

*М. В. Кононов, А. В Мисник,
С. П. Радченко, О. О. Судаков*

Моделювання фізичних процесів

*Методичні вказівки до проведення
практичних занять*

УДК 518+539.1

Рецензенти:

проф., докт. техн. наук С. Д. Погорілий,
доц., канд. фіз.-мат. наук Ю. В. Бойко

Рекомендовано до друку вченою радою радіофізичного факультету
Київського національного університету імені Тараса Шевченка
(протокол № 2 від 11 жовтня 2004 року).

**М. В. КОНОНОВ, А. В. МИСНИК, С. П. РАДЧЕНКО,
О. О. СУДАКОВ**

Моделювання фізичних процесів: К.: Видавничо-поліграфічний центр
“Київський університет” , 2006. – 92 с.

У виданні наведено загальні підходи до моделювання фізичних явищ, принципи побудови моделей та методи перевірки їх адекватності. Також методичні вказівки містять теоретичні основи для проведення практичних занять з метою засвоєння методів моделювання випадкових процесів, реконструкції томографічних даних, процесів переносу та дифузії в неоднорідних середовищах, штучних нейронних мереж. Зокрема, при побудові завдань використані реальні томографічні дані, отримані на томографічних приладах при дослідженні біологічної тканини. Для моделювання умов перебігу фізичних процесів в неоднорідних середовищах, якими є біологічні тканини, використовуються програмні засоби обробки даних та розрахунку просторових і часових параметрів фізичних процесів, розроблені на кафедрі медичної радіофізики радіофізичного факультету Київського національного університету імені Тараса Шевченка. Частина вказаних програмних засобів пройшла доклінічну перевірку і використовується для медико-біологічних досліджень в науково-лікувальних установах АМН України.

Завдання для практичних робіт також містять рекомендації для студентів при підготовці до них, описи робіт з основною та додатковою літературою.

УДК 518+539.1

Зміст

ЗМІСТ	3
ПРАКТИЧНЕ ЗАНЯТТЯ № 1. ОЦІНКА ПРОДУКТИВНОСТІ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ	5
Задача вибору обчислювальної системи	5
Хід виконання	9
Обладнання	9
Завдання	11
Контрольні запитання	15
ПРАКТИЧНЕ ЗАНЯТТЯ № 2. МОДЕЛЮВАННЯ ВИПАДКОВИХ ВЕЛИЧИН.....	16
Моделювання випадкових величин з різними законами розподілу	16
Завдання:.....	17
ПРАКТИЧНЕ ЗАНЯТТЯ № 3. МОДЕЛЮВАННЯ ВИПАДКОВИХ ПРОЦЕСІВ	19
Детермінований і статистичний опис фізичних процесів.....	19
Метод умовних розподілів	20
Завдання:.....	21
ПРАКТИЧНЕ ЗАНЯТТЯ № 4. МОДЕЛЮВАННЯ ВИПАДКОВИХ ПРОЦЕСІВ ІЗ ЗАДАНИМИ КОРЕЛЯЦІЙНИМИ ВЛАСТИВОСТЯМИ.....	22
Кореляційні властивості випадкового процесу	22
Моделювання нестационарних випадкових процесів із заданими кореляційними властивостями.	22
Метод лінійних перетворень.	23
Метод канонічного розкладу	25
ПРАКТИЧНЕ ЗАНЯТТЯ № 5. МОДЕЛЮВАННЯ ЛАЗЕРНОЇ ТЕРМОДЕСТРУКЦІЇ БІОЛОГІЧНИХ ТКАНИН	26
Задача лазерної термодеструкції	26
Дифузія тепла і перенесення випромінювання	26
Розв'язання рівнянь.....	31
Програма моделювання лазерної термодеструкції.....	35
Завдання.....	36

Контрольні запитання	37
ПРАКТИЧНЕ ЗАНЯТТЯ № 6. МОДЕЛЮВАННЯ СКАНУВАННЯ ТА РЕКОНСТРУКЦІЇ ЗОБРАЖЕНЬ В ПРОЕКТИВНІЙ ТОМОГРАФІЇ	38
Задача розрахунку томографічних проєкцій	38
Комп'ютерні томографічні системи	38
Перенесення рентгенівського випромінювання	41
Системи першого другого та третього поколінь	42
Моделювання томографічного сканування	43
Моделювання реконструкції	45
Моделювання факторів неідеальності	47
Програма моделювання сканування та реконструкції	47
ПРАКТИЧНЕ ЗАНЯТТЯ № 7. МОДЕЛЮВАННЯ МОЛЕКУЛЯРНИХ СТРУКТУР	52
Методи моделювання молекулярних структур	52
Квантово-хімічні методи	52
Метод Хартрі-Фока	55
Теорія збурень Моллера –Пелессета (MP)	59
Теорія функціоналу густини (DFT)	61
Базисні функції	62
Характеристики квантово-хімічних методів	62
Програми квантово-хімічного моделювання	63
Завдання	69
Контрольні запитання	70
ПРАКТИЧНЕ ЗАНЯТТЯ № 8. МОДЕЛЮВАННЯ НЕЙРОННИХ МЕРЕЖ	71
Штучні нейронні мережі	71
Штучні нейронні мережі та методи моделювання поведінки біологічних мереж	73
Алгоритм мережі перцептрона	79
Нейронна мережа на основі багатошарового перцептрону	80
Алгоритм мережі багатошарового перцептрону	80
Методи навчання штучних нейронних мереж	82
ДОДАТКИ	86
ПЕРЕЛІК ПОСИЛАНЬ	89

Практичне заняття № 1.

Оцінка продуктивності обчислювальних систем

Мета: практично ознайомитись з різними типами обчислювальних систем, практично засвоїти методи вимірювання характеристик ефективності обчислювальних систем та отримати навички вибору обчислювальних систем для різних типів задач комп'ютерного моделювання.

Задача вибору обчислювальної системи

Загальні відомості: Сьогодні комп'ютерне моделювання широко застосовується в різних галузях науки та практичної діяльності. Моделювання фізичних процесів не є інструментом лише наукових досліджень — воно широко застосовується в промисловості для розробки нових технологій та зразків продукції; в побутовій техніці та транспортних засобах з комп'ютерним керуванням відбувається моделювання фізичних процесів для прогнозування поведінки системи; загальновідомим прикладом комп'ютерного моделювання фізичних процесів є комп'ютерні ігри. Різні задачі комп'ютерного моделювання вимагають застосування комп'ютерних систем різної потужності і вартості в залежності від складності задачі. Правильний вибір обчислювальної системи для задач комп'ютерного моделювання визначає швидкість, а іноді і успішність вирішення задачі.

Основна мета вибору обчислювальної системи для задач комп'ютерного моделювання фізичних процесів — це зменшення часу обрахунку. Крім часу обрахунку, також слід враховувати такі аспекти, як час розробки програмного забезпечення, вартість та доступність обчислювальної системи, вартість та доступність програмного забезпечення та ін..

Оцінка об'єму оперативної пам'яті

Кількість обчислювальних операцій алгоритму залежить від об'єму даних задачі. Обчислювальна система може ефективно працювати, тільки якщо структури даних задачі повністю вміщуються в оперативній пам'яті обчислювальної системи. Тому при виборі обчислювальної системи необхідно визначати необхідний об'єм оперативної пам'яті. Об'єм оперативної пам'яті V можна оцінити за

формулою:

$$V = \sum_k n_k d_k$$

де індекс k — нумерує структури даних задачі, що одночасно мають знаходитись в оперативній пам'яті (наприклад, масиви, матриці); n_k — кількість елементів у k — й структурі даних; d_k — об'єм пам'яті елементу k — ї структури даних, який залежить від точності представлення даних. Основні типи даних та об'єм пам'яті для них наведено в табл. 1.1.

Таблиця 1.1 – Характеристики деяких типів даних

№	Тип даних	Об'єм пам'яті, d_k , байт	Відносна швидкість операцій, r
1	Символ (character, char, byte)	1	1
2	Ціле число (integer, int)	2, 4, 8 (в залежності від розрядності системи)	1
3	Коротке ціле число (short)	1, 2, 4 (в залежності від розрядності системи)	1
4	3 плаваючою точкою (real, float)	8	0.25
5	3 плаваючою точкою подвійної точності (double, double)	16	0.5
6	Комплексні числа (complex)	16	0.5
7	Комплексні числа подвійної точності (doublecomplex)	32	0.125

Оцінка часу обрахунку

Час обрахунку задачі залежить від алгоритму її розв'язання, об'єму вхідних даних, продуктивності та типу обчислювальної системи, надійності апаратури та програмного забезпечення. Для простих задач моделювання характерний час обрахунку може сягати від декількох секунд (обрахунок нескладних інтегралів) до декількох днів (розв'язання систем рівнянь в частинних похідних, інтегральні рівняння, зворотні задачі). Для найскладніших задач час обрахунку може сягати тижнів, місяців, років (квантова хімія, ядерна фізика, прогнозування погоди, моделювання міцності конструкцій та ін.).

Час обрахунку t можна просто оцінити, використовуючи формулу:

$$t = \sum_i \frac{L_i r_i}{S_i} \quad (1.1)$$

де індекс i — нумерує різні частини задачі, L_i — кількість операцій алгоритму, які обчислювальна система має виконати для того, щоб розв'язати i — ту частину задачі, S_i — продуктивність обчислювальної системи для i — ї частини задачі; r_i — відносна швидкість виконання операцій з даними задачі. Формула (1.1) передбачає, що всі операції алгоритму виконуються **послідовно**.

В наші дні широко застосовуються **паралельні** обчислювальні системи і паралельні алгоритми [1], які дозволяють обраховувати різні частини задачі за допомогою різних обчислювальних систем (процесорів), що працюють одночасно (паралельно). Час t_p обрахунку задачі за допомогою паралельної системи, яка має p процесорів, можна оцінити за формулою:

$$t_p = k_e \frac{t}{p}$$

де k_e — коефіцієнт ефективності, який для більшості задач лежить в діапазоні $0 \leq k_e \leq 1$ і визначає ефективність розпаралелювання алгоритму та "накладні витрати", що вносить паралельна обчислювальна система за рахунок необхідності обміну між процесорами. Не всі алгоритми ефективно розпаралелюються.

Ефективність алгоритмів

Кожний алгоритм, що використовується для розв'язання задачі, оперує з певними даними, кількість яких описується за допомогою характерної розмірності задачі N , яка може визначати розмірність матриці, довжину масиву, та ін.. Ефективність алгоритму характеризується кількістю операцій, яку потрібно виконати для розв'язання задачі заданої розмірності, крім того, алгоритми характеризуються об'ємом необхідної пам'яті та стійкістю. Характеристики ефективності деяких алгоритмів наведено в табл. 1.2.

При виборі алгоритму першою вимогою повинна бути стійкість, а лише після цього — швидкість. Для багатьох задач доступні алгоритми, які добре себе зарекомендували на практиці [2].

Таблиця 1.2 – Властивості деяких алгоритмів для характерної розмірності задачі.

№	Алгоритм	Об'єм пам'яті, ν	Число операцій, L	Стійкість	Характеристика
1	Сортування	$\sim N$	$\sim \ln N$	Висока	Швидкий
2	Пошук	$\sim N$	$\sim \ln N \div N$	Висока	Швидкий
3	Швидке перетворення Фур'є	$\sim N$	$\sim N \ln N$	Висока	Швидкий
4	Розв'язок симетричної системи лінійних рівнянь	$\sim N^2$	$\sim N^2$	Низька/Середня	Швидкий
5	Розв'язок системи лінійних рівнянь	$\sim N^2$	$\sim N^3$	Середня/висока	Повільний
6	Розв'язок нелінійних рівнянь	$> N^3$	$> N^3$	Низька	Дуже повільний
7	Ітеративні, α — кількість ітерацій	Залежить від алгоритму	$\sim L \cdot \alpha$	Висока/Низька	Повільний/Дуже повільний

Характеристики обчислювальних систем

Обчислювальні системи характеризуються кількістю процесорів (CPU), продуктивністю процесорів, об'ємом пам'яті, швидкістю каналів передачі (для багатопроцесорних систем), а також надійністю та вартістю. Продуктивність процесора вимірюється в кількості машинних інструкцій за секунду — **MIPS** (Million Instructions Per Second). Швидкість виконання операцій залежить від типу даних (табл. 1.1). Для задач фізичного моделювання найважливішою характеристикою є продуктивність операцій з плаваючою крапкою, яка вимірюється в одиницях **Flops** (floating point operations per second) — кількість операцій з плаваючою крапкою в секунду. На практиці застосовуються кратні одиниці **MFlops**= 10^6 Flops, **GFlops**= 10^9 Flops. У табл. 1.3 наведено характеристики деяких типів обчислювальних систем.

Для задач моделювання варто обирати систему мінімальної вартості, яка забезпечує необхідний об'єм пам'яті та необхідну швидкість обчислень. Крім того, слід враховувати, що багатопроцесорні системи мають меншу надійність.

Характеристики програмного забезпечення

Програмне забезпечення для задач моделювання можна розділити на програми користувачів, які написані користувачами за досить короткий час і розраховані на короткочасне використання, та

програмні пакети, які розробляються тривалий час, мають багато функцій і високу вартість. При виборі програмного забезпечення варто враховувати необхідні затрати часу на розробку програмного забезпечення та вартість існуючих програмних пакетів.

Таблиця 1.3 – Характеристики обчислювальних систем.

№	Тип системи	Платформа	Пікова продуктивність	Об'єм пам'яті	Вартість
1	Однопроцесорна робоча станція	X86(Intel, AMD, Alpha, Sparc)	~1 GFlops	><~1GByte	низька
2	SMP робоча станція	X86 (Intel, AMD, Alpha, Sparc) 2-4 CPU Intel, AMD 32-64 CPU Alpha	~1 Gflops /CPU	><~1GByte	середня
3	Векторно-конвеєрний суперкомп'ютер	Cray T90 16-32 CPU	~2GFlop /CPU	> 1 GByte	дуже висока
4	Масивно-паралельний суперкомп'ютер	Cray T3E до 2048 CPU	~1GFlops /CPU	>1 Gbytes	дуже висока
5	Високопродуктивна NUMA система	SGI Origin 3000 (тисячі CPU)	>1 TFlops	<~1TByte	дуже висока
6	Обчислювальний кластер	Набір однопроцесорних, SMP та NUMA систем, >1000 CPU	>50TFlops	>10 TBytes	Сума вартостей компонент

Хід виконання

Обладнання

Обладнання для виконання практичного завдання складається з робочих станцій з інстальованою операційною системою Linux, програмне забезпечення для підтримки кластерних обчислень openMosix, середовище MPI, тестові програми бібліотеки ATLAS, тестове програмне забезпечення для вимірювання продуктивності паралельних обчислювальних систем High Performance Computing Linpack Benchmark (HPL). Схема взаємодій між різними компонентами обладнання показана на рис. 1.1.

Різні компоненти схеми мають наступне призначення.

1. **Термінальний сервер**, що працює під операційною системою Linux.

Термінальний сервер – це програмне забезпечення, яке дозволяє створювати бездисккові робочі станції за рахунок підтримки доступу до спільних ресурсів через локальну мережу.

2. **Термінальні клієнти**, що працюють під операційною системою Linux.

Термінальний клієнт – це бездисктова робоча станція, що використовує термінальний сервер для доступу до дискових ресурсів через локальну мережу. В даній роботі в якості термінальних клієнтів використовуються робочі станції лабораторії, які завантажуються через мережу і працюють в бездисктовому варіанті. В даній роботі використовується поставка cluster-knoppix [3]



Рис. 1.1 – Схема зв'язків різних компонент апаратного та програмного забезпечення, що використовується для виконання практичного завдання.

3. Програмне забезпечення для підтримки кластерних обчислень openMosix.

Кластер – це декілька обчислювальних машин (вузлів), що працюють спільно для виконання деякого завдання. OpenMosix [4] – це програмне забезпечення (розширення ядра операційної системи Linux), яке дозволяє створювати обчислювальний кластер з функціями міграції процесів між вузлами кластера для динамічного балансування завантаження вузлів.

4. Середовище MPI (Lam-Mpi).

MPI (Message Passing Interface) — це бібліотеки, які реалізують стандарт обміну повідомленнями між процесами паралельної програми. *Lam-Mpi* — це одна з реалізацій стандарту MPI – Local Area Machine [5].

5. Тестові програми бібліотеки ATLAS.

ATLAS [6] — (Auto Tuning Linear Algebra Software) — оптимізоване програмне забезпечення для виконання розрахунків з лінійної алгебри та набір програм для тестування продуктивності.

6. Тестове програмне забезпечення для вимірювання продуктивності паралельних обчислювальних систем High Performance Computing Linpack Benchmark (HPL).

HPL — це програмне забезпечення, яке використовується в top500 найшвидших суперкомп'ютерів для тестування продуктивності [7].

Завдання

1. Створення середовища для роботи (потрібне тільки один раз на початку роботи або при збоях в роботі)

1.1. Завантаження робочих станцій.

У викладача отримайте дискету з завантажувачем і перезавантажте робочу станцію з дискети. У процесі завантаження робоча станція отримує IP адресу з використанням протоколу DHCP. Після цього з термінал-сервера через протокол TFTP завантажуються ядро та початковий образ системи. Після завантаження ядра з термінал-сервера підключається компакт-диск з образом системи за допомогою протоколу NFS. З компакт-диска завантажуються і запускаються всі необхідні системні служби, після чого термінальний клієнт готовий до роботи. Час завантаження може тривати близько 1 хвилини (в залежності від завантаженості мережі та її швидкості). Ознакою закінчення завантаження є запрошення для роботи в командному рядку у вигляді `root@knoppix #`. У випадку помилок завантаження зверніться до викладача.

1.2. Підключення домашніх каталогів.

Все необхідне для виконання роботи програмне забезпечення міститься на NFS сервері у домашньому каталозі. Для підключення домашнього каталогу після завантаження клієнта потрібно виконати команди:

```
mount <ім'я сервера>:/home/knoppix /home/knoppix <ENTER>  
mount <ім'я сервера>:/usr /home/knoppix/usr <ENTER>
```

Наприклад:

```
mount 10.0.11.2:/home/knoppix /home/knoppix <ENTER>  
mount 10.0.11.2:/usr /home/knoppix/usr <ENTER>
```

За допомогою команд

```
su knoppix <ENTER>  
cd <ENTER>
```

ви можете зайти до домашнього каталогу, де необхідно створити власну робочу директорію за допомогою команд

```
mkdir <ім'я директорії> <ENTER>  
cd <ім'я директорії> <ENTER>
```

1.3. Створення середовища для роботи MPI.

Перевірте наявність всіх робочих станцій у кластері за допомогою команди `mosmon`, яка видає номери станцій в системі openMosix. За допомогою команди

```
mosctl whois <openMosix номер станції>
```

ви можете визначити її IP адресу. Для запуску віртуальної машини MPI вам потрібен список IP адрес всіх машин у файлі `lamhosts`, який потрібно створити вручну. Приклад файлу

```
10.0.11.249  
10.0.11.247  
10.0.11.237  
10.0.11.236
```

Адреса вашого вузла повинна бути першою в списку. Для запуску

віртуальної машини MPI використовуйте команду

```
lamboot ~/lamhosts.
```

У випадку помилок зверніться до викладача.

2. Ознайомлення з послідовними та паралельними обчислювальними системами.

Файлова система

```
/proc
```

на кожному вузлі містить інформацію про апаратне забезпечення вашої станції. Визначте фізичні параметри вашої робочої станції: тип та швидкість процесора і об'єм оперативної пам'яті.

Каталог

```
/proc/hpc
```

на кожному вузлі містить інформацію про вузли кластера: номер вузла, кількість вузлів у системі та їх параметри. За допомогою цієї інформації визначте параметри кластера.

3. Тестування продуктивності однопроцесорної робочої станції для різних розмірів матриці даних та різних задач.

Для роботи використовуються програми

```
xclutst,  
xdlutst,  
xslutst,  
xzlutst,
```

які реалізують алгоритм розв'язання системи лінійних рівнянь методом LU факторизації (реалізація методу виключення Гауса) і дозволяють вимірювати час обрахунку. Ці програми є частиною бібліотеки ATLAS. Літери *s*, *d*, *c*, *z* в назвах виконуваних файлів означають відповідно `single`, `double`, `complex`, `double complex` (одинарна, подвійна точність, комплексні числа, комплексні числа з подвійною точністю). Кожна програма має набір аргументів, які дозволяють змінювати розмір матриці даних. Список аргументів

можна подивитись за допомогою опції -?. Важливою є опція

```
-N <мінімальний розмір> <максимальний розмір>  
<крок зміни розмірів> ,
```

яка дозволяє задавати порядки квадратних матриць, для яких буде виконуватись задача. Перед запуском програм потрібно заблокувати міграцію процесів з вашої робочої станції шляхом виконання команди

```
echo "1" > /proc/$$/lock.
```

Для задач різної точності виміряти продуктивність обчислювальної станції (в mflops) при 10 різних розмірах матриці даних в межах від 100x100 до 3000x3000.

Знайдіть максимальне значення обчислювальної ефективності вашої робочої станції.

6. Тестування продуктивності розподіленої обчислювальної системи для різних розмірів матриці даних, різної кількості вузлів, різних значень параметрів та різного розподілу процесів між вузлами.

Для виконання використовується програма

```
xhpl
```

Знайти її можна в каталозі

```
/home/knoppix/hpl/bin/Linux_PII_FBLAS.
```

Там же знаходиться вхідний файл **hpl.dat**, в якому можна задавати параметри програми. Програма реалізує паралельний алгоритм розв'язання системи лінійних рівнянь методом LU факторизації (варіант методу виключення Гауса) подвійної точності і дозволяє вимірювати час обрахунку. Для виконання вхідний файл та програму потрібно скопіювати в робочий каталог. Програма працює з використанням стандарту MPI, тому перед запуском потрібно запустити середовище MPI, як вказано в п. 1. Запуск програми відбувається наступним чином:

```
mpirun -np <кількість процесів> -c2c ./xhpl.
```

Виміряйте продуктивність паралельної обчислювальної системи для розв'язання вказаної задачі при розмірах матриці даних, що відповідали розмірам матриці при виконанні п. 3. Кількість процесів рівна — 4.

