

Использование кольцевой очереди в дискретно-событийном методе молекулярной динамики

*Национальный аэрокосмический университет им. Н. Е. Жуковского
«Харьковский авиационный институт»*

Рассмотрен дискретно-событийный метод молекулярной динамики. Проведен анализ эффективности работы основных этапов событийного алгоритма. Предложено использовать алгоритм ящичной сортировки для управления очередью событий, что дает возможность уменьшить затраты вычислительного времени. Описана методика преобразования случайной величины (времени наступления события) в случайную величину с равномерной функцией распределения, что позволяет осуществлять равномерное заполнение очередей в ящиках. Приведен алгоритм ящичной сортировки с использованием кольцевой очереди.

Ключевые слова: событийное моделирование, кольцевая очередь, ящичная сортировка, равномерное распределение, двоичное дерево, стек.

Введение

Компьютерная молекулярная динамика (МД) является одним из наиболее мощных вычислительных методов, эффективно применяемых для моделирования физических, химических и биологических систем [1 – 3]. Этот метод предусматривает расчет на ЭВМ эволюции системы моделей молекул или частиц с заданным законом взаимодействия.

Существуют два подхода к моделированию методом молекулярной динамики:

1. Дискретно-временное моделирование [4], в котором систему уравнений, описывающую движение частиц, решают численно, придавая основному параметру, которым является время, малые приращения. Новые координаты и скорости частиц вычисляют в конце каждого временного интервала и используют в качестве входных данных для вычислений на следующем интервале. Количество модельных объектов при использовании дискретно-временных алгоритмов имеет порядок $10^3 - 10^5$ даже при наличии самых совершенных компьютеров.

2. Дискретно-событийное моделирование [5, 6], в котором динамика системы управляется последовательностью различных событий - взаимодействий объектов. Эти события могут изменять свойства частиц (в частности, скорости). Между событиями частицы двигаются равномерно и прямолинейно.

В данной работе рассматривается дискретно-событийный метод (ДСМ) молекулярной динамики, который работает на несколько порядков быстрее, чем дискретно-временной.

В событийном моделировании выделяются три основные задачи, которые занимают большую часть вычислительного времени: поиск событий (обнаружение событий), управление очередью событий и обработка событий.

Основные шаги ДСМ для системы из N частиц следующие:

1. Тестирование событий $O(N^2)$. Для каждой частицы определяются тип и момент времени наступления следующего события. Рассчитанные моменты времени вставляются в очередь событий.

2. Сортировка событий $O(N \log N)$. События в очереди сортируются в порядке возрастания моментов времени наступления для определения ближайшего события, которое должно произойти.

3. Движение системы $O(N)$. Система эволюционирует до момента времени следующего события.

4. Обработка события $O(1)$. Для частиц, участвующих в событии, определяются новые скорости.

5. Обновление очереди событий $O(\log N)$. Для частиц, участвующих в событии, определяются моменты времени наступления следующих событий. Рассчитанные моменты времени вставляются в очередь.

6. Условие остановки алгоритма. Возвращаемся к шагу 2, пока не будет достигнуто необходимое время моделирования.

Пионерской работой этого направления является статья Б. Олдера и Т. Вэйнрайта [7]. С момента появления этой работы были достигнуты значительные успехи в развитии исходного алгоритма для уменьшения общего количества вычислений.

Б. Олдер и Т. Вэйнрайт [8] были первыми, кто предложил использование "списка соседей", техники для повышения эффективности поиска будущих событий. В этом методе вводятся в рассмотрение подсистемы исходной системы модельных частиц. В тестировании возможного события столкновения с частицей участвуют только те частицы, которые находятся в одной подсистеме. При этом вводятся новые события, связанные с переходом частицы из одной подсистемы в другую. В результате вычислительная сложность шагов 1 и 5 уменьшается до $O(1)$.

