

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(Самарский университет)

А.Н. АГАФОНОВ, А.В. ЕРЕМИН

МЕТОД КЛАССИЧЕСКОЙ МОЛЕКУЛЯРНОЙ ДИНАМИКИ В МОДЕЛИРОВАНИИ ФИЗИКО- ХИМИЧЕСКИХ ПРОЦЕССОВ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для студентов, обучающихся по основной образовательной программе высшего образования по направлениям подготовки 11.03.04 Электроника и нанoeлектроника

САМАРА
Издательство Самарского университета
2017

УДК: 539.19(075)

ББК: 22.36я7

А 235

Рецензенты: д-р техн. наук С.Б. К о н ы г и н,
канд. хим. наук В.И. П л а т о н о в

Агафонов, Андрей Николаевич

А 235 **Метод классической молекулярной динамики в моделировании физико-химических процессов:** учеб. пособие / *А.Н. Агафонов, А.В. Еремин.* – Самара: Изд-во Самарского университета, 2017. – 68 с.

ISBN 978-5-7883-1181-4

В пособии рассмотрены основные аспекты использования метода классической молекулярной динамики в моделировании физико-химических процессов. Приведены основные алгоритмы и их программная реализация на языке C++.

Пособие предназначено для студентов направления подготовки 11.03.04 Электроника и наноэлектроника.

УДК: 539.19(075)

ББК: 22.36я7

ISBN 978-5-7883-1181-4

© Самарский университет, 2017

СОДЕРЖАНИЕ

Введение	4
Используемые сокращения	8
Часть 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДА МОЛЕКУЛЯРНОЙ ДИНАМИКИ	9
1.1. Идея метода молекулярной динамики	9
1.2. Обоснование и границы применимости молекулярно-динамического метода	11
1.3. Силы межмолекулярного взаимодействия	13
1.4. Периодические граничные условия (ПГУ)	17
1.5. Термостатирование моделируемой системы	20
1.6. Применение ММД при описании физико-химиче- ских процессов на примере адсорбционно-десорбцион- ных системы	23
1.7. Дискретность времени	33
1.8. Численное решение уравнений движения	36
1.9. Алгоритмы и схемы интегрирования	39
Часть 2. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОГРАММЫ МОДЕЛИРОВАНИЯ	45
2.1. Общие сведения	45
2.2. Основная функция C++	49
2.3. Генерация системы и начальных условий	51
2.4. Алгоритм численного интегрирования уравнений движения на C++	57
2.5. Заголовочный файл (объявление класса и методов)	59
Заключение	62
Библиографический список	63
Приложение А. Таблица коэффициентов c_{mn}	67

ВВЕДЕНИЕ

Современные технологии позволяют создавать устройства, работа которых определяется физико-химическими процессами, протекающими в микро- и мезоразмерных структурах. К данным устройствам могут быть отнесены, например, современные сенсоры, микрофлюидные устройства, катализаторы и т. д. Традиционные подходы к моделированию процессов массопереноса, особенно на границе раздела фаз, имеют существенные ограничения при описании процессов в микро- и мезосистемах [1], в связи с чем возникает все больший интерес к моделированию процессов на микро- и наноуровнях, за счет интегрирования систем с большим количеством частиц, движение которых описывается уравнениями ньютоновской механики.

Численное интегрирование подобной системы во многих случаях дает достаточно полную картину эволюции молекулярных систем. Усреднение параметров частиц (пространственных и энергетических характеристик движения) позволяет получить информацию о структуре ансамбля частиц и о термодинамических и кинетических свойствах системы. Такой подход дает возможность исследовать макроскопические свойства молекулярных систем и в настоящее время получил широкое распространение в различных областях физики, химии и биологии [2]. Серьезной проблемой на ранних этапах развития данных методов являлось достаточно жесткое ограничение максимального количества частиц системы из-за недостаточной вычислительной мощности. Однако в последние десятилетия развитие игровой компьютерной индустрии дало серьезный толчок развитию доступных вычислительных устройств и созданию обширной алгоритмической базы. Такие технологии, как CUDA, OpenCL и прочие, позволяющие выполнять параллельные вычисления на видеопроцессорах (ГПУ), получили широкое применение во многих областях науки и техники из-за их превосходства в вычислительной мощности, что показано на рисунке 1.

Theoretical GFLOP/s

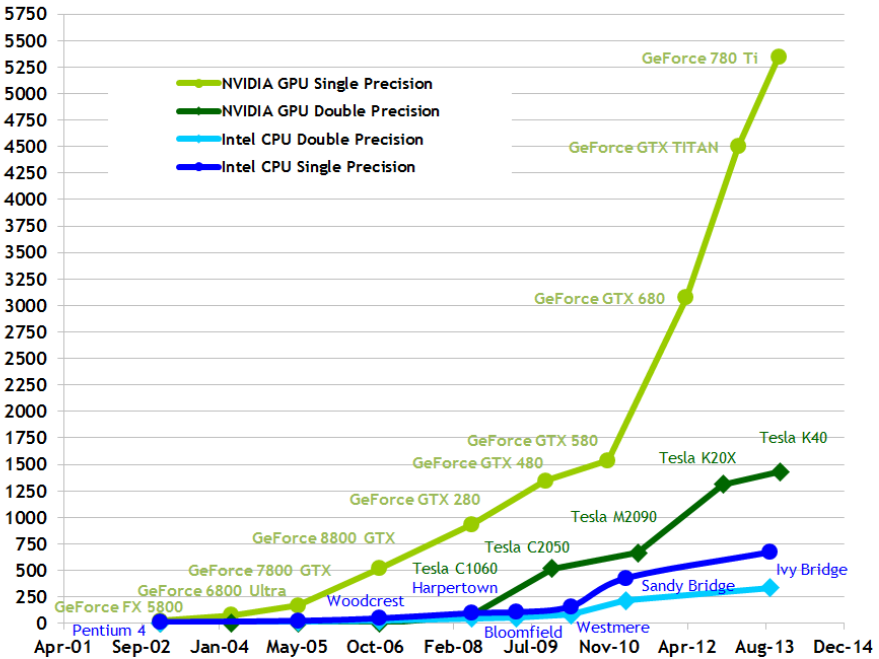


Рис. 1. Количество операций с плавающей точкой на ЦПУ и ГПУ

Данные технологии позволили значительно увеличить эффективность обработки больших объемов данных, что в свою очередь стало предпосылкой к скачку популярности методов частиц. Современные персональные компьютеры, имеющие тактовые частоты несколько гигагерц, позволяют за реальное время (порядка 10^4 сек.) решить систему уравнений движения для нескольких сотен тысяч частиц. Дополнение этих уравнений периодическими граничными условиями приближает систему к реальной, то есть многократно увеличивает в ней эффективное количество частиц, чтобы для нее соблюдалось второе начало термодинамики. К сожалению, любое посильное компьютеру число частиц все равно бесконечно мало по сравнению с реальным, имеющим порядок числа Авогадро $6.023 \cdot 10^{23}$ моль⁻¹.

Даже при реально используемом числе частиц $n = 10^3 \dots 10^6$ количество расчетов, необходимое для генерации одного шага фазовой траектории огромно. Поскольку при расчетах сил межчастичного взаимодействия необходимо учесть взаимодействие каждой частицы с каждой из оставшихся в системе, то есть количество взаимодействий равно $n(n-1)/2$ умноженное на количество типов взаимодействий. Другими словами трудоемкость этого алгоритма квадратична по числу частиц в системе, тогда как большинство других алгоритмов линейно по числу частиц. Несмотря на это, МД является весьма привлекательным способом изучения свойств вещества вследствие суммированных в нижеследующем списке возможностей:

- в отличие от натуральных экспериментов МД на одном и том же компьютере позволяет изучать различные свойства вещества, обусловленные коллективным поведением молекул, для исследования которых в натуральных экспериментах потребовалось бы множество специализированных установок;
- в отличие от моделей и теорий, которые обычно отбрасывают множество факторов вследствие невозможности их одновременного аналитического описания, МД позволяет получить результаты на основе любого количества заложенных в модель взаимодействий, лишь бы это позволили аппаратные и временные ресурсы, то есть ЧЭ (численный эксперимент) заполняет разрыв между теорией и реальным экспериментом;
- в отличие от натуральных экспериментов в ЧЭ по методу МД точно известно строение исследуемой системы, поскольку он начинается с ее конструирования;
- в отличие от натуральных экспериментов ЧЭ позволяет произвольно варьировать практически любые характеристики молекулярных систем в диапазонах, недоступных натурным экспериментам;
- расширение диапазона варьирования какого-либо параметра обычно используется для акцентирования его воздействия на систему;

- уникальное свойство ЧЭ – визуализация молекулярного движения и структуры моделируемой системы.

Данное методическое пособие посвящено основам моделирования физико-химических процессов с использованием метода классической молекулярной динамики и предназначено для студентов направления «Электроника и наноэлектроника».

Основная цель пособия - описание необходимых для реализации простейших вариантов метода молекулярной динамики уравнений, алгоритмов и их реализаций на языке C++, и достижение достаточной производительности на повсеместно распространенных современных персональных компьютерах. Последовательное изучение и собственная реализация представленных алгоритмов, использующих все аппаратные ресурсы персональных компьютеров, позволяет моделировать структуру некоторых простых веществ и физическую кинетику в любом их агрегатном состоянии: твердом, жидком и газообразном.

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

ГПУ – графическое процессорное устройство (видеопроцессор)

ЦПУ – центральное процессорное устройство

МД – молекулярная динамика

ЧЭ – численный эксперимент

ПЛД – потенциал Леннарда-Джонса

ПГУ – периодические граничные условия

ДУ – дифференциальное уравнение

ЭВМ – электронно-вычислительная машина

Часть 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДА МОЛЕКУЛЯРНОЙ ДИНАМИКИ

1.1. Идея метода молекулярной динамики

Метод молекулярной динамики (МД) – это метод, в котором временная эволюция системы взаимодействующих атомов или частиц отслеживается интегрированием их уравнений движения. Для описания движения атомов или частиц применяется классическая механика.

При моделировании силы межатомного взаимодействия представляют в форме классических потенциальных сил (как градиент потенциальной энергии системы). Наборы конфигураций, получаемые в ходе расчетов методом молекулярной динамики, распределены в соответствии с некоторой статистической функцией распределения, например, отвечающей микроканоническому распределению. Заметим, что точное знание траекторий движения частиц системы на больших промежутках времени не является необходимым для получения результатов макроскопического (термодинамического) характера [3].

Моделирование методом молекулярной динамики во многих отношениях очень похоже на проведение натуральных экспериментов. Проведение эксперимента связано с выполнением некоторой последовательности действий. Во-первых, необходимо подготовить исследуемый образец. Во-вторых, подключить к этому образцу измерительный прибор (например: термометр, манометр и т. п.). И, наконец, провести измерение (прямое или косвенное) интересующей физической величины на протяжении некоторого времени. Если результаты эксперимента содержат статистическую ошибку (шум), тогда чем дольше проводится измерение, тем точнее можно вычислить конечный результат. При моделировании методом МД выполняется та же самая последовательность действий. В начале выбирается модель системы, состоящей из N частиц, и производится решение уравнений движения Ньютона до тех пор, пока характеристики данной системы не перестанут изменяться с течением

времени (т. е. система перейдет в равновесное состояние). После уравновешивания происходит само измерение. По факту, ошибки, которые можно допустить в численном эксперименте, очень схожи с ошибками при натуральном эксперименте (некорректно подготовлен образец/модель, время измерения слишком мало, во время эксперимента система подвергается необратимому воздействию или измеренная величина неправильно интерпретируется).

