МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

М. М. ХУДИНЦЕВ

МОДЕЛЮВАННЯ ФІЗИКО-ХІМІЧНИХ ПРОЦЕСІВ У РАДІОЕКОЛОГІЇ

КОНСПЕКТ ЛЕКЦІЙ

ОДЕСА 2009

ББК 28.081
Х 98
УДК 539.1

Друкується за рішенням Вченої ради Одеського державного екологічного університету (протокол №9 від 30.10.2008 р.)

Худинцев М.М.

Моделювання фізико-хімічних процесів у радіоекології: Конспект лекцій. – Одеса: Вид-во «ТЕС», 2009, 90 с.

Конспект лекцій з відповідного навчального курсу призначений для теоретичного та практичного ознайомлення слухачів з сучасним станом аналітичного та чисельного моделювання простих фізико-хімічних процесів, які мають відношення до радіоекології та екології та передбачає наявність у слухачів підготовки на рівні курсу загальної фізики, що викладається у вищих навчальних закладах інженерно-технічного профілю.

Конспект лекцій використовується для денної форми навчання.

© Одеський державний
екологічний університет, 2009

ЗМІСТ

ВСТУП

Конспект лекцій з дисципліни «Моделювання фізико-хімічних процесів в радіоекології» створено для студентів спеціальності екологія та охорона навколишнього середовища (спеціалізація – радіоекологія).

Зміст конспекту адекватно відтворює зміст відповідної робочої програми та складається з таких основних частин:

- система інженерних і наукових розрахунків MATLAB;
- методи моделювання;
- моделі екологічних процесів;
- математичні основи методів моделювання;
- контрольні завдання.

Велика увага під час викладення дисципліни приділяється вивченю системи інженерних і наукових розрахунків MATLAB, яка є одною з найсучасніших та дуже зручною системою чисельних розрахунків, має інтуітивно порозумілий інтерфейс та широкі можливості для застосування – від найпростіших навчальних завдань до чисельних розрахунків високого наукового рівня.

Значне місце у матеріалах лекцій присвячено також застосуванню системи інженерних і наукових розрахунків MATLAB для безпосереднього чисельного моделювання різноманітних фізичних, хімичних та біологічних процесів у відповідних системах. Розглянуто великий перелік конкретних процесів, систем та методів моделювання.

Наведено також зразки контрольних завдань, які по суті є мінінауковими дослідженнями, що дозволяють не лише закріпити матеріали лекцій, а ще й надати слухачам методичні навички для подальшого виконання курсових та дипломних робіт і проектів.

1. Система інженерних і наукових розрахунків MATLAB

1.1.Загальні відомості MATLAB

MATLAB - це високопродуктивна мова для технічних розрахунків. Він містить у собі обчислення, візуалізацію і програмування в зручному середовищі, де задачі та розв'язки надаються у вигляді, який є наближеним до математичного. ATLAB дозволяє виконувати математичні розрахунки, створювати алгоритми, здійснювати моделювання, проводити аналіз даних, їх дослідження та візуалізацію з використанням наукової та інженерної графіки.

MATLAB - це інтерактивна система, у якій основним елементом даних є масив. Це дозволяє розв'язувати різні задачі, які пов'язані з розрахунками, насамперед у яких використовуються матриці й вектора, у кілька разів швидше, ніж при написанні програм з використанням "скалярних" мов програмування (зокрема, Сі або Фортран).

Слово MATLAB означає матрична лабораторія (matrix laboratory). MATLAB був спеціально написаний для забезпечення легкого доступу до LINPACK і EISPACK, які являють собою сучасні програмні засоби для матричних обчислень.

В MATLAB важлива роль приділяється спеціалізованим групам програм, які називаються toolboxes. Вони дуже важливі для більшості користувачів MATLAB, тому що дозволяють вивчати й застосовувати спеціалізовані методи. Toolboxes - це всебічна колекція функцій MATLAB (M-файлів), які дозволяють вирішувати значний клас задач. Toolboxes застосовуються для обробки сигналів, систем контролю, нейронних мереж, нечіткої логіки, вейвлетів, моделювання і т.д.

Система MATLAB складається з п'яти основних частин, які будуть розглянуті нижче:

- **мова** MATLAB;
- **середовище** MATLAB;
- **графіка** MATLAB;
- **бібліотека** математичних функцій;
- програмний **інтерфейс**.

Щоб запустити MATLAB на PC, двічі клацніть на іконку MATLAB. Для запуску в системі UNIX напишіть *matlab* у рядку операційної системи. Для виходу з MATLAB необхідно набрати *quit* у рядку MATLAB.

Якщо вам необхідно одержати додаткову інформацію, наберіть *help* у рядку MATLAB або виберіть **Help** у меню на PC.

Додаткова інформація по розділах MATLAB одержується за допомогою команди *help help.*

1.2. Матриці

Кращий спосіб почати роботу з MATLAB - це навчитися роботі з матрицями. В MATLAB матриця - це прямокутний масив чисел. Особливе значення надається матрицям 1x1, які є скалярами, і матрицям, що мають один стовбець або один рядок, - векторам. MATLAB використовує різні способи для зберігання чисельних і нечисельних даних, однак спочатку найкраще розглядати всі дані як матриці. MATLAB організований так, щоб всі операції були як можна більш природними. У той час як інші програмні мови працюють із числами як елементами мови, MATLAB дозволяє вам швидко й легко оперувати із цілими матрицями.

Побудова матриць

Ви можете задавати матриці в MATLAB декількома способами:

- записати повний перелік елементів
- завантажувати матриці із зовнішніх файлів
- генерувати матриці, використовуючи вбудовані функції
- створювати матриці за допомогою власних функцій у М-файлах

Запис елементів повинен задовольняти наступним умовам:

- відокремлення елементів рядка пробілами або комами
- викормстання крапки з комою для позначення закінчення кожного рядка
- оточення переліку елементів квадратними дужками [ ].

Наприклад: магічна матриця

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB відобразить матрицю A так:

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Матриця автоматично запам'ятовується середовищем MATLAB. Ми можемо до неї звернутися так:

A (та натиснить клавішу Enter).

Операції підсумовування елементів, транспонування та діагоналізації матриці

У наведеному прикладі, якщо знайти суми елементів уздовж будь-якого рядка, стовпця, або головної діагоналі, ви завжди отримаєте однакові числа. Для знаходження сум стобпців потрібно записати:

```
sum(A)
```

MATLAB видасть відповідь

```
ans =
  34     34     34     34
```

Коли вихідна змінна не визначена, MATLAB використовує змінну *ans,* коротко від answer - відповідь, для зберігання результатів обчислення.

Для одержання суми у рядках потрібно спочатку транспонувати матрицю (тобто змінити стовпці та рядки місцями), підрахувати суму в стовпцях, а потім транспонувати результат. Операція транспонування позначається апострофом або одинарними лапками. Вона дзеркально відображає матрицю щодо головної діагоналі й міняє рядки на стовпці. У такий спосіб

```
A'
```

викликає

```
ans =
  16    5     9     4
   3   10     6    15
   2   11     7    14
  13    8    12     1
```

А вираз

`sum(A')'` викликає результат вектор-стовпець, який містить суми

в рядках

```
ans =
 34 34
 34 34
```

Суму елементів на головній діагоналі можна легко одержати за допомогою функції *diag,* що вибирає цю діагональ.

```
diag(A)
```

```
ans =
 16 10
  7  1
```

А функція **sum(diag(A))** викликає

```
ans =
 34
```

Для суми елементів іншої діагоналі, яка називається побічною, немає спеціальної функції.

Індекси
Елемент у рядку і і столбці j матриці A позначається A(i,j). Наприклад, A(4,2) - це число в четвертому рядку й другому стовпці. Для нашого магічного квадрату A(4,2) =15. Таким чином, можна обчислити суму елементів у четвертому стовпці матриці A, набравши

```
A(1,4) + A(2,4) + A(3,4) + A(4,4)
```

одержимо

```
ans =
   34
```

Також можливо звертатися до елементів матриці через один індекс, A(k). Це звичайний спосіб посилання на рядки й стовпці матриці. Але його можна застосовувати тільки із двовимірними матрицями. У цьому випадку масив вважається вектором, який сформований із стовпців вихідної матриці.

Так, для нашого магічного квадрата, A(8) - це інший спосіб посилатися на значення 15, що зберігається в A(4,2).
Якщо ви намагаєтеся використати значення елемента поза матрицею, MATLAB ви^-дасть помилку:

```
t=A(4,5)
```

```
??? Index exceeds matrix dimensions.
```

З іншого боку, якщо ви зберігаєте значення поза матрицею, то розмір матри-цы збільшується.

```
X=A;
```

```
X(4,5) = 17
```

Відповідь має вигляд:

```
X =
    16     3     2    13     0
     5    10    11     8     0
     9     6     7    12     0
     4    15    14     1    17
```

Оператор двокрапки
Двокрапка „:” , - це один з найбільш важливих операторів MATLAB. Наприклад:

```
1:10
```

- це вектор-рядок, який містить цілі числа від 1 до 10

```
 1     2     3     4     5     6     7     8
 9    10
```

```
100:-7:50
```

дає

```
100    93    86    79    72    65    58    51
```

```
0:pi/4:pi
```

приводить до

```
 0    0.7854    1.5708    2.3562    3.1416
```

Індексний вираз з двокрапкою описує частину матриці:

```
A(1:k,j)
```

– це перші k елементів j -го стовпця матриці A.

```
sum(A(1:4,4))
```

обчислює суму четвертого рядка.

Відокремлена двокрапка звертається до всіх елементів у рядку й стовпці матриці, а слово ***end*** - до останнього рядка або стовпця.

```
sum(A(:,end))
```

обчислює суму елементів в останньому стовпці матриці A

```
ans =
    34
```

Функція **magic**
MATLAB має вбудовану функцію *magic*, яка створює магічний квадрат будь-якого розміру.

```
B=magic(4)
```

```
B =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Ця матриця майже співпадає з матрицею A, яка розглядалася вище. Єдина відміна полягає у тому, що два середніх стовпця помінялися місцями. Для перетворрення матриці B у матрицю A, переставимо їх місцями.

```
A=B(:,[1 3 2 4])
```

Це означає, що для кожного рядка матриці B елементи порядкуються за номерами рядків 1, 3, 2, 4

```
A =
    16     3     2    13
     5    10          11
          8
     9     6     7    12
     4    15          14
          1
```

1.3.Вирази

Як і більшість інших мов програмування, MATLAB надає можливість використання математичних виразів, але на відміну від багатьох з них, ці вирази в MATLAB включають матриці. Основні складові вирази:

- змінні
- числа
- оператори
- функції

Змінні
В MATLAB немає необхідності у визначенні типу змінних або розмірності. Коли MATLAB зустрічає нове ім'я змінної, він автоматично утворює змінну й виділяє відповідний об'єм пам'яті. Якщо змінна вже існує, MATLAB змінює її склад і, якщо це необхідно, виділяє додаткову пам'ять. Наприклад,

```
num_students = 25
```

створює матрицю 1x1 з ім'ям numstudents і зберігає значення 25 у її елементі.

Імена змінних складаються з літер, цифр або символів підкреслення. MATLAB використовує тільки перші 31 символ імені змінної. MATLAB чутливий до регістрів, він розрізняє заголовні й малі літери. Тому A і a – різні змінні. Щоб побачити матрицю, яка пов'язана із змінною, введіть назву змінної.

Числа
MATLAB використає прийняту десяткову систему числення, з необов'язковою десятковою крапкою й знаками плюс-мінус для чисел. Наукова система числення використовує літеру e для визначення множника ступеня десяти. Уявні числа використовують i або j як суфікс. Деякі приклади правильних чисел наведені нижче

3	-99	0.0001
9. 6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

Всі числа для зберігання використовують формат long з плаваючою крапкою. Числа із плаваючою крапкою мають обмежену точність - приблизно 16 значущих цифр і обмеженим діапазоном – приблизно від 10^{-3} до 10.

Оператори
Вирази використовують звичайні арифметичні операції й правила старшинства.

+ додавання

– вирахування

***** множення

/ ділення (для чисел та простих змінних)

**** ліве ділення (для векторів та матриць)

^ ступінь

' комплексно сполучене транспонування

( ) визначення порядку обчислення

Функції
MATLAB включає велику кількість елементарних математичних функцій, таких, як ***abs, sqrt, exp, sin***. Обчислення квадратного кореня негативного числа не є помилкою: у цьому випадку результатом є відповідне комплексне число. MATLAB також включає спеціальні функції, наприклад, Гама-функцію або функції Бесселя. Більшість з них мають комплексні аргументи.

До переліку усіх елементарних математичних функцій звертається команда

```
help elfun
```

Для спеціальних математичних або матричних функцій – наберіть

```
help
specfun
```

або

```
help elmat
```

відповідно.

Деякі функції, такі як ***sqrt*** і ***sin***, є вбудованими. Але деякі інші функції, такі, як ***gamma*** і ***sinh***, реалізовані в М-файлах. Тому ви можете легко побачити їхній код і, якщо буде потрібно, навіть модифікувати його.

Кілька спеціальних функцій надають значення часто використовуваних констант.

pi	3.14159265...
i	мнима одиниця
j	те ж саме, що й **i**
eps	відносна точність числа із плаваючою крапкою
realmin	найменше число з плаваючою крапкою
realmax	найбільше число з плаваючою крапкою
Inf	нескінченність
NaH	не число

Нескінченність з'являється після діленні на нуль або при виконанні математичної операції, яка призводить до переповнення, тобто до перевищення ***realmax***. Не число *(NaH)* генерується при обчисленні виразів типу **0/0** або ***Inf – Inf***, які не мають певного математичного значення.

Імена функцій не є зарезервованими, тому можливо змінювати їхні значення на нові, наприклад

```
eps = 1.e-6
```

і далі використовувати це значення в наступних обчисленнях. Початкове значення може бути відновлене в такий спосіб

```
clear eps
```

Вирази
Приклади з результатами:

```
rho = (1+sqrt(5))/2
```

```
rho =
   1.6180
```

```
a = abs(3+4i)
```

```
a =
     5
z = sqrt(besselk(4/3,rho-i))
z =
   0.3730 + 0.3214i
huge = exp(log(realmax))
huge =
     1.7977e+308
toobig = pi*huge
toobig =
     Inf
```

1.4. Робота з матрицями

Цей розділ присвячено різним способам утворення матриць.

Генерування матриць
MATLAB має чотири функції, які створюють основні матриці:

zeros	всі нулі
ones	всі одиниці
rand	рівномірний розподіл випадкових елементів
randn	нормальний розподіл випадкових елементів

Деякі приклади:

```
Z = zeros(2,4)
Z =
     0     0     0     0
     0     0     0     0
F = 5*ones(3,3)
F =
     5     5     5
     5     5     5
     5     5     5
N = fix(10*rand(1,10))
```

```
N =

     9     2     6

R = randn(4,4)

R =
   -0. 4326-1.14650.3273-0.5883
   -1. 66561.19090.1746  2.1832
    0. 12531.1892-0.1867 -0.1364
    0. 2877-0.03760.7258  0.1139
```

Завантаження матриць
Команда ***load*** зчитує файли, які містять матриці, що створені у MATLAB раніше, або текстові файли, які містять чисельні дані. Текстові файли повинні бути сформовані у вигляді прямокутної таблиці чисел, які виділятимуться пробілами, з рівною кількістю елементів у кожному рядку. Наприклад, створимо поза MATLAB текстовий файл з чотирьох рядків:

16.0	3.0	2.0	13.0
5.0	10.0	11.0	8.0
9.0	6.0	7.0	12.0
4.0	15.0	14.0	1.0

Збережемо цей файл з ім'ям magik.dat. Тоді команда load magik.dat прочитає цей файл і створить змінну magik, яка містить матрицю.

М-файли
Ви можете створювати свої власні матриці, використовуючи М-файли, які являють собою текстові файли, що містять код MATLAB. Достатньо створити файл з командами у командному рядку MATLAB. Збережіть його з ім'ям *.m, де * - сукупність літер, цифр та символу „_”.

Наприклад, створимо файл, що включає наступних 5 рядків:

```
A =  [16.0   3.0    2.0    13.0
       5.0   10.0   11.0          8.0
       9.0   6.0     7.0    12.0
       4.0   15.0   14.0     1.0 ];
```

Збережемо його з ім'ям magik.m. Тоді вираз