Еще одним важным улучшением является то, что перемещаются только те частицы, которые участвуют в событии [9] (шаг 3). Каждая частица хранит «локальное время» (момент времени наступления очередного события, в котором она принимает участие) и «глобальное время» (текущее время эволюции системы). Это приводит к уменьшению сложности шага 3 до $O(1)$.

Большое количество вычислительного времени тратится на управление очередью событий (шаг 2). Эта очередь должна быть отсортирована для определения следующего события и перестроена после вставки в нее нового события. Для подобных ситуаций используются различные методы реализации «очередей с приоритетами» [6, 10 – 12].

В данной работе предлагается использовать алгоритм карманной (ящичной) сортировки (bucket sort) [13]. Для обеспечения времени работы алгоритма в среднем случае, равного $O(N)$, на входные данные необходимо наложить условие подчинения равномерному закону распределения.

В рассматриваемом алгоритме сортируемые объекты распределяются между конечным количеством ящиков таким образом, чтобы в каждом последующем ящике все элементы были больше, чем в предыдущем. Поскольку объекты равномерно распределены, то в каждый ящик попадает одинаковое количество элементов.

В реальности функция распределения моментов времени наступления событий в молекулярной системе не известна.

В случае моделирования одноатомного газа выделяются два типа событий: столкновение модельных частиц между собой и пересечение центрами модель-

ных частиц граней подсистем, в которых они находятся. Функция распределения моментов времени наступления событий первого типа является следствием деления двух случайных величин длины свободного пробега на скорость. Длина свободного пробега имеет экспоненциальное распределение, скорость - распределение Максвелла (или Рэлея в двумерном случае). Второй тип событий возникает вследствие необходимости декомпозиции системы [6].

При моделировании молекулярных систем необходимо учитывать силы взаимодействия между атомами в молекуле. Для этого каждая частица представляется в виде пары концентрических сфер [6, 14] с различными радиусами. Внутренняя сфера ответственна за процесс отталкивания частиц, внешняя - притягивания. В результате возникает еще один процесс - колебание связанных частиц относительно их центра масс. Функция распределения этого процесса также не известна. Для ее описания необходимо учитывать не только расстояние между частицами, радиусы и скорость, но и направление вектора скорости.

В данной работе рассматриваются методы преодоления данной трудности для одноатомных моделей.

1 Преобразование случайной величины (времени наступления события) в случайную величину с равномерной функцией распределения

Предположим, что случайная величина τ имеет распределение, описываемое интегральной функцией распределения $F(\tau) = P(\tau < x)$. Тогда случайная величина $\eta = F_\tau(\tau)$ равномерно распределена на отрезке $[0, 1]$ [15]. Это открывает возможность для эффективного использования алгоритма ящичной сортировки за счет преобразования случайной величины (времени наступления события) в случайную величину с равномерной функцией распределения. Действительно, согласно алгоритму ящичной сортировки отрезок $[0, 1]$ разбивается на n

подинтервалов одинаковой длины $h = \frac{1}{n}$. Реализации случайной величины η сопоставляется номер ящика i как целая часть произведения: $i = [\eta \cdot n]$. Возникает семейство очередей, образованных в каждом отдельном ящике. В среднем при равномерном распределении длина очереди в отдельном ящике равна единице. Это и определяет то, что алгоритм вставки и изъятия имеет порядок $O(1)$.

Рассмотрим последовательность реализаций τ_i случайной величины τ , определяемой как разность случайной величины t (время наступления события) и момента осуществления последнего из событий, обозначенного как глобальное время t_{global} :

$$\tau = t - t_{global}. \quad (1)$$

Случайная величина τ в формуле (1) является минимумом двух случайных величин: τ_{bord} - времени достижения центром модельной частицы границы подобласти, которой частица принадлежит, и τ_{part} , наименьшего из моментов времени до взаимодействия данной частицы с остальными: $\tau = \min(\tau_{bord}, \tau_{part})$. На рис.1 приведен один из результатов построения графиков плотности распределения для каждой из этих случайных величин.