Для измерения наблюдаемых величин в молекулярно-динамическом эксперименте необходимо, чтобы данные величины можно было выразить как функции координат и импульсов частиц системы. Например, подходящее описание температуры для (классической) системы многих тел выражается через кинетическую энергию, входящую в состав выражения Гамильтониана системы. В частности, для среднего значения кинетической энергии на степень свободы имеем следующее выражение:

$$\left\langle \frac{p_\alpha^2}{2m} \right\rangle = \frac{1}{2} k_B T. \quad (1)$$

Данное выражение используется в численных экспериментах как рабочее определение температуры. Практически нужно измерить полную кинетическую энергию системы и разделить ее на кол-во степеней свободы $N_f (= 3N - 3$ для системы из N частиц с постоянным суммарным импульсом). Так как суммарная кинетическая энергия колеблется, моментальная температура может быть измерена следующим образом:

$$T(t) = \sum_{i=1}^N \frac{p_i^2(t)}{m_i k_B N_f}. \quad (2)$$

Флуктуации измеренной таким образом температуры будут порядка $1/\sqrt{N_f}$. При величине N_f порядка 10^2 - 10^3 , статистические отклонения температуры составляют 5–10%. Следовательно, для того чтобы дать точную оценку, необходимо взять среднее значение за некоторое время.

1.2. Обоснование и границы применимости молекулярно-динамического метода

Метод молекулярной динамики применим, если длина волны Де-Бройля атома (или частицы) много меньше, чем межатомное расстояние. Также классическая молекулярная динамика не применима для моделирования систем, состоящих из легких атомов, таких как гелий или водород. Кроме того, при низких температурах квантовые эффекты становятся определяющими и для рассмотрения таких систем необходимо использовать квантово-химические методы. Необходимо, чтобы времена на которых рассматривается поведение системы, были больше, чем время релаксации исследуемых физических величин.

Возможность классического описания движения таких квантово-механических объектов как молекулы является следствием того факта, что энергия трансляционного движения молекул в макроскопическом объеме практически не квантуется. Это является следствием решения стационарного уравнения Шредингера для волновой функции Ψ частицы со значениями энергии W в бесконечно глубокой потенциальной яме при поступательном движении [4]:

$$\frac{d^2\Psi}{dx^2} + \frac{2m}{\hbar}W\Psi = 0,$$
$$\Delta W_n = W_{n+1} - W_n = \frac{\hbar^2}{8ml^2}(2n + 1) \approx \frac{\hbar^2}{4ml^2}n. \quad (3)$$

Если взять массу $m \approx 10^{-25}$ кг (масса молекулы), размер потенциальной ямы $l \approx 10$ см (молекулы газа или жидкости в сосуде), получим расстояния между соседними энергетическими уровнями в зависимости от их номера n : $\Delta W \approx 10^{-20}n$ эВ. Столь густо расположенные энергетические уровни будут практически восприниматься как сплошной спектр энергии. Так что хотя квантование энергии в принципе будет иметь место, но на характере движения молекул сказываться не будет.

Даже если молекула находится внутри поры размером $l \approx 10^{-9}$ м, расстояние между соседними энергетическими уровнями будет $\Delta W \approx 10^{-4} n$ эВ, то есть все еще существенно меньше, чем средняя энергия теплового движения при комнатной температуре ~ 0.05 эВ. Более подробный теоретический анализ приведенных соображений содержится в работах [5, 6, 7].

Кроме приближенной применимости классических законов для описания движения необходимо еще упомянуть влияние накопления ошибки при численном решении дифференциальных уравнений движения частиц на большом интервале времени. С теоретической точки зрения здесь, на первый взгляд, возникает непреодолимая трудность, состоящая в том, что ошибка численного решения эквивалентна смене начальных условий решения дифференциальных уравнений в небольших пределах или, другими словами, переброс системы на другую фазовую траекторию. В данном случае проблема разрешается тем, что переброс происходит между различными микросостояниями системы, относящимися к одному и тому же макросостоянию.

Коррекция энергии системы, проводимая периодически в ходе численного решения уравнений движения частиц, принудительно возвращает систему на начальную макротраекторию, которая, таким образом, является суперпозицией фазовых траекторий всех микрочастиц системы, отвечающих данному ее термодинамическому состоянию. Наглядно это можно представить, как и следует из определения микросостояния [4], что неразличимые частицы системы как бы обмениваются своими фазовыми траекториями, лежащими в широкой полосе, отвечающей данной фазовой макротраектории.

1.3. Силы межмолекулярного взаимодействия

Для описания межмолекулярного взаимодействия в ММД обычно используют различные потенциалы парного взаимодействия. Одним из самых распространенных является потенциал Леннарда-Джонса (ПЛД), также известный как потенциал m - n , который применяется для моделирования движения простейших сферических бесструктурных частиц в трехмерном каноническом ансамбле ввиду хорошего воспроизведения взаимодействия между реальными неполярными молекулами [8]. Далее приведена общая форма записи данного потенциала.

$$U_{mn} = \varepsilon c_{mn} \left[\left(\frac{\sigma}{r} \right)^n - \left(\frac{\sigma}{r} \right)^m \right], \quad n > m, \quad (4)$$

где ε – глубина потенциальной ямы в равновесном состоянии, σ – характерный ван-дер-ваальсовдиаметр молекулы, r – расстояние между центрами частиц, а c_{mn} имеет следующий вид:

$$c_{mn} = \left(\frac{n}{m} \right)^{m/n-m} \left(1 - \frac{m}{n} \right)^{-1}. \quad (5)$$

Рассчитанные значения коэффициента c_{mn} для некоторых пар mn представлены в приложении А.

Расстояние, соответствующее минимальному значению потенциальной энергии, имеет следующий вид:

$$r_{min} = \sigma \left(\frac{n}{m} \right)^{1/n-m}. \quad (6)$$

Характерный вид кривой, соответствующей форме ПЛД представлен на рисунке 2.

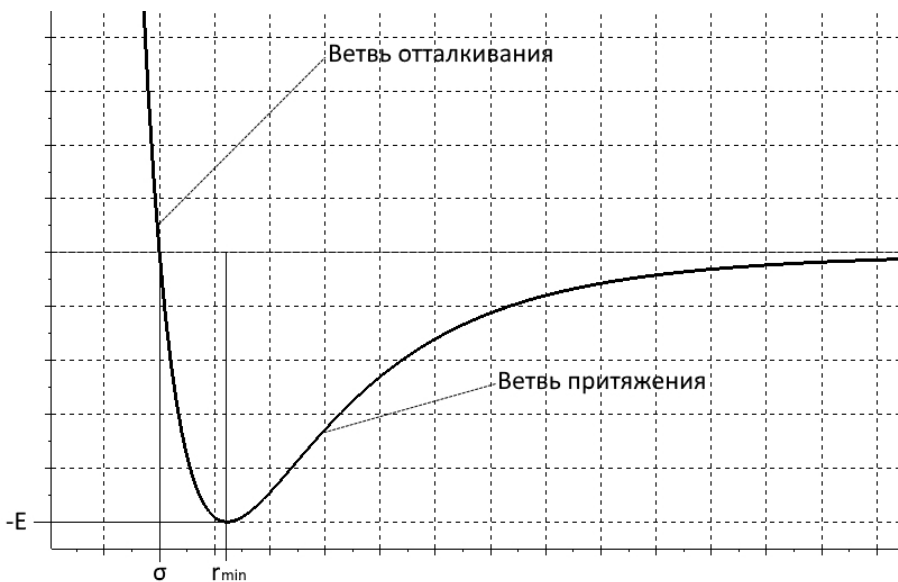


Рис. 2. Характерный вид потенциала ЛД

Наиболее известна форма, называемая ПЛД 6-12, имеет вид:

$$U = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]. \quad (7)$$

В реальном ЧЭ обычно используется усеченная форма, ограничивающая расстояние взаимодействия несколькими ван-дер-ваальсовыми диаметрами. Математическое представление данной формы приведено ниже.

$$U_c = \begin{cases} \varepsilon c_{mn} \left[\left(\frac{\sigma}{r} \right)^n - \left(\frac{\sigma}{r} \right)^m \right] + U_{mn}(r_c), & r \leq r_c \\ 0, & r > r_c \end{cases} \quad (8)$$

Здесь r_c – расстояние усечения потенциала, $U_{mn}(r_c)$ – значение неусеченного потенциала на расстоянии усечения. Обычно расстояние усече-

ние принимается равным $2.5 \dots 3\sigma^1$. Такой выбор обусловлен тем, что на этом расстоянии значение энергии взаимодействия составляет всего примерно 1.6 % от глубины потенциальной ямы.

При рассмотрении взаимодействия сферических частиц разного вида, отличающихся энергией взаимодействия и ван-дер-ваальсовыми диаметрами, перекрестные величины определяются правилом Бергло-Лоренца [10]:

$$\varepsilon_{12} = \sqrt{\varepsilon_1 \cdot \varepsilon_2}; \quad \sigma_{12} = 0.5(\sigma_1 + \sigma_2). \quad (9)$$

Если среднеарифметическое соотношение для межцентровых расстояний частиц обосновано геометрическими соображениями, то среднегеометрическое значение для глубины взаимной потенциальной ямы никак не обосновано и является, всего лишь, общепринятым правилом, не характеризующим никак внутреннюю структуру частиц.

ПЛД является двухпараметрическим, поэтому он имеет очень ограниченные возможности для вариации макроскопических параметров моделируемого им материала. Фактически он позволяет удовлетворить значению только одного макроскопического параметра – это, например, модуль упругости или предел прочности в статике, скорость распространения продольных волн и др. С другой стороны, данный потенциал весьма точно описывает свойства ряда веществ (прежде всего, кристаллических инертных газов), а также достаточно точно описывает силы взаимодействия Ван-дер-Ваальса. К несомненным достоинствам ПЛД относится также его вычислительная простота, не требующая вычисления иррациональных и трансцендентных функций. ПЛД широко применяется как классический модельный потенциал, особенно в работах, в

¹Если принять $r_c = r_{min}$, то такое соотношение называется Weeks-Chandler-Anderson – потенциал [9] – эти авторы установили, что структура и динамика конденсированного состояния вещества обусловлена именно отталкиванием составляющих его частиц, а притяжение определяет, в основном, термодинамические свойства.

которых основной задачей является не получение точных количественных результатов, а исследование общих физических закономерностей [11].

Дополняя ПЛД дополнительным членом дипольного взаимодействия можно получить потенциал Штокмайера. В силу того, что рассматриваются молекулы обладают постоянными дипольными моментами, энергия взаимодействия пары таких молекул будет зависеть не только от расстояния между ними, но и от их взаимной ориентации. В СГС потенциал Штокмайера запишется в следующем виде:

$$U_{\text{Ш}} = \varepsilon c_{mn} \left[\left(\frac{\sigma}{r} \right)^n - \left(\frac{\sigma}{r} \right)^m \right] - \frac{\mu_a \mu_b}{r^3} g(\theta_a, \theta_b, \varphi), \quad (10)$$

где θ_a, θ_b – полярные углы молекул; $\varphi = \varphi_b - \varphi_a$ – разность азимутальных углов молекул (см. рисунок 3); $g(\theta_a, \theta_b, \varphi) = 2 \cos \theta_a \cos \theta_b - \sin \theta_a \sin \theta_b \cos \varphi$.

