

*Зіновчук Андрій,
кандидат фізико-математичних наук, доцент,
доцент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна*

КОМП'ЮТЕРНА МОДЕЛЬ СТАТИСТИЧНОЇ СИСТЕМИ БАГАТЬОХ ЧАСТИНОК З ВЗАЄМОДІЄЮ

Більшість реальних фізичних систем складаються з дуже великої кількості (порядку $N \approx 10^{23}$) взаємодіючих частинок, які знаходяться в стані постійного руху. Фізичні особливості взаємодії визначають характеристики руху частинок, а це, в свою чергу, визначає всі макроскопічні властивості фізичних систем. У зв'язку з цим виникає закономірне питання як вирішити задачу про опис поведінки складної багаточасткової системи, тобто як зв'язати мікроскопічні характеристики руху частинок з макроскопічними властивостями цілої

статистичної системи. Така задача не може бути розв'язана аналітично навіть у випадку невзаємодіючих частинок (модель ідеального газу) з огляду на дуже велику кількість ступенів вільності (координат та швидкостей частинок) [1]. Тільки підходи, орієнтовані на комп'ютерне моделювання, можуть дати результати, що адекватно відповідають об'єктивній фізичній реальності. При цьому, найбільший інтерес становлять не самі траєкторії руху частинок, а обчислювані з їх використанням макроскопічні характеристики досліджуваних систем. Відповідну групу комп'ютерних моделей часто називають методом молекулярної динаміки (МД) [2,3]. Слід зазначити, що обчислювальні можливості навіть сучасних CPU обмежують метод МД за кількістю частинок моделі. Тому комп'ютерні моделі такого типу обов'язково треба перевіряти на предмет валідності відносно кількості частинок.

У цій роботі ми розглядаємо модель, що базується на чисельному розв'язанні мікроскопічних рівнянь руху частинок з довільним виглядом потенціалу взаємодії. Модель не враховує квантово-механічні ефекти, тому стан частинки повністю визначається координатами та швидкостями, а рівняння руху мали класичну ньютонівську форму. Для чисельного розв'язання рівнянь руху використовувався модифікований алгоритм Верле з підвищеним порядком точності [4]. Програмна реалізація моделі була побудована на мові Python використанням бібліотек Numpy та Matplotlib. Дослідження показали, що вже при досить невеликих числах частинок $N \approx 300$ відносна похибка результатів моделювання макроскопічних характеристик системи не перевищує 10 %. Це дає можливість використовувати нашу модель для адекватних фізичних оцінок поведінки складних багаточастинкових систем за допомогою одного (без використання паралельних обчислень) CPU навіть середньої обчислювальної потужності.

Математична модель

Якщо вважати, що сила взаємодії двох частинок залежить тільки від відстані між ними, то потенціальна енергія взаємодії U визначається сумою енергій

двохчастинкової взаємодій $U = \sum_{i < j}^N V(r_{ij})$, де $V(r_{ij})$ залежить лише від відстані між i та j частинками $V(r_{ij}) = V(|\vec{r}_i - \vec{r}_j|)$. Наша модель дозволяє застосування довільного виду двохчастинкової взаємодії $V(r_{ij})$, однак, в цій роботі ми слідуємо найбільш часто застосовуваному вибору потенціалу Ленарда-Джонса

$$V(r_{ij}) = 4V_0 \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right) \quad (1)$$

де числові параметри V_0 і σ фізично означають енергетичну глибину мінімуму потенціалу та ефективний діаметр частинки.

Таким чином, при використанні моделі двохчастинкової взаємодії задача опису поведінки статистичної системи зводиться до інтегрування системи з N диференціальних рівнянь у ньютонівій формі

$$m \frac{d^2 \vec{r}_i}{dt^2} = -\vec{\nabla} \sum_{i < j}^N V(r_{ij}) \quad (2)$$

де m – маса частинок, а $\vec{\nabla}$ – градієнт по координатах j частинки.

Для однозначності розв’язків система рівнянь (2) має бути доповнена початковими умовами:

$$\begin{aligned} \vec{r}_i(t_0) &= \vec{r}_{i,init} \quad (i = 1 \dots N) \quad (\text{початкові положення}) \\ \vec{v}_i(t_0) &= \frac{d\vec{r}_i}{dt}(t_0) = \vec{v}_{i,init} \quad (i = 1 \dots N) \quad (\text{початкові швидкості}) \end{aligned}$$

Набір початкових даних $\vec{r}_{i,init}$ задавався за допомогою генератора псевдовипадкових чисел. Щодо набору початкових швидкостей $\vec{v}_{i,init}$, то генерація відбувалася таки чином, щоб середнє значення сумарної кінетичної енергії було рівним $\langle E \rangle = (3/2)kT$, де k – числова константа, яка впливає на розмірність фізичних одиниць, а T – задана температура системи. У фізиці константа k називається сталою Больцмана і має порядок $\sim 10^{-23}$. Однак, для того щоб запобігти неконтрольованим помилкам обчислень, пов’язаним із виходом за межі діапазону типів даних, ми приймали $k=1$ та проводили моделювання у відносних фізичних одиницях. Система (2) разом із початковими умовами є замкнутою та однозначною математичною моделлю розглядуваної статистичної системи.