```
magik
```

прочитає файл і створить змінну A, яка містить вихідну матрицю.

1.5. Командне вікно

Дотепер, ми використовували лише командний рядок MATLAB, записуючи команди й вирази та отримаючи результати. Якщо ваша система дозволяє вам вибирати шрифт, рекомендується використовувати шрифти з фиксованою шириною, такі як Fixedsys або Courier, для забезпечення коректного інтервалу.

Команда format
Команда format управляє чисельним форматом значень. Ця операція впливає тільки на вигяд даних на екрані. Нижче наведені формати чисел, які використовуються для відображення вектора x з різними компонентами:

```
x = [4/3 1.2345e-
6] format short
1.3333    0.0000
format short e
1.3333e+000  1.2345e-006
format short g
1.3333  1.2345e-006
format long
1.33333333333333   0.00000123450000
format long e
1.333333333333333e+000    1.234500000000000e-006
format long g
1. 33333333333333      1.2345e-006
format bank
1.33      0.00
```

Якщо найбільший елемент матриці більше 10 або сам маленький менше 10^{-3}, MATLAB застосовує загальний масштабний коефіцієнт для форматів ***short*** та ***long***.

У додавання до команд ***format*** ще існує

```
format compact
```

який забирає порожні рядки, що з'являються на екрані. Це дозволяє бачити на екрані більше інформації.

Скорочення вихідних даних
Якщо наприкінці рядка ви поставите крапку з коми, MATLAB проведе обчислення, але не відобразить їх на екрані. Це дуже важливе зауваження для великих масивів даних. Наприклад,

```
A = magic(100);
```

Редактор командного рядка
Використовуйте кнопки з стрілками для звертання до окремих рядків програми без їхнього повторного набіру.

1.6. Графіка

MATLAB має широкі можливості для графічного зображення векторів і матриць, а також для створення коментарів і друку графіки.

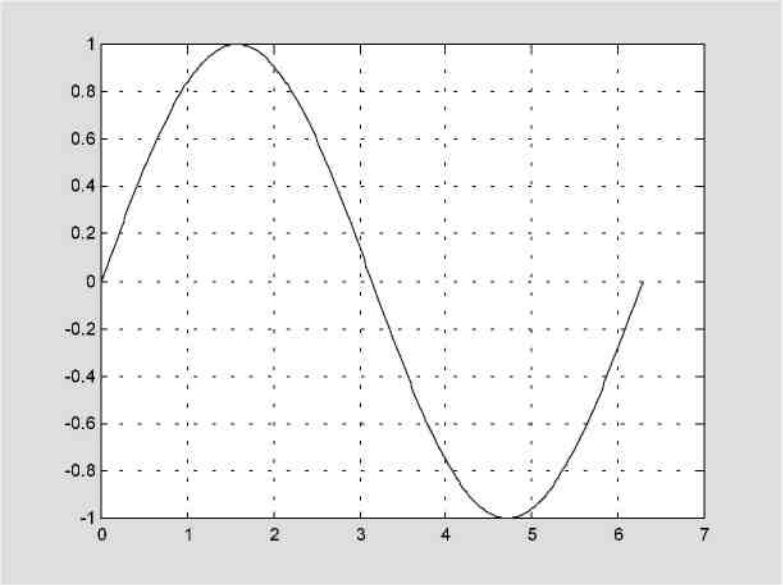
Створення графіка
Функція ***plot*** має різні форми, що пов'язані із вхідними параметрами, наприклад ***plot(y)*** створює кусочно-лінійний графік залежності елементів y від їхніх індексів. Якщо задати два вектори, як аргументи, ***plot(x,y)*** створить графік залежності *y* від *x*.

Наприклад, для побудови графіка значень функції ***sin*** від нуля до *2π,* потрібно зробити наступне

```
t = 0:pi/100:2*pi
;

y = sin(t);

plot(t,y)
```



Можливо створити водночас декілька різнокольорових графіків

```
y2 = sin(t-.25);
y3 = sin(t-.5);
plot( t, y, t, y2, t, y3)
```

Можливі зміна кольорів, стилю ліній й маркерів. Наприклад, вираз

```
plot(x,y,'y:+')
```

будує жовтий пунктирний графік і поміщає маркери ′ + ′у кожну крапку даних. Якщо ви визначаєте тільки тип маркера, але не визначаєте тип стилю ліній, то MATLAB виведе тільки маркери.

Детальніше про кольори та маркери дивіться за допомогою команди **help plot**.

Вікна зображень
Функція ***plot*** автоматично відкриває нове вікно зображення (далі вікно), якщо його не було на екрані. Якщо воно вже існує, ***plot*** використовуватиме його за замовчуванням. Для відкриття та вибору нового вікна наберіть

```
figure
```

або

```
figure(n)
```

де n - це номер у заголовку вікна. У цьому випадку результати всіх наступних команд будуть виводитися в це вікно.

Додавання кривих на існуючий графіки
Команда ***hold*** дозволяє додавати криві на існуючий графік. Коли ви набираєте

```
hold on
```

MATLAB не стирає існуючий графіки, а додає в нього нові дані, змінюючи осі, якщо це необхідно. Наприклад, наведений елемент програми спочатку створює контурні лінії функції ***peaks,*** а потім накладає графік тієї ж функції:

```
[x,y,z] = peaks;
contour(x,y,z,20,
k') hold on
pcolor(x,y,z)
shading interp
```

Команда ***hold on*** спичиняє комбінацію графіку ***pcolor*** з графіком ***contour*** в одному вікні.

Підграфіки
Функція ***subplot*** дозволяє виводити множину графіків в одному вікні або роздруковувати їх на одному аркуші паперу.

```
subplot(m,n,p)
```

розбиває вікно зображень на матрицю m на n підграфіків та обирає p-ий підграфік поточним. Графіки нумеруються уздовж першого у верхньому рядку, потім у другий і т.д. Наприклад, для того, щоб навести графічні дані в чотирьох різних підвікнах необхідно виконати наступне:

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1)
mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```

Уявні та комплексні дані
Якщо аргумент функції ***plot*** комплексне число, уявна частина ігнорується, за винятком випадків, коли комплексний аргумент один. Тому

```
plot(Z),
```

де Z – комплексний вектор або матриця,

еквівалентно

```
plot(real(Z),imag(Z))
```

Наприклад,

```
t = 0:pi/10:2*pi;
plot(exp(i*t),'-o')
```

відобразить двадцятибічний багатокутник з символами **o** на вершинах.

Управління вісями
Функція ***axis*** має кіілька можливостей для налагодження масштабу, орієниації й коефіцієнта стиснення.

Звичайно MATLAB обирає максимальне й мінімальне значення й вибирає відповідний масштаб і маркіровку вісей. Функція *axis* змінює значення за замовчуванням граничними значення, які вводяться користувачем.

```
axis( [xmin xmax ymin ymax] )
```

У функції ***axis*** можна також використовувати ключові слова для управління зовнішнім виглядом вісей. Наприклад

```
axis square
```

створює x і y осі рівної довжини,

```
plot(exp(i*t))
```

наступна або за ***axis square,*** або за ***axis equal*** перетворює овал у правильне коло,

```
axis auto
```

повертає значення за замовчуванням і переходить в автоматичний режим,

```
grid off
```

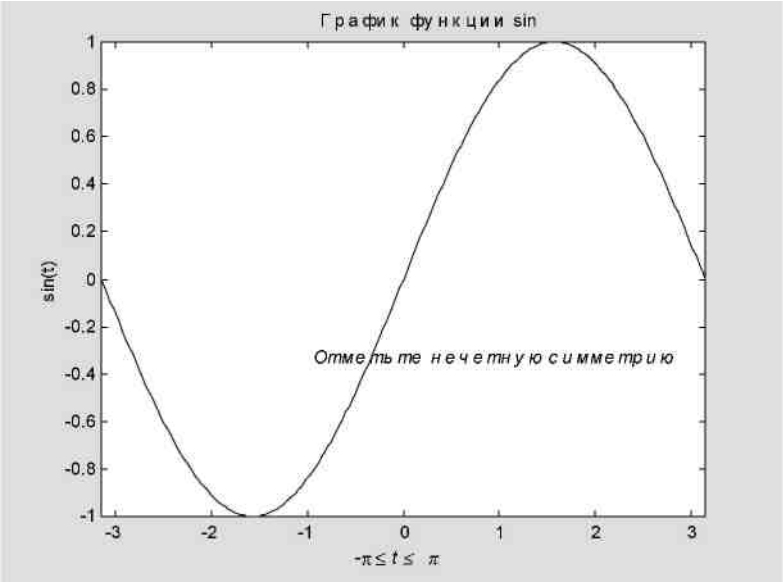
виключає сітку координат, а

```
grid on
```

включає її заново.

Підписи до вісей і заголовки
Функції *xlabel, ylabel, zlabel* додають підписи до відповідних вісей, функція title додає заголовок у верхню частину вікна, а функція text додає текст у будь-яке місце графіка. Використання TEX подання дозволяє застосовувати грецькі літери, математичні символи й різні шрифти. Наступний приклад демонструє цю можливість.

```
t  = -pi:pi/100:pi;
y = sin(t);
plot(t,y)
axis([-pi pi -1 1])
xlabel( ' -\pi \leq \itt \leq \pi ' )
ylabel( ' sin(t) ' )
title( ' Графік функції sin ' )
text(-1, -1/3, ' \it{Отметьте нечетную
симметрию}' )
```



Функції mesh і surface
MATLAB визначає поверхню, як z координати точок над координатною сіткою площини x-y, використовуючи прямі лінії для з'єднання сусідніх точок. Функції *mesh* і *surface* відображають поверхню у трьох вимірах. При цьому *mesh* створює каркасну поверхню, де кольорові лінії з'єднують тільки задані точки, а функція *surface* разом з лініями відображає кольором й саму поверхню.

Друк графіки
Опція **Print** у меню **File** і команда ***print*** друкують графікові MATLAB. Меню **Print** викликає діалогове вікно, яке дозволяє обирати загальні стандартні варіанти друку. Команда ***print*** забезпечує значну гнучкість надання вихідних даних і дозволяє контролювати друк з М-файлів. Результат може бути надісланий безпосереднью на принтер, обраний за замовчуванням, або збережений у заданому файлі. Можливо широке варіювання формату вихідних даних, включаючи використання PostScript.

Наприклад, команда

```
print -depsc2 magicsquare.eps
```

збереже поточне вікно зображення, як кольоровий PostScript Level 2 Encapsulated у файлі magicsquare.eps

Слід знати, що не всі Postscript принтери підтримують Level 2.

1.7. Довідка та поточна документація

Є кілька способів отримати поточну документацію по функціях MATLAB.

- Команда help
- Вікно довідки
- MATLAB Help Desk
- Поточні довідкові сторінки
- Зв'язок з The MathWorks, Inc.

Команда help

Команда ***help*** – основний спосіб визначення синтаксису й поводження окремих функцій. Інформація відображається у командному вікні. Наприклад

```
help magic
```

видасть

```
MAGIC Magic square.
   MAGIC(N) is an N-by-N matrix constructed
   from the integers 1 through N^2 with equal
   row, column, and diagonal sums. Produces
   valid magic squares for N = 1,3,4,5,...
```

MATLAB у поточній довідці використовує великі літери для функцій і змінних. Однак імена функції складаються з малих літер.

Всі функції MATLAB організовані в логічні групи. Наприклад, всі функції лінійної алгебри перебувають у директорії matfun. Щоб вивести імена всіх функцій у цій директорії з коротким описом, треба набрати

```
help matfun

Matrix functions - numerical linear algebra.

Matrix analysis.
  norm       - Matrix or vector
  norm. normest     - Estimate the
  matrix 2-norm.
```

Команда

```
help
HELP topics:

matlab\general       -  General purpose commands.
matlab\ops      -  Operators and special characters
```

Вікно довідки

Вікно довідки MATLAB з'являється на PC або Mac після вибору опції **Help Window** у меню **Help** або натисканням кнопки питання на панелі інструментів. Та сама операція може бути виконана при наборі команди

```
helpwin
```

Для отримання вікна довідки по окремих розділах потрібна команда

```
helpwin topic
```

Вікно довідки надає таку саму інформацію, що і команда ***help,*** але віконний інтерфейс забезпечує більш зручний зв'язок з іншими розділами довідки.

1.8. Середовище MATLAB

Середовище MATLAB містить у собі як сукупність змінних, створених за час сеансу роботи MATLAB, так і набір файлів, що містять програми й дані, які продовжують існувати між сеансами роботи.

Робочий простір
Робочий простір – це область пам'яті, доступна з командного рядка MATLAB. Дві команди, ***who*** і ***whos,*** показують поточний зміст робочого простору. Команда ***who*** видає короткий список, а команда ***whos*** розмір і використовувану пам'ять.

Наприклад,

```
whos
```

Name	Size	Bytes	Class
A	4x4	128	double
D	5x3	120	double
M	10x1	3816	cell array
S	1x3	442	struct
H	1x11	22	char array
N	1x1	8	double
S	1x5	10	char array
V	2x5	20	char array

Grand total is 471 elements using 4566 bytes.

Для видалення усіх існуючих змінних з робочого простору MATLAB, наберіть

```
clear
```

Команда save
Команда ***save*** зберігає зміст робочого простору у М-файлі, що може бути прочитаний командою ***load*** у наступних сеансах роботи MATLAB. Наприклад,

```
save August17th
```

зберігає зміст усього робочого простору у файлі August17th.mat. Якщо потрібно, ви можете зберегти тільки певні змінні, указуючи їхні імена після імені файлу.

-ascii Використовує 8-знаковий текстовий формат

-ascii –double Використовує 16-знаковий текстовий

формат

-ascii -double –tabs Розділяє елементи масиву табуляцією

-v4 Створює файл для MATLAB 4

-append Додає дані в існуючий MAT-файл.

Команда

```
path
```

указує маршрут пошуку по усіх платформах. На PC і Mac оберіть опцію **Set Path** з меню **File** для перегляду і зміни маршруту.

Операції над дисковими файлами
Команди ***dir, type, delete*** і ***cd*** здійснюють комплекс групових операційних системних команд для маніпуляцій над файлами.

Наприклад:

MATLAB	MS-DOS	UNIX	VAX/VMS
dir	Dir	Ls	dir
type	Type	cat	type
delete	del. erase	rm	delete
cd	Chdir	cd	set defalt

Команда diary
Команда ***diary*** створює щоденник сеансу MATLAB у дисковому файлі. Ви можете переглянути й відредагувати кінцевий текстовий файл, використовуючи будь-який текстовий процесор. Для створення файлу під ім'ям diary, який містить всі команди, які ви використовували, уведіть

```
diary
```

Для збереження сеансу MATLAB у файлі з певним ім'ям,

використайте

```
diary filename
```

При зупинці запису сеансу роботи, наберіть

```
diary off
```

1.9. Операції з матрицями та масивами

Ви вже бачили операцію транспонування матриці, A'. Додавання матриці до її транспонованого дає симетричну матрицю.

```
A + A'
```

```
ans =
   32    8   11   17
    8   20   17   23
   11   17   14   26
   17   23   26    2
```

Символ множення, *, позначає матричний добуток. Множення матриці на її транспоновану також дає симетричну матрицю.