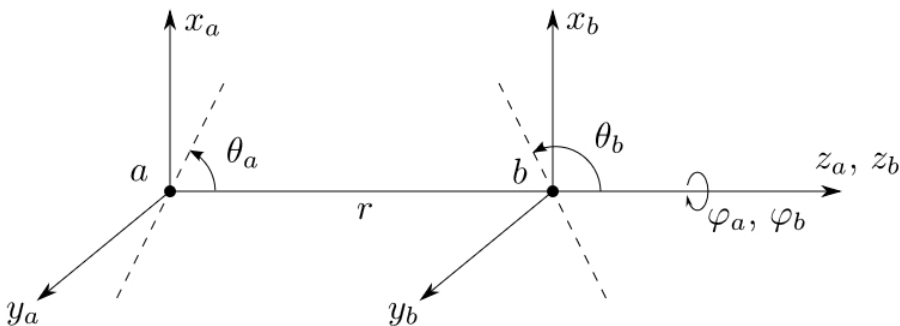


Рис. 3. Иллюстрация обозначений, использованных в записи потенциала Штокмайера

Легко увидеть, что потенциал Штокмайера представляется в виде суперпозиции двух более простых потенциалов $U_{\text{Ш}} = U_{\text{ЛД}} + U_d$. Первый из

них представляет собой потенциал Леннард-Джонса, второй – потенциал взаимодействия двух диполей.

Следует помнить что, потенциал Штокмайера наследует все ограничения потенциала Леннард-Джонса и использует выражения для взаимодействия только точечных диполей, в то время как реальные молекулы имеют конечные размеры [12].

1.4. Периодические граничные условия (ПГУ)

Надо отметить, что минимально возможное для расчета количество частиц в модельной системе определяется масштабом тех молекулярных процессов, которые необходимо смоделировать. Очевидно, что если исследуются крупномасштабные процессы порядка размера полимерной цепи, то система должна быть в несколько раз больше размеров этой цепи. Но если моделируются локальные процессы перестройки ячейки в жидкости или энергетические взаимодействия в ней, то количество частиц в модельной системе должно быть лишь в несколько раз больше, чем их количество в сфере молекулярного действия. Это показано в ряде статей [13, 14, 15, 16] по моделированию так называемого ландшафта потенциальной энергии частиц путем построения зависимостей целевой функции от количества частиц в системе, начиная с 65. Исследование целевой физической характеристики от количества частиц в системе полезно проводить во всех случаях перед началом длительного ЧЭ, чтобы попытаться минимизировать его время или не потратить его зря, проводя ЧЭ с недостаточными размерами системы. Однако, как уже было отмечено ранее, даже самые передовые вычислительные машины не в состоянии моделировать реальные физические системы целиком, что приводит к необходимости использования различных синтетических упрощений модельной системы. Одним из подобных решений является использование ПГУ, что делает модельную систему квази-бесконечной. Стоит

отметить, что использование ПГУ возможно только при условии, что минимальный размер (по любой оси) системы уже достиг некоторой величины – 20-кратного ван-дер-ваальсова диаметра частицы, после которой не проявляются гидродинамические эффекты, вносимые периодическими граничными условиями (называемые эффектами конечного размера системы) [17, 18, 19, 20, 21, 22, 23, 24].

Рассмотрим систему, представляющую собой параллелепипед со сторонами L_x , L_y и L_z и ограниченную плоскостями $\alpha = 0$ и $\alpha = L_n$, где $n = x, y, z$ (ячейка). Такой выбор объема системы удобен для численных расчетов и не накладывает никаких дополнительных ограничений в случае жидкостей и газов. Такая система с заданным значением энергии E и состоящая из постоянного числа частиц N представляет микроканонический ансамбль [25], для моделирования которого характерно использование ПГУ [26]. Эти условия фактически устремляют объем системы и эффективное количество частиц к бесконечности, что приводит к уменьшению влияния ограничивающих поверхностей на поведение системы.

Использование ПГУ подразумевает, что ячейка (и все находящиеся в ней частицы) повторяется в пространстве бесконечное количество раз. Математически это означает, что для любой наблюдаемой величины Ψ выполняется соотношение:

$$\Psi(\vec{r}) = \Psi(\vec{r} + \vec{\gamma}), \quad \vec{\gamma} = aL_x\vec{i} + bL_y\vec{j} + cL_z\vec{k}, \quad a, b, c \in Z. \quad (11)$$

Фактически, это означает, что если частица вылетела за границу исходной ячейки, то такая же частица влетела через противоположную стенку с той же скоростью. Стоит отметить, что в случае ПГУ изменится выражение (34) для потенциальной энергии. Теперь необходимо учитывать не только взаимодействия с частицами исходной ячейки, но также с частицами во всех остальных ячейках:

$$\frac{d^2\vec{r}_i}{dt^2} = \frac{1}{m_i} \left[\sum_{\substack{j=1..N \\ j \neq i}}^N \left(\frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} F(|\vec{r}_{ij}|) \right) + \right. \\ \left. + \sum_{\vec{n} \neq \vec{0}} \sum_{\substack{j=1..N \\ j \neq i}}^N \left(\frac{\vec{r}_{ij} + \vec{y}}{|\vec{r}_{ij} + \vec{y}|} F(|\vec{r}_{ij} + \vec{y}|) \right) \right]. \quad (12)$$

ПЛД (4) короткодействующий, поэтому в сумме (12) ограничимся только ближайшими к данной частице соседями, вне зависимости от того, лежат они в исходной ячейке или нет. То есть, под расстоянием $|\vec{r}_{ij}|$ между частицами с номерами i и j мы будем понимать:

$$|\vec{r}_{ij}| = \min_{\vec{n}} |\vec{r}_{ij} + \vec{y}|. \quad (13)$$

Данное действие фактически обрезает потенциал U на расстояниях $r_c = \sqrt{3}L_n/2$.

При выполнении численного моделирования необходимо проверять переход частицы за границы ячейки и возвращать назад в соответствии с ПГУ.

1.5. Термостатирование моделируемой системы

Число частиц моделируемой системы n и ее объем v чаще всего неизменны в процессе генерации фазовой траектории. Кроме того, при решении уравнений движения производится дополнительная принудительная коррекция какой-либо физической величины, являющейся согласно постановке задачи интегралом движения. Если постоянной поддерживается полная энергия системы W , то она называется микроканоническим или nvW – ансамблем. Если постоянна температура системы T , то есть средняя кинетическая энергия частиц, то она называется каноническим или nvT – ансамблем [27, 28, 29]. В этих двух случаях уравнения движения имеют простой физический смысл уравнений Ньютона и кинематических уравнений.

Реже для приближения свойств системы к реальному производится моделирование с постоянным давлением P и температурой, а система называется каноническим nPT – ансамблем. В этом случае приходится дополнять уравнения движения частиц уравнениями изменения объема системы, физический смысл которых отсутствует, но которые призваны описать термическое расширение системы для уравнивания давления в ней с давлением внешней среды.

В данном пособии рассмотрим лишь nvT – ансамбль, как самый простой и физически понятный. На основе отлаженных для него алгоритмов затем, в случае необходимости, можно написать алгоритмы и для всех остальных более сложных ансамблей.

Для сохранения постоянной температуры системы на каждом шаге интегрирования вычисляется мгновенная средняя кинетическая энергия одной частицы, которая согласно принципу равнораспределения энергии по степеням свободы с точностью до постоянного множителя равна безразмерной абсолютной температуре:

$$\varepsilon_K(t) = \frac{1}{2n} \sum_{i=1}^n m_i [v_{xi}^2(t) + v_{yi}^2(t) + v_{zi}^2(t)] = \frac{3}{2} k_B T. \quad (14)$$

В формуле в явном виде записаны компоненты скорости частиц. Эта энергия, также один раз на каждом шаге добавляется или убавляется за счет введенной ниже термостатирующей силы (15), согласно наиболее популярному алгоритму Nose-Hoover [30, 27]:

$$f_{Ti}(t) = (\chi \cdot m_i) \cdot \left[\frac{\varepsilon_{K0}}{\varepsilon_K(t)} - 1 \right] \cdot v_{ai}(t), \quad (15)$$

где ε_{K0} – кинетическая энергия, соответствующая заданной температуре, χ – коэффициент пропорциональности, который стараются сделать как можно меньше, обычно порядка 0.01...0.20, чтобы как можно меньше возмущать детерминированное начальными условиями движение.

Практически этот коэффициент можно сделать равным нулю при уменьшении шага интегрирования, но тогда производительность алгоритма, соответственно, упадет. Поэтому его постепенно увеличивают от нулевого значения вместе с увеличением шага интегрирования, следя за тем, чтобы целевые динамические характеристики системы не менялись. Останавливаются при этом условия на величинах Δt и χ , обеспечивающих максимально производительность и неограниченно долгую устойчивую работу алгоритма.

Как видно из (15), термостатирующая сила практически представляет собой знакопеременную жидкоподобную силу сопротивления, которая уменьшает скорость частиц, если она завышена, и увеличивает ее, если она занижена. При этом пропорциональность силы скорости частицы сохраняет максвелловское распределение скоростей, а не просто нивелирует скорости всех частиц к средней тепловой. Масса частицы в данном случае играет лишь формальную роль согласования размерностей и в тексте алгоритма может быть включена в χ при подборе ее величины, что указывается объединением этих характеристик в скобках, лучше с учетом принадлежности частицы к какой-то группе в неомогенной системе.

Дополнительно для сохранения постоянной температуры системы, можно периодически через несколько сотен или тысяч шагов интегриро-

вания перемасштабировать скорости всех частиц для точного принудительного восстановления ее полной кинетической энергии, опять же с сохранением максвелловского распределения скоростей:

$$v_{\alpha i}(t) = v_{\alpha i}(t) \cdot \sqrt{\frac{\varepsilon_{K0}}{\varepsilon_K(t)}}. \quad (16)$$

С той же периодичностью необходимо повторять процедуру восстановления неподвижности центра масс моделируемой системы частиц, которая нарушается как конечностью шага интегрирования, так и термостатированием. Для этого из скорости каждой частицы, естественно по координатно, вычитается скорость центра масс системы на данном шаге:

$$v_{\alpha i}(t) = v_{\alpha i}(t) - \left(\sum_{j=1}^n m_j \cdot v_{\alpha j}(t) / \sum_{j=1}^n m_j \right). \quad (17)$$

Если ошибка не накапливается, то вычитаемое всегда будет мало. Периодичность этой процедуры также подбирается эмпирически.

В завершение раздела следует упомянуть, что все константные выражения и связанные с ними арифметические операции выносятся за пределы любого цикла и считаются только один раз на данном шаге, как например χ , или один раз при запуске программы, как например масса системы, фигурирующая в знаменателе (17).