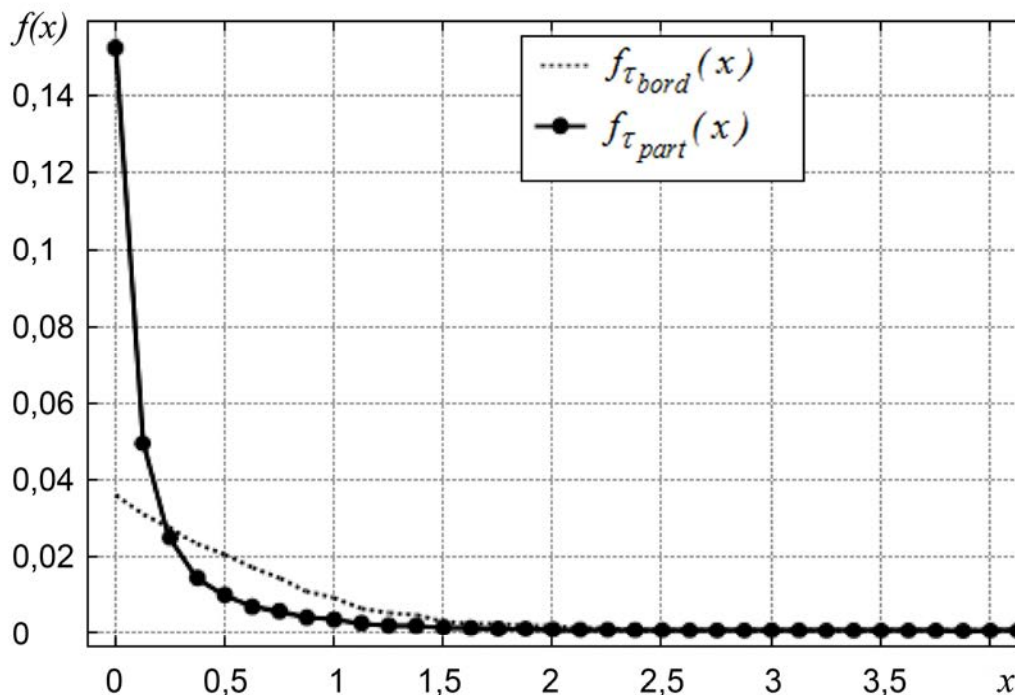


Рис. 1. Графики плотности распределения случайных величин τ_{bord} , τ_{part}

Реальная функция распределения, как правило, не известна и может быть восстановлена лишь эмпирически. Тем не менее практика применения дискретно-событийного моделирования движения модельных частиц показывает, что такие распределения близки к экспоненциальным.

Для перевода в равномерное распределение с последующим использованием алгоритма ящичной сортировки приходится аппроксимировать эмпирическую функцию распределения. Кусочно-линейная аппроксимация, как и приближение сплайн-функциями, достаточна сложна и не приводит к приемлемым результатам из-за большой величины производной в начале координат. В результате длины очередей в ящиках с начальными номерами оказываются слишком большими. Поэтому предлагается использовать простое представление эмпирической функции распределения в виде, точном для экспоненциального:

$$\eta = 1 - \exp\left(-\frac{\tau}{T}\right). \quad (2)$$

Вопрос заключается в выборе такого значения усредненного параметра T , при котором распределение случайной величины η будет в наибольшей степени близким к равномерному. Для оценивания оптимального значения параметра T проведена серия численных экспериментов. Пользуясь полученными данными, удастся приблизить распределение моментов событий к равномерному с той или иной погрешностью.

Таким образом, каждому моменту t_i наступления события с номером i после обработки этого события в момент времени t_{global} ставится в соответствие

число $\eta_i = 1 - \exp(-\frac{t_i - t_{global}}{T})$, для которого легко найти соответствующий номер ящика, причем порядок наступления событий не нарушается.