```
A'*A
```

```
ans =
   378  212  206  360
   212  370  368  206
   206  368  370  212
   360  206  212  378
```

Визначник матриці обчислюється за допомогою

```
d = det(A)
```

```
d =
     0
```

Оскільки задана матриця є сингулярною (det=0), вона не має зворотної. Якщо зробити спробу визначити зворотню матрицю

```
X = inv(A),
```

отримаємо попередження

```
Warning: Matrix is close to singular or badly
scaled.
       Results may be inaccurate. RCOND =
       1.175530e-017.
```

Власні значення визначаються

```
e = eig(A)
```

```
e =
  34.0000
   8.0000
  -0.0000
  -8.0000
```

Одне із власних значень дорівнює нулю, що є наслідком сінгулярності. Найбільше власне значення дорівнює 34, магічній сумі. Це відбувається тому, що вектор, який складається з одиниць, є власним вектором.

Ступінь матриці позначається

A^2

Коефіцієнти характеристичного поліному отримуються за допомогою

```
poly(A)
```

```
ans =
  1.0e+003 *
    0.0010   -0.0340   -0.0640    2.1760    0.0000
```

Масиви

Коли ми виходимо з миру лінійної алгебри, матриці стають двовимірними чисельними масивами. Арифметичні операції з масивами здійснюються поелементно. Це означає, що підсумовування

й вирахування є однаковими операціями для матриць і масивів, а множення – по-різному.

Список операторів містить у собі:

+	підсумовування
–	вирахування
.*	множення
. /	ділення
.	ліве ділення
.^	піднесення в ступінь
.'	матричне транспонування

Елементарні математичні функції працюють із масивами поелементно. Так

```
format short g x =
(1.5:0.1:2)';

logs = [x log10(x)]
```

створює таблицю логарифмів

```
logs =

    1.5  0.17609

    1.6  0.20412

    1.7  0.23045

    1.8  0.25527

    1.9  0.27875

    2    0.30103
```

1.10. Управління потоками

MATLAB має п'ять видів структур керування потоками:

- оператор ***if***
- оператор ***switch***
- цикли ***for***
- цикли ***while***
- оператор ***break***

if

Оператор *if* обчислює логічний вираз й виконує групу операторів, якщо значення виразу „істина”. Необов'язкові ключові слова ***elseif*** і ***else*** слугують для виконання альтернативних груп операторів. Ключове слово ***end,*** що погоджується з ***if***, завершує останню групу операторів без залучення фігурних або звичайних дужок.

Алгоритм MATLAB для створення магічного квадрата порядку n поділяється на три різних випадки : n непарне, n парне, але не ділиться на 4, і n парне й ділиться на 4. Нижче наведенj приклад відповідного коду

```
if rem(n,2) ~= 0
  M = oddmagic(n)
elseif rem(n,4) ~= 0
  M =
single even magic(n
) else
  M =
doubleevenmagic(n)
end
```

switch і case

Оператор ***switch*** виконує групу операторів, базуючись на значенні змінний або вираження. Ключові слова ***case*** і ***otherwise*** розділяють ці групи. Виконується тільки перший відповідний випадок. Необхідно використовувати ***end*** для узгодження з ***switch***.

Логіка алгоритму магічних квадратів може бути описана на наступному прикладі

```
switch (rem(n,4)==0) +
  (rem(n,2)==0) case 0
   M = oddmagic(n) case 1
   M = single even magic(n) case
 2
   M = doubleevenmagic(n)
   Otherwise error( ' Це
неможливо' ) end
```

for

Цикл ***for*** повторює групу операторів фіксоване, визначене число раз. Ключове слово ***end*** окреслює тіло циклу.

```
for n = 3:32
  r(n) =
rank(magic(n));
  end r
```

Крапка з комою після виразув тілі циклу запобігає повторення виводу результатів на екран, а r після циклу виводить остаточний результат.

while

Цикл ***while*** повторює групу операторів певне число раз, поки виконується логічна умова. Ключове слово ***end*** окреслює тіло циклу.

Нижче наведена повна програма, яка ілюструє роботу операторів ***while, if, else*** і ***end,*** що використає метод ділення відрізка навпіл для знаходження нулів поліному.

```
a = 0;

fa = -Inf;

b = 3;

fb = Inf;

while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
  if sign(fx) == sign(fa)
    a = x; fa = fx;
  else
    b = x; fb = fx;
  end
end x
```

Результатом буде корінь полінома

```
x =
2.09455148154233
```

Для оператора while вірні ті ж застереження щодо матричного порівняння, що й для оператора if, які обговорювалися раніше.

break

Оператор ***break*** дозволяє достроково виходити із циклів ***for*** або ***while***. У вкладених циклах *break* здійснює вихід тільки із самого внутрішнього циклу.

Нижче представлений варіант попереднього прикладу. Чому використання оператора ***break*** у цьому випадку вигідно?

```
a = 0; fa = -
Inf; b = 3; fb
= Inf; while
b-a > eps*b x
= (a+b)/2; fx
= x^3-2*x-5;
if fx ==0
    break elseif
  sign(fx) == sign(fa)
    a = x; fa =
  fx; else
  b = x; fb = fx;
end end x
```

1.11. Сценарії та функції

MATLAB - це потужна мова програмування, також як і інтерактивне обчислювальне середовище. Файли, які містять код мовою MATLAB, називаються М-файлами. Ви створюєте М-файли, який є водночас функцію або командою MATLAB.

Існує два види М-файлів

- Сценарії, які не мають вхідних і вихідних аргументів. Вони оперують з даними з робочого простору.
- Функції, які мають вхідні й вихідні аргументи. Вони оперують з локальними змінними.

Якщо ви є новачком в MATLAB програмуванні, створюйте М-файли у поточній директорії. Якщо ж ви розробили багато М-файлів, для їх группування у різні директорії й персональні пакети програм (toolboxes) необхідно додати їхній маршрут пошуку MATLAB.

Якщо ви повторюєте ім'я функції, то MATLAB викликають лише першу з них.

Зміст М-файлу редактуватиметься звичайним способом за допомогою будь-якого текстового редактору.

Сценарії

Коду ви викликаєте сценарій, MATLAB викликає команди з файлу. Сценарії можуть оперувати існуючими даними в робочому просторі або вони можуть створювати ці дані. Хоча сценарії не повертають значень, всі змінні, які вони створюють, залишаються у робочому просторі для використання в наступних обчисленнях.

Функції

Функції – це М-файли, які матимуть вхідні й вихідні дані. Ім'я М-файлу й функції повинне бути тим самим. Функції працюють із змінними в межах їх власного робочого простору.

Приклад функції *rank.* М-файл *rank.m* перебуває у директорії

toolbox/matlab/matfun

Ви можете переглянути його зміст за допомогою

```
type rank
```

```
function r = rank(A,tol)
 %RANK  Matrix rank.
 %  RANK(A) provides an estimate of the number
of linearly
 %   independent rows or columns of a matrix A.
 %  RANK(A,tol) is the number of singular values
of A
 %  that are larger than tol.
 %  RANK(A) uses the default tol = max(size(A))
* norm(A) * eps.
 %   Copyright (c) 1984-98 by The MathWorks, Inc.
%   $Revision: 5.7 $  $Date: 1997/11/21 23:38:49 $


s = svd(A);
if nargin==1
  tol = max(size(A)') *
max(s) * eps; end r = sum(s
> tol);
```

Перший рядок функції М-файлу починається з слову ***function***. Тут відбувається визначення імені та переліку аргументів. У нашому випадку, використовується до двох вхідних аргументів і один вихідний.

Функція ***rank*** може бути використана у інших сценаріях у різних формах:

```
rank(A)
r = rank(A)
```

```
r = rank(A, 1.e-6)
```

Глобальні змінні
Якщо ви хочете, щоб більше однієї функції використали окрему копію змінної, просто оголосите її як ***global*** у всіх функціях:

```
global d1 p s
```

Функція від функцій
Клас функцій з умовною назвою "функція від функцій", працює з нелінійними функціями скалярних змінних. Тобто одна функція працює з іншою. Функції від функцій містять у собі

- Знаходження нуля
- Оптимізація
- Інтегрування
- Звичайні диференціальні рівняння

MATLAB представляє нелінійні функції через M-файли. Наприклад, нижче наведена спрощена версія функції ***humps*** з директорії *matlab/demos*

```
function y = humps(x)
y = 1./((x-.3).^2+.01)+1./((x-.9)^2+.04)-6;
```

Наприклад, функція

```
p = fmins( 'humps', .5)

p =
    0.6370
```

визначає положення мінімуму функції **y** ,

а **humps(p)**

```
ans =
   11.2528
```

значення сааї функції **y** точці мінімуму.

```
Q = quad8('humps',0,1)
```

обчислює площу під графіком функції на інтервалі [0,1] за допомогою визначеного інтегралу

```
Q =
   29.8583
```

1.12. Керована графіка

MATLAB надає комплекс функцій низького рівня, які дозволяють створювати й обробляти лінії, поверхні й інші графічні об'єкти. Ця система називається керована графіка (Handle Graphics).

Графічні об'єкти
Графічні об'єкти - це базисні елементи системи керованої графіки в MATLAB. Вони сформовані в дерево структурної ієрархії.

Типи об'єктів керованої графіки:

- Об'єкти ***Root*** є вершиною ієрархії. Вони відповідають екрану комп'ютера. MATLAB автоматично їх створює спочатку сеансу роботи.
- Об'єкти ***Figure*** – це вікна на екрані, крім командного вікна.
- Об'єкти ***Uicontrol*** – це користувальницьке керування інтерфейсом. Коли користувач активує об'єкт, викликається відповідна функція. Вони містять у собі ***pushbutton, radio button*** **і** ***slider***.
- Об'єкти ***Axes*** визначають область у вікні ***Figure*** і орієнтацію дочірніх об'єктів у цій області.
- Об'єкти ***Uimenu*** являють собою меню интерфейсу користувача, що розташовано у верхній частині вікна ***Figure***.
- Об'єкти ***Image*** – це двовимірні об'єкти, які виводить MATLAB, використовуючи елементи прямокутного масиву як індекси в палітрі.
- Об'єкти ***Line*** є основними графічними базисними елементами для більшості двовимірних графіків.
- Об'єкти ***Surface*** – це тривимірне подання матриці, створене шляхом графічного відображення даних як висот над площиною x-y.
- Об'єкти ***Text*** – це рядки символів.
- Об'єкти ***Light*** визначають джерело світла, що діє на всі об'єкти в межах ***Axes***.

Анімація
MATLAB надає кілька способів для створення анімаційної графіки. Використання властивості ***EraseMode*** призначено для довгої послідовності простих графіків, де зміна від кадру до кадру мінімально.

Нижче наведено приклад, що моделює броунівський рух.

Визначимо кількість частинок `n = 20;`

Температуру або швидкість як `s = .02;`

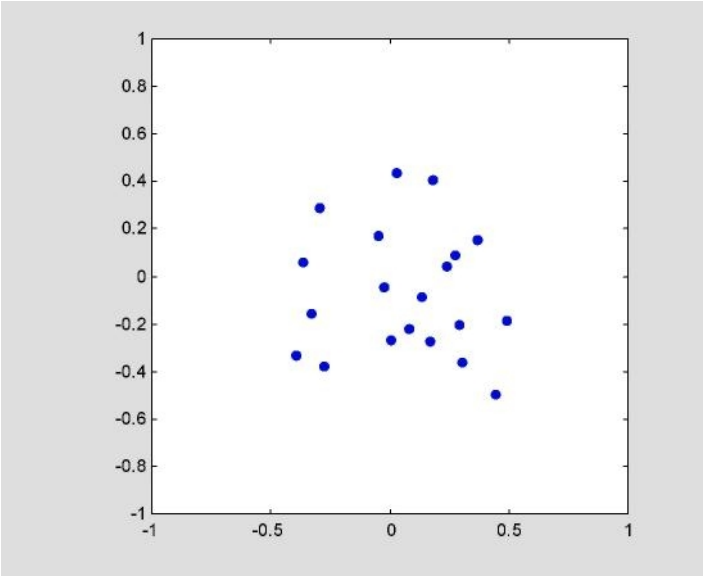
Згенеруємо n випадкових частинок з координатами (x,y) у межах від `-0.5` до `0.5`

```
x = rand(n,1)-
0.5; y =
rand(n,1)-0.5;


h = plot(x ,y,'.');
axis([-1 1 -1 1])
axis square
grid off
set(h,'EraseMode','xor','MarkerSize',18);


while 1
  x = x + s*randn(n,1);
  y = y + s*randn(n,1);
  set(h,'Xdata',x,'Ydata',y)
  end
```

Одна з можливих конфігурацій стану системи броунівськи частинок, наведено у рисунку



1.13. Демонстрація

Щоб побачити додаткові приклади MATLAB, виберіть **Examples and Demos** з меню, або наберіть

