І. О. Князь, А. М. Вітренко

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ДИНАМІЧНИХ СИСТЕМ

Розділ
"Моделювання фізичних систем"

Навчальний посібник

Рекомендовано вченою радою
Сумського державного університету

Суми
Сумський державний університет
2011

УДК 004.94 (075.8)
ББК 32.973.2–018я73
К 54

Рецензенти:

О. С. Мазманішвілі – доктор фізико-математичних наук, професор Сумського державного університету;
О. В. Хоменко – доктор фізико-математичних наук, професор Сумського державного університету;

Рекомендовано вченою радою Сумського державного університету як навчальний посібник
(протокол № 12 від 12.05.2011р.)

Князь І. О.

К 54 Комп'ютерне моделювання динамічних систем. Розділ "Моделювання фізичних систем" : навч. посіб. / І. О. Князь, А. М. Вітренко. — Суми : Сумський державний університет, 2011.— 140 с.

У посібнику на наочних прикладах розглянуто методику та основні підходи до моделювання фізичних процесів та систем. Подано широкий спектр задач: від руху матеріальної точки у силових полях до статистичного моделювання систем багатьох частинок.

Посібник розрахований насамперед на студентів фізичних та математичних спеціальностей, а також аспірантів та науковців, які займаються комп'ютерним моделюванням.

УДК 004.94 (075.8)
ББК 32.973.2–018я73

Зміст

Розділ 1

Рух частинки у силовому просторі

У цьому розділі на наочних прикладах розглядаються основні підходи до моделювання руху частинок у різних силових полях: однорідному, гравітаційному, електричному, електромагнітному. У розділі подано основні критерії, якими повинен користуватися дослідник при підборі різницевої схеми для інтегрування відповідних рівнянь руху.

1.1 Постановка задачі

Побудуйте траєкторії руху матеріальної частинки масою m у силовому полі $\overrightarrow{F} = \overrightarrow{F}(x, y)$ (або $\overrightarrow{F} = \overrightarrow{F}(x, y, z, t)$ у загальному випадку) при заданих початкових умовах: $x_0 = x(0)$, $y_0 = y(0)$, $v_{0x} = v_x(0)$, $v_{0y} = v_y(0)$.

1.2 Теоретичний матеріал

Розглянемо два види сил першочергової важливості — це електричні та гравітаційні сили, причому і ті, й інші є далекодіючими і потенціальними. Рух окремої частинки визначається дією прикладеної до неї заданої зовнішньої сили $\overrightarrow{F}$ [1]. Якщо це поле потенціальне, то його можна подати за допомогою скалярного потенціалу U:

$$\overrightarrow{F} = -\overrightarrow{\nabla} U, \qquad \overrightarrow{\nabla} U \equiv \frac{\partial}{\partial \overrightarrow{x}}. \tag{1.1}$$

Стан точкової частинки маси m визначається чотирма координатами (у двовимірному випадку): вектором просторового положення $\overrightarrow{x} = \{x, y\}$ і вектором швидкості $\overrightarrow{v} = \{v_x, v_y\}$. Координати частинки задовольняють рівняння руху

$$\begin{aligned} \frac{d\overrightarrow{x}}{dt} = \overrightarrow{v} \quad &\rightarrow \quad \left\{ \frac{dx}{dt} = v_x, \quad \frac{dy}{dt} = v_y \right\}, \\ \frac{d\overrightarrow{v}}{dt} = \frac{1}{m}\overrightarrow{F} \quad &\rightarrow \quad \left\{ \frac{dv_x}{dt} = \frac{1}{m}F_x, \quad \frac{dv_y}{dt} = \frac{1}{m}F_y \right\}. \end{aligned} \tag{1.2}$$

Основними прикладами такого руху є рух частинки в однорідному силовому полі, у центрально-симетричному силовому полі, у центральному полі сил пружності і т.п. В узагальнених випадках можуть бути враховані сили в'язкого тертя.

У зв'язку з тим, що сила F залежить лише від просторових координат x (і від часу t у загальному випадку), права частина кожного з рівнянь (1.2) не залежить від тієї величини, для якої записане рівняння. Отже, для чисельного моделювання цієї системи рівнянь особливо підходить різницева схема з переступом, оскільки координату та швидкість частинки потрібно визначати тільки у моменти часу, що чергуються.

Важливо відмітити, що під час руху частинки у потенціальному силовому полі виявляються дві надзвичайно важливі властивості: оборотність часу і збереження енергії. Отже, будь-яка схема інтегрування траєкторії частинки повинна забезпечувати оборотність часу і збереження енергії (якщо не строго, то хоча б із високим порядком точності) [2]. Покажемо, що метод з переступом враховує дані властивості. Розглядаючи моменти часу t, $t+\triangle t$, $t+2\triangle t$, $t+3\triangle t$ і т.д., маємо

$$\begin{aligned} \overrightarrow{x}^{n} &= \overrightarrow{x}^{n-2} + 2\Delta t\,\overrightarrow{v}^{n-1}, \\ \overrightarrow{v}^{n+1} &= \overrightarrow{v}^{n-1} + 2\Delta t \frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^{n}, t^{n}). \end{aligned} \tag{1.3}$$

Зрозуміло, що розглянута різницева схема є оберненою у часі, тому що якщо $\Delta t' = -\Delta t$, то

$$\begin{aligned} \overrightarrow{x}^{n-2} &= \overrightarrow{x}^{n} + 2\Delta t'\,\overrightarrow{v}^{n-1}, \\ \overrightarrow{v}^{n-1} &= \overrightarrow{v}^{n+1} + 2\Delta t' \frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^{n}, t^{n}), \end{aligned} \tag{1.4}$$

і отримані рівняння цілком аналогічні рівнянням із прямим ходом часу (1.3). Очевидно, що у випадку, коли схема інтегрування траєкторії частинки порушувала б оборотність часу, то в системі таких частинок спостерігалося б зростання ентропії, обумовлене чисельними ефектами.

Для перевірки умови збереження енергії перепишемо друге рівняння (1.3) для швидкостей у вигляді

$$\overrightarrow{v}^{n+1} - \Delta t \frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^{n}, t^{n}) = \overrightarrow{v}^{n-1} + \Delta t \frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^{n}, t^{n}). \tag{1.5}$$

Підносячи це рівняння до квадрата, маємо

$$(v^{n+1})^2 - 2\Delta t\,\overrightarrow{v}^{n+1}\frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^n, t^n) + (\Delta t)^2\left(\frac{1}{m}F(x^n, t^n)\right)^2 =$$
$$= (v^{n-1})^2 + 2\Delta t\,\overrightarrow{v}^{n-1}\frac{1}{m}\overrightarrow{F}(\overrightarrow{x}^n, t^n) + (\Delta t)^2\left(\frac{1}{m}F(x^n, t^n)\right)^2.$$

Таким чином, зміна кінетичної енергії на двох часових шарах становить

$$\frac{1}{2}m(v^{n+1})^2 - \frac{1}{2}m(v^{n-1})^2 = (\overrightarrow{v}^{n+1} + \overrightarrow{v}^{n-1})\overrightarrow{F}(\overrightarrow{x}^n, t^n)\Delta t. \qquad (1.6)$$

Використовуючи різницеві рівняння (1.3) для координат на двох часових шарах, виключимо швидкості з правої частини рівняння (1.6):

$$\frac{1}{2}m(v^{n+1})^2 - \frac{1}{2}m(v^{n-1})^2 = \frac{1}{2}(\overrightarrow{x}^{n+2} - \overrightarrow{x}^{n-2})\overrightarrow{F}(\overrightarrow{x}^n, t^n). \qquad (1.7)$$

Права частина цього рівняння являє собою зміну потенціальної енергії частинки на проміжку часу $[t^{n-1}, t^{n+1}]$, узяту з протилежним знаком. Але оскільки цю зміну потенціальної енергії не можна виразити у вигляді різниці значень потенціалу, енергія частинки цілком не зберігається. Проте права частина рівняння (1.7) становить різницеву апроксимацію інтеграла $\int_{n-1}^{n+1}\overrightarrow{F}(\overrightarrow{x}^n, t^n)d\overrightarrow{x}$ із другим порядком точності. Таким чином, хоча строге збереження енергії і не спостерігається, похибка є малою, і поки схема обернена, викривлення відсутні.

Проаналізуємо основні типи силових полів [3].

Рух частинки в однорідному полі. В усіх точках простору вектор сили має сталі проекції на осі координат. У такому випадку за відсутності сили тертя частинка рухається по параболі, а за її наявності — по більш складній траєкторії.

Рух у центрально-симетричному полі. Проекції на осі координат можна визначити, виходячи із рис. 1.1а. Якщо частинка притягується до центра, то залежно від початкових координат і швидкостей вона рухається по гіперболі, параболі або еліпсу; якщо відштовхується — по гіперболі.

Рух у магнітному полі. Рух зарядженої частинки у магнітному полі з індукцією $\overrightarrow{B}$ буде двовимірним, якщо початкова швидкість частинки перпендикулярна до силових ліній магнітного поля. При цьому з боку поля діє сила Лоренца, яка спрямована перпендикулярно до вектора швидкості (рис. 1.1б). Заряджена частинка буде рухатися по колу, радіус якого зменшується за наявності гальмівної сили.

Рух частинки в електричному та магнітному полях. Нехай силові лінії електричного поля $\overrightarrow{E}$ лежать у площині рисунка і спрямовані вгору, а силові лінії магнітного поля $\overrightarrow{B}$ спрямовані на нас і є перпендикулярні до вектора $\overrightarrow{E}$. Якщо заряд частинки додатний, то на нього з боку електричного поля діє стала сила qE, спрямована вгору, якщо від'ємний — униз (див.табл. 1.1). Якщо початкова швидкість частинки дорівнює нулю, то траєкторією її руху є циклоїда.

1.3 Алгоритми

Розглянемо алгоритм чисельного моделювання руху частинки у зазначених вище полях. У випадку центрально-симетричного або однорідного поля за відсутності сил тертя метод із переступом дає непогані результати. У такому разі алгоритм моделювання зводиться до таких кроків:

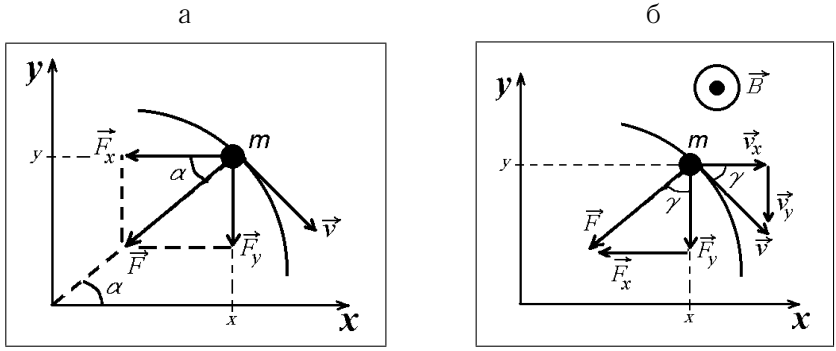

Рисунок 1.1 — Рух у центрально-симетричному (а) та магнітному (б) полях

Таблиця 1.1 — Сили, що діють на частинку, та їх проекції на осі координат

Тип поля	Сили	Проекції сил
Однорідне	$\overrightarrow{F} = \overrightarrow{k}\,\overrightarrow{x}$	$F_x = k_x x$ $F_x = k_y x$ k_x, k_y — константи
Центрально-симетричне	$F = G\frac{mM}{r^2}$ або $F = k\frac{qQ}{r^2}$, $r = \sqrt{x^2+y^2}$	$F_x = -F\cos\alpha = -F\frac{x}{r}$ $F_y = -F\sin\alpha = -F\frac{y}{r}$
Магнітне	$\overrightarrow{F} = k\overrightarrow{v}\,\overrightarrow{B}$	$F_x = -\lvert F\rvert\sin\gamma = \lvert F\rvert\frac{v_y}{\lvert v\rvert}$ $F_y = -\lvert F\rvert\cos\gamma = -\lvert F\rvert\frac{v_x}{\lvert v\rvert}$ $\lvert v\rvert = \sqrt{v_x^2+v_y^2}$
Електро-магнітне	$\overrightarrow{F}_1 = k\overrightarrow{v}\,\overrightarrow{B}$, $\overrightarrow{F}_2 = q\overrightarrow{E}$	$F_x = \lvert F_1\rvert\frac{v_y}{\lvert v\rvert}$ $F_y = \lvert F_2\rvert - \lvert F_1\rvert\frac{v_x}{\lvert v\rvert}$

[1] Задаємо параметри фізичної системи: масу частинки m, початкові координати $x(t=0)$, $y(t=0)$; компоненти вектора швидкості $v_x(t=0)$, $v_y(t=0)$, проекції сили F_x, F_y (відповідно до отриманих виразів (табл. 1.1), крок за часом $\triangle t$ та параметри поля.

[2] Беремо $t=0$. Просуваємося на півкроку за часом, перераховуючи швидкість: $v_x(t+1/2\triangle t) = v_x(t) + F_x(x(t),y(t))/\triangle t/(2m)$, $v_y(t+1/2\triangle t) = v_y(t) + F_y(x(t),y(t))/\triangle t/(2m)$.

[3] Цикл за часом.

[3.1] Розраховують координату та швидкість частинки у наступний момент часу:

$$\begin{aligned} x(t+\triangle t) &= x(t) + v_x(t+\triangle t/2)\triangle t,\\ y(t+\triangle t) &= y(t) + v_y(t+\triangle t/2)\triangle t,\\ v_x(t+\triangle t+1/2\triangle t) &= v_x(t+1/2\triangle t) + \frac{F_x(t+\triangle t)}{m}\triangle t,\\ v_y(t+\triangle t+1/2\triangle t) &= v_y(t+1/2\triangle t) + \frac{F_y(t+\triangle t)}{m}\triangle t. \end{aligned} \quad (1.8)$$

На цьому етапі результати обчислень, наприклад $(x(t+\triangle t), y(t+\triangle t))$, виводимо на екран або у файл.

(3.2) Збільшуємо час t на крок $\triangle t$. Якщо цикл по t закінчився — вихід із циклу.

Розглянемо більш загальний випадок — наявність сили тертя. У такому разі права частина ДУ, що описує рух тіла, залежить не тільки від координати, але й від швидкості. У такому випадку більш прийнятним є модифікований метод Ейлера[1].

1.4 Приклади

Подана у додатку програма дозволяє вивчити рух частинки у силових полях різних типів (які задаються відповідними проекціями сил (табл.1.1) за наявності сили тертя $\overrightarrow{F} = -r\overrightarrow{v}$.

Результати роботи програми подані на рис. 1.2.

Як бачимо з рис. 1.2, результати повністю відповідають зазначеним у розділі припущенням про можливу форму траєкторії частинки. У випадку однорідного поля (проекції сил збігаються з напрямком осей координат) частинка рухається по довільній траєкторії (рис. 1.2а). У випадку центрально-симетричного та гравітаційного поля радіус траєкторії поступово зменшується (рис. 1.2б-в). У випадку електромагнітного поля траєкторія становить циклоїду зі змінним радіусом витка; через певний час траєкторія частинки вироджується у пряму лінію.

[1] Сила, що діє на частку в магнітному полі, залежить від швидкості, і навіть схема Ейлера, яка використана у прикладі, виявляється тут недостатньо точною (хоча за умови малого кроку $\triangle t$ вона дає непогані результати). Можна рекомендувати схему розрахунку, що базується на рівності вигляду

$$x(t+\triangle t) = x(t) + \dot{x}(t)\triangle t + \ddot{x}(t)\frac{\triangle t^2}{2} + \dddot{x}(t)\frac{\triangle t^3}{6} + O(\triangle t^4), \qquad (1.9)$$

причому $\dot{x}(t) = v(t)$, $\ddot{x}(t) = F(t)/m$, а $\dddot{x}(t)$ з тією самою точністю виражається як $(f(t) - f(t-\triangle t))/\triangle t$.

Питання для самоконтролю

1 Якими критеріями необхідно користуватися при підборі чисельного методу для моделювання руху частинки у силовому просторі?

2 Напишіть рівняння руху частинки у різних силових полях за наявності сили тертя.

3 Побудуйте модель руху частинки у тривимірному центрально - симетричному силовому просторі.

4 Побудуйте модель руху частинки у тривимірному електричному полі.

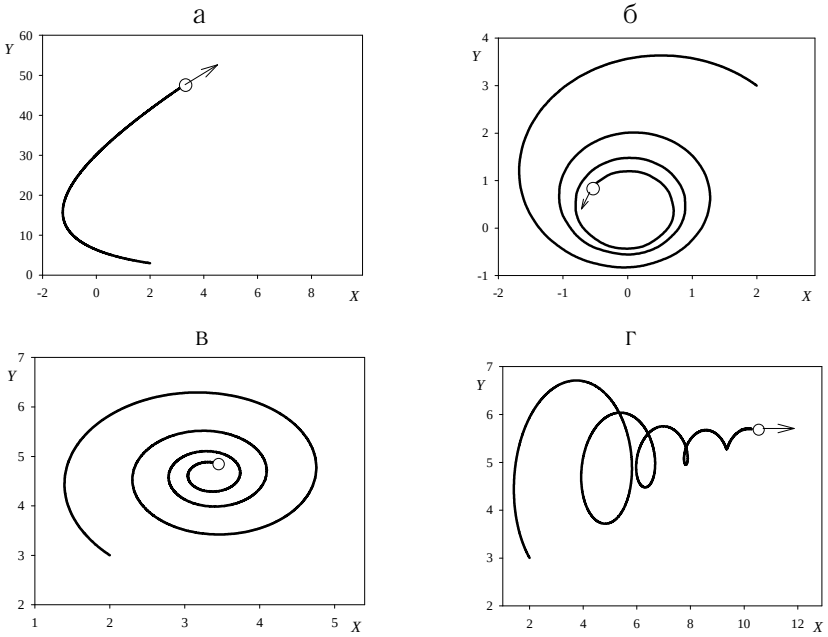

Рисунок 1.2 — Чисельне моделювання руху частинки у силових полях: а) однорідне поле; б) центрально-симетричне поле; в) магнітне поле; г) електромагнітне поле

Завдання для самостійної роботи

1 Проаналізуйте рух точкового об'єкта у полі гравітаційних сил, які діють за законом обернених квадратів $F = GmM/r^2$. Промоделюйте ситуації, при яких об'єкт рухається по гіперболі, параболі та еліпсу. Вивчіть характер руху штучного супутника Землі, який входить у верхні шари атмосфери (поява сили в'язкого тертя).

2 Вивчіть рух точкового об'єкта у полі сил відштовхування. Промоделюйте досліди Резерфорда щодо відхилення альфа-частинок ядрами атомів золота. Припускається, що альфа-частинка не проникає в ядро: взаємодія між частинкою та ядром описується законом Кулона. В експерименті припускають, що частинка спочатку розташована на великій відстані від ядра (де кулонівська сила є малою) і прямолінійно рухається уздовж осі x у бік ядра зі сталою швидкістю v_1. Після розсіювання частинка буде рухатися прямолінійно із сталою швидкістю v_2. Зрозуміло, що частинка та ядро повинні бути зсунутими уздовж осі y на певну відстань (прицільний параметр). Змінюючи прицільний параметр, проведіть серію комп'ютерних експериментів.

3 Промоделюйте рух зарядженої частинки у камері Вільсона, яка знаходиться в однорідному магнітному полі. Врахуйте, що в міру свого руху частинка втрачає кінетичну енергію. Магнітне поле спрямоване перпендикулярно до площини екрана.

4 Вивчіть рух зарядженої частинки у схрещених електричному і магнітному полях, спрямованих паралельно і перпендикулярно до площини екрана відповідно.

5 Вивчіть рух матеріальної точки в гравітаційному полі двох масивних тіл. Проведіть комп'ютерні експерименти при різних початкових умовах.

6 Задайте довільне силове поле, наприклад періодичне, і промоделюйте рух частинки у ньому.

Розділ 2

Моделювання руху систем взаємодіючих частинок

У розділі розглядається один із найпоширеніших комп'ютерних методів моделювання систем багатьох частинок (тверді тіла, класичні рідини та гази тощо) — метод молекулярної динаміки. Подано основні теоретичні відомості щодо чисельного визначення макроскопічних (термодинамічних) властивостей класичних середовищ за допомогою усереднення за часом. На наочному прикладі показано основні прийоми щодо реалізації зазначеного методу: усунення поверхневих ефектів, підвищення точності розрахунків і т.п.

2.1 Постановка задачі

Побудуйте траєкторії руху N матеріальних частинок з масами m_i ($i = 1..N$), що взаємодіють між собою із силами $F_{ij} = F_{ij}(m_i, m_j, r_{ij})$ (r_{ij} — відстань між частинками i та j) у силовому полі $\overrightarrow{F}_i = \overrightarrow{F}(m_i, \overrightarrow{x}_i)$ при заданих початкових координатах $\overrightarrow{x}(t = 0)$ та швидкостях $\overrightarrow{v}(t = 0)$. Визначте внутрішню енергію (або температуру) системи у рівноважному стані.

2.2 Теоретичний матеріал

2.2.1 "Точне"моделювання системи N тіл. Метод молекулярної динаміки

Побудуємо машинну модель системи N взаємодіючих тіл, яку будемо використовувати для одержання інформації, по-перше, про макроскопічні термодинамічні властивості системи і, по-друге, про мікроскопічні молекулярні властивості. Зазначимо, що для реалізації такої системи можна використати два підходи. У *методі молекулярної динаміки*, який пропонується у цьому розділі, будується система диференціальних рівнянь, кожне з яких описує стан окремої частинки. Точна еволюція у

часі системи з N частинок простежується шляхом поступового інтегрування рівнянь руху для кожної частинки. Після достатнього числа кроків за часом вважається, що термодинамічна рівновага настала, і термодинамічні властивості визначаються за допомогою усереднення за часом тих мікроскопічних властивостей, які є предметом вивчення. У *методі Монте-Карло* (див. наступний розділ) використовуються принципи статистичної механіки Гіббса, і термодинамічні властивості визначаються шляхом усереднення за ансамблем.

Розглянемо основні ідеї методу молекулярної динаміки. Припустимо, що між $N(N-1)/2$ парами частинок діє двочастинковий потенціал. Як придатний приклад такого потенціалу можна розглянути потенціал Леннарда-Джонса

$$U(r_{12}) = 4\varepsilon \left\{ \left(\frac{\sigma}{r_{12}} \right)^{12} - \left(\frac{\sigma}{r_{12}} \right)^{6} \right\}, \tag{2.1}$$

де r_{12} — відстань між частинками 1 і 2, а ε ("відстань" взаємодії частинок 1 та 2) і σ (глибина потенціальної ями або "енергія") — сталі. Такий потенціал добре описує притягання у випадку, коли частинки віддалені на значну відстань, і відштовхування, коли вони зближені. У такому разі еволюція системи у часі відбувається відповідно до детерміністичних законів руху кожної частинки, причому сила взаємодії F_{ij} спрощується і подається так:

$$\overrightarrow{F}(|\overrightarrow{x}_i - \overrightarrow{x}_j|) = -\frac{\partial}{\partial \overrightarrow{x}_i} U(|\overrightarrow{x}_i - \overrightarrow{x}_j|). \tag{2.2}$$

В останньому рівнянні просторова похідна визначає градієнт, розрахований у точці розміщення i-ї частинки[2].

Оскільки нашою метою є вивчення якісних властивостей систем багатьох частинок, то можна ввести деякі спрощення: динаміка системи — класична, частинки — хімічно-інертні тіла. Стан системи задається $6N$-вимірним вектором, що утворений просторовими координатами і компонентами швидкостей усіх частинок. Відповідно до законів руху Ньютона

[2]При розрахунках звичайно використовується скалярна сила $F(r) = -\partial U/dr$, а не потенціал, що дозволяє уникнути різницевого диференціювання за простором.

еволюція системи у часі визначається системою:

$$\begin{aligned} \frac{d\overrightarrow{x}_i}{dt} &= \overrightarrow{v}_i, \\ m_i\frac{d\overrightarrow{v}_i}{dt} &= \overrightarrow{F}_i + \sum_{j=1,j\neq i}^{N} F(m_i, m_j, |\overrightarrow{x}_i - \overrightarrow{x}_j|)\frac{(\overrightarrow{x}_i - \overrightarrow{x}_j)}{|\overrightarrow{x}_i - \overrightarrow{x}_j|}, \end{aligned} \quad (2.3)$$

де $\overrightarrow{F}_i$ — рівнодіюча зовнішніх сил, що діють на i-ту частинку з боку тіл, які не входять у систему; $F(m_i, m_j, |\overrightarrow{x}_i - \overrightarrow{x}_j|)$ — внутрішня сила, що діє на i-ту частинку з боку j-ї частинки; множник $(\overrightarrow{x}_i - \overrightarrow{x}_j)/|\overrightarrow{x}_i - \overrightarrow{x}_j|$ вводиться для врахування напрямку дії сили.

Модель (2.3) має широке застосування й охоплює велике число механічних систем. Крім вивчення класичних рідин та твердих тіл (див. нижче), ця модель дозволяє вивчити рух частинки у центрально-симетричному полі іншої частинки, абсолютно пружний і непружний удари, рух молекул газу, дифузію, рух планет, рух взаємодіючих частинок в однорідному полі, рух взаємодіючих частинок у центрально-симетричному полі і т.п.

Труднощі вивчення системи з далекодіючими силами полягають у тому, що кожна частинка ефективно взаємодіє з будь-якою іншою частинкою. Тому в системі з N частинок у загальному випадку необхідно врахувати $N(N-1)/2$ взаємодій. Отже, навіть для чисельних методів число N обмежене зверху величиною порядку 10^3, оскільки виникає необхідність стежити за мільйоном взаємодій. Однак 10^3 частинок досить для прояву статистичних властивостей ансамблю і це дає нам можливість "експериментального" вивчення статистичної механіки.

Розглянемо спочатку різницеву апроксимацію рівнянь (2.3). Як і у випадку однієї частинки (розділ 4), найбільш простий підхід полягає в апроксимації рівнянь за методом з переступом:

$$\begin{aligned} \overrightarrow{x}_i^{\,n} &= \overrightarrow{x}_i^{\,n-2} + \overrightarrow{v}_i^{\,n-1}2\Delta t, \\ \overrightarrow{v}_i^{\,n+1} &= \overrightarrow{v}_i^{\,n-1} + \frac{1}{m_i}\left[\overrightarrow{F}_i(m_i, \overrightarrow{x}_i^{\,n}) + \right. \\ &\quad \left. + \sum_{j=1,j\neq i}^{N} F(m_i, m_j, |\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n}|)\frac{(\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n})}{|\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n}|}\right] 2\Delta t. \end{aligned} \quad (2.4)$$

Ми забезпечимо точне збереження енергії й одночасно спростимо обчислення, якщо будемо використовувати третій закон руху Ньютона: збільшення імпульсу i-ї частинки, обумовлене взаємодією з j-ю частинкою, дорівнює зменшенню імпульсу j-ї частинки, обумовленому i-ю частинкою.

Якщо повний об'єм, зайнятий системою, дорівнює V, то з кожною частинкою можна зв'язати вільний об'єм:

$$\frac{1}{n} = \frac{4\pi}{3}a^3 = \frac{V}{N}. \tag{2.5}$$

Тут n — концентрація частинок; a — характерна відстань взаємодії. Варто вибирати досить малий крок за часом, щоб для усіх i виконувалася умова

$$\Delta t << \frac{a}{|\overrightarrow{v}|}. \tag{2.6}$$

2.2.2 Визначення термодинамічних властивостей класичних середовищ за допомогою усереднення за часом

Можливості безпосереднього застосування наведеної вище моделі до нестаціонарних систем багатьох тіл незначні. Найбільш корисне застосування такої машинної моделі полягає у визначенні термодинамічних (рівноважних) макроскопічних властивостей системи, що досягається усередненням станів окремих частинок за усією системою. Очевидно, для точнішого визначення статистичного середнього потрібно взяти досить велике число частинок у системі. Так, у системах частинок із короткодіючими силами для прояву статистичних властивостей достатньо узяти 1000 частинок. Тому, зокрема, під час вивчення класичних рідин і твердих тіл, де молекулярні сили є короткодіючими (наприклад, ван-дер-ваальсівського типу), моделі двочастинкової взаємодії набули найбільшого поширення.

Якщо відомі потенціали чи сили, що діють між частинками (молекулами), які утворюють класичну рідину, машинна модель системи взаємодіючих частинок дозволяє одержати рівняння стану середовища, описує такі термодинамічні властивості, як фазові переходи, магнітна і діелектрична проникності. Однак часто сили, що діють між молекулами,

невідомі. Незважаючи на це, постулюючи вигляд цих сил, можна за допомогою таких моделей одержати можливі властивості цих систем і внаслідок порівняння з експериментом можна, у свою чергу, визначати поліпшені молекулярні потенціали.