Знайдіть максимальне значення обчислювальної ефективності паралельної системи і порівняйте отримане значення з відповідним значенням для однопроцесорної системи.

Змініть кількість процесів від 2 до 8 та знайдіть максимальне значення обчислювальної ефективності, порівняйте результати з попереднім випадком.

Зробіть висновки про те, у яких випадках ефективнішою є однопроцесорна система, багатопроцесорна система, а також про вплив кількості процесів, кількості вузлів, розподілу процесів між вузлами на ефективність обрахунку задачі. Яке може бути пояснення для цих висновків?

Контрольні запитання

Звіт про виконання практичного заняття приймається в електронному вигляді, як відповіді на наступні контрольні запитання.

1. Які параметри вашої робочої станції?
2. Оцініть кількість операцій, об'єм пам'яті та час обрахунку, які потрібні для вирішення послідовної задачі в даній роботі.
3. Для яких задач ефективність послідовного обрахунку виходить найбільшою? Який характерний час обрахунку задачі за допомогою послідовного алгоритму? Який об'єм оперативної пам'яті робочої станції використовують програми при різних розмірах матриці даних?
4. Чому рівна максимальна виміряна продуктивність вашої робочої станції? При яких умовах і на яких задачах це значення досягається, чому?
5. Яка продуктивність паралельної обчислювальної системи, яке її максимальне значення, при яких параметрах воно досягається і чому?
6. У яких випадках послідовна система є ефективнішою за паралельну, а коли навпаки?

Практичне заняття № 2.

Моделювання випадкових величин

Мета: засвоїти методи моделювання випадкових величин, провести кореляційну обробку та отримати оцінки якості згенерованої послідовності випадкових величин.

Моделювання випадкових величин з різними законами розподілу

Загальні відомості: Моделювання випадкових величин з різними законами розподілу ґрунтується на теоремі:

Якщо випадкова величина x має функцію густини розподілу $p(x)$, то розподіл випадкової величини $y = F(x)$ буде рівномірним на інтервалі $[0, 1]$.

$F(x)$ - функція розподілу випадкової величини x , яка за означенням $F(a) = P(x < a) = \int_{-\infty}^a p(x) dx$. Тобто моделювання

випадкової величини з заданим законом розподілу зводиться до аналітичних, графічних, числових методів розв'язання інтегрального

рівняння $\int_{-\infty}^{x_i} p(x) dx = y_i$.

Випадкова величина ξ рівномірно розподілена на відрізку $[a, b]$,

має функцію густини розподілу $p(\xi) = \begin{cases} \frac{1}{b-a}, & a \leq \xi \leq b \\ 0, & \xi < a \text{ і } b < \xi \end{cases}$. Її

математичне сподівання $M_\xi = \frac{b+a}{2}$ і дисперсія $D_\xi = \frac{1}{12}$.

Випадкова величина ξ , яка рівномірно розподілена на відрізку $[0, 1]$, має відповідно функцію густини розподілу

$$p(\xi) = \begin{cases} 1, & 0 \leq \xi \leq 1 \\ 0, & \xi < 0 \text{ і } 1 < \xi \end{cases}.$$

її математичне сподівання $M_\xi = \frac{1}{2}$ і дисперсія $D_\xi = \frac{1}{12}$.

Метод рекурентних співвідношень отримання випадкової величини ξ , яка рівномірно розподілена на відрізку $[0, 1]$, полягає в генерації послідовності чисел за $\xi_i = \frac{B_i}{M}$, де $B_{i+1} = (KB_i + C) \pmod{M}$, де операція \pmod{M} - залишок від ділення $\frac{KB_i + C}{M}$. Строго кажучи, отримана послідовність є псевдовипадковою, а розподіл – квазірівномірним. Ступінь наближення згенерованої послідовності чисел до випадкової величини, яка рівномірно розподілена на відрізку $[0, 1]$, визначається вибором початкових значень параметрів. Для їх вибору використовують такі правила:

- K – просте число;
- B_0 – будь-яке достатньо велике число;
- M – достатньо велике парне число, але не кратне K ;
- C – будь-яке число близьке до B_0 .

Для оцінки якості згенерованої послідовності випадкових чисел використовують статистичні критерії:

- близькості математичного сподівання і дисперсії теоретичного та експериментального розподілів;
- близькості теоретичного та експериментального розподілів за критерієм χ^2 ;
- наявності невідповідної ділянки на основі значення нормованої автокореляційної функції.

Завдання:

1. Створити генератор послідовності випадкових величини ξ , рівномірно розподіленої на відрізку $[0, 1]$, який реалізує метод рекурентних перетворень.

2. Підібрати початкові параметри для надійної роботи генератора (завдання 1).
3. Дослідити якість згенерованої послідовності випадкових величин:
 - a. наявність не випадкової ділянки на основі значення нормованої автокореляційної функції ($|R_i| < 0.1$ з рівнем значимості 0.95).
 - b. близькість математичного сподівання і дисперсії теоретичного та експериментального розподілів з рівнем значимості 0.95;
4. Отримати послідовність із 1000 випадкових чисел, які мають:
 - a. нормальний розподіл з параметрами (a, σ) , $a = 1$, $\sigma = 1$ (додаток 1);
 - b. розподіл Сімпсона на відрізку $[a, b]$, $a = 1$, $b = 2$ (додаток 2);
 - c. експоненціальний розподіл з параметром α , $\alpha = 2$ (додаток 3).

Практичне заняття № 3.

Моделювання випадкових процесів

Мета: засвоїти методи моделювання випадкових процесів, провести кореляційну обробку та отримати оцінки якості згенерованої послідовності.

Детермінований і статистичний опис фізичних процесів

Загальні відомості: Розгляд будь-якого фізичного процесу оснований на його математичному описі, який може бути детермінованим або статистичним (імовірнісним).

При детермінованому описі у нас є всі дані для точного вказування часової залежності параметрів процесу $y = f(t)$. В основі такого підходу лежить уявлення, що повторення експерименту призведе до тих самих результатів.

Класичним прикладом невідповідності такого підходу є рух броунівської частинки (наприклад: зміна її координати x з часом $x = f(t)$). Для опису цього процесу використовують функцію розподілу імовірності $\omega(x, t)$, яка визначає імовірність значень x в

момент часу t . $P(x_1 \leq x < x_2, t) = \int_{x_1}^{x_2} \omega(x, t) dx$, очевидно, що

$\int_{-\infty}^{\infty} \omega(x, t) dx = 1$. Процес може бути і комплексним $z(t) = x(t) + iy(t)$.

В загальному випадку випадковий процес $x(t)$ характеризує його багатовимірна функція густини імовірності $p(x_1, x_2, x_3, \dots, x_n, t_1, t_2, t_3, \dots, t_n)$. В ідеальному випадку для повного опису процесу $x(t)$ $n \rightarrow \infty$. Чим більше значення n , тим детальніший опис процесу $x(t)$, але ж n також визначає і кількість відліків процесу при моделюванні (розмір необхідних ресурсів, час моделювання).

Розглянемо кілька реалізацій випадкового процесу за період часу $(0, T_M)$. В межах цього проміжку одна реалізація може бути подана як набір із m випадкових величин x_j^i з відповідними розподілами, де $t_j \in (0, T_M)$, $j = 1 \div m$.

Якщо $\Delta t = t_{i+1} - t_i = \text{const}$, то можна записати $p(x_1, x_2, x_3, \dots, x_m, t_1, t_2, t_3, \dots, t_m)$ як $p(x_1, x_2, x_3, \dots, x_m)$, із виконанням всіх умов нормування

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(x_1, x_2, x_3, \dots, x_m) dx_1 dx_2 dx_3 \dots dx_m = 1.$$

Функція розподілу густини імовірності $p(x_1, x_2, x_3, \dots, x_m)$ отримується із $p(x_1, x_2, x_3, \dots, x_n, t_1, t_2, t_3, \dots, t_n)$

$$p(x_1, x_2, x_3, \dots, x_m) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(x_1, x_2, x_3, \dots, x_n) dx_{m+1} dx_{m+2} dx_{m+3} \dots dx_n$$

, $m < n$

Умовна густина імовірності запишеться як

$$p(x_1, x_2, x_3, \dots, x_i / x_{i+1}, x_{i+2}, x_{i+3}, \dots, x_m) = \frac{p(x_1, x_2, x_3, \dots, x_m)}{p(x_{i+1}, x_{i+2}, x_{i+3}, \dots, x_m)}.$$

Це загальний підхід до моделювання випадкових процесів. Він дозволяє змодельовати довільний випадковий процес, але є дуже повільним і громіздким.

Метод умовних розподілів

Схема методу умовних розподілів походить із очевидної залежності

$$p(x_1, x_2, x_3, \dots, x_n) = p_1(x_1) p_2(x_1 / x_2) p_3(x_3 / x_1, x_2) \dots p_n(x_n / x_1, x_2, x_3, \dots, x_{n-1})$$

. Суть методу полягає у визначенні одновимірних умовних густин імовірності для кожного відліку, тобто, кожне значення x_i (кожен відлік випадкового процесу) генерується датчиком випадкових чисел із своїм розподілом:

$$- p_1(x_1) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(x_2, x_3, \dots, x_n) dx_2 dx_3 \dots dx_n ;$$

$$- p_2(x_2 / x_1) = \frac{p_{12}(x_1, x_2)}{p_1(x_1)}, \text{ де}$$

$$p_{12}(x_1, x_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(x_3, x_4, x_5, \dots, x_n) dx_3 dx_4 dx_5 \dots dx_n ;$$

$$- p_3(x_3 / x_1, x_2) = \frac{p_{123}(x_1, x_2, x_3)}{p_{12}(x_1, x_2)} ;$$

- ...;

$$- p_k(x_k / x_1, x_2, x_3, \dots, x_{k-2}, x_{k-1}) = \frac{p_{123\dots k}(x_1, x_2, x_3, \dots, x_k)}{p_{123\dots k-1}(x_1, x_2, x_3, \dots, x_{k-1})},$$

де

$$p_{123\dots k}(x_1, x_2, x_3, \dots, x_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(x_1, x_2, x_3, \dots, x_n) dx_{k+1} dx_{k+2} dx_{k+3} \dots dx_n$$

Маючи густини імовірності $p_1, p_2, p_3, \dots, p_k$, за допомогою датчика випадкових чисел генеруються відліки $x_1, x_2, x_3, \dots, x_k$.

Завдання:

1. Побудувати модель випадкового процесу, який має поліноміальний розподіл з параметрами $n, a_1, a_2, a_3, \dots, a_n$ (додаток 4).
2. Підібрати початкові параметри $n, a_1, a_2, a_3, \dots, a_n$ для моделювання випадкового процесу з $n = 4$ (завдання 1).
3. Використати генератор для побудови текстури (додаткове завдання).

Практичне заняття № 4.

Моделювання випадкових процесів із заданими кореляційними властивостями

Мета: засвоїти методи моделювання випадкових процесів із заданими кореляційними властивостями.

Кореляційні властивості випадкового процесу

Загальні відомості: Реальним випадкам моделювання випадкових процесів частіше відповідає ситуація, коли відомі кореляційні властивості процесу при невідомих його статистичних характеристиках.

Така ситуація може скластися з багатьох причин, зокрема:

- багато фізичних процесів моделюються нормальним (гаусовим) випадковим процесом, для опису якого достатньо знати лише кореляційну функцію;
- часто для задач радіофізики та радіотехнічних застосувань необхідно змодельовати випадковий процес, який можна легко (без значних часових ресурсів і додаткової підготовки) отримати саме із нормального випадкового процесу (див. поперед. пункт);
- при теоретичних і експериментальних дослідженнях випадкових процесів часто трапляються випадки, коли вимірююю або теоретично описаною є лише його кореляційна функція;
- при моделюванні трапляються випадки, коли кінцевий результат залежить лише від кореляційних властивостей випадкового процесу (наприклад: при дії випадкового процесу на лінійний інерційний пристрій на його виході отримують нормальний випадковий процес, характеристики якого визначаються виключно кореляційними властивостями вхідного випадкового процесу).

Моделювання нестационарних випадкових процесів із заданими кореляційними властивостями.

Для нестационарного випадкового процесу $x(t)$ кореляційна функція залежить від двох параметрів t_1 і t_2 .

$$R(t_1, t_2) = M(x(t_1)x(t_2)) = M(x_1 x_2) \quad (4.1)$$

Для зручності моделювання вважають, що процес має нульове математичне сподівання $M(x(t)) = 0$. При необхідності моделювання процесу $y(t)$, який має відмінне від нуля математичне сподівання $M(y(t)) = m_y(t)$, використовують перетворення

$$y(t) = x(t) + m_y(t) \quad (4.2)$$

Моделювання випадкового процесу здійснюється для дискретних моментів часу t_i ($i = 1 \div n$), тобто, необхідно змоделювати випадковий вектор X , який містить n складових (див. попереднє заняття). При цьому кореляційні властивості будуть задаватися кореляційною матрицею R , елементи якої визначаються із кореляційної функції $R_{ij} = R(t_i, t_j)$.

Метод лінійних перетворень.

Метод застосовують для моделювання відліків випадкового процесу з відомими значення кореляційної функції в певні проміжки часу (дискретність має довільний крок). Суть методу полягає в лінійному перетворенні $\vec{Y} = A\vec{X}$ випадкового вектору \vec{X} з некорельованими складовими x_i у випадковий вектор \vec{Y} з корельованими складовими y_i ($i = 1 \div n$).

$$\vec{X} = \{x_1, x_2, x_3, \dots, x_n\}, \quad \vec{Y} = A\vec{X}, \quad A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{Bmatrix},$$

$$\vec{Y} = \{y_1, y_2, y_3, \dots, y_n\}^T.$$

Таким чином, необхідно визначити елементи матриці перетворення A через задані елементи кореляційної матриці R .

Нехай реалізований датчик випадкових чисел забезпечує послідовність випадкових величин з математичним сподіванням $M(x_i) = 0$ і дисперсією $D(x_i) = 1$ ($i = 1 \div n$).

$$M(x_i x_j) = 0, \quad i \neq j \quad (4.3)$$

Додатково накладається умова незалежності випадкових чисел. Це забезпечує виконання умови некорельованості складових вектора \vec{X} (4.3).

Якщо подати матрицю A у трикутному вигляді

$$A = \begin{Bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} \end{Bmatrix}, \quad \text{то знаходження залежності}$$

елементів a_{ij} матриці від елементів кореляційної матриці R стає простим і однозначним, оскільки тепер:

$$\begin{aligned} y_1 &= a_{11}x_1 \\ y_2 &= a_{21}x_1 + a_{22}x_2 \\ y_3 &= a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \\ &\dots \\ y_n &= a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n \end{aligned} \quad (4.4)$$

і, враховуючи (4.1), з (4.4) отримаємо

$$\begin{aligned} R_{11} &= M(y_1^2) = M(a_{11}^2 x_1^2) = a_{11}^2 M(x_1^2) = a_{11}^2 D(x) = a_{11}^2 \\ R_{12} &= M(y_1 y_2) = M(a_{11} x_1 (a_{21} x_1 + a_{22} x_2)) = a_{11} M(a_{21} x_1 x_1 + a_{22} x_1 x_2) = \\ &= a_{11} a_{21} M(x_1^2) + a_{11} a_{22} M(x_1 x_2) = a_{11} a_{21} \\ R_{22} &= M(y_2^2) = M((a_{21} x_1 + a_{22} x_2)(a_{21} x_1 + a_{22} x_2)) = a_{21}^2 + a_{22}^2 \\ &\dots \end{aligned} \quad (4.5)$$

З рівнянь (4.5) можна отримати значення коефіцієнтів a_{ij} , які є елементами матриці перетворення A .

$$\begin{aligned} a_{11} &= \sqrt{R_{11}} \\ a_{i1} &= \frac{R_{i1}}{a_{11}}, \quad i = 2 \div n \end{aligned}$$

$$a_{ij} = \frac{R_{ij} - \sum_{k=1}^{j-1} a_{ik} a_{jk}}{a_{jj}}, \quad j = 2 \div i - 1$$

$$a_{ii} = \sqrt{R_{ii} - \sum_{k=1}^{i-1} a_{ki}^2}$$

Метод канонічного розкладу

Випадковий процес $y(t)$ можна подати у вигляді ряду

$$y(t) = \sum_{i=1}^{\infty} u_i \omega_i(t), \quad (4.6)$$

де u_i – коефіцієнти розкладу, $\omega_i(t)$ утворюють систему функцій. Відповідно, коефіцієнти розкладу в ряді (4.6) є випадковими величинами, які змінюються для кожної реалізації. Вони мають задовільняти умові некорельованості

$$M(u_i u_j) = 0, \quad \text{при } i \neq j. \quad (4.7)$$

Також для зручності покладемо, що $M(u_i) = 0$ і $M(y(t)) = 0$. При моделюванні процесу з ненульовим середнім необхідно скористатися (4.2).

Умову (4.7) можна задовільнити лише при виборі в якості системи функцій у ряді (4.6) лише $\omega_i(t)$, які задовільняють рівняння

$$\int_0^{T_M} R(t_1, t_2) \omega_i(t_2) dt_2 = k_i \omega_i(t_1) \quad (4.8)$$

Завдання:

1. Проаналізувати отриману кореляційну функцію та вимоги до моделювання випадкового процесу.
2. Визначити метод моделювання випадкового процесу за заданими кореляційними властивостями.
3. Провести розрахунок матриці перетворення та функцій в залежності від вибраного методу.
4. Змоделювати випадковий процес із заданими кореляційними властивостями.
5. Оцінити якість змодельованого процесу.

Практичне заняття № 5.

Моделювання лазерної термодеструкції біологічних тканин

Мета: практично ознайомитись з основами комп'ютерного моделювання фізичних процесів переносу випромінювання та дифузії тепла в середовищі на прикладі біологічних тканин.

Задача лазерної термодеструкції

Загальні відомості: В сучасних медичних технологіях широко застосовується моделювання фізичних процесів, які відбуваються в біологічних тканинах. Прикладом такої задачі є моделювання фізичних процесів, що відбуваються в біологічних тканинах при дії на них лазерного випромінювання. Під дією лазерного випромінювання тканина нагрівається і гине. Цей процес називається лазерною термодеструкцією [8, 9]. На кафедрі медичної радіофізики радіофізичного факультету Київського національного університету було розроблено технологію, яка дозволяє виконувати хірургічні операції на головному мозку з використанням лазерної термодеструкції [10, 11]. При цьому в точку операції за допомогою мікро-трепанції вводиться світловод, через який подається лазерне випромінювання, що руйнує патологічне утворення. На відміну від випадку механічного руйнування тканин лазерна термодеструкція є малоінвазивним методом. Однак, для успішності операції потрібно розрахувати параметри лазерного випромінювання, яке подається в зону операції і дозволяє отримати область ураження потрібного розміру без пошкодження нормальної тканини. Параметри лазерного випромінювання можна знайти шляхом моделювання процесів поширення випромінювання і тепла в біологічних тканинах при дії випромінювання різної інтенсивності і знаходження зони, в якій температура буде вищою 43°C [12], що відповідає температурі коагуляції білка.

Дифузія тепла і перенесення випромінювання

Геометрію задачі лазерної термодеструкції біологічних тканин показано на рис. 5.2.

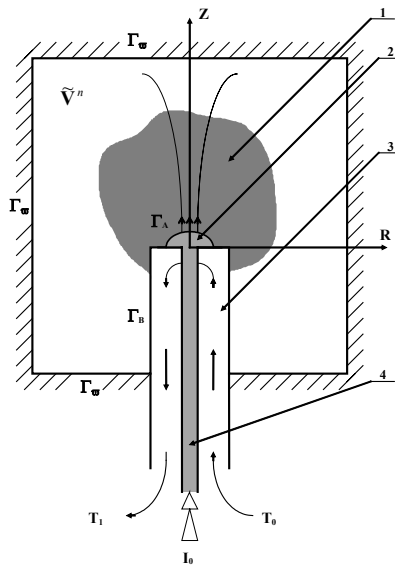


Рисунок 5.2 – Схема обчислення теплових полів при лазерній термотерапії: 1 — пухлина, 2 — оптична насадка, 3 — аплікатор, 4 — світловод, T_0 — рідина для охолодження, I_0 — лазерне випромінювання.

В область пухлини вводиться голка аплікатора — хірургічного інструмента, в якому проходить світловод для доставки лазерного випромінювання інтенсивності I_0 . На кінці світловоду знаходиться розсіювач світла чи інший оптичний прилад, що формує діаграму випромінювання. Між стінками аплікатора циркулює рідина для охолодження, яка має температуру T_0 . Область, в якій потрібно знайти розподіл температур, позначимо \tilde{V}^n . Межа цієї області $\Gamma = \partial\tilde{V}^n$ складається з віддаленої границі Γ_∞ , де можна знехтувати впливом випромінювання, і поверхні аплікатора $\Gamma_A + \Gamma_B$. Вісь аплікатора направлено вздовж осі Z , а радіус циліндричної системи координат R — перпендикулярно осі Z .

Поширення тепла в тканинах, що насичені кровоносними судинами, описується квазілінійним рівнянням теплопровідності з гладкою залежністю коефіцієнтів від температури.

$$C_V \frac{\partial T}{\partial t} = \vec{\nabla}(\lambda \vec{\nabla} T) - B_V(T - T_0) + F, \quad (5.3)$$

де $T(\vec{r}, t)$ — температура в точці \vec{r} ; $C_V(\vec{r}, T)$ — об'ємна теплоємність; $\lambda(\vec{r}, T)$ — коефіцієнт теплопровідності; $B_V(\vec{r}, T)$ — коефіцієнт об'ємного тепловідводу за рахунок кровотоку; T_0 — температура крові, які рівна температурі рідини охолодження; $F(\vec{r}, t)$ — функція джерела нагріву.