```
demo
```

у командному рядку MATLAB. Виберіть з меню приклади і додержуйтеся інструкцій на екрані.

2. Методи моделювання

2.1. Метод класичної молекулярної динаміки

Метод молекулярної динамики – це метод, у якому еволюція системи взаємодіючих частинок описується за допомогою інтегрування кінцевого числа рівнянь руху.

Опису руху окремих частинок системи здійснюється за допомогою законів класичної механіки Н'ютона. Сили міжчастинкової взаємодії описуються за допомогою модельного потенціалу (наприклад, гравітаційної взаємодії).

Рівняння руху частинки:

$$m_i \boldsymbol{r_i}''(t) = \boldsymbol{F_i}, \qquad i=1,2,\ldots,N,$$

де N – кількість частинок, m_i – маса, $\boldsymbol{r_i}(t)$ – радіус-вектор, як функція часу t, *i-oї* частинки, $\boldsymbol{F_i}$ – сума сил, які діють на *i-y* частинку, штрих означає похідну за часом.

Дискретна зміна часу:

$$t_{j+1} = t_j+\Delta t, \qquad j=0,1,2,\ldots,J,$$

де $J\Delta t$ – загальний час спостереження.

Метод полягає в інтегруванні рівнянь руху (як правило, для кінцевого числа частинок):

1. $\boldsymbol{r_i}(t_0)$, $\boldsymbol{v_i}(t_0)$ – початкові умови;
2. $\boldsymbol{r_i}(t_1)= \boldsymbol{r_i}(t_0)+ \boldsymbol{v_i}(t_0)t_0+ \boldsymbol{a_i}\, t_0^{\,2}/2$,
   $\boldsymbol{v_i}(t_1)= \boldsymbol{v_i}(t_0)+\boldsymbol{a_i}\, t_0, \qquad \boldsymbol{a_i}=\boldsymbol{F_i}/m_i$
3. …
4. $\boldsymbol{r_i}(t_{j+1})= \boldsymbol{r_i}(t_j)+ \boldsymbol{v_i}(t_j)t_j+ \boldsymbol{a_i}\, t_j^{\,2}/2$,
   $\boldsymbol{v_i}(t_j)= \boldsymbol{v_i}(t_j)+\boldsymbol{a_i}\, t_j, \qquad \boldsymbol{a_i}=\boldsymbol{F_i}/m_i$

Таким чином, у методі молекулярної динаміки розраховуються класичні (н'ютонівські) траєкторії руху частинок у потенційному полі,

що моделюється. Основу методу становить чисельний розв'язок класичних рівнянь Н'ютона для системи взаємодіючих частинок, причому силу надаються у вигляді

$$\vec{F}_i(\vec{r}) = -\frac{\partial U(\vec{r})}{\partial \vec{r}_i}$$

де $U(\boldsymbol{r})$ – потенційна енергія, що залежить від взаємного розташування частинок.

Після завдання координат і швидкостей всіх частинок у початковий момент часу рівняння руху розв'язуються чисельно шляхом послідовних розрахунків сил, координат і швидкостей частинок за часом.

Температура визначається, як середня кінетична енергія, ящо припадає на один ступінь свободи системи:

$$T(t) = \frac{1}{3Nk_B}\sum_{i=1}^{n} m_i \vec{v}_i^{\,2}, \qquad \vec{v}_i = \frac{d\vec{r}_i}{dt}.$$

При цьому N – повне число ступенів волі молекули, k_b – стала Больцмана.

Потенційна енергія, наприклад, молекули може бути змодельована за допомогою співвідношень:

$$U(\boldsymbol{r}) = U_b + U_{ld} + U_c + U_H,$$

де доданки відповідають різним типам взаємодій (U_b – хімічному; U_{ld} – Леннард-Джонса; U_c – кулонівському; U_H – донорно-акцепторному):

хімічна взаємодія:

$$U_b = \frac{1}{2}\sum_b K_b (r - b_0)^2,$$

де підсумовування проводиться по всіх хімічних зв'язках, b_0 – рівноважна валентна довжина, r – поточна довжина зв'язків, K_b – відповідна силова константа; ван-дер-ваальсова взаємодія атомів, розділених трьома й більше валентними зв'язками, описуються за допомогою потенціалів Леннард-Джонса:

$$U_{ld} = \Sigma\, (A/r_{ij}^{12} - B/r_{ij}^{6}),$$

де A і B – константи взаємодії, r_{ij} – відстані між атомами, підсумовування проводиться по всіх незбіжних індексах, що нумерують атоми системи; кулонівський потенціал має звичайний вигляд $\sim q_i q_j / r_{ij}$ (q_i – заряди частинок системи), а водневий потенціал є аналогічним потенціалу Леннард-Джонса, але є слабкішим за нього та приписується лише донорно-акцепторним парам.

Набори конфігурацій, які було одердано у ході розрахунків методом молекулярної динаміки, розподілені відповідно до деякої статистичної функції розподілу, яка, наприклад, відповідає мікроканонічному розподілу.

Метод молекулярної динаміки не застосовується у випадках, коли

1) длина волны Де Бройля частинки є порівняльною з межчастинковою відстанню;
2) для моделювання систем, які складаються з легких атомів;
3) при низьких температурах, коли визначальними є квантові ефекти;
4) час, протягом якого розглядається поведінка системи, менше, ніж час релаксації фізичних величин, що досліджувалися.

Метод класичної молекулярної динаміки дозволяє розглядати системи, що складаються з десятків тисяч атомів на протязі часу порядку сотен наносекунд.

Одна з перших робіт з моделювання методом молекулярної динаміки була присвячена дослідженню фазової діаграми системи твердих сфер. Метод молекулярної динаміки значно поширився у біохімії і біофізиці, зокрема, у дослідженні білка та деталізації його властивостей.

2.2. Метод Монте-Карло

Метод Монте-Карло – метод статистичних випробувань, чисельний метод розв'язку математичних проблем за допомогою моделювання випадкових процесів і подій. Термін виник у 1949 при моделюванні процесів розділення урану, хоча деякі розрахунки шляхом моделювання випадкових подій здійснювалися статистиками й раніше. Назва походить від міста Монте-Карло, відомого своїм ігорним домом. Значне поширення метод отримав після появи швидкодіючих обчислювальних машин, тому що програми для розрахунків порівняно прості й, як правило, дозволяють обходитися без великої оперативної пам'яті.

Спочатку метод Монте-Карло використовувався для розв'язку складних задач теорії переносу випромінювання й нейтронної фізики, де традиційні чисельні методи виявилися мало придатними. Потім його було поширено на великий клас проблем статистичної фізики, дуже різних за своїм змістом. Метод застосовується для розв'язку задач теорії ігор, теорії масового обслуговування і математичної економіки, теорії передачі повідомлень при наявності перешкод і т.д.

Для розв'язку *детермінованої проблеми*, насамперед, розробляють ймовірну модель, представляють величину, яка вишукується, наприклад, багатомірний інтеграл, у вигляді математичного очікування функціонала від випадкового процесу, який далі моделюється на ЕОМ. Відомі ймовірні моделі для обчислення інтегралів, для розв'язку інтегральних рівнянь 2-го роду, для систем лінійних алгебраїчних рівнянь, для крайових проблем для еліптичних рівнянь, для оцінки власних значень лінійних операторів і т.д.

Моделювання випадкових величин із заданими розподілами здійснюється шляхом перетворення одного або декількох незалежних значень випадкового числа *a*, яке розподілено рівномірно у інтервалі *(0,1)* – так званий, генератор випадкових чисел, який реалізовано, зокрема, у системі MATLAB (оператори rand і randn). Такі числа називаються "псевдовипадковими", вони перевіряються статистичними тестами й розв'язками типових задач.

Якщо в розрахунку моделюються випадкові величини, які обумовлені реальним змістом явища, розрахунок виявляється процесом "прямого моделювання". Такий розрахунок є неефективним, якщо вивченню підлягають рідкі події, тому що реальний процес містить про них обмаль інформації. Ця неефективність звичайно виявляється в занадто великій ймовірній погрішності (дисперсії) випадкових оцінок. Способи зменшення дисперсії зазначених оцінок, як правило, засновані на модифікації моделювання за допомогою інформації про "функції цінності" значень випадкових величин щодо величин, які обчислюються.

Пряме моделювання (метод прямого статистичного моделювання Монте-Карло) – метод обчислювальної газодинамики, призначений для розв'язку задач динаміки розріджених газів (найчастіше – рівняння Больцмана). Ідея методу полягає у моделюванні реальної поведінки частинок розрідженого газу за допомогою великої кількості (до 10^7) моделюючих молекул. Одна моделююча частинка описує дуже велику кількість реальних молекул. Розрахункова область поділяється на зіштовхувальні осередки, у яких газ є «замороженим», а характерним є час «вільного пробігу» осередку. Після опису руху зіштовхувальних осередків далі у кожному осередку розглядаються внутрішні зіткнення між молекулами. При цьому в найпростішому випадку зазначається, що різні осередки не обмінюються молекулами газу. Вперше метод прямого статистичного моделювання, який використовує розщеплення по процесах зіткнення й переносу молекул був запропонований Г.Бердом (1963). Розвинутий підхід у цей час називається методом Берда Non-Time Counter.

Застосування методу обмежується випадком превалювання подвійних зіткнень молекул газу над багаточастинковими. Тому метод застосовується для опису течії газу у вільно-молекулярному, перехідному й континуальному режимах. Наприклад, повітря задовольняє цим умовам аж до тиску в сотні атмосфер. Режим течії визначається за допомогою числа Кнудсена $K_n = l / d$, де l – довжина вільного пробігу, d – діаметр молекули. Таким чином, метод Монте-Карло може застосовуватися для великих чисел Кнудсена. Інше обмеження на застосовність методу пов'язане з порушенням умови про молекулярний хаос, яка використовується під час обгрунтування рівняння Больцмана. Виникнення статистичної залежності між

моделюючими молекулами приводить до необхідності збільшення числа моделюючих молекул. При K_n < 0.01 цей фактор змушує використовувати паралельні рорахункові методи.

Безкоштовний виклад методу Монте-Карло у проблемах статистичної фізики й фізичної кінетики знаходитиметься в Інтернет-мережі за наступною адресою:
http://www.npac.syr.edu/users/paulc/lectures/montecarlo/p_montecarlo.html,
а безкоштовна симуляція методу – за адресою:
http://www.people.nnov.ru/fractal/perc/Monte_r.htm

2.3. Мікроканонічний ансамбль

2.2.1. Ансамблі й розподіли

Мікроканонічний ансамбль – статистический ансамбль для ізольованих (не обмінюються енергією з навколишніми тілами) макроскопічних систем у постійному об'ємі для постійного числа частинок. Енергія макроскопічної системи є постійною. Поняття мікроскопічного ансамблю вперше застосовано Дж. У. Гиббсом в 1901 і є ідеалізацією, тому що в дійсності повністю ізольованих систем не існує.

У класичній статистиці статистичний ансамбль характеризується функцією розподілу $f(q_i, p_i)$, яка залежить від координат q_i і імпульсів p_i всіх частинок системи. Ця функція визначає ймовірність мікроскопічного стану системи, тобто ймовірність того, що координати і імпульси частинок системи мають певні значення. Відповідно до мікроканонічного розподілу Гиббса, всі мікроскопічні стани, які відповідають існуючій енергії, рівноймовірні. Звичайно, що значення енергії системи може бути реалізоване при різних значеннях координат і імпульсів частинок системи.

Якщо позначити енергію системи у залежності від координат і імпульсів (функцію Гамильтона) $H(q_i, p_i)$, а її енергію E, маємо

$$f(q_i, p_i) = A\,\delta\{H(q_i, p_i) - E\} \; -$$

мікроканонічний розподіл Гиббса, де $\delta\{z\}$ – дельта-функція Дирака, стала A визначається умовою нормівки (сумарна ймовірність перебування системи у всіх можливих станах, обумовлена інтегралом від $f(q_i, p_i)$ по всім q_i, p_i, дорівнює 1) і залежить від об'єму та енергії системи.

У квантовій статистиці розглядається ансамбль енергетично ізольованих квантових систем (з постійними об'ємом V і повним числом частинок N), які мають однакову енергію E з точністю до $\Delta E << E$. Передбачається, що для таких систем усі квантовомеханичні стани з енергією у інтервалі $[E, E + \Delta E]$ рівноймовірні. Такий розподіл ймовірностей w станів системи, коли

$$w(E_k) = \begin{cases} \Omega^{-1}(E,N,V) & \\ 0 & E \le E_k \le E + \Delta E \end{cases},$$

називається мікроканонічним розподілом. Тут w – статистический вес, обумовлений умовою нормівки $\Sigma\, w(E_k) = 1$ (підсумовування відбувається по всіх можливих значеннях енергії), і дорівнює числу квантових станів у інтервалі $[E, E + \Delta E]$. Величину ΔE звичайно обирають малою, але кінцевою (тому, що точна фіксація енергії відповідно до співвідношення невизначеностей для енергії і часу, зажадала б нескінченого часу спостереження). Однак мікроканонічний ансамбль є малочутливим до вибору ширини енергетичного інтервалу ΔE, якщо вона значно менше повної енергії системи.

За допомогою статистичної ваги $w(E,N,V)$ обчислимо ентропію S системи:

$$S = k_b \ln w(E, N, V)$$

(k_b – стала Больцмана) і інші термодинамические потенціали.
Оскільки ентропія системи пропорційна числу частинок N, статистична вага має порядок величини експоненціальної функції від N і для розглянутих макроскопічних систем є дуже великою.

Мікроканонічний розподіл виявляється незручним для практичного застосування, тому що для обчислення статистичної ваги потрібно знайти розподіл квантових рівнів системи, яка складається з великої кількості частинок, є дуже складною проблемою. Зручніше розглядати не енергетично ізольовані системи, а системи, які перебувають у тепловому контакті та рівновазі з навколишнім середовищем, температура якого вважається постійною, і застосовувати канонічний *розподіл Гиббса*.

Теорема Гиббса: частина мікроскопічного ансамблю розподілена в ансамблі канонічно. Доказ теорема полягає в обґрунтуванні канонічного розподілу Гиббса, якщо мікроканонічний розподіл вважати постулатом статистичної фізики.

Для систем, які перебувають у тепловій рівновазі з навколишнім середовищем з постійною температурою (термостат), виконується *канонічний розподіл Гиббса*. Відповідно до цього розподілу, ймовірність певного мікроскопічного стану пропорційна функції розподілу f (q_i, p_i), яка залежить від координат q_i і імпульсів p_i частинок системи:

$$f(q_i, p_i) = Ae^{-H(q_i, p_i)/kT},$$

де H (q_i, p_i) – функція Гамильтона системи (повна енергія, виражена через координати і імпульси частинок), k_b – стала Больцмана, T – абсолютна температура; стала A не залежить від q_i і p_i і визначається умовою норміровки (сума ймовірностей перебування системи у всіх можливих станах дорівнює одиниці). Таким чином, ймовірність мікростану визначається відношенням енергії системи до величини $k_b T$ (яка є мірою інтенсивності теплового руху молекул) і не залежить від конкретних значень координат і імпульсів частинок, що реалізують дане значення енергії.

У квантовій статистиці ймовірність w_n даного мікроскопічного стану визначається значенням енергетичного рівня системи E_n.

$$w_n = Ae^{-E_n/kT}.$$

Для ідеального газу, тобто газу, у якому енергією взаємодії частинок можна зневажити, канонічний розподіл Гиббса переходить у розподіл Больцмана. З функцією розподілу, що характеризує ймовірність перебування частинки в даному стані, що, наприклад, для ідеального газу молекул, що перебувають у поле зовнішніх сил, має вигляд:

$$f(p,r) = A\exp\left(-\frac{p^2/2m + U(r)}{kT}\right),$$

де $\boldsymbol{p}^2/2m$ – кінетична енергія молекули маси m, $U(r)$ – потенційна енергія у зовнішнім полі.

Якщо система ізольована, то її енергія постійна; у цьому випадку виконується мікроканонічний розподіл Гиббса, відповідно до якого всі мікроскопічні стани ізольованої системи рівноймовірні.

У відповідності до загальноприйнятого у статистичній фізиці підходу, розглянемо замкнену систему, яка складається з N частинок. Оберемо макроскопічними характеристиками системи об'єм V і повну енергію системи E, які будемо вважати постійними. До того ж припустимо, що система є ізольованою, тобто впливом на неї зовнішніх факторів можна знехтувати. Замкнені макроскопічні системи, як відомо, прагнуть до стаціонарного рівноважного стану з максимальною ентропією. Відзначимо, що якщо на макроскопічному рівні система характеризується трьома величинами N, V, E, те на мікроскопічному рівні існує величезне число конфігурацій, що реалізують заданий макростан*(N, V, E)*.

Таким чином, обмеження на існування мікростану полягає у зберіганні числа частинок, об'єму, повної енергі. Отже система в будь-який момент часу може виявитися з рівною ймовірністю в одному з можливих мікростанів (*постулат про рівність ймовірностей*).

Для визначення середніх значень фізичних величин, які характеризують макроскопічну систему, використовують два способи: по мікростанах і за часом. Гіпотеза про збіг середніх значень величин, що обчислюють усередненням за часом і усередненням по ансамблям, що реалізуюься, називається *эргодичною гіпотезою.*

2.2.1. Моделювання мікроканонічного ансамблю

Загальна ідея методу Монте-Карло для мікроканонічного ансамблю полягає в отриманні представницької вибірки з повного числа мікростанів і оцінці ймовірності. Для цього використовують досить прозору процедуру: потрібно зафіксувати повне число частинок N, об'єм V і повну енергію системи E, випадково змінюючи координати й швидкості окремих частинок, і розглядати мікростани, що „виникають".

Однак більш ефективною виявляється процедура розрахунків методом Монте-Карло для мікроканонічного ансамблю, яка полягає в розділенні вихідної макроскопічної системи на дві підсистеми: вихідну, яку надалі будемо називати „системою", і підсистеми, яка складається з одного елемента, названого авторами моделі *демоном* (за аналогією з демоном Максвелла). Роль демона аналогічна ролі члена кінетичної енергії в методі молекулярної динаміки. Обходячи елементи системи й передаючи енергію, він забезпечує зміну конфігурації системи. Якщо енергії у „мантії демона" виявляється достатньо, він віддає енергію тому елементу системи, якому вона потрібна для здійснення зміни конфігурації. І навпаки, якщо для зміни конфігурації потрібно зменшити енергію системи, енергія передається частинкою демонові. Єдине обмеження полягає в тому, що енергія демона повинна бути позитивною.

Алгоритм процедури демона:

1) Генерація початкової конфігурації системи із заданим параметром повної енергії.
2) Вибір випадковим образом частинки (демона) з певною енергією й пробна зміна її координат.
3) Обчислення повної енергії системи у новому стані.
4) Збереження нового стану при зменшенні або збереженні енергії.
5) Збереження нового стану при збільшенні енергії, якщо енергії демона достатньо для передачі її системі. У противному випадку нова конфігурація не зберігається, і частинка не змінює координати.
6) Повторення алгоритму до одержання репрезентативної вибірки мікростанів і переходу системи у рівноважний стан (з енергією, що зберігається).

Приклад. Одномірний ідеальний газ

Застосуємо описаний алгоритм до класичного одновимірного ідеального газу, швидкості частинок якого безперервні й необмежені, а енергія частинок не залежить від положення частинок, тому повна енергія частинок є сумою кінетичних енергій окремих частинок. Для зміни конфігурації будемо випадково змінювати швидкість випадково обраної частинки.

Для реалізації описаного алгоритму в пакеті MATLAB створимо два m-файли:

1) файл Init.m, що містить опис функції, яка повертає абсолютні значення швидкостей у момент часу t = 0;
2) файл Demon.m, що містить опис функції, яка повертає середні по ансамблю реалізацій значення: енергії демона (Edave), середньої енергії системи на одну частинку (Esysave), середньої швидкості на одну частинку системи (Vave), середнього числа прийняття (Accept).