В гомогенной системе, состоящей из частиц одинаковой единичной массы, ее символ вообще не будет фигурировать в уравнениях движения, а во всех уравнениях символ массы стоит для сохранения, во-первых, их физического смысла, во-вторых, общности алгоритмов, которые могут быть использованы для моделирования систем с частицами разного вида.

1.6. Применение ММД при описании физико-химических процессов на примере адсорбционно-десорбционных системы

Адсорбция малых и цепных молекул на поверхности твердого тела, или просто ограничение объема для движения молекул непроницаемой поверхностью различной формы является, с одной стороны, распространенной в микромире ситуацией, а с другой стороны, достаточно подробно теоретически проработано, например, для полимерных цепей [31, 32, 33, 34]. С точки зрения МД, такие объекты также весьма удобны для исследования, поскольку устраняют ПГУ вдоль одной оси координат при исследовании адсорбции и вдоль всех трех осей при исследовании подвижности в замкнутом объеме [35, 36, 37, 38, 39]. Последнее обстоятельство сразу повышает статистику ЧЭ, так как в этом случае частицы не могут покинуть пределы базовой ячейки.

Поверхность обычно вводится в систему путем интегрирования ПЛД по плоскости или по объему полупространства, в которых расположены неподвижные частицы твердого тела. Такой механизм перехода от отдельных частиц к непрерывной поверхности демонстрируется на рисунке 4. Полученный интегральный потенциал определяет закономерность спада получившегося ван-дер-ваальсова потенциального поля, зависящего теперь только от одной координаты, перпендикулярной поверхности [3].

При этом коэффициенты при разных степенях получающегося при интегрировании полинома, зависят от предположенной кристаллической или аморфной структуры твердого тела. Однако, решающую роль играют степени полинома, равные в данном случае 10 и 4 – именно они определяют закономерность спада получившегося ван-дер-ваальсова потенциального поля, зависящего теперь только от одной координаты, перпендикулярной поверхности.

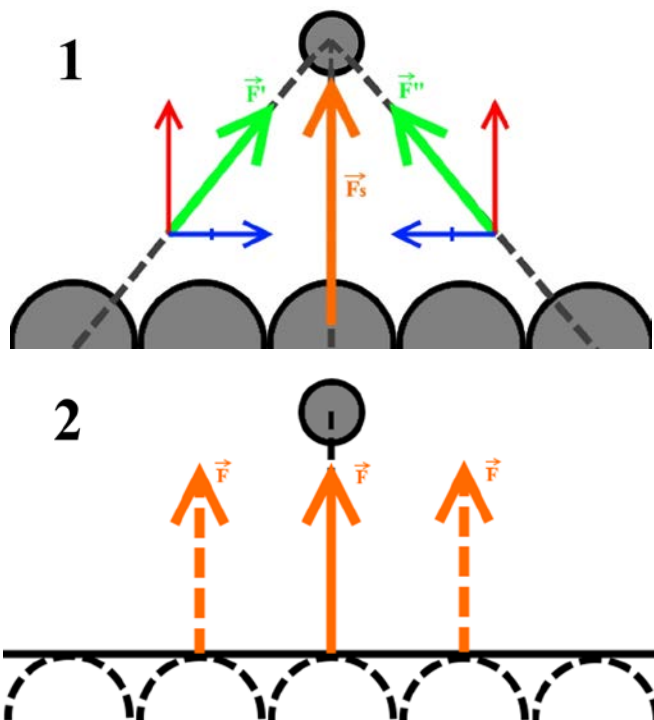


Рис. 4. Переход к плоской стенке

Если две поверхности, параллельные плоскости XU , образованы частицами, центры которых лежат в плоскостях $z = 0$ и $z = L_z$, то результирующее потенциальное поле в объеме базовой ячейки удобно описывать следующим обобщенным выражением [40, 41, 42, 43, 44, 45]:

$$\begin{aligned}
 U_{A0}(z) &= c_{10_4} \varepsilon_W \left[\left(\frac{\sigma}{z} \right)^{10} - \left(\frac{\sigma}{z} \right)^4 \right], \\
 U_{Az}(z) &= c_{10_4} \varepsilon_W \left[\left(\frac{\sigma}{L_z - z} \right)^{10} - \left(\frac{\sigma}{L_z - z} \right)^4 \right],
 \end{aligned}
 \tag{18}$$

где U_{A0} и U_{Az} – потенциальная энергия взаимодействия с двумя адсорбирующими стенками; ε_W – глубина потенциальной ямы взаимодействия

со стенкой, когда расстояние равно равновесному, а c_{10_4} – коэффициент для потенциала ЛД10-4.

Использование выражения (18) дает две адсорбирующие стенки, ограничивающие базовую ячейку перпендикулярно оси Z. Вдоль осей X и Y по-прежнему сохраняются ПГУ.

Только надо помнить, что адсорбирующий эффект начинает проявляться, когда ε_W становится больше ε_K (средней кинетической энергии теплового движения частиц), лучше в несколько раз. В этом случае взаимодействие со стенками становится термодинамически более выгодным, чем между частицами жидкости, и ближайшие из них начинают адсорбироваться на стенках. Количество устойчиво адсорбированных монослоев при этом тем больше, чем больше энергия адсорбции по сравнению с энергией теплового движения [42].

Однако стоит отметить, что при таком подходе граница раздела становится адиабатической, т. е. не происходит обмен энергией между моделируемыми частицами и стенкой. Данную проблему можно решить внедрением термостатирующей силы, но наблюдение за процессами, требующими непосредственного взаимодействия частиц твердого тела и жидкой или газовой фазы (например, физическая адсорбция), становится затруднительным.

Существует достаточно много методов реализации теплообмена при взаимодействии частицы со стенкой. В данной работе принято допущение, что каждый атом в узле кристаллической решетки представляет собой единичный осциллятор, совершающий автоколебания, амплитуда которых зависит от температуры кристалла. Общий вид автоколебательной системы схематически показан на рисунке 5.

В такой модели роль источника энергии играет твердое тело, колебательной системой является ячейка поверхности, а регулятор и цепь обратной связи работают по схеме, представленной на рисунке 6.

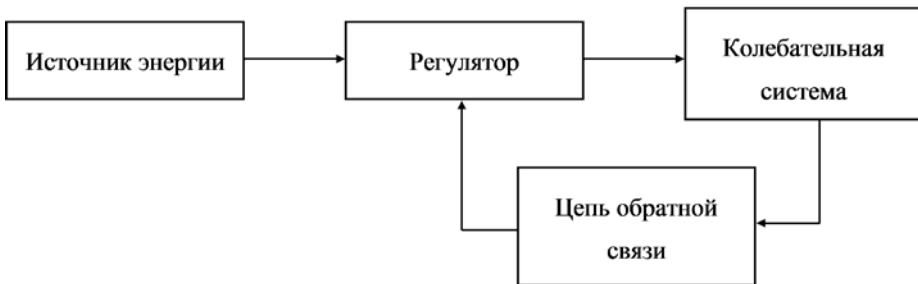


Рис. 5. Функциональная схема автоколебательной системы

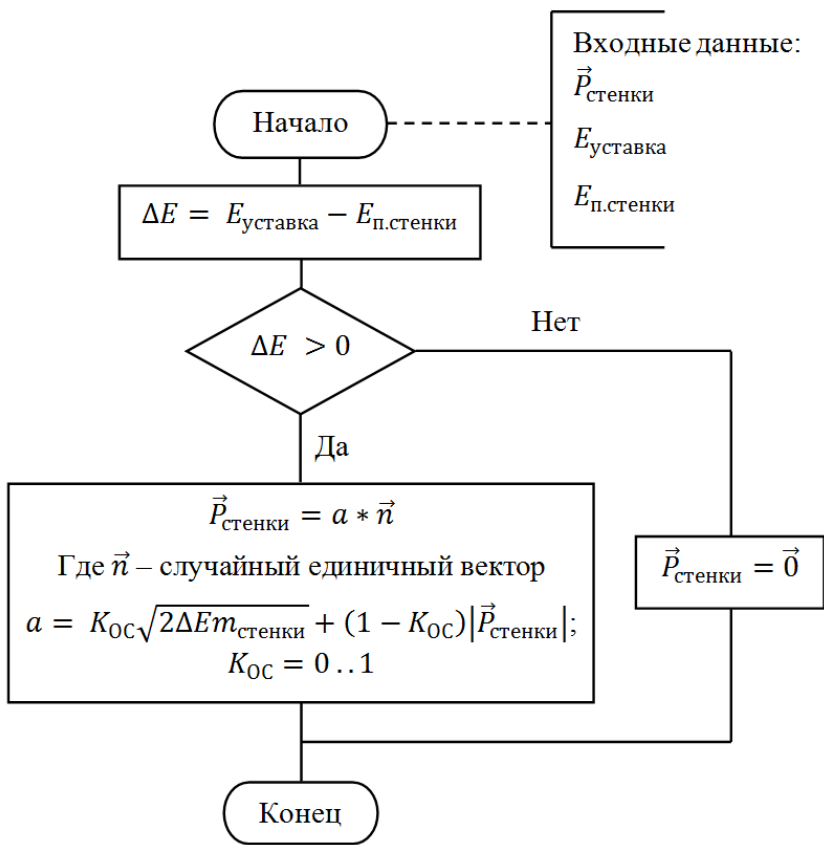


Рис. 6. Блок-схема работы цепи обратной связи и регулятора

Здесь $E_{\text{установка}}$ – энергия, соответствующая тепловым колебаниям иона твердого тела, $E_{\text{п.стенки}}$ – текущее значение потенциальной энергии колебания иона твердого тела, K_{OC} – коэффициент обратной связи, $\vec{P}_{\text{стенки}}$ – текущее значение импульса иона твердого тела.

При реализации данной схемы для системы из одной свободной частицы и одной частицы поверхности можно проследить динамику работы механизма теплообмена (рисунки 7-9).

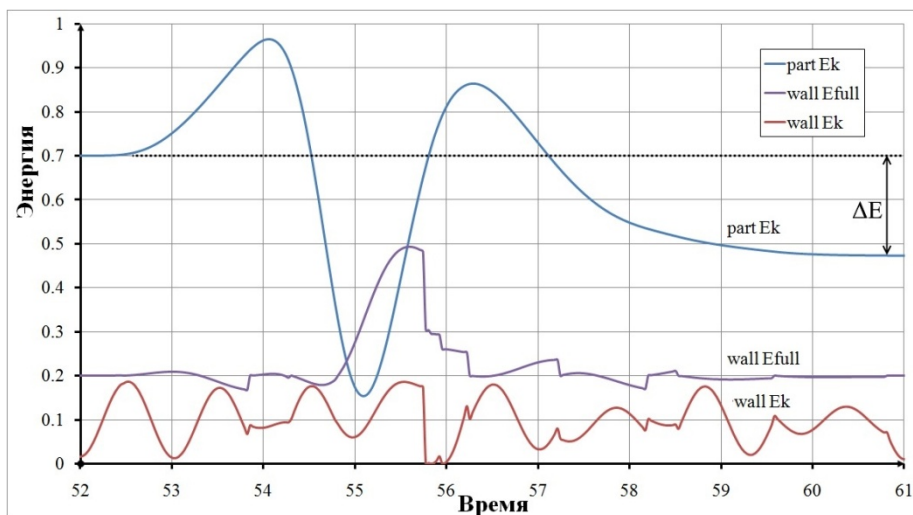


Рис. 7. Динамика единичного взаимодействия частицы с ячейкой поверхности при $E_{\text{стенки}} < E_{\text{частицы}}$

На данном рисунке видно, что за одно взаимодействие с ячейкой поверхности частица теряет некоторую энергию ΔE , при этом энергия стенки после взаимодействия возвращается к исходному состоянию. Такое поведение системы свидетельствует о протекании процесса передачи энергии от частицы газа к стенке. Скачки в кинетической энергии ячейки обусловлены работой механизма регуляции (рисунок 6).