Алгоритм

Після складання математичної моделі системи, слід вибрати чисельний алгоритм, від вибору якого залежить як точність так і ефективність (швидкість) отримання розв’язків. Контроль за стійкістю чисельного розв’язку можна здійснювати, шляхом обчислення на кожному кроці алгоритму повної енергії системи, величина якої в ідеальному випадку має залишатися постійною. Використання алгоритмів низького порядку точності, таких як алгоритм Ейлера та алгоритм Ейлера-Кромера [5] показало, що дані алгоритми не можуть забезпечити збереження енергії на часових інтервалах. Тому доводиться застосовувати обчислювальні алгоритми, що мають більш високий порядок точності, одним із яких є алгоритм Верле. Цей алгоритм ще називається неявною симетричною різницевою схемою. Він зводиться до двох рекурентних співвідношень

$$\begin{aligned} \vec{r}_{i,n+1} &= 2\vec{r}_{i,n} - \vec{r}_{i,n-1} + \vec{a}_{i,n} (\Delta t)^2 \\ \vec{v}_{i,n+1} &= (\vec{r}_{i,n+1} - \vec{r}_{i,n-1}) / (2\Delta t) \end{aligned} \quad (3)$$

де Δt – інтервал часової дискредитації, $\vec{r}_{i,n}$ та $\vec{v}_{i,n}$ – положення і швидкості частинок в момент часу $t_n = t_0 + n \cdot \Delta t$, ($n = 1 \dots \infty$), де n – номер ітерації. Зважаючи на те, що за визначенням прискорення частинок є $\vec{a} = d^2 \vec{r} / dt^2$, із (2) отримаємо

$$\vec{a}_{i,n} = -\frac{1}{m} \vec{\nabla} \sum_{i < j}^N V(|\vec{r}_{i,n} - \vec{r}_{j,n}|) \quad (4)$$

Градiєнт $\vec{\nabla}$ розраховувався за допомогою функції *numpy.gradient* із бібліотеки Numpy [6].

Як видно із (3)-(4), недолік оригінального алгоритму Верле полягає в тому, що він не є самостартуючим і доводиться використовувати додаткові обчислення і фізичні допущення для отримання набору даних першої ітерації ($n=1$), після якої вже починає працювати різницева схема (3)-(4). Для обходу згаданого недоліку, ми використали математичний аналог рекурентних співвідношень (3), у вигляді

$$\begin{aligned} \vec{r}_{i,n+1} &= \vec{r}_{i,n} + \vec{v}_{i,n} \Delta t + \frac{1}{2} \vec{a}_{i,n} (\Delta t)^2 \\ \vec{v}_{i,n+1} &= \vec{v}_{i,n} + \vec{a}_{i,n} \Delta t \end{aligned} \quad (5)$$

Як видно із (5) дана схема є самостартуючою і не вимагає додаткових обчислень.

В макросистемах із відносно слабою взаємодією, частинки мають можливість зміщуватися на великі макроскопічні відстані. Тому, під час моделювання, традиційно вважається, що частинки знаходяться в деякій обмеженій області простору. В цій роботі ми розглядали систему в кубічній комірці з лінійним розміром L та об'ємом $\text{Vol}=L^3$. Така постановка забороняє частинкам проникати за межі комірки. Частинки, відбиті від граней межі, будуть повертатися всередину, тому грані будуть вносити значний внесок в макроскопічні характеристики системи, особливо для систем з малою кількістю частинок. Це в свою чергу буде негативно впливати на релевантність отриманих результатів моделювання. Для зменшення описуваного ефекту ми ввели періодичні граничні умови. З точки зору математичної моделі, це означає що фізичні величини стають періодичним функціями координат у всіх трьох напрямках з періодами, що рівні лінійній довжині комірки L . За таких граничних умов потенціальна енергія системи доповниться додатковими додатками, що являтимуть собою взаємодію частинок всередині комірки з частинками в всіх "періодично відображених" комірках.

$$U = \sum_{i < j}^N V(|\vec{r}_i - \vec{r}_j|) + \sum_{\vec{l}} \sum_{i < j}^N V(|\vec{r}_i - \vec{r}_j + \vec{l}|) \quad (6)$$

В другому додатку сумування проводиться по безкінечній кількості векторів періодичності $\vec{l} = (l_1 L; l_3 L; l_2 L)$, що визначаються цілими числами (l_1, l_3, l_2) . Для того, щоб позбавитися від безкінечної суми і оптимізувати ефективність обчислень ми залишили лише сумування по тих "відображених" частинках, що знаходяться на відстані не більше ніж лінійна довжина комірки. Зважаючи на те, що реальні міжатомні та міжмолекулярні сили переважно мають короткодiючих характер, це наближення не повинно сильно впливати на фізичну релевантність отриманих результатів моделювання.

