```

var

```

```

  EnDataForm: TEnDataForm;

```

```

implementation

```

```

{$R *.dfm}

```

```

uses unMain, types_procs, unsave;

```

```

var k:integer;

```

```

procedure TEnDataForm.WMChangeCBChain(VAR Msg:
TWMChangeCBChain);

```

```

begin

```

```

IF PrevHwnd = Msg.Remove THEN

```



```

PrevHWnd := Msg.Next;
IF Msg.Remove <> Handle THEN
SendMessage (PrevHWnd, WM_CHANGECHAIN, Msg.Remove,
Msg.Next);
end;

procedure GridFill(svalue, fvalue, step:real; var
grid:TStringgrid);
var i,n:integer;
begin
n:=trunc((fvalue-svalue)/step)+1;
grid.ColCount:=n+1;
for i:=1 to n do
begin
grid.Cells[i,0]:=floattostr(svalue+(i-1)*step);
end;
end;

procedure TEnDataForm.FormCreate(Sender: TObject);
var wid:integer;
begin
data.Cells[0,0]:='Довжина хвилі, нм';
data.Cells[0,1]:='Оптична густина, відн.од.';
WITH data do
BEGIN
Wid := Canvas.TextWidth(Cells[0,1]+' ');
IF Wid > ColWidths[0] THEN
ColWidths[0] := Wid;
END;
end;

procedure TEnDataForm.le_stepKeyPress(Sender: TObject; var
Key: Char);
begin
if key=#13 then
begin
data.SetFocus;
gridfill(strtfloat(le_svalue.Text),
strtfloat(le_fvalue.Text),
strtoint(le_step.Text),
data);
end;
end;

procedure TEnDataForm.le_svalueKeyPress(Sender: TObject; var
Key: Char);
begin
if key=#13 then le_fvalue.SetFocus;
end;

procedure TEnDataForm.le_fvalueKeyPress(Sender: TObject; var
Key: Char);

```

```

begin
if key=#13 then le_step.SetFocus;
end;

procedure TEnDataForm.bconfirmClick(Sender: TObject);

  procedure savebase(fname:string);
  var f:textfile;
      i:integer;
  begin
  assignfile(f,fname);
  rewrite(f);
  for i:=1 to endataform.data.ColCount-2 do

writeln(f,endataform.data.Cells[i,0],chr(27),endataform.data.
Cells[i,1]);

write(f,endataform.data.Cells[i,0],chr(27),endataform.data.Ce
lls[i,1]);
  closefile(f);
  end;

  function check:boolean;
  var i:integer;
  begin
  check:=false;
  for i:=1 to data.ColCount-1 do
    if data.Cells[i,1]<>' ' then check:=true
  end;

var i:integer;
begin
if check=true then
  begin
  mainform.lagrData.Clear;
  mainform.peaks.Clear;
  mainform.enData.Clear;
  mainform.Maxs.Clear;
  mainform.data.ColCount:=endataform.data.ColCount;
  types_procs.n_d:=mainform.data.ColCount-1;
  for i:=0 to endataform.data.ColCount-1 do
    begin
    mainform.data.Cells[i,0]:=endataform.data.Cells[i,0];
    mainform.data.Cells[i,1]:=endataform.data.Cells[i,1];
    if i<>0 then
      begin

types_procs.mas[i,0]:=strtofloat(mainform.data.Cells[i,0]);
types_procs.mas[i,1]:=strtofloat(mainform.data.Cells[i,1]);
mainform.enData.AddXY(types_procs.mas[i,0],types_procs.mas[i,
1]);
      end;
    end;

