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.

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.

Теоретичний матеріал

"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

$$U(r_{12}) = 4\varepsilon \left\{ \left(\frac{\sigma}{r_{12}} \right)^{12} - \left(\frac{\sigma}{r_{12}} \right)^{6} \right\}, \qquad (1)$$

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:

$$\overrightarrow{F}(|\overrightarrow{x}_i - \overrightarrow{x}_j|) = -\frac{\partial}{\partial \overrightarrow{x}_i} U(|\overrightarrow{x}_i - \overrightarrow{x}_j|). \qquad (2)$$

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:

$$\begin{aligned} \frac{d\overrightarrow{x}_i}{dt} &= \overrightarrow{v}_i, \\ m_i\frac{d\overrightarrow{v}_i}{dt} &= \overrightarrow{F}_i + \sum_{j=1,j\neq i}^{N} F(m_i, m_j, |\overrightarrow{x}_i - \overrightarrow{x}_j|)\frac{(\overrightarrow{x}_i - \overrightarrow{x}_j)}{|\overrightarrow{x}_i - \overrightarrow{x}_j|}, \quad (3) \end{aligned}$$

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:

$$\begin{aligned} \overrightarrow{x}_i^{\,n} &= \overrightarrow{x}_i^{\,n-2} + \overrightarrow{v}_i^{\,n-1} 2\Delta t, \\ \overrightarrow{v}_i^{\,n+1} &= \overrightarrow{v}_i^{\,n-1} + \frac{1}{m_i}\Big[\overrightarrow{F}_i(m_i, \overrightarrow{x}_i^{\,n}) + \quad (4) \\ &\quad + \sum_{j=1,j\neq i}^{N} F(m_i, m_j, |\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n}|)\frac{(\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n})}{|\overrightarrow{x}_i^{\,n} - \overrightarrow{x}_j^{\,n}|}\Big] 2\Delta t. \end{aligned}$$

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:

$$\frac{1}{n} = \frac{4\pi}{3}a^3 = \frac{V}{N}. \qquad (5)$$

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

$$\Delta t << \frac{a}{|\overrightarrow{v}|}. \qquad (6)$$

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

$$E = \frac{d}{2}kT + \overline{U}, \tag{7}$$

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:

$$\frac{d}{2}kT = \frac{1}{N}\frac{1}{K}\sum_{n=1}^{K}\sum_{i=1}^{N}\frac{1}{2}m(v_i^n)^2 \tag{8}$$

й аналогічно усереднимо повну потенціальну енергію:

$$\overline{U} = \frac{1}{2N}\frac{1}{K}\sum_{n=1}^{K}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}U(|\overrightarrow{x}_i^n - \overrightarrow{x}_j^n|). \tag{9}$$

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

$$(\overrightarrow{r_i} + \overrightarrow{v_i}t' - \overrightarrow{r_j} - \overrightarrow{v_j}t')^2 = 4R^2. \qquad (10)$$

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

Зміна швидкості після зіткнення розраховується згідно з формулою

$$\overrightarrow{v_i} = \overrightarrow{v_i} - \frac{m_j}{m_i + m_j}\triangle\overrightarrow{v}, \qquad \overrightarrow{v_j} = \overrightarrow{v_j} + \frac{m_i}{m_i + m_j}\triangle\overrightarrow{v}, \qquad (11)$$

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.

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:

$$\begin{aligned} r_x &= x_i(t)-x_j(t), \\ r_y &= y_i(t)-y_j(t), \\ r &= \sqrt{r_x^2+r_y^2}. \end{aligned} \qquad (12)$$

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:

$$\begin{aligned} S_x &= S_x + F(m_i, m_j, r)\frac{r_x}{r}, \\ S_y &= S_y + F(m_i, m_j, r)\frac{r_y}{r}, \\ a_{xj} &= a_{xj} - F(m_i, m_j, r)\frac{r_x}{r}, \\ a_{yj} &= a_{yj} - F(m_i, m_j, r)\frac{r_x}{r}. \end{aligned} \tag{13}$$

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:

$$\begin{aligned} v_{xi}\left(t + \frac{1}{2}\triangle t\right) &= v_{xi}(t) \quad + \quad (S_x + a_{xi})\frac{1}{2}\triangle t, \\ v_{yi}\left(t + \frac{1}{2}\triangle t\right) &= v_{yi}(t) \quad + \quad (S_y + a_{yi})\frac{1}{2}\triangle t. \end{aligned} \tag{14}$$

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

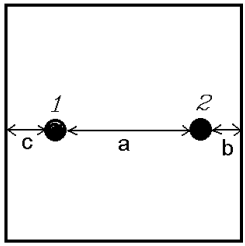

Рисунок 1 — Мінімальна відстань між частинками 1 та 2 становить $r_x = b + c, r_x < a$ через періодичність прямокутного резервуара

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:

$$\begin{aligned} x_i(t+\triangle t) &= x_i(t) + v_{xi}\left(t+\frac{1}{2}\triangle t\right)\triangle t, \\ y_i(t+\triangle t) &= y_i(t) + v_{yi}\left(t+\frac{1}{2}\triangle t\right)\triangle t. \end{aligned} \qquad (15)$$

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.

Приклади

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 $T \approx 240$.

"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?

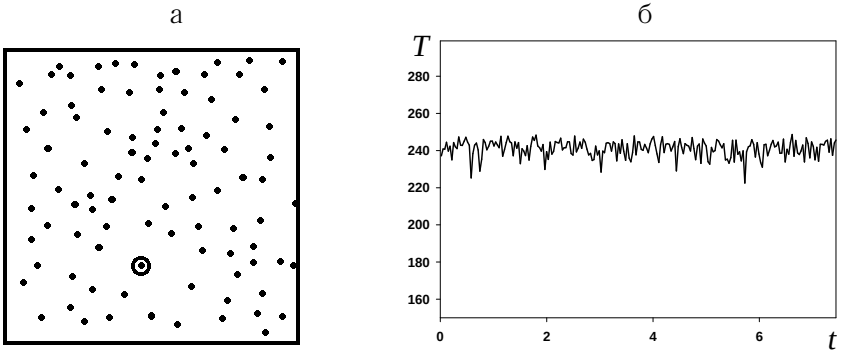

Рисунок 2 — Чисельне моделювання руху N взаємодіючих частинок з потенціалом міжмолекулярної взаємодії Леннарда-Джонса:
а) візуалізація руху частинок у періодичному резервуарі;
б) експериментально отримана залежність температури T від часу t

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).

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;
```