Граничні умови на поверхні аплікатора у випадку примусового охолодження:

$$\lambda \frac{\partial T}{\partial n} \Big|_{\Gamma} = B_{\Gamma}(T - T_0) \Big|_{\Gamma}, \Gamma = \Gamma_A, \Gamma_B, \quad (5.4)$$

де B_{Γ} — коефіцієнт теплопередачі через стінку аплікатора.

На нескінченності температура рівна температурі крові:

$$T \Big|_{\Gamma} = T_0 \Big|_{\Gamma}, \Gamma = \Gamma_{\infty}, \quad (5.5)$$

Початкові умови в області \tilde{V}^n визначаються температурою крові:

$$T(\vec{r}, t = 0) = T_0, \vec{r} \in \tilde{V}^n \quad (5.6)$$

Враховуючи циліндричну симетрію, тривимірну задачу можна звести до двовимірної

$$C_V \frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left[r \lambda \frac{\partial T}{\partial r} \right] + \frac{\partial}{\partial z} \left[\lambda \frac{\partial T}{\partial z} \right] - B_V(T - T_0) + F \quad (5.7)$$

Значення функції джерела можна виміряти на експерименті або обрахувати чисельно за формулою:

$$F(\vec{r}) = k_a c U(\vec{r}) \quad (5.8)$$

де k_a — коефіцієнт поглинання випромінювання, c — швидкість світла, $U(\vec{r})$ — густина енергії випромінювання в точці \vec{r} .

Густину випромінювання можна знайти з розв'язку рівняння переносу випромінювання в мутному (з високим коефіцієнтом розсіювання) середовищі, яким є біологічні тканини.

$$\frac{1}{c} \frac{\partial I}{\partial t} = -\vec{n} \vec{\nabla} I - (k_a + k_s) I + k_s \int_{4\pi} I(\vec{n}') W(\vec{n}, \vec{n}') d\vec{n}' + S \quad (5.9)$$

де c — швидкість світла; $I(\vec{r}, \vec{n}, t)$ — інтенсивність випромінювання,

що поширюється в напрямку \vec{n} в точці \vec{r} в момент часу t ; $k_a(\vec{r}, T)$, $k_s(\vec{r}, T)$ — коефіцієнти поглинання та розсіювання випромінювання; $W(\vec{n}, \vec{n}')$ — індикатриса розсіювання з напрямку \vec{n}' в напрямку \vec{n} ; $d\Omega$ — елемент тілесного кута в напрямку поширення; $S(\vec{r}, \vec{n}, t)$ — об'ємне джерело випромінювання.

Всі процеси зміни інтенсивності за рахунок переносу випромінювання відбуваються з швидкістю світла в середовищі, тому порівняно з процесами дифузії тепла процес переносу світла

можна вважати стаціонарним: $\frac{\partial I}{\partial t} = 0$ — час, за який

встановлюється стаціонарний розподіл густини випромінювання значно менший, ніж час встановлення температури.

Перехід до об'ємної густини енергії $U(\vec{r}) = (1/c) \int_{4\pi} I(\vec{r}, \vec{n}) d\Omega$

та потоку енергії $\vec{J}(\vec{r}) = \int_{4\pi} \vec{n} I(\vec{n}, \vec{r}) d\Omega$ з врахуванням стаціонарності

процесу дає можливість спростити¹ (5.9) і отримати рівняння дифузії випромінювання:

$$0 = \vec{\nabla} \left(\frac{1}{k_a + k_s} \vec{\nabla} U \right) - k_a U + S_{4\pi} \quad (5.10)$$

де $S_{4\pi} = \int_{4\pi} S(\vec{n}) d\Omega = k_s U_L$ — енергія ізотропного джерела

випромінювання в кожній точці, яку можна вважати рівною енергії розсіяного лазерного випромінювання; U_L — початкова енергія лазерного випромінювання, яка залежить від потужності лазера та типу розсіювача. Для простих типів розсіювача це значення рівне:

$$U_L(\vec{r}) = \frac{P_0}{2\pi|\vec{r}|^2} \exp(-k_a|\vec{r}|) — \text{точкове джерело;}$$

¹ Перехід відбувається шляхом множення (5.9) на \vec{n} , інтегрування по всім тілесним кутам, прирівнювання всіх похідних по часу нулеві та виключення невідомого потоку енергії.

$$U_L(\vec{r}) = \frac{2P_0}{\pi a_z^2} \exp\left(-\frac{2(|\vec{r}|^2 - z^2)}{a_z^2}\right) \exp(-k_a |\vec{r}|) -$$

одномодовий світловод радіуса a_0 , $a_z^2 = a_0^2 + \left(\frac{2\Lambda z}{\pi a_0}\right)^2$;

$$U_L(\vec{r}) = \frac{P_0}{\pi |\vec{r}|^2} \frac{z}{|\vec{r}|} \exp(-k_a |\vec{r}|) - \text{ідеальний ламбертівський розсіювач,}$$

де P_0 — потужність лазера, Λ — довжина хвилі випромінювання.

Граничні умови для розв'язання (5.10) можна записати наступним чином.

На прозорій поверхні світловода:

$$U|_{\Gamma} = 0, \Gamma = \Gamma_A,$$

тобто все випромінювання проходить через поверхню аплікатора.

На поверхні ідеального розсіювача:

$$\frac{1}{(k_a + k_s)} \frac{\partial U}{\partial \Gamma} \bigg|_{\Gamma} = \frac{P_0}{c S_{ap}}, \Gamma = \Gamma_A,$$

де S_{ap} — площа розсіювача.

Тобто зміна енергії, яка проходить через розсіювач, рівна енергії, що розсіюється на його поверхні.

На поверхні аплікатора (ідеальний відбивач):

$$\frac{1}{(k_a + k_s)} \frac{\partial U}{\partial \Gamma} \bigg|_{\Gamma} = 0, \Gamma = \Gamma_B.$$

Тобто потік енергії через поверхню аплікатора рівний нулю.

На віддаленій границі:

$$\frac{1}{(k_a + k_s)} \frac{\partial U}{\partial \Gamma} \bigg|_{\Gamma} = U|_{\Gamma}, \Gamma = \Gamma_{\infty}.$$

Тобто вся енергія, яка проходить через границю, поглинається.

Враховуючи циліндричну симетрію задачі, (5.10) можна записати у вигляді:

$$\frac{1}{r} \frac{\partial}{\partial r} \left[r \frac{1}{k_a + k_s} \frac{\partial U}{\partial r} \right] + \frac{\partial}{\partial z} \left[\frac{1}{k_a + k_s} \frac{\partial U}{\partial z} \right] - k_a U + S_{4\pi} = 0 \quad (5.11)$$

Розв'язання рівнянь

Для отримання розподілу температури у просторі і залежності цього розподілу в часі потрібно розв'язати рівняння (5.7) та (5.11) при відповідних початкових та граничних умовах. Ці рівняння мають аналогічний вигляд, тому різницеву схему для них можна вибрати однаковою.

Для розв'язання рівнянь в частинних похідних застосовуються різницеві апроксимації задач [13]. Неперервний час замінюється дискретними моментами з кроком τ : $t_n = n\tau$, неперервні просторові координати замінюються сіткою з кроком h : $x_{i,j,k} = (i,j,k)h$. Апроксимація здійснюється шляхом розкладу наших невідомих функцій у ряд Тейлора певного порядку точності по часовим і просторовим координатам з малими параметрами h і τ і відкиданням членів вищого порядку малості по h і τ . Кількість членів ряду Тейлора, які лишаються, називається порядком схеми. Позначимо $u_{i,j,k}^n$ — значення нашої невідомої функції у вузлі просторової сітки, що має номер i, j, k , і момент часу, що має номер n .

Приклад схеми другого порядку по координатам і першого порядку по часу для одновимірної задачі

$$\frac{u_i^{n+1} - u_i^n}{\tau} = \frac{\lambda_i \left(\frac{u_{i+1}^n - u_i^n}{h} \right) - \lambda_{i-1} \left(\frac{u_i^n - u_{i-1}^n}{h} \right)}{h} + k_i u_i^n + F_i.$$

Звідси можна знайти розв'язок відносно u_i^{n+1} :

$$u_i^{n+1} = u_i^n (1 - \tau k_i) + \frac{\tau}{h^2} (\lambda_i u_{i+1}^n - u_i^n (\lambda_i + \lambda_{i-1}) + \lambda_{i-1} u_{i-1}^n) + \tau F_i.$$

У багатовимірному випадку з'являються доданки, пов'язані з апроксимаціями похідних за іншими просторовими координатами. Повторюючи процедуру для наступних моментів часу, знаходимо розв'язок рівняння у наступний момент часу через розв'язок у попередній момент.

Різницевої схеми характеризуються стійкістю і точністю. Точність характеризує те, як змінюється похибка розв'язку задачі при зміні довжини кроку у часі та просторі. Схеми вищого порядку мають вищу точність, ніж схеми нижчого порядку. Стійкість різницевої схеми характеризує наскільки похибка розв'язку, яка виникла на одному

часовому кроці, змінюється при переході на наступний часовий крок. Якщо похибка не збільшується, то схема стійка, якщо похибка збільшується, то схема — нестійка.

Для розв'язання нашої задачі було вибрано дві схеми розв'язку задачі: чисто неявна схема та схема Франка-Нікольсона. Перша схема, яка є абсолютно стійкою, в одновимірному випадку має вигляд:

$$u_i^{n+1} = u_i^n (1 - tk_i) + \frac{t}{h^2} (\lambda_i u_{i+1}^{n+1} - u_i^{n+1} (\lambda_i + \lambda_{i-1}) + \lambda_{i-1} u_{i-1}^{n+1}) + tF_i$$

Точність цієї схеми $O(h^2 + \tau)$.

Схема Франка-Нікольсона також є абсолютно стійкою, але має вищу точність по часу:

$$u_i^{n+1} = u_i^n (1 - tk_i) + \frac{t}{2h^2} (\lambda_i u_{i+1}^{n+1} - u_i^{n+1} (\lambda_i + \lambda_{i-1}) + \lambda_{i-1} u_{i-1}^{n+1}) + \frac{t}{2h^2} (\lambda_i u_{i+1}^n - u_i^n (\lambda_i + \lambda_{i-1}) + \lambda_{i-1} u_{i-1}^n) + F_i$$

Точність цієї схеми є $O(h^2 + \tau^2)$.

Обидві схеми є неявними, тобто вимагають розв'язання системи рівнянь для знаходження розв'язку у наступний момент часу.

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \quad (5.12)$$

Де вектор \mathbf{u} — це набір значень функції у всіх вузлах просторової сітки. Значення цього вектора в момент часу $n+1$ позначимо як u_i^{n+1} . Матриця \mathbf{A} залежить від параметрів задачі, значення розв'язку в момент часу n — u_i^n , а також від розв'язку u_i^{n+1} , тобто ця система рівнянь є квазілінійною. Вектор \mathbf{f} залежить від параметрів задачі. Матриця \mathbf{A} має *стрічкову структуру*, тобто більшість елементів цієї матриці є нульовими, ненульові ж елементи лежать на стрічках, що паралельні діагоналі. Кількість стрічок для одновимірної задачі рівна 3, для двовимірної — 5, для тривимірної — 7. Кількість операцій розв'язання системи лінійних рівнянь з стрічковою матрицею порядку Nd , де d — ширина стрічки, а N — розмірність матриці.

Для розв'язання (5.12) звичайно використовують ітераційні методи. Початковим наближенням розв'язку є розв'язок у момент часу n . На кожній ітерації розв'язується система лінійних рівнянь з стрічкоподібною матрицею \mathbf{A} , для чого використовується швидкий

метод типу методу прогонки, після чого знову обчислюється матриця \mathbf{A} і так повторюється до збіжності.

В нашому випадку було вибрано метод верхньої релаксації. Матриця \mathbf{A} розбивається на суму нижньотрикутної — \mathbf{A}^- , верхньотрикутної — \mathbf{A}^+ , і діагональної \mathbf{A}_d : $\mathbf{A} = \mathbf{A}^- + \mathbf{A}_d + \mathbf{A}^+$.

Розв'язок шукається з ітераційного процесу:

$$\begin{aligned}\hat{\mathbf{u}}^{k+1} &= \mathbf{A}_d^{-1} \left(-\mathbf{A}^- \mathbf{u}^{k+1} - \mathbf{A}^+ \mathbf{u}^k + \mathbf{f} \right) \\ \mathbf{u}^{k+1} &= \omega \hat{\mathbf{u}}^{k+1} + (1 - \omega) \mathbf{u}^k\end{aligned}$$

де $(1 \leq \omega \leq 2)$ — параметр релаксації, підбором якого можна покращити збіжність. Ітерації закінчуються, коли виконується умова:

$$\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_2 < \varepsilon$$

де ε — допустима точність розв'язку.

Таким чином, остаточно нашу задачу можна записати у наступному вигляді. Задаються табличні значення теплофізичних параметрів задачі в точках простору з циліндричними координатами r, z при температурі T : $C_V(r, z | T)$ — теплоємність, $\lambda(r, z | T)$ — коефіцієнт теплопровідності, $B_V(r, z | T)$ — коефіцієнт відтоку тепла за рахунок кровотоку та інші; а також оптичні параметри: $k_a(r, z | T)$ — коефіцієнт поглинання, $k_s(r, z | T)$ — коефіцієнт розсіювання, параметри аплікатора та інші. Як для рівняння дифузії тепла, так і для рівняння переносу випромінювання потрібно розв'язати систему рівнянь (5.12), яку у розгорнутому вигляді можна записати наступним чином:

$$-\mathbf{A}_{r-1,z}^t U_{r-1,z}^t - \mathbf{A}_{r,z-1}^t U_{r,z-1}^t + \mathbf{A}_{r,z}^t U_{r,z}^t - \mathbf{A}_{r+1,z}^t U_{r+1,z}^t - \mathbf{A}_{r,z+1}^t U_{r,z+1}^t = \mathbf{f}_{r,z}$$
де $U_{r,z}^t$ — розподіл температури, або густини енергії випромінювання в момент часу t , $\mathbf{A}_{r,z}^t$ — матриця параметрів задачі.

У випадку чисто неявної схеми для рівняння теплопровідності коефіцієнти мають вигляд

$$\mathbf{A}_{r-1,z}^t = \begin{cases} \frac{\tau}{h^2} \left(1 - \frac{1}{2r} \right) \frac{\lambda(r, z | T_{r,z}^t) + \lambda(r-1, z | T_{r-1,z}^t)}{2C_V(r, z | T_{r,z}^t)} & ; r \neq 0 \\ 0 & ; r = 0 \end{cases}$$

$$A_{r+1,z}^t = \begin{cases} \frac{\tau}{h^2} \left(1 + \frac{1}{2r}\right) \frac{\lambda(r,z|T_{r,z}^t) + \lambda(r+1,z|T_{r+1,z}^t)}{2C_V(r,z|T_{r,z}^t)} & ; r \neq 0 \\ 2 \frac{\tau}{h^2} \frac{\lambda(0,z|T_{0,z}^t) + \lambda(1,z|T_{1,z}^t)}{C_V(0,z|T_{0,z}^t)} & ; r = 0 \end{cases}$$

$$A_{r,z-1}^t = \frac{\tau}{h^2} \frac{\lambda(r,z|T_{r,z}^t) + \lambda(r,z-1|T_{r,z-1}^t)}{2C_V(r,z|T_{r,z}^t)}$$

$$A_{r,z+1}^t = \frac{\tau}{h^2} \frac{\lambda(r,z|T_{r,z}^t) + \lambda(r,z+1|T_{r,z+1}^t)}{2C_V(r,z|T_{r,z}^t)}$$

$$A_{r,z}^t = 1 + A_{r-1,z}^t + A_{r,z-1}^t + A_{r+1,z}^t + A_{r,z+1}^t + \tau \frac{B_V(r,z|T_{r,z}^t)}{C_V(r,z|T_{r,z}^t)}$$

$$f_{r,z} = T_{r,z}^{t-1} + \tau \frac{B_V(r,z|T_{r,z}^t)}{C_V(r,z|T_{r,z}^t)} T_0 + \tau \frac{1}{C_V(r,z|T_{r,z}^t)} F(r,z)$$

де $F(r,z)$ — функція джерела випромінювання (густина поглинутої енергії випромінювання, яка визначається з рівняння дифузії тепла).
Для рівняння дифузії випромінювання:

$$\tilde{\lambda} = \frac{1}{k_a + k_s}$$

$$A_{r-1,z}^t = \begin{cases} \left(1 - \frac{1}{2r}\right) \frac{\tilde{\lambda}(r,z|T_{r,z}^t) + \tilde{\lambda}(r-1,z|T_{r-1,z}^t)}{2h^2} & ; r \neq 0 \\ 0 & ; r = 0 \end{cases}$$

$$A_{r+1,z}^t = \begin{cases} \left(1 + \frac{1}{2r}\right) \frac{\tilde{\lambda}(r,z|T_{r,z}^t) + \tilde{\lambda}(r+1,z|T_{r+1,z}^t)}{2h^2} & ; r \neq 0 \\ 2 \frac{\tilde{\lambda}(0,z|T_{0,z}^t) + \tilde{\lambda}(1,z|T_{1,z}^t)}{h^2} & ; r = 0 \end{cases}$$

$$A_{r,z-1}^t = \frac{\tilde{\lambda}(r,z|T_{r,z}^t) + \tilde{\lambda}(r,z-1|T_{r,z-1}^t)}{2h^2}$$

$$A_{r,z+1}^t = \frac{\tilde{\lambda}(r,z|T_{r,z}^t) + \tilde{\lambda}(r,z+1|T_{r,z+1}^t)}{2h^2}$$

$$A_{r,z}^t = A_{r-1,z}^t + A_{r,z-1}^t + A_{r+1,z}^t + A_{r,z+1}^t + k_a(r,z|T_{r,z}^t)$$

$$f_{r,z} = S_{4\pi}(r,z),$$

де $S_{4\pi}(r,z)$ — густина ізотропного лазерного випромінювання, що залежить від типу та потужності лазерного джерела.

Програма моделювання лазерної термодеструкції

Для моделювання лазерної термодеструкції було розроблено прикладну програму, яка виконує наступні дії:

1. Зчитування параметрів задачі з файлу.
2. Корекція параметрів за допомогою інтерактивного діалогу.
3. Знаходження розподілу енергії випромінювання.
4. Знаходження розподілу теплового поля для різних моментів часу.
5. Виведення даних.

Інтерфейс програми показано на рис. 5.3.

Файл `zzz.txt` в робочій директорії програми містить опис можливих значень параметрів. Параметри зчитуються з файлу `teplo.ini` і їх можна поміняти інтерактивним чином за допомогою діалогу "Param". Зокрема, можна встановити наступні параметри:

1. Тип аплікатора, `type` (табл. 5.1).
2. Потужність лазерного випромінювання, I_0 .
3. Режим подачі випромінювання (неперервний, імпульсний).
4. Параметри інтегрування по часу (довжина кроку `dt` і кількість кроків `tmax`).
5. Параметри візуалізації і побудови графіків (`graphing`).

Таблиця 5.1 – Типи аплікатора

Значення параметра <code>type</code>	Тип аплікатора
10	Боковий розсіювач
101	Гаусів пучок
105	Сферичний розсіювач

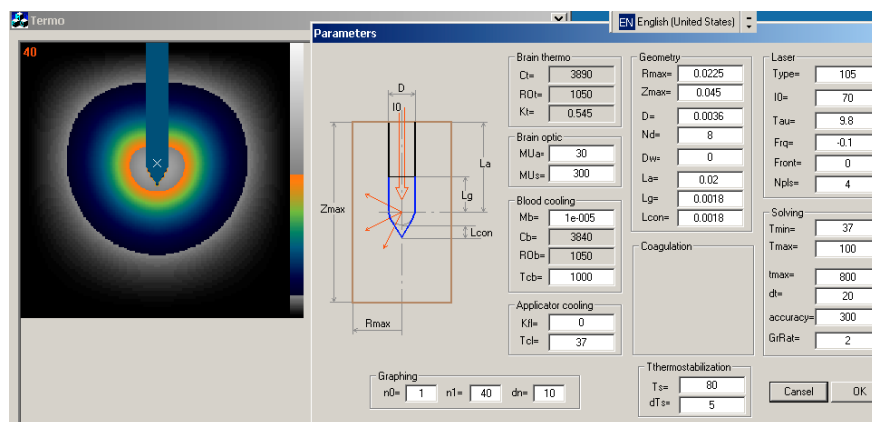


Рисунок 5.3 – Інтерфейс користувача програми моделювання лазерної термодеструкції.

Розрахунок виконується на просторовій сітці розміром 64×128 . Перший крок по часу вимагає для розв'язання системи лінійних рівнянь близько $10 \div 30$ ітерацій. Наступні кроки близько двох ітерацій. При цьому точність знаходження температури — 0.1 градуса, крок просторової сітки — $0,1 \div 0,2$ мм, крок по часу — $0,1 \div 2$ с.

Обраховані програмою ізотерми можна вивести на томографічному зображенні для співставлення ізотерм з анатомічними структурами.

Завдання

1. Оцініть, яка кількість операцій і скільки часу потрібно для обрахунку одного кроку по часу при розв'язання рівняння теплопровідності для вашої обчислювальної системи.
2. Проведіть моделювання розподілу теплового поля для будь-яких значень параметрів, визначте, скільки часу зайняв обрахунок одного часового кроку, і порівняйте отримані дані з обрахованими в першому завданні. Обрахунок запускається кнопкою "Calc". Кількість кроків по часу задається параметром діалогу tmax.
3. Визначте, для якого типу аплікатора і при яких параметрах отримується задовільний ефект термодеструкції. Для того, щоб побудувати просторовий розподіл температури, потрібно вказати номери часових кроків, при яких будуються графіки, за допомогою групи "Graphing" діалогу параметрів (n0 — номер початкового кроку по часу, n1 — номер кінцевого кроку по часу, dn — крок з

- яким будувати графіки).
4. Дайте відповідь на контрольні запитання.