```

```

    end;
    mainform.Chart1.LeftAxis.Automatic:=false;
    mainform.Chart1.LeftAxis.Minimum:=0;

mainform.Chart1.LeftAxis.Maximum:=1.2*types_procs.findmax(mas
,1,n_d);
    mainform.Chart1.BottomAxis.Automatic:=false;
    mainform.Chart1.BottomAxis.Minimum:=0;

mainform.Chart1.BottomAxis.Maximum:=mainform.Chart1.BottomAxis
.Minimum;
    mainform.Chart1.BottomAxis.Maximum:=mas[n_d,0];
    mainform.Chart1.BottomAxis.Minimum:=mas[1,0];
    unsave.SaveDialog.ShowModal;
    if unsave.SaveDialog.ModalResult=mrYes
    then
        begin

unMain.mainform.savedialog1.InitialDir:=ExtractFilePath(Param
Str(0));
        unMain.mainform.savedialog1.FilterIndex:=1;
        unMain.mainform.savedialog1.execute;
        if unMain.mainform.savedialog1.FileName<>' ' then
            begin
                savebase(unMain.mainform.savedialog1.FileName);

mainform.Chart1.Title.Text.Insert(0,mainform.savedialog1.File
Name);
            end;
        end;
        enddataform.Hide;

MainForm.e_s.Text:={data.Cells[1,0]}floattostr(types_procs.ma
s[1,0]);

MainForm.e_e.Text:={data.Cells[1,0]}floattostr(types_procs.ma
s[n_d,0]);
    end
    else showmessage('Присутні незаповнені комірки!');
end;

procedure TEnDataForm.dataSelectCell(Sender: TObject; ACol,
ARow: Integer;
    var CanSelect: Boolean);
VAR Wid: Integer;
begin
k:=ACol;
WITH Sender AS TStringGrid DO
    BEGIN
        Wid := Canvas.TextWidth(Cells[Col,Row]+' ');
        IF Wid > ColWidths[Col] THEN
            ColWidths[Col] := Wid;

```

```

    END;
end;

procedure stringtograd(s:ansistring; var m:TStringgrid);
var i:integer;
t,t1:real;
f:textfile;
b:boolean;
begin
n:=0;
k:=0;
assign(f,'tmp');
rewrite(f);
write(f,s);
reset(f);
b:=true;
while not eof(f) do
begin
try
begin
read(f,t);
read(f,t1);
end
except on EInOutError do
begin
b:=false;
showmessage('Невірний формат даних у буфері обміну!');
closefile(f);
DeleteFile('tmp');
exit;
end
end;
end;
if b=true then
begin
reset(f);
for i:=1 to length(s) do
begin
if (s[i]=#13) then inc(k);
if (s[i]=#13) or (s[i]=#9) then inc(n);
end;
i:=0;
if k>=2 then m.ColCount:=k+1;
while not eof(f) do
begin
inc(i);
read(f,t);
readln(f,t1);
m.Cells[i,0]:=floattostr(t);
m.Cells[i,1]:=floattostr(t1);
end;
closefile(f);

```

```

DeleteFile('tmp');
EnDataForm.le_svalue.Text:=EnDataForm.data.cells[1,0];
EnDataForm.le_fvalue.Text:=EnDataForm.data.cells[i,0];

EnDataForm.le_step.Text:=floattostr(strtfloat(EnDataForm.data
a.cells[2,0])
-strtfloat(EnDataForm.data.cells[1,0]));
end;

end;

procedure TEnDataForm.WMDrawClipboard(VAR Msg:
TWMDrawClipboard);
VAR
  s:ansistring;
  P : PChar;
  H : THandle;
begin
SendMessage(PrevHwnd, WM_DRAWCLIPBOARD, 0, 0);
IF Clipboard.HasFormat(CF_TEXT) THEN
  BEGIN
  H := Clipboard.GetAsHandle(CF_TEXT);
  P := GlobalLock(H);
  mem01.SetTextBuf(P);
  s:=mem01.Text;
  stringtograd(s,EnDataForm.data);
  GlobalUnlock(H);
  END;
Msg.Result := 0;
end;
{вставити з буфера обміну}
procedure TEnDataForm.N1Click(Sender: TObject);
begin
PrevHwnd := SetClipboardViewer(Handle);
ChangeClipboardChain(Handle, PrevHwnd);
end;

procedure TEnDataForm.dataEnter(Sender: TObject);
begin

if (le_fvalue.Text<>'') and
(le_step.Text<>'') and
(le_svalue.Text<>'') then
  gridfill(strtfloat(le_svalue.Text),
          strtfloat(le_fvalue.Text),
          strtoint(le_step.Text),
          data);

end;

end.

```

Модуль форми діалогу збереження створеної бази UnSave.pas

```
unit UnSave;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
  StdCtrls,
  Buttons, ExtCtrls;

type
  TSaveDialog = class(TForm)
    OKBtn: TButton;
    CancelBtn: TButton;
    Label1: TLabel;
    procedure OKBtnClick(Sender: TObject);
    procedure CancelBtnClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  SaveDialog: TSaveDialog;

implementation

{$R *.dfm}

procedure TSaveDialog.OKBtnClick(Sender: TObject);
begin
  Savedialog.ModalResult:=mrYes;
  savedialog.Hide;
end;

procedure TSaveDialog.CancelBtnClick(Sender: TObject);
begin
  Savedialog.ModalResult:=mrCancel;
  savedialog.Close;
end;
end.
```