На останньому етапі алгоритму в кожній ітерації розраховуються макроскопічні параметри системи, як середні статистичні величини по наборах координат і швидкостей. В даній роботі таким параметрам були тиск та температура системи.

Остаточні етапи алгоритму можна сформулювати наступним чином: 1) Задати кількість частинок системи N .

2) Згенерувати початкову конфігурацію системи (набір початкових даних $\vec{r}_{i,init}, \vec{v}_{i,init}$).

3) Встановити інтервал часової дискретизації Δt .

4) Задати число ітерацій, у яких обчислюються розв'язки системи (2).

5) Обчислення потенціальної енергії (6) та її градієнту.

6) Запуск різницевої схеми (5) та отриманні набору даних для наступної ітерації.

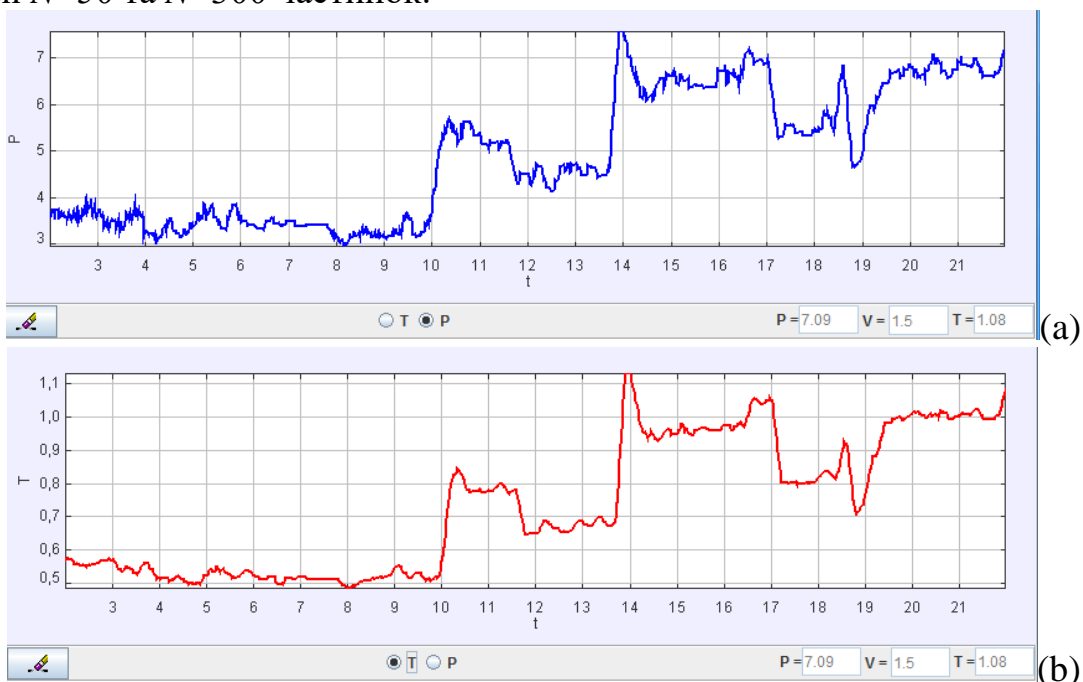
7) Обчислення повної енергії системи

8) Якщо різниця між повними енергіями системи в двох послідовних ітераціях більша за критично допустиме значення, то аварійно завершити виконання алгоритму. В іншому випадку, перейти до обчислення макроскопічних характеристик системи.

Результат

Як вже зазначалося описана математична модель програмно була реалізована на мові Python з бібліотекою Numpy. Контроль стабільності роботи алгоритму в процесі виконання ітерацій забезпечувався за допомогою графічної візуалізації засобами бібліотеки Matplotlib. Числові параметри потенціалу двохчастинкової взаємодії (1) бралися близькими до реальних значень молекули азоту: $V_0=0,01$ еВ та $\sigma=3,5$ А. Програмна реалізація моделі запускалася на виконання на одноядерному 2.2 ГГц CPU.

На рис.1 показані часова розгортка миттєвих значень тиску та температури в системі $N=50$ та $N=300$ частинок.