Контрольні запитання

1. Якими рівняннями описується поле температури і густина енергії випромінювання? У яких випадках ці рівняння можуть бути одного типу? Чому не потрібно розв'язувати в часі задачу переносу випромінювання?
2. Які характеристики різницевих схем для розв'язку задачі дифузії тепла при термодеструкції?
3. Яка кількість операцій потрібна для розрахунку одного часового кроку задачі дифузії тепла? Чи можна цю задачу розв'язувати в реальному часі?
4. При яких параметрах аплікатора отримується найкращий ефект термодеструкції?

Практичне заняття № 6.

Моделювання сканування та реконструкції зображень в проективній томографії

Мета: практично ознайомитись з методами комп'ютерного моделювання процесу отримання інтроскопічних даних та з чисельними методами розв'язання некоректних задач на прикладі задачі моделювання процесу отримання проекцій в комп'ютерній томографії та побудови зображень за цими проекціями.

Задача розрахунку томографічних проекцій

Загальні відомості: Моделювання процесів сканування та реконструкції в томографії є ефективним засобом дослідження та розробки нових фізичних методів збору томографічних даних, нових методів реконструкції зображень, методів фільтрації шуму та ін., які використовуються в сучасних медичних технологіях і т.п.. Задача реконструкції зображень за томографічними даними є математично некоректно поставленою [14]. Це означає, що задача може не мати розв'язку, розв'язок може бути не єдиним, а також задача може бути нестійкою за вхідними даними (невелика зміна вхідних даних призводить до великої зміни розв'язку). Для таких задач застосовуються різні методи регуляризації, тобто накладаються додаткові умови, за яких задача стає коректно поставленою. Моделювання процесу реконструкції дає можливість вибрати додаткові умови найкращим чином.

До проективних методів належить трансмісійна рентгенівська комп'ютерна томографія (КТ), емісійна томографія (ПЕТ, ОФЕТ)², ультразвукова томографія в наближенні коротких хвиль та інші методи.

Комп'ютерні томографічні системи

Розглянемо задачу на прикладі КТ. Інші методи мають свої особливості, проте загальна ідея залишається такою самою. Рентгенівська комп'ютерна томографія (звичайно називається просто

² ПЕТ — позитронна емісійна томографія, ОФЕТ — однофотонна емісійна томографія.

комп'ютерною томографією або КТ) — це метод, який дозволяє будувати зображення розподілу коефіцієнту поглинання рентгенівського випромінювання в середині об'єктів неінвазивним чином (тобто без руйнування об'єктів). Основна ідея показана на рис. 6.4, де подана схема отримання томографічних проекцій у випадку двовимірного об'єкта. Рентгенівське випромінювання проходить через досліджуваний об'єкт. Всередині об'єкту випромінювання поглинається. В залежності від внутрішньої будови та форми об'єкту інтенсивність випромінювання, що пройшло через об'єкт, буде різною. Інтенсивність випромінювання, що пройшло через об'єкт, реєструється детектором.

Якщо рентгенівську трубку і детектор рухати поступально біля об'єкта вздовж прямої s , то ми отримаємо проекцію об'єкта на пряму, паралельну прямій s , а координата s буде координатою вздовж проекції. Якщо систему трубка–детектор повернути навколо об'єкта на кут φ , то можна одержати проекцію з іншим ракурсом. При наявності повного набору проекцій, тобто значень сигналу на виході детектора при всіх положеннях системи трубка–детектор, за цими проекціями можна відновити розподіл коефіцієнта поглинання всередині об'єкта. Даний метод використовується для отримання зображень поперечних (аксіальних) перерізів анатомічних структур.

Існують різні реалізації комп'ютерних томографічних систем [15]. Томографічні системи першого покоління мали одну гармату, що випромінювала вузький пучок випромінювання, і один детектор. Система гармата-детектор рухалась поступально перпендикулярно тілу пацієнта для отримання даних однієї проекції, а також оберталась навколо тіла пацієнта. Така система відповідає ідеальній схемі, проте завдяки необхідності механічних рухів є найповільнішою (декілька хвилин на отримання повного набору проекцій) і висуває серйозні вимоги до стабільності механіки.

Система другого покоління має декілька детекторів (8-16) і гармату, що випромінює розширений пучок випромінювання. В таких системах зменшується час отримання проекцій (декілька десятків секунд), проте вимоги до механіки залишаються високими, а також додаються вимоги щодо однакових параметрів всіх детекторів і багатоканального підсилювача.

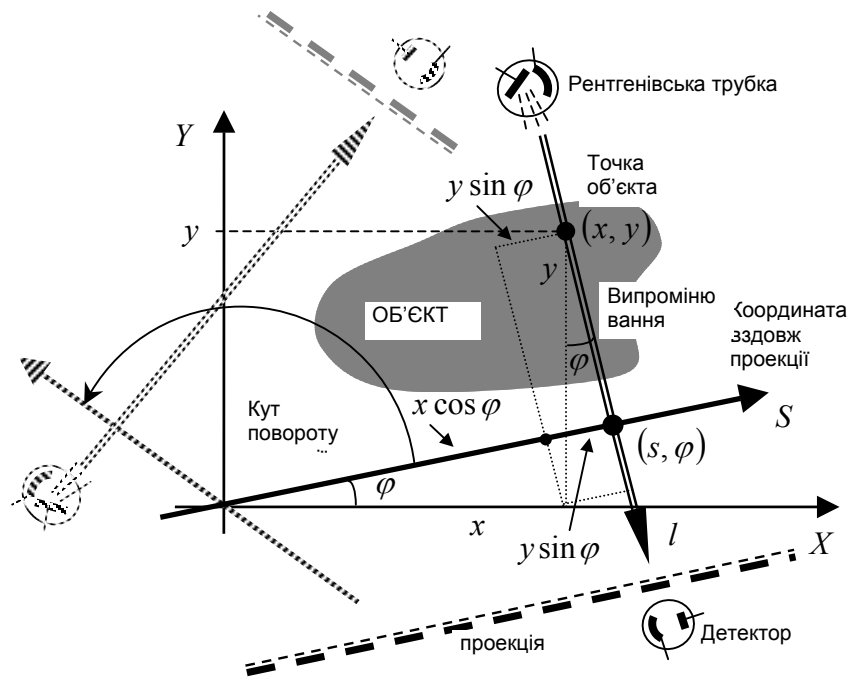


Рисунок 6.4 – Геометрія моделі процесу сканування в комп'ютерній томографії.

Системи третього покоління мають гармату, що випромінює віялоподібний пучок, і велику кількість (300-600) детекторів, що дозволяє отримати проекцію за один імпульс трубки і позбавляє необхідності поступальних рухів. Система трубка-детектори рухається поступально. Томографічні системи третього покоління є найпоширенішими і мають час отримання повного набору проекцій близько 1 секунди.

Томографи четвертого покоління мають систему детекторів, що розташовані по колу, і рухається лише гармата. В таких системах час отримання проекцій ще менший, проте з'являється вплив розсіяних фотонів, що ускладнює алгоритм реконструкції зображень. Справа в тому, що в системах четвертого покоління діаграма направленості детекторів має бути широкою, щоб мати можливість реєструвати випромінювання з різних кутів при повороті трубки. В томографічних системах старших поколінь впливом розсіювання можна знехтувати, оскільки кожний детектор реєструє лише вузький пучок

випромінювання з одного напрямку. Варіантом систем четвертого покоління є спіральні томографи, в яких трубка рухається також поступально вздовж тіла пацієнта.

Томографічні системи п'ятого покоління не мають рухомих частин. Розгортка променя виконується електронним чином, що ще зменшує час накопичення даних. Інші апаратні складові аналогічні до систем четвертого покоління.

При моделюванні необхідно враховувати всі важливі особливості реалізації томографічної системи.

Перенесення рентгенівського випромінювання

Фізичні процеси, що відбуваються при проходженні рентгенівського випромінювання через речовину, можна математично описати за допомогою рівняння переносу:

$$\frac{\partial I(\vec{r}, \vec{n}, \lambda)}{\partial t} = -\vec{n} \vec{\nabla} I - k_a(\vec{r}, \vec{n}, \lambda) I(\vec{r}, \vec{n}, \lambda) - k_s(\vec{r}, \vec{n}, \lambda) I(\vec{r}, \vec{n}, \lambda) + k_s(\vec{r}, \vec{n}, \lambda) \iint I(\vec{n}', \lambda') W(\vec{n}, \vec{n}', \lambda, \lambda') d\vec{n}' d\lambda' + S(\vec{r}, \vec{n}, \lambda) \quad (6.13)$$

де $I(\vec{r}, \vec{n}, \lambda)$ — інтенсивність випромінювання з довжиною хвилі λ , що поширюється в напрямку \vec{n} в точці з координатами \vec{r} ; k_a — коефіцієнт поглинання випромінювання з даною довжиною хвилі в заданій точці; k_s — коефіцієнт розсіювання випромінювання, що поширюється в заданому напрямку; W — індикатриса розсіювання випромінювання. Слід зауважити, що в загальному випадку коефіцієнти, що характеризують властивості речовини, можуть бути анізотропними, тобто залежать від напрямку поширення випромінювання (наприклад, для кристалічних структур), проте для біологічних об'єктів залежності від напрямку майже не спостерігається. Дане рівняння виражає наступне твердження: інтенсивність випромінювання заданої довжини хвилі, що поширюється в даній точці простору в даному напрямку, зменшується за рахунок поширення випромінювання з даної точки в сусідню, за рахунок поглинання і за рахунок розсіювання з даного напрямку в інші та з даної довжини хвилі в інші. Збільшення інтенсивності відбувається за рахунок розсіювання з інших напрямків в даний та за рахунок джерел випромінювання. Ефекти розсіювання рентгенівського випромінювання можуть бути нелінійними, такі, як ефект Комптона.

В загальному випадку (6.13) розв'язується важко, крім того, для цього рівняння можна ставити різні задачі. *Задача трансмісійної*

томографічної реконструкції: за відомим розподілом інтенсивності випромінювання зовні об'єкту знайти розподіл параметрів речовини (коефіцієнта поглинання). *Задача моделювання променевої терапії*: за відомими значеннями коефіцієнтів поглинання та розсіювання розрахувати поглинуту дозу випромінювання. *Зворотна задача розсіювання*: за відомою інтенсивністю випромінювання ззовні об'єкту розрахувати розподіл коефіцієнту розсіювання. *Задача емісійної томографічної реконструкції*: за інтенсивністю зовнішнього випромінювання розрахувати розподіл інтенсивності випромінювання, що випромінюється об'єктом. Ми розглянемо лише найпростішу задачу моделювання трансмісійної комп'ютерної томографії для томографічних систем першого, другого та третього поколінь.

Системи першого, другого та третього поколінь

Задача томографічної реконструкції є стаціонарною, тобто всі перехідні процеси, пов'язані із зміною інтенсивності в часі, закінчуються значно швидше, ніж триває процес збору томографічних даних, тому похідну в часі в рівнянні (6.13) можна вважати нульовою. В томографічних системах, що розглядаються, вплив розсіюваних фотонів є незначним, оскільки кожен детектор є вузьконаправленим і можна вважати, що він приймає випромінювання лише з виділеного напрямку. Градієнт замінюється похідною за напрямком поширення випромінювання l . Таким чином, (6.13) зводиться до:

$$\frac{dI(l, \lambda)}{dl} = -k_a(l, \lambda)I(\vec{r}, \lambda) + S(\lambda) \quad (6.14)$$

Для томографічних систем вище третього покоління рівняння (6.14) не є справедливим. При розсіюванні може змінюватись довжина хвилі (наприклад, за рахунок ефекту Комптона), у цьому випадку також не можна нехтувати розсіюванням. Ми цей ефект враховувати не будемо, що призведе до спотворення значень коефіцієнта поглинання. Рівняння (6.14) має простий розв'язок:

$$I(l, \lambda) = S(\lambda) e^{-\int_{-\infty}^l k_a(l', \lambda) dl'} \quad (6.15)$$

Реальні джерела рентгенівського випромінювання не є монохроматичними, тому інтенсивність сигналу на вході детектора буде мати вигляд:

$$I(l) = \int_{-\infty}^{\infty} S(\lambda) e^{-\int_{-\infty}^l k_a(l', \lambda) dl'} d\lambda \quad (6.16)$$

Рівняння (6.16) дає можливість моделювати проходження випромінювання через об'єкт з заданим значенням розподілу коефіцієнта поглинання, заданими залежностями коефіцієнта поглинання від довжини хвилі та заданими характеристиками випромінювача та детектора.

Моделювання томографічного сканування

Для розробки та дослідження алгоритмів реконструкції на перших етапах можна не враховувати залежність коефіцієнта поглинання від довжини хвилі. Будемо вважати випромінювання монохроматичним, при цьому з (6.15) легко отримати значення випромінювання, що пройшло через об'єкт.

Задамо значення розподілу коефіцієнта поглинання у вузлах прямокутної декартової сітки (рис. 6.5):

$$k_a(x, y) = k_a(n_x \Delta x, n_y \Delta y) = k_a(n_x, n_y),$$

де $\Delta x, \Delta y$ — розмір кроку в напрямку координат x, y відповідно, а n_x, n_y — цілі числа, номер вузла сітки. Розподіл $k_a(n_x, n_y)$ зручно задавати у вигляді зображення розміром $N_x \times N_y$ від 64×64 до 512×512 точок. Менші розміри не дають можливості оцінити якість реконструкції, а більші розміри не є доцільними, бо це збільшує кількість операцій. Глибину пікселя доцільно вибрати 16 біт, або більше, оскільки реальні зображення реконструюються з глибиною пікселя 16 біт. Значення коефіцієнта розподілу в будь-якій точці можна обрахувати шляхом інтерполяції. Для отримання значення інтенсивності сигналу проекцій $I(s, \varphi)$ вибирається сітка в просторі проекцій розміром $N_s \times N_\varphi$:

$$I(s, \varphi) = I(n_s \Delta s, n_\varphi \Delta \varphi) = I(n_s, n_\varphi),$$

де $\Delta s, \Delta \varphi$ — відповідно відстань між детекторами і крок по куту, n_s, n_φ — цілі числа, номери вузлів. Значення $\Delta x, \Delta y$ і Δs можна вибрати рівними 1, оскільки вони входять лінійно, а значення $\Delta \varphi = \frac{2\pi}{N_\varphi}$, де N_φ — кількість проекцій, N_s — кількість точок в

проекції. Для отримання значення проекції потрібно обрахувати променевий інтеграл:

$$p(s, \phi) = \int k_a[x(l, s, \phi), y(l, s, \phi)] dl \quad (6.17)$$

де за рис. 6.4. видно, що:

$$\begin{aligned} x(l, s, \phi) &= s \cos \phi + l \sin \phi \\ y(l, s, \phi) &= l \cos \phi + s \sin \phi \end{aligned} \quad (6.18)$$

Позитивний напрямок l співпадає з напрямком променя, а нуль відповідає точці перетину з прямою s .

Інтеграл (6.17) зручно брати за методом прямокутників. Вибирається крок інтегрування Δl вздовж променя, його значення має бути в 3-10 разів менше, ніж значення кроку просторової сітки $\Delta x, \Delta y$, значення інтеграла апроксимується сумою:

$$p(n_s, n_\phi) \approx \sum_i k_a(i\Delta l) \Delta l$$

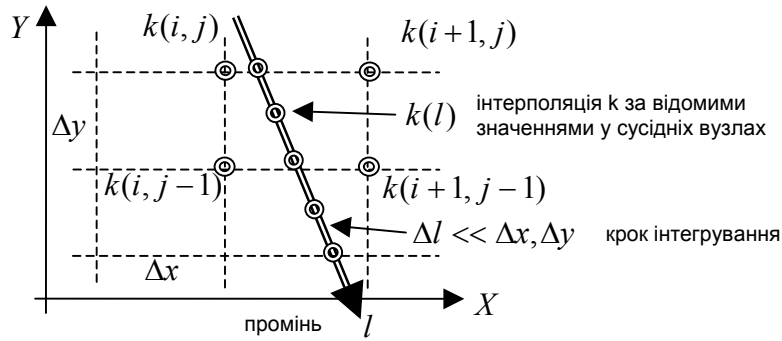


Рисунок 6.5 – Сітка для обчислення променевих інтегралів.

Значення $k_a(i\Delta l)$ одержуються шляхом інтерполяції за відомими значеннями у найближчих вузлах. Один з можливих варіантів — білінійна інтерполяція:

$$k(l) = k(x(l), y(l))$$

$$= (1-t)(1-u)k(i, j) + t(1-u)k(i+1, j) + u(1-t)k(i, j-1) + tuk(i+1, j-1),$$

де коефіцієнти визначаються за формулами:

$$i = [x / \Delta x] + N_x / 2$$

$$j = [y / \Delta y] + N_y / 2$$

$$u = (x - j\Delta x) / \Delta x$$

$$u = (y - j\Delta y) / \Delta y$$

Значення $x(l), y(l)$ обчислюються за формулою (6.18). Кількість операцій отримання повного набору проекцій рівна близько $N_s N_\varphi N_l \sim N^3$, мінімальний необхідний об'єм пам'яті близько $N_x N_y N_s N_\varphi \sim N^4$.

Моделювання реконструкції

Найпоширенішим методом реконструкції є метод згортки та зворотного проєціювання. Суть його полягає в обертанні рівняння (6.15). З цього рівняння видно, що променеві інтеграли рівні:

$$p(l) = p(s, \varphi) = -\ln \frac{I(l, \lambda)}{S(\lambda)} = \int_{-\infty}^l k_a(l', \lambda) dl'$$

Враховуючи, що $s = x \cos \varphi + y \sin \varphi$ (рис. 6.4), можна записати:

$$p(s, \varphi) = \int k_a(x, y) \delta(s - x \cos \varphi - y \sin \varphi) dx dy$$

Якщо взяти від останнього рівняння перетворення Фур'є за змінною s , то можна показати, що воно рівне перерізу двовимірного Фур'є образу функції $k_a(x, y)$, який відповідає прямій, що проходить через початок координат Фур'є простору під тим же кутом φ , що і проекція:

$$\begin{aligned} \int p(s, \varphi) e^{-i\omega_s s} ds &= \hat{k}_a(\omega_s \cos \varphi, \omega_s \sin \varphi) \\ \hat{k}_a(\omega_x, \omega_y) &= \int k_a(x, y) e^{-i(x\omega_x + y\omega_y)} dx dy \end{aligned} \quad (6.19)$$

За повним набором проекцій можна відновити повні дані Фур'є простору функції $k_a(x, y)$. Метод згортки і зворотного проєціювання виводиться з (6.19) і має наступний вигляд.

Спочатку обчислюються фільтровані проекції як згортка променевих інтегралів з функцією ядра $F(t)$:

$$P(x \cos \varphi + y \sin \varphi, \varphi) = \int p(s - t, \varphi) F(t) dt \Big|_{s=x \cos \varphi + y \sin \varphi} \quad (6.20)$$

Значення коефіцієнта поглинання обраховується як сума фільтрованих проекцій в заданій точці:

$$k_a(x, y) = \int P(x \cos \varphi + y \sin \varphi, \varphi) d\varphi \quad (6.21)$$

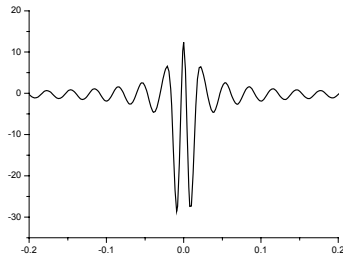
Фільтровані проекції (6.20) повинні бути обчислені у заданій точці $s = x \cos \varphi + y \sin \varphi$, проте доступний лише дискретний набір даних і значення у необхідній точці може бути невідомим, при цьому використовується лінійна інтерполяція:

$$P(s, \varphi) = P((i-1)\Delta s, \varphi) + \frac{P(\Delta s, \varphi) - P((i-1)\Delta s, \varphi)}{\Delta s} (s - i\Delta s)$$

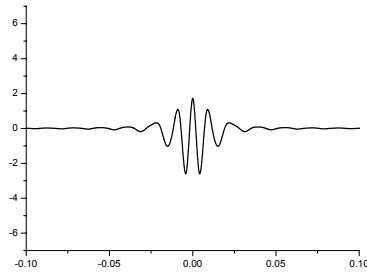
Функція ядра згортки $F(t)$ визначається наступним чином:

$$F(t) = \int |\omega| e^{i\omega t} d\omega$$

Даний інтеграл розбігається, тому доводиться використовувати різні наближені значення ядра згортки. Застосування наближень відповідає процедурі регуляризації. В якості наближення підінтегральна функція покладається рівною нулеві за межами певного інтервалу, або зменшується її зростання. Вигляд функції ядра сильно впливає на точність реконструкції, на роздільну здатність, а також на стійкість. Чим більш гладка функція ядра, тим вища стійкість, але менша роздільна здатність і навпаки. На рис. 6.6 показано вигляд функцій ядра згортки: з меншим згладжуванням (а) та з більшим згладжуванням (б).



а)



б)

Рисунок 6.6 – Вигляд ядра згортки незгладжене а) та згладжене б).

Вибір ядра згортки та інші методи фільтрації проєкцій суттєво впливають на якість реконструкції і використовуються для керування стійкістю, боротьби з шумами, компенсації факторів неідеальності.

Моделювання факторів неідеальності

До факторів неідеальності належать всі ті ефекти, які не враховуються моделлю реконструкції. Такими є немонохроматичність випромінювання, вплив розсіювання, нестабільність руху системи трубка-детектори, нестабільність підсилювачів, шуми детекторів. Врахування такого фактора, як розсіювання, вимагає розв'язання точного рівняння (6.13). Врахування немонохроматичності вимагає використання виразу (6.16). Вплив квантових шумів моделюється, як випадковий процес з Пуасонівським розподілом з дисперсією $\sigma = \sqrt{N}$, де N — кількість фотонів, що пройшла до детектора, яка пропорційна інтенсивності випромінювання. Нестабільність параметрів за рахунок руху детекторів зручно моделювати, як адитивний періодичний процес:

$$\sigma = \sum_i A_i \sin(\omega_i t + \alpha_i)$$


Для точного моделювання також можна врахувати нелінійність характеристики детектора, шуми підсилювача та ін..