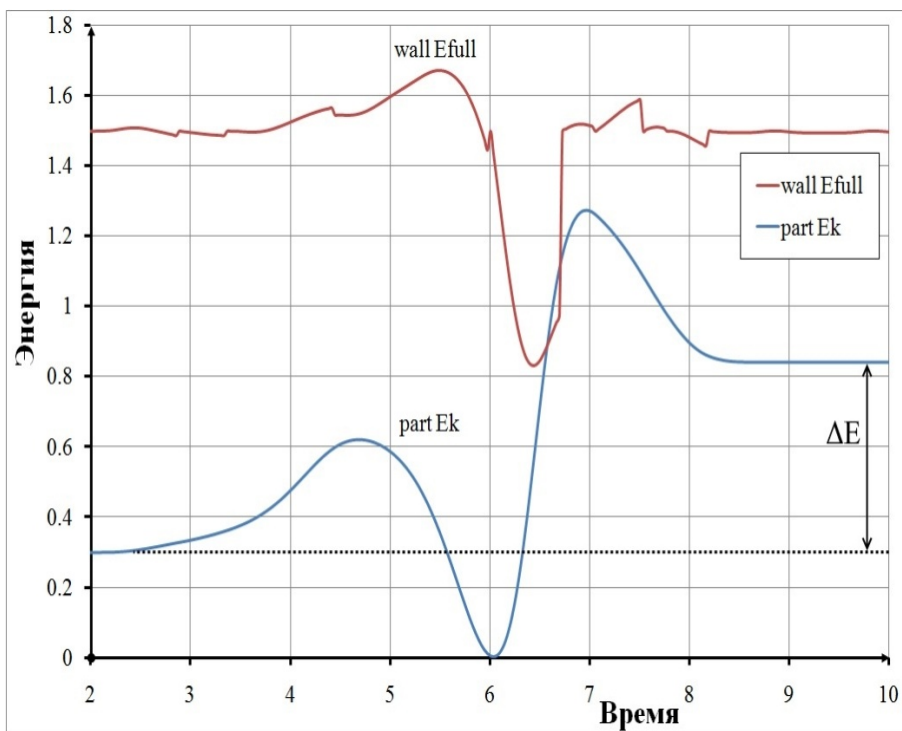


Рис. 8. Динамика единичного взаимодействия частицы с ячейкой поверхности при $E_{\text{стенки}} > E_{\text{частицы}}$

На данном графике виден процесс передачи энергии от стенки к частице газа. Это свидетельствует, что алгоритм взаимодействия описывает оба возможных направления энергообмена.

Из представленного графика (рис. 9) можно увидеть, что энергия частицы сравнивается с энергией ячейки поверхности за 6 взаимодействий, что хорошо согласуется с известными из литературы данными [46].

При использовании описанного выше механизма теплообмена на границе газ-твёрдое тело, для модельной системы в ходе численных экспериментов были получены некоторые характерные зависимости, такие как кинетические кривые адсорбции и изотермы адсорбции.

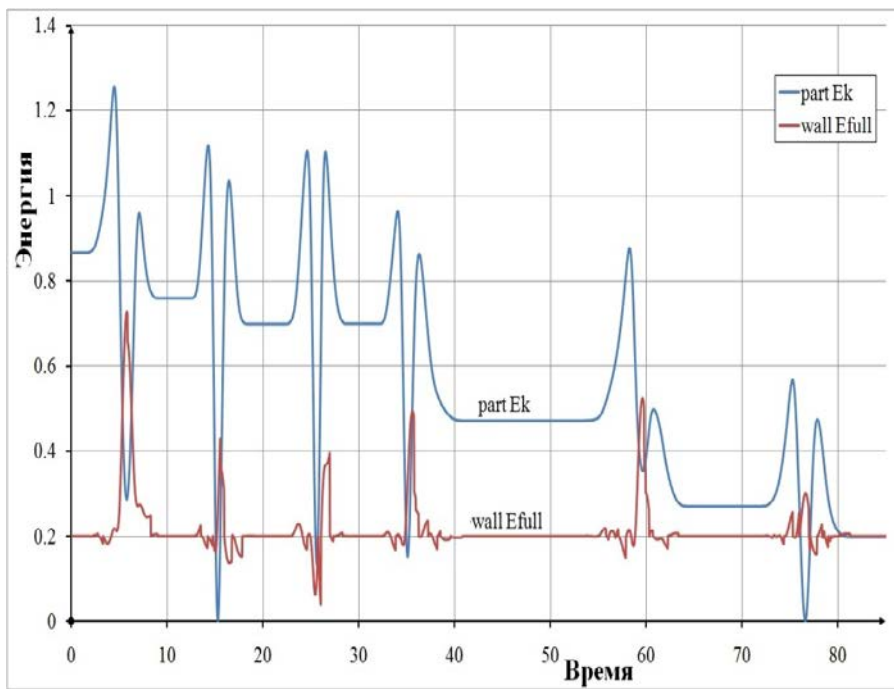


Рис. 9. Динамика установления энергетического равновесия в модельной системе

Для удобства обработки результатов моделирования был введен критерий адсорбированности частицы, заключающийся в следующем: частица является адсорбированной, если кратчайшее расстояние между частицей и поверхностью меньше четырех эквивалентных ван-дер-ваальсовых диаметров [8].

В начальный момент времени кол-во адсорбированных частиц равно нулю.

Параметры модельной системы были приняты следующими: 1) Температура = 180 К; 2) Давление = 2.04 атм. Результат единичного эксперимента приведен на рисунке 10.

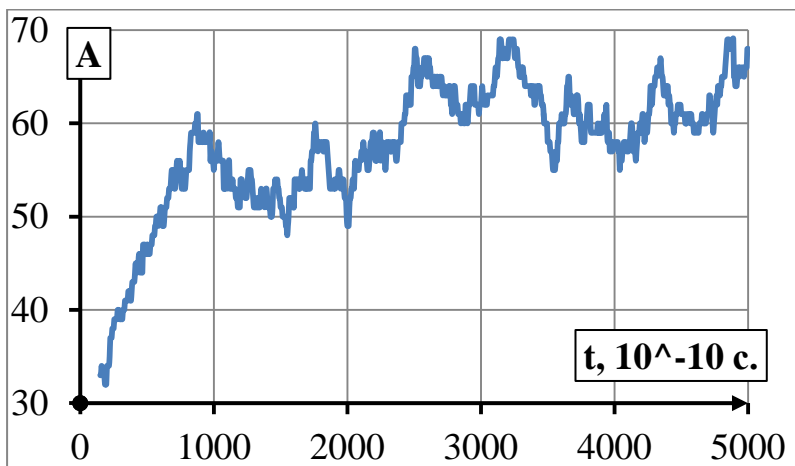


Рис. 10. Кинетическая кривая без осреднения

Представленный результат позволяет оценить размер флуктуаций, характерный для модельной системы. На рисунке 11 представлен результат осреднения по 10-ти экспериментам с аналогичными параметрами.

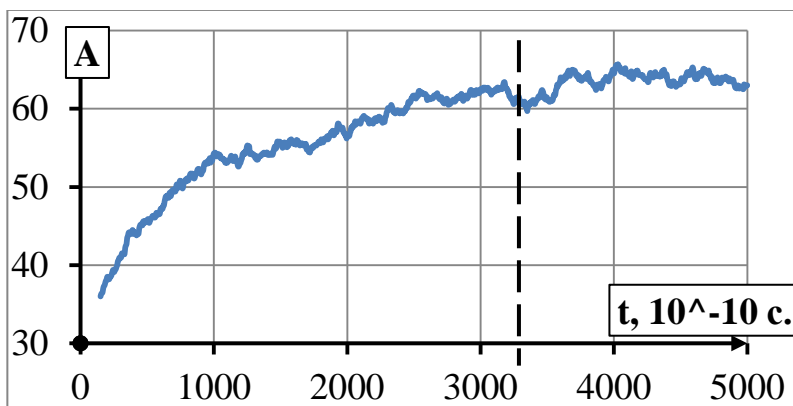


Рис. 11. Кинетическая кривая с осреднением по 10-ти опытам
Штриховой линией обозначено примерное время установления
адсорбционного равновесия

На рисунке 12 приведены кинетические кривые адсорбции, полученные при различных исходных давлениях газа.

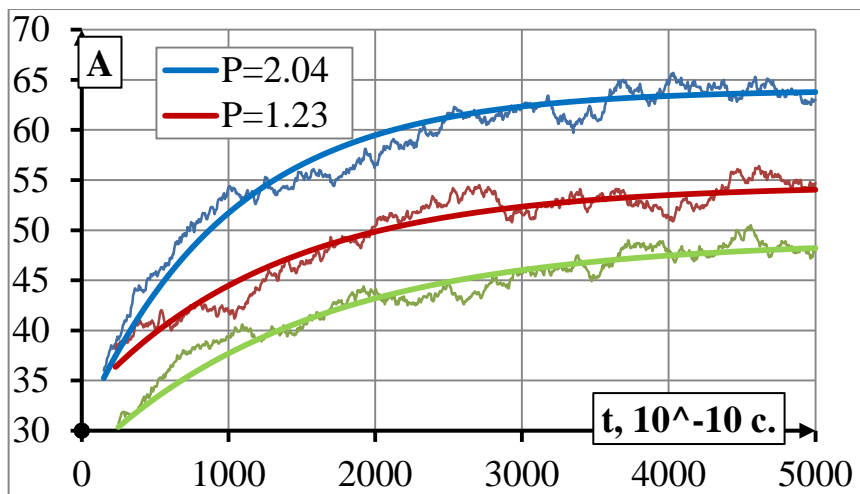


Рис. 12. Кинетические кривые при различных значениях P

Полученные результаты на качественном уровне согласуются с известными кинетическими зависимостями.

В отличие от классических методов описания кинетики адсорбции, использование ММД позволяет оценить размер термодинамических флуктуаций, а также в явном виде получить пространственные распределения частиц.

Построение кинетических кривых адсорбции позволило определить время установления адсорбционного равновесия в модельной системе. Осредняя участки кинетических кривых после установления равновесия, можно построить участки изотерм адсорбции для модельной системы, которые представлены на рисунке 13.

Отличия в виде полученных изотерм можно объяснить тем, что при снижении температуры, значение P_0 также снижается. Это приводит к

тому, что на изотерме, соответствующей температуре 120 К, наблюдается ветвь перехода к конденсации.