```
 % листинг файлу Init.m
 function z = Init(N,Esystem)
 % функція, що повертає абсолютні значення
швидкостей у момент
% часу t = 0
 % N число частинок системи
 % Esystem енергія системи
 V=(Esystem./N).^0.5;
 for i=1:N
   z(i)=V;
 end;
 % листинг файлу Demon.m
 function [Edave,Esysave,Vave,Accept] =
Demon(N,Esystem,NTrial,Vel,d)
 % функція, що повертає усереднені по ансамблю
реалізацій
% значення: енергії демона (Edave), середньої
енергії системи на одну
% частинку (Esysave), середньої швидкості на одну
частинку системи (Vave),
% середнього числа прийняття (Accept).
Edemon=0;
 Vtot=sum(Vel);
```

```
 Vcum=0;
 Escum=0;
 Edcum=0;
 Accept=0;
 for j=1:NTrial
   for i=1:N
      dv=(2*rand(1) -1)*d; % випадкова зміна
швидкості
      Ip=floor(N*rand(1) +1); % випадковий вибір
частинки
      VTrial=Vel(Ip)+dv; % пробна зміна швидкості
      de=0.5*(VTrial.^2-Vel(Ip)^2); % пробна зміна
енергії
      if de<=Edemon % якщо енергія зменшується
зміну приймається
         Vel(Ip)=VTrial;
         Vtot=Vtot+dv;
         Accept=Accept+1;
         Edemon=Edemon-de;
         Esystem=Esystem+de;
      end;
   Edcum=Edcum+Edemon;
   Escum=Escum+Esystem;
   Vcum=Vcum+Vtot;
   end;
 end;
 Edave=Edcum/(NTrial*N);
 Accept=Accept/(NTrial*N);
 Esysave=1/N*Escum/(NTrial*N);
 Vave=1/N*Vcum/(NTrial*N);
```

Далі необхідно виконати наступну послідовність команд:

```
<STRONG>>>Esystem=40;
% енергія системи

 <STRONG>>>N=40;
% число частинок системи

 <STRONG>>>NTrial=4000;
```

```
% число випробувань

 <STRONG>>>d=2*2^0.5;
% максимальне значення зміни швидкості

 <STRONG>>>Vel=Init(N,Esystem);
% завдання початкових швидкостей

 % обчислення характеристик стану ідеального газу
 <STRONG>>>[Edave Esysave Vave
Accept]=Demon(N,Esystem,NTrial,Vel,d)

 Edave =
    0.9566
 Esysave =
    0.9761
 Vave =
    1.5695e-004
 Accept =
    0.5119


 % листинг файлу Demon.m
 function z = Demon(N,Esystem,NTrial,Vel,d)
 % функція, що повертає миттєві значення енергії
демона
 % N число частинок системи
 % Esystem енергія системи
 % NTrial число випробувань
 % Vel вектор, який повернено функцією Init
 % d максимальна зміна швидкості
 Edemon=0;
 z(1) =Edemon;
 k=1;
 for j=1:NTrial
   for i=1:N
      dv=(2*rand(1) -1)*d;
     % випадкова зміна швидкості
      Ip=floor(N*rand(1) +1);
     % випадковий вибір номера частинки
      Vtrial=Vel(Ip)+dv;
      de=0.5*(Vtrial.^2-Vel(Ip).^2);
```

```
      % пробна зміна енергії

     if de<=Edemon
     % якщо енергія зменшується, пробна зміна
приймається
          Vel(Ip)=Vtrial;
          Edemon=Edemon-de;
       end;
       k=k+1;
       z(k)=Edemon;
    end;
 end;
```

Далі необхідно виконати наступну послідовність команд:

```
 >> Esystem=40; % енергія системи
 >> N=40; % число частинок системи
 >> NTrial=500; % число випробувань
 >> d=2*2^0.5; % максимальна зміна швидкості

 >> Vel=Init(N,Esystem);
     % обчислення початкових швидкостей частинок
 >> Ed=Demon(N,Esystem,NTrial,Vel,d);
     % обчислення миттєвих значень енергії демона
     % візуалізація залежності миттєвих значень
енергії
     % демона від часу
 >> i=1:length(Ed);
 >> figure(1);plot(i,Ed,'k');axis([0 20000 0 8])
     % обчислення й візуалізація розподілу
ймовірностей
 >> x1=min(Ed);
 >> x2=max(Ed);
 >> Nint=50;
 >> i=1:Nint;
 >> x(i)=x1+(x2-x1)/(Nint-1)*(i-1);
 >> h=hist(E,x);
 >> figure(2);bar(x,h); colormap white
     % обчислення й візуалізація функції, що
апроксимує
     % розподіл
 >> F10=inline('u(1) *exp(-u(2) *z)','u','z')
```

```
    % завдання апроксимуючої функції
 F10 =
    Inline function:
    F10(u,z) = u(1) *exp(-u(2) *z)
 >> beta=nlinfit(x,h,F10,[3000 1]);
    % обчислення коефіцієнтів апроксимуючої
функції
    % візуалізація гістограми послідовності
миттєвих
    % значень енергії демона й апроксимуючої
функції
 >> Ni=500;
 >> i=1:Ni;
 >> X(i)=x1+(x2-x1)/(Ni-1)*(i-1);
 >> figure(3);bar(x,h); colormap white
 >> hold on
 >> plot(X(i),F10(beta,X(i)),'k')
 >> hold off
```

Завдання 1. Проведіть чисельні розрахунки для заданих значень числа частинок (*N=10*, *20*, *40*, *80)* і різних значень відносини $K=d\sqrt{N}/E_{system}$. Побудувати залежності коефіцієнта прийняття *Accept* від *K* при зазначеному числі частинок. Використовуючи метод найменших квадратів, підберіть аналітичне співвідношення, яке описує залежність коефіцієнта прийняття *Accept* від *K*. Коли значення коефіцієнта прийняття виявляється ≈ 0.5?

Завдання 2. Застосуємо кінетичне визначення температури, у відповідність із яким, температура визначається зі співвідношення для середньої кінетичної енергії на одну частинку й температурою газу, яка вимірюється в одиницях k_b

$$\frac{1}{2}m\left\langle v^2\right\rangle = \frac{1}{2}T_{kin},$$

Використовуючи співвідношення, отримайте T_{kin}. Як пов'язані значення температури і енергії демона?

Завдання 3. Змоделюйте канонічний ансамбль методом Монте-Карло, зберігаючи на кожному кроці значення енергії демона. Для знайденої

реалізації значень енергії демона отримайте розподіл ймовірностей $P(E_d)\Delta E$. Визначити функціональну залежність $P(E_d)\Delta E$ від температури.

Завдання 4. Дослідити залежність температури, яку розраховано за допомогою мікроканонічного й канонічного ансамблів від числа частинок системи N і числа кроків методу Монте-Карло.

Температура ідеального газу, яку визначено, як середню кінетичну енергію, що припадає на одну частинку, T_{kin} збігається з температурою термостата T_{term}. Таким чином, демон перебуває в термодинамічній рівновазі з термостатом.
Розподіл ймовірності називається розподілом Больцмана або канонічним розподілом.

2.4. Модель Ізінга

2.4. 1. Одновимірна модель Ізінга

Однієї з найпростіших моделей, які використовуються у статистичній фізиці для моделювання фазових переходів у магнітних речовинах або бінарних сполуках, є *модель Ізінга*. Дана модель ставиться до широкого класу граткових моделей, у яких розглядаються локальні взаємодії, тобто взаємодії між найближчими вузлами решітки. У магнітних системах локальні взаємодії обумовлені *спінами*, розташованими у вузлах гратки або решітки. Спіни можуть являти собою, наприклад, магнітні моменти атомів у твердому тілі, взаємодіючі один з одним і зовнішнім магнітним полем.

Розглянемо решітку, що складається з N вузлів. Зв'яжемо з кожним *i-м* вузлом решітки число $s_i=\pm1$, що характеризує напрямок магнітного моменту системи, де $s_i=1$, якщо спін орієнтовано у позитивному напрямку осі OZ, і $s_i=-1$, якщо спін орієнтований у негативному напрямку осі OZ. (Розглянута картина характерна для частинок з напівцілим спіном, хоча далі ми розглядаємо спіни як класичні ступені волі й не вводимо для них використовуваних у квантовій механіці правил комутації кутового моменту).

Будь-який мікростан решітки задається набором змінних $\{ s_1, s_2,..., s\}$. Тому що макроскопічне властивості системи визначаються

властивостями її досяжних мікростанів, необхідно визначити залежність енергії E від конфігурації спінів.

Повна енергія за наявністю магнітного поля h у моделі Ізінга дорівнює

$$E = -J\sum_{\langle i,j\rangle}^{N} s_i s_j - h\sum_{i=1}^{N} s_i \ ,$$

де кутові дужки $<i,j>$ означають, що сума береться по всіх найближчих сусідніх парах спінів, константа обмінного зв'язку J характеризує силу взаємодії сусідніх спінів:

$$\begin{array}{cc} \uparrow\uparrow \quad \downarrow\downarrow & \uparrow\downarrow \quad \uparrow\downarrow \\ E = -J & E = +J \end{array}$$

Якщо $J>0$, то в стані ↑↑ й ↓↓, тобто при однаковій орієнтації спінів найближчих сусідів, енергетично вигідніше станів ↑↓і ↓↑, у яких сусідні спіни орієнтовані в протилежні сторони. Отже, очікується, що для $J>0$ стан з найменшою повною енергією є феромагнітним, тобто всереднє число спінів, які зорієнтовано в одному напрямку, не дорівнює нулю. Якщо $J<0$, то з енергетичної точки зору більше кращими виявляються стани ↑↓й ↓↑, для яких сусідні спіни антипаралельні. Отже, середнє число спінів, зорієнтованих в одному напрямку, дорівнює нулю, тобто спіни впорядковані через один (антиферомагнітний стан).

При накладенні зовнішнього магнітного поля, спрямованого паралельно осі OZ, спіни ↑ й ↓ здобувають додаткову внутрішню енергію, рівну $-h$ і $+h$, відповідно.

Зазначимо основні припущення, які покладені в основу моделі Ізінга:

1) кінетична енергія вузлів решітки приймається такою, що дорівнює нулю;
2) енергія взаємодії враховує лише внески від найближчих сусідів;
3) спіни володіють тільки двома дискретними станами.

Тому що надалі нас будуть цікавити термодинамічні характеристики даної системи, виявляється зручним вимірювати параметри J і h в одиницях температури. Можливі конфігурації системи визначаються

визначенням усіх спінівих змінних, число яких становить *2N*, а внесок кожної з *2N* спінівих конфігурацій визначається функцією розподілу Гиббса для канонічного ансамблю

$$w(s) = \frac{e^{-E(s)}}{Z}, \qquad Z(J,h) = \sum_{i}^{N} e^{-E(s_j)}$$

Алгоритм демона може бути застосований до дослідження моделі Ізінга методом мікроканонічного ансамблю. Для цього необхідно одержати вираження, що зв'язує енергію демона й температуру термостата. Нагадаємо, що в безперервному випадку розподіл енергії демона підкорюється формулі Больцмана. Припустимо, що даний розподіл ймовірностей справедливо для будь-якої макроскопічної системи, що перебуває в стані термодинамічної рівноваги. Тоді $\langle E_d \rangle$ дорівнює

$$\left\langle E_d \right\rangle = \frac{\sum E_d e^{-\frac{E_d}{T}}}{\sum e^{-\frac{E_d}{T}}},$$

де суми обчислюються по всіх можливих значеннях E_d.

Мінімальна ненульова втрата енергії системи в нульовому магнітному полі становить *2s*, де *s* – сумарний спін найближчих сусідів перекидна спіна. В одномірному випадком сумарний спін найближчих сусідів дорівнює *0*, або *2*, тобто мінімальна ненульова втрата енергії рівняється *2J*. Отже, енергія демона, може рівнятися *0*, *2J*, *4J*, ... Якщо ввести позначення $x=2J/T$, те останнє вираження для нескінченної решітки можна представити у вигляді

$$\left\langle \frac{E_d}{T} \right\rangle = \frac{\sum_{n=0}^{\omega} (x \cdot n) e^{-n x}}{\sum_{n=0}^{\omega} e^{-n x}}$$

Нескінченні суми у чисельнику та знаменнику можуть бути розраховані аналітично або чисельно. Можна переконатися, що розв'язки наведеного рівняння відносно температури визначатимуться співвідношенням

$$T = \frac{2J}{\ln(1 + 2J/\langle E_d \rangle)}.$$

Для моделювання одновимірної моделі Ізінга методом мікроканонічного ансамблю в пакеті MATLAB необхідно створити файл *Ising.m*, що містить опис функції, яка повертає значення повної енергії системи, енергії демона, намагніченості та середнє число прийняття.

```
 % листинг файлу Ising.m

 function
[Es,Ed,Sp,Accept]=Ising(Nspin,J,h,Esi,NTrial)

 % функція, що повертає миттєві значення сповненої
енергії
 % (Es), енергії демона (Ed), намагніченість (Sp)
і середнє
 % число прийняття (Accept)
 % Nspin число спінів системи
 % J константа обмінної взаємодії
 % h зовнішнє магнітне поле
 % Esi кінцева енергія системи
 % NTrial число випробувань
 % завдання конфігурації спінів у момент часу t =
0

for i=1:Nspin
  s(i)=1;
 end;
 M=Nspin;
 Esystem=-(J+h)*Nspin;
 Edemon=2*J*ceil((Esi-Esystem)/(2*J));
 Es(1) =Esystem; % енергія системи в момент часу t
= 0
```

```
 Ed(1) =Edemon;  % енергія демона в момент часу t
= 0
 Sp(1) =M;  % магнітний момент системи в момент
часу t = 0

 Accept=0;
 k=1;

 % реалізація методу мікроканонічного ансамблю
 for i=1:NTrial
  for j=1:Nspin
    Ispin=floor(Nspin*rand(1) +1);
    % випадковий вибір номера спіна
    % періодичні граничні умови
    if Ispin==1
      Left=s(Nspin);
    else
      Left=s(Ispin-1);
    end;
    if Ispin==Nspin
      Right=s(1);
    else
      Right=s(Ispin+1);
    end;
    de=2*s(Ispin)*(-h+J*(Left+Right));
    % пробна зміна енергії спіна
    if de<=Edemon
    % прийняття пробної зміни
      s(Ispin)=-s(Ispin);
      Accept=Accept+1;
      Edemon=Edemon-de;
      Esystem=Esystem+de;
    end;
    k=k+1;
    Es(k)=Esystem;
    Ed(k)=Edemon;
    Sp(k)=sum(s);
  end;
 end;
 Accept=Accept/(NTrial*Nspin);
```

Далі необхідно виконати наступну послідовність команд:

```
 >> Nspin=200;   % число спінів системи
 >> J=1;         % константа обмінної взаємодії
 >> h=0;         % зовнішнє поле
 >> Esi=-10;          % завдання кінцевого значення
енергії
 >> NTrial=100;  % число випробувань

% обчислення миттєвих значень енергії системи(Es),
% енергії демона (Ed), миттєвої намагніченості
(Sp), числа
% прийняття рішень

 >> [Es Ed Sp Accept]=Ising(Nspin,J,h,Esi,NTrial);
        % візуалізація залежностей миттєвих
значень енергії
        % системи й енергії демона від часу
        >> i=1:1200;
        >> figure(1);plot(i,Es(i),'k');
        >> figure(2);plot(i,Ed(i),'k');

% обчислення розподілу ймовірностей миттєвих
значень
% енергії демона
 >> Nint=50;
 >> i=1:Nint;
 >> x1=min(Ed);
 >> x2=max(Ed);
 >> x(i)=x1+(x2-x1)/Nint*i;
 >> h=hist(z2,x);
 % знаходження і візуалізація функції, що описує
розподіл
 >> F10=inline('u(1) *exp(-u(2) *z)','u','z')

 F10 =
     Inline function:
     F10(u,z) = u(1) *exp(-u(2) *z)

 >> beta=nlinfit(x,h,F10,[10000 0.05]);
 >> bar(x,h);colormap white
 >> hold on
 >> Ni=500;j=1:Ni;
 >> X(j)=x1+(x2-x1)/Ni*j;
```

```
>> plot(X,F10(beta,X),'k')
>> hold off

>> mean(Es)/Nspin
   % середня енергія системи на один спін
ans =
  -0.0941

>> mean(z3)
   % середня намагніченість системи
ans =
  1.4352

>> mean(z3)/Nspin
   % середня намагніченість на один спін
ans =
  0.0072

>> 2/log(1+2./mean(z2))
ans =
  9.7904
```

Завдання 5. Обчислите E_{system} для систем, що складаються з різного числа спінів N. Як отриманий результат залежить від числа спінів N і кількості кроків методу Монте-Карло? Порівняти отримані результати з точною відповіддю для нескінченної одномірної решітки

$$\frac{\left\langle E_{system} \right\rangle}{N} = -th\left(\frac{J}{T}\right).$$

Завдання 6. Для ненульового магнітного поля h обчислите $\langle E_{demon} \rangle$, як функцію h і T для заданої повної енергії. Порівняти значення рівноважних температур для випадків $h=0$, $h\neq 0$ при однакових значеннях повної енергії.

2.4.2. Двовимірна модель Ізінга

Алгоритм моделювання двовимірної системи N спінів методом Монте-Карло:

1. Визначення числа спінів решітки N_{spin}.
2. Визначення числа кроків методу Монте-Карло на спін.
3. Визначення початкової конфігурації орієнтації спінів у вузлах квадратної решітки в момент часу t=1.
4. Випадковой вибір одного зі спінів системи.
5. Обчислення пробної зміни енергії.
6. Збереження конфігурації при зменшенні або збереженні енергії системи.
7. Збереження конфігурації при збільшенні енергії системи, якщо пробний спін (демон) має більшу власну енергію в порівнянні зі зміною енергії системи.
8. Повторення пунктів 4-7 (по числу спінів у системі).

Для реалізації описаного алгоритму моделювання двовимірної системи спінів методом Монте-Карло в пакеті MATLAB створимо файл *Ising2.m*, що містить опис функції, що повертає миттєві значення: енергії системи, енергії демона, намагніченості, числа прийняття рішень, а також миттєві конфігурації спінів.