Мікроскопічні чи молекулярні властивості легко усереднити за усіма частинками і за великим числом кроків за часом, у результаті чого будуть отримані шукані макроскопічні термодинамічні властивості. Наприклад, термодинамічна величина — внутрішня енергія — визначається за формулою

$$E = \frac{d}{2}kT + \overline{U}, \tag{2.7}$$

де $d/2kT$ — середня кінетична енергія на одну частинку (d — вимірність простору; k — стала Больцмана; T — температура), а $\overline{U}$ — усереднена потенціальна енергія, що припадає на одну частинку. Для визначення температури усереднимо повну кінетичну енергію системи за багатьма кроками K за часом:

$$\frac{d}{2}kT = \frac{1}{N}\frac{1}{K}\sum_{n=1}^{K}\sum_{i=1}^{N}\frac{1}{2}m(v_i^n)^2 \tag{2.8}$$

й аналогічно усереднимо повну потенціальну енергію:

$$\overline{U} = \frac{1}{2N}\frac{1}{K}\sum_{n=1}^{K}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}U(|\overrightarrow{x}_i^n - \overrightarrow{x}_j^n|). \tag{2.9}$$

2.2.3 Детерміністичний хаос

Зазначимо, що "оборотність" руху, яку в розглянутій задачі забезпечує чисельний метод, є чисто математичною. У рамках комп'ютерної реалізації у системі частинок із часом обов'язково проявиться непередбачуваність за рахунок наростання, підсилення малих невизначеностей[3] — так званий детерміністичний хаос (див. розд. 7). Розглянемо реалізацію зазначеного ефекту "молекулярного хаосу" на прикладі простої

[3] що є наслідком округлення при машинних розрахунках.

моделі, що носить назву "більярд Синая". Це гіпотетичний більярд, у якому кулі не зупиняються з часом. Достатньо велика послідовність зіткнень куль неминуче приводить до наростання малих відхилень від розрахованих траєкторій за рахунок неможливості з нескінченною точністю розрахувати кути зіткнень. Наростання ефекту малих впливів відбувається так швидко, що для розрахунку положень куль вже через хвилину, необхідно враховувати такі надмалі відхилення, що в реальному світі можуть бути викликані впливом гравітації сусідніх галактик!

Алгоритм розрахунку руху куль протягом інтервалу часу Δt є достатньо простим [4]:

1 Визначимо моменти зіткнення для кожної пари куль без врахування можливих перешкод з боку інших куль та стінок, а також моменти зіткнення кожної із куль зі стінками.

2 Виберемо найбільш раннє з числа цих зіткнень (відкинувши попередньо ті, які відбувалися в минулому). Якщо проміжок часу до цього зіткнення t' більший, ніж заданий інтервал Δt, то протягом часу Δt не відбудеться ніяких зіткнень, так що координата i-ї кулі в момент t визначаються очевидним чином:

$$\overrightarrow{r_i}(t) = \overrightarrow{r_i} + \overrightarrow{v_i}\Delta t,$$

де $\overrightarrow{r_i}$, $\overrightarrow{v_i}$ — радіус-вектор та швидкість i-ї кулі в попередній момент часу. Якщо ж $t' < \Delta t$, то потрібно розрахувати зсунення куль за час t', — у результаті цього зсунення пара куль (скажімо, i-та та j-та) досягає зіткнення. Потім розраховуємо зміну швидкостей цієї пари куль, що відбувається у результаті зіткнення.

3 Зменшуємо час на величину $(\Delta t - t')$ і, якщо заданий інтервал часу не вичерпаний, переходимо до п.1, інакше переходимо до розгляду наступного інтервалу часу.

Момент зіткнення кулі, наприклад, із правою стінкою, визначається з рівняння $x + v_x t' = L - R$, де L — лінійний розмір площини, R — радіус кулі. Зміна швидкості при такому зіткненні $v_x = -v_x$. Момент зіткнення пари куль t' (час, коли центри куль зближаються на відстань $2R$) визначається з рівняння

$$\left|\overrightarrow{r_i}(t') - \overrightarrow{r_j}(t')\right| = 2R,$$

або

$$(\overrightarrow{r_i} + \overrightarrow{v_i}t' - \overrightarrow{r_j} - \overrightarrow{v_j}t')^2 = 4R^2. \qquad (2.10)$$

Розв'язуючи отримане квадратне рівняння відносно t' отримуємо шуканий час.

Зміна швидкості після зіткнення розраховується згідно з формулою

$$\overrightarrow{v_i} = \overrightarrow{v_i} - \frac{m_j}{m_i + m_j}\triangle\overrightarrow{v}, \qquad \overrightarrow{v_j} = \overrightarrow{v_j} + \frac{m_i}{m_i + m_j}\triangle\overrightarrow{v}, \qquad (2.11)$$

де $\triangle\overrightarrow{v} = \overrightarrow{n}(\overrightarrow{n}(\overrightarrow{v_i} - \overrightarrow{v_j}))$, $\overrightarrow{n} = (\overrightarrow{r_i} - \overrightarrow{r_j})/2R$.

Поданий алгоритм реалізовано мовою MATLAB у програмі `Sinaj` (наводиться у додатку).

Порівнюючи подані у наступних підрозділах програми (на C++ та MATLAB), можна побачити, що програми такого типу на MATLAB є значно компактнішими та більш прозорими. Оскільки MATLAB оперує цілими матрицями, зникає необхідність виконувати обчислення над компонентами векторів (у даному випадку векторів швидкостей та координат). З іншого боку, програми на C++ виконуються значно швидше, що надає їм переваги за умови значної кількості частинок у системі.

2.3 Алгоритми

Як зазначалося, нашою метою є вивчення систем багатьох частинок ($N \approx 10^{23}$), а модель молекулярної динаміки обмежена лише $N \approx 10^3$ частинками, ми не можемо помістити систему у резервуар із жорсткими стінками[4]. Це обов'язково призведе до значних поверхневих ефектів (відбиття від стінок), що значно зменшить якість отриманих результатів. Для мінімізації подібних ефектів розглянемо періодичні граничні умови. Для двовимірного випадку це означає, що частинка, досягаючи стінки прямокутного плоского резервуара, з'являється з протилежного боку. Інакше кажучи, ми згортаємо плоский резервуар у тор, при цьому протилежні сторони прямокутної області з'єднуються.

[4]Нагадаємо, що стала Авогадро $N_A \approx 10^{23}$ — кількість структурних елементів (атомів, молекул і т.п.) в 1 молі речовини.

Ураховуючи наведене зауваження, подамо алгоритм моделювання системи так:

1 Задаємо параметри фізичної системи: кількість частинок N, лінійні розміри резервуара Lx та Ly, масу частинок m_i, початкові координати кожної частинки $x_i(t=0)$, $y_i(t=0)$; компоненти векторів швидкостей $v_{xi}(t=0)$, $v_{yi}(t=0)$; проекції зовнішніх сил F_{xi}, F_{yi}, параметри потенціалу (ε та σ у випадку потенціалу Леннарда-Джонса), крок за часом $\triangle t$.

2 Розраховуємо повний імпульс системи в $x-$ та $y-$ напрямках і коригуємо швидкості таким чином, щоб повний імпульс системи дорівнював нулю. Для цього окремо підсумовуємо проекції v_{xi} та v_{yi}, знаходимо середнє значення та віднімаємо від кожної проекції отримане середнє.

3 Беремо $t=0$. Згідно із запропонованим методом знаходимо компоненти швидкості $v_{xi}(t+1/2\triangle t)$, $v_{yi}(t+1/2\triangle t)$ у наступний момент часу:

3.1 Візьмемо $i=1$, $a_{xi}=0$, $a_{yi}=0$, $i=1..N$ (i — номер частинки).

3.2 $S_x=0$, $S_y=0$, $j=i+1$.

3.3 Якщо $j>N$, то переходимо до п. 3.5, інакше розраховуємо мінімальну відстань між частинками i та j:

$$\begin{aligned} r_x &= x_i(t)-x_j(t), \\ r_y &= y_i(t)-y_j(t), \\ r &= \sqrt{r_x^2+r_y^2}. \end{aligned} \tag{2.12}$$

Через періодичність граничних умов відстань між частинками не може перевищувати половини лінійного розміру резервуара (рис. 2.1). Відповідно віднімаємо від r_x (або r_y) величину Lx (або Ly), якщо відстань між частинками перевищила $Lx/2$ ($Ly/2$).

Розраховуємо сумарну силу, яка діє на i-ту частинку з боку сусідніх частинок (змінні S_x, S_y) та коригуємо швидкості частинок, які провзає-

модіяли з i-ю частинкою, наприклад, так:

$$\begin{aligned} S_x &= S_x + F(m_i, m_j, r)\frac{r_x}{r}, \\ S_y &= S_y + F(m_i, m_j, r)\frac{r_y}{r}, \\ a_{xj} &= a_{xj} - F(m_i, m_j, r)\frac{r_x}{r}, \\ a_{yj} &= a_{yj} - F(m_i, m_j, r)\frac{r_x}{r}. \end{aligned} \tag{2.13}$$

3.4 Беремо $j = j + 1$ і переходимо до п.3.3

3.5 Ураховуючи третій закон Ньютона (збільшуючи імпульс однієї із взаємодіючих частинок, зменшуємо імпульс іншої), розраховуємо швидкість i-ї частинки на наступному кроці:

$$\begin{aligned} v_{xi}\left(t + \frac{1}{2}\triangle t\right) &= v_{xi}(t) \quad + \quad (S_x + a_{xi})\frac{1}{2}\triangle t, \\ v_{yi}\left(t + \frac{1}{2}\triangle t\right) &= v_{yi}(t) \quad + \quad (S_y + a_{yi})\frac{1}{2}\triangle t. \end{aligned} \tag{2.14}$$

3.6 Беремо $i = i + 1$. Якщо $i = N$, то переходимо до наступного пункту, інакше до пункту 3.2.

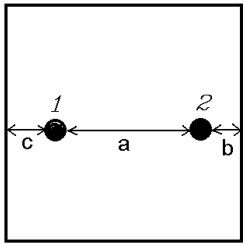

Рисунок 2.1 — Мінімальна відстань між частинками 1 та 2 становить $r_x = b + c, r_x < a$ через періодичність прямокутного резервуара

4 Розраховуємо координати частинок у наступні моменти часу.

4.1 Припускаємо, що $t = 0$.

4.2 Цикл по i до N. Перебираємо всі частинки, перераховуючи координати згідно з формулою:

$$\begin{aligned} x_i(t+\triangle t) &= x_i(t) + v_{xi}\left(t+\frac{1}{2}\triangle t\right)\triangle t, \\ y_i(t+\triangle t) &= y_i(t) + v_{yi}\left(t+\frac{1}{2}\triangle t\right)\triangle t. \end{aligned} \tag{2.15}$$

Корегуємо координати за умови виходу за межі резервуара.

4.3 Виконуємо розрахунки згідно з пунктами 3.1-3.6.

4.4 Отримані швидкості та координати застосовуємо для розрахунків термодинамічних характеристик системи, виводимо на екран або у файл.

4.5 Збільшуємо час на крок $\triangle t$: $t = t + \triangle t$ та переходимо до п.4.2. Якщо цикл по t закінчився — вихід із програми.

2.4 Приклади

2.4.1 Рух системи взаємодіючих частинок

Подана у додатку програма дозволяє провести моделювання руху N частинок з масами $m_i = 1$, взаємодія яких описується потенціалом Леннарда-Джонса. Щоб уникнути зайвих ускладнень, ми не враховуємо дію зовнішнього поля ($\overrightarrow{F}_i = 0$). Програма дозволяє візуалізувати рух частинок у прямокутному резервуарі та простежити за зміною перенормованої температури системи.

Результати роботи програми зображені на рис. 2.2.

За допомогою розробленої програми був проведений ряд експериментів, пов'язаних з вимірюванням температури у системі. У випадку, коли початковий розподіл не є рівномірним уздовж площини резервуара, на графіку температури простежується певний перехідний режим, що супроводжується значними флуктуаціями. З часом температура набуває певного стаціонарного значення з невеликими відхиленнями в око-

лі середнього. Флуктуації температури можна зменшити, збільшуючи кількість частинок у системі. При рівномірному початковому розподілі частинок перехідний режим практично не простежується (рис. 2.2), при цьому, як бачимо з рисунка, значення перенормованої температури становить $T \approx 240$.

2.4.2 "Більярд Синая"

Подана у додатку програма дозволяє провести моделювання руху 2 куль однакової маси радіусом R у прямокутному резервуарі з лінійними розмірами $Lx \times Ly$. Програма складається з двох M-файлів: головний модуль Sinaj та функція Raschet, що виконує розрахунок траєкторії руху куль на заданому проміжку часу $\triangle t$.

Питання для самоконтролю

1 У чому полягає суть методу молекулярної динаміки?

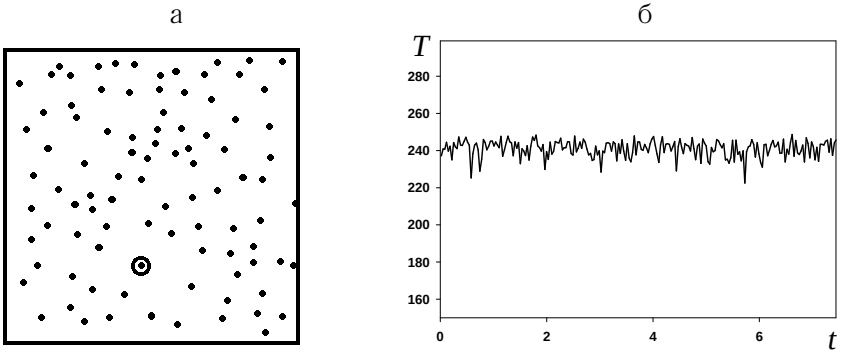

Рисунок 2.2 — Чисельне моделювання руху N взаємодіючих частинок з потенціалом міжмолекулярної взаємодії Леннарда-Джонса:
а) візуалізація руху частинок у періодичному резервуарі;
б) експериментально отримана залежність температури T від часу t

2 У яких випадках (з якою метою) застосовується метод молекулярної динаміки?

3 Які обмеження накладаються на систему при застосуванні методу молекулярної динаміки?

4 Які фізичні параметри систем багатьох частинок можна обчислити за допомогою комп'ютерного моделювання?

5 Дайте визначення детерміністичного хаосу. Назвіть причини його появи.

6 Чи завжди рух системи багатьох частинок є непередбачуваним?

7 Яким чином можна уникнути впливу граничних умов ("ефекту стінки") при точному моделюванні руху N частинок?

Завдання для самостійної роботи

1 Використовуючи наведений у розділі алгоритм, промоделюйте поведінку системи із 200 частинок та визначте повну енергію системи, температуру та тиск. Критерій рівноваги підберіть самостійно.

2 Проведіть моделювання системи багатьох частинок на значних інтервалах часу. Зачекайте, поки система не опиниться у стані рівноваги. Визначте рівноважну густину ймовірності $P(v)$ того, що швидкість частинки знаходиться в інтервалі від v до $v + \Delta v$; побудуйте відповідний графік. Для розв'язання задачі розбиваємо інтервал швидкостей $[0..v_{max}]$ на n частин і визначаємо кількість частинок, швидкості яких лежать у межах кожного проміжку даного інтервалу. Відповідно ймовірність реалізації певної швидкості дорівнює відношенню кількості частинок у цьому інтервалі до загальної кількості частинок у системі. Результуючий графік побудуйте шляхом усереднення результатів принаймні за 200 кроками за часом. Чи збігається отримана залежність з відомим розподілом Максвелла-Больцмана?

3 Отримайте залежність повної енергії системи від температури (у рівноважному стані). Оцініть окремо вклади кінетичної та потенціальної енергії та побудуйте відповідні графіки.

4 Розгляньте систему із N частинок. Використовуючи відому формулу Ейнштейна, розрахуйте коефіцієнт самодифузії:

$$D = \frac{R(t)^2}{2dt}, \quad (t \to \infty)$$

де d — вимірність простору; $R(t)^2$ — середній квадрат зсуву, розрахований за формулою

$$R(t)^2 = \langle |r_i(t_2) - r_i(t_1)|^2 \rangle,$$

де r_i — координата i-ї частинки; $\langle\rangle$ — оператор усереднення за всіма частинками.

5 На основі комп'ютерної моделі вивчіть рух двох або трьох частинок, між якими діють сили притягання.

6 Вивчить рух двох матеріальних частинок, між якими діють сили відштовхування. Промоделюйте центральний та нецентральний удари.

7 Промоделюйте розрив снаряду на кілька осколків різної маси в однорідному полі тяжіння. Після вибуху виникає сила відштовхування, що швидко зменшується у міру віддалення осколків.

8 Вивчіть рух матеріальної точки в гравітаційному полі двох масивних тіл. Припустіть обчислювальні експерименти при різних початкових координатах і швидкостях точки.

9 Промоделюйте рух декількох планет Сонячної системи. Припустіть, що маса Сонця в багато разів більша за масу будь-якої планети системи.

10 Промоделюйте рух молекул газу в прямокутній замкнутій посудині. Врахуйте, що при зіткненні молекули з горизонтальною (вертикальною) стінкою посудини вертикальна (горизонтальна) проекція її швидкості змінює свій знак на протилежний.

11 Промоделюйте дифузію двох газів. Нехай спочатку молекули з масами m заповнюють ліву половину посудини, а молекули з масами M — праву. Задайте випадкові значення швидкостей молекул. Як змінюється концентрація молекул газів у посудині з часом?

12 Промоделюйте рух молекул газу в однорідному полі тяжіння. Доведіть, що в міру збільшення висоти концентрація молекул газу зменшується за експоненціальним законом.

13 Промоделюйте рух молекул газу в гравітаційному полі кулі великої маси.

14 Використовуючи програму Sinaj, переконайтеся у наявності хаосу у системі двох куль. Для цього у деякий момент часу t поміняйте швидкості куль на протилежні $\overrightarrow{v_i} = -\overrightarrow{v_i}$ та простежте, чи будуть кулі повертатися у вихідний стан "прокладеними"раніше траєкторіям. Для якісного дослідження досить зафіксувати шляхи частинок на екрані (використовуючи замість маркера 'o' маркер '.' і вибравши для нього варіант виведення без видалення попереднього зображення (`set(plothandle,'EraseMode','none'))`,а при повторному запуску куль можна змінити колір траєкторій.

15 Отримайте на екрані картину розподілу частинок у площині (v_x, v_y).

Розділ 3

Моделювання систем багатьох частинок методом Монте-Карло

У розділі розглядається "швидкий" статистичний спосіб вивчення систем багатьох частинок — метод Монте-Карло [5]. Метод грунтується на статистичній механіці Гіббса і використовує ідею усереднення за ансамблем замість усереднення за часом, що дозволяє уникнути розв'язання диференціальних рівнянь руху системи частинок. Як приклад метод застосовується для вивчення фазових перетворень у відомій із теорії магнетизму моделі Ізинга.

3.1 Постановка задачі

Проведіть мікроканонічне моделювання двовимірної моделі Ізинга методом Монте-Карло з використанням динаміки перевертання спіну. Розрахуйте намагніченість системи та проаналізуйте її залежність від температури.

3.2 Теоретичний матеріал

3.2.1 Мікроканонічний ансамбль

Розглянемо замкнуту систему, що складається з N елементів (частинок). За макроскопічні характеристики системи оберемо об'єм V та повну енергію E, які будемо вважати сталими. Крім того, припустимо, що система є ізольованою, тобто впливом на неї зовнішніх факторів можна знехтувати. Відзначимо, що якщо на макроскопічному рівні система характеризується трьома величинами N, V, E, то на мікроскопічному рівні існує величезне число конфігурацій, що реалізують заданий макростан. Таким чином, на можливі мікростани накладається єдине обмеження: вони повинні бути такими, щоб забезпечити зазначені макроскопічні характеристики системи. Як відомо [6], для такої системи виконується по-

стулат про рівність апріорних імовірностей: система у будь-який момент часу може опинитися з однаковою ймовірністю в одному з можливих мікростанів. Це означає, що для системи з W досяжними станами ймовірність знайти систему у мікростані ν дорівнює

$$P_\nu = 1/W. \tag{3.1}$$

Для визначення середніх значень фізичних величин, що характеризують макроскопічну систему, можна використовувати два способи. Перший спосіб — вимірюють фізичні величини протягом досить великого проміжку часу, за який у системі реалізується велика кількість досяжних мікростанів, і проводять усереднення. Подібне усереднення у рамках методу молекулярної динаміки виконувалось у попередньому розділі. Другий спосіб — статистичний, що базується на ергодичній гіпотезі[5]. Суть цього методу полягає у такому. Незважаючи на очевидний фізичний зміст середніх за часом, у статистичній фізиці вводять поняття статистичних середніх у даний момент часу, що обчислюються за ансамблем мікростанів, які відповідають заданому макростану. Тобто замість розгляду еволюції однієї системи у часі розглядається набір, або *ансамбль*, систем, що відповідають одному й тому самому макроскопічному стану. Повне число систем в ансамблі дорівнює числу можливих мікростанів. Імовірність знайти в ансамблі тотожних систем ν-ту реалізацію (конфігурацію елементів (частинок) визначається виразом (3.1). Ансамбль систем, що характеризується величинами N, E, V та описується розподілом імовірностей виду (3.1), називається *мікроканонічним ансамблем*.

Будемо вважати, що система характеризується деякою величиною m, значення якої в ν-му стані дорівнює m_ν. Тоді середнє значення m за ансамблем визначається як

$$\langle m \rangle = \sum_{\nu=1}^{W} m_\nu P_\nu. \tag{3.2}$$

[5]Гіпотеза про збіг середніх значень величин, що обчислюються усередненням за часом та усередненням за ансамблем реалізацій, називається ергодичною гіпотезою.

З останнього рівняння бачимо, що для розрахунку середнього за ансамблем необхідно знати значення ймовірностей P_ν. При невеликій кількості елементів значення ймовірностей можна визначити методом перебору: фіксуючи N та V, розглянути всі мікростани, що забезпечують повну енергію E. Однак при збільшенні числа елементів цей метод стає неприйнятним через великий об'єм обчислень. У цих умовах доводиться використовувати наближені методи, серед яких одним із найефективніших є метод Монте-Карло.

3.2.2 Метод Монте-Карло

Загальна ідея методу Монте-Карло для мікроканонічного ансамблю складається в одержанні репрезентативної вибірки з повного числа мікростанів й оцінці з її використанням значення $\langle m \rangle$:

$$\langle m \rangle = \frac{1}{M} \sum_{i=1}^{M} m_i, \tag{3.3}$$

де M — велике число — кількість систем у вибірці. У найпростішому вигляді алгоритм одержання репрезентативної вибірки має такий вигляд:

1. Фіксуємо повне число частинок N та об'єм системи V.

2. Змінюємо випадковим чином координати і швидкості окремих частинок.

3. Беремо конфігурації мікростанів, що мають задану повну енергію.

Необхідно зазначити, що ця процедура виявляється досить неефективною, тому що більшість конфігурацій узагалі не матимуть необхідної повної енергії і повинні бути відкинуті.

Розглянемо більш ефективну процедуру, яка враховує вимоги статистичної механіки, а саме: ансамбль буде канонічним, якщо ймовірність того, що система належить ансамблю, пропорційна $\exp(-E/kT)$. Такий ансамбль можна побудувати чисельно за допомогою процесу Маркова.

Алгоритм побудови ансамблю:

1. Нехай N частинок спочатку розподілені усередині об'єму V будь-яким довільним чином. Візьмемо певну частинку (наприклад, $\mu = 1$) і

перемістимо її випадковим чином у будь-яку точку квадрата зі стороною a:

$$\begin{aligned} x_\mu^{i+1} &= x_\mu^i + a(R_1 - \frac{1}{2}), \\ y_\mu^{i+1} &= y_\mu^i + a(R_2 - \frac{1}{2}), \end{aligned} \tag{3.4}$$

де R_1 і R_2 — випадкові числа з інтервалу $0 \leq R \leq 1$.

2. У результаті на кроці $i+1$ утвориться нова система, стан якої описується вектором $(x_1^{i+1}, y_1^{i+1}, .., x_N^{i+1}, y_N^{i+1})$. Оскільки випадкові числа рівноймовірні, то новий стан отримано способом, що відповідає вимозі рівності апріорних імовірностей. При переміщенні частинки відбулася зміна енергії ΔE, яку можна обчислити так:

$$\Delta E = \sum_{\nu=1,\nu\neq\mu}^{N} \left[U(|\overrightarrow{x}_\mu^{i+1} - \overrightarrow{x}_\nu^{i+1}|) - U(|\overrightarrow{x}_\mu^{i} - \overrightarrow{x}_\nu^{i}|) \right], \tag{3.5}$$

де U — потенціал взаємодії.

3. Перед нами є задача побудувати канонічний ансамбль так, щоб частота появи систем, що його утворюють, була пропорційною до величини $\exp(-E/kT)$. Такий розподіл за ансамблем можна одержати, вирішуючи питання, буде отримана система дозволеною чи ні. Якщо $\Delta E < 0$, тобто переміщення приводить до системи з меншою енергією, то система включається в ансамбль. З іншого боку, якщо $\Delta E > 0$, тобто утворилася система з більшою енергією, то система включається в ансамбль лише з імовірністю $\exp(-\Delta E/kT)$. Для цього вибирається третє випадкове число R_3, і якщо $R_3 < \exp(-\Delta E/kT)$, то нова система приєднується до ансамблю, а якщо ні, то нова система відкидається, і до ансамблю ще один раз додається стара система. На цьому даний крок завершений, і протягом наступного кроку переміщується інша частинка μ'.

Очевидно, що чим більше систем міститься в ансамблі, тим кращий буде розв'язок, але для задоволення умови рівності апріорних імовірностей необхідно побудувати принаймні $M_{min} = NL/a$ систем. Вибір параметра a теж може бути довільним, однак установлено, що при визначенні термодинамічних величин збіжність буде більш швидкою, якщо

значення a буде порядку відстані між частинками. Шукану термодинамічну змінну можна одержати просто як середнє за ансамблем.

У загальному випадку метод усереднення за ансамблем за методом Монте-Карло значно перевищує за швидкодією прямий метод усереднення за часом з використанням детерміністичних рівнянь руху. Однак цей метод зорієнтований лише на рівноважні системи, що обумовлює більш вузьку область його застосування.

3.2.3 Ідеальний газ

Розглянемо методику застосування методу Монте-Карло до вивчення фізичних характеристик простих систем, типу класичного ідеального газу (швидкості частинок неперервні та необмежені, енергія частинок не залежить від їх положення, повна енергія є сумою кінетичних енергій окремих частинок). У даному випадку можна запропонувати спрощену процедуру Монте-Карло, запропоновану в [5].

Цей метод полягає у поділі вихідної макроскопічної системи на дві підсистеми: вихідну, яку в подальшому будемо називати просто системою, і підсистему, що складається з одного елемента, який ми назвемо "конденсатором". Обходячи елементи системи і передаючи енергію, "конденсатор" забезпечує зміну конфігурації системи. Якщо енергії, яка накопичена у "конденсаторі", виявляється досить, він віддає енергію тому елементу системи, якому потрібна енергія для здійснення зміни конфігурації. І навпаки, якщо для зміни конфігурації потрібно зменшити енергію системи, енергія частково передається "конденсатору". Єдине обмеження полягає в тому, що енергія "конденсатора" повинна бути позитивною. Описана процедура алгоритмічно має такий вигляд:

1 Задаємо початкову конфігурацію системи із заданим значенням повної енергії.

2 Вибираємо випадковим чином частинку і робимо спробу змінити її швидкість.

3 Обчислюємо повну енергію системи у новому стані.

4 Якщо у новому стані енергія системи виявляється меншою, то система віддає енергію "конденсатору" і нова конфігурація приймає-

ться. Якщо у новому стані енергія системи виявляється більшою, то нова конфігурація приймається, якщо енергії "конденсатора" досить, щоб передати її системі. У протилежному разі нова конфігурація не приймається і частинка зберігає свою вихідну швидкість.

6 Якщо зміна швидкості не змінює енергію системи, то нова конфігурація приймається.

Перелічені вище дії повторюються до одержання репрезентативної вибірки мікростанів. Очікується, що через деякий проміжок часу система досягне рівноважного стану, в якому система та "конденсатор" матимуть деякі середні (для кожного) значення енергії. При цьому значення повної енергії системи залишається сталим.

Зазначимо, що моделювання мікроканонічного ансамблю при сталій енергії проводиться незалежно від температури. При цьому температура системи може бути визначена на основі кінетичних енергій усіх частинок із співвідношення $(1/2)m\langle v^2\rangle = (d/2)k_BT$, де d — вимірність простору.

Відмітимо, що мікроканонічний ансамбль не можна вважати абсолютно адекватною моделлю реальних статистичних систем, оскільки реальні системи не є ізольованими, а знаходяться у тепловому контакті з навколишнім середовищем. Оскільки розмір системи, що досліджується, як правило, набагато менший за розмір навколишнього середовища, її звичайно називають мікросистемою, а оточуюче середовище — тепловим резервуаром. При цьому закон збереження енергії належить до всієї системи, що складається з мікросистеми та теплового резервуара. Розглянемо велике число уявних копій мікросистеми і теплового резервуара, що можуть бути описані за допомогою мікроканонічного ансамблю. При цьому мікросистему разом із тепловим резервуаром можна вважати ізольованою термодинамічною системою. Оскільки найбільший інтерес становлять рівноважні значення фізичних величин, що описують мікросистему, необхідно знати ймовірність, з якою мікросистема опиниться у стані із заданою енергією. Ансамбль, який описує розподіл імовірностей станів мікросистеми, що знаходиться в термодинамічній рівновазі з тепловим резервуаром, називається *канонічним ансамблем* . У загальному випадку, мікросистемою можна вважати одну або декілька частинок серед інших частинок теплового резервуара. При-

кладом такої мікросистеми є "конденсатор" (див. поперед. підрозділ), який можна вважати мікросистемою, мікроканонічний стан якої визначається лише її енергією E_c. Таким чином, метод відшукання розподілу ймовірностей у канонічному ансамблі полягає у тому, щоб виконати чисельне моделювання "конденсатора", що обмінюється енергією з ідеальним газом та визначити густину розподілу $p(E_c)$. Обчислювальний алгоритм, що дозволяє розв'язати цю задачу, досить очевидний:

1 Проводимо моделювання канонічного ансамблю методом Монте-Карло, зберігаючи на кожному кроці Монте-Карло значення енергії "конденсатора" E_c.