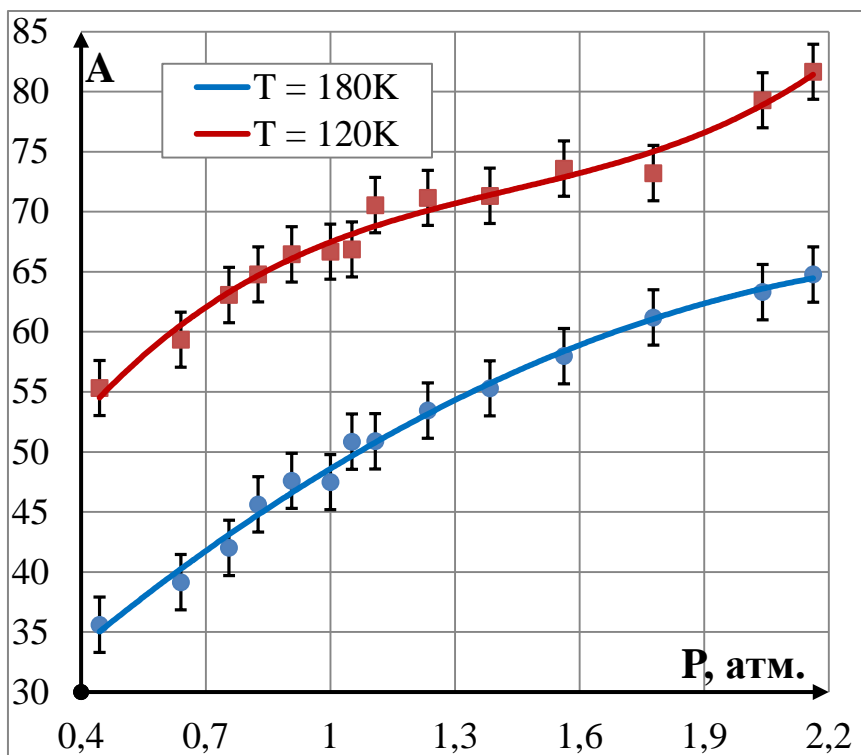


Рис. 13. Изотермы адсорбции модельной системы

Изотерма, полученная при температуре 180 К, соответствует лэнгмюровскому участку изотермы БЭТ, тогда как при температуре 120 К наблюдается качественное согласие с изотермой БЭТ 2-го типа (полимолекулярная адсорбция на непористых или макропористых адсорбентах.), что ожидаемо при принятых допущениях.

Приведенные результаты моделирования позволяют проиллюстрировать некоторые особенности ММД, такие, как органичный учет влияния термодинамических флуктуаций, возможность исследовать неравновесные состояния и переходные процессы и т. д.

1.7. Дискретность времени

Решение, получаемое в ходе ЧЭ, дискретно как по времени, так и по пространственным координатам. Это вызвано ограниченной точностью машинного расчета, однако если для последнего шага определяется только количеством разрядов в представлении числа, то первое выбирается наибольшим из тех, при которых сохраняется достаточная точность решения (погрешность определения выбранной величины не превосходит заданную). В связи с этим возникает необходимость введения критерия оценки величины Δt . Учитывая, что основное влияние на устойчивость решения и накопление погрешности в ходе ЧЭ оказывает нелинейность сил, действующих на частицы (правая часть ДУ (34)), следует выбрать критерий, исходя из формы потенциала взаимодействия.

Т. к. ПЛД является самым часто используемым при численном моделировании, рассмотрим систему из двух тел, взаимодействующих по данному потенциалу, с целью определения частоты их собственных колебаний. Для этого запишем лагранжиан:

$$L = L(\vec{r}, \vec{v}) = K - U = \frac{\mu v^2}{2} - U(\vec{r}). \quad (19)$$

Т. к. константа $(\mu v^2)/2$ не оказывает влияние на конечный вид уравнений Лагранжа, в дальнейшем ей можно пренебречь. Для ПЛД все положения равновесия выражаются следующим образом:

$$\vec{r} = r_{min} \vec{e}, \quad |\vec{e}| = 1, \quad (20)$$

где r_{min} соответствует (34). Устойчивость данных точек является следствием формы самого потенциала.

Произведем линейную интерполяцию потенциальной энергии в окрестности положения (20) при фиксированном векторе \vec{e} . Совершим замену:

$$\vec{r} = a\vec{e} + \vec{q}, \quad \vec{v} = \frac{d\vec{r}}{dt} = \frac{d\vec{q}}{dt}, \quad a = r_{min}, \quad (21)$$

где $|\vec{q}| \rightarrow 0$. Произведя разложение данных выражений в ряд Тейлора с остаточным членом в форме Пеано, получим:

$$\begin{aligned} U(\vec{r}) &= U(a\vec{e} + \vec{q}) = \\ &= U(a\vec{e}) + \sum_{i=1}^3 \frac{\partial U}{\partial r_i} \Big|_{a\vec{e}} q_i + \\ &+ \frac{1}{2} \sum_{i,j=1}^3 \frac{\partial^2 U}{\partial r_i \partial r_j} \Big|_{a\vec{e}} q_i q_j + o(|\vec{q}|^2) = \\ &= -\varepsilon + \frac{1}{2} \sum_{i,j=1}^3 \frac{\partial^2 U}{\partial r_i \partial r_j} \Big|_{a\vec{e}} q_i q_j + o(|\vec{q}|^2). \end{aligned} \quad (22)$$

Вторую смешанную производную в точке $a\vec{e}$ можно представить как

$$\frac{\partial^2 U}{\partial r_i \partial r_j} \Big|_{a\vec{e}} = J e_i e_j, \quad i, j = \overline{1..3}, \quad J = \frac{nm\varepsilon}{a^2}. \quad (23)$$

Тогда лагранжиан можно приближенно представить следующим образом:

$$L \approx \frac{1}{2} \dot{\vec{q}}^T A \dot{\vec{q}} - \frac{1}{2} \vec{q}^T C \vec{q}. \quad (24)$$

Матрицы A и C представлены в виде:

$$\begin{aligned} A &= \|a_{ij}\|_{i,j=\overline{1..3}}, \quad C = \|c_{ij}\|_{i,j=\overline{1..3}}; \\ a_{ij} &= \mu \delta_{ij}, \quad c_{ij} = J e_i e_j, \quad i, j = \overline{1..3}, \end{aligned} \quad (25)$$

где δ_{ij} – символ Кронекера. Приведем уравнение Лагранжа к линейному виду.

$$A\ddot{\vec{q}} + C\dot{\vec{q}} = 0. \quad (26)$$

Решение системы (26) выразим в виде $\vec{q}(t) = ue^{i\omega t}$. После преобразований получаем систему линейных алгебраических уравнений для определения амплитудного вектора u :

$$(-A\omega^2 + C)u = 0. \quad (27)$$

Система (27) имеет нетривиальное решение при выполнении следующего условия:

$$\det(-A\omega^2 + C) = 0. \quad (28)$$

Найдя детерминант матрицы $(-A\omega^2 + C)$, который не зависит от \vec{e} , перепишем уравнение (28)

$$\mu^2\omega^4(J - \mu\omega^2) = 0. \quad (29)$$

Как можно увидеть, выражение (29) имеет единственное нетривиальное решение для ω^2 :

$$\omega_{osc} = \sqrt{\frac{J}{\mu}} = \frac{1}{a} \sqrt{\frac{nm\varepsilon}{\mu}} = \frac{1}{r_{min}} \sqrt{\frac{2nm\varepsilon}{m_p}}, \quad (30)$$

где m_p – масса частицы, а $\mu = m_p/2$ т.к. частицы тождественны. Период колебания может быть найден по определению:

$$T_{osc} = \frac{2\pi}{\omega_{osc}} = \pi r_{min} \sqrt{\frac{2m_p}{nm\varepsilon}}. \quad (31)$$

Максимальный временной шаг τ_{max} выбирается таким образом, чтобы период собственных колебаний T_{osc} на дне потенциальной ямы был разбит хотя бы на N частей, где N принимает значения от 10 до 1000. Учитывая вышесказанное, запишем окончательный критерий:

$$\tau_{max} = \frac{T_{osc}}{N} = \frac{\pi r_{min}}{N} \sqrt{\frac{2m_p}{nm\varepsilon}} \ll T_{osc}. \quad (32)$$

Выбранный таким образом шаг по времени не только связан с физической составляющей поставленной задачи, но и гарантирует устойчивость схемы численного интегрирования [47].

1.8. Численное решение уравнений движения

Очевидно, что для качественной реализации метода МД необходимо выбрать *хороший* алгоритм численного интегрирования уравнений движения. Но, хотя определить *плохой* алгоритм достаточно просто, критерий, которому должен соответствовать *хороший* алгоритм, не является очевидным. Поэтому необходимо рассмотреть данный вопрос с нескольких разных сторон.

Несмотря на то, что на первый взгляд скорость исполнения алгоритма является ключевым критерием, на деле этот параметр не играет существенной роли при выборе, т. к. время затрачиваемое на исполнение самого алгоритма зачастую мало, по сравнению со временем расчета сил межчастичного взаимодействия, даже для относительно небольших систем.

Точность при большом шаге по времени является более важной чертой, потому что чем больше шаг по времени можно использовать, тем меньше расчетов сил нужно использовать на единицу модельного вре-

мени. Следовательно, больше преимуществ дает использование уточненных схем, позволяющих выбирать большие шаги по времени при интегрировании.

Большинство алгоритмов достигают выполнения последнего условия за счет сохранения в памяти ЭВМ информации о производных от координат более высоких порядков. Как следствие, такие схемы имеют тенденцию к использованию больших объемов памяти. Правда, для большинства ЧЭ, данная особенность не является серьезным недостатком, т. к. объем памяти, необходимый для хранения данных производных, мал в сравнении с объемом, доступным даже на обычных компьютерах.

Еще одним важным критерием является сохранение энергии при интегрировании. Тут, правда, стоит сразу выделить два вида сохранения: короткосрочный (на протяжении нескольких шагов интегрирования) и долгосрочный (на протяжении всего численного эксперимента). Более сложные схемы высоких порядков обычно хорошо сохраняют энергию на коротких промежутках времени, но при этом имеют неприятную тенденцию к дрейфу энергии на больших временах. Если же рассматривать Верле-подобные алгоритмы, то на коротких интервалах они показывают меньшую точность, но дрейф энергии в долгосрочной перспективе у них практически отсутствует.

Также может показаться, что наиболее важным для алгоритма является его точность расчета всех траекторий частиц системы как на коротких, так и на длительных промежутках времени. Но таких алгоритмов попросту не существует. На деле любая система, моделируемая методом МД, находится в таком состоянии, что фазовая траектория (в 6-ти мерном пространстве координат-импульсов) очень чувствительна к малейшим отклонениям в начальных условиях. Это означает, что две траектории, изначально очень близкие (но не одинаковые), со временем экспоненциально расходятся. Можно показать, что ошибка интегрирования, вызванная алгоритмом, соответствует небольшому отклонению рассчитанной траектории системы от *истинной*. Следовательно, можно ожидать, что любая, даже самая малая, ошибка вызовет экспоненциальное расхожде-

ние расчетной фазовой траектории системы от *истинной*, соответствующей тем же начальным условиям. Данное явление, называемое неустойчивостью Ляпунова, на первый взгляд является разрушительным для всей идеи молекулярной динамики, но есть причины не считать серьезным проблемой даже его.