Програма моделювання сканування та реконструкції

Для моделювання сканування та реконструкції в проєктивній трансмісійній комп'ютерній томографії на кафедрі медичної радіофізики було створено програму, яка також може використовуватись в ролі інтерфейсу керування комп'ютерним томографом (рис. 6.7).

Програма дає можливість проводити розрахунок модельних проєкцій заданого розподілу коефіцієнта поглинання за описаною вище моделлю, а також проводити відновлення розподілу коефіцієнта поглинання рентгенівського випромінювання за проєкціями.

Для розрахунку проєкцій розподіл коефіцієнта поглинання задається у вигляді зображення глибиною пікселя 8 біт і розміром до 512×512 точок. Для зображень більшого розміру на краях зображення інформацію буде втрачено. Зображення потрібно відкрити за допомогою меню "**Файл->Открыть**". Запуск розрахунку проєкцій відбувається за допомогою пункту меню "**Run->Emul**". Моделюється 480 проєкцій розмірністю 512 точок. Розраховані

проекції на екрані не відображаються. Для візуалізації проєкцій потрібно їх записати на диск за допомогою пункту меню "Run->Save Prj Set". Проєкції записуються у дисковий файл з іменем _INOUT\tmp.prs у каталозі, що містить проєкт програми. При повторному записі даний файл переписується, тому варто зробити його копію з іншим іменем. Відкрити проєкції можна за допомогою меню "Файл->Открыть" і вибрати тип даних *Projection*. Проєкції мають глибину пікселя 16 біт, тому для коректного відображення необхідно користуватись діалогом , на якому вибрати режим корекції за допомогою миші (кнопка "Мышью").

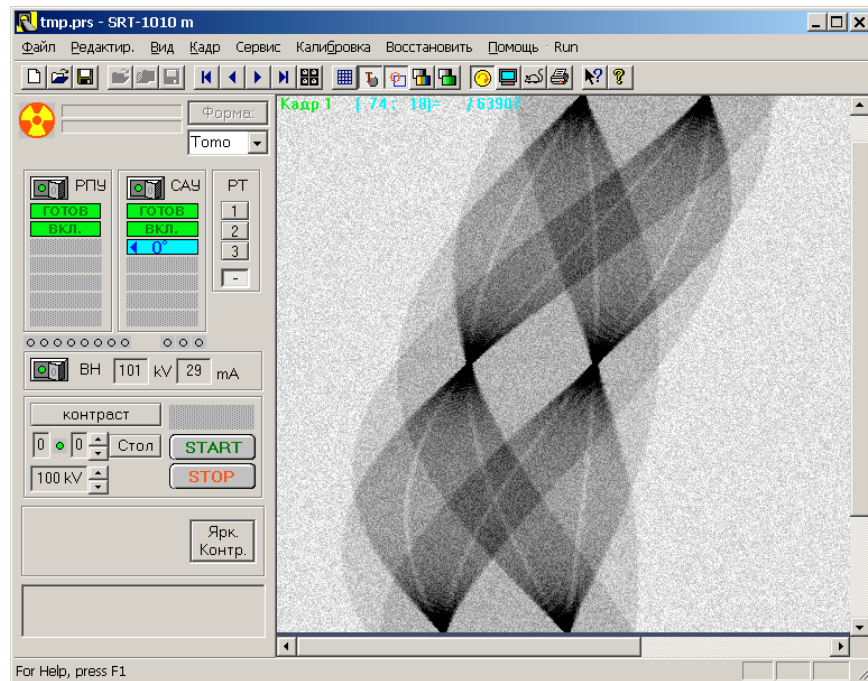




Рисунок 6.7 – Інтерфейс програми моделювання сканування та реконструкції зображень в комп'ютерній томографії.

Відновлення проєкцій з відкритого файлу запускається за допомогою меню "Восстановить". Цей пункт має два підпункти: "Трубо" та "Точно", які дозволяють запускати реконструкцію відповідно за неповним та повним набором проєкцій. В першому випадку швидкість реконструкції буде вищою. Реконструйоване

зображення відображається на екрані. За допомогою діалогу  є можливість проводити статистичний аналіз зображення. Для цього на зображення виділяється замкнута область за допомогою відповідних інструментів (коло, квадрат, замкнутий контур). За допомогою інструменту редагування  необхідно активізувати **контур** виділеної області **лівою кнопкою миші**. Після цього натиснення **правої кнопки миші** у середині виділеної області запускає розрахунок статистичних параметрів виділеної області, які відображаються у вікні "**ред. текста**". Статистична інформація дає можливість оцінити співвідношення сигнал/шум заданої області зображення.

Моделювання сканування реалізується за допомогою класу `class CProjSet;`

Функція

`void CProjSet::Emul(CPicCT* source, double mmp=0.5);`

розрахуває двовимірні проекції розподілу коефіцієнта поглинання, що задається зображенням `source`. Важливими є також наступні поля цього класу:

<code>int CProjSet::Nd</code>	Кількість точок в проекції
<code>int CProjSet::Orient</code>	Кількість проекцій (орієнтацій)
<code>A_real* CProjSet ::</code>	Отримати адресу першого елемента
<code>Get(int n_Pr)</code>	проекції з номером <code>n_Pr</code> . В масиві зберігаються саме проекції, а не променеві інтеграли!

Реконструкція підтримується за допомогою класу

`class CIP_mP;`

Важливими елементами цього класу є:

<code>CProjSet* CIP_mP::PrjSet</code>	Посилання на набір проекцій.
<code>void CIP_mP::DoPolar_Low()</code>	Відновити зображення з швидким методом
<code>void</code>	відновити зображення високої якості (повільно)
<code>CIP_mP::DoPolar_High()</code>	Ядро згортки — масив даних розміром <code>CIP_mP::nM</code> . Масив містить "половину" ядра згортки, яке є симетричним. Останній елемент масиву містить значення ядра згортки в точці 0.
<code>A_real* CIP_mP::my_KERm</code>	

CIP_mPAT* CIP_mP::myPAT	Посилання на таблиці
	акселерації
void CIP_mP::Convolve(int nom_Pn)	Обрахувати згортку проекції
	номер nom_Pn з ядром.

Для встановлення ядра згортки можна присвоїти елементу A_real* CIP_mP::my_KERm адресу масиву, що містить ядро згортки, або скористатись готовими ядрами таблиці акселерації. На поточне ядро таблиці акселерації вказує елемент myPAT->v_KERm. Значення вказівника на ядро згортки також можна задати за допомогою функції:

```
int CIP_mPAT::SetConvolutorAT(A_real* pKERm, int ConvType).
```

Завдання

1. Створіть на локальному комп'ютері копію проекту програми моделювання сканування-реконструкції. Оригінал міститься в каталозі, вказаному викладачем. Перевірте, що програма компілюється та запускається коректно.
2. За допомогою довільного графічного редактора створіть тестове зображення розподілу коефіцієнта поглинання для оцінки якості сканування та реконструкції. Зображення повинно містити елементи (тонкі лінії, плавні перепади яскравості, однорідні ділянки) для оцінки роздільної здатності, стійкості та наявності артефактів.
3. Оцініть кількість операцій та час, які потрібні для моделювання сканування та реконструкції.
4. Запустіть моделювання сканування створеного тестового розподілу коефіцієнта поглинання. Проведіть реконструкцію зображення за модельними проекціями в різних режимах. Визначте час сканування та реконструкції.
5. Оцініть, на скільки реконструйоване зображення відрізняється від оригіналу за такими характеристиками, як співвідношення сигнал/шум та роздільна здатність. Оцініть наявність артефактів реконструкції.
6. Змодельуйте вплив ядра згортки на якість реконструкції, для цього у функції void CIP_mP::DoPolar_Low() чи void CIP_mP::DoPolar_High() замініть ядро згортки на інше шляхом виклику функції int CIP_mPAT::SetConvolutorAT(A_real* pKERm, int ConvType).
7. Змодельуйте, як впливає фільтрація проекцій перед реконструкцією на якість реконструкції, співвідношення

сигнал/шум та роздільну здатність реконструйованих розподілів. Для фільтрації можна застосувати будь-який фільтр згладжування. Обробку проекцій варто помістити у функції

```
void CIP_mP::DoPolar_Low()
void CIP_mP::DoPolar_High()
```

чи

8. Дайте відповіді на контрольні запитання.

Контрольні запитання

1. Які фізичні ефекти потрібно враховувати для моделювання сканування та реконструкції?
2. Скільки операцій потрібно для моделювання сканування та реконструкції в даній роботі, скільки часу це займе на вашій обчислювальній машині?
3. Скільки часу займає процес моделювання сканування та реконструкції на вашій обчислювальній машині? Чи є відмінність від оцінених значень і чому?
4. Як впливає форма ядра згортки на роздільну здатність, співвідношення сигнал/шум та рівень артефактів реконструкції? При якій формі ядра отримано найкращий результат?
5. Як впливає попередня фільтрація проекцій на якість реконструкції? При яких параметрах фільтру отримано найкращу якість реконструкції?

Практичне заняття № 7.

Моделювання молекулярних структур

Мета: практично ознайомитись з методами комп'ютерного моделювання квантово-механічних систем на прикладі розрахунку енергетичних, геометричних та спектральних характеристик речовини.

Методи моделювання молекулярних структур

Загальні відомості: Сьогодні комп'ютерне моделювання фізичних процесів, що відбуваються на молекулярному рівні, широко застосовується у різних галузях науки та промисловості, таких як хімія, мікроелектроніка, молекулярна біологія. Розрахунки на основі квантово-механічних моделей дозволяють непрямим чином отримати ту інформацію про механізми функціонування молекулярних структур, яку неможливо отримати ніякими експериментальними методами. Прикладом може бути нова галузь молекулярної біофізики — комп'ютерне моделювання лікарських препаратів. Шляхом комп'ютерного моделювання на основі квантово-механічних та напівкласичних моделей виявляється можливим знаходження хімічних формул лікарських препаратів, які оптимальним чином діють на задані вірусні чи клітинні білки. При цьому на вхід подаються дані рентгеноструктурного аналізу білка, а на виході одержують формули оптимальних лікарських препаратів.

Крім цього, сьогодні ведуться роботи по створенню структур, які дозволяють виконувати різні корисні функції на молекулярному рівні, наприклад, працювати як електронні пристрої (наноелектроніка), як пристрої, що вибірково діють на складні біологічні молекули (нанороботи) та ін.. Всі ці розробки об'єднують під одним терміном *нанотехнологія*. Розвиток цього напрямку, взагалі не можливий без комп'ютерного моделювання, оскільки на експерименті вдається виміряти лише зовнішні прояви дії цих пристроїв, а комп'ютерна модель, хоч і є наближеною, але дозволяє вивчати та оптимізувати механізми їх роботи.

Квантово-хімічні методи

Моделювання молекулярних структур зводиться до розрахунку

енергії цих структур в необхідних зовнішніх умовах. Знаючи енергію, з'являється можливість знайти багато інших характеристик, як самої молекули, так і макроскопічних характеристик речовини. Наприклад, оптимальна геометрія молекули відповідає мінімуму її енергії, значення власних частот інфрачервоного (ІЧ) спектру — через перші похідні від енергії і т.д..

Всі фізичні характеристики на молекулярному рівні можна з достатньою точністю розрахувати лише на основі квантової механіки. Для опису положення атомів у молекулі в будь-який момент часу t вводиться хвильова функція $\psi(\vec{R}_1, \vec{R}_2, \dots, \vec{R}_N, \vec{r}_1, \vec{r}_2, \dots, \vec{r}_n, t)$, яка залежить від координат атомних ядер \vec{R}_N , що входять у молекулу,

та від координат електронів \vec{r}_n , які належать всій молекулі. Хвильова функція визначає імовірність того, що координати всіх атомних ядер та електронів мають певні значення, або іншими словами — імовірність того, що молекула знаходиться в певному стані. Стан відповідає положенням та характеру руху всіх складових частин молекули. Положення та швидкості руху будь-яких мікрооб'єктів можуть бути визначені лише з певною імовірністю. Густина імовірності n того, що всі ядра та електрони мають задані значення координат, виражається через хвильову функцію наступним чином:

$n = |\Psi|^2 = \Psi^* \Psi$. Імовірність того, що координати всіх складових частин молекули мають хоча б якесь значення, очевидно рівна 1. Тому маємо умову нормування хвильової функції:

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \Psi^*(\vec{R}_1, \dots, \vec{R}_N, \vec{r}_1, \dots, \vec{r}_n) \Psi(\vec{R}_1, \dots, \vec{R}_N, \vec{r}_1, \dots, \vec{r}_n) d\vec{R}_1 \dots d\vec{R}_N d\vec{r}_1 \dots d\vec{r}_n = 1$$

Фізичні характеристики, які можна спостерігати на експерименті, є усередненням по всім можливим значенням у заданому стані. Кожна фізична величина характеризується оператором, який діє на хвильові функції. Якщо фізична величина A має оператор \hat{A} , то значення величини A , яке буде спостерігатись на експерименті для молекули, що знаходиться у стані, який описується хвильовою функцією Ψ , буде рівне:

$$A = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \Psi^*(\vec{R}_1, \dots, \vec{R}_N, \vec{r}_1, \dots, \vec{r}_n) \hat{A} \Psi(\vec{R}_1, \dots, \vec{R}_N, \vec{r}_1, \dots, \vec{r}_n) d\vec{R}_1 \dots d\vec{R}_N d\vec{r}_1 \dots d\vec{r}_n = \langle \Psi^* | \hat{A} | \Psi \rangle$$

Оператори деяких фізичних величин наведено нижче (Табл. 7.1).

Таблиця 7.1 – Оператори деяких фізичних величин.

№	Фізична величина	Оператор
1	Координата, \vec{r}	$\hat{r} = \vec{r}$
2	Імпульс в напрямку координати \vec{r}	$\hat{p} = i\hbar\vec{\nabla} = i\hbar\frac{\partial}{\partial\vec{r}}$, де \hbar — стала Планка, i — уявна одиниця
3	Кінетична енергія	$\hat{T} = \frac{\hbar^2}{2m}\Delta^2 = \frac{\hbar^2}{2m}\left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right]$, де \hbar — стала Планка
4	Потенціальна енергія	$\hat{U} = U$, де U — класичне значення потенціальної енергії

Хвильові функції, які дозволяють обчислити середнє значення будь-якої фізичної величини, знаходяться з рівняння Шредінгера. Рівняння Шредінгера для стаціонарного стану (наприклад, коли середні значення фізичних характеристик не залежать від часу) має вигляд:

$$\hat{H}\Psi = E\Psi,$$

де $\hat{H} = \hat{T} + \hat{U}$ — оператор Гамільтона (Гамільтоніан, оператор повної енергії), а E — значення повної енергії системи. Розв'язок рівняння Шредінгера для стаціонарного стану являє собою дискретний набір значень енергії E_n , кожному значенню енергії відповідає хвильова функція Ψ_n — стан, у якому досягається дане значення енергії.

Всі методи квантової хімії дають можливість шляхом розв'язання рівняння Шредінгера для заданого набору і розташування атомів молекули знайти різні характеристики цієї молекули, такі як енергія, дипольний момент, тощо. Різні методи квантової хімії мають різну точність і відповідно різну швидкість обрахунку. Щоб з'ясувати основні фактори, які впливають на швидкість і точність обрахунку, а також успішно використовувати паралельне прикладне програмне забезпечення для квантово-хімічних розрахунків, потрібно мати основні уявлення про методи розв'язання квантово-хімічних задач.

Метод Хартрі-Фока

Розглянемо аспекти теорії Хартрі-Фока, яка є основою більшості квантово-хімічних методів [16]. У звичайних умовах молекули знаходяться в стані з мінімальною можливою енергією — в основному стані. Для того, щоб мати можливість знаходити всі фізичні характеристики в цьому стані, потрібно мати хвильову функцію цього стану. В загальному випадку задача розв'язання рівняння Шредінгера є складною. Теорія Хартрі-Фока дозволяє знайти значення енергії молекули у основному стані та хвильову функцію молекули у цьому стані.

Стан молекули описується за допомогою хвильової функції $\Psi(\vec{R}_1, \dots, \vec{R}_M, \vec{r}_1, \dots, \vec{r}_N)$, яка визначає імовірність того, що координати складових частин молекули (ядер та електронів) відповідно будуть $\vec{R}_1, \dots, \vec{R}_M, \vec{r}_1, \dots, \vec{r}_N$. Оскільки ми розраховуємо систему в стаціонарному стані, то для хвильових функцій справедливо стаціонарне рівняння Шредінгера (7.1).

$$H\Psi = E\Psi \quad (7.1)$$

де H — оператор Гамільтона (гамільтоніан) молекули, E — значення повної енергії системи. Для молекули в розглянутій моделі гамільтоніан рівний:

$$H = T + U = -\frac{\hbar^2}{2m} \sum_{i=1}^N \Delta_i - \frac{\hbar^2}{2} \sum_{k=1}^M \frac{\Delta_k}{M_k} - \sum_{k=1}^M \sum_{i=1}^N \frac{e^2 Z_k}{r_{ik}} + \sum_{k_1=1}^M \sum_{k_2 > k_1}^M \frac{e^2 Z_{k_1} Z_{k_2}}{R_{k_1 k_2}} + \sum_{i_1=1}^N \sum_{i_2 > i_1}^N \frac{e^2}{r_{i_1 i_2}} \quad (7.2)$$

де i, i_1, i_2 - нумерують електрони, k, k_1, k_2 - нумерують ядра, T — оператор кінетичної енергії, U — оператор потенціальної енергії, \hbar — стала Планка, m - маса електрона, M_k — маса k -го ядра, Δ_k — оператор Лапласа за координатами k -го ядра, Z_k — заряд k -го ядра, r_{ik} відстань від k -го ядра до i -го електрона, e — заряд електрона, $r_{i_1 i_2}$ — відстань між двома електронами, $R_{k_1 k_2}$ — відстань між двома ядрами. Знехтуємо кінетичною енергією ядер, порівняно з кінетичною енергією електронів (наближення Борна-Опенгеймера), перейдемо до системи атомних одиниць ($\hbar = 1, m = 1, e = 1$), а також врахуємо, що потенціальну енергію взаємодії ядер

$\sum_{k_1=1}^M \sum_{k_2>k_1}^M \frac{Z_{k_1} Z_{k_2}}{R_{k_1 k_2}}$ можна обрахувати незалежно і додати потім. З цих

міркувань отримуємо електронний гамільтоніан:

$$H_e = -\frac{1}{2} \sum_{i=1}^N \Delta_i - \sum_{k=1}^M \sum_{i=1}^N \frac{Z_k}{r_{ik}} + \sum_{i_1=1}^N \sum_{i_2>i_1}^N \frac{1}{r_{i_1 i_2}} \quad (7.3)$$

Розкладемо електронний гамільтоніан на одноелектронну частину, яка не залежить від взаємодії між електронами

$$h(r_i) = -\frac{\hbar^2}{2m} \Delta_i - \sum_{k=1}^M \frac{e^2 Z_k}{r_{ik}}, \text{ та двоелектронну частину, яка залежить}$$

$$\text{лише від взаємодії між електронами } \sum_{i_1=1}^N \sum_{i_2>i_1}^N \frac{1}{r_{i_1 i_2}}.$$

Хвильові функції молекули в наближенні Борна-Опенгеймера можна представити у вигляді $\Psi(\vec{r}_1, \dots, \vec{r}_N) X(\vec{R}_1, \dots, \vec{R}_M)$, оскільки ми вважаємо, що ядра нерухомі. При цьому множники $X(\vec{R}_1, \dots, \vec{R}_M)$ скорочуються при підстановці в (7.1).

Якщо в (7.1) використати одноелектронний Гамільтоніан, то ми отримаємо одноелектронні рівняння Шредінгера:

$$h_i \psi_i = \varepsilon_i \psi_i \quad (7.4)$$

де ψ_i — хвильова функція i -го електрона, ε_i — енергетичний спектр i -го електрона. Для розрахунку багатоелектронної хвильової функції використаємо наближення Хартрі, в якому хвильова функція шукається у вигляді лінійної комбінації добутків виду: $\Psi(\vec{r}_1, \dots, \vec{r}_N) = \psi_1(\vec{r}_1) \psi_2(\vec{r}_2) \dots \psi_N(\vec{r}_N)$, де $\psi_m(\vec{r}_k)$ означає, що k -й електрон перебуває в стані, що має одноелектронну хвильову функцію ψ_m , тобто вважається, що електрони між собою не взаємодіють. Враховуючи, що електрони молекули нічим один від одного не відрізняються, загальна хвильова функція має містити суму відповідних добутків з різними перестановками електронів по станах. Враховуючи принцип Паулі (при зміні двох електронів місцями хвильова функція має поміняти знак), а також умову нормування, загальний вигляд хвильової функції:

$$\Psi(\vec{r}_1, \dots, \vec{r}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\vec{r}_1) & \psi_1(\vec{r}_2) & \psi_1(\vec{r}_3) & \dots & \psi_1(\vec{r}_N) \\ \psi_2(\vec{r}_1) & \psi_2(\vec{r}_2) & \psi_2(\vec{r}_3) & \dots & \psi_2(\vec{r}_N) \\ \psi_3(\vec{r}_1) & \psi_3(\vec{r}_2) & \psi_3(\vec{r}_3) & \dots & \psi_3(\vec{r}_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_N(\vec{r}_1) & \psi_N(\vec{r}_2) & \psi_N(\vec{r}_3) & \dots & \psi_N(\vec{r}_N) \end{vmatrix} \quad (7.5)$$

Вираз (7.5) називається детермінантом Слетера.