2 Для знайденої реалізації значень енергії "конденсатора" обчислюємо розподіл імовірностей $P(E_c)\triangle E$.

3 Визначаємо функціональну залежність $P(E_c)\triangle E$ від температури теплового резервуара.

Зазначимо, що при коректному обчисленні отриманий розподіл енергії "конденсатора" матиме вигляд канонічного розподілу (розподілу Больцмана) [6]. Цей розподіл прийнято записувати у такому вигляді:

$$P(E_c) = \frac{1}{Z} \exp\left(-\frac{E_c}{k_B T}\right), \tag{3.6}$$

де Z - нормувальний множник. Як бачимо, канонічний розподіл характеризується температурою. Звідси робимо висновок, що у канонічному ансамблі будь-який макростан визначається параметрами T, N, V.

Остання формула забезпечує простий спосіб обчислення температури на основі середньої енергії "конденсатора", яка, за визначенням, дорівнює

$$\langle E_c \rangle = \frac{\int_0^\infty E_c \exp\left(-\frac{E_c}{k_B T}\right) dE_c}{\int_0^\infty \exp\left(-\frac{E_c}{k_B T}\right) dE_c} = k_B T. \tag{3.7}$$

Таким чином, середня енергія "конденсатора" дорівнює температурі теплового резервуара (в одиницях k_B). Цей результат справедливий лише за умови, якщо енергія набуває неперервних значень. При комп'ютерному моделювання температура буде дещо відрізнятися від передбачуваного значення (3.7).

3.2.4 Модель Ізинга

Розглянемо одновимірну гратку із N закріплених вузлів (спінів), кожний з яких характеризується числом s_i, де $s_i = +1$, якщо спін спрямований "вгору", і $s_i = -1$, якщо він спрямований "униз". Таким чином, будь-яка конкретна конфігурація такої системи задається набором змінних $s_1, s_2, .., s_N$. Повна енергія системи залежить від конфігурації спінів і визначається за формулою

$$E = -J \sum_{i,j}^{N} s_i s_j - h \sum_{i}^{N} s_i, \tag{3.8}$$

де перша сума береться за всіма парами спінів, які є сусідніми, друга — за усіма спінами. J — константа обмінної взаємодії; константа h характеризує зовнішнє поле. Помітимо, що коли $J > 0$, то стани з однонапрямленими спінами $\uparrow\uparrow$ (або $\downarrow\downarrow$) є енергетично вигіднішими, ніж пари різнонапрямлених спінів ($\downarrow\uparrow$), оскільки така конфігурація зменшує енергію системи. Відповідно при застосуванні цієї моделі до опису фазових перетворень у магнетиках необхідно очікувати, що при $J > 0$ стан з мінімальною повною енергією спінів є феромагнітним, тобто таким, коли середнє сумарне число спінів, орієнтованих в одному напрямку, не дорівнює нулю. Якщо $J < 0$, більш вигіднішими є стани зі спінами протилежної спрямованості, і стан системи є антиферомагнітним.

Для вивчення часової поведінки системи (наприклад, переходу системи у рівноважний стан) вводять так звану динаміку "перевертання спіну". У цій динаміці на кожному кроці ітераційного процесу випадковим чином вибирається і перевертається ($s_i = -s_i$) спін, якщо новий стан спіну є енергетично вигіднішим (метод Монте-Карло). Цей алгоритм, наприклад, є придатним наближенням до реальної динаміки анізотропного магніту, спіни якого пов'язані з коливаннями гратки.

При моделюваннях системи Ізинга практичне значення має розрахунок середньої намагніченості

$$\langle m \rangle = \langle \sum_{i}^{N} s_i \rangle, \tag{3.9}$$

теплоємності

$$C = \frac{1}{k_B T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right), \tag{3.10}$$

магнітної сприйнятливості (за відсутності зовнішнього поля)

$$\chi = \frac{1}{k_B T} \left(\langle m^2 \rangle - \langle m \rangle^2 \right). \tag{3.11}$$

Оскільки нас цікавлять властивості нескінченної системи, при моделюванні бажано ввести періодичні граничні умови (рис. 3.1а). Крім того, помітимо, що для двовимірної моделі Ізинга ймовірність перевертання спіну зводиться до п'яти випадків, поданих на рис. 3.1б, що може бути використано для оптимізації розрахункових витрат.

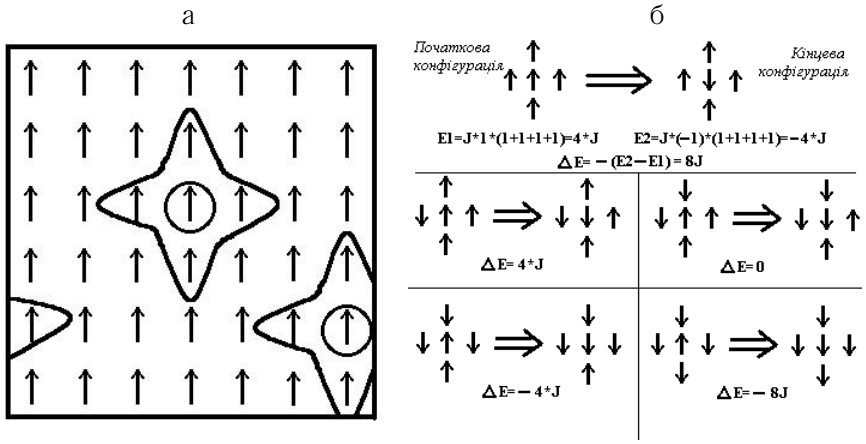

Рисунок 3.1 — Модель Ізинга: (а) тороїдальні періодичні граничні умови (на рисунку показані сусідні спіни до спінів, окреслених колами); (б) п'ять можливих конфігурацій, які описують зміну енергії при перевертанні центрального спіну

3.3 Алгоритми

Розглянемо алгоритм реалізації методу Монте-Карло для дослідження моделі Ізинга.

[1] Випадковим чином формуємо початкову конфігурацію (випадкові напрямки спінів).

[2] Виконуємо випадкову зміну в поточній конфігурації. Наприклад, випадковим чином вибираємо спін і перевертаємо його.

[3] Розраховуємо зміну енергії $\triangle E$ системи, обумовлену зміною конфігурації.

[4] Якщо $\triangle E \leq 0$, то беремо нову конфігурацію. Якщо $\triangle E > 0$, то розраховуємо ймовірність переходу у новий стан $P = \exp(-\triangle E/k_B T)$. Генеруємо випадкове число R: $0 \leq R \leq 1$. Якщо $R \leq P$, беремо нову конфігурацію, інакше зберігаємо попередню конфігурацію.

[5] Якщо виконано достатньо ітерацій за методом Монте-Карло (система наближається до рівноваги), переходимо до наступного пункту, інакше переходимо до кроку 2.

[6] Розраховуємо фізичні параметри системи.

[7] Переходимо до кроку 2 або завершуємо ітераційну процедуру при виконанні значної кількості ітерацій.

Для виведення поточної просторової конфігурації спінів у MATLAB зручно використовувати графічні функції *quiver3* (тривимірне векторне поле) або *bar3* (тривимірна діаграма). Наприклад, якщо поточна конфігурація спінів міститься у двовимірному масиві S, то застосування функції *quiver3* можливе у такий спосіб:

```
i=1:9;
j=1:9;
% Задаємо координати точок у тривимірному просторі
% для виведення векторних стрілок; у нашому випадку
% (U,V,S) - компоненти вектора, Z - площина, яка є
% основою векторних стрілок.
U(i,j)=0;
V(i,j)=0;
Z(i,j)=0;
quiver3(Z,U,V,S);
```

Результат роботи функції *quiver3* поданий на рис.3.2а. Аналогічним чином можна застосувати функцію *bar3*:

```
bar3(S);
```

Результат роботи функції *bar3* поданий на рис.3.2б.

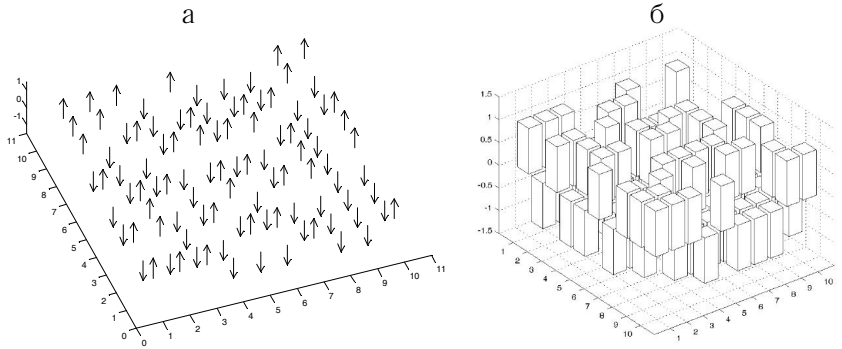

Рисунок 3.2 — Графічне зображення поточної конфігурації спінів за допомогою функцій *quiver3* (а) та *bar3* (б)

3.4 Приклади

3.4.1 Ідеальний газ

Виконаємо чисельне моделювання "конденсатора", що обмінюється енергією з ідеальним газом, та визначимо густину розподілу $p(E_c)$. Для цього, фіксуючи енергію "конденсатора"кожного разу при зміні швидкості певної частинки, отримуємо масив можливих значень енергії E_c. Далі, поділяючи діапазон можливих значень енергії на окремі ділянки, визначаємо, скільки разів енергія системи (протягом експерименту) потрапила в ту чи іншу ділянку. У результаті отримуємо шукану густину розподілу. Приклад програми, що виконує таке моделювання, наводиться у додатку.

Результати роботи програми:

`E_c =0.5910` (середня енергія конденсатора);

`Aver_v = -5.0861e-004` (середня швидкість);

`Num = 0.4665` (відносна кількість прийнятих конфігурацій).

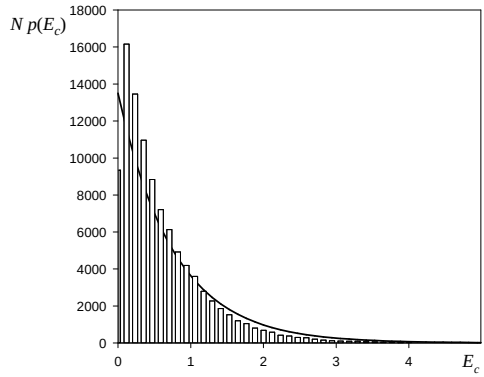

Рисунок 3.3 — Густина розподілу енергії лабораторної системи (" конденсатора"). Суцільна крива — результат апроксимації

Аналізуючи результати роботи програми, можна зробити такі висновки:

1. Кількість конфігурацій, які прийняті до ансамблю, становить 50% від загальної кількості розглянутих систем.

2. Середня швидкість, що припадає на кожну частинку, як це й очікувалося, фактично дорівнює нулю.

3. Як бачимо з рис. 3.3, розподіл імовірностей описується експоненціальною залежністю, яка має вигляд розподілу Больцмана.

4. Аналізуючи результати апроксимації, робимо висновок, що температура теплового резурвуара ($b = 0.66$) у межах точності моделі приблизно дорівнює середній енергії конденсатора ($E_c = 0.591$), що добре узгоджується з (3.7).

3.4.2 Модель Ізинга

Подана у додатку програма дозволяє провести моделювання двовимірної моделі Ізинга з періодичними межами методом Монте-Карло. Припускається, що система знаходиться у тепловому резервуарі із температурою T, зовнішнє поле відсутнє. Програма виводить на екран залежність середньої намагніченості від знерозміреної температури резервуара, при цьому реєстрація зміни намагніченості відбувається на кожному кроці методу Монте-Карло. Результати роботи програми зображені на рис. 3.4.

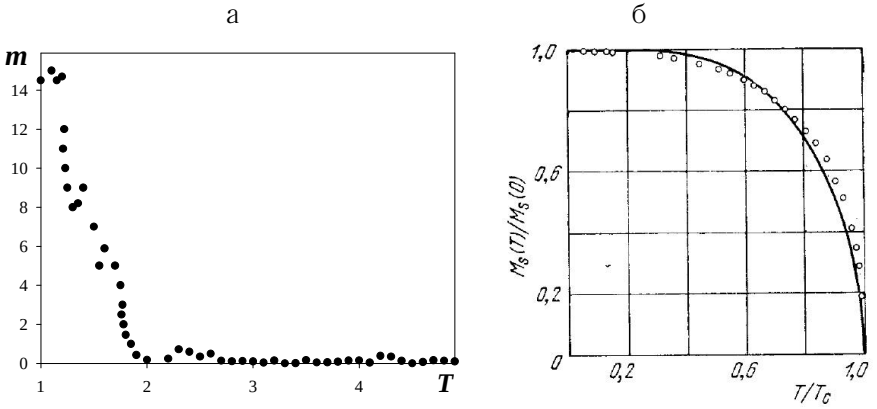

Рисунок 3.4 — (а) Модель Ізинга. Експериментально отримана залежність намагніченості $\langle m \rangle$ від температури T теплового резервуара. Результати моделювання на гратці розміром 64x64. Вихідні дані: $J/k_B = 1$, зовнішнє поле відсутнє $h = 0$, кількість ітерацій за методом Монте-Карло $N_{iter} = 1000$. (б) Намагніченість нікелю як функція температури. Суцільна крива — теоретична, побудована на основі теорії середнього поля. Точки — експеримент. T_c — температура Кюрі. Дані взяті з роботи [7]

На рис. 3.4 показана залежність намагніченості від температури, розрахована в рамках проведеного комп'ютерного експерименту (рис.3.4а)

та реального експерименту з нікелем (рис. 3.4б). Як бачимо, результати якісно збігаються, що свідчить про адекватність моделі Ізинга при описі переходів феромагнетик-парамагнетик у реальних системах. Більш плавну залежність $\langle m \rangle(T)$ можна отримати, збільшивши розмір системи та кількість ітерацій методу Монте-Карло. Як бачимо з рисунка, при збільшенні температури намагніченість поступово зменшується і стає нульовою при певному критичному значенні T_c. Така поведінка намагніченості дає підставу віднести цей перехід із феромагнітного стану до парамагнітного до числа фазових переходів другого роду. Як відомо з теорії магнетизму, критична температура, при якій зникає намагніченість, називається точкою Кюрі ($T_c = 358^o C$).

Питання для самоконтролю

1 У чому полягає суть методу Монте-Карло? Назвіть недоліки та переваги цього методу.

2 Назвіть основні ідеї вивчення систем багатьох частинок за допомогою методу Монте-Карло.

3 Якими факторами визначається точність методу Монте-Карло?

4 У чому полягає різниця між методами молекулярної динаміки та методом Монте-Карло?

5 Назвіть основні етапи побудови мікроканонічного ансамблю.

6 Які обмеження накладаються на систему багатьох частинок при застосуванні методу Монте-Карло?

7 Наведіть алгоритм моделювання системи "Ідеальний газ".

8 Назвіть основні ідеї, покладені в основу моделі Ізинга

9 Яким чином можна уникнути врахування граничних умов у моделі Ізинга?

10 Назвіть основні чинники, що впливають на точність моделювання спінових систем за допомогою моделі Ізинга.

11 Які фізичні параметри спінових систем можна розрахувати за допомогою моделі Ізинга?

Завдання для самостійної роботи

1 Промоделюйте рух однієї класичної частинки наведеним у розділі методом Монте-Карло. Швидкість частинки змінюється на випадкову величину $\pm\triangle v$. Частинка повністю характеризується своєю кінетичною енергією. При заданій температурі T та початковій швидкості v_0 знайдіть середню енергію $\langle E\rangle$ та середню швидкість $\langle v\rangle$. Побудуйте залежність густини ймовірності P реалізації стану з енергією E. Переконайтеся, що отримана залежність є розподілом Больцмана.

2 Використовуючи умови задачі 1, розгляньте випадок N невзаємодіючих частинок (ідеальний газ). Надайте всім частинкам однакову початкову швидкість v_0.

3 Використовуючи умову задачі 2, оцініть час переходу системи у стан теплової рівноваги. Критерій переходу у рівноважний стан підберіть самостійно.

4 Використовуючи умову задачі 2, побудуйте залежність середньої енергії, яка припадає на одну частинку, від температури резервуару T. Середню енергію розраховуйте після переходу системи у рівноважний стан.

5 Використовуючи умову задачі 2, оцініть середній квадрат флуктуацій енергії $\langle\triangle E^2\rangle = \langle E^2\rangle - \langle E\rangle^2$ при різних температурах.

6 Промоделюйте поведінку одновимірної моделі Ізинга з періодичними межами (ланцюжок замкнутий у кільце) за відсутності зовнішнього поля. Візуально простежте за зміною конфігурації спінів

і оцініть час переходу системи у рівноважний стан. Реалізуйте розрахунок намагніченості та енергії системи.

7 Використовуючи умову задачі 6, проведіть моделювання за умови наявності зовнішнього поля ($h \neq 0$).

8 Використовуючи двовимірну модель Ізинга, розрахуйте теплоємність та магнітну сприйнятливість системи. При розрахунках необхідно врахувати, що перехід у рівноважний стан (який є базовим для обчислень фізичних параметрів) із початкової конфігурації займає певний проміжок машинного часу.

9 Використовуючи двовимірну модель Ізинга, оцініть час переходу системи у рівноважний стан. Критерій рівноваги підберіть самостійно.

10 Використовуючи умову задачі 8, нарисуйте графіки залежності середньої енергії $\langle E \rangle$, теплоємності C та сприйнятливості χ від температури T. При якій температурі відбувається фазовий перехід (зникає намагніченість)?

11 Використовуючи кінетичне визначення температури, відповідно до якого для одновимірної моделі температура визначається із співвідношення $m\langle v^2 \rangle/2 = T/2$ (де $m\langle v^2 \rangle/2$ — середня кінетична енергія на частинку, T — температура в одиницях k_B), та наведену у додатку функцію `Ideal_gas`, отримайте "кінетичну" температуру T газу.

12 Використовуючи функцію `Ideal_gas` (наведена у додатку), побудуйте графік густини ймовірності $P(E_c)$ того, що "конденсатор" має енергію E_c. Покажіть, що отримана залежність є розподілом Больцмана (або канонічним розподілом):

$$P(E_c) = Ae^{-\frac{E_c}{T}}.$$

13 Використовуючи функцію `Ideal_gas` (наведена у додатку), проведіть дослідження залежності точності визначення температури від числа частинок системи та числа кроків методу Монте-Карло.

Розділ 4

Моделювання дифузійних процесів

У розділі розглядаються ключові питання щодо побудови моделей дифузійних процесів. Наводяться основні чисельні методи розв'язання рівнянь у частинних похідних та визначаються межі їх застосовності. На наочному прикладі моделювання поширення тепла у прямокутній пластинці показані основні прийоми програмної реалізації наведених у розділі алгоритмів.

4.1 Постановка задачі

Є прямокутна пластина зі сталим коефіцієнтом температуропровідності D. Задано початковий розподіл температури $T = T(x, y)$, потужність W_i та координати (x_i, y_i) джерел тепла. Проаналізуйте зміну температури різних точок пластини з часом.

4.2 Теоретичний матеріал

4.2.1 Рівняння дифузії

У цьому розділі ми будемо використовувати комп'ютер для чисельного розв'язання рівняння у частинних похідних (рівняння дифузії), що описує багато фізичних явищ, наприклад, тепловий потік або дифузію домішок у нерухомій рідині. Перебіг дифузії можна уявити як процес розчину краплі фарби у склянці води. Фарба складається із великої кількості окремих частинок, кожна з яких неодноразово відштовхується від молекул води, що її оточують, тобто рухається випадковим щляхом. Через величезну кількість частинок фарби їхні індивідуальні випадкові рухи утворюють надзвичайно детерміновану загальну модель. Для побудови такої моделі ми переходимо на вищий ієрархічний рівень опису процесу розчинення фарби (від мікроскопічного до мезоскопічного опису). На цьому рівні будується рівняння еволюції концентрації фарби у просторі та часі, при цьому покладається, що рідина є суцільним середовищем.

Подібним чином можна уявити теплову енергію, яка поширюється завдяки випадковим взаємодіям сусідніх частинок. Зазначимо, що поняття суцільних середовищ і неперервних полів є абстрактними. Однак вони знаходять широке застосування у фізиці: у класичній електродинаміці рівняння Максвелла формулюються за допомогою визначення неперервних функцій джерела; тверде тіло часто для простоти трактується як суцільне; різноманітні середовища (рідини, гази, плазма і т.п.) можна спрощено трактувати як суцільні. На основі цих уявлень у просторі й часі визначають неперервні функції, що описують властивості середовища і, застосовуючи кількісні закони фізики, одержують рівняння у частинних похідних, що зв'язують властивості середовища у просторі й часі.

Оскільки нас цікавить саме процес теплопровідності, отримаємо відповідне рівняння дифузії, на базі якого будуватимемо розв'язання поставленої задачі [2]. За законом збереження енергії швидкість зміни енергії в об'ємі V повинна дорівнювати потоку енергії $\overrightarrow{q}$ через поверхню S, яка охоплює даний об'єм:

$$\frac{\partial}{\partial t}\iiint \varepsilon(\overrightarrow{x},t)d\tau = -\iint \overrightarrow{q}\overrightarrow{dS}, \tag{4.1}$$

де $\varepsilon(\overrightarrow{x},t)$ — густина енергії; $d\tau$ — елементарний об'єм. Застосовуючи теорему Остроградського-Гаусса до правої частини рівняння (4.1), одержуємо

$$\iiint \left(\frac{\partial \varepsilon}{\partial t} + \frac{\partial}{\partial \overrightarrow{x}}\overrightarrow{q}\right) d\tau = 0, \tag{4.2}$$

де $\overrightarrow{x} = (x, y, z)$. Нехай густина енергії ε є пропорційною температурі . З експерименту відомо, що потік тепла $\overrightarrow{q}$ залежить від градієнта температури. Тому, вводячи коефіцієнт пропорційності (коефіцієнт теплопровідності) D, одержимо рівняння дифузії

$$\frac{\partial T}{\partial t} - \frac{\partial}{\partial \overrightarrow{x}} D \frac{\partial}{\partial \overrightarrow{x}} T = 0. \tag{4.3}$$

Зрозуміло, що у випадку сталого коефіцієнта дифузії та наявності джерел тепла W_i усереднені об'єму V рівняння (4.3) перепишеться так:

$$\frac{\partial T}{\partial t} - D\frac{\partial^2}{\partial \overrightarrow{x}^2} T = \sum_{i=1}^{N} W_i(\overrightarrow{x}), \tag{4.4}$$

де N — кількість джерел тепла. Розглянемо далі основні методи та помилки, які виникають при чисельному інтегруванні рівнянь типу (4.4).

4.2.2 Огляд методів для параболічних рівнянь

Основні принципи різницевих методів, наведені у попередніх розділах, показують, що функції, неперервні у просторі й часі, можна замінити векторами, компоненти яких визначаються лише у дискретних точках простору і часу [8]- [20]. У випадку звичайних диференціальних рівнянь точність та стійкість розв'язку пов'язана у першу чергу з характерними часовими характеристиками цих рівнянь (наприклад, із часом загасання або часом осциляції). Аналогічно точність і стійкість чисельного розв'язку рівнянь у частинних похідних залежать від характерних часових масштабів процесів, які описуються цими рівняннями. Тому у загальному випадку перед застосуванням різницевого методу до рівнянь у частинних похідних важливо встановити деякі істотні фізичні властивості таких рівнянь.

Явний метод першого порядку точності. Найпростіший шлях розв'язання рівняння дифузії за часом — використати явний метод першого порядку, аналогічний методу Ейлера для звичайних диференціальних рівнянь (розділ 3). Як і раніше, у момент часу $t = 0$ початкові умови визначають залежну змінну на просторовій сітці $\{x_j\}$ (x_j — координата j-го вузла просторової сітки). Розглянемо одновимірний випадок (ланцюжок із N вузлів, розміщених на відстані $\triangle$, в кожному з яких визначена змінна u). Нам необхідно проінтегрувати рівняння

$$\frac{\partial u^n}{\partial t} - D\frac{\partial^2 u}{\partial x^2} = 0 \tag{4.5}$$

у межах кроку за часом Δt. Просторовий оператор $\partial^2/\partial x^2$ — друга похідна за простором, яка за аналогією до другої похідної за часом визначається так:

$$\frac{\partial^2}{\partial x^2} u \approx \frac{u^n_{j+1} - 2u^n_j + u^n_{j-1}}{\triangle^2}, \tag{4.6}$$

де $\triangle$ — крок за простором, $u^n_j = u(t^n, x_j)$. У такому випадку u^{n+1}_j зна-

ходиться з різницевого рівняння

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{\triangle^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \tag{4.7}$$

Зазначимо, що опис методів дослідження стійкості різницевих схем виходить за рамки цього посібника; без доведення наведемо умову стійкості цього методу:

$$\Delta t \leq 0.5\frac{\triangle^2}{D}. \tag{4.8}$$

Отже, щоб одержати стійкий чисельний розв'язок, ми повинні вибирати крок за часом, менший за максимально припустимий крок (4.8). Цей результат можна пояснити фізично. Максимальний припустимий крок за часом Δt_{max} дорівнює часу дифузії на характерній довжині $\triangle$, що збігається з просторовим кроком сітки. Величина Δt_{max} є часом поширення інформації уздовж довжини $\triangle$. Оскільки метод розв'язання є явним, то інформація на $n+1$ часовому кроці в точці j може бути отримана лише з навколишніх точок. Тому в явному методі інформація "поширюється" по сітці лише зі швидкістю $\triangle/\Delta t$, і якщо ця швидкість через великий крок за часом виявляється занадто малою, можна чекати появи некоректних результатів.

Неявний метод Кранка-Нікольсона. Метод Кранка-Нікольсона для параболічних рівнянь аналогічний неявному методу другого порядку для звичайних диференціальних рівнянь (розділ 3). Усереднюючи просторовий дифузійний член за часом, отримуємо неявну схему

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{2\triangle^2}\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right) + \frac{D\Delta t}{2\triangle^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right). \tag{4.9}$$

Метод Кранка - Нікольсона є безумовно стійким. Крім того, він має точність другого порядку як за часовим, так і за просторовим кроком і завдяки цим перевагам широко застосовується. Однак точність і стійкість схеми були отримані ціною ускладнення системи рівнянь для визначення величин u_j^{n+1}. У (4.9) нові значення u_j^{n+1} визначені неявно, що потребує додаткового розв'язання матричного рівняння на кожному кроці за часом.

Нестійкий метод "з переступом". У випадку звичайних диференціальних рівнянь метод "з переступом" має ряд позитивних властивостей. На перший погляд, цей метод можна застосовувати і для розв'язання рівняння дифузії

$$u_j^{n+1} = u_j^{n-1} + 2\frac{D\Delta t}{\triangle^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right). \qquad (4.10)$$

Можна довести[6], що один із коренів для множника переходу[7] завжди менший за (-1), і тому **метод є безумовно нестійким**.

Метод Дюфора - Франкеля

$$u_j^{n+1} = u_j^{n-1} + 2\frac{D\Delta t}{\triangle^2}\left[u_{j+1}^n - \left(u_j^{n+1} + u_j^{n-1}\right) + u_{j-1}^n\right]. \qquad (4.11)$$

Використовуючи нескладні перетворення (4.11), знайдемо явний вираз для функції u_j^{n+1} у кожному вузлі сітки:

$$u_j^{n+1} = \left(\frac{1-\alpha}{1+\alpha}\right)u_j^{n-1} + \frac{\alpha}{1+\alpha}\left(u_{j+1}^n + u_{j-1}^n\right), \qquad (4.12)$$

де

$$\alpha = 2\frac{D\Delta t}{\triangle^2}. \qquad (4.13)$$

Наведена явна схема є стійкою. Зрозуміло, що поданий метод має великі можливості, але необхідно відзначити, що для великих кроків за часом різницева схема призводить до коливань, хоча і незростаючих.

---

[6]Доведення виходить за рамки цього посібника, тому зацікавленому читачу пропонуємо ознайомитись із спеціальною літературою за чисельними методами.

[7]Множник переходу $g = \varepsilon^{n+1}/\varepsilon^n$, де ε^{n+1} та ε^n — відповідно похибка на $n+1$- та n-му часових кроках.

4.3 Розв'язання рівнянь у частинних похідних з використанням MATLAB

4.3.1 Розв'язання методом скінченних різниць

Програма розв'язання рівняння дифузії методом скінченних різниць на MATLAB наводиться у підрозділі 4.5.2[8].

4.3.2 Розв'язання з використанням пакета ToolBox Partial Differential Equations (PDE)

Пакет ToolBox Partial Differential Equations (PDE) призначений для розв'язання задач механіки, теплопровідності, текучості рідини, електростатики та електродинаміки, які зводяться до диференціальних рівнянь у частинних похідних у двовимірних областях складної форми. До складу TOOLBOX входить додаток PDETOOL із графічним інтерфейсом користувача, а також набір функцій, корисних під час написання власних додатків для розв'язання задач методом скінченних елементів. Середовище PDETOOL дозволяє задати геометрію області, тип і коефіцієнти диференціального рівняння, граничні й початкові умови, зробити розбивання області на скінченні елементи (тріангуляцію), розв'язати отриману систему лінійних рівнянь та візуалізувати результат.