Описанная методика позволяет удовлетворительно осуществить заполнение очередей в ящиках с начальными номерами. Но численные эксперименты показывают, что большинство событий скапливаются в последних ящиках, оставляя слабо заполненными первые.

2 Использование кольцевой очереди в ящичном алгоритме сортировки

Для устранения вышеперечисленных недостатков предлагается все события, которые превышают максимальное значение, помещать в начало очереди. Пусть в очереди имеется m ящиков. Для поиска номера ящика, в который необходимо поместить событие, необходимо время события поделить на длину ящика. Если полученное значение больше m , то для определения номера ящика предлагается выполнить операцию деления по модулю m от полученного значения. Графически это можно представить в виде кольцевого связанного списка ящиков (рис. 2).

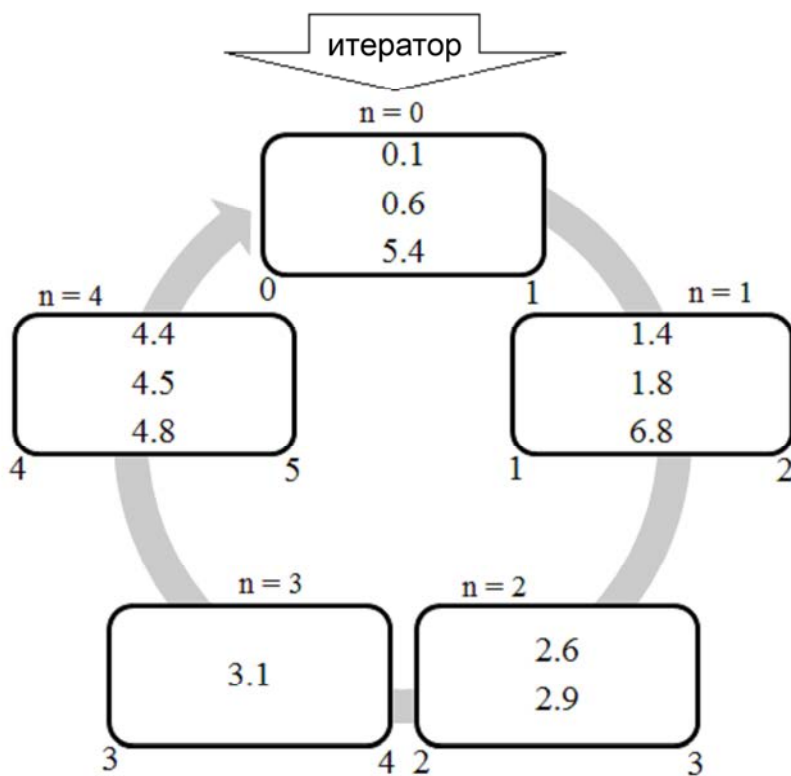


Рис. 2. Графическое представление кольцевой очереди

Необходимо также в структуре, которая описывает событие, предусмотреть поле, которое хранит номер круга r . В начале работы алгоритма итератор последовательно обрабатывает элементы текущего круга из ящика с номером 0. Воз-

можно удаление этих элементов в случае, когда вычисленные величины η_i выходят за пределы начального ящика. После очистки начального ящика происходит переход к следующему ящику и т. д. При возвращении к ящику с номером 0 значение переменной «текущий круг» увеличивается, после чего действия повторяются.

Для метода дискретно-событийного моделирования нет необходимости в упорядочивании всего множества элементов, достаточно на каждом этапе знать номер элемента, время наступления события которого минимально. В связи с этим структуру данных в отдельном ящике предлагается реализовать в виде хипа. На вершине хипа располагается номер элемента с минимальным временем наступления события. В элементах нижнего яруса хранятся номера событий, принадлежащих ящику.

Так как количество элементов в каждом ящике не постоянно, номера свободных элементов в нижнем ярусе хранятся в стеке (рис. 3).