Последнее утверждение требует некоторых разъяснений. Во-первых, целью молекулярно-динамического расчета не является определение *в точности* того, что случится с системой, находившейся в *точно* определенных начальных условиях. Наиболее важными являются статистические предсказания. Интересным является моделирование среднего поведения системы, о начальном состоянии которой известно что-то (например, полная энергия), но не все. В данном отношении, метод молекулярной динамики фундаментально отличается от, например, численных схем для расчета траекторий полета спутников, которые необходимо вычислять точно. В последнем случае нет возможности запустить множество спутников, сделав только статистическое предсказание того, куда они в конечном счете попадут. По-другому дело обстоит при МД расчете, для которого статистического предсказания более чем достаточно. При этом неточные фазовые траектории, рассчитанные численно, в какой-то степени, все-таки оказываются близки к истинной.

Хотя последнее утверждение не имеет строгого доказательства, существует масса свидетельств в пользу его истинности [47]. Исходя из этого, можно предположить, что существуют, так называемые, теньевые траектории, являющиеся истинными для системы многих тел. Также такие траектории оказываются близки к численно рассчитанным на временных интервалах больших, чем время развития неустойчивости Ляпунова. Следовательно, результат моделирования является одной из истинных фазовых траекторий, пусть даже нельзя априори сказать, какой именно.

В заключение, стоит отметить, что, даже при использовании идеальной схемы интегрирования, будет иметь место ошибка округления, т.к. при расчете используются вычисления с плавающей точкой, имеющие конечную точность.

1.9. Алгоритмы и схемы интегрирования

ММД основывается на численном решении системы дифференциальных уравнений движения вида:

$$\frac{d^2\vec{r}_i}{dt^2} = -\frac{1}{m_i}\vec{\nabla}U(\vec{r}_1, \dots, \vec{r}_N), \quad (33)$$

$$\frac{d^2\vec{r}_i}{dt^2} = \frac{1}{m_i} \sum_{\substack{j=1..N \\ j \neq i}}^N \left(\frac{\vec{r}_j}{|\vec{r}_{ij}|} F(|\vec{r}_{ij}|) \right), \quad (34)$$

$$\vec{f}_i = \frac{1}{m_i} \sum_{\substack{j=1..N \\ j \neq i}}^N \left(\frac{\vec{r}_j}{|\vec{r}_{ij}|} F(|\vec{r}_{ij}|) \right). \quad (35)$$

Для численного решения данной системы необходимо представить уравнение (34) с учетом равенства (35) в виде системы уравнений первого порядка следующим образом:

$$\begin{cases} \frac{d\vec{r}_i}{dt} = \vec{v}_i \\ \frac{d\vec{v}_i}{dt} = \vec{f}_i \end{cases} \quad (36)$$

Как можно увидеть из (35), основная вычислительная нагрузка приходится на расчет результирующей силы парного взаимодействия, следовательно, алгоритмы для решения данной системы не должны производить более одного расчета правой части на итерацию.

Для итерационной схемы верны следующие равенства:

$$\vec{v}_{t+\Delta t} = \vec{v}(t + \Delta t), \quad (37)$$

$$\vec{r}_{t+\Delta t} = \vec{r}(t + \Delta t). \quad (38)$$

Далее, разлагая (37) и (38) в ряд Тейлора, можно получить выражения для всех нижеприведенных схем.

Начнем рассмотрение с самой простой схемы интегрирования, а именно с алгоритма Эйлера.

$$\vec{v}_{t+\Delta t} = \vec{v}_t + \vec{a}_t \Delta t + O(\Delta t), \quad (39)$$

$$\vec{r}_{t+\Delta t} = \vec{r}_t + \vec{v}_t \Delta t + O(\Delta t). \quad (40)$$

Как показано в (39, 40), алгоритм Эйлера имеет первый порядок точности по координатам и скоростям. Главным недостатком данной схемы является асимметричность относительно шага интегрирования, т. е. при продвижении решения на временной шаг dt используется информация о производных только в начальной точке интервала, что ведет к активному накоплению погрешности. Данная схема приведена, преимущественно, в качестве примера, так как на практике не используется.

Следующим рассмотрим модификацию Кромера для алгоритма Эйлера (далее алгоритм Эйлера-Кромера).

$$\vec{v}_{t+\Delta t} = \vec{v}_t + \vec{a}_t \Delta t + O(\Delta t), \quad (41)$$

$$\vec{r}_{t+\Delta t} = \vec{r}_t + \vec{v}_{t+\Delta t} \Delta t + O(\Delta t). \quad (42)$$

Хотя схема (41, 42) имеет, аналогично (39, 40), первый порядок точности, она является симметричной, что способствует меньшему накоплению погрешности и дает более устойчивые решения для колебательных систем. Данный алгоритм может быть использован в демонстрационных целях, например, в рамках лабораторных или курсовых студенческих работ.

Последней схемой первого порядка точности является метод центральной точки.

$$\vec{v}_{t+\Delta t} = \vec{v}_t + \vec{a}_t \Delta t + O(\Delta t), \quad (43)$$

$$\vec{r}_{t+\Delta t} = \vec{r}_t + (\vec{v}_{t+\Delta t} + \vec{v}_t) \frac{\Delta t}{2} + O(\Delta t^2). \quad (44)$$

В отличие от алгоритмов Эйлера и Эйлера-Кромера, данный метод имеет второй порядок точности определения координаты для систем с малыми значениями производной ускорения по времени.

Перейдем к рассмотрению более точных схем интегрирования. Наиболее простой и распространенной из них является алгоритм Верле в скоростной форме.

$$\vec{r}_{t+\Delta t} = \vec{r}_t + \vec{v}_t dt + \frac{1}{2} \vec{a}_t \Delta t^2 + O(\Delta t^3), \quad (45)$$

$$\vec{v}_{t+\Delta t} = \vec{v}_t + (\vec{a}_t + \vec{a}_{t+\Delta t}) \frac{\Delta t}{2} + O(\Delta t^2). \quad (46)$$

Основным достоинством данной схемы является простота реализации. Как можно заметить из (46), для корректной работы необходимо сохранять в памяти ЭВМ дополнительные значения ускорений. Данная схема может применяться для реализации численных экспериментов, направленных на выявление статистических или макроскопических явлений.

Последней рассмотренной в данной работе схемой рассматривается алгоритм Бимана в явной форме.

$$\vec{r}_{t+\Delta t} = \vec{r}_n + \vec{v}_t dt + \frac{1}{6} (4\vec{a}_t - \vec{a}_{t-\Delta t}) \Delta t^2 + O(\Delta t^4), \quad (47)$$

$$\vec{v}_{t+\Delta t} = \vec{v}_t + (4\vec{a}_{t+\Delta t} + 5\vec{a}_t - \vec{a}_{t-\Delta t}) \frac{\Delta t}{6} + O(\Delta t^3). \quad (48)$$

Как видно из (47, 48) алгоритм Бимана определяет координату с четвертым порядком точности, а скорость – с третьим, таким образом, алгоритм имеет общую погрешность третьего порядка. Также стоит отметить, что достоинством алгоритма является хорошее сохранение энергии и низкая чувствительность к ошибкам округления благодаря тому, что в алгоритме вычисляется разность значительно отличающихся чисел ($4a_t - a_{t-\Delta t}$). Он может применяться в случаях, когда весомое значение

имеют фазовые траектории отдельных частиц системы. Главным недостатком данной схемы является то, что она не обладает самостартуемостью, т. к. для вычисления состояния в последующий момент времени используются значения в предыдущий момент. Для решения данной проблемы первый шаг интегрирования обычно производят менее точными алгоритмами, например, Эйлера-Кромера или Верле [48].

Более подробно рассмотрим последнюю схему интегрирования. Для этого значение функции, ее четных и нечетных производных определим на разных временных слоях [49], как это показано на рисунке 14. На основании этого:

$$x_{t+\Delta t} = x_t + v_{t+\frac{1}{2}\Delta t} \Delta t, \quad (49)$$

$$v_{t+\frac{1}{2}\Delta t} = v_t + \frac{1}{2} a_{t+\frac{1}{4}\Delta t} \Delta t. \quad (50)$$

Значение $a_{t+\frac{1}{4}\Delta t}$ экстраполируем по значениям ускорения с двух предыдущих временных шагов ($a_{t-\Delta t}$ и a_t):

$$a_{t+\frac{1}{4}\Delta t} = \frac{1}{4} (5a_t - a_{t-\Delta t}). \quad (51)$$

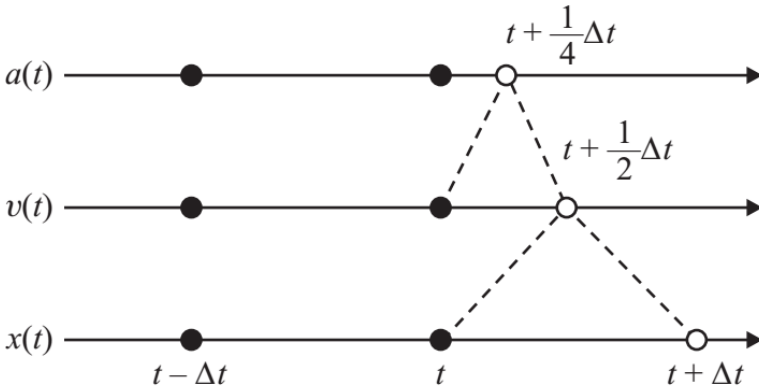


Рис. 14. Расположение временных шагов, используемое в рассматриваемой конечно-разностной схеме

Определим порядок точности данного алгоритма. Для этого выразим с помощью ряда Тейлора основные выражения. Для выражения (49) представим x_t и $x_{t+\Delta t}$ в следующем виде:

$$x_{t+\Delta t} = x_{t+\frac{1}{2}\Delta t+\frac{1}{2}\Delta t} = x_{t+\frac{1}{2}\Delta t} + \frac{1}{2}v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{8}a_{t+\frac{1}{2}\Delta t}\Delta t^2 + \frac{1}{48}b_{t+\frac{1}{2}\Delta t}\Delta t^3 + O(\Delta t^4), \quad (52)$$

$$x_t = x_{t+\frac{1}{2}\Delta t-\frac{1}{2}\Delta t} = x_{t+\frac{1}{2}\Delta t} - \frac{1}{2}v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{8}a_{t+\frac{1}{2}\Delta t}\Delta t^2 - \frac{1}{48}b_{t+\frac{1}{2}\Delta t}\Delta t^3 + O(\Delta t^4). \quad (53)$$

Вычтем (53) из (52):

$$x_{t+\Delta t} - x_t = v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{24}b_{t+\frac{1}{2}\Delta t}\Delta t^3 + O(\Delta t^4). \quad (54)$$

Выполним аналогичные операции для выражения (50)

$$\begin{aligned} v_{t+\frac{1}{2}\Delta t} &= v_{t+\frac{1}{4}\Delta t+\frac{1}{4}\Delta t} = \\ &= v_{t+\frac{1}{4}\Delta t} + \frac{1}{4}a_{t+\frac{1}{4}\Delta t}\Delta t + \frac{1}{32}b_{t+\frac{1}{4}\Delta t}\Delta t^2 + O(\Delta t^3), \end{aligned} \quad (55)$$

$$\begin{aligned} v_t &= v_{t+\frac{1}{4}\Delta t-\frac{1}{4}\Delta t} = \\ &= v_{t+\frac{1}{4}\Delta t} - \frac{1}{4}a_{t+\frac{1}{4}\Delta t}\Delta t + \frac{1}{32}b_{t+\frac{1}{4}\Delta t}\Delta t^2 + O(\Delta t^3). \end{aligned} \quad (56)$$

После вычитания получим:

$$v_{t+\frac{1}{2}\Delta t} - v_t = a_{t+\frac{1}{4}\Delta t}\Delta t + O(\Delta t^3). \quad (57)$$

Для (50) имеем:

$$a_{t+\frac{1}{4}\Delta t} = a_t + \frac{1}{4}b_t\Delta t + O(\Delta t^2), \quad (58)$$

$$a_{t-\Delta t} = a_t - b_t\Delta t + O(\Delta t^2). \quad (59)$$

Используя выражения (58) и (59), выразим $a_{t+\Delta t}$, умножив (58) на 3 и прибавив (59). В результате получим следующее выражение:

$$a_{t+\frac{1}{4}\Delta t} = \frac{1}{3}(4a_t - a_{t-\Delta t}) - \frac{1}{21}b_t\Delta t + O(\Delta t^2). \quad (60)$$

Подставляя (60) в (57), а результат в (54) получим выражение для предсказания координаты частицы, соответствующее алгоритму Бимана (47) [51].