Як приклад застосування пакета PDE розглянемо таку задачу: знайти розподіл температури в області, що зображена на рис. 7.1. Границі прямокутника є теплоізольованими, а краї отвору піддаються нагріванню; температура на краях змінюється лінійно у часі. Всередині області немає розподілених джерел тепла; у початковий момент часу температура дорівнює нулю. Температура буде зростати на краях отвору протягом 10 секунд.

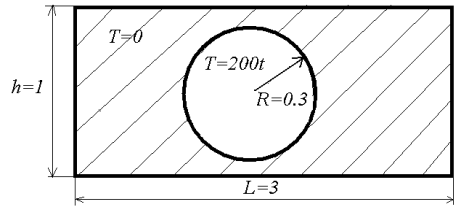

Рисунок 4.1: — Геометрія задачі

[8]Для повноцінного ознайомлення з основами візуального проектування та моделювання динамічних систем пропонуємо ознайомитися з роботами [21]- [28]

Розв'язання поставленої задачі проводиться у декілька етапів:

1-й етап — конструювання області задачі.

За допомогою команди `pdetool` завантажуємо пакет PDE. Конструювання області передбачає створення прямокутника та кола заданих розмірів і вилучення кола з прямокутника. Зручно розмістити область таким чином, щоб центр збігався з початком координат, тоді координати нижнього лівого кута прямокутника будуть $(-1.5, -0.5)$, а його висота і ширина 1 і 3 відповідно. Для створення прямокутника необхідно скористатися командою меню $Draw \rightarrow Rectangle/square$. Утримуючи ліву кнопку мишки, рисуємо мишкою необхідну прямокутну область. Точно задаємо її положення і розміри. Для цього подвійним клацанням мишки викликаємо діалогове вікно **Object Dialog**. У рядках `Left` і `Bottom` задаємо координати нижнього лівого кута прямокутника: `-1.5` і `-0.5` відповідно. У рядках `Width` і `Height` — ширину і висоту прямокутника: `3` і `1` відповідно. `Name` за замовчуванням буде `R1`.

Рисуємо коло з центром у точці $(0, 0)$ і радіусом 0.3 (команда меню $Draw \rightarrow Ellipse/circle(centered)$). Рисування при натиснутій клавіші *Ctrl* дозволяє створити правильне коло або квадрат. У діалоговому вікні **Object Dialog** задаємо положення кола та його радіус. Встановлюємо зв'язок між зображеними примітивами (прямокутником і колом), з яких складається область; коло повинне бути вилучене з прямокутника. Для цього в рядку області введення `Set formula` записуємо формулу `R1 - C1`.

2-й етап — встановлення коефіцієнтів та граничних умов.

Меню *Options* містить підменю *Application*, що дозволяє задати тип розв'язуваної задачі. Пункт `Heat Transfer` відповідає задачі про розподіл тепла. Визначаємо модельне рівняння: за допомогою команди меню $PDE \rightarrow PDESpecification$ викликаємо діалогове вікно **PDE Specification**, у якому потрібно задати коефіцієнти рівняння, що описує поставлену задачу. На панелі `Equation` міститься загальний вигляд рівняння теплопровідності: `rho*C*T'-div(k*grad(t))=Q+h*(Text-T)`. Ліва панель вікна **PDE Specification** дозволяє вибрати тип задачі: `Elliptic` — відповідає задачі про стаціонарний розподіл тепла; `Parabolic` — відповідає нестаціонарному випадку. Для нашої задачі встановлюємо пере-

микач у полі `Parabolic`. Права панель призначена для визначення коефіцієнтів рівняння:

- густина `rho=1`;
- теплоємність `c=1`;
- коефіцієнт теплопровідності `k = 1`;
- потік тепла `Q = 0`;
- коефіцієнт конвективного теплообміну `h = 0`;
- зовнішня температура `Text = 0`.

За допомогою команди меню $Boundary \rightarrow BoundaryMode$ переводимо середовище PDETOOL у режим встановлення граничних умов. У цьому режимі відображаються тільки межі області. Зверніть увагу, що прямокутник має чотири межі, за числом сторін, і коло теж складається з чотирьох дуг. Задаємо граничні умови для верхньої межі прямокутника. Для цього клацанням мишки активізуємо верхню межу прямокутника і за допомогою команди $Boundary \rightarrow SpecifyBoundaryConditions$ викликаємо вікно **Boundary Condition**, призначене для вибору типу граничних умов [9]. Оскільки потік через межу прямокутної області є відсутнім, у вікні **Boundary Condition** встановлюємо перемикач у положення `Neuman` (умова Неймана `n*k*grad(T)+q*T=g`).

Очевидно, що для одержання теплоізольованих меж потрібно задати коефіцієнти `q` і `g` такими, що дорівнюють `0`. Згрупуємо чотири частини кола й у рядку `Text` вводимо формулу `100*t` (умова Діріхле), змінна `t` позначає час. Задаємо початковий розподіл температури та інтервал часу, на якому необхідно знайти наближений розв'язок. Для цього за допомогою команди меню $Solve \rightarrow Parameters$ викликаємо діалогове вікно **Solve Parameters**. У вікні `Time` встановлюємо вектор моментів часу (від `0` до `10` з кроком `0.5`), а у вікні `u(t0)` — початковий розподіл температури (нуль у нашому випадку).

3-й етап — розв'язання і візуалізація результатів

Розв'язання складається з двох кроків:

1) тріангуляції — покриття сконструйованої області сіткою, що складається з трикутників;

[9] Оскільки граничні умови є однаковими, то при натиснутій клавіші *Shift* можна виділити відразу всі чотири сторони прямокутника.

2) розв'язання задачі на цій розрахунковій сітці.

Перехід у режим тріангуляції здійснюється за допомогою команди $Mesh \rightarrow MeshMode$. Область задачі розбивається на досить великі трикутні елементи. Для одержання розв'язку з високою точністю початкової тріангуляції недостатньо, потрібно зменшити крок розбиття області. Це можна зробити за допомогою команди $Mesh \rightarrow RefineMesh$. Кожен вибір цієї команди приводить до рівномірного зменшення розмірів трикутників. Необхідно мати на увазі, що вибір занадто дрібної сітки може призвести до значних витрат часу на розв'язання системи лінійних рівнянь методом скінченних елементів. Повернення до початкової тріангуляції — команда меню $Mesh \rightarrow InitializeMesh$.

Для знаходження розв'язку активізуємо меню $Solve \rightarrow SolvePDE$. У вікні **pdetool** з'являється розподіл температури в області у кінцевий момент часу (`t=10` у нашому випадку) (рис.4.2). За допомогою команди меню $Plot \rightarrow Parameters$ можна викликати діалогове вікно **Plot Selection**, що дозволяє вибрати тип та вигляд результату.

4.3.3 Розв'язання за допомогою пакета SIMULINK

Розглянемо розв'язання одновимірного рівняння теплопровідності за допомогою пакета SIMULINK. Рівняння еволюції у нашому випадку має вигляд

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}. \tag{4.14}$$

Перейдемо від неперервної до дискретної моделі: отримуємо систему звичайних диференціальних рівнянь, кожне з яких описує динаміку зміни величини u в окремих вузлах простору. У матричному вигляді така система рівнянь може бути подана таким чином:

$$\frac{d\mathbf{u}}{dt} = D(\mathbf{Au} + \mathbf{B}), \tag{4.15}$$

де права частина являє собою дискретне наближення до $D(\partial^2 u/\partial x^2)$, (ми заміняємо $(\partial^2 u/\partial x^2)$ у (t, x) на $u(t, x_{j+1}) - 2u(t, x_j) + u(t, x_{j-1})$).

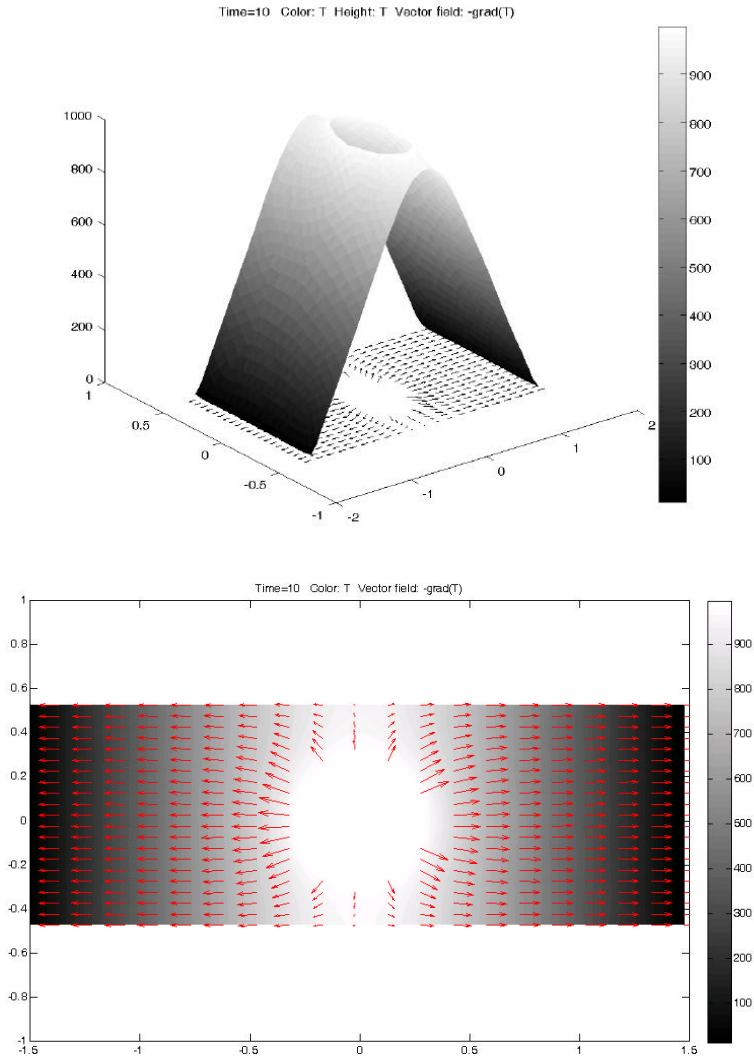

Рисунок 4.2 — Результати моделювання з використанням пакета ToolBox Partial Differential Equations: розподіл тепла у двовимірній пластині з отвором

Матриця $\mathbf{A}$ має вигляд

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix}. \qquad (4.16)$$

Вектор $\mathbf{B}$ являє собою граничні умови. Далі, відповідно до системи рівнянь (4.15), сконструюємо модель (рис. 4.3). Використовуючи меню

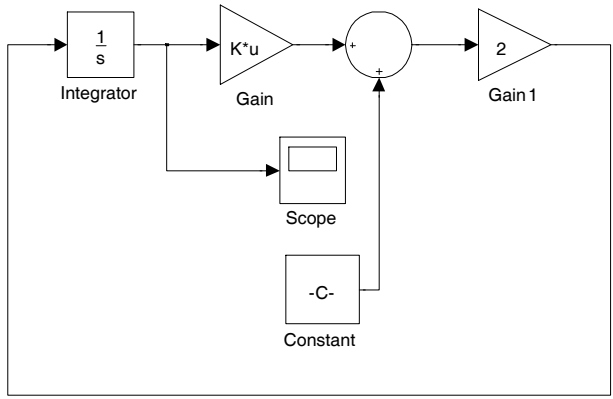

Рисунок 4.3 — Модель рівняння (4.15) у програмі SIMULINK

програми SIMULINK задамо початкові ($D = 4$, $-10 \le x \le 10$, $\triangle x = 1$, $t \in [0, 10]$) та граничні (на кінцях стрижня температура підтримується сталою і дорівнює 10 та 20 відповідно) умови. Початкові умови для $\mathbf{u}$ задаються у блоці INTEGRATOR. У вікні **Block Properties** уводимо рядок: `[10*ones(9,1);15;20*ones(9,1)]` (тобто задаємо початкове значення температури у кожній з 19 внутрішніх точок стрижня (крім граничних точок): $[10, 10, .., 10, 15, 20, 20, .., 20]$). Тут використовується число 15 як компроміс у точці $x = 0$ (середнє значення між 10 та 20). Аналогічним

чином змінюємо властивості блока GAIN: у вікні властивостей уводимо матрицю $\mathbf{A}$:

```
-2*eye(19)+diag(ones(18,1),1)+diag(ones(18,1),-1)
```

та обираємо тип множення — матричне `Matrix(K*u)`. У блоці GAIN1 вводимо константу множення D. Блок CONSTANT являє собою граничні умови (вектор $\mathbf{B}$). Перший елемент вектора $\mathbf{B}$ дорівнює 10, останній 20, елементи між ними дорівнюють 0. Таким чином, у вікні властивостей даного блока вводимо рядок `[10; zeros(17,1);20]`. Далі зберігаємо модель під ім'ям `heat_1d_model.mdl` і запускаємо процес симуляції. Результати моделювання подаються у вікні SCOPE у вигляді 19 кривих, що відображають часову зміну температури кожної з 19 внутрішніх точок стрижня.

Зазначимо, що результати моделювання можна зберегти та використати у робочому просторі системи MATLAB. Запуск моделі із робочого вікна MATLAB виконується командою

```
>> [T,U]=sim('heat_1d_model.mdl',[0,5])
```

Результат моделювання на проміжку $t \in [0, 5]$ — масиви `T` (час) та `U` (матриця із 19 стовпців — температура кожної з точок у кожний момент часу із масиву `T`) стають доступними для використання. До матриці `U` додамо додаткові стовпці значень температури у кінцевих точках:

```
>> u=[10*ones(length(T),1),U,20*ones(length(T),1)]
```

Далі виконуємо команди:

```
>> x=-10:10
```

```
>> surf(x,T,u)
```

Результат моделювання подано на рис. 4.4.

Як бачимо з рисунка, температура точок лівої половини зростає з часом; температура правої половини зменшується. Відповідно через проміжок часу $\triangle t = 5$ температура кожної з точок стрижня виходить на певний стаціонарний рівень, при цьому профіль температури описується лінійною залежністю $T(x)$.

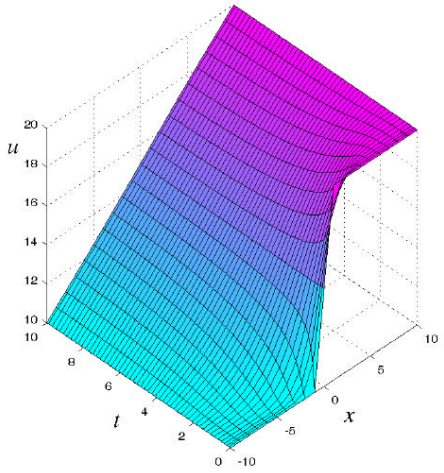

Рисунок 4.4 — Розв'язання рівняння теплопровідності у пакеті SIMULINK

4.3.4 Розв'язання за допомогою команди *pdepe*

Функція *pdepe* застосовується для розв'язання рівнянь у частинних похідних (з однією просторовою координатою) вигляду

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right), \tag{4.17}$$

де c — діагональна матриця; f — потік; s — джерела. Використання параметра m дозволяє розв'язувати одновимірні задачі з лінійною геометрією ($m=0$), розглянути циліндричну (полярну) ($m=1$) або сферичну ($m=2$) систему координат.

Початкові умови визначаються для вектора невідомих:

$$u(x,t_0)=u_0(x). \tag{4.18}$$

Граничні умови задаються у такому вигляді:

$$p(x,t,u)+q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0. \tag{4.19}$$

Як приклад розглянемо розв'язання рівняння

$$\frac{\partial u}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right), \tag{4.20}$$

де $-5 \le x \le 5, t \in [0,4], D = 4$
з початковими

$$u(x,0) = \cos(\pi x/4) \tag{4.21}$$

та граничними

$$u(-5,t) = 0 \tag{4.22}$$

$$\pi e^{-t} + \frac{\partial u}{\partial x}(5,t) = 0 \tag{4.23}$$

умовами.

Наступний M-файл розв'язує подану задачу.

```
function pdepe_my
  m = 0; % геометрія є лінійною
  x = linspace(-5,5,21);
  % розбиваємо інтервал x сіткою із 21 вузла
  t = linspace(0,4,81);
  sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
  % виділяємо перший елемент розв'язку як u.
  u = sol(:,:,1);
  surf(x,t,u)
  % будуємо залежність u(x,t) та підписуємо осі
  xlabel('Distance x')
  ylabel('Time t')
% Функція, що описує рівняння (7.34)
function [c,f,s] = pdex1pde(x,t,u,DuDx)
  c = 1;
  f = 4*DuDx;
  s = 0;
% Функція, що задає початкові умови
function u0 = pdex1ic(x)
  u0 = cos(pi*x/4);
```

```
% Функція, що задає граничні умови згідно зі
% співвідношеннями (7.36-7.37)
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
  % умова Діріхле на лівому кінці
  pl = ul;
  ql = 0;
  % умова Неймана на правому кінці
  pr = pi*exp(-t);
  qr = 1;
```

Результат роботи подано на рис. 4.5.

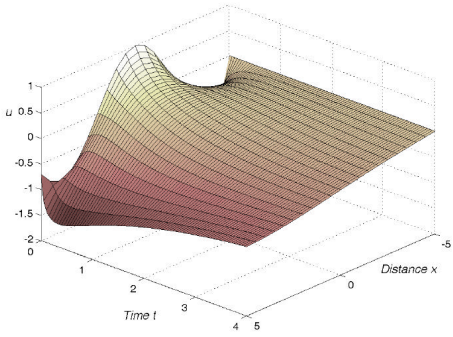

Рисунок 4.5 — Розв'язання рівняння (4.20) за допомогою команди *pdepe*

Як бачимо з рисунка, система досить швидко "забуває" про початковий неоднорідний розподіл температури. Через проміжок часу $\triangle t = 4$ за рахунок дифузії температура вздовж стрижня вирівнюється, виходячи на стаціонарний рівень, що відповідає граничним умовам. У даному випадку температура вздовж стрижня дорівнює нулю.

4.4 Алгоритми

У цьому підрозділі наводиться алгоритм числового розв'язання рівняння дифузії за допомогою явного методу першого порядку точності. Розглянемо двовимірний випадок (для одновимірного випадку опускає-

мо індекс j в усіх рівняннях):

$$\frac{\partial T}{\partial t} - D\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) T = \sum_i^N W_i(x_i, y_i).$$

Алгоритм розв'язання задачі можна описати так:

1 Розбиваємо континуальний простір поля пластини на однакові домени розміром $\triangle \times \triangle$. Вважаємо, що температура вздовж окремого домена є сталою величиною.

2 Задаємо коефіцієнт дифузії D (у неоднорідному випадку — окремо вздовж кожного виміру D_x, D_y), початковий розподіл температури $T_{i,j}(t = 0)$ (де i, j визначають координати окремого домена на гратці), координати і потужності джерел тепла $W_{i,j}$. Припускаємо, що $t = 0$.

3 Запускаємо цикл за t.

3.1 Шляхом послідовного перебору всіх вузлів гратки (окремих доменів) за допомогою різницевої схеми перераховуємо температури доменів на наступному кроці за часом. На цьому етапі створюємо два цикли за i та за j і перераховуємо температуру кожного домена за формулою обраного методу. У випадку явної схеми першого порядку точності розрахункова формула має вигляд

$$T_{i,j}(t + \Delta t) = T_{i,j}(t) + D\left(\frac{T_{i,j+1}(t) - 2T_{i,j}(t) + T_{i,j-1}(t)}{\triangle^2}\Delta t + \right.$$
$$\left. + \frac{T_{i+1,j}(t) - 2T_{i,j}(t) + T_{i-1,j}(t)}{\triangle^2}\Delta t\right) + W_{i,j}\triangle t. \quad (4.24)$$

3.2 Виводимо поточний розподіл температури на екран, зафарбовуючи елементи так, що різним температурам відповідають різні кольори.

3.3 Збільшуємо час на крок Δt. Повертаємося до операції 3.1. Якщо цикл за t закінчився — переходимо до п.4.

4 Вихід із програми.

4.5 Приклади

4.5.1 Моделювання поширення тепла у двовимірній пластині

Розглянемо програмну реалізацію алгоритму моделювання поширення тепла у тонкій двовимірній пластинці, на краях якої підтримується стала температура $T_1 = 10$ та $T_2 = 100$. Початкова температура всіх внутрішніх точок - 0. Ми припускаємо, що додаткових джерел тепла не існує. Для розв'язання рівняння дифузії

$$\frac{\partial T}{\partial t} - D\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)T = 0$$

розіб'ємо континуальний простір поля пластини на однакові домени (подамо у вигляді гратки $M \times N$ з кроком $\triangle = 1$) (рис.4.6) і будемо вважати, що температура вздовж окремого домена є сталою. Для візуалізації

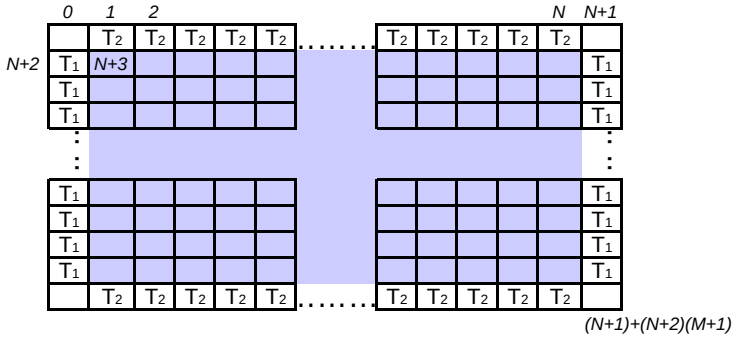

Рисунок 4.6 — Моделювання поширення тепла у двовимірній пластині: континуальний простір розбивається на окремі домени $M \times N$ з кроком $\triangle = 1$

процесу поширення тепла зручним є подання степеня нагрітості окремих ділянок пластини градаціями одного (наприклад, сірого) кольору, при цьому припускаємо, що мінімальній температурі $T = T_1$ відповідає чорний колір, а максимальній T_2 — білий. У поданому в додатку прикладі використовується 16 градацій сірого кольору. Результат роботи

Рисунок 4.7 — Поширення тепла у тонкій пластині, межі якої підтримуються при сталій температурі

програми (наведена у додатку) поданий на рис.4.7, де показана картина розподілу тепла у пластині у момент часу $t = 5$. Візуально простежуючи за характером нагрівання пластини, можна зробити висновок, що даний процес, візуально, абсолютно тотожний процесу розчинення краплі фарби у склянці води. Це пояснюється тим, що, як зазначалося на початку розділу, мезаскопічні рівняння, що описують обидва процеси, абсолютно однакові. Температура плавно "перетікає" від більш нагрітих ділянок до менш нагрітих, тобто у системі відбувається процес дифузії тепла.

4.5.2 Розв'язання рівняння дифузії у програмі MATLAB

Наступна програма на MATLAB (наводиться у додатку) дозволяє розрахувати температурний профіль пластини зі сталою температурою на межах і вивести розподіл температури на екран.

Результат роботи програми поданий на рис. 4.8.

Як і очікувалося, температура у системі поступово вирівнюється з часом, наближаючись до значення $T = 10$ для усіх точок пластини.

Питання для самоконтролю

1 Отримайте рівняння дифузії.

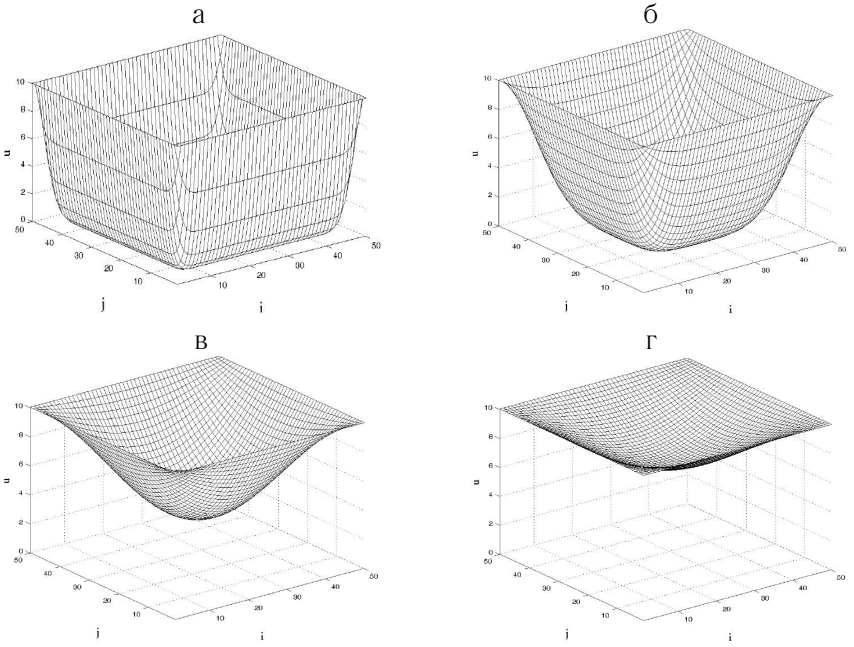

Рисунок 4.8 — Розподіл температури у полі пластини у різні моменти часу: а) $t = 1$; б) $t = 10$; в) $t = 50$; г) $t = 100$

2 Назвіть стійкі методи, які можуть бути застосовані для розв'язання рівняння теплопровідності.

3 Назвіть стійкі методи, які можуть бути застосовані для розв'язання гіперболічних рівнянь.

4 Яким чином формулюється умова стійкості для багатовимірної задачі?

5 Назвіть основні ідеї, які лежать в основі алгоритму моделювання дифузійних процесів.

6 Яким чином можна уникнути врахування граничних умов при моделюванні дифузійних процесів?

Завдання для самостійної роботи

1 Виберіть декілька активних елементів у центрі одновимірного стрижня, температура яких є високою. Побудуйте графік залежності температури від координати і проаналізуйте зміну розподілу температури вздовж стрижня з часом.

2 Розв'яжіть попередню задачу у випадку, коли стрижень є неоднорідним, наприклад, коефіцієнт температуропровідності його лівої половини більший, ніж правої.

3 Задайте джерело тепла, потужність якого періодично змінюється з часом з достатньо малою частотою. Промоделюйте теплові хвилі.

4 Нехай температура групи елементів, що знаходяться у центрі пластини, є досить високою. Проаналізуйте зміну розподілу температури з часом у випадках однорідної та ізотропної пластини.

5 Розв'яжіть попередню задачу для випадку, коли пластина неоднорідна.

6 Промоделюйте нагрівання анізотропної пластини джерелами тепла, розміщеними в центрі.

7 Поблизу центра пластини є група поглиначів тепла (джерел тепла з від'ємною потужністю). Вивчіть зміну розподілу температури з часом.

8 Пластина з отвором містить джерело і поглинач тепла. Проаналізуйте розподіл температури у різні моменти часу.

9 Температура групи елементів поблизу центра пластини підтримується сталою. Проаналізуйте розподіл температури, якщо пластина має джерела тепла з додатною (від'ємною) потужністю.

10 Розв'яжіть попередню задачу у випадку, коли пластина анізотропна, тобто її коефіцієнт температуропровідності залежить від обраного напрямку.

Розділ 5

Моделювання руху системи зв'язаних осциляторів

У розділі розглядається методика моделювання системи зв'язаних осциляторів, які здійснюють одновимірні поперечні або поздовжні коливання.

5.1 Постановка задачі

Промоделюйте рух системи із N осциляторів з масами m_i та жорсткістю пружин k_i, зв'язаних між собою пружинками жорсткістю q_i. Ланцюжок осциляторів розміщений у в'язкому середовищі. Нехай початковий зсув $x_i(t=0)$ та швидкість $v_i(t=0)$ кожного із осциляторів відомі. Розгляньте випадки, коли на окремі осцилятори діє змушувальна сила $f_i(t)$ [29, 30].

5.2 Теоретичний матеріал

5.2.1 Поперечні коливання

Розглянемо ланцюжок осциляторів з масами m_i та коефіцієнтами жорсткості пружин k_i; між окремими осциляторами існують пружні сили з коефіцієнтами q_i (рис. 5.1).

Сила, яка діє на окремий осцилятор, визначається стисканням та розтягуванням зв'язаних з ним пружинок, силою пружності $-k_i x_i$ самого осцилятора та силою в'язкого тертя $-rv_i$ (v_i — швидкість i-го осцилятора). На кожний i-й осцилятор з боку сусідніх $i+1$-го та $i-1$-го осциляторів діє сила

$$F_i = q_i(x_{i-1} - x_i) + q_{i+1}(x_{i+1} - x_i), \tag{5.1}$$

де x_i — зсув i-го осцилятора від положення рівноваги. Отже, згідно з другим законом Ньютона рух окремого осцилятора описується рівнянням

$$m_i a_i = m_i \frac{d^2 x_i}{dt^2} = F_i - rv_i - k_i x_i, \tag{5.2}$$

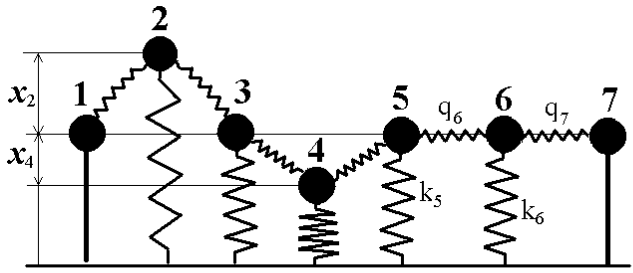

Рисунок 5.1 — Поперечні коливання точкових осциляторів, з'єднаних пружинками. x_2, x_4 — відхилення від положення рівноваги відповідно другого та четвертого осциляторів; осцилятори 1 та 7 є закріпленими

де a_i — прискорення i-го осцилятора. Отже, маємо систему N-рівнянь, яка може бути розв'язана методами, запропонованими у розділі 2.