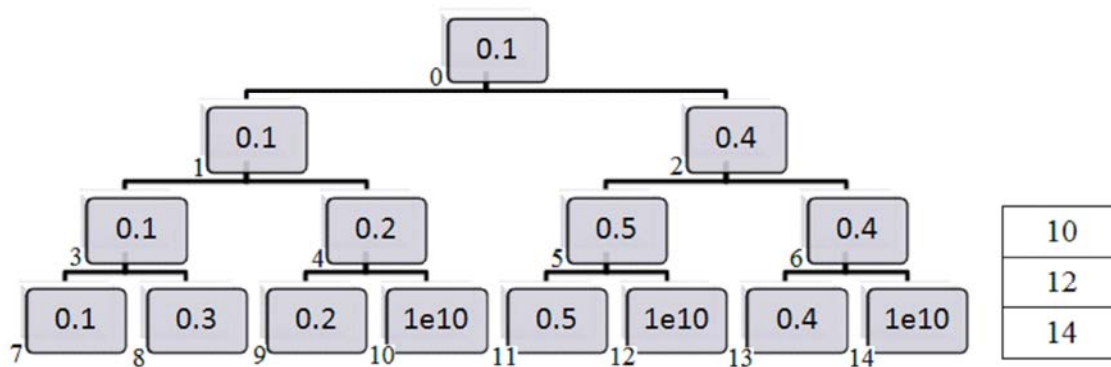


Рис. 3. Двоичное дерево со стеком

Удалив минимальный элемент из вершины хипа, листу, соответствующему этому элементу, присваивается заведомо большое время события, а номер листа помещается в стек. Дерево перестраивается.

Для добавления элемента в ящик необходимо выполнить следующие шаги:

1. Если элемент принадлежит другому ящику, то удалить его из того ящика.
2. Взять из стека номер свободного листа.
3. Свободному листу присвоить номер текущего элемента.
4. Перестроить дерево от свободного листа.

Для удаления события с минимальным временем наступления из ящика необходимо выполнить следующие шаги:

1. Запомнить номер минимального события.
2. Присвоить листу с минимальным событием заведомо большое время наступления события.
3. Поместить в стек номер листа.
4. Перестроить дерево для обработанного листа.
5. Вернуть номер минимального события вызывающей программе.

3 Выводы

В данной работе рассмотрено использование линейной сортировки в рамках дискретно-событийного алгоритма.

Теоретически линейный алгоритм эффективней логарифмического. Однако следует учесть, что константа при порядке в случае ящичной сортировки достаточно велика. Практика показывает, что в связи с необходимостью проведения большого количества операций с плавающей точкой в процессе приведения к равномерному закону распределения общее количество машинного времени может оказаться большим.

Тем не менее, следует ожидать, что линейный алгоритм ощутимо превысит логарифмические по производительности при объемах выборки модельных частиц порядка $10^8 - 10^9$. Такие объемы должны стать достижимыми с ростом производительности компьютеров.

Главным преимуществом данного метода перед методом турнира является непостоянство количества событий в очереди. В классической реализации метода дискретно-событийного моделирования событие связывается с каждой частицей, в связи с чем при событии столкновения частиц в очереди хранятся два одинаковых события. Следовательно общее количество событий может колебаться от N (только события пересечения границ подсистем) до $\frac{N}{2}$ (только события столкновения частиц).