Значення повної електронної енергії системи можна знайти, як середнє значення оператора Гамільтона системи (7.6):

$$E_0 = \int \Psi^*(\vec{r}_1, \dots, \vec{r}_N) H \Psi(\vec{r}_1, \dots, \vec{r}_N) d\vec{r}_1, \dots, d\vec{r}_N = \langle \Psi | H | \Psi \rangle$$

$$H_e = \sum_{i=1}^N h_i + \sum_{i_1=1}^N \sum_{i_2 > i_1}^N \frac{1}{r_{i_1 i_2}} = h + H^2 \quad (7.6)$$

В результаті підстановки (7.5) в (7.6), отримаємо:

$$E = E_1 + E_2 = \sum_{i=0}^N \langle i | h | i \rangle + \sum_{i=0}^N \sum_{j>i}^N \langle ij || ji \rangle \quad (7.7)$$

де $\langle i | h | i \rangle = \int \psi_i^*(\vec{r}) h \psi_i(\vec{r}) d\vec{r}$ — одноелектронні інтеграли,

$$\langle ij || kl \rangle = \int \psi_i^*(\vec{r}_1) \psi_j^*(\vec{r}_2) \frac{1}{r_{12}} \psi_k(\vec{r}_1) \psi_l(\vec{r}_2) d\vec{r}_1 d\vec{r}_2 —$$

двоелектронні інтеграли.

Розкладемо хвильові функції по деякому базису відомих функцій

χ_i :

$$\psi_k(\vec{r}) = \sum_i C_k^i \chi_i(\vec{r}) \quad (7.8)$$

При цьому задача знаходження хвильових функцій зводиться до знаходження коефіцієнтів розкладу C_k^i . В якості базису частіше всього вибирається хвильові функція для руху електрона в центральній симетричному полі, або близькі до них Гаусівські функції. Значення енергії виражені через інтеграли від базисних функцій:

$$E = \sum_{m,n}^{basis} D_{mn} h_{mn} + \sum_{m,n,p,q}^{basis} D_{mn} D_{pq} (2 \langle mn | pq \rangle - \langle mn | qp \rangle) \quad (7.9)$$

Сума береться по всім базисним функціям, де

$D_{mn} = \sum_{i=1}^N C^{*m}_i C^n_i$ - матриця густини, $h_{mn} = \langle m | h | n \rangle$. Вираз

(7.9) також враховує, що в стані з одним значенням енергії може знаходитись два електрона з протилежними значеннями спінів (метод RHF – restricted Hartree-Fock).

Для знаходження хвильових функцій в найнижчому стаціонарному стані знайдемо мінімальне значення електронної енергії при умові, що хвильові функції мають бути нормованими. Це еквівалентно наступній варіаційній проблемі:

$$\begin{aligned} & \sum_a^N \langle a | h | a \rangle + \frac{1}{2} \sum_{ab}^N [\langle ab | ab \rangle - \langle ab | ba \rangle] - \\ & - \sum_{ab}^N e_{ab} [\langle a | b \rangle - \delta_{ab}] = \min \end{aligned} \quad (7.10)$$

В результаті знаходження варіації (7.10) і прирівнювання її до нуля отримуємо рівняння Хартрі-Фока:

$$[h + \sum_b^N (J_b(\vec{r}_1) - K_b)] \varphi_a(\vec{r}_1) = \sum_b^N e_{ab} \varphi_b(\vec{r}_1) \quad (7.11)$$

де $J_b(\vec{r}_1) = \int d\vec{r}_2 \varphi_b^*(\vec{r}_2) \frac{1}{r_{12}} \varphi_b(\vec{r}_2)$ — кулонівський оператор,

$K_b \varphi_a(\vec{r}_1) = \varphi_b(\vec{r}_1) \int d\vec{r}_2 \varphi_b^*(\vec{r}_2) \frac{1}{r_{12}} \varphi_a(\vec{r}_2)$ — обмінний оператор,

$h + \sum_b^N (J_b(\vec{r}_1) - K_b)$ — оператор Фока.

В результаті використання розкладу (7.8) з (7.11) легко отримати рівняння самоузгодженого поля (Self Consistent Field, SCF) для коефіцієнтів розкладу C :

$$FC = eSC, \quad (7.12)$$

де $F_j^i = h_{ij} + \sum_{pq}^{basis} D_{pq} (2 \langle ip | qj \rangle - \langle ip | jq \rangle)$ — матриця Фока,

$S_j^i = \langle i | j \rangle$ — обмінна матриця,

$D_{ij} = \sum_a^N C_a^{*i} C_a^j$ — матриця густини.

Рівняння (7.12) є основними рівняннями для знаходження енергії та хвильових функції молекул за методом Хартрі –Фока. Це є узагальнена проблема власних значень, з розв'язку якої отримуються значення енергії кожного електрона — матриця e , та значення коефіцієнтів розкладу хвильової функції кожного електрона по базису — матриця C .

Алгоритм розв'язку (7.12) наступний:

1. Вибір початкового наближення матриці Фока у вигляді:

$$F_{ij}^{(0)} = \langle i | \sum_{m=1}^{occ} \left[-\frac{1}{2} \Delta_m + \sum_{k=1}^M \frac{Z_k}{r_{km}} \right] | j \rangle$$

2. Вибір базису.
3. Обрахунок одно- та двоелектронних інтегралів і збереження їх.
4. Обрахунок матриці перекриття.
5. Обрахунок матриці Фока.
6. Вирішення проблеми власних значень.
7. Обрахунок густини та енергії.
8. Повторити п. 5 до збіжності.

Таким чином, задачі квантової хімії зводяться до операцій з матрицями.

1. Обчислення елементів матриць.
2. Множення матриць.
3. Розв'язання проблеми власних значень.

Всі ці задачі, як було показано, добре розпаралелюються і ефективність розпаралелювання близька до ідеальної для матриць великої розмірності.

Теорія збурень Моллера–Пелессета (MP)

Метод Хартрі-Фока не враховує кореляції хвильових функцій електронів, оскільки вираз (7.5) виведений в припущенні, що електрони не взаємодіють між собою. Для розрахунку структур складних біологічних молекул неврахування кореляції може призвести до великих похибок обчислення. Врахувати “кореляцію електронів” можна на основі теорії збурень, суть якої полягає в наступному.

Нам потрібно розв'язати рівняння Шредінгера (7.1), гамільтоніан якого складається з двох частин: H_0 і λV . Для H_0 розв'язок рівняння відомий, V - збурення гамільтоніану, яке не дає можливості точно розв'язати рівняння. При цьому загальний гамільтоніан

$$H = H_0 + \lambda V \text{ - де } \lambda \text{ - малий параметр } 0 \leq \lambda \leq 1. \text{ При } \lambda = 0$$

рівняння розв'язується:

$$H_0 \Psi_0 = E_0 \Psi_0$$

Збурення впливає на енергію та хвильові функції.

$$E = E_0 + \lambda E_1 + \lambda^2 E_2 + \dots$$

$$\Psi = \Psi_0 + \lambda \Psi_1 + \lambda^2 \Psi_2 + \dots \quad (7.13)$$

Після підстановки (7.13) в (7.1) і прирівнювання коефіцієнтів при однакових степенях λ отримаємо рівняння:

$$E_{nk} = \langle \Psi_0^k | V | \Psi_{n-1}^k \rangle$$

$$\Psi_l^n = \sum_k C^{(l)n}_k \Psi_0^k$$

$$C^{(k)n}_l = \sum_{i \neq n}^k \frac{C^{(k-1)n}_i \langle \Psi_0^l | V | \Psi_0^i \rangle - \sum_{i=1}^k C^{(k-i)n}_i E_{in}}{E_{0n} - E_{0l}}, \quad (7.14)$$

які визначають нові наближення хвильових функцій і енергій через відомі наближення більш низького порядку.

В теорії збурень Моллера–Плессета в якості збурення вибирається різниця між оператором Фока і точним значенням гамільтонаіану

$$V = \sum_{i < j}^N \left[\frac{1}{r_{ij}} - \sum_p^N (J_p(i) - K_p(i)) \right].$$

В результаті наближення першого порядку співпадає з методом Хартрі-Фока, а наближення другого порядку (MP2) дозволяє знайти поправку до енергії та хвильових функцій:

$$\Psi_{1,n} = \sum_{a>b, y>z} C^{(1)}_{n,abyz} \Psi_{yz}^{ab},$$

Ψ_{yz}^{ab} - відповідає детермінанту Слетера, в якому стани (a, b) - зайняті в Ψ_0 і вільні в Ψ_1 , а стани (y, z) – вільні в Ψ_0 і зайняті в Ψ_1

$$C^{(1)}_{n,abyz} = \sum_{a>b, y>z} \frac{\langle \Psi_{yz}^{ab} | \Psi_{0n} \rangle}{\varepsilon_a + \varepsilon_b - \varepsilon_y - \varepsilon_z}$$

$$E_2 = \frac{1}{4} \sum_{abyz} \frac{|\langle ab | yz \rangle - \langle ab | zy \rangle|^2}{\varepsilon_a + \varepsilon_b - \varepsilon_y - \varepsilon_z} \quad (7.15)$$

Теорія функціоналу густини (DFT)

Ще один метод врахування кореляції електронів – метод (DFT). Ідея методу полягає в тому, щоб розв'язати рівняння Шредінгера для невзаємодіючих електронів, які знаходяться в деякому зовнішньому полі. При цьому потенціал зовнішнього поля підбирається так, щоб отримати одноелектронну густину імовірності для кожного електрона такою ж самою, як в реальній системі.

Електронна густина виражається в наступному вигляді:

$$n(\vec{r}) = \int \Psi^*(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n) \Psi(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n) d\vec{r}_1 d\vec{r}_2 \dots d\vec{r}_n \quad (7.16)$$

Згідно першої теореми Гохенберга-Кона в найнижчому енергетичному стані системи через електронну густину можна виразити значення будь-якої фізичної характеристики системи. Друга теорема Гохенберга-Кона доводить, що функціонал $F[n] = \min \langle \Psi | T + V_{ee} | \Psi \rangle$ (мінімум шукається по всім функціям, що дають задану густину $n(\vec{r})$), є універсальним для будь-якої системи. При цьому енергія системи шукається як $E = \min_n \langle F[n] + \int V_{ext}(r) n(r) dr \rangle$.

Практично для знаходження конформаційних та енергетичних характеристик методом DFT використовуються рівняння Кона-Шама:

$$(T + V_s) \psi_i = E_i \psi_i$$

$$n(r) = \sum_i^N |\psi_i(r)|^2 \quad (7.17)$$

$$V_s(r) = V_{ext}(r) + \int \frac{n(r')}{|r - r'|} dr' + V_{xc}(n, r)$$

де $V_{xc}(n, r)$ - обмінний член, $V_{ext}(r)$ - зовнішнє поле ядер. Для обмінного члена використовуються різні апроксимації.

Ці рівняння мають той самий вигляд, що і рівняння в методі Хартрі-Фока. Складність полягає в тому, що точний вигляд обмінного члена невідомий і доводиться користуватись різними апроксимаціями. При цьому вплив кореляції електронів враховується обмінним членом.

Базисні функції

Для квантово-хімічного розрахунку електронні хвильові функції розкладаються по певному вибраному базису. Базис вибирається з міркувань точності та швидкості. Чим більше базисних функцій, тим вища точність, але нижча швидкість та стійкість методу. В якості базисних функцій вибираються електронні хвильові функції, які отримуються в результаті розв'язку рівняння Шредінгера для атома водню, або близькі до них:

$$\chi_{l_x l_y l_z \xi}(\vec{r}, \vec{R}) = x^{l_x} y^{l_y} z^{l_z} r^{n-1} e^{-\xi r}.$$

Для кожного ядра вибираються свої базисні функції. Цей базисний набір називають базисом Слетера. Базисні функції відповідають різним станам електрона в атомі, наприклад, s, p, d, f і т.д.. Для спрощення розрахунків застосовується гаусовий базис:

$$\chi_{l_x l_y l_z \xi}(\vec{r}, \vec{R}) = x^{l_x} y^{l_y} z^{l_z} r^{2n-2-l} e^{-\xi r^2}.$$

Існують різні стандартні набори базисних функцій [17].

Характеристики квантово-хімічних методів

Теорія квантово-хімічного розрахунку є складною, тому підсумуємо основні аспекти. Квантово-хімічні методи дозволяють знайти наближений розв'язок рівняння Шредінгера для заданих типів атомів і заданого положення цих атомів у просторі.

Структура молекули задається за допомогою декартових координат або z-матриці (Рис. 7.8). Для побудови z-матриці вибирається один атом як основний. Для другого атома задається відстань від першого. Для третього атома необхідно задати відстань до одного з визначених атомів та кут між зв'язками. Для четвертого і всіх інших атомів потрібно задати три параметра: відстань до одного з відомих атомів, кут між зв'язками та двограний кут між площинами, в яких знаходяться атоми.

Z-матриця молекули (Рис. 7.8) має вигляд:

О							
С	1	R_{12}					
С	2	R_{23}	1	α_{123}			
Н	3	R_{34}	2	α_{234}	1	θ_{1234}	

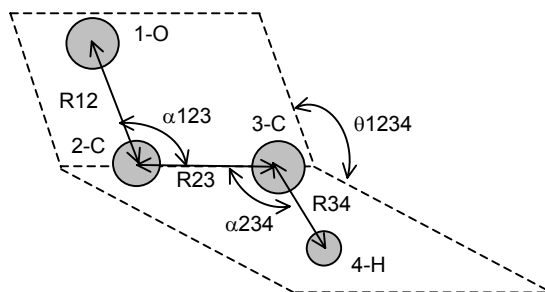


Рисунок 7.8 – Координати z-матриці.

У z-матриці всі просторові координати задаються, як правило в ангстремах, в кути — у градусах.

В результаті моделювання знаходиться енергія і хвильові функції всіх електронів молекули. Точність залежить від вибраного рівня теорії. Характеристики деяких методів наведено в табл. 7.2. Для розкладу електронних хвильових функцій необхідно вибрати базис, який дозволяє найточніше представити електронну конфігурацію. Основні характеристики базисів наведено в табл. 7.3.

Знаючи енергію та хвильові функції, є можливість знайти середні значення та зміни в часі різних характеристик, такі як спектральні частоти інфрачервоного, видимого та ультрафіолетового діапазону, швидкості руху атомів, значення спектрів ядерного магнітного резонансу та ін.. Деякі методи розрахунку фізичних параметрів, що часто застосовуються, наведено в табл. 7.4. Як правило, результати моделювання представляються у вигляді текстових файлів, які можуть бути візуалізовані спеціальним програмним забезпеченням.

Програми квантово-хімічного моделювання

Існує ряд широкоживаних програм для квантово-хімічного, напівемпіричного та напів-класичного моделювання молекулярних структур. У даній роботі буде використано програму gamess (General Atomic and Molecular Electronic Structure System) [18]. Ця програма сприймає вхідний текстовий файл, у якому задається структура молекули, системні параметри та параметри розрахунку. Текстовий файл повинен мати розширення **inp**, наприклад **H2O.inp**.

Приклад вхідного файла:

```
$CONTRL SCFTYP=RHF RUNTYP=HESSIAN COORD=ZMT $END
```

```

$SYSTEM TIMLIM=30 MEMORY=3000000 COREFL=.TRUE. $END
$BASIS GBASIS=N21 NGAUSS=3 $END
$GUESS GUESS=HUCKEL $END
$SCF DIRSCF=.TRUE. $END
$STATPT NSTEP=800 $END
$PCM SOLVNT=WATER $END
$DATA

c1
c
c 1 1.45000
o 2 1.40000 1 109.471
h 3 0.95000 2 109.471 1 180.000
h 2 1.08900 1 109.471 3 180.000
h 2 1.08900 3 109.471 5 120.000
h 1 1.08900 2 109.471 3 180.000
h 1 1.08900 2 109.471 6 120.000
h 1 1.08900 2 109.471 8 120.000
$END

```

Таблиця 7.2. Основні характеристики квантово-хімічних методів та наближень

Метод	Кількість операцій	Об'єм пам'яті	Характеристика
RHF	$\sim N^4$	$\sim N^3$	Метод Хартрі-Фока для замкнених електронних оболонок (restricted Hartree Fock). Не враховує кореляцію електронів. Дозволяє знайти енергію молекули та хвильові функції електронів. Є першим етапом для методів підвищеної точності. Застосовується для молекул, не застосовується для атомів та вільних радикалів.
ROHF	$\sim N^4$	$\sim N^3$	Метод Хартрі-Фока для незамкнених електронних оболонок (restricted Openshell Hartree Fock). Застосовується для атомів та вільних радикалів.
UHF	$\sim N^4$	$\sim N^3$	Метод Хартрі-Фока для будь-яких електронних оболонок (unrestricted Hartree Fock). Має гірші характеристики збіжності, ніж попередні методи

MP(k)	$\sim N^{3+k}$	$\sim N^4$	Поправки k-го порядку до енергії та хвильових функцій на основі теорії збурень Моллера-Плессета. Враховує кореляцію електронних хвильових функцій. Найчастіше застосовується метод другого порядку. Є одним з найточніших методів для знаходження енергії та похідних від енергії.
DFT	$\sim N^4$	$\sim N^3$	Метод на основі теорії функціонала густини. Дозволяє врахувати кореляцію електронних хвильових функцій. Точність залежить від правильності вибору апроксимації обмінного потенціалу.
CI	$\sim N^{6-7}$	$\sim N^{3-4}$	Метод конфігураційної взаємодії. Враховує кореляцію електронних хвильових функцій, може застосовуватись для збуджених станів, проте дуже ресурсоємний. Має багато різних варіантів і наближень, застосовується для точних розрахунків
CC	$\sim N^{6-7}$	$\sim N^{3-4}$	Метод зв'язаного кластеру (coupled cluster). Є одним з найточніших, проте дуже ресурсоємний. Застосовується для точних розрахунків.
MCSCF	$\sim N^{6-7}$	$\sim N^{3-4}$	Мультиконфігураційний SCF. Метод SCF, який однаково підходить як для основних, так і для збуджених станів.
ECP			Effective core potential — дозволяє враховувати лише валентні електрони, а вплив внутрішніх електронних оболонок замінюється ефективним потенціалом взаємодії.
MD/FF			Молекулярна динаміка та метод силових полів. Застосовуються напівкласичні наближення.

PCM			Модель неперервного середовища, що поляризується. Може використовуватись для моделювання молекулярних структур у розчинах
-----	--	--	---

Таблиця 7.3 – Характеристики деяких базисів.

Позначення	Характеристика
STO-(n)G	Слєтерівський базис, для якого n гаусових функцій використовуються для апроксимації однієї слєтерівської орбіталі. Наприклад, STO-2G — кожна слєтерівська орбіталь апроксимується двома гаусівськими функціями. Використовується для точних розрахунків енергії, проте не завжди точно передає розподіл заряду
N-kmlG	Звужений базисний набір. Використовується одна гаусівська функція з лінійної комбінації N примітивних гаусівських функцій для кожної внутрішньої оболонки і дві або три гаусівських функції з різними значеннями експонент, кожна з яких є лінійною комбінацією відповідно k, m, l примітивних гаусівських функцій — для зовнішньої оболонки. Наприклад, 6-31G — кожна внутрішня оболонка описується однією гаусівською функцією, яка є лінійною комбінацією з 6 примітивних гаусівських функцій. Кожний валентний електрон описується двома функціями, одна — з лінійної комбінації 2-х гаусівських функцій, а друга — з однієї. Застосовується для складних молекул для точнішої передачі розподілу заряду.
Наявність *	Поляризаційні функції. Додаються d-функції до p-елементів або p-функції до s-елементів. Наприклад 6-311*G — додаються d-функції.
Наявність +	Дифузні функції. Додаються s та p функції з повільним затуханням для валентних електронів. Наприклад 6-31+G. Застосовується для опису водневих та інших слабких зв'язків.
Наявність (ndf,mpd)	Додаються поляризаційні функції n – df типу, та m — pd типу.

Таблиця 7.4 – Методи розрахунку фізичних характеристик.

Метод	Призначення
ENERGY	Розрахунок енергії молекули E . Необхідно задати геометрію молекули, рівень теорії і базис.
GRADIENT	Розрахунок сил $\vec{f} = \text{grad}E$, що діють на ядра атомів.
OPTIMIZE	Знаходження оптимальної геометрії молекули, що відповідає локальному мінімуму енергії $\text{grad}E \rightarrow 0$. Розрахунок проводиться ітеративно $\vec{R}_i^{\text{new}} = \vec{R}_i^{\text{old}} + \alpha \vec{f}_i(\vec{R}_i^{\text{old}})$, де α — малий параметр.
SADPOINT	Знаходження сідлової точки (активаційного бар'єру), що відповідає максимуму енергії. Принцип розрахунку той самий, що і для оптимізації геометрії, проте перевірка наявності максимуму вимагає додаткових затрат.
IRC	Внутрішні координати реакції. Знаходження кривої у фазовому просторі, через яку піде хімічна реакція.
HESSIAN	Розрахунок частот та інтенсивностей інфрачервоних спектрів молекули шляхом діагоналізації матриці других похідних від енергії.
VSCF	Розрахунок ангармонічних поправок до коливальних спектрів.
RAMAN	Розрахунок інтенсивності комбінаційного розсіювання.
NMR	Розрахунок тензора хімічного екранування.
TRANSITN	Розрахунок перехідних моментів та сил осциляторів для спин-орбітальної взаємодії.

Для виконання роботи важливими є параметри, що підкреслені, всі інші можна використовувати без зміни. Всі рядки мають починатись з пробілу.

SCFTYP=RHF Рівень теорії для розв'язання SCF рівнянь. У даному випадку — restricted Hartree-Fock

RUNTYP=HESSIAN Фізичні параметри, що розраховуються.
HESSIAN — коливальні спектри
OPTIMIZE — оптимізація геометрії

COORD=ZMT Тип координат, у даному випадку z-матриця

\$BASIS GBASIS=N21 Базис, у даному випадку 3-21G

NGAUSS=3 \$END

`$PCM SOLVNT=WATER` Якщо група вказана, то використовується модель розчинника, як неперервного середовища, якщо ні, то розрахунок проводиться у вакуумі. В даному прикладі розчинник — вода.