5.2.2 Поздовжні коливання

Розглянемо коливальний рух лінійного ланцюжка частинок, з'єднаних пружинками з коефіцієнтами пружності q_i (рис. 5.2). Крайові осцилятори є закріпленими:

$$x_0 = x_{N+1} = 0. \tag{5.3}$$

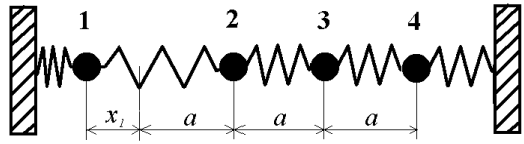

Рисунок 5.2 — Поздовжні коливання точкових осциляторів, з'єднаних пружинками: x_1 — відхилення першого осцилятора від положення рівноваги; a — відстань між осциляторами

Оскільки сила, що діє на i-й ($i = 2..N-1$) осцилятор, визначається лише стисканням та розтягуванням пружинок ліворуч та праворуч і силою тертя, то рівняння руху i-го осцилятора набирає вигляду

$$m_i \frac{d^2 x_i}{dt^2} = q_i(x_{i-1} - x_i) + q_{i+1}(x_{i+1} - x_i) - r v_i. \tag{5.4}$$

Рівняння руху крайових осциляторів мають вигляд

$$m_1 \frac{d^2 x_1}{dt^2} = -q_1 x_1 + q_2(x_2 - x_1) - r v_1, \tag{5.5}$$

$$m_N \frac{d^2 x_N}{dt^2} = -q_{N+1} x_N + q_N(x_{N-1} - x_N) - r v_N. \tag{5.6}$$

5.3 Алгоритми

1 Задаємо параметри системи й початкові умови: число осциляторів N, їхні маси m_i, коефіцієнти пружних зв'язків k_i та q_i, зсув x_i та швидкість v_i частинок у початковий момент часу $t = 0$, крок за часом $\triangle t$, сили f_i, що діють на окремі частинки.

2 Беремо $t = 0$.

3 Цикл за t.

3.1 Перебираємо всі осцилятори і для кожного з них розраховуємо швидкість та зсув у наступний момент часу $t + \triangle t$, наприклад, за модифікованим методом Ейлера:

$$F_i(t) = q_i(x_{i-1}(t) - x_i(t)) + q_{i+1}(x_{i+1}(t) - x_i(t)), \tag{5.7}$$

$$v_i(t + \triangle t) = v_i(t) + \frac{F_i(t) - r v_i(t) - k_i x_i(t)}{m_i} \triangle t, \tag{5.8}$$

$$x_i(t + \triangle t) = x_i(t) + v_i(t + \triangle t) \triangle t, \tag{5.9}$$

3.2 Виводимо результати на екран або до файла.

3.3 Якщо цикл за t закінчився — вихід із циклу.

Однією з основних проблем чисельного розв'язання диференціальних рівнянь (ДР) і систем ДР є проблема вибору кроку інтегрування, оскільки при досить великому кроці інтегрування виникають нестійкі розв'язки, тобто розв'язки, похибка яких починає зростати у часі експоненціально швидко. Один із способів перевірки стійкості методу полягає у контролі величини повної енергії, що у разі вільних коливань повинна зберігатись. Отже для перевірки правильності вибору кроку інтегрування можна використовувати такий алгоритм:

1 Задаємо початкові зсуви і швидкості осциляторів.

2 Задаємо часовий інтервал, на якому шукається розв'язок системи ДУ, та число точок, в яких шукається чисельний розв'язок системи ДУ.

3 Знаходимо розв'язок системи ДУ.

4 Обчислюємо значення енергії системи зв'язаних осциляторів у кожен момент часу.

5 Аналізуємо зміну енергії системи у часі на заданому часовому інтервалі та оцінюємо точність виконання закону збереження енергії.

6 При незадовільній точності розв'язку змінюємо параметри розв'язку (п. 2) (або обираємо інший метод) та повторюємо пп.3-5.

5.4 Приклади

Розглянемо декілька прикладів програм (наведені у додатку). Перша програма дозволяє провести моделювання руху N однакових осциляторів з масами m. Крайні осцилятори є закріпленими. Програма моделює поширення імпульсу вздовж ланцюжка у випадку, коли крайній лівий осцилятор робить півколивання під дією вимушувальної гармонічної сили $f_1 = A\sin(\omega t)$ ($t \in [0...T/2]$, $T = 2\pi/\omega$ — період коливання; A — амплітуда коливань; ω — частота коливань). Через половину періоду лівий осцилятор повертається у вихідне положення й у наступні моменти часу він залишається нерухомим. Програма припиняє свою роботу при натисканні на будь-яку клавішу.

Друга програма дозволяє провести моделювання руху ланцюжка осциляторів, які здійснюють поздовжні коливання уздовж осі x.

Третя програма (мовою MATLAB) дозволяє провести перевірку стійкості обраного чисельного методу шляхом контролю величини повної енергії у випадку вільних коливань.

5.4.1 Візуалізація руху системи осциляторів: поперечні коливання та поздовжні коливання

Результат роботи програми поданий на рис. 5.3. З рисунка бачимо, що середній час поширення фронту хвилі уздовж ланцюжка (від лівого осцилятора до крайнього правого осцилятора) при даних параметрах системи становить $t \approx 0.3$.

5.4.2 Перевірка виконання закону збереження енергії для системи зв'язаних осциляторів

Результат роботи програми поданий на рис. 5.4. Аналіз залежності повної енергії коливальної системи від часу показує, що повна енергія на обраному часовому інтервалі фактично не змінюється (не більше ніж на 1%). Отже, обрана кількість точок (крок інтегрування) є достатньою для проведення експериментів з даною моделлю.

Питання для самоконтролю

1 Побудуйте двовимірну модель руху N осциляторів.

2 Яким чином модифікується модель при переході від поперечних коливань до поздовжніх?

3 Наведіть алгоритм розв'язку системи ДУ, що описує рух ланцюжка осциляторів.

4 Наведіть алгоритм перевірки придатності чисельної схеми для проведення моделювання зазначеної системи.

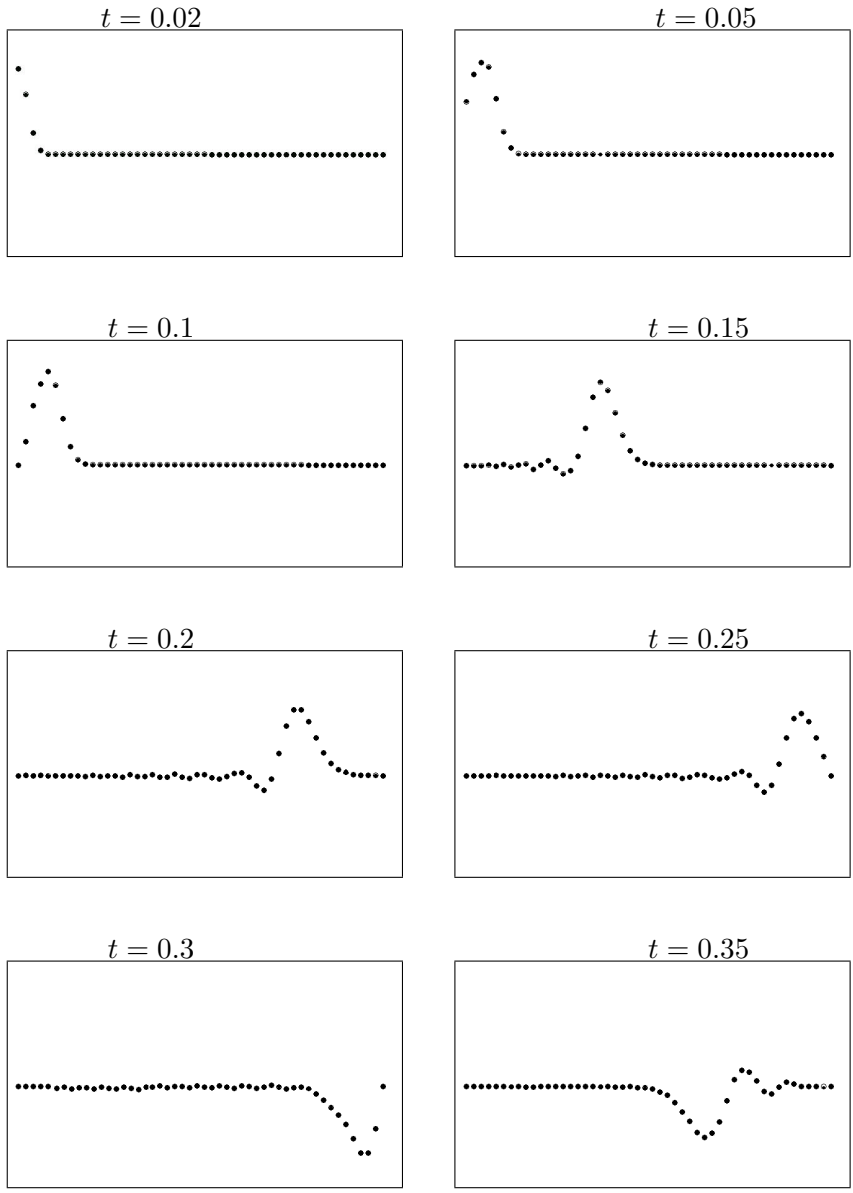

Рисунок 5.3 — Візуалізація процесу поширення хвилі вздовж ланцюжка зв'язаних осциляторів

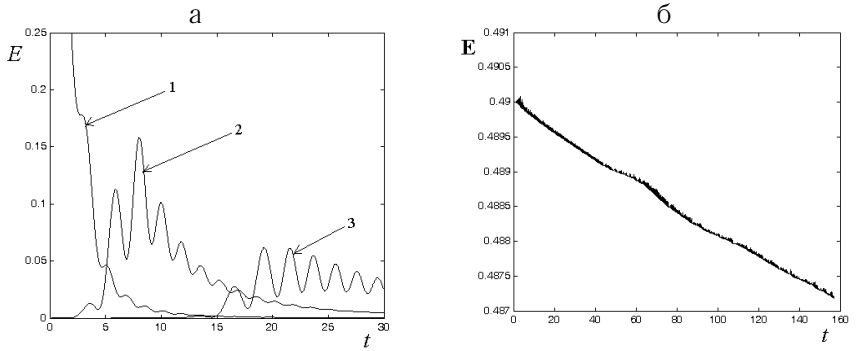

Рисунок 5.4 — (а) Зміна енергії 1-го (крива 1), 5-го (крива 2) та 15-го (крива 3) осциляторів з часом; (б) зміна повної енергії системи з часом

Завдання для самостійної роботи

1 Промоделюйте коливання двох зв'язаних осциляторів. Розгляньте випадки: 1) на один із осциляторів діє зовнішня сила; 2) один з осциляторів має початковий зсув; 3) один з осциляторів має початкову швидкість. Виконайте комп'ютерні експерименти при різних параметрах моделі.

2 Проаналізуйте коливання трьох зв'язаних осциляторів, розглянувши всі перелічені вище випадки та виконавши комп'ютерні експерименти при різних параметрах моделі.

3 Промоделюйте коливання 50 осциляторів, зв'язаних пружинами, у випадку, коли на лівий крайній осцилятор подіяла короткочасна сила. Розгляньте випадки, коли правий крайній осцилятор є закріпленим (а) та незакріпленим (б).

4 Вивчіть зміну фази імпульсу при відбиванні від "більш щільного" та "менш щільного" середовища. Для цього промоделюйте поширення імпульсу вздовж ланцюжка осциляторів у випадку, коли маса однієї половини осциляторів та жорсткість їх пружин відрізняється від аналогічних параметрів іншої половини.

5 Розгляньте рух коливальної системи, що складається із N зв'язаних осциляторів ($N > 20$), вважаючи, що $m_i = 1$, $k_i = 1$, при початкових умовах $x_i(0) = 0$, $\dot{x}_i(0) = 0$ під дією зовнішньої сили $f(t) = A\sin(\Omega t)$, прикладеної до крайнього лівого осцилятора.

- Знайдіть частоти власних коливань ω_i ($i = 1..N$) розглянутої системи зв'язаних осциляторів.
- Вивчіть рух кожного з тіл системи зв'язаних осциляторів при $\Omega = \omega_i$, використовуючи для цього залежності x_i, v_i, E_i та їхні спектри.
- Вивчіть рух коливальної системи у випадках $\Omega < \min(\omega_i)$ та $\Omega > \max(\omega_i)$.
- Вивчіть залежність швидкості поширення збурювання у ланцюжку від частоти зовнішньої сили. Поясніть отриманий результат.
- Вивчіть залежність швидкості поширення збурювання у ланцюжку від маси осциляторів.
- Вивчіть залежність швидкості поширення збурювання у ланцюжку від жорсткості пружин.

У результаті розв'язку наведеної задачі доведіть такі положення:

- розглянута система є фільтром, що не пропускає коливання з частотами $\Omega << \min(\omega_i)$ та $\Omega >> \max(\omega_i)$;
- при частоті зовнішньої сили $\Omega << \min(\omega_i)$ система робить коливання як єдине ціле з частотою, що дорівнює частоті зовнішньої сили;
- при частоті зовнішньої сили $\Omega >> \max(\omega_i)$ амплітуди коливань частинок спадають у напрямку правого кінця ланцюжка.

6 Використовуючи умову попередньої задачі та вважаючи, що система складається з осциляторів різних мас, дайте відповідь на таке запитання: як впливає подібний розподіл мас на поводження системи і швидкість поширення збурювання?

7 Використовуючи умову попередньої задачі та вважаючи, що система складається з різних пружинок, дайте відповідь на запитання: як впливає така конфігурація на поводження системи і швидкість поширення збурювання?

8 Побудуйте залежність, що відображає зміну повної кінетичної енергії системи осциляторів з часом. Виконайте розрахунок кінетичнлї енергії окремих осциляторів. Проаналізуйте отримані залежності.

9 Побудуйте залежність, що відображає зміну повної потенціальної енергії з часом. Виконайте розрахунок потенціальної енергії окремих осциляторів. Проаналізуйте отримані залежності.

10 Порівняйте результати двох попередніх задач для енергії окремих осциляторів та енергії системи в цілому. Покажіть, що результуюча крива становить пряму лінію.

Розділ 6

Моделювання поширення механічної хвилі

Розділ присвячений моделюванню одновимірних механічних хвиль у суцільних пружних середовищах. Моделювання проводиться на основі лінійного хвильового рівняння, яке виводиться у граничному випадку великого числа зв'язаних лінійних осциляторів (розділ 5).

6.1 Постановка задачі

Промоделюйте поширення одновимірної хвилі у суцільному пружному середовищі. Як вихідні дані задачі взято: закони руху (коливань) окремих ділянок середовища і швидкість поширення збурювання. Необхідно розрахувати зсув елементів середовища у наступні моменти часу.

6.2 Теоретичний матеріал

6.2.1 Побудова моделі

У розділі 5 ми розглянули мікроскопічну картину поширення механічної хвилі: коливальний рух ланцюжка із N зв'язаних осциляторів, який у спрощеному випадку може бути описаний системою рівнянь

$$m_i \frac{d^2 x_i}{dt^2} = q_i(x_{i-1} - x_i) + q_{i+1}(x_{i+1} - x_i), \tag{6.1}$$

де x_i — поперечний зсув i-го осцилятора від положення рівноваги; параметр q_i описує взаємодію i-го осцилятора з сусіднім $(i-1)$-м осцилятором. Для спрощення аналізу ми не враховуємо силу в'язкого тертя, яка легко може бути додана до правої частини кожного з рівнянь поданої системи (розділ 5). У випадку однакових осциляторів ($q = q_i$, $m = m_i$, $i = 1..N$) останнє рівняння переписується у вигляді

$$m \frac{d^2 x_i}{dt^2} = q(x_{i+1} - 2x_i + x_{i-1}). \tag{6.2}$$

Перейдемо тепер від коливального руху до хвильового, або, іншими словами, від мікроскопічного до макроскопічного опису процесу поширення хвилі. Використовуючи (6.2), отримаємо неперервне хвильове рівняння, яке буде базовим для моделювання процесів поширення одновимірних хвиль. Для цього розглянемо граничний випадок $N \to \infty$. Замінимо $x_i(t)$, де i — дискретна змінна на функцію $x(y,t)$, де y — неперервна змінна. Тоді рівняння (6.2) перепишеться у вигляді

$$\frac{\partial^2 x(y,t)}{\partial t^2} = \frac{qa^2}{m}\frac{1}{a^2}(x(y+a,t) - 2x(y,t) + x(y-a,t)), \tag{6.3}$$

де a — відстань між окремими осциляторами. Далі, розвиваючи функцію $x(y \pm a, t)$ у ряд Тейлора (до другого порядку включно)

$$x(y \pm a, t) = x(y,t) \pm a\frac{\partial x}{\partial y} + \frac{a^2}{2}\frac{\partial^2 x}{\partial y^2} \tag{6.4}$$

та скорочуючи відповідні складові, перепишемо рівняння (6.3) у вигляді

$$\frac{\partial^2 x(y,t)}{\partial t^2} = \frac{qa^2}{m}\frac{\partial^2 x(y,t)}{\partial y^2}. \tag{6.5}$$

Припускаючи, що

$$\frac{qa^2}{m} = V_f^2, \tag{6.6}$$

де V_f — фазова швидкість хвилі, отримуємо хвильове рівняння

$$\frac{\partial^2 x(y,t)}{\partial t^2} = V_f^2\frac{\partial^2 x(y,t)}{\partial y^2}. \tag{6.7}$$

Останнє рівняння описує процес поширення одновимірної хвилі уздовж осі y (рис. 6.1).

Хвильове рівняння другого порядку (6.7) можна записати у вигляді системи двох рівнянь першого порядку:

$$\frac{\partial v}{\partial t} = V_f\frac{\partial \theta}{\partial y}, \tag{6.8}$$

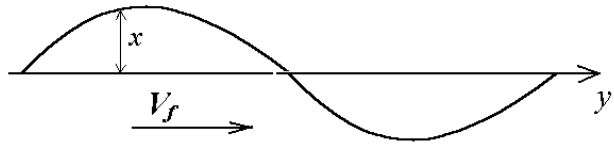

Рисунок 6.1 — Одновимірна хвиля: $x(y,t)$ — відхилення від положення рівноваги; V_f — фазова швидкість

$$\frac{\partial \theta}{\partial t} = V_f \frac{\partial v}{\partial y}, \tag{6.9}$$

де $v = \partial x/\partial t$ — швидкість зсуву; $\theta = \partial x/\partial y$ — кутове відхилення. Перш ніж розглянути методи, які можуть бути використані для розв'язання даної системи рівнянь, зазначимо, що (6.7) можна переписати так:

$$\frac{\partial x}{\partial t} = v, \tag{6.10}$$

$$\frac{\partial v}{\partial t} = V_f^2 \frac{\partial^2 x(y,t)}{\partial y^2} \tag{6.11}$$

і застосувати явну схему інтегрування (розділ 4).

6.2.2 Огляд методів для гіперболічних рівнянь

Явний метод першого порядку точності. Розглянемо найпростіший спосіб інтегрування одновимірного рівняння перенесення

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \tag{6.12}$$

за допомогою явної схеми першого порядку точності. Наближаючи (6.12) у межах кроку за часом Δt, маємо

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\triangle}(u_{j+1}^n - u_{j-1}^n). \tag{6.13}$$

Різницевій схемі (6.13) властива нестійкість для будь-яких часових кроків Δt, і вона показує, що не всі явні та прості схеми є корисними.

Консервативний метод Лакса. Замінюючи u_j^n у схемі (6.13) на просторове середнє, маємо

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{v\Delta t}{2\triangle}(u_{j+1}^n - u_{j-1}^n). \tag{6.14}$$

Запишемо умову застосованості даного методу:

$$\Delta t \leq \frac{\triangle}{|v|}. \tag{6.15}$$

Для цього методу крок за часом потрібно вибирати менше найменшого характерного фізичного часу задачі, що у випадку рівняння перенесення є не що інше, як час, за який швидкість v приводить до перебігу на відстань $\triangle$.

Консервативний метод "з переступом". При інтегруванні гіперболічних рівнянь з першим порядком точності за часом у рівняння вноситься дестабілізувальна складова, оскільки часовий інтервал не є центрованим за часом. Для стабілізації стійкості додають більш сильну просторову складову (метод Лакса). Як результат — сильно згладжений чисельний розв'язок.

Більш якісну апроксимацію дає метод "з переступом".

Допоміжний крок

$$u_j^{n+1} = u_j^{n-1} - \frac{v\Delta t}{\triangle}(u_{j+1}^n - u_{j-1}^n); \tag{6.16}$$

Основний крок

$$u_{j+1}^{n+2} = u_{j+1}^n - \frac{v\Delta t}{\triangle}(u_{j+2}^{n+1} - u_j^{n+1}). \tag{6.17}$$

Зрозуміло, що при застосуванні даного методу необхідний об'єм пам'яті подвоюється у порівнянні з методом першого порядку. Виконуючи аналіз стійкості, можна довести, що метод може застосовуватися лише для рівнянь перенесення і лише при виконанні умови

$$\Delta t \leq \frac{\triangle}{|v|}. \tag{6.18}$$

Двокрокова схема Лакса - Вендрофа. Двокроковий метод Лакса - Вендрофа забезпечує центрування за часом шляхом визначення проміжних значень функцій на напівцілих кроках за часом $t^{n+1/2}$.
Допоміжний крок

$$u_{j+1/2}^{n+1/2} = \frac{1}{2}(u_j^n + u_{j+1}^n) - \frac{v\Delta t}{2\triangle}(u_{j+1}^n - u_j^n); \tag{6.19}$$

Основний крок

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{\triangle}(u_{j+1/2}^{n+1/2} - u_{j-1/2}^{n+1/2}). \tag{6.20}$$

Після кожного основного кроку проміжні значення величин $u_{j+1/2}^{n+1/2}$ стають непотрібними й у подальшому не використовуються.

Умова стійкості методу:

$$\Delta t \leq \frac{\triangle}{|v|}. \tag{6.21}$$

Відзначимо, що у методі Лакса-Вендрофа не вноситься ніяких сторонніх і додаткових чисельних мод. Ця перевага методу визначає його широке застосування.

6.2.3 Багатовимірні явні методи

Умова стійкості $\Delta t \leq \triangle/|v|$, накладена на крок за часом, застосовна до цілого ряду методів й у N-вимірному просторі переходить у таку умову

$$\Delta t \leq \frac{\triangle}{|\overrightarrow{v}|\sqrt{N}}, \tag{6.22}$$

де вектор швидкості $\overrightarrow{v}$ — найбільша швидкість поширення сіткою.

У N - вимірному просторі методу Лакса, наприклад, набирає вигляду

$$\overrightarrow{u}^{n+1} = \frac{1}{2N}\sum_{\alpha=-N}^{N}\overrightarrow{u}_{\alpha}^{n} - \sum_{\alpha=1}^{N}(\overrightarrow{u}_{\alpha}^{n} - \overrightarrow{u}_{-\alpha}^{n})\frac{\overrightarrow{v}\Delta t}{2\triangle}. \tag{6.23}$$

Індекси, що визначають конкретну точку сітки, опущені, й рівняння необхідно розглядати у кожному вузлі гратки; індекс α належить до $2N$ сусіднім точкам у N-вимірному просторі.

У поширеній двовимірній постановці схема набирає вигляду

$$u_{i,j}^{n+1} = \frac{1}{4}(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n) - \frac{\Delta t}{2\triangle} v_x (u_{i+1,j}^n - u_{i-1,j}^n) - \\ - \frac{\Delta t}{2\triangle} v_y (u_{i,j+1}^n - u_{i,j-1}^n). \tag{6.24}$$

Умова стійкості

$$\Delta t \le \frac{\triangle}{(v_x^2 + v_y^2)^{1/2}\sqrt{2}}. \tag{6.25}$$

6.3 Алгоритми

1 Дискретизуємо модель: розбиваємо неперервне пружне середовище (вісь y в одновимірному випадку), у якому поширюється хвиля, граткою із N вузлів (в одновимірному випадку), відстань між якими (крок за граткою) $\triangle y$. Створюємо два масиви `X[N]` та `V[N]`, в яких будемо зберігати відповідно зсув та швидкість точок середовища у N точках гратки.

2 Задаємо параметри моделі: крок за часом $\triangle t$, швидкість поширення хвилі V_f, початкові координати та швидкості кожної із N точок середовища ($x_i(0)$, $y_i(0)$, $i = 1..N$), а також рівняння коливань окремих точок середовища, наприклад рівняння коливань крайньої лівої точки $x_1 = A\sin(\omega t)$. Припускаємо, що $t = 0$.

3 Цикл за t.

3.1 Збільшуємо час на $\triangle t$: $t = t + \triangle t$.

3.2 Цикл за i.

3.2.1 Розраховуємо швидкість i-ї точки середовища у даний момент часу згідно з методом Ейлера:

$$v_i(t+\triangle t) = v_i(t) + V_f^2 \frac{x_{i+1}(t) - 2x_i(t) + x_{i-1}(t)}{(\triangle y)^2} \triangle t. \qquad (6.26)$$

3.2.2 Записуємо розраховану швидкість i-ї точки у масив `V`.

3.2.3 Збільшуємо i на 1 та повертаємося до пункту 3.2.1. Якщо цикл за i закінчився, беремо $i = 1$ та переходимо до пункту 3.3.

3.3 Цикл за i.

3.3.1 Розраховуємо координату i-ї точки середовища у даний момент часу згідно з методом Ейлера:

$$x_i(t+\triangle t) = x_i(t) + v_i(t+\triangle t)\triangle t. \qquad (6.27)$$

3.3.2 Якщо відомі закони коливання окремих точок середовища, розраховуємо їхній зсув у даний момент часу згідно із заданими законами.

3.3.3 Записуємо розраховану координату i-ї точки у масив `X`.

3.3.4 Збільшуємо i на 1 та повертаємося до пункту 3.3.1. Якщо цикл за i закінчився, переходимо до пункту 3.4.

3.4 Записуємо отримані дані (масив `X` та `V`) у файл або виводимо на екран. В останньому випадку стираємо попередній рисунок та накреслюємо новий, з'єднуючі окремі елементи масиву `X` лініями.

3.5 . Повернення до пункту 3.1. Якщо цикл за t закінчився — виходимо із циклу.

6.4 Приклади

Подана програма (наводиться у додатку) дозволяє провести моделювання поширення одновимірної хвилі у пружному середовищі. Початкові координати та швидкості елементів середовища ми беремо такими, що дорівнюють нулю. Крайня ліва точка виконує два повних коливання згідно із законом $A\sin(\omega t)$ та у подальшому є нерухомою. Правий кінець є закріпленим. Початкові дані: амплітуда коливань `A=4`, частота `omega=1`, швидкість поширення хвилі `V_f=2`, крок за часом `dt=0.01`,

крок за простором `dy=1`. Програма припиняє свою роботу при натисканні на будь-яку клавішу. Результат роботи програми поданий на рис. 6.2. З рисунка бачимо, що середній час поширення фронту хвилі (при даних параметрах системи) становить $t \approx 19$.

Питання для самоконтролю

1 На основі яких припущень будується модель хвилі?

2 Які різницеві схеми можуть бути застосовані при моделюванні хвильових процесів?

3 Який параметр моделі хвилі визначає густину середовища, в якому вона поширюється?

4 Отримайте модель двовимірної хвилі.

Завдання для самостійної роботи

1 Модифікуючи наведену вище програму, промоделюйте проходження і відбиття хвилі від середовища з більшою густиною. Для цього розділіть середовище на дві частини і припустіть, що швидкість поширення хвилі в одній із частин системи (яка вважається більш густою) є меншою.

2 Вивчіть поширення і відбиття хвилі (одиночного імпульсу або цугу) від закріпленого і незакріпленого кінця пружного середовища.

3 Промоделюйте інтерференцію хвиль, яка виникає в результаті відбиття падаючої хвилі або випромінювання двох когерентних хвиль.

4 Проаналізуйте залежності довжини хвилі від частоти та швидкості поширення.

5 Промоделюйте поширення хвилі і її відбиття від закріпленого (або незакріпленого) кінця середовища у випадку, коли його інший крайовий елемент здійснює гармонічні коливання.