```
 % листинг файлу Ising2.m

function [Es,Ed,Sp,A,S] =
Ising2(Nspin,J,h,Esi,NTrial)

 % функція, що повертає миттєві значення: енергії
системи,
 % енергії демона, намагніченості, числа прийняття
рішень,
 % а також миттєві конфігурації спінів
 % Nspin число спінів решітки
 % J константа обмінної взаємодії
 % h напруженість зовнішнього магнітного поля
 % Esi кінцева енергія системи
 % NTrial число випробувань

 Ns=Nspin.^0.5;
```

```
   % число спінів уздовж однієї сторони
 s=ones(Ns,Ns);
   % початкова конфігурація спінів
 Esystem=-(J+h)*Nspin;
   % початкова енергія системи
 Edemon=4*J*floor((Esi-Esystem)/(4*J));
   % початкова енергія демона

Es(1) =Esystem;
Ed(1) =Edemon;
S=s;
k=1;
 for i=1:NTrial
  Accept=0;
  for j=1:Nspin
     % випадковий вибір вузла сітки
     Ix=floor(Ns*rand(1)+1);
     Iy=floor(Ns*rand(1)+1);
     % граничні умови
     if Ix==1
        Left=Ns;
     else
        Left=Ix-1;
     end;
     if Ix==Ns
        Right=1;
     else;
        Right=Ix+1;
     end;
     if Iy==1
        Down=Ns;
     else;
        Down=Iy-1;
     end;
     if Iy==Ns
        Up=1;
     else
        Up=Iy+1;
     end;
     % пробна зміна енергії
de=2*s(Iy,Ix)*(-
h+J*(s(Iy,Left)+s(Iy,Right)+s(Down,Ix)+
```

```
s(Up,Ix)));

    if de<=Edemon % прийняття пробної зміни
енергії
        s(Iy,Ix)=-s(Iy,Ix);
        Accept=Accept+1;
        Edemon=Edemon-de;
        Esystem=Esystem+de;
     end;
     k=k+1;
     Es(k)=Esystem;
     Ed(k)=Edemon;
     A(k-1)=Accept;
     s1=sum(s);
     Sp(k)=sum(s1);
     S=cat(3,S,s);
  end;
 end;
 A=A/NTrial;
```

Далі необхідно виконати наступну послідовність команд:

```
 >> Nspin=81;    % число спінів системи
 >> J=1;     % константа обмінної взаємодії
 >> h=0;         % напруженість зовнішнього
магнітного поля
 >> Esi=-10;     % кінцева енергія системи
 >> NTrial=50;   % число випробувань
 >> [Es,Ed,Sp,A,S]=Ising2(Nspin,J,h,Esi,NTrial);
            % візуалізація залежності миттєвих
         значень
            % сповненої енергії системи й енергії
            демона % від часу

 >> i=1:2000;
 >> figure(1);plot(i,Es(i))
 >> figure(2);plot(i,Ed(i))
 % візуалізація миттєвих конфігурацій системи у
вигляді
 % векторного поля
 >> i=1:9;
```

```
 >> j=1:9;
 >> V(i,j)=0;
 >> U(i,j)=0;
 >> Z(i,j)=0;
 >> figure(3);quiver3(Z,U,V,S(:,:,1)); colormap
white
 >> figure(4);quiver3(Z,U,V,S(:,:,100)); colormap
white
 >> figure(5); quiver3(Z,U,V,S(:,:,1000));
colormap white
 >> figure(6); quiver3(Z,U,V,S(:,:,2000));
colormap white

% візуалізація миттєвих конфігурацій системи у
вигляді
% прямокутних паралелепіпедів

  >> figure(7);bar3(z5(:,:,1));axis([0 10 0 10 -
1.5 1.5]);
     colormap white
  >> figure(8);bar3(z5(:,:,100));axis([0 10 0 10 -
1.5 1.5]);
     colormap white
 >> figure(9);bar3(z5(:,:,1000));axis([0 10 0 10 -
1.5 1.5]);colormap white
 >> figure(10);bar3(z5(:,:,2000));axis([0 10 0 10
-1.5 1.5]);colormap white
 >> mean(Es); % середня енергія системи
 ans =
   -14.1790
 >> mean(z1)/Nspin; % середня енергія системи на
один спін
 ans =
   -0.1750
 >> mean(z3)/Nspin % середня намагніченість на
один спін
 ans =
   0.6627
 >> mean(A) % середнє значення коефіцієнта
прийняття на один крок
 ans =
   0.1463
```