Список литературы

1. Rapaport, D. C. Art of Molecular Dynamics Simulation [Text] / D. C. Rapaport . – Cambridge University Press, 2004. – 564 p.
2. Слепичева, М. А. Дискретно-событийное моделирование адсорбции водорода углеродными структурами [Текст] / М. А. Слепичева, В. А. Горячая, Ю. К. Чернышев // Труды НТК с международным участием «Компьютерное моделирование в наукоёмких технологиях» (КМНТ-2010), Харьков, 18-21 мая, 2010 г. – Т.1. – С . 118 – 122.
3. Chernyshev, Yu. Multiagent simulation of contact disease distribution based on event approach [Text] / Yu. Chernyshev, O. Sokolov // Proceedings of the International Conference on Modelling and Simulation 2010, 22–25 June 2010, Prague, Czech Republic. – P. 172 – 175.
4. Allen, M. P. Computer Simulations of Liquids [Text] / M. P. Allen, D. J. Tildesley. – Oxford Science Publications, 1987. – 408 p.
5. Donev, A. Asynchronous Event-Driven Particle Algorithms [Text] / A. Donev // Simulation. – 2009. – Vol. 85. – P. 229 – 242.
6. Чернышев, Ю. К. Событийное программирование. Применение к решению некоторых задач физики [Текст] / Ю. К. Чернышев. – Х.: Нац. аэрокосм. ун-т "ХАИ", 2008. – 68 с.
7. Alder, B. J. Phase transition for hard sphere system [Text] / B. J. Alder, T. E. Wainwright // J. Chem. Phys. – 1957. – Vol. 27. – P. 1208 – 1209.
8. Alder, B. J. Studies in molecular dynamics. 1. General method [Text] / B. J. Alder, T. E. Wainwright // J. Chem. Phys. – 1959. – Vol. 31. – P. 459 – 466.
9. Marin, M. Efficient Algorithms for Many-Body Hard Particle Molecular Dynamics [Text] / M. Marin, D. Risso, P. Cordero // J. Comp. Phys. – 1993. Vol. 109. – P. 306 – 317.
10. Левин, С.С. Применение RB-дерева для оптимизации алгоритма обработки потока событий при имитационном моделировании течения газа методом

частиц [Текст] / С. С. Левин, Ю. К. Чернышев // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. НАКУ «ХАИ». – Вып. 17. – Х., 2003. – С. 77 – 81.

11. Jones, D. W. An empirical comparison of priority-queue and event-set implementations [Text] / D. W. Jones // Comm. ACM. – 1986. – Vol. 29. – P. 300 – 311.

12. Marín, M. An empirical assessment of priority queues in event-driven molecular dynamics simulation [Text] / M. Marín, P. Cordero // Comput. Phys. Comm. – 1995. – Vol. 92. – P. 214 – 224.

13. Кормен, Т. Алгоритмы. Построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 2001. – 955 с.

14. Чернышев, Ю. К. Моделирование самоорганизации химических соединений Fm3m-кристаллографической группы в изотермическом приближении [Текст] / Ю. К. Чернышев, О. В. Хайленко // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. НАКУ «ХАИ». – Вып. 61. – Х., 2013. – С. 98 – 104.

15. Чернова, Н. И. Теория вероятностей [Текст] Учеб. пособие / Н. И. Чернова. – Новосиб. гос. ун-т. Новосибирск, 2007. – 160 с.

Поступила в редакцию 10.06.2015

Використання кільцевої черги в дискретно-подійному методі молекулярної динаміки

Розглянуто дискретно-подійний метод молекулярної динаміки. Проведено аналіз ефективності роботи основних етапів подійного алгоритму. Запропоновано використовувати алгоритм ящиків сортування для керування чергою подій, що дає можливість зменшити витрати обчислювального часу. Описано методику перетворення випадкової величини (часу настання події) у випадкову величину з рівномірною функцією розподілу, що дозволяє здійснювати рівномірне заповнення черг в ящиках. Наведено алгоритм ящиків сортування з використанням кільцевої черги.

Ключові слова: подійне моделювання, кільцева черга, ящикове сортування, рівномірний розподіл, двійкове дерево, стек.

Using a circular queue in discrete event-driven method of molecular dynamics

The discrete-event method of molecular dynamics was considered. The analysis of the efficiency of the main stages of event algorithm was done. It was proposed to use a bucket sort algorithm for queue management of events, which makes it possible to reduce the cost of computing time. A method for converting a random value (time of occurrence) as a random value with a uniform distribution function, which allows using uniform filling at the boxes, was proposed.

Keywords: event simulation, circular queue, bucket sort, uniform distribution, binary tree, stack.