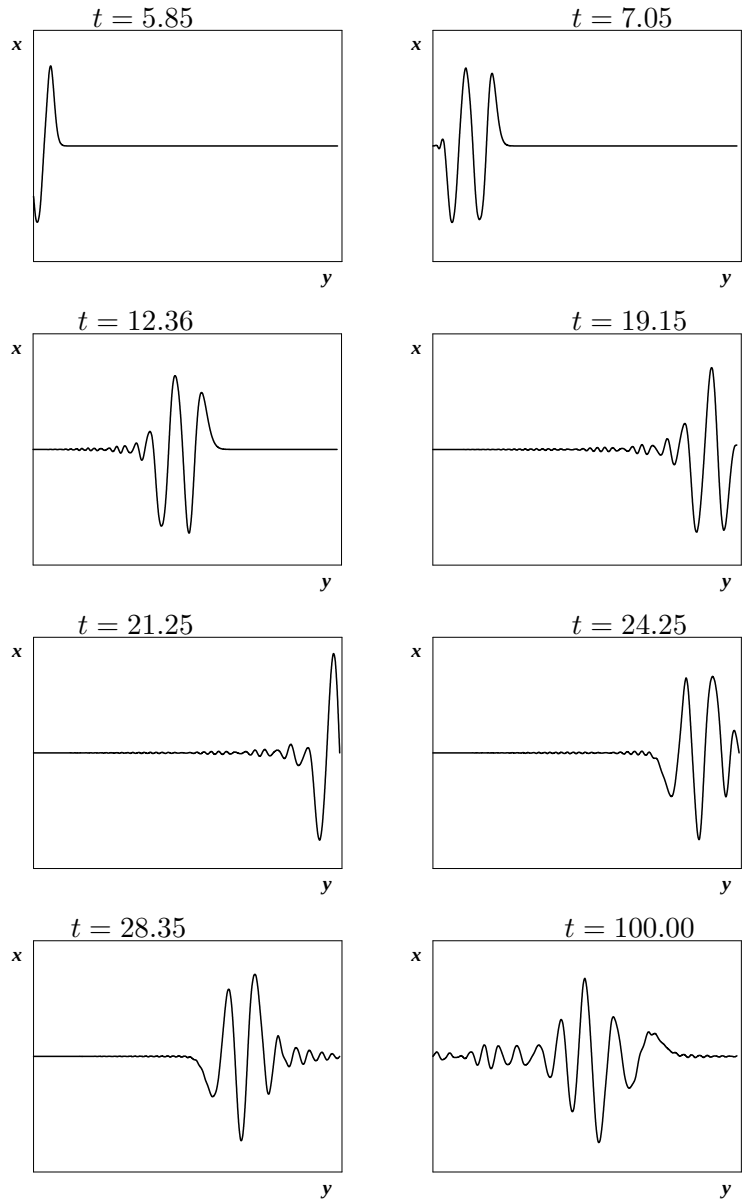

Рисунок 6.2 — Візуалізація процесу поширення хвилі у пружному середовищі. Джерело хвиль здійснює два повних коливання. Правий кінець є закріпленим

6 Вивчіть поширення і відбиття хвилі у випадку, коли лівий елемент середовища здійснює півколивання.

7 Простежте за суперпозицією хвиль, що випромінюються двома, віддаленими один від одного на певну відстань, елементами, які коливаються з однаковими (різними) частотами.

8 Промоделюйте виникнення стоячої хвилі при відбитті гармонічної хвилі від правого закріпленого (незакріпленого) кінця шнура.

9 Промоделюйте інтерференцію двох хвиль, які поширюються назустріч одна одній.

10 Використовуючи наведену модель, експериментально отримайте залежність довжини хвилі від частоти.

Розділ 7

Моделювання хаотичних динамічних систем

У розділі розглядаються нелінійні динамічні системи, поведінка яких є випадковою, незважаючи на те, що вона визначається детерміністичними законами. На простих прикладах наводяться основні підходи до аналізу та моделювання таких систем, більш відомих під назвою "хаотичні системи".

7.1 Постановка задачі

На прикладі логістичного відображення продемонструйте ефект чутливості хаотичної системи до малих збурень початкових умов, визначте показник Ляпунова та оцініть можливий час прогнозування поведінки хаотичної системи. Проведіть моделювання багатопараметричної системи Лоренца та визначте параметри, при яких система демонструє хаотичну поведінку.

7.2 Теоретичний матеріал

Динамічний (або детермінований) хаос – нерегулярна непередбачувана поведінка виключно нелінійної динамічної системи, яка виникає за відсутності впливу жодного випадкового чинника. Існування хаосу пов'язане з чутливістю до малих збурень початкових умов системи. Така невизначеність у початкових умовах може бути обумовлена, наприклад, кінцевою точністю вимірювальних приладів.

Критерієм хаосу є наявність додатного старшого показника Ляпунова, розрахунок якого може бути виконано як аналітично (наприклад, для простих рекурентних відображень), так і чисельно (для більш реалістичних моделей динамічних систем) [31]. Розглянемо обидва випадки більш детально.

7.2.1 Опис хаотичних систем за допомогою відображень

Нехай $\delta x(0)$ – нескінченно мала відстань між двома точками у фазовому просторі, які належать різним фазовим траєкторіям у момент часу $t = 0$, $\delta x(t)$ – відстань між цими точками у момент часу t. Тоді можна записати

$$|\delta x(t)| \approx |\delta x(0)| \exp(\lambda t), \tag{7.1}$$

де параметр λ називається показником Ляпунова. Якщо $\lambda > 0$, то дві фазові траєкторії, які виходять із малого околу певної точки простору (початкові координати зсунуті на незначну відстань), з часом розходяться експоненціально швидко (рис. 7.1(а)). Із співвідношення (7.1) можна отримати формулу для розрахунку показника Ляпунова:

$$\lambda = \lim_{\substack{t\to\infty \\ \delta x(0)\to 0}} \frac{1}{t} \ln \left| \frac{\delta x(t)}{\delta x(0)} \right|. \tag{7.2}$$

У загальному випадку він є функцією початкової координати.

Фазові траєкторії динамічної системи, як правило, не можуть розходитися до нескінченності внаслідок фізичної обмеженості руху. Тому через деякий час вони наближаються до межі фазового об'єму, яку не можуть перетнути. Це призводить до перемішування фазових траєкторій (рис. 7.1б) – властивість, що притаманна статистичним системам. Таким чином, чутливість динамічної системи до малих збурень початкових умов та обмеженість її фазового об'єму обумовлюють виникнення динамічного хаосу. Відповідно прогноз поведінки такої системи на тривалий час стає неможливим.

Розгляд хаотичних динамічних систем зручно почати з простих прикладів – одновимірних дискретних відображень, які мають вигляд

$$x_{n+1} = f(x_n), \tag{7.3}$$

де x – динамічна змінна; $f(x)$ – певна однозначна нелінійна функція; $n = 0, 1, 2, \ldots$.

Отримаємо загальний вигляд формули для розрахунку показника Ляпунова у випадку, коли динамічна система описується одновимірним відображенням вигляду (7.3). Для цього у виразі (7.2) час t замінимо на

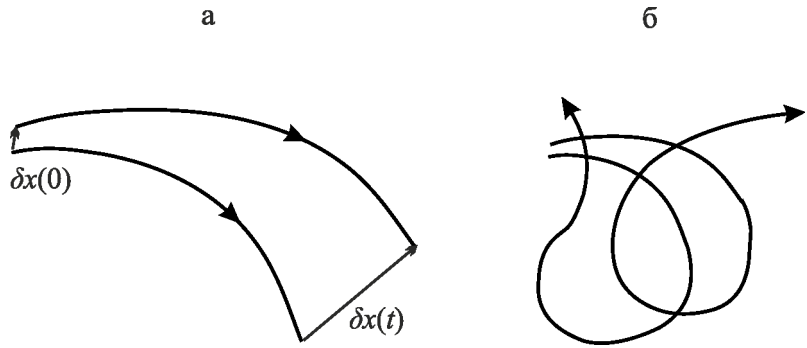

Рисунок 7.1 — Розбіжність фазових траєкторій, які виходять із близьких точок фазового простору (а) та їх перемішування при досягненні межі фазового об'єму (б)

номер ітерації N, відстань δx_0 між початковими точками траєкторій позначимо як ε, відстань δx_N між точками після N ітерацій запишемо як $|f_N(x_0+\varepsilon) - f_N(x_0)|$. У результаті отримаємо

$$\lambda = \lim_{\substack{N\to\infty \\ \varepsilon\to 0}} \frac{1}{N} \ln \left| \frac{\delta x_N}{\varepsilon} \right| = \lim_{\substack{N\to\infty \\ \varepsilon\to 0}} \frac{1}{N} \ln \left| \frac{f_N(x_0+\varepsilon) - f_N(x_0)}{\varepsilon} \right|.$$

Виконавши в цьому виразі граничний перехід $\varepsilon \to 0$, одержимо

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \ln \left| \frac{df_N(x_0)}{dx_0} \right|. \tag{7.4}$$

Для подальшого перетворення виразу (7.4) використаємо формулу похідної від складної функції. Наприклад, при $N = 2$ можна записати

$$\left.\frac{df_2(x)}{dx}\right|_{x_0} = \left.\frac{d}{dx} f[f(x)]\right|_{x_0} = f'[f(x_0)]f'(x_0) = f'(x_1)f'(x_0),$$

де $x_1 = f(x_0)$. А для довільного N вираз (7.4) набирає вигляду

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \ln \left| \frac{df^N(x_0)}{dx_0} \right| = \lim_{N\to\infty} \frac{1}{N} \ln \left| \prod_{i=0}^{N-1} f'(x_i) \right|.$$

Таким чином, остаточний вигляд формули для розрахунку показника Ляпунова у випадку одновимірного відображення, поданого у вигляді (7.3), є таким

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln \left| f'(x_i) \right|. \tag{7.5}$$

Відображення "зуб пили". Як перший приклад розглянемо модельну динамічну систему, яка описується таким рівнянням

$$x_{n+1} = \{2x_n\}, \tag{7.6}$$

де фігурні дужки позначають дробову частину числа, $x_0 \in [0, 1]$. Нескладно перевірити, що усі наступні значення x_n також належать одиничному інтервалу. У зв'язку з цим рівняння (7.6) називають відображенням одиничного інтервалу на самого себе. Розв'язок відображення "зуб пили" має вигляд

$$x_n = \{2^n x_0\}. \tag{7.7}$$

Рівняння (7.6) можна записати у вигляді (див. рис. 7.2)

$$x_{n+1} = \begin{cases} 2x_n, & 0 \le x < 1/2, \\ 2x_n - 1, & 1/2 \le x < 1, \\ 0, & x = 1. \end{cases} \tag{7.8}$$

Виберемо як початкову умову $x_0 = 1/8$. У такому випадку траєкторія динамічної системи (7.6) задається послідовністю чисел

$$\frac{1}{8}, \frac{2}{8}, \frac{4}{8}, 0, 0, \dots ,$$

яка збігається до 0. Якщо початкова умова, наприклад, $x_0 = 1/9$, тоді ми отримаємо періодичну траєкторію (рис. 7.3):

$$\frac{1}{9}, \frac{2}{9}, \frac{4}{9}, \frac{8}{9}, \frac{7}{9}, \frac{5}{9}, \frac{1}{9}, \dots .$$

Для подальшого дослідження властивостей послідовності чисел, що утворюється відображенням (7.6), перейдемо до їх двійкового подання.

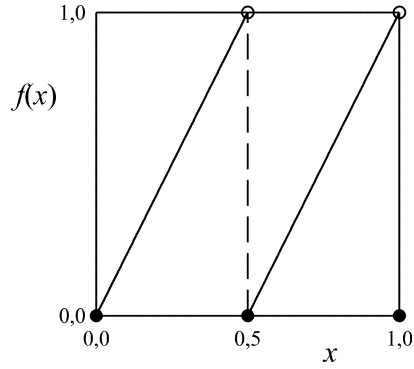

Рисунок 7.2 — Графік відображення "зуб пили"

Початкова умова у двійковому поданні має вигляд

$$x_0 = \sum_{k=1}^{\infty} a_k 2^{-k} = 0, a_1 a_2 a_3 \ldots ,$$

де цифри a_k набувають значення або 0, або 1, тобто $a_k \in \{0, 1\}$. Якщо початкове значення x_0 знаходиться у лівій половині одиничного інтервалу ($x_0 < 1/2$), тоді перший розряд після коми у двійковому поданні числа x_0 дорівнює нулю ($a_1 = 0$); якщо $x_0 \geq 1/2$, тоді $a_1 = 1$. Дія перетворення $f(x) = \{2x\}$ на двійкове число x_0 зводиться до переміщення коми праворуч на один розряд і видалення цілої частини. Таке перетворення двійкової послідовності називається зміщенням Бернуллі. Отже, на першій ітерації отримаємо

$$x_1 = f(x_0) = 0, a_2 a_3 a_4 \ldots ,$$

на n-й ітерації

$$x_n = 0, a_{n+1} a_{n+2} a_{n+3} \ldots$$

Число $1/8$ у двійковому поданні – скінченний дріб 0,001. При початковій умові $x_0 = 1/8$ після третьої ітерації ми отримаємо 0. Число $1/9$ у

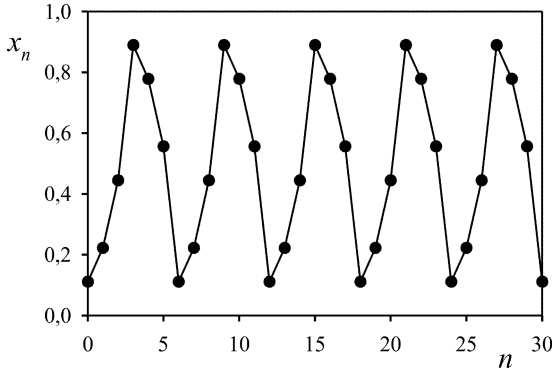

Рисунок 7.3 — Графік періодичної траєкторії для відображення «зуб пили» при $x_0 = 1/9$. Точки з'єднані суцільною лінією для більш наочного подання

двійковому поданні — періодичний дріб $0{,}(000111)$. При початковій умові $x_0 = 1/9$ після шостої ітерації послідовність значень відображення повторюється. Отже, якщо початкове значення x_0 — раціональне число з інтервалу $[0, 1]$, тоді x_n — або періодична послідовність, якщо x_0 — нескінчений двійковий дріб, або збігається до 0, якщо x_0 — скінченний двійковий дріб.

Ірраціональні числа з інтервалу $[0, 1]$ подають у вигляді неперіодичних двійкових дробів. Якщо початкове значення — ірраціональне число, тоді динамічна система (7.6) має неперіодичну траєкторію. Ми можемо задати початкову умову довільною послідовністю нулів та одиниць у двійковому поданні. Випадкову послідовність цифр отримаємо шляхом підкидання монети: якщо випадає орел, записуємо 0, решка — 1. Наприклад, при початковій умові $x_0 = 0{,}1010000\ldots$ динамічна система перебуває під час своєї еволюції в лівій та в правій половині одиничного інтервалу відповідно до випадкової послідовності, тобто вона має статистичні властивості, як і при підкиданні монети.

Ірраціональні числа у вигляді двійкових дробів задаються з кінцевою точністю. При цьому, якщо дві початкові умови починають різнитися з $n + 1$ двійкового розряду після коми, то ця різниця з кожною ітераці-

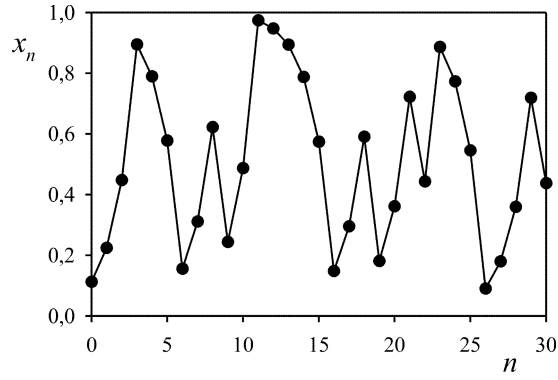

Рисунок 7.4 — Графік хаотичної траєкторії для відображення «зуб пили» при $x_0 = 1/\sqrt{80}$. Точки з'єднані суцільною лінією для більшої наочності

єю збільшується. Після n ітерацій значення будуть відрізнятися вже в першому розряді після коми, тобто динамічна система (7.6) повністю забуває свій початковий стан, що притаманно хаотичній поведінці. Таким чином, якщо x_0 – ірраціональне число, то траєкторія динамічної системи (7.6) – хаотична, наприклад, як на рис. 7.4 при $x_0 = 1/\sqrt{80}$.

Знайдемо показник Ляпунова для відображення "зуб пили" за формулою (7.5). Оскільки похідна $f'(x)$ дорівнює 2, маємо $\lambda = \ln 2 \approx 0,693$. Отже, показник Ляпунова є більшим за 0, що свідчить про експоненціальну розбіжність фазових траєкторій динамічної системи.

Зазначимо, що експоненціальна розбіжність фазових траєкторій притаманна і динамічній системі, що описується рівнянням $x_{n+1} = 2x_n$. Проте хаос в ній не спостерігається. Це пояснюється тим, що в цій системі, на відміну від системи (7.6), рух є необмеженим, унаслідок чого перемішування траєкторій стає неможливим.

Логістичне відображення. Як другий приклад розглянемо логістичне відображення, яке описується таким рівнянням:

$$x_{n+1} = 4x_n(1 - x_n), \tag{7.9}$$

Початкове значення x_0 належить одиничному інтервалу, тому ко-

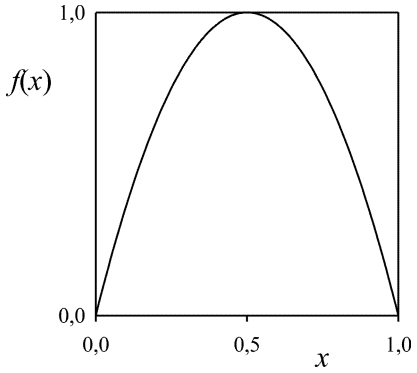

Рисунок 7.5 — Графік логістичного відображення

жне наступне значення x_n також належить одиничному інтервалу. Графік функції $f(x) = 4x(1 - x)$ подано на рис. 7.5.

Логістичне відображення (7.9) може описувати, наприклад, динаміку чисельності біологічної популяції в замкненому середовищі. При цьому x_n – нормована на одиничний інтервал чисельність особин в n-й рік. Кількість особин x_{n+1} у наступний рік пропорційна, з одного боку, кількості особин x_n у попередній рік, а з іншого – вільній частині життєвого простору, яка дорівнює $(1 - x_n)$. Отже, $x_{n+1} \approx x_n(1 - x_n)$.

Покажемо, що логістичне відображення (7.9) можна звести до відображення "зуб пили" (7.6). Для цього, виконуючи заміну змінної

$$x_n = (y_n + 1)/2,$$

отримаємо квадратичне відображення

$$y_{n+1} = 1 - 2y_n^2, \tag{7.10}$$

яке еквівалентне логістичному відображенню. При цьому значення y_n належать інтервалу $[-1, 1]$. Наступне перетворення координати зробимо за формулою $y_n = -\cos 2\pi z_n$. Враховуючи, що функція $\cos 2\pi z_n$ – періодична з періодом 1, область значень для z_n можна обмежити, $z_n \in [0, 1]$. Підставимо останній вираз для y_n в (7.10) та застосуємо три-

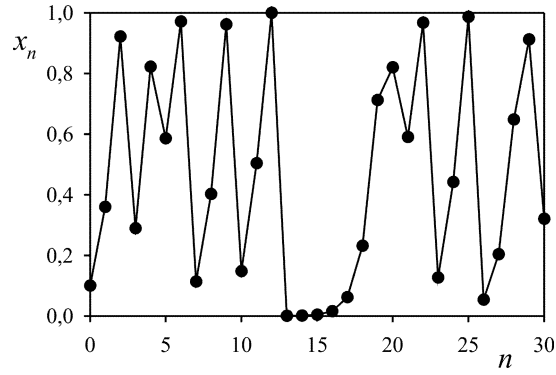

Рисунок 7.6 — Графік хаотичної траєкторії для логістичного відображення при $x_0 = 0,1$. Точки з'єднані суцільною лінією для більшої наочності

гонометричне перетворення $2\cos^2\varphi - 1 = \cos 2\varphi$. У результаті отримаємо рівняння

$$\cos 2\pi z_{n+1} = \cos 4\pi z_n,$$

розв'язок якого можна подати у вигляді $z_{n+1} = 2z_n + k$ (k – ціле число), або, враховуючи обмеженість для z_n, у вигляді

$$z_{n+1} = \{2z_n\}. \tag{7.11}$$

Таким чином, логістичне відображення (7.9) зводиться до відомого відображенням "зуб пили"(7.11), розв'язок якого подається виразом (7.7). Якщо початкове значення x_0 таке, що

$$z_0 = (1/2\pi)\arccos(1 - 2x_0)$$

– ірраціональне число, тоді логістичне відображення демонструє хаотичну поведінку (рис. 7.6).

Ураховуючи виконані перетворення змінних, точний розв'язок рівняння (7.9) можна записати у вигляді

$$x_n = \frac{1}{2} - \frac{1}{2}\cos\left[2^n \arccos(1 - 2x_0)\right]. \tag{7.12}$$

Показник Ляпунова для логістичного відображення можна отримати чисельно (див. програму у додатку), використовуючи формулу (7.5). Враховуючи, що похідна $f'(x) = 4 - 8x$ для $N = 100$ ітерацій маємо $\lambda \approx 0,693$. Отже, показник Ляпунова є додатним, що свідчить про експоненціальну розбіжність фазових траєкторій.

На прикладі логістичного відображення продемонструємо чутливість даної системи до малих збурень початкових умов, притаманну всім динамічним хаотичним системам. Для цього порівняємо дві траєкторії з $x_0 = 0,1$ та $x_0 = 0,10001$ (рис. 7.7). Відносна похибка у визначенні початкових умов становить $0,01\%$. Як бачимо з рисунка, починаючи з 10-ї ітерації значення розходяться. Згодом абсолютна похибка Δx_n може бути порівнянною з розміром області, $\Delta x_n \sim 1$.

Динамічна система, яка чутлива до малих збурень початкових умов, також буде чутливою до малих збурень на кожній ітерації. При чисельному моделюванні малими збуреннями є похибки округлення. Якщо мантиса двійкового числа дорівнює N, тоді для хаотичної траєкторії початкові умови повністю забуваються приблизно через N ітерацій. Проілюструємо це твердження на прикладі логістичного відображення, використавши при цьому точний розв'язок (7.12). Для зберігання дійсних чи-

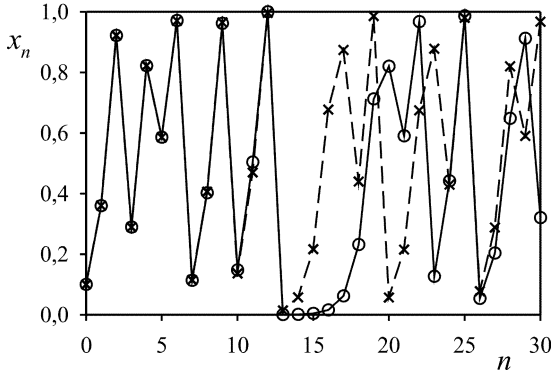

Рисунок 7.7 — Графіки хаотичних траєкторій для логістичного відображення при $x_0 = 0,1$ (кола, суцільна лінія) та $x_0 = 0,10001$ (хрестики, пунктирна лінія))

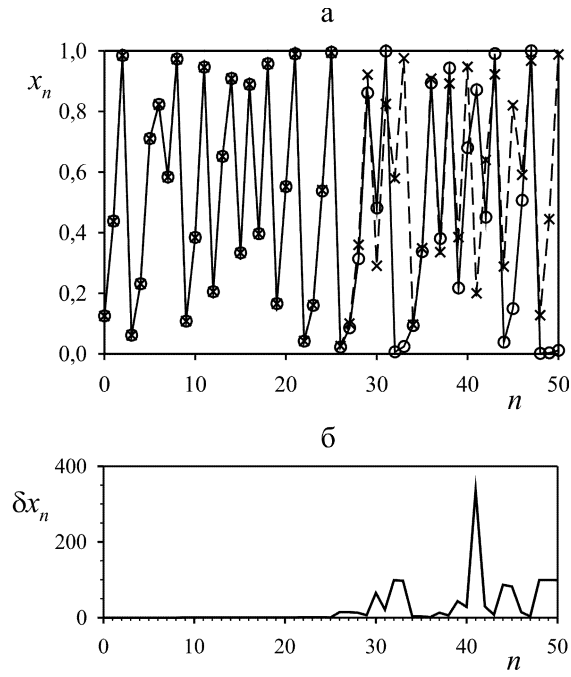

Рисунок 7.8 — Порівняння наближених (кола, суцільна лінія) і точних (хрестики, пунктирна лінія) значень логістичного відображення при $x_0 = 1/8$ (а) та відносна похибка (б)

сел використаємо тип float (23 біти для зберігання мантиси). Критерієм розбіжності точних значень $\tilde{x}_n$ з наближеними x_n є відносна похибка $\delta x_n = \Delta x_n / \tilde{x}_n \times 100\%$, де $\Delta x_n = |x_n - \tilde{x}_n|$ – абсолютна похибка. Результати порівняння наведені на рис. 7.8. Початкове значення $x_0 = 1/8$ у двійковому поданні – скінченний дріб, тому воно задається абсолютно точно, і розбіжності, що виникають, пов'язані виключно з похибками округлення. Як бачимо з рисунка, чисельний прогноз поведінки хаотичної системи можливий лише для перших $N = 24$ ітерацій[10].

[10] кількість розрядів мантиси числа.

7.2.2 Детерміністичний хаос у реалістичних моделях

У попередньому підрозділі були розглянуті приклади динамічних систем, що мають штучний характер. У зв'язку з цим постають питання, чи може виникати хаос у більш реалістичних моделях, опис яких базується на використанні диференціальних рівнянь, і якщо це так, які підходи можуть бути застосовані при комп'ютерному моделюванні та аналізі таких систем. Відповідь на ці питання дамо на прикладі відомої моделі Лоренца [37, 38].

Система Лоренца. Система Лоренца складається із трьох диференціальних рівнянь першого порядку, що описують конвекцію рідини, яка підігрівається знизу. У 1961 році при чисельному розв'язанні побудованої системи рівнянь Едвард Лоренц виявив установлення хаотичного режиму, який характеризувався складною неперіодичною поведінкою змінних системи у часі. Пізніше його модель була застосована для опису роботи лазера, дисипативного осцилятора з інерційним збудженням тощо, спектр динамічних режимів яких містить хаотичну складову.

У загальному випадку модель Лоренца має вигляд

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x), \\ \frac{dy}{dt} &= rx - y - xz, \\ \frac{dz}{dt} &= -bz + xy, \end{aligned} \tag{7.13}$$

де σ, r, b — константи. Змінна x характеризує швидкість обертання конвекційних валів, величини y та z відповідають за розподіл температури по горизонталі та вертикалі. Параметр b визначається геометрією конвекційної комірки; параметр σ є пропорційним числу Прандтля; параметр r пов'язаний із числом Релея і відіграє роль керуючого параметра. Збільшення r до певного порогового рівня супроводжується виникненням конвекційних валів; при великих значеннях керуючого параметра система демонструє хаотичну поведінку.

Для чисельного аналізу моделі (7.13) застосуємо адаптивний метод Рунге-Кутта 3-го порядку з довільним вибором кроку за часом та оцін-

кою похибки на кожному кроці. Динамічний вибір кроку за часом є корисним у тому разі, коли траєкторія системи випробовує різкі зміни, що є характерною рисою хаотичних систем.

Метод Рунге-Кутта. Як правило, математична модель хаотичної системи (наприклад, система Лоренца) складається із декількох диференціальних рівнянь першого порядку:

$$\frac{d\overrightarrow{x}}{dt} = f(\overrightarrow{x}, t),$$

де $\overrightarrow{x} = x, y, z....$

Для розв'язання даної системи використаємо такий алгоритм (схема Рунге-Кутта):

1. Задаємо необхідну точність обчислень ε та максимально допустиме значення кроку за часом $\triangle t_{max}$.

2. На кожному кроці за часом розраховуємо коефіцієнти:

$$\begin{aligned} \overrightarrow{k}_1 &= f(\overrightarrow{x}(t), t)\triangle t, \\ \overrightarrow{k}_2 &= hf(\overrightarrow{x}(t) + \overrightarrow{k}_1, t + \triangle t)\triangle t, \\ \overrightarrow{k}_3 &= hf(\overrightarrow{x}(t) + (\overrightarrow{k}_1 + \overrightarrow{k}_2)\triangle t/4, t + \triangle t/2)\triangle t, \end{aligned} \tag{7.14}$$

де $\triangle t$ — крок за часом.

3. Оцінюємо поточну δ та максимально допустиму δ_{max} похибки:

$$\begin{aligned} \delta &= \max((\overrightarrow{k}_1 - 2\overrightarrow{k}_3 + \overrightarrow{k}_2)\triangle t/3), \\ \delta_{max} &= \varepsilon \cdot \max(\max(\overrightarrow{x}), 1.0), \end{aligned} \tag{7.15}$$

де max — максимальне (за модулем) значення серед компонентів вектора, вказаного у дужках.

4. Якщо $\delta < \delta_{max}$, розраховуємо значення $\overrightarrow{x}$ у наступний момент часу $t + \triangle t$:

$$\overrightarrow{x}(t + \triangle t) = \overrightarrow{x}(t) + (\overrightarrow{k}_1 + 4\overrightarrow{k}_3 + \overrightarrow{k}_2)\triangle t/6. \tag{7.16}$$

5. Змінюємо величину кроку $\triangle t$:

$$\triangle t = \min(\triangle t_{max}, 0.9\triangle t(\delta_{max}/\delta)^{1/3}), \tag{7.17}$$

де min — мінімальне (за модулем) значення із заданого масиву.