```
 >> 4/log(1+4/mean(z2)) % рівноважна температура
системи
 ans =
   2.7028
```

Завдання 7. Покажіть, що розподіл енергії демона описується функцією розподілу Больцмана.

2.5. Метод точкових відображень

Поняття про точкові відображення. Усякий коливальний процес є еволюційним у часі. Природними, такими, що використовуються сторіччями, для опису таких процесів служили (і служать) диференціальні рівняння. Однак останнім часом для тих же цілей стали використати іншу альтернативний підхід – *дискретні (точкові) відображення*.

Найбільш природне уявлення про дискретні відображення можна одержати при описі динаміки біологічних популяцій. Нехай ми вивчаємо чисельність деякої популяції рік у рік. Ясно, що в цьому випадку доцільно порівнювати між собою чисельність, розраховану один раз у рік у якийсь певний момент часу. Можна чекати, що чисельність популяції в *(n+1)*-й рік є функцією її чисельності в попередній, *n-й* рік, і можна записати співвідношення

$$x_{n+1} = f(x_n).$$

Це і є найпростіше дискретне відображення. Ми можемо конкретизувати його вид, якщо зробимо кілька простих припущень. Нехай при малій чисельності популяції її величина рік у рік міняється в геометричній прогресії. Тоді

$$x_{n+1} = \lambda x_n,$$

де λ – параметр, що визначає швидкість розмноження. Зрозуміло, однак, що при більшій чисельності популяції, вона не зростатиме

надто швидко. Ефекти конкуренції за джерела харчування призведуть до обмеження чисельності.

Припустимо, що закон падіння чисельності пов'язаний із квадратичною нелінійністю:

$$x_{n+1} = \lambda x_n - x_n^2$$

Отримане нами співвідношення називається логістичним відображенням. Незважаючи на свій простий вид, воно служить одним з основних, фундаментальних моделей нелінійної динаміки й теорії динамічного хаосу. Еволюцію, що описується дискретними відображеннями, зручно представляти на *ітераційних діаграмах* (їх іноді ще називають діаграмами Ламерея). На діаграмі відкладають залежність $x_{n+1}(x_n)$, тобто функцію $f(x_n)$.

Еволюційний процес, що описується дискретними відображеннями, звичайно має дві стадії: перехідний процес і його результат – деякий усталений рух (звичайно, чіткої границі між цими стадіями немає – вона визначається точністю обчислень).

Найпростішим варіантом розвитку подій служить такий, коли чисельність популяції дорівнює константі. Очевидно, що для наведеного відображення згаданий випадок відповідатиме співвідношенню $f(y) = y$. У такому випадку говорять, що відображення має *нерухому точку*. Нерухома точка може бути стійкою або нестійкою. Критерієм цього є величина похідної, яка обчислена у нерухомій точці. Якщо ця похідна за модулем менше одиниці, збурювання загасає при ітераціях, і точка є стійкою, у противному випадку збурювання буде наростати, і нерухома точка буде нестійкою.

Стаціонарні режими можуть носити й більш складний коливальний характер. Наприклад, чисельність популяції може змінюватися з періодом в 2 роки, тобто задаватися послідовністю $x_1, x_2, x_1, x_2, \ldots$ У цьому випадку говорять, що відображення має цикл періоду 2. Також можливий нерегулярний у часі процес – *динамічний хаос*.

Завдання 1. Знайдіть нерухому крапку логістичного відображення й досліджуйте її стійкість залежно від параметра λ.

Завдання 2. Знайдіть у явному виді елементи 2-циклу логістичного відображення. У якому інтервалі параметра існує 2-цикл?

Завдання 3. Побудуйте за допомогою комп'ютера ітераційні діаграми для логістичного відображення для значень параметра , 1.2, 1.4, 1.5 і 1.9. Який процес буде встановлюватися в кожному випадку по закінченні досить великого часу?

Наведемо кілька прикладів, коли дискретні відображення виникають природно.

Генератор пілоподібних коливань. Нехай вихідний сигнал генератора коливань, яким є напруга, змвнюється за лінійним законом:

$$V(t)=U_0 - \alpha(t - t_n),$$

де U_0 і α – постійні, доти, поки не досягне нуля в деякий момент часу t_n. Після цього напруга стрибком збільшується до величини U_0. Коливання будуть періодичними в часі з періодом $T = U_0 / \alpha$. Далі розглянемо ситуацію за наявністю зовнішнього впливу, який змінює верхній поріг за гармонійним законом:

$$U(t)=U_0 + U_m \cos \omega t$$

Відтепер «схил» імпульсу описується рівнянням

$$V(t)=U(t_n) - U_0 (t - t_n)/T.$$

Неважко одержати співвідношення, щоб зв'язати моменти t_{n+1} і t_n. Оскільки в момент часу t_{n+1} напруга звертається в нуль, маємо:

$$U_0 + U_m \cos \omega t_n - U_0 (t_{n+1} - t_n) / T = 0.$$

Дозволяючи це співвідношення відносно t_{n+1} і переходячи до безрозмірної змінного $\Theta_n = \omega t_n$, одержуємо:

$$\Theta_{n+1} = \Theta_n + \Omega + f \cos \Theta_n,$$

де $\Omega=\omega T$ – растройка, $f = \Omega\ U_m\ /U_0$ – безрозмірна амплітуда зовнішнього впливу. Це – одна з еталонних моделей нелінійної динаміки, що називають відображенням окружності.

Системи під імпульсним періодичним впливом.
Важливий клас проблем, у яких виникають дискретні відображення, пов'язаний з вивченням систем, що перебувають під імпульсним періодичним впливом. Як приклад, розглянемо дискретне відображення для осцилятора з кубічною нелінійністю та загасанням, яке спостерігається для коливання під впливом періодичної послідовності δ-імпульсів:

$$x'' + 2\,\gamma x + \omega_0^2\, x + \beta\, x^3 = \Sigma\, C\, \delta(t - n).$$

Будемо вважати, амплітуди коливання між імпульсами змінюються повільно. У цьому випадку розв'язок рівняння надається у вигляді квазігармонійного коливання з повільно мінливою амплітудою:

$$x = A\, e^{\,i\omega ot} + A^*\, e^{\,-i\omega ot}$$

де A і A* – повільно мінливі функції часу (амплітуди).

У проміжку між імпульсами для комплексної амплітуди справедливе рівняння

$$A' = -\,\gamma A + 3i\beta\, |\, A\, |^2 A\ \ / 2\omega_0,$$

у чому можна переконатися безпосередньою підстановкою.

Розв'язок останнього рівняння можна шукати у вигляді $A = a\, e^{\,i\varphi}\ /2.$ Тоді для дійсні амплітуди a і фази φ отримаємо наступні рівняння:

$$a' = -\gamma a\ ,\ \ \varphi' = 3\beta a^{2/8}\omega_{0.}$$

Вирішуючи систему рівнянь щодо амплітуди a і фази φ, отримаємо наступні співвідношення в проміжку між *n-м* і *(n+1)*-м імпульсами:

$$a(t) = a_n e^{-\gamma t},$$
$$\varphi(t) = 3\beta a_n^2 (1 - e^{-2\gamma t}) / 16\gamma\omega_0 + \varphi_n,$$

де a_n і φ_n – початкові амплітуда та фаза відразу після *n-го* імпульсу.

Підставляючи всі отримані співвідношення у формулу для *x*, на розглянутому інтервалі часу між імпульсами отримаємо:

$$x_{n+1} = a_n e^{-\gamma T} \cos\{\omega_0 T + 3\beta a_n^2 (1 - e^{-2\gamma T}) / 16\gamma\omega_0 + \varphi_n\},$$

$$v_{n+1} = -\omega_0 a_n e^{-\gamma T} \sin\{\omega_0 T + 3\beta a_n^2 (1 - e^{-2\gamma T} / 16\gamma\omega_0 + \varphi_n\} + C.$$

При цьому внаслідок властивостей δ-потенціалу враховані безперервність координати й кінцевий розрив швидкості у точках $t = n$.

Для комплексної змінної z, яка складається з x и v за наступним правилом

$$z = z_a (ix + v / \omega_0),$$

де $z_a = \sqrt{3\beta a_n^2 (1 - e^{-2\gamma T}) / 16\gamma\omega_0}$, існує наступне відображення:

$$z_{n+1} = C z_a / \omega_0 + z_n e^{-\gamma T} \exp\{i(|z_n|^2 + \omega_0 T)\}$$

Аналогічне відображення було отримано K.Ikeda (1980) для кільцевого оптичного резонатора, який заповнено середовищем, показник переломлення якого залежить від амплітуди поля.

Дане відображення є *двовимірним*. Відповідна *фазова діаграма* (залежність швидкості від координати) може бути побудована чисельно.

3. Класифікація моделей екологічних процесів

Основні типи моделей: дескриптивні моделі й моделі поводження.

Дескриптивна модель дозволяє одержати інформацію про взаємозв'язки між найбільш важливими змінними системи. Реалізується такий тип моделі методами стохастичного моделювання, заснованого на інструментах теорії ймовірностей і математичної статистики. Розрізняють статичні методи, що не враховують час у якості змінної (наприклад, проста і множинна лінійна й нелінійна кореляція й регресія, дисперсійний аналіз), і динамічні методи, які враховують залежність від часу (наприклад, аналіз Фур'є, кореляційний і спектральний аналіз). Моделі, які відтворюють собою регресійні й інші емпірично встановлені кількісні залежності та не претендують на розкриття механізму описуваного процесу, є описовими.

Моделі поведінки описують системи під час перехідного періоду від одного стану до іншого. Вони, як правило, ураховують:

1) структуру сигналів на вході й виході системи;
2) реакцію системи на особливі перевірочні сигнали;
3) внутрішню структуру системи.

Етапи моделювання:

1) формування концепції;
2) визначення системи керуючих параметрів;
3) імітація (одержання чисельні розв’язків модельних рівнянь при фіксованих значеннях параметрів і початкових умов).

Критерії підбору системи керуючих параметрів:

1) оцінка на основі спостережень;
2) оптимальність параметрів (з використанням методів оптимізації);
3) чутливість (як модель реагує на зміну значень параметрів).

Способи імітації:

1) перевірка (якісне або кількісне порівняння даних, які отримано у результаті моделювання, з дійсними значеннями);
2) перевірка значимості моделі (проведення експериментів для вивчення поведінки моделі та системи з метою виявлення їхньої подібності, а також для порівняння тенденцій поведінки моделі й системи).

Типи фізико-математичних моделей екологічних процесів:

1) описові моделі;
2) якісні моделі (які з'ясовують динамічний механізм процесу, що досліджується, та здатні відтворити динамічні ефекти в поведінці системи, яка состерігається);
3) імітаційні моделі конкретних складних систем, які б враховували наявну інформацію про об'єкт (і що дозволяють прогнозувати поведінку систем або розв'язувати проблему оптимізації їхньої експлуатації). Особливе значення надається саме останньому класу моделей, оскільки він виявляється корисним для практичних цілей.

Зразковий алгоритм побудови імітаційної моделі:

1) формулювання предмета й проблеми дослідження, визначення початкових і граничних умов;
2) декомпозиція системи на окремі блоки, зв'язані, але відносно незалежні;
3) визначення параметрів стану кожного блоку;
4) формулювання гіпотез, що визначають поводження окремих блоків;
5) розробка алгоритмів поводження, що відповідають окремим блокам;
6) верифікація кожного блоку при незмінних граничних і початкових умовах;
7) об'єднання розроблених блоків, при цьому досліджуються різні схеми їхньої взаємодії;
8) верифікація імітаційної моделі й перевірка її адекватності;
9) проведення чисельних експериментів з моделлю з метою виявлення її адекватності системі й границь її застосовності.

4. Математичні методи, що застосовуються для побудови моделей

4.1. Диференціальні рівняння. Для опису екологічних систем використають диференціальні рівняння (системи диференціальних рівнянь), які описують динаміку чисельності (біомаси) кожної популяції, що входить у досліджувану систему. У загальному виді можна записати залежність

$$\frac{dx_i}{dt} = F_i\left(t, x_1, x_2, ..., x_w \right), \quad i = 1, ..., w,$$

де w – число видів у співтоваристві, x_i – чисельності *i-го* виду, t – час. Подальше теоретичне дослідження в більшості випадків проводиться для конкретного виду функцій F_i.

4.2. Варіаційне обчислення. Принципово іншим є метод моделювання, який засновано на застосуванні екстремальних принципів. Відповідно до них у реальності здійснюються лише деякі стани системи, а саме, стану з екстремальним значенням функціонала, що описує, наприклад, повну енергію системи. Широке застосування екстремальні принципи отримали у фізиці, механіці, термодинаміці, економіці, теорії керування.

Приклади функціоналів для опису екологічної системи:

1) потік енергії (Lotka, 1922);
2) дисипативна енергія (Ulanowicz, Hannon, 1987; Schnieder, Kay, 1994; Mauersberger, 1996);
3) «індекс зрілості» (ентропія) (Perez-Espana, Arreguin-Sanchez, 1999);
4) непрямі ефекти (потік енергії через границю системи) (Patten, 1986; 1995);
5) біомаса (Margalef, 1968);
6) ієрархічна організація (перенос ентропії) (O'Neill et al., 1986);
7) потік мас із неорганічної в органічну фазу (Whittaker, Woodwell, 1971);
8) узагальнена ентропія (з урахуванням екологічного компонента) (Левич, 1980).

4.3. Клітинні автомати. Клітинні автомати – динамічні моделі з дискретним часом, простором і станами. Простий клітинний автомат визначається решіткою *L*, простором станів *Q*, шаблоном сусідів δ і функцією місцевих переходів *f*. Кожна клітка решітки *L* може перебувати в стані із простору *Q*. Клітки можуть з'єднуватися різними способами, у найпростішому випадку вони утворюють квадратну або шестикутну решітку. Клітки можуть змінювати свій стан за кроками дискретного часу, звичайно вони це роблять синхронно. Доля клітки залежить від навколишніх її сусідів (сусіди першого роду - центральна клітка й чотири що примикають, сусіди другого роду - центральна клітка й 8 що примикають) і відповідної функції переходу. Правила переходу визначаються формулою

$$a_{t-1}^{s} = f\left(a_t^{s-\tau},...,a_t^{s},a_t^{s+\tau}\right),$$

де a_t^s – стан клітки *s* у момент часу *t*; τ – πанг сусідніх кліток; f – функція місцевого переходу.

Набір величин $\left\{ a_t^s \,\middle|\, \forall\, s \in I \right\}$ називається конфігурацією клітинного автомата в момент часу *t* (*I* – безліч індексів кліток).

Клітинний автомат із сусідами другого роду можна описувати за допомогою марковских ланцюгів.

Нехай $X_{t_1}\left(t_i \in T\right)$ – дискретні випадкові величини

$t \in T$, де $T = \left\{ t_0, t_1, ... \right\}$ – простір параметрів марковського ланцюга.

Для марковських ланцюгів визначальною є властивість

$$P\left(i_{n+1} \,\middle|\, i_n \right) = P\left(i_n \middle|\, i_{n-1}\right).$$

Поширюючи індекс *t* на тимчасовий і просторовий вимір, можна описати клітинний автомат із сусідами другого роду. У цьому випадку просторова залежність полягає у тому, що майбутнє клітки залежить від її оточення, що опусується простим марковським ланцюгом.

Залежно від часу майбутнє клітки залежить тільки від останнього стану.

Теорія клітинних автоматів використається, зокрема, для моделювання динаміки рослинного покриву, сукцесії лісів, міграції радіонуклідів.

4.4. Нейронні мережі.

Типи нейронних мереж:

1. Багатошарова нейронна мережа із вхідного, одного або декількох внутрішніх і вихідних шарів. Шари утворяться нелінійними елементами (нейронами), кожний нейрон одного шару зв'язаний з усіма нейронами наступній, кожній сполуці приписаний відповідна вага, зворотний зв'язок відсутній, а також неможливі ніякі сполуки між елементами одного шару. Кількість елементів вхідного й вихідного шарів визначається об'єктом дослідження. У роботі з нейронною мережею виділяють два етапи: навчання й тестування. Основний принцип навчальної процедури полягає в тому, що якщо мережа дає неправильну відповідь, то ваги коректують так, щоб зменшити помилку. Набори значень, що використаються для навчання мережі, повинні мати досить прикладів, щоб у цілому описувати проблему.
2. Мережа складається тільки із вхідного й вихідного шарів. Вихідний шар звичайно складається з елементів, об'єднаних у двовимірну квадратну (або іншої геометричної форми) решітку. Кожний нейрон пов'язаний з найближчими сусідами. Нейрони містять ваги (вектор ваг), кожний з яких відповідає вхідному значенню.

4.5. "Организменні" (individual-based) моделі.

Основним об'єктом моделі є індивід, що розглядається як унікальну, дискретна одиницю, у якої є, принаймні, одна характеристика, що залежить від віку (крім самого віку), наприклад, маса, ранг у соціальній ієрархії й т.п.

5. Моделі, які засновані на диференційних рівняннях

5.1. Приклади рівнянь.

У розділі про моделювання за допомогою диференціальних рівнянь, у першу чергу, розглядаються моделі фітопланктонних і мікробіологічних співтовариств.

5.1. 1. Моделювання співтовариств фітопланктону.

Традиційний шлях дослідження співтовариств мікроорганізмів полягає в моделюванні безперервних культур. Загальне рівняння, що описує кінетику концентрації кліток у такому процесі, має вигляд

$$\frac{dx}{dt} = x(\mu - d),$$

де x – концентрація кліток у культиваторі, μ – функція, що описує розмноження популяції, d – швидкість вимивання.

Швидкість розмноження може залежати від концентрації кліток, концентрації субстрату *s*, температури середовища та інших факторів.

У мікробіологічних системах, як правило, швидкість росту лімітується концентрацією субстрату, що відбивається залежністю, запропонованої Ж.Моно (Monod, 1942):

$$\mu = \frac{\mu_O \, s}{K_3 + s},$$

де μ_O – максимальна швидкість росту організмів за даних умов, K_3 – видова стала, яка дорівнює концентрації субстрату, для якої швидкість росту культури дорівнює половині максимальної (константа напівнасичення).

У моделюванні динаміки фітопланктону важливу роль відіграє облік впливу рівня освітленості на швидкість росту. У книзі С.Йоргенсена (1985) описані деякі види рівнянь, які застосовуваються у моделях. Залежність між швидкістю росту (швидкістю первинного

продукування) і освітленістю може бути описана рівнянням Михаеліса-Ментен

$$\mu = \frac{\mu_I \ I}{K_I + I},$$

тут I – освітленість, K_I – константа напівнасичення по освітленості.

При освітленості вище граничної, починаючи з якої відбувається гноблення фотосинтезу, можна записати наступну залежність:

$$\mu \frac{\mu_I \ K_I}{K_I + I - IH}.$$

Значення всіх констант (μ_I, K_I, IH) залежать від адаптації до освітленості й температури. Залежність K_I від температури, як правило, лінійна $K_I = K_{I0} + a_m T$, a_m, K_{I0} – константи, K_{I0} – константа напівнасичення по освітленості в нульовій крапці температурної шкали. Залежність IH від температури так само може бути виражена лінійною функцією: $IH = IH_0 + a_H T$; IH_0 – гранична освітленість у нульовій крапці температурної шкали.

Як вже було згадано, швидкість росту залежить від концентрації біогенних речовин. Для фітопланктону елементами, здатними лімітувати ріст, можуть бути, наприклад, азот, фосфор і вуглець. Можливі способи відбиття цього факту в роботі С.Йоргенсена описані за допомогою

$$\mu = \hat{\mu} \frac{P}{K_p + P} \cdot \frac{N}{K_N + N} \cdot \frac{C}{K_C + C},$$

де P, N, C – концентрація розчиненого фосфору, азоту й вуглецю; K_P, K_N, K_C – відповідні константи напівнасичення. Можливо також використання середньої величини факторів, що лімітують $\mu = \hat{\mu} \frac{\frac{P}{K_p + P} + \frac{N}{K_N + N} + \frac{C}{K_C + C}}{3}$.

Кромі того, у моделях досліджуються ефекти метаболічного впливу, як, наприклад, це було зроблено Ю. А. Домбровським зі співавторами (1990). Швидкість росту фітопланктону *i* -го виду було описано формулою

$$\mu_i = \frac{(1-\pi_i)M_i s}{H_i + \alpha_{11}x_1 + \alpha_{12}x_2 + s},$$

де x_1 , x_2 , s – концентрації двох видів фітопланктону й мінеральної речовини, виражені в одиницях біогенної речовини, що лімітує; π_i – коефіцієнти метаболізму; M_i - максимальна швидкість фотосинтезу; H_i – параметр насичення; α_{ij} – емпіричний коефіцієнт *інгібірования* і-го *виду* j-м.

5.1.2. Диференціальні рівняння в мікробіології.

Диференціальні рівняння, які описуються у термінах концентрації мікробної біомаси (x) і концентрації субстрату, що лімітує (s) в умовах хемостатичного культивування, які були виведені Ж.Моно виходячи з умов матеріального балансу, склали першу модель росту мікробних популяцій (Паніков, 1991)

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{dx_T}{dt} - \dfrac{dx_p}{dt} = \mu\, x - D\, x, \\ \dfrac{ds}{dt} = D\,(s_0 - s) - \dfrac{\mu\, x}{Y}, \end{cases} \qquad \mu = \hat{\mu}s / K_s + s\ ,$$

де x – концентрація мікроорганізмів у приймачі, куди зливається наростаюча бактеріальна суспензія; x_j – сумарна концентрація мікроорганізмів; μ – питома швидкість росту, $\hat{\mu}$ – максимальна швидкість росту; s – концентрація ресурсу, що лімітує, у середовищі; Ks – константа напівнасичення при лімітуванні даним субстратом; s_0 – величина концентрації субстрату, що лімітує, на вході в культиватор; D – швидкість розведення, рівна відношенню швидкості надходження живильного до середовища об'єму культури; Y – економічний коефіцієнт (вихід біомаси на одиницю спожитого субстрату).