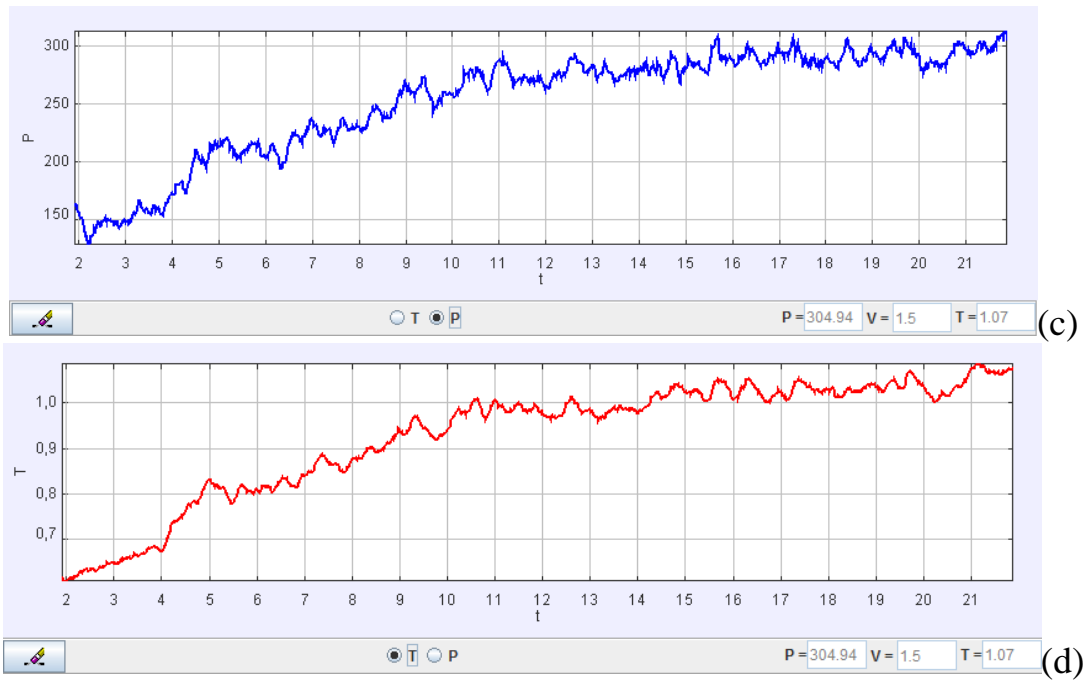


Рис. 1 Часова розгортка миттєвих значень тиску (а, с) та температури (b, d) в системі з $N=50$ (а, b) та $N=300$ (с, d) частинок.

Як видно при $N=50$ миттєві значення хаотично змінюються з великою амплітудою. Середнє квадратичне відхилення в цьому випадку є занадто великим щоб таку модель можна було використовувати для релевантних оцінок. Необхідно збільшувати число частинок. Звичайно, при цьому збільшиться і розрахункове навантаження на CPU. Алгоритмічний клас складності нашої моделі є N^3 (належить до групи відносно швидких алгоритмів), однак для розрахунку динаміки (часової еволюції) системи складність збільшується в таку кількість разів, яка рівна кількості інтервалів часової дискретизації, на яких проводиться розрахунок. Тому, за прийнятний час розрахунку, для занадто великих багаточастинкових систем, стандартні обчислювальні пристрої дозволяють отримати лише початкові етапи динаміки, що не є достатнім для актуальних фізичних оцінок. Однак, як видно із рис.1 вже при $N=300$ середнє квадратичне відхилення не перевищує 10 % від середнього значення макроскопічних параметрів, що цілком задовільною величиною для використання такої моделі у обчислювальних експериментах.

Як підсумок, відзначимо, що складена нами математична модель динаміки статистичної системи та її програмна реалізація демонструє задовільні характеристики, як по швидкості обчислень, так і по відповідності отриманих результатів об'єктивним фізичним даним. Вона може бути використана в обчислювальних експериментах для отримання нових та перевірки вже відомих фактів про фізичні процеси і явища статистичної фізики та фізичної кінетики. Також модель може бути інкорпорована, як складова частина, більш загального симулятора фізичних процесів в макросистемах.

Список використаних джерел та літератури

1. R. K. Pathria and P. D. Beale, *Statistical Mechanics*, Academic Press, 2011.
2. D. C. Rapaport, *The Art of Molecular Dynamics Simulation*, Cambridge

University Press, 2004

3. D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Academic Press, San Diego, 2002.

4. L. Verlet, *Computer “experiments on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules*, Phys. Rev., 159, 98–103, 1967.

5. D. Vasileska, S. M. Goodnick, G. Klimeck, *Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation*.

6. NumPy. The fundamental package for scientific computing with Python: URL: <https://numpy.org>.