Показник Ляпунова. Чисельна процедура обчислення старшого показника Ляпунова для багатопараметричної динамічної системи наводиться у підрозділі "Алгоритми".

7.3 Алгоритми

Розглянемо алгоритм обчислення старшого показника Ляпунова.

1 Отримуємо чисельний розв'язок динамічних рівнянь на інтервалі часу, який є достатнім для того, щоб траєкторія система вийшла на атрактор. У результаті одержуємо деяку точку фазового простору $\overrightarrow{x}(0)$, яку будемо вважати за вихідну.

2 Розраховуємо траєкторію, що виходить із точки $\overrightarrow{x}(0)$, та збурену траєкторію, що стартує з точки $\overrightarrow{x}(0)+\overrightarrow{\delta x}_0$. При цьому норма $||\overrightarrow{\delta x}_0|| = \varepsilon$. Для цього знаходимо чисельний розв'язок системи на інтервалі часу T і отримуємо вектор стану $\overrightarrow{x}_1 \equiv \overrightarrow{x}(T)$ і його збурення $\overrightarrow{\delta x}_1$ у даний момент часу. Відношення $||\overrightarrow{\delta x}_1||/\varepsilon$ характеризує зміну норми вектора збурення за час T.

3 Перевизначимо цей вектор так, щоб його напрямок залишився тим самим, а норма дорівнювала вихідному значенню ε, а саме

$$\overrightarrow{\delta x}_1 = \varepsilon \overrightarrow{\delta x}_1 / ||\overrightarrow{\delta x}_1||.$$

Виконуємо розв'язання на наступному інтервалі часу T, узявши за початкову точку та початкове збурення $\overrightarrow{x}(0) = \overrightarrow{x}_1$ та $\overrightarrow{\delta x}_0 = \overrightarrow{\delta x}_1$ відповідно (пункт 2). Далі процес триває. Після достатньої кількості ітерацій N переходимо до пункту 4.

4 Розраховуємо старший показник Ляпунова:

$$\lambda \simeq \frac{1}{NT} \sum_{i=1}^{N} \ln \frac{||\overrightarrow{\delta x}_i||}{\varepsilon}. \tag{7.18}$$

Більш точну статистичну оцінку показника Ляпунова можна отримати, повторивши процедуру 1-4 для різних початкових умов та різних збурень у початковий момент часу з подальшим усередненням за кількістю експериментів.

7.4 Приклади

7.4.1 Моделювання відображення "зуб пили"

Подана у додатку програма дозволяє провести моделювання відображення "зуб пили". За початкову умову обрано ірраціональне число, $x_0 = 1/\sqrt{80}$. Результат роботи програми подано на рис. 7.4, опис якого наводиться у відповідному підрозділі даного розділу.

7.4.2 Моделювання логістичного відображення

Подана у додатку програма дозволяє провести моделювання логістичного відображення та отримати значення показника Ляпунова. Результат роботи програми подано на рис. 7.6-7.8, опис яких наводиться у відповідному підрозділі даного розділу.

7.4.3 Система Лоренца

Подана у додатку програма дозволяє провести моделювання системи Лоренца методом Рунге-Кутта 3-го порядку. Вихідні дані: $\sigma = 10$, $r = 28$, $b = 8/3$; початкові умови $x(0)$, $y(0)$, $z(0)$ обираються випадковим чином; час моделювання $T = 200$; крок за часом динамічно змінюється у процесі моделювання (вихідне значення $\triangle t = 0.2$). Результати моделювання наводяться на рис. 7.9а-в.

Як бачимо з рисунка, еволюція кожної змінної у часі становить хаотичну залежність, з якої важко отримати будь-яку інформацію про властивості системи. Якщо об'єднати всі три змінні на одному графіку в тривимірному просторі, можна отримати досить нетривіальну картину (рис. 7.9г). Розглянемо цей результат більш детально. Якщо зв'язок між змінними був би відсутній, у тривимірному фазовому просторі ми отримали б хаотичну хмару точок. Якщо взаємозалежність була б однозначною та лінійною, результатом була б лінія або петля. Замість цього ми отримали графік, розміщений у певних межах, що має характерні контури, які нагадають два крила метелика. Крім того, як показав детальний аналіз, фазова точка ніколи не проходить по одних і тих самих траєкторіях, що відображає лише часткову упорядкованість системи (стан систе-

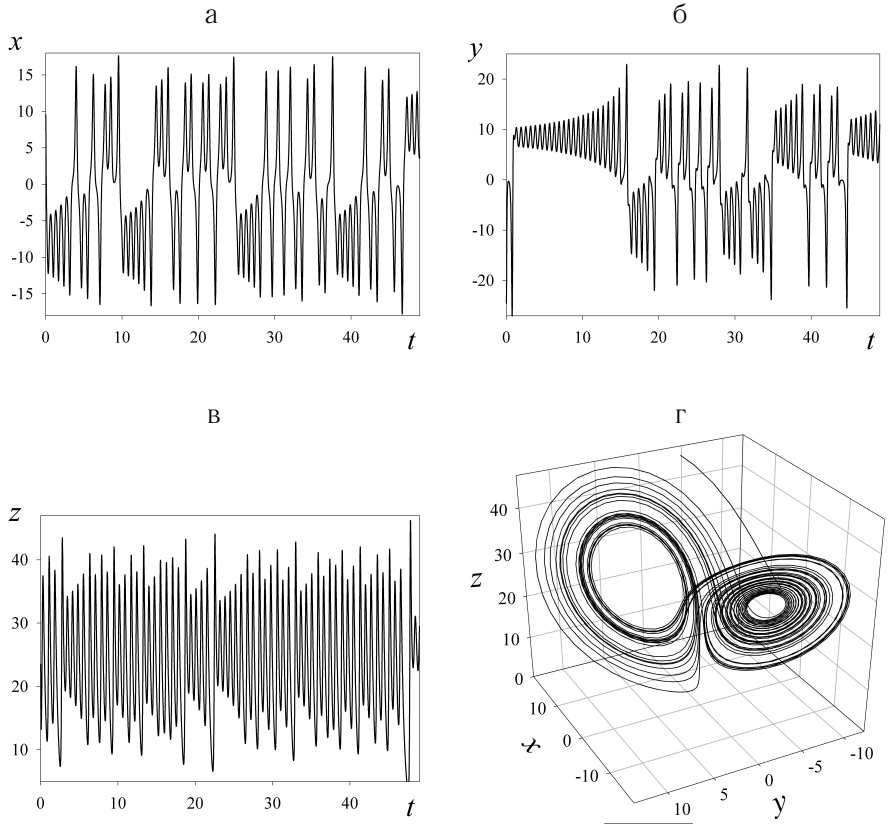

Рисунок 7.9 — Еволюція змінних (а)-(в) та фазовий портрет (г) системи Лоренца

ми ніколи не повторюється). Таким чином, отриманий графік є заплутаним клубком траєкторій, який має фрактальну структуру. Будь-яка траєкторія, що починається у віддаленій точці фазового простору, з часом потрапляє у цей клубок, проходячи по власній траєкторії[11]. Такий атрактор, як відомо, відображає поведінку детерміновано хаотичної системи

[11] Така множина точок у фазовому просторі, до якої прямують усі траєкторії називається атрактором.

і належить до атракторів особливого типу — дивним (хаотичним, стохастичним) атракторам.

Таким чином, на основі виконаного чисельного аналізу системи можна зробити такі висновки:

1. Фазові траєкторії, які утворюють дивний атрактор, ніколи не накладаються (система ніколи не проходить у точності таким самим шляхом, яким уже пройшла). Крім того, якщо збільшити будь-яку частину дивного атрактора, що буде відповідати прорахунку поведінки системи при деяких параметрах, узятих з більшою точністю, можна побачити, що окремі траєкторії розпадаються на ряд дочірніх траєкторій, з порожніми проміжками між ними. Збільшивши невелику ділянку збільшеної області ще раз, можна помітити наступний ряд траєкторій, які є невидимими при меншому розділенні. Наприклад, якщо при значенні змінної $y = 1.5$ у даній ділянці фазового простору (тобто при певних значеннях параметрів x та z, що відповідають даній області фазового простору) проходить одна траєкторія, що свідчить про можливість реалізації такого стану системи, то при збільшенні цієї ділянки можна побачити, що ця траєкторія розпадається на чотири траєкторії, які відповідають значенням змінної $y = 1.492, 1.515, 1.535, 1.567$, у той час як проміжні стани, наприклад між 1.492 та 1.515, стають недосяжними при даних x та z. Ця цікава топологічна особливість характеризує обмежену передбачуваність поведінки системи, що залежить від точності початкових даних.

2. Як показав проведений аналіз, хаотична система є стійкою до зовнішніх шумів. Якщо відхилити початкове положення точки усередині дивного атрактора, тобто змінити початковий стан системи, вже через невеликий проміжок часу слід даного збурення під шумовою завісою власних флуктуацій системи буде втрачено. В той же час схожа за силою дія на лінійну систему призведе до значної зміни динаміки, а можливо, і до повного руйнування системи.

7.4.4 Розрахунок старшого показника Ляпунова

Поданий у розділі "Алгоритми" алгоритм розрахунку показника Ляпунова реалізовано у програмі, лістинг якої наводиться у додатку.

Питання для самоконтролю

1 Назвіть причини виникнення динамічного хаосу.

2 Отримайте формулу для показника Ляпунова у випадку одновимірного відображення.

3 При якій умові поведінка відображення "зуб пили" регулярна / хаотична?

4 Що таке зміщення Бернуллі?

5 Через скільки кроків хаотична динамічна система "забуває" свій початковий стан?

6 Отримайте розв'язок для логістичного відображення.

7 Наведіть алгоритм моделювання хаотичної системи, модель якої описується диференціальними рівняннями.

8 Шо таке хаотичний атрактор?

9 Скільки вимірів може мати простір дивного атрактора?

10 Назвіть властивості дивного атрактора.

11 Яким чином розраховується показник Ляпунова для багатопараметричних систем, що описуються диференціальними рівняннями?

Завдання для самостійної роботи

1 Відображення "тент"дається рівнянням

$$x_{n+1} = \begin{cases} 2x_n, & 0 \le x_n < 1/2; \\ 2(1 - x_n), & 1/2 \le x_n \le 1. \end{cases}$$

Знайдіть показник Ляпунова.

2 Проаналізуйте, при яких початкових умовах логістичне відображення матиме періодичну траєкторію. Побудуйте відповідний графік.

3 Виконайте перевірку впливу кінцевої точності подання дійсних чисел у комп'ютері на можливість прогнозу поведінки хаотичної системи. Для цього, використовуючи подані у розділі відображення, отримайте траєкторію $x^n(n)$ системи при певній фіксованій початковій умові. Потім, використовуючи ту саму початкову умову, на кожній ітерації виконайте послідовно операції ділення, а потім множення x^n на 10. Цей прийом дозволяє виконати обрізання останнього розряду. Порівняйте отримані траєкторії. Зробіть висновки.

4 Використовуючи програми, наведені у додатку, розрахуйте залежність показника Ляпунова від параметрів системи Лоренца σ, r, b. Експериментально знайдіть порогові значення даних параметрів, при яких динаміка системи стає хаотичною.

5 Побудуйте карту динамічних режимів (у площині σ — r або b — r) для системи Лоренца. На основі показника Ляпунова визначте області, які визначають хаотичний режим.

6 Проаналізуйте чутливість системи Лоренца до збурень початкових умов. Оцініть залежність величини кінцевого відхилення траєкторії від початкового збурення для різних часових інтервалів.

7 Для дослідження внутрішньої будови атрактора Лоренца виконайте його поперечний переріз, що має назву "перетин Пуанкаре". Цей прийом зменшує число вимірів із трьох до двох. Щоразу, коли траєкторія перетинає площину, вона залишає на ній точку. Виконайте аналіз отриманого рисунка. Доведіть, що, незважаючи на те, що при такому аналізі втрачається частина інформації, він є корисним для вивчення властивостей хаотичної системи.

Список літератури

1. Арцимович Л.А. Движение заряженных частиц в электрических и магнитных полях / Л.А. Арцимович, С.Ю. Лукьянов. — М.: Наука, 1978. — 325 с.

2. Поттер Д. Вычислительные методы в физике. — М.: Москва, 1975. — 392 с.

3. Майер Р.В. Основы компьютерного моделирования: учебное пособие. — Глазов: ГГПИ, 2005. — 25 с.

4. Коткин Г.Л. Компьютерное моделирование физических процессов с использованием MATLAB / Г.Л. Коткин, В.С. Черкасский. — Новосибирск: Изд-во НГУ, 2001. — 173 с.

5. Гулд Х. Компьютерное моделирование в физике: в 2 частях / Х. Гулд, Я. Тобочник. — М.: Мир, 1990.–Ч.2.— 400 с.

6. Ландау Л.Д. Статистическая физика. Часть 1 / Л.Д. Ландау, Е.М. Лифшиц. — М.: Наука, 1976.

7. Киттель Ч. Введение в физику твердого тела. — М.: Наука, 1978.

8. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. — М.: Наука, 1987.

9. Каханер Д. Численные методы и программное обеспечение / Д. Каханер, К. Моулер, С. Неш. — М.: Мир, 1998. — 575 с.

10. Зельдович Я.Б. Элементы прикладной математики / Я.Б. Зельдович, А.Д. Мышкис. — М.: Наука, 1965. — 615 с.

11. Тихонов А.Н. Уравнения математической физики / А.Н. Тихонов, А.А. Самарский. — М.: Наука, 1966. — 724 с.

12. Хемминг Р.В. Численные методы для научных работников и инженеров: пер. с англ. — М.: Наука, 1972.

13. Демидович Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. — М.: Наука, 1966.

14. Загускин В.Л. Справочник по численным методам решения уравнений. — М.: ФИЗМАТГИЗ, 1960. — 216 с.

15. Березин И.С. Методы вычислений: в 2 т. / И.С. Березин, Н.П. Жидков. — М.: Наука, 1966.

16. Калиткин Н.Н. Численные методы. — М.: Наука, 1978. — 512 с.

17. Самарский А.А. Математическое моделирование: Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. — М.: Наука, 1997. — 320 с.

18. Форсайт Дж. Машинные методы математических вычислений / Дж. Форсайт, К. Молер, К. Малькольм. — М.: Мир, 1980. — 278 с.

19. Фаддеев Д.К. Вычислительные методы линейной алгебры / Д.К. Фаддеев, В.Н. Фаддеева. — М.: Наука, 1980.

20. Мэтьюз Д. Численные методы. Использование MATLAB, 3-е издание. : пер. с англ / Д. Мэтьюз, К. Финк. — М.: Издательский дом "Вильямс", 2001. — 720 с.

21. Терёхин В.В. Моделирование в системе MATLAB: учебное пособие. — Новокузнецк: Кузбассвузиздат, 2004. — 376 с.

22. Мироновский Л.А. Введение в MATLAB: учебное пособие / Л.А. Мироновский, К.Ю. Петрова. –– СПб.: ГУАП., 2006. — 164 с.

23. Гультяев А. Визуальное моделирование в среде MATLAB: учебный курс. — СПб.: Питер, 2000. — 364 с.

24. Hunt B.R. MatLab R2007 с нуля! — М.: Лучшие книги, 2008. — 352 с.

25. Веселова И.Ю. Моделирование: Вычислительный практикум / И.Ю. Веселова, Ю.Б. Сениченков. — СПб: Изд-во СПбГТУ, 1999. — 99 с.

26. Бусленко Н.П. Моделирование сложных систем. — М.: Наука, 1978. — 384 с.

27. Бенькович Е.С. Практическое моделирование динамических систем / Е.С. Бенькович, Ю.Б. Колесов, Ю.Б. Сениченков. — СПб.: БХВ-Петербург, 2002. — 464 с.

28. Колесов Ю.Б. Визуальное моделирование сложных динамических систем / Ю.Б. Колесов, Ю.Б. Сениченков. — СПб.: Изд-во Мир и Семья & Интерлайн, 2000. — 242 с.

29. Элементарный учебник физики: в 3 т. / под ред. Г.С. Ландсберга. — М.: АОЗТ "Шрайк", 1995.

30. Пановко Я.Г. Введение в теорию механических колебаний. — М.: Наука, 1971. — 242 с.

31. Кузнецов С.П. Динамический хаос. — М.: Физматлит, 2001.

32. Лихтенберг А. Регулярная и стохастическая динамика / А. Лихтенберг, М. Либерман. — Череповец: Меркурий-ПРЕСС, 2000.

33. Шустер Г. Детерминированный хаос: введение. — М.: Мир, 1988.

34. Заславский Г.М., Сагдеев Р.З. Введение в нелинейную физику: от маятника до турбулентности и хаоса / Г.М. Заславский, Р.З. Сагдеев. — М.: Наука, 1988.

35. Steeb W.-H. The Nonlinear workbook / W.-H. Steeb, Y. Hardy, R. Stoop. — Singapore: World Scientific, 2008.

36. Ruelle D. Where can one hope to profitably apply the ideas of chaos? // Physics Today. — 1994. — Vol. — P. 24-30.

37. Nicolis G. Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations / G. Nicolis, I. Prigogine. — New York: JohnWiley, 1977.

38. Haken H. Synergetics, An Introduction. — Berlin: Springer-Verlag, 1983.

Покажчик

Додаток А
(довідковий)

Рух частинки у різних силових полях (до розділу 1)

```
...
double dt, M, m, r, B, q, E;
void F_const(double x, double y, double vx,double vy,
                                  double *Fx, double *Fy)
    // Функція повертає проекції сили F у випадку
    // однорідного поля.
{
  *Fx=1.0;
  *Fy=3.0;
};
void F_centr(double x, double y, double vx,double vy, \
                                  double *Fx, double *Fy)
   // Функція F_centr повертає проекції сили F під час
   // руху у центрально-симетричному полі.
{
    double r=sqrt(x*x+y*y);
    double F=M*m/(r*r);
    if (fabs(r)<1e-2) exit(0);
    *Fx=-F*x/r;
    *Fy=-F*y/r;
};
void F_Magnit(double x, double y, double vx,double vy, \
                                   double *Fx, double *Fy)
  // Функція F_Magnit повертає проекції сили F під час
  // руху у магнітному полі.
{
    double v=sqrt(vx*vx+vy*vy);
    double F=B*v*q;
```

```
    *Fx=F*vy/v;
    *Fy=-F*vx/v;
};
void F_Magnit_Electr(double x, double y, double vx,
                        double vy, double *Fx, double *Fy)
  // Функція  F_Magnit  повертає  проекції  сили  F  під час
  // руху частинки у магнітному та електричному полі.
{
    double v=sqrt(vx*vx+vy*vy);
    double F=B*v*q;
    *Fx=F*vy/v;
    *Fy=q*E-F*vx/v;
};
void E_K (void(*Func)(double,double,double,double,double*,
         double*),double x0, double y0,double v_x0,
         double v_y0)
    // Реалізація методу "з переступом".
{
  double  t, x, y, v_x, v_y, Fx, Fy;
  t=0;
    // виводимо на екран точку з координатами (x0,y0)
  x=x0;
  y=y0;
  v_x=v_x0;
  v_y=v_y0;
  while (!kbhit())
   {
    Func(x,y,v_x,v_y,&Fx,&Fy);
    v_x=v_x+(Fx-r*v_x)/m*dt;
    v_y=v_y+(Fy-r*v_y)/m*dt;
    x=x+v_x*dt;
    y=y+v_y*dt;
      // виводимо на екран точку з координатами (x,y)
    t+=dt;
```

```
   };
};

void main()
{
  B=2; q=1; dt=0.001, M=500, m=10, r=0.1, E=1;
  double x0=-10, y0=10, v_x0=3, v_y0=2;
    // Викликаємо функцію Perestup. Як перший параметр
    // передаємо адресу відповідної функції (фактично -
    // тип поля).
   E_K(&F_const,x0,y0,v_x0,v_y0);
    // За необхідності знімаємо коментар
// E_K(&F_centr,x0,y0,v_x0,v_y0);
// E_K(&F_Magnit,x0,y0,v_x0,v_y0);
// E_K(&F_Magnit_Electr,x0,y0,v_x0,v_y0);
};
```

Рух системи взаємодіючих частинок (до розділу 2)

```
const N=100;    // Кількість частинок
const Lx=20;    // Лінійні розміри прямокутного резервуара
const Ly=20;
const Scale=10; // Масштаб виведення точок на екран
double vx[N], vy[N], x[N], y[N], xbak[N], ybak[N], vmax=20;
double t=0, dt=1e-3, epsilon=0.5, sigma=1;
void Initial()
 {
  int j=1, i=1, k=0;
  double rx, ry, r;
  randomize();
         // Випадковим чином задаємо координати та проекції
         // швидкостей кожної частинки
  for (i=0; i<N; i++)
   {
```

```
  x[i]=random(Lx);
  y[i]=random(Ly);
  vx[i]=vmax*(2.0*random(1001)/1000.0-1);
  vy[i]=vmax*(2.0*random(1001)/1000.0-1);
 };
    // Коригуємо  координати  таким  чином, щоб
    // частинки були розподілені у  резервуарі
    // більш-менш рівномірно
for(;;)
 {
  int flag=0;
  for (i=0; i<N-1; i++)
   for (int j=i+1; j<N; j++)
    {
       rx=x[j]-x[i];
       ry=y[j]-y[i];
       r=sqrt(rx*rx+ry*ry);
       if (r<1)
        {
         x[j]=random(Lx);
         flag=1;
        };
    };
 if (!flag) break;
};
    // Розраховуємо повний імпульс системи в x-
    // та  y-напрямках і коригуємо  швидкості
    // таким чином, щоб повний імпульс системи
    // дорівнював нулю
double SumVx=0, SumVy=0;
for (i=0; i<N; i++)
  {
   SumVx+=vx[i];
   SumVy+=vy[i];
```

```
    }
  SumVx/=N;
  SumVy/=N;
  for (i=0; i<N; i++)
    {
     vx[i]-=SumVx;
     vy[i]-=SumVy;
    }
 };
    // Якщо  відстань між частинками перевищує половину
    // геометричного розміру резервуара - зменшуємо  її
    // відповідно  до  припущення про періодичність меж
    // резервуара (див. опис до програми)
void Test1(double *dx, double *dy)
 {
   int sign=*dx>0?1:-1;
   if (fabs(*dx)>Lx/2.0)
       *dx=*dx-sign*Lx;
   sign=*dy>0?1:-1;
   if (fabs(*dy)>Ly/2.0)
       *dy=*dy-sign*Ly;
 };
      // Якщо  частинка  досягає  стінки  резервуара, то
      // вона  з'являється з іншого боку
void Test2(double *x, double *y) {
  if (*x<0) *x=*x+Lx;
  if (*x>Lx) *x=*x-Lx;
  if (*y<0) *y=*y+Ly;
  if (*y>Ly) *y=*y-Ly;
};
         // Рисуємо частинки на екрані
void Out(double x[N], double y[N], double x1[N], \
                                          double y1[N])
{
```

```
  for (int i=0; i<N; i++)
   {
   ...
     // Видаляємо   частинки  з координатами  x[N], y[N].
     // Для  цього як поточний колір вибираємо колір фону
     // і малюємо даним кольором  частинки на екрані.
     // Координати перераховуємо згідно заданого масштабу
     // x[N/2]*Scale, y[N/2]*Scale
   ...
     // Вибираємо інший колір для виведення частинок.
     // Виводимо  частинки  у  точки  з  координатами
     // x1[i]*Scale, y1[i]*Scale
   }
};
     // Виводимо на екран залежність температури від часу
void GrT(double t, double T)
 {
  ...
     // Виводимо точку з координатами ((400+t*50),(300-T))
     // на графік
  ...
};
     // Розраховуємо траєкторії руху системи частинок
double x_n[N], y_n[N], vx_n[N], vy_n[N], a[N],b[N];
void Simula()
 {
  int i;
  double rx, rx6, ry, ry6, Fx, Fy, r, r6, F;
  double t=0, T, SumX, SumY;
    // На наступних кроках за часом координати та швидкості
    // частинок розраховуємо за методом з переступом
  for (i=0; i<N-1; i++)
   {
    SumX=SumY=0;
```

```
   for (int j=i+1; j<N; j++)
    {
     rx=x[i]-x[j];
     ry=y[i]-y[j];
     Test1(&rx,&ry);
     r=sqrt(rx*rx+ry*ry);
     r6=pow(sigma/r,6);
     F=24*epsilon*sigma/r*r6*(2*r6-1);
     SumX=SumX+F*rx;
     SumY=SumY+F*ry;
           // Згідно з третім законом Ньютона, збільшуючи
           // імпульс (швидкість) однієї із  взаємодіючих
           // частинок, зменшуємо імпульс іншої
    a[j]-=F*rx;
    b[j]-=F*ry;
   };
 vx[i]=vx[i]+(SumX+a[i])*dt/2;
 vy[i]=vy[i]+(SumY+b[i])*dt/2;
 };
vx[i]=vx[i]+a[i]*dt/2;
vy[i]=vy[i]+b[i]*dt/2;
while (!kbhit())
 {
   T=0;
   for (i=0; i<N; i++)
    {
     x_n[i]=x[i]+vx[i]*dt;
     y_n[i]=y[i]+vy[i]*dt;
     Test2(&x_n[i],&y_n[i]);
    };
   ...
      // Обнуляємо масиви a та b
   ...
   for (i=0; i<N-1; i++)
```

```
 {
  SumX=0;
  SumY=0;
  for (int j=i+1; j<N; j++)
   {
    rx=x_n[i]-x_n[j];
    ry=y_n[i]-y_n[j];
    Test1(&rx,&ry);
    r=sqrt(rx*rx+ry*ry);
    r6=pow(sigma/r,6);
           // Похідна від потенціалу Леннарда-Джонса
    F=24*epsilon*sigma/r*r6*(2*r6-1);
    SumX=SumX+F*rx;
    SumY=SumY+F*ry;
    a[j]-=F*rx;
    b[j]-=F*ry;
   };
  vx_n[i]=vx[i]+(SumX+a[i])*dt;
  vy_n[i]=vy[i]+(SumY+b[i])*dt;
 };
vx_n[i]=vx[i]+a[i]*dt;
vy_n[i]=vy[i]+b[i]*dt;
           // Розраховуємо температуру системи
for (i=0; i<N; i++)
  T+=vx_n[i]*vx_n[i]+vy_n[i]*vy_n[i];
T=T/N;
GrT(t,T);
Out(x,y,x_n,y_n);
...
     // копіюємо елементи масиву $x_n$  у  масив $x$,
     // відповідно  $y_n$ у масив $y$, $vx_n$ у $vx$,
     // $vy_n$ у масив $vy$
...
t+=dt;
```

```
   };
};
void main() {
  ...
  Initial();
  Simula();
  ...
}
```

Більярд Синая (до розділу 2)

```
%%%%%%%%%%%%%%%%%
% Програма Sinaj %
%%%%%%%%%%%%%%%%%
global R Lx Ly; R=5; Lx=100; Ly=100;
r=[10 60; 40 80];  % x- та y-координати куль
v=[10 3; 6 7];     % x- та y-компоненти швидкостей куль
plothandle = plot(r(1,:),r(2,:),'o', ...
      'Color','black', ...
      'MarkerSize',25);
 % Створюємо графічний  об'єкт - plot, за допомогою якого
 % рисуємо на екрані кулі у точках з x-координатами r(1,:)
 % (усі стовпці першого рядка матриці r) та y-координатами
 % r(2,:). Кулі розміром 25 чорного кольору виводяться на
 % екран у формі кола  ('o'). У подальшому робота з даним
 % об'єктом виконується через  дескриптор  plothandle  за
 % допомогою команди set.
axis([0 Lx 0 Ly]);
           % Встановлюємо  межі  зміни x- та y-координат.
dt=0.05;   % крок за часом
t=0;
pause;
 % пауза забезпечує негайне виведення зображення на екран;
 % для продовження необхідно натиснути будь-яку клавішу
```

```
while t<100,   % час розрахунків
  [r,v]=Raschet(r,v,dt);      % Викликаємо функцію Raschet,
      % передаємо початкові координати (масив r), швидкості
      % (масив v) та часовий інтервал. Як вихідні дані
      % [r,v] функція повертає нові координати та швидкостя
      % через час dt.
  pause(0.001);           % пауза для високошвидкісних ЕОМ
  set(plothandle,'xdata',r(1,:),'ydata',r(2,:));
                          % Перерисовуємо кулі на екрані.
  t=t+dt;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Підпрограма розрахунку траєкторій руху куль на проміжку %
% часу dt                                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [r,v]=Raschet(rin,vin,dt);
global R Lx Ly;
RR=4*R*R;
eps=1E-8;
r=rin; v=vin;
Left_Down_Corner=[R R; R R];
       % Координати лівої нижньої межі області для обох
       % куль, яка є доступною центрам куль
Right_Upper_Corner=[Lx-R Ly-R ;Lx-R Ly-R];
       % Координати правої верхньої межі
t=dt;
while (t>0)
 time_Left_Down_col=(Left_Down_Corner-r)./v;
       % Розраховуємо усі можливі моменти зіткнення з лівою
       % (у x-напрямку) та нижньою (у y-напрямку) стінками
       % для обох куль.
 time_Left_Down_col=time_Left_Down_col+...
                         1E8*((time_Left_Down_col <=0));
 % Якщо зіткнення відбулося у минулому
```

```
% time_Left_Down_col<=0,
% даний момент часу зіткнення нескінченно   збільшуємо
% (у подальшому він використовуватися не буде), інакше
% залишаємо без змін
time_Right_Upper_col=(Right_Upper_Corner-r)./v;
time_Right_Upper_col=time_Right_Upper_col+...
                    1E8*((time_Right_Upper_col <=0));
[t1,j1]=min(time_Left_Down_col(:));
[t2,j2]=min(time_Right_Upper_col(:));
% Вибираємо найбільш  раннє  зіткнення із стінками
% (t1 та t2), і кулю, яка  зіткнулася  із  стінкою
% (j1 та j2).
if(t1<t2)
 t0=t1;j0=j1;
else
 t0=t2;j0=j2;
end;
  % згідно з наведеними формулами знаходимо момент
  % зіткнення куль
r0=r(:,1)-r(:,2);
v0=v(:,1)-v(:,2);
rr=r0'*r0;
vv=v0'*v0;
rv=r0'*v0;
d=rv*rv-(rr-RR)*vv;      % дискримінант
if (d>0)&(rv<0)&(vv>eps)
 time_col=-(sqrt(d)+rv)/vv;
else
 time_col=inf;           % нескінченність
end;
if(t<t0)&(t<time_col)
          % Якщо поточний час менший за час
          % зіткнення куль зі стінкою або одна з одною
 r=r+v.*t;         % розраховуємо нові координати куль
```

```
 elseif (t0<time_col)  % якщо досягли часу зіткнення із
                       % стінкою
  r=r+v.*t0;
  v(j0)=-v(j0);   % змінюємо напрямок руху частинки при
                  % зіткненні із стінкою
 else             % зіткнення куль одна з одною
  t0=time_col;
  r=r+v.*t0;
  r0=r(:,1)-r(:,2);
  v0=v(:,1)-v(:,2);
  dv=r0'*v0/RR*r0;
  ddv=[-dv,dv ];
      % змінюємо швидкості відповідно до наведених
      % у розділі формул
  v=v+ddv;
 end;
 t=t-t0;
end;
```