`$DATA` Містить координати молекули. C1 — група симетрії молекули. Після ключового слова `$DATA`

c1

...

`$END` має бути пустий рядок або рядок з назвою вашого розрахунку.

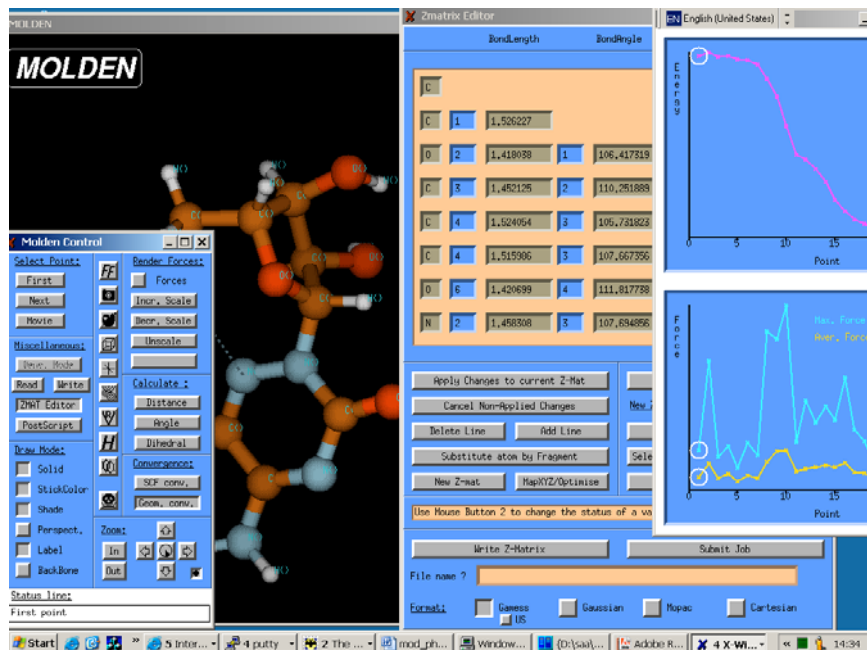


Рисунок 7.9 – Інтерфейс програми molden.

Запускається програма за допомогою команди:

`startgamess <ім'я вхідного файлу без розширення>`

Наприклад,

`startgamess H2O`

Результати розрахунку програма видає на стандартний вивід, тому їх необхідно перенаправити у файл, наприклад:

`startgamess H2O > H2O.results 2>&1.`

У цьому випадку всі результати будуть направлені у файл **H2O.results**.

Для того, щоб слідкувати за процесом розрахунку, можна скористатись командою:

```
tail -f <ім'я файла> ,
```

наприклад:

```
tail -f H2O.results .
```

Для створення z-матриці і візуалізації результатів можна скористатись програмою **molden** [19] (Рис. 7.9). Програма запускається за допомогою команди **molden3.7.linux**.

Для відкриття файлу z-матриці або вихідного файлу програми **gamess** служить діалог **read** на панелі "**Molden Controls**". Для редагування та створення z-матриці служить діалог "**Zmat editor**", у якому за допомогою миші можна вказувати типи атомів та їх положення. Для запису потрібно вказати ім'я файлу у полі "**File name**" і перед записом встановити параметри **Format=Gamess US** і поле **Set Status on All Variables** у значення **constant**.

Все інше обладнання співпадає з обладнанням заняття №1 "Оцінка продуктивності обчислювальних систем".

Завдання

1. Створіть середовище для роботи по аналогії з завданням заняття №1 "Оцінка продуктивності обчислювальних систем".
2. За допомогою програми **molden** створіть z-матрицю простої молекулярної структури, вказаної викладачем. Для скорочення часу кількість атомів потрібно вибрати не більше 10 і бажано, щоб вони були легкими (C, N, H, O і т.д.). Для потужної машини можна створити складнішу молекулу.
3. За допомогою програми **gamess** виконайте оптимізацію геометрії вашої молекулярної структури і візуалізуйте її за допомогою програми **molden**. Порівняйте оптимізовану геометрію з початковою. Для скорочення часу варто вибрати невеликий базис (наприклад, 3-21G) та рівень теорії (наприклад, RHF).
4. Оцініть час обрахунку енергії для вашої молекулярної структури і вашої обчислювальної системи та необхідний об'єм пам'яті і порівняйте оцінений результат з отриманим.
5. Знаючи оптимізовану геометрію молекули, розрахуйте коливальні спектри вашої молекули за допомогою програми **gamess** і візуалізуйте розраховані коливальні моди за допомогою програми **molden**.
6. Повторіть пункти 2, 5 для моделі PCM розчинника, вказаного

викладачем, і порівняйте результати моделювання в розчині і без нього.

7. Дайте відповідь на контрольні запитання.

Контрольні запитання

1. Які фізичні ефекти враховуються при квантово-хімічному моделюванні?
2. Скільки операцій потрібно для розрахунку енергії вашої молекулярної структури, який необхідний об'єм пам'яті і які оцінки часу розрахунку?
3. На скільки оцінені значення часу і об'єму пам'яті відрізняються від експериментальних і чому?
4. На скільки відрізняється геометрія вашої молекули у розчині та у вакуумі, чому?
5. На скільки відрізняються коливальні спектри вашої молекули у вакуумі та у розчині, чому?

Практичне заняття № 8.

Моделювання нейронних мереж

Мета: практично ознайомитися з модельними штучними нейронними мережами, оцінити характеристики ефективності мереж та отримати навички вибору структури мереж для різних типів задач розпізнавання та аналізу даних.

Штучні нейронні мережі

Загальні відомості: В загальному випадку нейронною мережею називають деяку сукупність нервових клітин живого організму, які виконують функцію сприйняття та обробки інформації, яка поступає з навколишнього середовища і внутрішніх систем організму, та генерують нервові імпульси для керування цими системами [20].

Штучні нейронні мережі є деякими спрощеними моделями біологічних нейронних мереж, але, незважаючи на значні спрощення, за допомогою нейронних мереж досягнуті значні успіхи. По-перше, вони допомагають у дослідженні головного мозку, є потужним інструментом перевірки гіпотез стосовно його діяльності. По-друге, вони використовуються для вирішення широкого класу задач розпізнавання образів, статистичного аналізу і прийняття рішень. Найбільш типовими задачами для нейронних мереж є [20, 21, 22]:

- апроксимація функцій по набору точок,
- класифікація даних по заданому набору класів, кластеризація даних і виявлення заздалегідь невідомих класів-прототипів,
- асоціативна пам'ять і стискання інформації,
- відновлення втрачених, зашумлених і неповних даних,
- оптимізація та оптимальне керування,
- передбачення часових рядів і прогнозування.

Цей список можна продовжувати і далі. Однак, ці, здавалось би, зовсім різні задачі, насправді дуже близькі. Розглянемо базові складові біологічного нейрону, а також принцип його функціонування, а потім проведемо порівняння із штучним нейроном.

Біологічний нейрон складається із: ядра (соми), декількох тисяч входів, що єднаються з ядром (дендритів і синапсів), вихідного аксона [23]. Нейрон проводить інформацію із входу на вихід і при цьому виконує певні операції:

1. Додає сигнали із вхідних аксонів, вхідний сигнал підсилює або послаблює за рахунок синаптичного контакту і його біохімічної провідності.

2. Вирішує як, і чи буде, представлена інформація на вихідному аксоні.

Механізм передачі та обробки інформації здійснюється за рахунок зміни біохімічної провідності, яка викликана інжекцією нейромедіаторів, щільними контактами, зміною біохімічної проникності мембрани за рахунок різних типів мембранних каналів. Штучний нейрон — модель, що частинам біологічного нейрону зіставляє у відповідність математичні операції і містить такі частини: елемент обробки (ядро), який складається з двох частин — суматора і блока прийняття рішення; вхідні ваги — синапси і аксони (рис. 8.1).

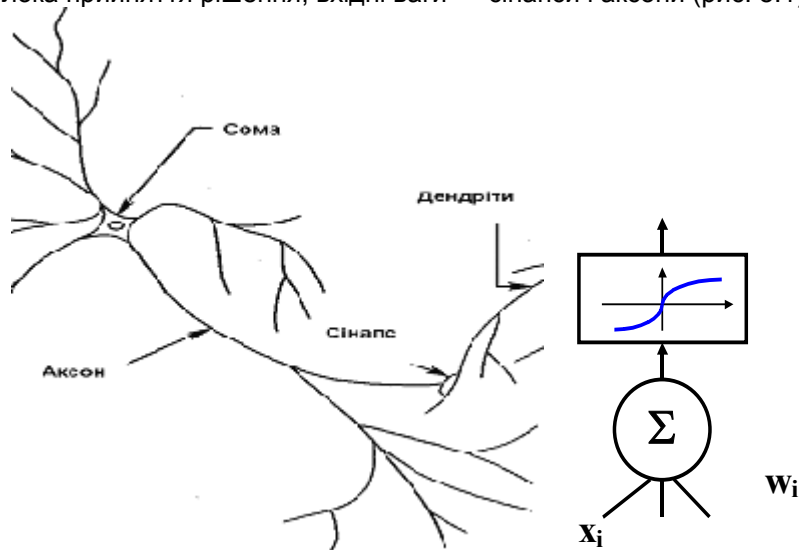


Рисунок 8.1 – Штучний нейрон містить суматор і блок прийняття рішення.

В моделі штучного нейрону об'єднання N входів переходить в операцію додавання, яка розраховує зважену суму вхідних сигналів:

$$I_j = \sum_{i=0}^{n-1} w_{ij} x_i, \quad (8.1)$$

де w_{ij} — ваговий коефіцієнт зв'язку між i -м нейроном попереднього шару і j -м вузлом даного шару, x_i — сигнал від i -го нейрону попереднього шару, I_j — зважена сума сигналів від попереднього

шару нейронів. Прийняття рішення моделюється нелінійною операцією:

$$Y_j = f(I_j, \theta) \quad (8.2)$$

де θ — величина порога спрацювання, $f(I_j, \theta)$ — так звана функція нелінійності або перетворення, яка в стандартних термінах нейронних мереж має назву функції прийняття рішення. Вигляд даної функції для випадку, коли $\theta = 1$, показано на рис. 8.2 [24].

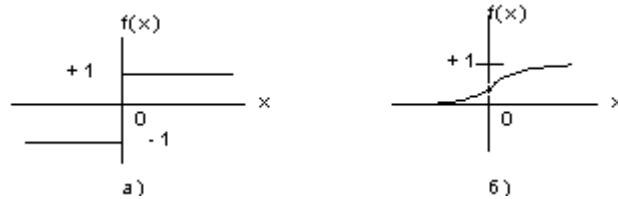


Рисунок 8.2 – Функції нелінійності: а) порогова; б) функція Фермі.

Від того, яка функція перетворення обрана для конкретної мережі, залежать принципи функціонування самої мережі, а також методи створення конгломерації нейронів, а тому принципи і процедури ініціалізації і навчання мережі. Так, для порогової функції, яка відповідає бінарній або тригерній архітектурі мережі, створені дані цілком природно подавати у бінарному вигляді, для використання аналогових сигналів є необхідним певне кодування інформації для представлення даних на вхід мережі. Слід зауважити, що для аналітичного дослідження мереж використовують сігмоїдальну функцію перетворення, тому що в цьому випадку спрощується процес пошуку точного аналітичного рішення мережі за рахунок гладкості та неперервності функції.

Таким чином, обравши тип елементарних комірок-нейронів і вид функції перетворення, можна побудувати найпростішу нейронну мережу певної архітектури, що є об'єднанням нейронів по типу, вказаному на рис. 8.3, або більш складному. Крім того, існують мережі, що самі будують себе під час процесу навчання.

Штучні нейронні мережі та методи моделювання поведінки біологічних мереж

Організму вцілому, окремим його системам і органам притаманна властивість самоорганізації [23], крім того, більшість систем

організму є розподіленими і складаються із однакових або близьких за функціями елементів, наприклад, в нервовій системі — нейрони, кровоносній — судини, в дихальній системі, а саме легенях, — альвеоли і т. ін.. Тому можна застосовувати такий опис, який розглянуто в [25], для таких систем. Нехай система, що складається із близьких за структурою чи однакових елементів, описується вектором:

$$\vec{\eta} = \vec{\eta} \{ \vec{a}_n, \vec{\tau} \} \quad (8.3)$$

де \vec{a}_n — вектор стану елементу, $\vec{\tau}$ — матриця зв'язків між елементами.

Об'єднання підсистем однієї системи чи систем, наприклад, організм людини, можна описати гіпервектором:

$$\vec{\Lambda} = \vec{\Lambda} \{ \vec{\eta}, \vec{\zeta}, \vec{\sigma} \dots \} \quad (8.4)$$

$\vec{\eta}, \vec{\zeta}, \vec{\sigma} \dots$ — параметри-вектори типу (8.3), що відносяться до різноманітних систем організму і описують деякі їх властивості — фізичні, хімічні, електричні і т. ін., або самі системи. Подібний підхід лежить в основі ідеї моделювання нервової системи і нервових шляхів людини — нейронних мереж.

Нервова система людини — розподілена, самоорганізуюча, адаптивна система з паралельною обробкою даних. Більшість існуючих алгоритмів і моделей в основному відображають певні аспекти і властивості нервової системи.

Моделі нейронних мереж — це алгоритми для інтелектуальних завдань, таких як навчання і оптимізація, ідеї яких, в деякому тлумаченні, взяті із досліджень природи головного мозку людини. В термінах математики нейронна мережа — це модель, що описується спрямованим графом із наступними властивостями [21, 26]:

1. Є змінна стану x_i , асоційована з кожним вузлом i (вихідний сигнал).
2. Є дійсна змінна, вага w_{ij} , асоційована з кожним зв'язком (ij) між двома вузлами i і j .
3. Є дійсна змінна, поріг θ_i , асоційована з кожним вузлом за номером i .

4. Задається функція перетворення $f_i = [n_i, w_{ij}, \theta_i, (j \neq i)]$ для кожного вузла i , що визначає стан вузла в залежності від порогу, вагових коефіцієнтів зв'язків і стану вузлів.

В стандартних термінах вузли називаються нейронами, зв'язки

називаються синапсами, а поріг — порогом активації. Функція перетворення звичайно має вигляд:

$$f\left(\sum_i w_{ij} \cdot x_i - \theta_j\right), \quad (8.5)$$

де $f(x)$ або дискретно-крокова, або гладка функція сигмоїдальна функція.

Вузли без вхідних зв'язків називаються вхідними нейронами, вузли без вихідних зв'язків — вихідними нейронами. Мережею нормальної топології — прямого і однонаправленого типу — називають мережу, що не містить замкнутих шляхів і передає дані в одному напрямку (рис. 8.3).

При об'єднанні в мережу нейронів-елементів система виявляє такі властивості [26]: спроможність навчатись; дані розподілені по всій системі (в вигляді вагових коефіцієнтів), отже, система є відмовостійкою — помилка в одному нейроні не призводить до руйнування всієї системи; в мережі відбиті асоціативні властивості, тобто невідомий вхідний об'єкт розподіляється між асоціативно промаркованими навчаючими прикладами; звідси — можливість використання для класифікації; можливість розпізнавання образів чи даних, які містять шум або неповні дані [27].

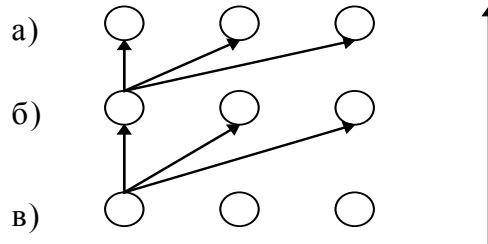


Рисунок 8.3 – Багатошарова нейронна мережа: а) вихідні вузли, б) прихований шар, в) вхідні вузли.

Далі слідує процес навчання, що можуть багаторазово повторюватися, і починається, власне, застосування мережі — її робота.

Одна із найважливіших особливостей мереж — можливість застосування різноманітних алгоритмів роботи і навчання відповідно до існуючого завдання. Алгоритми, за якими будуються нейронні мережі, звичайно поділяють на два основних класи [21–24]:

1. Адаптивні — коли всі вагові коефіцієнти системи адаптуються

на кожному кроці, наприклад, розраховується вихідний сигнал і порівнюється із відомим по навчанню результатом, помилка зменшується повільною адаптацією всіх вагових коефіцієнтів.

2. Конкурентні — коли збільшуються ваги одного чи більше “нейронів-переможців”. Для певного набору вхідних об’єктів активними є певні скупчення нейронів (кластери), що відображають вхідні класи об’єктів.

Алгоритми навчання залежно від принципу роботи звичайно поділяють наступним чином [26]:

1. Контрольовані — на кожній ітерації вихідний сигнал нейронної мережі порівнюється із відомим (по навчанню на попередньому циклі) і використовується для спостереження за процесом.

2. Неконтрольовані (самоорганізуючі) — навчання спонтанно направляється навчаючим алгоритмом.

Треба зауважити, що алгоритми навчання і роботи постали із відомих в анатомії і фізіології фактів і, частково, із здогадів про принципи роботи головного мозку, а також із досліджень молекулярної нейрофізіології. Нейронні мережі знаходять все більш широке застосування саме завдяки тому, що дозволяють успішно вирішувати неформалізовані завдання. Такі системи використовуються для моделювання задач, що не мають формальних алгоритмів рішення. На даний момент застосування мереж обмежено одним із серйозних недоліків — нейронні мережі є “чорним ящиком”. Як змінюються ваги вхідних сигналів і що за структури створюються — це залишається невідомим для більшості випадків. Але і цей недолік цілком можливо усунути, і системи можуть працювати в контакт з експертами та проводити внутрішню перебудову після узгодження.

Штучні контрольовані нейронні мережі на основі персептрону

Існує декілька основних ознак, за якими обираються алгоритми для різноманітних завдань медичної практики. По-перше, якщо потрібна класифікація за деякими параметрами, а не за самим зображенням, обираються алгоритми, які не потребують багато ресурсів обчислювальної техніки (розміри мережі досить малі), тобто щоб програми не використовували надмірно потужної техніки. Бажано також, щоб алгоритм використовував самоорганізацію, тому що зараз існують тільки емпіричні методи вибору кількості вузлів та порогових значень. По-друге, вимагається, щоб алгоритм був збіжним і похибка при цьому повинна бути якомога меншою. По-третє, необхідно, щоб існував метод декодування та інтерпретації отриманої від нейронної мережі інформації.

Найчастіше в медицині використовуються декілька класичних алгоритмів, такі як перцептрон, багатошаровий перцептрон, алгоритм Кохонена, а також модифіковані варіанти алгоритму навчання прямого розвитку, або їх різноманітні версії. Найбільш простим та поширеним алгоритмом є модель перцептрона (від лат. *perceptio* — сприйняття), який, незважаючи на простоту, є дуже потужним і перцептроноподібні мережі можуть вирішувати різноманітні задачі.

Мережа на основі одношарового перцептрону

Це один із найбільш простих алгоритмів нейронних мереж. Існує багато різноманітних його варіацій, об'єднаних загальною ідеєю. Наприклад, вагові коефіцієнти та порогові значення можуть бути як фіксованими, так і змінюватися в процесі роботи мережі. Розглянемо алгоритм перцептрона [20], який найбільш часто використовується в прикладних технічних задачах. Він відноситься до нейронних мереж з прямим розповсюдженням інформації, без зворотних зв'язків. Основою його є декілька шарів нейронів — нелінійних суматорів, які є моделлю пірамідальних нейронів кори головного мозку. Структура мережі і функція прийняття рішення наведена на рис. 8.4.

Кожен i -й вхід з входів x_1, \dots, x_n приймається з відповідним ваговим значенням w_i , так що їх зважена сума буде приймати вигляд (8.1). Для мережі вводять також порогове значення збудження θ для кожного нейрона і функцію активації.

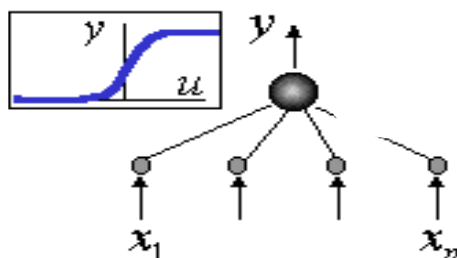


Рисунок 8.4 – Одношарова перцептронна мережа і функція прийняття рішення.

Функцію обирають, виходячи з конкретної задачі. Звичайно

використовують основні типи функцій, які наведено вище. Для персептрона характерне пошарове включення нейронів. В залежності від кількості шарів розрізняють одношарові і багатшарові персептрони.

Одношаровим персептроном є просте об'єднання нейронів, коли на кожен з нейронів подається одне і те ж вхідне значення. В задачах класифікації кількість нейронів визначає кількість класів, а кількість входів — розмірність даних, які потрібно класифікувати.

Для навчання персептронів найчастіше використовують алгоритм градієнтного навчання. На вхід подають навчаючі об'єкти, для кожного прикладу задається бажане значення виходів мережі. Далі можна порівняти дійсні і бажані значення виходів і ввести функцію помилки, яка в загальному випадку залежить від вагових коефіцієнтів, порогового значення і функції активації кожного нейрона:

$$E = E(\mathbf{w}, \vec{\theta}, f(u)), \text{ де } \vec{\theta} = (\theta_1, \theta_2, \dots, \theta_n), \text{ і}$$

$$\mathbf{w} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix} \quad (8.6)$$

В (8.6) як w_{ij} позначено ваговий коефіцієнт, який пов'язаний з j -м входом i -го нейрона. Часто в прикладних застосуваннях поріг збудження не змінюють в процесі навчання, а змінюють лише синаптичні ваги. Функцію похибки вводять як квадрат відстані від бажаного до реального виходу в багатовимірному гіперпросторі ознак (8.5).

В градієнтному навчанні мінімізація функції похибки досягається зміною величини вагових коефіцієнтів проти градієнту похибки. Таким чином забезпечується найшвидший спуск її в мінімум. В ідеальному випадку це — глобальний мінімум, але найчастіше знаходиться деяке субоптимальне рішення — в локальному мінімумі (рис. 8.5) [21]. Взагалі кажучи, наведена проблема локального мінімуму притаманна практично будь-якому алгоритмові нейронної мережі і може бути вирішена шляхом подальшого ускладнення алгоритму, архітектури мережі, або процедури навчання.