Принциповою особливістю даної відкритої системи є можливість досягнення динамічної рівноваги: у стаціонарному стані $\mu = D$.

5.2. Концепція факторів, що лімітують.

Існування й успіх будь-якого організму або будь-якої групи організмів залежить від комплексу певних умов. Будь-яка умова, що наближається до межі толерантності або перевищує його, називається умовою, що лімітує, або фактором, що лімітує.

Використання в математичних моделях залежності швидкості росту популяції як функції одного елемента харчування $\mu(s)$ припустимо тільки для систем з постійним характером лімітування. Однак, зміна факторів лімітування зустрічається дуже часто і представляє як теоретичний, так і практичний інтерес. Існує кілька різних підходів до теоретичного опису механізму зміни факторів, що лімітують зростання чисельності популяції. Найпоширенішими є моделі, у яких передбачається, що в будь-який момент часу зростання контролюється тільки одним елементом харчування.

Таким, що лімітує, є елемент харчування, який призводить до мінімальної швидкості зростання чисельності популяції.

Приклад. Динаміка біомас видів x_1, x_2 і живильних речовин s^1, s^2, описується рівняннями

$$\begin{cases} \dot{x}_1 = (\mu_1 - D)x_1; \\ \dot{x}_2 = (\mu_2 - D)x_2; \\ \dot{s}^1 = D(s_0^1 - s^1) - \mu_1 x_1 / Y_{11} - \mu_2 x_2 / Y_{21}; \\ \dot{s}^2 = D(s_0^2 - s^2) - \mu_1 x_1 / Y_{12} - \mu_2 x_2 / Y_{22} \end{cases},$$

де x_i – концентрація біомаси *i-го* виду; D – швидкість протоки у хемостаті; $\mu_1 = \min\{\hat{\mu}_1, \beta_{11}s^1, \beta_{12}s^2\}$, $\mu_1 = \min\{\hat{\mu}_1, \beta_{11}s^1, \beta_{12}s^2\}$– питома швидкість росту організмів; s_0^j– концентрація *j-го* речовини; s^j – концентрація елементів харчуванняі; Y_{ij} – економічний коефіцієнт використання *j-го* речовини організмами *i-го* виду; β_{ij}– коефіцієнт пристосованості *i-го* виду до *j-му* ресурсу.

Дослідження стаціонарних станів системи дало наступні результати. Якщо обидва види споживають переважно той самий ресурс, то в системі стає домінуючий один вид, той, у якого максимальний коефіцієнт β по ресурсу, що лімітує. Якщо ж види споживають

переважно різні ресурси, то результат конкурентної боротьби залежить від значень s_0^j – концентрацій речовини j у середовищі.

Таким, що лімітує, є елемент харчування, для якого відношення „концентрація у середовищі”/”концентрація в організмі” є найменшим у порівнянні з іншимисуттєвими компонентами харчування.

5.3. Моделювання конкуренції за ресурси.

Одним з факторів, що впливають на формування структури екологічних співтовариств, є конкуренція. Конкуренція в самому широкому сенсі – це взаємодія організмів, які прагнуть отримати той самий ресурс.

Конкурентна взаємодія стосується простору, їжі або біогенних елементів, світла, залежності від хижаків і т.д. Міжвидова конкуренція може призвести до встановлення рівноваги між двома видами, або до заміни популяції одного виду на популяцію іншого, або до того, що один вид витисне іншої в інше місце або ж змусить його перейти на використання іншої їжі. При конкуренції близьких або подібних видів спостерігається тенденція до їхнього екологічного поділу. Ця тенденція утотожнюється принципу конкурентного виключення (принцип Гаузе). Наприклад, спільне споживання ресурсів харчування організмами одного трофічного рівня (модель динаміки концентрації кліток Абрикосіва).

Приклад. *“Графічна” теорія міжвидової боротьби.*

Ключовим поняттям теорії є засіб існування, яким вважається будь-яка речовина або фактор, що приводить до прискорення росту при збільшення його кількості в навколишнім середовищі, і який споживається організмом. У випадку, коли якийсь вид потенційно лімітується не одним, а декількома ресурсами, фактор буде вважатися засобом існування, коли є якийсь діапазон наявності або доступності іншого фактора, що лімітує, у якому перший задовольняє вимогам даного вище визначення. Як наслідок - можливість класифікації засобів існування на основі характеру взаємодії різних факторів, що визначають швидкість росту якого-небудь виду.

В основі класифікації лежать загальні рівняння, які показують, як засоби існування впливають на зростання чисельності популяції і як споживачі впливають на кількість засобів існування

$$\frac{dx_i}{dt} = f_i(s^1,...,s^m) - d_i, \qquad i = \overline{1,w},$$

$$\frac{ds^j}{dt} = g^j(s^j) - \sum_{i=1}^{w} x_i \, f_i(s^1,...,s^m) \, h_{ij}(s^1,...,s^m), \qquad j = \overline{1,m},$$

де x_i – щільність популяції виду i; s^j – запаси засобу існування j; d_i – показник смертності виду i; f_i – функція, що описує залежність швидкості розмноження (розраховуючи на одну особину) від кількості засобу існування; g^j – функція, що характеризує швидкість надходження засобу існування j; h_{ij} – функція, що описує кількість засобу j, що вимагається для створення кожної нової особини виду i; усього w видів конкурують за m засобів існування.

В цих рівняннях крім очевидних (безперервне розмноження, гомогенність популяцій і місць перебування й ін.) зроблений ряд важливих допущень: різні види взаємодіють між собою тільки через користування засобами існування; між засобами існування немає ніякої взаємодії.

5.4. Моделювання впливу міграції видів на стійкість системи

Інший аспект застосування систем диференціальних рівнянь для опису динаміки біологічного співтовариства - вивчення впливу міграції на стійкість системи.

Розглянемо n місцеперебувань (зон забруднення радіонуклідами), усередині яких динаміка описується системою рівнянь

$$\frac{dx_i}{dt} = f_i(x_1,...,x_w), \qquad i = 1,2,...,w,$$

де $x_i(t)$ – чисельність *i-го* виду (маса *i-го* радіонукліда), у співтоваристві w видів (радонуклидів) ($w \geq 2$). Далі збережена термінологія для співтовариства видів.

Схема міграційного потоку задається матрицею $M^{(n)} = \|m_{ks}\|$; елемент матриці m_{ks} указує потік мігрантів з місцеперебування s у місцеперебування k; елемент m_{kk} (зі знаком " – ") дає суму потоків емігрантів з *k-го* місцеперебування.

Для системи, замкнутої по міграції, виконується співвідношення

$$m_{kk} = -\sum_{s=1, s\neq k}^{n} m_k, \quad k = 1,2,...,n.$$

Середовище передбачається ізотропним по міграції, що для матриці $M^{(n)}$ означають рівність $m_{ks} = m_{sk}$ і рівність одиниці всіх ненульових недіагональних елементів. Число мігрантів будь-якого виду на будь-якому маршруті в одиницю часу вважається пропорційним чисельності даного виду в тім місцеперебуванні, звідки відбувається еміграція. Біологічні особливості видів описуються параметрами $m_i \geq 0, \ i = 1,...,w$. Таким чином, динаміка композиції співтовариств, що займають n місцеперебувань, буде описуватися рівняннями

$$\frac{dx_i^k}{dt} = f_i(x_1^k,...,x_w^k) + m_i \sum_{s=1}^{n} m_{ks} x_i^s, \quad k = \overline{1,n}, \quad i = \overline{1,w}.$$

Дослідження стану рівноваги цієї системи дозволяє зробити висновок про те, що при об'єднанні підсистем за допомогою міграції в ізотропному середовищі властивості стійкості не поліпшуються в порівнянні з ізольованим випадком.

5.5. Управління ростом і врожаєм мікроводоростей

Ріст водоростей у періодичній культурі з елементами мінерального харчування як лімітуючих факторів та описується рівняннями

$$\begin{cases} \dot{x} = \hat{\mu}\left(1 - \dfrac{q_{\min}}{q}\right) x \\ \dot{q} = v - \mu\, q \\ \dot{s} = -v\, x \end{cases},$$

де x – концентрація біомаси кліток; q – внутрішньоклітинний зміст елемента харчування; $q_{\min}$ – мінімальний зміст біогену в клітці, при якому питома швидкість росту μ дорівнює нулю.

6. Функціональні моделі

6.1. Функціонал дії.

Один із способів застосування цільової функції полягає у формулюванні загального твердження щодо поведінки системи. Добре відомі екстремальні принципи відносяться до цього випадку. Найвідоміший з них - принцип Гамильтона, відповідно до якого, кожна механічна система поводиться так, щоб дія (інтеграл за часом від функції Лагранжа) було мінімальним. В екологічних моделях існували спроби використання цього підходу (Wilhelm, Bruggemann, 2000) для одержання рівняння росту популяції, точніше, розглядалося зворотне завдання: записати дію, що приведе до спеціального рівняння росту. Одна з найбільш удалих спроб дозволити це завдання, запропоноване М.Гатто із співавторами (Gatto et al., 1988a,b), наведена в роботі Дж.Вебба (Webb, 1995).

Як функціонал дії, що приведе до так званого логістичного рівняння росту популяції чисельності n, було розглянуто наступний вираз

$$S = \int dt \left[\frac{1}{2}\left(\frac{\dot{n}}{n}\right)^2 + \frac{1}{2} r^2 \left(1 - \frac{n}{k}\right)^2 \right].$$

Для спрощення обчислення була зроблена заміна змінних

$$\begin{cases} S = \int dt \left[\frac{1}{2} \dot{x}^2 - V(x) \right], \quad x = \ln\left(\frac{n}{k} \right); \\ V(x) = -\frac{1}{2} r^2 (1 - e^x)^2 \end{cases}$$

Відповідно до варіаційного принципу, рівняння еволюції $x(t)$ задається вимогою екстремальності дії, тобто $d = 0$. Після необхідних обчислень було отримано динамічне рівняння

$$\ddot{x} = -r^2 e^x (1 - e^x) .$$

Для порівняння результату з логістичним рівнянням

$$\frac{dn}{dt} = rn\left(1 - \frac{n}{k}\right),$$

його переписали у змінних $x = \ln\left(\frac{n}{k} \right)$; $\dot{x} = r\,(1 - e^x)$, й продиференцювали: $\ddot{x} = -r^2 e^x (1 - e^x)$.

Отриманий збіг показує, що будь-які розв’язки логістичного рівняння є розв’язк динамічного рівняння, отриманого з функціонала дії. Однак, не будь-який розв’язок рівняння є розв’язком логістичного рівняння. Для виявлення взаємозв'язку між даними рівняннями було проведене дослідження отриманого рівняння еволюції. Після деяких перетворень і інтегрування було отримано

$$\frac{1}{2}\left(\frac{\dot{n}}{n}\right)^2 - \frac{r^2}{2}\left(1 - \frac{n}{k}\right)^2 = R .$$

Рівняння еволюції характеризується константою R: при $R > 0$ популяція необмежено зростає, при $R < 0$ популяція досягає максимального значення, а потім зменшується до 0. Значення $R = 0$ приводить до логістичного рівняння, показуючи, що логістичне зростання – це особливий випадок рівноваги між необмеженим зростанням і згасанням.

6.2. Мальтузіанський параметр.

Нехай співтовариство складається з *w* популяцій. У кожний момент часу воно може бути описане чисельностями (або біомасами) складових його популяцій x_i. Нехай $x = \sum_{i=1}^{w} x_i$ – сумарна біомаса співтовариства. Нехай кожна популяція характеризується мальтузіанським параметром $\mu_i(t)$ з рівняння $\frac{dx_i}{dt} = \mu_i x_i$. Нехай $p_i = \frac{x_i}{x}$ – відносний достаток *i-го* виду в співтоваристві. Тоді набір $p = \{p_1, \ldots, p_w\}$ називається структурою співтовариства; величина $\hat{\mu} = (\mu, p)$ – середнім мальтузіанським параметром співтовариства ($(\mu, p) = \mu_1 p_1 + \mu_2 p_2 + \ldots + \mu_w p_w$), а динаміка загальної біомаси описується рівнянням $\frac{dx}{dt} = \hat{\mu} x$.

Постулюється принцип максимуму середнього мальтузіанського параметра: співтовариство взаємодіючих популяцій еволюціонує таким чином, що його середній мальтузіанський параметр завжди зростає, досягаючи в стійкій рівновазі максимуму.

6.3. Принцип виживання.

Нехай динаміку екосистеми, до якої входить розглянутий вид, адекватно описує система рівнянь із невідомими чисельностями особин всіх елементів екосистеми. Як параметри рівнянь виступають екологічні умови, а також структурно-функціональні параметри особин всіх елементів екосистеми. Виділяють *s-я* популяція й деякий структурний або функціональний параметр α_{s_k} цієї популяції. Роблять припущення про те, що популяція складається із двох підпопуляцій, які розрізняються величиною фенотипічного параметру. Нехай $x_s^{(1)}$, $x_s^{(2)}$, $\alpha_{s_k}^{(1)}, \alpha_{s_k}^{(2)}$ – чисельності та величини фенотипічного параметру двох підпопуляцій.

Дослідження динамічної системи, у яку внесені відповідні зміни, які б враховували розходження фенотипічного параметру в особин *s-oї* популяції, дозволяє аналізувати асимптотичні властивості чисельності підпопуляцій. Один з можливих варіантів поведінки – витиснення однієї підпопуляції іншою (фенотипічний параметр $\alpha_{s_k}^{(1)}$ має селективну перевагу в порівнянні з параметром $\alpha_{s_k}^{(2)}$ у заданих екологічних умовах).

У ряді досліджень як критерій оптимальності виступала вимога максимуму відносної швидкості росту чисельності популяції:

$$k = \frac{d \ln x}{dt} = \max$$

Цей критерій може бути застосований для визначення оптимальних величин структурно-функціональних параметрів, якщо відносна швидкість росту чисельності представлена у вигляді функції цих параметрів.

6.4. Моделі динамічної структури.

Досліджується узагальнений потік енергії через систему

$$TST = \sum_{j=1}^{n} T_j$$

$$T_j = \sum_{\substack{i=0 \\ i \neq j}}^{n} f_{ij} \quad j=1,\ldots,n.$$

де T_j являє собою загальний потік енергії (речовини) з *j-oї* частини системи, f_{ij} – потік від j до i, значення індексу $i = 0$ позначає зовнішнє середовище.

Останнім часом найбільше поширення отримала функція эксергії або енергії переходу, під якою розуміють максимальну вільну енергію, яку система здатна виділити в простір при переході в стан термодинамічної рівноваги. Эксергія визначається середовищем і

може бути розглянута як міра термодинамічного порядку. Крім того, енергія переходу безпосередньо пов'язана з термодинамічною інформацією *I*: $E = IT$ (T – температура).

Для систем з неорганічними потоками й пасивним продукуванням органічних речовин енергія переходу задається формулою

$$E_x = RT \sum_{i=0}^{n} \left[c_i \ln\left(\frac{c_i}{c_i^{eq}}\right) - (c_i - c_i^{eq}) \right],$$

де R – газова стала; T – абсолютна температура; c_i - концентрація в екосистемі *i-го* компоненту; c_i^{eq} – відповідна концентрація *i-го* компоненту в термодинамічній рівновазі; індекс $i = 0$ відповідає неорганічним компонентам розглянутої хімічної речовини.

6.5. Принцип максимуму Понтрягіна

Принцип максимуму Понтрягіна ілюструється на прикладі моделі вилову двох конкуруючих видів риб, що підкоряється закону логістичного росту.

Система рівнянь, що описує дану смистему, має вигляд:

$$\begin{cases} \dfrac{dx}{dt} = rx\left(1 - \dfrac{x}{K}\right) - \alpha xy - q_1 E_x, \\ \dfrac{dy}{dt} = sy\left(1 - \dfrac{y}{L}\right) - \beta xy - q_2 Ey \end{cases}$$

Тут *r, s* являють собою екологічний потенціал, *K, L* – пропускні здатності двох видів, члени $-\alpha xy$ й $-\beta xy$ відбивають боротьбу за загальний ресурс, q_1 і q_2 – коефіцієнти вилову видів і E – зусилля, витрачені на їхній спільний вилов. Передбачається, що існує зовнішній ресурс, що підтримує кожний вид.

Чистий дохід у будь-який момент часу визначається формулою: $П(x, y, E) = p_1 q_1 xE + p_2 q_2 yE - CE$, де *З* – ціна рибного лову, пропорційна зусиллям.

6.6. Ентропійні моделі

Як цільова функція використаовується ентропія

$$S = -\sum_i p_i \ln p_i .$$

Ентропія популяції записується у вигляді

$$S = -\sum_{i=1}^{n} \frac{N_i}{N} \ln \frac{N}{N} = const ;$$

загальна фіксована чисельність задається виразом

$$N = \sum_{i=1}^{n} N_i = const ,$$

а фіксована маса популяції – виразом $M = \sum_{i=1}^{n} m_i N_i = const$

(m_i – маса окремої особини *i-го* віку).

Проблема знаходження умов досягнення стаціонарного стану, який би задовольняв перерахованим вище умовам, вирішується застосуванням методу невизначених множників Лагранжа до системи

$$\begin{cases} \delta\, N = 0; \\ \delta\, M = 0; \\ \delta\, S = -\sum\limits_{i=1}^{n} \dfrac{\delta\, N_i}{N} \ln \dfrac{N_i}{N} = 0. \end{cases}$$

У результаті була отримана формула, що визначає зв'язок між віковою чисельністю *i-й* групи N_i, масою m_i окремої особини *i-го* віку й загальною чисельністю популяції N

$$N_i = \frac{Ne^{-\frac{N m_i}{\theta}}}{\sum_{i=1}^{n} e^{\frac{N m_i}{\theta}}},$$

де θ - модуль статистичного розподілу різних особин по віках, що визначається з експерименту на основі фізичних міркувань.

Остання формула може бути наведена в іншому вигляді, якщо як фіксований параметр взяти середній час життя популяції $T = \frac{1}{N}\sum_i t_i N$ (t_i – вік *i-й* групи популяції)

$$N_i = \frac{Ne^{-\frac{N t_i}{\theta}}}{\sum_{i} e^{\frac{N t_i}{\theta}}}.$$

Може також бути використаний універсальний критерій еволюції Гленсдорфа-Пригожіна у вигляді принципу мінімуму виробництва ентропії: тимчасова похідна енропії прагне в стан мінімуму. Розманітість моделей визначається способом моделювання самої ентропії *S*. Спосіб моделювання матиме у своїй основі термодинамічну або статистичну форму. Можуть також моделюватися інформаційні (екологічні) внески до ентропії.

7. Контрольні завдання

7.1. Здійснити моделювання росту кластера частинок в квадратній (прямокутній) області за допомогою моделі клітинних автоматів (Монте-Карло), якщо відома кількість зародкових центрів.

7.2. Отримати розподіл невзаємодіючих частинок, які напорошені на безструктурну (структурну) підкладку методом Монте-Карло.

7.3. Змоделювати сезонні коливання ентропії біосистеми. Чисельно перевірити універсальний критерій еволюції Гленсдорфа-Пригожіна.

7.4. Описати протікання струму (перколяцію) у кластері сферичних не взаємодіючих (взаємодіючих) частинок, які розподілені на підкладці з металевими контактами. Побудувати анімаційну картину відповідного перколяційного кластера. Чисельно визначити поверхневу концентрацію частинок на початку перколяції.

7.5. Дослідити залежність середнього квадрата зсуву броунівської частинки від часу методом Монте-Карло.

7.6. Змоделювати процес дифузії радіонуклідної домішки у рідкому (газовому) середовищі.

7.7. Побудувати фазову діаграму системи масивних матеріальних точок методом молекулярної динаміки. Ускладнити завдання, вважаючи частинки іонами.

Література

1. Потемкин В.Г. Система инженерных и научных расчетов Matlab 5.x. В 2-х т.т. М.: Диалог-МИФИ, 1999.
2. Самарский А.А. Введение в численные методы: Учеб. Пособие. М.:Наука, 1982.
3. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. Часть 1 и 2. М.:Мир, 1990.
4. Методы Монте-Карло в статистической физике/Под ред. К. Биндера. М.: Мир, 1982.
5. Румер Ю.Б., Рывкин М.Ш. Термодинамика, статистическая физика и кинетика М.: Наука, 1972.
6. Базаров И.П., Геворкян Э.В., Николаев П.Н. Неравновесная термодинамика и физическая кинетика. М.: Изд-во МГУ, 1989.
7. Ландау Л.Д., Лифшиц. Физическая кинетика. М.: Наука, 1986.
8. Крышев И.И., Сазыкина Т.Г. Математическое моделирование миграции радионуклидов в водных экосистемах. М.: Энергоатомиздат, 1982.
9. Кивва С.Л., Колябина И.Л. и др. Моделирование процесса миграции радионуклидов в почвогрунтах. Чернобыль: НПО «Припять», 1992.
10. Климонтович Ю.Л. Статистическая теория открытых систем, т.1. М.: ТОО “Янус”, 1995.
11. Хакен Г. Информация и самоорганизация. Макроскопический подход к сложным системам. М.: Мир. 1991. – 240 с.
12. Абросов Н.С. Экологические факторы и механизмы формирования видового разнообразия экосистем и проблема совместимости видов // Экология в России на рубеже XXI века. М.: Научный мир. 1999. – С.54 – 69.
13. Алексеев В.В., Крышев И.И., Сазыкина Т.Г. Физическое и математическое моделирование экосистем. С.-Пб.: Гидрометеоиздат. 1992. – 367 с.
14. Барабашева Ю.М., Девяткова Г.Н., Тутубалин В.Н., Угер Е.Г. Некоторые модели динамики численностей взаимодействующих видов с точки зрения математической статистики // Журнал общей биологии. – 1996. 57, N.2. – С.123 – 139.
15. Левич А.П., Алексеев В.Л. Энтропийный экстремальный принцип в экологии сообществ: результаты и обсуждение // Биофизика. – 1997. 42, вып.2. – С.534 – 541.

Навчальне видання

Худинцев Микола Миколайович

Моделювання фізико-хімічних процесів у радіоекології

Конспект лекцій

Підписано до друку 30.10.2008 р. Формат 60×84/16
Папір офсетний. Ум. друк. арк. 5,35
Наклад 50 прим. Замовлення 394
Видавництво та друкарня ”ТЕС”
(Свідоцтво ДК № 771) Одеса, Канатна 81/2.

Надруковано з готового оригінал-макета

Одеський державний екологічний університет

65016, м.Одеса, вул..Львівська, 15