Моделювання класичного ідеального газу (до розділу 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Програма мікроканонічного моделювання методом  %
% Монте-Карло   класичного   ідеального   газу.  %
% Складається із двох модулів, що розташовані у  %
% файлах main.m та ideal_gas.m.                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% Лістінг файлу main.m %
%%%%%%%%%%%%%%%%%%%%%%%%
E=30;    % енергія системи
N=100;   % число частинок
NTrial=1000;     % кількість випробувань
dV=2.5;          % максимальне значення зміни швидкості
```

```
V=(2*E./N).^0.5; % швидкість, що припадає на одну частинку
                 % для заданої енергії системи E=N (mv^2/2)
                 % m=1
for i=1:N
  v(i)=V;        % масив швидкостей
end;
[E_c Aver_v Num E_mas]=Ideal_gas(N,E,NTrial,v,dV);
  % Функція Ideal_gas повертає середнє за ансамблем
  % значення енергії ''конденсатора'' E_c,  середню
  % швидкість на одну частинку системи Aver_v    та
  % відсоток прийнятих конфігурацій  Num.
E_c
Aver_v
Num
             % Будуємо розподіл енергії "конденсатора".
x_min=min(E_mas);      % мінімальне  значення енергії
                       % "конденсатора"
x_max=max(E_mas);      % максимальне значення енергії
                       % "конденсатора"
Nint=50;   % кількість інтервалів, на які розбивається
           % відрізок  [x_min; x_max]  для    побудови
           % гістограми
i=1:Nint;
x(i)=x_min+(x_max-x_min)/(Nint-1)*(i-1);
h=hist(E_mas,x);   % функція  повертає  розподіл  E_mas
                   % серед length(x) інтервалів, центри
                   % яких зазначені у масиві z
bar(x,h);          % будуємо гістограму
axis([0 x_max 0 max(h)])
colormap white
hold on
   % Далі, використовуючи  пакет  Curve  Fitting  Tool
   % виконуємо аналіз гістограми і знаходимо параметри
   % функції, що наближує представлені  на  гістограмі
```

```
   % дані. Результат  апроксимації виводимо на екран ...
y=a*exp(-x/b); % a=13500 b=0.66 - результат апроксимації
           % за отриманими даними поточного експерименту
plot(x,y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Лістінг файлу ideal_gas.m %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [E_k Aver_v Num E_mas]=Ideal_gas(N,E,NTrial,v,dV)
   % функція, яка повертає усереднені значення:
   % енергія конденсатора (E_k),
   % швидкість на одну частинку (Aver_v),
   % число прийнятих конфігурацій (Accept).
   % Миттєві значення енергії конденсатора (E_mas).
E_k=0;
E_mas(1)=E_k;
Vtot=sum(v);
Vsum=0;
Esum=0;
Num=0;
m=2;
for j=1:NTrial
   for i=1:N
      dv=(2*rand(1)-1)*dV;    % випадкова зміна швидкості
      i_r=floor(N*rand(1)+1); % випадковим чином вибираємо
                              % частинку
      VTrial=v(i_r)+dv;       % змінюємо її швидкість
      de=0.5*(VTrial.^2-v(i_r)^2);
                  % розраховуємо зміну кінетичної енергії,
                  % що відбулася у результаті зміни
                  % швидкості частинки
      if de<=E_k  % якщо енергія зменшилася, приймаємо
                  % конфігурацію
         v(i_r)=VTrial;
```

```
                    % міняємо швидкість вибраної частинки
          Vtot=Vtot+dv;
          Num=Num+1;
          E_k=E_k-de;
       end;
       Esum=Esum+E_k;
       Vsum=Vsum+Vtot;
       E_mas(m)=E_k;
       m=m+1;
    end;
end;
E_k=Esum/(NTrial*N);
Num=Num/(NTrial*N);
Aver_v=1/N*Vsum/(NTrial*N);


%%%%%%%%%%%%%%%%%%%%%%%%
% Результати розрахунків %
%%%%%%%%%%%%%%%%%%%%%%%%
E_c =0.5910
Aver_v = -5.0861e-004
Num = 0.4665
```

Модель Ізинга (до розділу 3)

```
     // Розмір системи
const int Lx=64;
const int Ly=64;
     // Кількість ітерацій за методом Монте-Карло
const N_iter=1000;
int Spins[Lx][Ly];
long int M;
double T=2;   // Температура теплового резервуара,
              // в якому знаходиться система
```

```
int Sum(int i, int j)
 // Функція  повертає  суму  за усіма найближчими
 // сусідами вузла (i,j). При  знаходженні   суми
 // враховуємо періодичність граничних умов.
{
int left, right, up, down;
if (i==0)
  left=Spins[Lx-1][j];
else
    left=Spins[i-1][j];
if (i==Lx-1)
  right=Spins[0][j];
else
  right=Spins[i+1][j];
if (j==0)
  up=Spins[i][Ly-1];
else
  up=Spins[i][j-1];
if (j==Ly-1)
  down=Spins[i][0];
else
  down=Spins[i][j+1];
return left+right+up+down;
};

void Monte()
     // Реалізація методу Монте-Карло
{
int Sum_Nb=0, left, right, up, down;
double DeltaE, P;
for (int n=1; n<=Lx*Ly; n++)
{
 int i=random(Lx); //Випадковим чином вибираємо спін
 int j=random(Ly);
```

```
 // Розраховуємо енергію,  яку вносить вибраний спін
 // та його оточення  у загальну  енергію  системи з
 // протилежним знаком. Якщо отримане значення менше
 // нуля, відповідно у  повну  енергію  воно входить
 // з  додатним   знаком, збільшуючи енергію   усієї
 // системи. Отже,  перевертаємо спін і розраховуємо
 // намагніченість. Спін  перевертається  і у  разі,
 // коли енергія збільшується, але не перевищує певне
 // значення згідно з розподілом Гіббса (див. опис до
 // програми).
Sum_Nb=Spins[i][j]*Sum(i,j);
if (Sum_Nb<=0)
{
 Spins[i][j]=-Spins[i][j];
 M=M+2*Spins[i][j];
}
else
 {
 DeltaE=2*Sum_Nb;
 P=exp(-DeltaE/T);
 if (random(11)/10.0<P)
  { Spins[i][j]=-Spins[i][j];
    M=M+2*Spins[i][j];
  };
 };
};
};
void Out(double x, double y)
{
 // Виводимо на екран точку  з координатами
 // (x*10,y*50); 10,50 - масштаби виведення
};
void main()
{
```

```
 double NMagn;
 randomize();
 while (T>0.1)
 {
  M=0;
  NMagn=0;
    // Випадковим чином розташовуємо спіни на гратці
  for (int i=0; i<Lx; i++)
   for (int j=0; j<Ly; j++)
    {
     if (random(11)/10.0 > 0.5)
       Spins[i][j]=1;
      else
        Spins[i][j]=-1;
     M=M+Spins[i][j];
    };
  for(i=0; i<N_iter; i++)
  {
   Monte();
   NMagn=NMagn+M;
  };
    // Середня намагніченість на спін
  NMagn=NMagn/(N_iter*Lx*Ly);
  Out(T,fabs(NMagn));
  T=T-0.1;
 };
};
```

Моделювання поширення тепла у пластині (до розділу 4)

```
...
#define GRAD 16.0 // Кількість градацій сірого кольору
                  // Лінійні розміри пластини
const Lx=100;
const Ly=40;
```

```
const time_max=200;
const T1=10, T2=100; // Температура на краях пластини
int   SideX=3, SideY=3, dX, dY, LenX, LenY;
float D=200, dt=0.001, t;
   // Температуру комірок пластини будемо зберігати в
   // в одновимірному масиві T
float T[Lx*Ly+2*(Lx+Ly)];
float func(int i, int j)
{
 return T[i+(Lx+2)*j]+D*dt*(T[i+1+(Lx+2)*j]+T[i-1+(Lx+2)*j]+
 T[i+(Lx+2)*(j-1)]+T[i+(Lx+2)*(j+1)]-4*T[i+(Lx+2)*j]);
}
void CalcNext()
{
 float TempMas[Lx*Ly+2*(Lx+Ly)];

 for(int j=1; j<=Ly; j++)
   for(int i=1; i<=Lx; i++)
     TempMas[i+(Lx+2)*j]=func(i,j);
 for (int i=0;i<=Lx+1;i++)
   TempMas[i]=TempMas[i+(Ly+1)*(Lx+2)]=T1;
 for (j=1;j<=Ly+1;j++)
   TempMas[(Lx+2)*j]=TempMas[(Lx+2)*j-1]=T2;
 ...
     // Поновлюємо масив T шляхом копіювання в нього
     // значень елементів з масиву TempMas
 ...
}
void OutPix()
{
 for (int j=1; j<=Ly; j++)
  for (int i=1; i<=Lx; i++)
   {
    ...
```

```
     // Виводимо на екран комірки відповідного кольору
     // з номером (GRAD-1)/(T2-T1)*(T[j*(Lx+2)+i]-T1))
    ...
   }
}
void Simula()
{
 ...
 t=dt;
 for(int j=1;j<=Ly;j++)
  for(int i=1;i<=Lx;i++)
    T[i+(Lx+2)*j]=T1;
 for (int i=0;i<=Lx+1;i++)
  T[i]=T[i+(Ly+1)*(Lx+2)]=T1;
 for (j=1;j<=Ly+1;j++)
  T[(Lx+2)*j]=T[(Lx+2)*j-1]=T2;
 OutPix();
 while (t<=time_max && !kbhit())
  {
   CalcNext();
 // Для зменшення витрат машинного часу поновлення
 // екрана виконуємо лише кожну 33-тю ітерацію за часом.
   if (!fmod(ceil(t*1000), 33))
     OutPix();
   t+=dt;
  }
 ...
}
void main()
{
 ...
 Simula();
}
```

Розрахунок температурного профілю (до розділу 4)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Розрахунок розподілу тепла у тонкій двовимірній пластині %
% із сталою температурою на межах. Програма складається %
% із двох файлів: головного модуля main та M-файла-функції %
% heat2d, що реалізує алгоритм розрахунку та виведення %
% результатів на екран. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%
% Текст файлу main.m %
%%%%%%%%%%%%%%%%%%%%%%
dx=1; % крок за простором (однаковий у x- та y-напрямках)
dt=0.01;% крок за часом
Tmax=60; % час моделювання
D=2;% коефіцієнт дифузії
% розмір пластини
Nx=50; Ny=50;
T=0:dt:Tmax;
init=10*ones(Nx,Ny);  % температура на межах дорівнює 10
init(2:Nx-1,2:Ny-1)=0; % температура усередині області 0
heat2d(T,init,D,dx,dt); % виклик процедури розрахунків

%%%%%%%%%%%%%%%%%%%%%%%
% Текст файла heat2d.m %
%%%%%%%%%%%%%%%%%%%%%%%
function heat2d(t,init,D,dx,dt)
 [i,j]=size(init);
 n=length(t);
 koef=D*dt/dx^2;
 u=init;
 [X,Y]=meshgrid(1:i,1:j);
 for k=1:n-1
    u(2:i-1,2:j-1)=koef*(u(3:i,2:j-1)+u(1:i-2,2:j-1)+...
```

```
        u(2:i-1,3:j)+u(2:i-1,1:j-2))+...
        (1-4*koef)*u(2:i-1,2:j-1);
    mesh(X,Y,u)
    axis([1 i 1 j 0 10]);
    pause(0.01);
 end;
```

Поперечні коливання осциляторів (до розділу 5)

```
const int N=60;    // Кількість осциляторів
  // Параметри моделі (див. опис до програми)
float m=1, r=0.2, k=0.02, dt=0.001, A=30, Omega=4, q=80;
void main() {
  static float v[N], x[N];
  int x_centr, y_centr;
  float x_bak, accel, F, t;
  t=0;
  while (!kbhit())
  {
   for (int i=0; i<N; i++)
    {
     x_bak=x[i];
       // Розрахунок  швидкості  та  координати у
       // наступний момент часу за методом Ейлера
     F=q*(x[i-1]-x[i])+q*(x[i+1]-x[i]);
     accel=(F-r*v[i]-k*x[i])/m;
     v[i]=v[i]+accel*dt;
     x[i]=x[i]+v[i]*dt;
       // Зсув крайнього лівого осцилятора.
       // 3.14/Omega - півперіод  коливань
     if (Omega*t<3.14)
       x[0]=A*sin(4*t);
      else
        x[0]=0;
```

```
     x[N-1]=0; // Зсув крайнього правого осцилятора
     x_centr=10*i;
     y_centr=240-ceil(x[i]*5);
     ...
      // Видаляємо  попередній рисунок, для чого
      // кольором фону рисуємо осцилятор у точці
      // (x_centr,240-ceil(x_bak*5))
      // Встановлюємо новий колір для  виведення
      // осцилятора та рисуємо осцилятор у точці
     // (x_centr,y_centr);
     ...
    }
   t=t+dt;
  };
};
```

Поздовжні коливання осциляторів (до розділу 5)

```
  // Моделювання руху лінійного ланцюжка осциляторів,
  // які здійснюють поздовжні коливання уздовж осі x.
const int N=10;   // Кількість осциляторів
  // Параметри моделі (див. опис до програми)
float m=1, r=0, dt=0.001, q=80;
float v[N], x[N], a[N], x_bak[N];
void Out() {
  int x_centr, y_centr;
  for (int i=0; i<N; i++)
   {
     x_centr=60*i+ceil(x[i]*20);
     ...
       // Видаляємо  попередній рисунок, для чого
       // кольором фону рисуємо осцилятор у точці
       // (60*i+ceil(x_bak[i]*20),240)
       // Встановлюємо новий колір для  виведення
```

```
      // осцилятора та рисуємо осцилятор у точці
      // (x_centr,240);
    ...
  };
};

void main() {
  int size=sizeof(x_bak);
  float F, t;
    // Припускаємо, що усі осцилятори знаходяться у
    // положенні рівноваги. Початкова     швидкість
    // кожного  осцилятора  дорівнює нулю.
  for (int i=0; i<N; i++)
   {
    v[i]=0;
    x_bak[i]=0;
    x[i]=0;
   };
  x[0]=2;  // Виводимо  перший  осцилятор  із положення
           // рівноваги. Надаємо йому початковий зсув 2.
  x_bak[0]=2;
  Out();
  t=0;
  while (!kbhit())
  {
   for (int i=1; i<N-1; i++)
    {
     F=q*(x[i-1]-x[i])+q*(x[i+1]-x[i]);
     a[i]=(F-r*v[i])/m;
    }
    F=q*(x[1]-x[0])-q*x[0];
    a[0]=(F-r*v[0])/m;
    F=q*(x[N-2]-x[N-1])-q*x[N-1];
    a[N-1]=(F-r*v[N-1])/m;
```

```
      // Розраховуємо нову швидкість та координату за
      // методом Ейлера.
   for (i=0; i<N; i++)
    {
     v[i]=v[i]+a[i]*dt;
     x[i]=x[i]+v[i]*dt;
    };
   Out();
   ...
      // Поновлюємо масив x_bak шляхом копіювання в
      // нього значень елементів з масиву x
   ...
  };
  t=t+dt;
};
```

Розрахунок енергії системи осциляторів (до розділу 5)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Опис функції, що повертає значення правих частин %
% системи ДУ. Як вхідні дані функція отримує два %
% масиви: масив моментів часу, масив початкових координат %
% та швидкостей кожного із N осциляторів %
%(x(1)- координата 1-го осцилятора, x(2)- його швидкість, %
% x(3)- координата 2-го осцилятора, x(4)- його швидкість %
% і т.д. Текст програми розміщується у файлі oscilr.m. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=oscilr(t,x)
 global k m N % Глобальні змінні (відповідно жорсткість
              % пружинок, маси та кількість осциляторів)
 y=zeros(2*N,1);
 % Масив $y$ містить праві частини системи рівнянь.
 % y(1) - швидкість першого осцилятора
 % y(2) - сила, що діє на перший осцилятор
```

```
 % y(3) - швидкість другого осцилятора
 % y(4) - сила, що діє на другий осцилятор
 % і т.д.
 y(1)=x(2);
 y(2)=-(k(1)/m(1))*x(1)-(k(2)/m(1))*(x(1)-x(3));
 j=3;
 for i=2:N-1
   y(j)=x(j+1);
   y(j+1)=-(k(i)/m(i))*(x(j)-x(j-2))-...
                              (k(i+1)/m(i))*(x(j)-x(j+2));
   j=j+2;
 end;
 y(2*N-1)=x(2*N);
 y(2*N)=-(k(N)/m(N))*(x(2*N-1)-x(2*N-3))-...
                                    k(N+1)/m(N)*x(2*N-1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Опис функції, що розраховує енергію кожного з %
% осциляторів у заданих точках часового інтервалу. %
% Текст програми розміщується у файлі Energ.m. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function E=Energ(N,m,k,M)
 NT=length(M); % Кількість точок на часовому інтервалі
 c=1;
 for j=1:N     % цикл за усіма осциляторами
  for i=1:NT   % цикл за усіма моментами часу
    if j>1
      if j<N
        e(i)=0.5*m(j)*M(i,2*j).^2...
              +0.25*k(j)*(M(i,2*j-1)-M(i,2*j-3)).^2+...
              0.25*k(j+1)*(M(i,2*j-1)-M(i,2*j+1)).^2;
        % Розраховуємо повну енергію i-го осцилятора у
        % j-й момент часу  як суму кінетичної $mv^2/2$
        % та потенціальної $kx^2/2$ енергій.
```

```
      end;
    end;
    if j==1
      e(i)=0.5*m(1)*M(i,2)^2+0.5*k(1)*M(i,1)^2+...
                          0.25*k(2)*(M(i,1)-M(i,3))^2;
    end;
    if j==N
      e(i)=0.5*m(N)*M(i,2*N)^2+0.5*k(N+1)*M(i,2*N-1)^2+...
                       0.25*k(N)*(M(i,2*N-1)-M(i,2*N-3))^2;
    end;
  end;
  energ(:,c)=e'; % c - індекс осцилятора.
                 % energ(:,c) - усі рядки c-го стовпця
  c=c+1;
 end;
E=energ;
```

Для обчислення миттєвих значень енергії тіл коливальної системи та її повної енергії у командному вікні виконуємо таку послідовність команд:

```
>> global k m N  % опис глобальних змінних
>> N=30;         % кількість осциляторів
>> n=1:N;
>> m(n)=1;       % задаємо маси осциляторів
>> i=1:N+1;
>> k(i)=1;       % задаємо коеф. жорсткості пружинок
>> n=1:2*N;
>> xv(n)=0;      % початкові координати та швидкості
>> xv(1)=0.7;    % координата першого осцилятора
>> Tfin=50*pi;   % задаємо часовий інтервал
>> NumPoint=2^14-1;
                 % кількість точок часового інтервалу
>> [T,M]=ode45('oscilr',[0:Tfin/NumPoint:Tfin],xv);
>> E=Energ(N,m,k,M);
     % Масив E має розмірність [(NumPoints) x (N)] і
```

```
     % містить величину енергії кожного з
     % осциляторів у задані моменти часу
>> for i=1:NumPoint+1
     s=E(i,:);   % s - одновимірний масив, що містить
                 % енергії  усіх  осциляторів  у  i-й
                 % момент часу
     Efull(i)=sum(s); % Знаходимо суму за усіма
                      % осциляторами у i-й момент часу
>> end;
>> subplot(1,2,1);
>> plot(T,E(:,1),T,E(:,5),T,E(:,15))   % Виводимо на екран
   % енергію 1-го, 5-го та 15-го осциляторів в усіх точках
   % часового інтервалу
>> axis([0 30 0 0.25]);
>> subplot(1,2,2);
>> plot(T,Efull)
   % Виводимо на екран повну енергію системи в усіх точках
   % часового інтервалу
```

Моделювання поширення хвилі (до розділу 6)

```
const N=140;
float t=0, dt=0.01, dy=1, A=4, omega=1, V_f=2;
float v[N],x[N],x_bak[N];
void Init()
{
  for(int i=0; i<N-1; i++)
   {
     x[i]=0;
     v[i]=0;
   };
  x[0]=A*sin(omega*t);
  ...
     // Заповнюємо масив x_bak шляхом копіювання в нього
```

```
     // значень елементів з масиву x
  ...
};
void Out()
{
  for(int i=1; i<N; i++)
   {
    ...
       // Видаляємо  попередній рисунок,  для чого
       // кольором фону проводимо лінію  від точки
       // (i*4-4,240-ceil(x_bak[i-1]*30)) до точки
       // (i*4,240-ceil(x_bak[i]*30)).
       // Встановлюємо новий колір для   виведення
       // оновленої ділянки   лінії  від     точки
       // (i*4-4,240-ceil(x[i-1]*30)) до     точки
       // (i*4,240-ceil(x[i]*30))
    ...
   };
};
void main()
{
  Init();
  while (!kbhit())
   {
    t=t+dt;
    for(int i=1; i<N-1; i++)
     v[i]=v[i]+V_f*V_f*(x[i+1]-2*x[i]+x[i-1])/(dy*dy)*dt;
    for(i=1; i<N-1; i++)
        x[i]=x[i]+v[i]*dt;
       // Ліва точка здійснює   два   повних коливання
       // (t=2T=2*2*pi /omega),  а  потім  перебуває у
       // нерухомому стані.
    if (t<2*2*3.14/omega)
      x[0]=A*sin(omega*t);
```

```
     else
       x[0]=0;
    x[N-1]=0; // Остання точка пружного середовища є
              // закріпленою
    Out();
    ...
      // Заповнюємо масив x_bak шляхом копіювання в нього
      // значень елементів з масиву x
    ...
   }
};
```

Моделювання відображення "зуб пили"(до розділу 7)

```
...
double frac(double y);
void main()
  {
   double x=1/sqrt(80);
   int i = 0, N = 30;
   ...       // i та x зберігаємо у файлі
   for (i=1; i<=N; i++)
     {
       x = frac(2*x);
       ...  // i та x зберігаємо у файлі
     }
   ...
  }
double frac(double y)
  {
   double w;
   w = y - floor(y);
   return w;
  }
```

Логістичне відображення (до розділу 7)

```
   // Порівняння наближених і точних значень логістичного
   // відображення
...
typedef float realtype;
void main()
  {
   realtype x0=1/8., x=x0, x_ex, abs_err=0, rel_err=0;
   int i = 0, N = 60;
   ...   // записуємо у файл i, x, x_ex, abs_err, rel_err
   for (i=1; i<=N; i++)
{
     x = 4*x*(1-x);
     x_ex = 1/2.*(1-cos(pow(2,i)*acos(1-2*x0)));
     abs_err = fabs(x-x_ex);
     rel_err = abs_err/x_ex*100;
     ...
        // записуємо у файл i, x, x_ex, abs_err, rel_err
}
...
}

  // Знаходження показника Ляпунова для логістичного
  // відображення
...
double f_(double y);
void main()
 {
  double x=0.1, s=0;
  int i = 0, N = 100;
  s += log(fabs(f_(x)));
  for (i=1; i<=N-1; i++)
   {
    x = 4*x*(1-x);
```

```
    s += log(fabs(f_(x)));
   }
  s = s/N;
  ... // виводимо s  на екран
 }
double f_(double y)
 {
  double z;
  z = 4-8*y;
  return z;
 }
```

Система Лоренца (до розділу 7)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Програма складається із двох функцій. Головної - %
% функція lorenz та допоміжної lor, яка містить праву   %
% частину модельних рівнянь системи                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function lorenz
global sigma r b
   % Задаємо значення параметрів системи
sigma = 10;
r = 28.0;
b = 8.0/3.0;
FunFcn='lor';
   % Випадковим чином вибираємо початкові умови
y0(1)=rand*30+5;
y0(2)=rand*35-30;
y0(3)=rand*40-5;
t0=0;
t = t0;
y = y0(:);
tol = 0.001; % Точність розв'язку
```

```
tf=200;       % Час моделювання
pow = 1/3;    % Параметр оцінки кроку за часом (див. розд 7)
hmax = (tf-t)/10;   % Максимально допустимий крок за часом
h = (tf - t)/1000;  % Поточний крок за часом
cla;
handle = plot3(y(1),y(2),y(3),'Marker','.', 'markersize',...
                                          1,'erase','none');
axis([-10, 10,-10, 10, 0, 60]);
xlabel('X');
ylabel('Y');
zlabel('Z');
while t<tf
     % Розраховуємо коефіцієнти згідно з методом Рунге-Кутта
   k1 = feval(FunFcn, t, y);
   k2 = feval(FunFcn, t+h, y+h*k1);
   k3 = feval(FunFcn, t+h/2, y+h*(k1+k2)/4);
     % Оцінюємо похибку
   delta = max(abs((h*(k1 - 2*k3 + k2)/3)));
     % Розраховуємо припустиму похибку
   tau = tol*max(max(abs(y)),1.0);
     % Розраховуємо нові координати
   if delta <= tau
       t = t + h;
       y = y + h*(k1 + 4*k3 + k2)/6;
       set(handle,'xdata',y(1),'ydata',y(2),'zdata',y(3))
       drawnow;
   end;
   if delta ~= 0.0
     % Розраховуємо оптимальний крок за часом
     h = min(hmax, 0.9*h*(tau/delta)^pow);
   end
end

function yd = lor(t,y)
```

```
global sigma r b
yd = [sigma*(y(2)-y(1)); r*y(1)-y(2)-y(1)*y(3); ...
                                  -b*y(3)+y(1)*y(2)];
```

Розрахунок показника Ляпунова (до розділу 7)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Розрахунок показника Ляпунова для системи Лоренца.  %
% Функція lor, яка містить праву частину  модельних   %
% рівнянь системи                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function lyapunov
global sigma r b
    % Задаємо значення параметрів системи
sigma = 10;
r = 28.0;
b = 8.0/3.0;
    % Час експерименту - 200;
    % час виходу системи на атрактор - 100;
    % часовий інтервал розрахунку на одній
    % ітерації - 5
time=[100 5 200];
p=0.5;  % Початкове збурення
L=0;
N=0;
   % Випадковим чином вибираємо початкові умови
Y(1)=rand*30+5;
Y(2)=rand*35-30;
Y(3)=rand*40-5;
[T,Y]=ode23s(@lor,[0,time(1)],Y);
Yend=Y(end,:);
for i=time(1):time(2):time(3)-time(2),
       % Розрахунок вихідної траєкторії на одній
       % ітерації
```

```
    [T,Y]=ode23s(@lor,[i,i+time(2)],Yend);
       % Розрахунок збуреної траєкторії
    [T,dY]=ode23s(@lor,[i,i+time(2)],Yend+p);
    Delta=Y(end,:)-dY(end,:);
       % Розрахунок норми
    norma=norm(Delta,2);
    L=L+log(norma/p);
    Yend=Y(end,:);
    N=N+1;
end;
       % Розраховуємо показник Ляпунова
L=L/N

function yd = lor(t,y)
global sigma r b
yd = [sigma*(y(2)-y(1)); r*y(1)-y(2)-y(1)*y(3); ...
                                    -b*y(3)+y(1)*y(2)];
```

Навчальне видання

Князь Ігор Олександрович
Вітренко Андрій Миколайович

Комп'ютерне моделювання динамічних систем

Розділ
"МОДЕЛЮВАННЯ ФІЗИЧНИХ СИСТЕМ"

Навчальний посібник

Дизайн обкладинки А. М. Вітренко
Редактор Н. А. Гавриленко
Комп'ютерне верстання І. О. Князя

Формат 60×84/16. Ум. друк. арк. 8,37. Обл.-вид. арк. 7,14. Тираж 40 прим. Зам. №

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб'єкта видавничої справи ДК № 3062 від 17.12.2007.