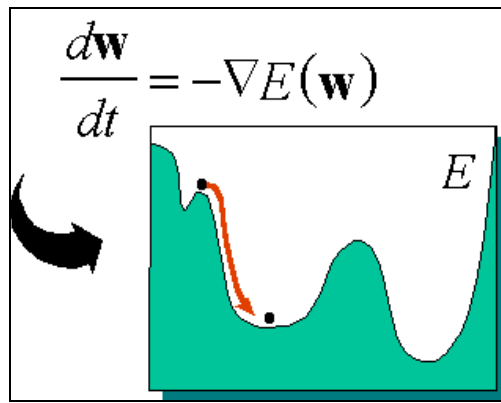


Рисунок 8.5 – Глобальний і локальний мінімум рішення мережі.

Алгоритм мережі персептрона

Припустимо, що є набір даних, які містять вектори двох класів, яким потрібно зіставити у відповідність два рівня вихідного сигналу мережі: -1 і +1, тобто класи А і Б. Необхідно виконати навчання мережі.

Крок 1. Виконується ініціалізація вагових коефіцієнтів і функції порога, тобто задаються ваги $w_i(0)$ і θ — малі випадкові числа. $w_i(t)$ — ваговий коефіцієнт i -го входу в момент часу t , θ — поріг спрацювання.

Крок 2. Введення невідомого об'єкту і бажаного вихідного результату. На вхід мережі послідовно подаються дані, тобто вектор вхідних значень $\vec{x} = (x_1, \dots, x_N)$, а також фіксується відповідний до вхідного вихідний сигнал $d(t)$ і порівнюється з бажаним.

Крок 3. Обчислення дійсних вихідних сигналів проводиться за формулою:
$$y(t) = f_n \left(\sum_{i=0}^{N-1} w_i(t) x_i(t) - \theta \right) \quad (8.7)$$

Крок 4. Далі виконується адаптація вагових коефіцієнтів.

$$w_i(t+1) = w_i(t) + \eta(d(t) - y(t)), \quad (8.8)$$

де $0 \leq i \leq N$ — кількість компонентів вхідного вектора,

$d(t) = \begin{cases} +1, \text{ клас } A \\ -1, \text{ клас } B \end{cases}$, η — довільний малий параметр ($0 < \eta < 1$) і $d(t)$ — бажаний вихід мережі. Вагові коефіцієнти запам'ятовуються, якщо

отримано необхідну точність, або на наступній ітерації вагові коефіцієнти не змінюються.

Крок 5. Повторення. (повернення на крок 2.).

Одношаровий персептрон може бути використаний для побудови швидкодіючої невеликої мережі з прозорою архітектурою, але він має декілька суттєвих недоліків, що обмежує його застосування. По-перше, він може бути використаний тільки для вузького кола завдань, тому що завдяки певній структурі він не може виконувати деякі логічні операції, наприклад, булевське АБО. По-друге, не гарантована збіжність мережі, тобто отримання стійкого рішення. Цілком природним є розвиток ідеї персептрону до багатошарової мережі, яка не має таких недоліків.

Нейронна мережа на основі багатошарового персептрону

Ця мережа близька за своєю ідеологією до алгоритму персептрона, але має декілька шарів між входом і виходом. Така мережа не має недоліків одношарового персептрону. Області в просторі рішень, які отримані за допомогою такого алгоритму, обмежені ламаними, а не гладенькими кривими (це відбувається за рахунок того, що області рішень — напівплощини — кожного вхідного вузла перетинаються, тому число сторін багатокутника відповідає числу вхідних вузлів). Як показано в [23], такі мережі можуть розпізнавати змішані області і для цього цілком достатньо трьох шарів нейронів.

Багатошаровим персептроном називають структуру з декількох шарів нейронів (рис. 8.6), в якій кожний шар побудовано з персептронів. Кожен вхід наступного шару з'єднаний з усіма виходами попередніх шарів.

Алгоритм мережі багатошарового персептрону

Це ітеративний градієнтний алгоритм з мінімальним квадратичним відхиленням між вхідними даними і потрібним виходом [25]. При цьому як функція порога використовується сігмоїд в такому вигляді:

$$g(x) = \frac{1}{1 + e^{-Ax}}, \quad (8.9)$$

де A — константа.

Крок 1: ініціалізація. Задаються значення всіх вагових коефіцієнтів як малі випадкові величини.

Крок 2. Представлення мережі невідомого об'єкту і бажаного вихідного результату. На цьому кроці виконується введення на вхід нових даних, які змінюються \vec{x} разом з бажаним вихідним сигналом $d(t)$.

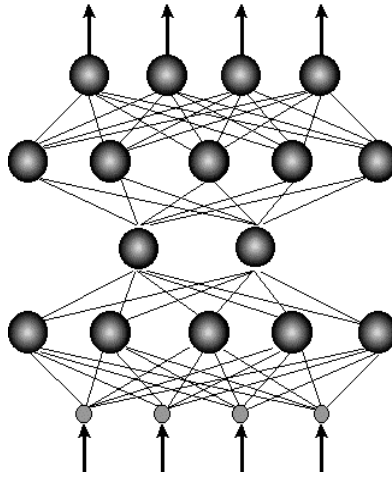


Рисунок 8.6 – Структура мережі багатошарового персептрону.

Крок 3. Обчислення вихідних значень проводиться за формулою:

$$y_i(t) = f \left(\sum_{i=0}^N w_i(t)x_i(t) - \theta \right), \quad (8.10)$$

де N — кількість вхідних даних.

Крок 4. Адаптація вагових коефіцієнтів. Рекурсивно, починаючи із вихідних вузлів і закінчуючи першим шаром, адаптуються ваги за формулою (8.11):

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \eta \delta_j x_i + \alpha (w_{ij}(t) - w_{ij}(t-1)), \\ \delta_j &= y_j(1 - y_j)(d_j - y_j), \end{aligned} \quad (8.11)$$

де $w_{ij}(t)$ — вага між i вузлом попереднього шару та вузлом j , η, α — малі параметри, δ_j — похибка для вузла j , d_j — очікуваний вихідний сигнал вузла j , а y_j — розрахований сигнал.

Якщо вузол j лежить глибше, ніж на один шар, то похибка може бути розрахована наступним чином:

$$\delta_j = x_j(1 - x_j) \sum_k \delta_k w_{jk}, \quad (8.12)$$

де сума по k — сума по всіх вузлах. Алгоритм має швидку збіжність, якщо вагові коефіцієнти розраховуються за емпіричною формулою:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i(t) - w_{ij}(t)) \quad (8.13)$$

де $0 < \eta < 1$ — малий параметр. В такому записі формула адаптації вагових коефіцієнтів нагадує мережу Кохонена.

Крок 5. Повернення на крок 2.

Алгоритм персептрону є простим і потужним методом вирішення різноманітних завдань класифікації, аналізу та розпізнавання даних, але має декілька суттєвих недоліків, які ускладнюють розробку нейросистем. По-перше, під час створення мережі існують тільки емпіричні правила вибору архітектури мережі, тому потрібно розробляти алгоритм навчання мережі, який гарантує збіжність рішення та стійку поведінку системи. По-друге, існує проблема кодування та інтерпретації даних та отриманого результату роботи мережі. Найбільш складним є декодування отриманого рішення мережі, яке також вирішується за допомогою специфічних алгоритмів навчання. По-третє, рішення мережі (інакше кажучи, рішення системи рівнянь, що відповідають мережі) може опинитися в локальному мінімумі замість глобального, як наслідок, в результаті ми отримаємо часткове рішення системи і тому нейронна мережа буде генерувати значну похибку при розпізнаванні невідомих даних. Перші дві проблеми відомі як визначення правил прийняття рішення і можуть бути вирішені шляхом створення самоорганізуючої мережі, де такі недоліки усунуті за рахунок самої ідеології та архітектури мережі. Третя із вказаних проблем може бути вирішена тільки шляхом створення складного алгоритму навчання мережі, який гарантує збіжність рішення до головного максимуму.

Методи навчання штучних нейронних мереж

Існує декілька класичних ідеологій навчання нейронних мереж, більшість з них базується на методиці зворотного розповсюдження похибки, але для отримання функції відображення вхідного простору в деякі об'єктні класи кращі результати дають алгоритми, що базуються на додаванні нейронів до скритого шару. В самому простішому випадку нейронної мережі, одношаровому персептрону, взагалі не потрібно ніякої процедури навчання. Достатньо побудувати (на базі логічних елементів, мікросхем, або шляхом створення програмного продукту) одношаровий персептрон і представити відомі дані на вхід для того, щоб сформувати вагові коефіцієнти. При подачі наступних даних мережа буде мати змогу їх аналізувати або розпізнавати. Але така мережа може бути використана тільки для певного дуже вузького класу задач, крім того, доведено, що одношаровий персептрон значно обмежений за

класом завдань, які він може вирішити таким чином. Цілком аналогічна ситуація для багат шарового персептрону — після ініціалізації мережі та подачі даних на вхід він може аналізувати інформацію, але ймовірність того, що він невірно розпізнає вхідні дані або на виході не буде отримано взагалі ніякого рішення, велика. Тому після створення та ініціалізації мережі, для підвищення ймовірності вірного рішення, гарантованості його збіжності, підвищення швидкодії та стійкості нейронні мережі потребують навчання.

Для персептроноподібних алгоритмів найбільш розповсюдженим є метод “зворотного розповсюдження” Поля Вербоза (*back-propagation*) [26].

Ідея алгоритму полягає в тому, що якщо ми маємо звичайну мережу, яка побудована за принципом персептрону і подаємо на вхід деякі дані, то вихідний сигнал розраховується за формулою звичайної нейронної мережі:

$$I_{ij} = f \left(\sum_i (I_{ij-1} w_{ij}, \theta) \right), \quad (8.14)$$

де I_{ij} — сигнал нейронів наступного шару, I_{ij-1} — вхідні сигнали нейронів попереднього шару, а w_{ij} — вагові коефіцієнти зв'язків між шарами j і $j-1$. Індекс j є номером шару. Якщо для створеної архітектури мережі ми подамо відомі дані і зафіксуємо вихідні значення I_{ij+1} , тобто так званий “бажаний” вихід мережі, або відомий результат, то при подальшому просуванні даних крізь мережу під час фази навчання для кожного циклу проходження даних, які розраховані за формулою (8.14), на виході проводиться порівняння отриманого результату за формулою:

$$I_{ij} - \sum_i f(I_{ij-1} w_{ij}, \theta) < \varepsilon \quad (8.15)$$

де ε — апіорі задана точність, i — індекс нейрона певного шару j . Після першої ітерації необхідна точність не буде отримана, тому починаючи від вихідного шару, починають змінюватись вагові коефіцієнти w_{ij} за формулою:

$$w_{ij} = w_{ij} \Delta_j + w_{ij} \quad (8.16)$$

де Δ_j — параметр зміни вагових коефіцієнтів, так званий градієнт зміни міжнейронних зв'язків, який на кожному циклі переходу від наступного шару до попереднього, тобто в зворотному напрямку, розраховується наступним чином:

$$\Delta_j = \alpha (I_j - \sum_j (I_{ij-1} w_{ij})) \quad (8.17)$$

α — малий постійний коефіцієнт. Таким чином, після того як дані пройшли крізь мережу і проведено порівняння з апіорі заданим вихідним значенням, проводиться адаптація коефіцієнтів, причому для зв'язків до кожного наступного шару вираховуються свої параметри зміни вагових коефіцієнтів Δ_{j-k} , які дещо змінені з врахуванням коефіцієнтів попереднього шару, k — відстань шару від вихідного. Вказані параметри зміни, як було доведено Полем Вербозом у 1986 році, неперервним чином змінюються по мірі віддалення від вихідного шару і утворюють дещо схоже на градієнт поля — тому така методика навчання зветься звичайно “градієнтним навчанням”, а завдяки тому, що адаптація вагових коефіцієнтів відбувається в напрямку від виходу на вхід, то альтернативною є назва “алгоритм зворотного розповсюдження”.

Наведений метод навчання є збіжним і гарантує навчання мережі для багатшарового персептрону, але має декілька суттєвих недоліків. Жорстке визначення структури мережі — в даному випадку це є контрольоване навчання — та контроль процесу навчання дають стійке рішення, але дозволяють створити не дуже гнучку нейронну мережу. Взагалі кажучи, для вирішення конкретного завдання потрібно створювати нову мережу і навчати її для розпізнавання або аналізу нового типу даних. Крім того, із формалізму навчального алгоритму видно, що процес навчання займає тривалий час і потребує великих обчислювальних потужностей, оскільки при просуванні даних крізь мережу спочатку самі дані проходять на вихід, потім відбувається градієнтна зміна вагових коефіцієнтів. Природним аналогом вказаної процедури навчання можуть бути інгібіторні нейронні зв'язки, адже в природних нейронних мережах інформація рухається взагалі в одному напрямку — зі входу на вихід.

Завдання.

За допомогою лабораторного програмного пакету емуляції штучних нейронних мереж виконати наступні завдання.

Завдання 1.

1. Побудувати двошарову мережу, яка буде виконувати логічну операцію XOR (2 входи, 2 шари, в першому шарі — 2 нейрона, в другому — 1, вихідний канал 1).
2. Занотувати час навчання, записати похибку.
3. Побудувати трьох- и чотирьохшарову мережі. Виміряти час навчання, записати похибку (середньоквадратичне відхилення).
4. Порівняти час навчання и похибку для різних структур мереж.
5. Зробити висновки.

Завдання 2.

1. Побудувати багат шарову мережу ЦАП. (Вхідних каналів — 8, вихідний — 1, прихованих шарів — довільна кількість, кожна підгрупа обирає собі незалежну від інших кількість прихованих шарів і кількість нейронів в них). Кількість нейронів необхідно планувати згідно такої формули:
$$(\text{розмірність даних}) \times 1,5 \leq (\text{кількість нейронів}) \leq (\text{розмірність даних}) \times 15$$
2. Створити навчаючу множину (патерн) в табличному процесорі (Excel, Origin) ступенів двійки для чисел від 0 до 10. Провести навчання мережі за цими даними. Записати похибку.
3. Розпізнати за допомогою розробленої мережі 5-10 бінарних чисел. Визначити похибку.
Порівняти результати різних груп і зробити висновки.

Додатки

Додаток 1.

Випадкова величина ξ має нормальний розподіл з параметрами

$$(a, \sigma), \text{ якщо густина її розподілу } p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right).$$

Додаток 2.

Випадкова величина ξ має розподіл Сімпсона на відрізку $[a, b]$,
якщо густина її розподілу

$$p(x) = \begin{cases} \frac{2}{b-a} - \frac{2}{(b-a)^2} |a+b-2x|, & a \leq x \leq b \\ 0, & x < a \text{ і } b < x \end{cases}.$$

Додаток 3.

Випадкова величина ξ має експоненціальний розподіл з параметром $\alpha > 0$, якщо густина її розподілу

$$p(x) = \begin{cases} \alpha \exp(-\alpha x), & x \geq 0 \\ 0, & x < 0 \end{cases}.$$

Додаток 4.

Випадковий вектор $\vec{\xi} = \{\xi_1, \xi_2, \xi_3, \dots, \xi_i, \dots, \xi_n\}$ має
поліноміальний розподіл з параметрами $n, a_1, a_2, a_3, \dots, a_n$

($0 < a_i < 1, \sum_{i=1}^n a_i = 1$), якщо

$$p(\vec{\xi} = \vec{m}) = p(\xi_1 = m_1, \xi_2 = m_2, \xi_3 = m_3, \dots, \xi_n = m_n) = \frac{n!}{m_1! m_2! m_3! \dots m_n!} a_1^{m_1} a_2^{m_2} a_3^{m_3} \dots a_n^{m_n}$$

для $\vec{m} = \{m_1, m_2, m_3, \dots, m_n\}$, якщо $\sum_{i=1}^n m_i = n$.

Таблиця 1 - Значення $t_{q,v}$ розподілу Стюдента.

v	q		
	0.1	0.05	0.01
1	6.314	12.706	63.657
2	2.92	4.303	9.925
3	2.353	3.182	5.841
4	2.132	2.776	4.604
5	2.015	2.571	4.032
6	1.943	2.447	3.707
7	1.895	2.365	3.499
8	1.86	2.306	3.355
9	1.833	2.262	3.25
10	1.812	2.228	3.189
12	1.812	2.179	3.055
14	1.782	2.145	2.977
16	1.761	2.12	2.921
18	1.746	2.101	2.878
20	1.734	2.086	2.845
22	1.725	2.074	2.819
24	1.717	2.064	2.797
26	1.711	2.056	2.779
28	1.706	2.048	2.763
30	1.697	2.042	2.75
∞	1.645	1.96	2.576

Таблиця 2 - Значення верхньої межі $\chi^2_{q,v}$.

v	q		
	0.1	0.05	0.01
1	2.7	3.8	6.6
2	4.6	6	9.2
3	6.3	7.8	11.3
4	7.8	9.5	13.3
5	9.2	11.1	15.1
6	10.6	12.6	16.8
7	12	14.1	18.5
8	13.4	15.5	21
9	14.7	16.9	21.7
10	16	18.3	23.2
12	18.5	21	26.2
14	21.1	23.7	29.1
16	23.5	26.3	32
18	26	28.9	34.8
20	28.4	31.4	37.6
22	30.8	33.9	40.3
24	33.2	36.4	43
26	35.6	38.9	45.6
28	37.9	41.3	48.3
30	40.3	43.8	50.9

Перелік посилань

-
- 1 И. Н Молчанов. Введение в алгоритмы параллельных вычислений. — К.: Наукова Думка, 1990. — 128 с.
 - 2 National Institute of Standards and Technologies
<http://math.nist.gov/>.
 - 3 Wim Vandersmissen. ClusterKnoppix is a modified Knoppix distro using the OpenMosix kernel. <http://bofh.be/clusterknoppix/>.
 - 4 Moshe Bar. OpenMosix Linux kernel extension for single-system image clustering. <http://www.openmosix.org>.
 - 5 LAM/MPI Parallel Computing. <http://www.lam-mpi.org>.
 - 6 Automatically Tuned Linear Algebra Software (ATLAS)
<http://www.netlib.org/atlas>.
 - 7 TOP500 SUPERCOMPUTER SITES. <http://www.top500.org/>
 - 8 Bown S. G. : Phototherapy of tumors. World J. Surg. 7 : 700-709 (1983).
 - 9 Masters A, Bown S. : Interstitial laser hyperthermia in the treatment of tumors. Lasers Med. Sci. , 129-136 (1990).
 - 10 М.В. Кононов, О.В.Кононов, М.К.Новоселець, І.М.Палій. Числове моделювання лазерної термотерапії пухлин головного мозку //Вісник Харківського університету, №488, 2000 - Біофізичний вісник, 2000, вип.1(6).с.85-95.
 - 11 М.В. Кононов, О.В.Кононов, М.К.Новоселець. Числове моделювання лазерної та мікрохвильової термотерапії пухлин передміхурової залози // Вісник Київського університету. Сер.Фізико-математичні науки,2001,вип.2.с.288-295.
 - 12 P. W. Ascher, F. Ebner, R. Stollberger. MR-guided laser-assisted thermotherapy of cerebral BFMN tumors. In “ Laser-induced Interstitial Thermotherapy” - SPIE Optical Ingeneering Press, USA, 1995, - p. 377-382.
 - 13 С. Самарский А.А., Теория разностных схем. - Г.: Наука, - 1989, 616 с.
 - 14 А. Н. Тихонов, В. Я. Арсенин. Методы решения некорректных задач. -Москва: Наука, 1979.-228 с.
 - 15 The physics of medical imaging, edited by S. Webb, IOP Publishing Ltd, 1995. - 640 p. (Физика визуализации изображений в медицине: В 2-х томах.: Пер. с англ./ Под ред .С. Уэбба. - М.: Мир, 1991).

-
- 16 Jack Simons. Theoretical Chemistry.
<http://simons.hec.utah.edu/NewUndergradBook/TableofContents.html>
- 17 EMSL Gaussian Basis Set Order Form
<http://www.emsl.pnl.gov/forms/basisform.html>
- 18 M.W.Schmidt, K.K.Baldrige, J.A.Boatz, S.T.Elbert, M.S.Gordon, J.H.Jensen, S.Koseki, N.Matsunaga, K.A.Nguyen, S.J.Su, T.L.Windus, M.Dupuis, J.A.Montgomery J.Comput.Chem. 14, 1347-1363(1993).
- 19 G.Schaftenaar and J.H. Noordik, "Molden: a pre- and post-processing program for molecular and electronic structures", J. Comput.-Aided Mol. Design, 14 (2000) 123-134
- 20 Muller B., Reinhardt J., Strickland M. T. Neural Networks: An Introduction. — Springer, 1995. — 329 p.
- 21 Lippman R. P. An introduction to computing with Neural Nets // IEEE ASSP Magazine. — 1987. — Vol.1. — P. 4-22.
- 22 Волькенштейн М.В. Биофизика. — М.: Наука, 1988. — 590 с.
- 23 Биофизика /Костюк П.Г., Гродзинский Д.М., Зима В.Л., Магура И.С., Сидорик Е.П., Шуба М.Ф./ Под ред. П.Г. Костюка. — К.: Вища школа, 1988. — 504 с.
- 24 Ежов А.А., Шумский С.А. Нейрокомпьютинг и его приложения в экономике. — МИФИ, 1998. — 238 с.
- 25 Olmsted D. History and Principles of Neural Networks. — Springer-Verlag, 1998. P. 100-126.
- 26 Kaski S., Kohonen T. Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World // Proc. of the Third International Conference on Neural Networks in the Capital Markets. — London. — 1995. —P.11-13.
- 27 Куssуль Э. М. Нейроподобные сети и нейрокомпьютеры. — К.: Наукова думка, 1993. — 84 с.