Аналогичным образом можно показать порядок точности выражения для определения скорости.

Часть 2. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОГРАММЫ МОДЕЛИРОВАНИЯ

2.1. Общие сведения

Чтобы вышеописанные уравнения превратить в работающую программу моделирования, их необходимо переписать в виде алгоритмов, реализованных на каком-либо языке программирования. Каждый алгоритм может быть представлен в виде отдельной функции. В данном случае и ниже термин «функция» будет употребляться не в алгебраическом смысле, а согласно определению информатики, как некоторый независимый, изолированно исполняемый алгоритм по обработке переданных в него числовых данных.

Любая программа представляет собой совокупность обменивающихся данными функций – в результате такого обмена получается искомым результат ЧЭ. Объем вычислений и, соответственно, время работы каждой функции зависит от вида и количества обрабатываемых данных. В первом приближении это оценивается по количеству членов суммы в вышеприведенных дискретных формулах. Двойное суммирование дает количество расчетов, пропорциональное произведению пределов этих сумм.

Например, количество расчетов межчастичных расстояний в процессе генерации фазовой траектории квадратично растет с числом частиц. Интегрирование уравнений движения, являющееся неотъемлемой частью работы по генерации фазовой траектории, и ее последующая обработка для получения различных динамических характеристик требует количества расчетов, линейно возрастающих с числом частиц, но квадратично по числу шагов фазовой траектории.

Для получения значимых в настоящее время результатов ЧЭ пределы суммирования, определяемые количеством частиц в системе, должны быть не менее тысячи. Соответственно, различие во времени исполнения

различных функций также будет порядка тысячи. Поэтому общая производительность программы определяется этими объемными функциями и именно для них необходимо писать алгоритмы, использующие все ресурсы современных персональных компьютеров – о чем и будет идти речь ниже.

Все описанные функции могут работать только в совокупности с некоторым минимумом других, конечно менее критичных с точки зрения производительности функций, поставляющих для них исходные данные и обрабатывающих полученные. В следующем разделе будет описан минимальный набор функций, который в совокупности представляет собой основу программы МД.

Для работы на персональном компьютере существуют среды разработки программных продуктов на основе различных языков программирования: фортран, Бейсик, паскаль, Ява, несколько разновидностей С++ и т. д. Для написания программы МД, в принципе, можно использовать любой из них, согласно предпочтениям разработчика, но с разным эффектом, поскольку каждый из языков разрабатывался для каких-то определенных целей. С этих позиций, автор отдает предпочтение С++, как наиболее гибкому средству программирования, алгоритмы которого, к тому же, не сильно отличаются от привычных физики алгебраических конструкций. Других оснований для выбора нет, поскольку все множество языков программирования постоянно развивается.

В данном пособии автором была выбрана среда разработки MicrosoftVisualStudio 2013.

Все нижеописанные функции С++ предложены в качестве состава общей программы. При самостоятельном выполнении желательно предусмотреть графический или файловый интерфейс, через которые можно вводить числовые значения исходных параметров системы, как-то ее размер, количество простых бесструктурных частиц и/или цепей, концентрацию последних, коэффициент упаковки и т. д. В противном случае

придется для изменения параметров численного эксперимента каждый раз перекомпилировать программу.

Создание такого интерфейса выходит за рамки данного пособия и, к тому же, является предметом рассмотрения множества учебных пособий по программированию или описанию интегрированных сред программирования.

Графический вывод (если таковой будет иметься) должен быть достаточно редким, чтобы не понижать производительность программы, или по мере надобности вызываться какой-то клавишей клавиатуры или мышью.

Графический интерфейс удобен в работе, но не обязателен, а вот периодический цифровой контроль интегралов движения – средней кинетической энергии частиц, потенциальной энергии, давления и т. п. необходим, как автоматически внутри программы, так и визуально. Так же важна информация о выводе количества шагов интегрирования – основной характеристике развития системы и длины фазовой траектории. Иногда именно заданная длина и является сигналом к автоматическому оставанию программы.

Прежде, чем перейти к рассмотрению алгоритмов МД, необходимо написать и отладить класс, содержащий методы, генерирующие модельную систему и производящие интегрирование уравнений движения – лишь при наличии этих методов программа работоспособна.

Листинги всех методов снабжены необходимыми минимальными пояснениями, так как их смысл определяется назначением и представляет собой алгоритмическое изложение формул предыдущего раздела, как и большинство используемых символов, исключая греческие, остальные – поясняются короткими комментариями после символа `</>`. Комментарии даны на английском, чтобы гарантированно читаться в любой операционной системе и любой интегрированной среде программирования, куда они напрямую могут быть скопированы из данного текста.

Каждый листинг удобно поместить в отдельный файл, имя которого совпадает с именем определяемого в нем метода, тогда между ними можно быстро переключаться в интегрированной среде разработки, любая из которых рекомендует большие программы представлять в виде совокупности отдельных файлов, образующих проект.

Начнем рассмотрение проекта МД с класса и составляющих его методов, которые могут вызываться как из основной функции проекта, так и друг из друга. Распределение алгоритмов по методам позволяет отлаживать их совершенно независимо друг от друга, посылая в них из основной функции какие-то числовые данные, результат обработки которых известен и может быть выведен либо на дисплей, либо в файл в виде, удобном для последующей обработки. Процесс отладки должен начинаться с метода, который генерирует систему, то есть готовит исходные данные для обработки всеми остальными методами – с него и начнем описание алгоритмов на языке C++ в следующем разделе.

1. Считывание параметров, которые определяют условия запуска (начальная температура, число частиц, плотность, шаг по времени).
2. Инициализация системы.
3. Вычисление силы для всех частиц.
4. Интегрирование уравнения движений Ньютона для частиц системы. (Этот шаг и предыдущий составляют ядро моделирования. Они повторяются, пока не будет рассчитана эволюция системы для заданной продолжительности времени).
5. После завершения центрального цикла, вычисляются и записываются средние значения измеряемых величин, и происходит остановка программы.

Программа построена следующим образом:

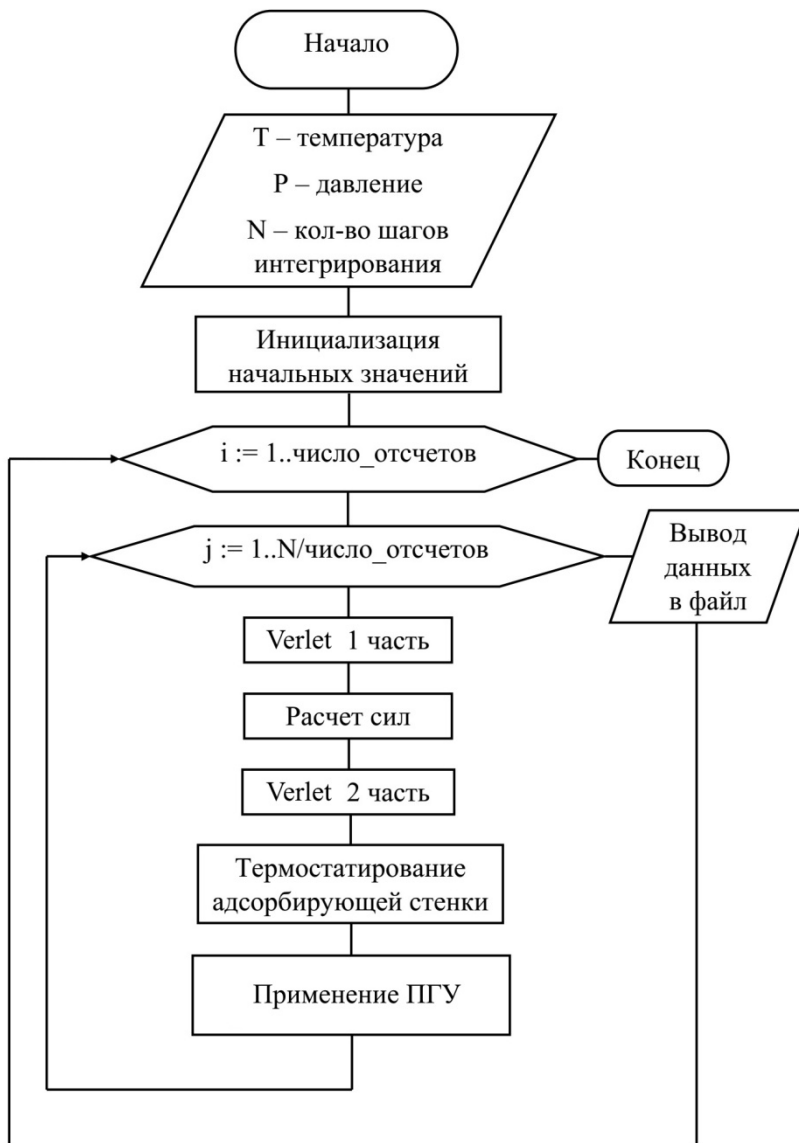


Рис. 15. Блок-схема ЧЭ

2.2. Основная функция C++

В листинге 2.7 приведена точка входа в программу, откомпилированную в консольном стиле, который задается в интерактивной среде программирования для программ, не имеющих графического интерфейса. Программа исполняется до тех пор, пока не дойдет до конца функции *main*, исполнив все вызываемые в ней функции.

Листинг 2.7. main.cpp

```
1. #include "md.h"
2. #define n 5 // number of particles along base box side
3.
4. void main(){
5.     md m = new md_System(n);
6.     m->Init(20, 2, 1, 12, 200, 0.01);
7.     m->Calc(100000, 1000);}
```

В 5-ой строке происходит вызов конструктора класса. После, при помощи оператора выбора элемента по указателю, осуществляется обращение к методу *Init*. В круглых скобках задаются необходимые параметры исследуемой системы. В 7-ой строке происходит активация цикла расчета. Первый параметр – количество шагов фазовой траектории; второй – частота записи в файл полученных результатов.

2.3. Генерация системы и начальных условий

Чтобы начать моделирование, необходимо задать начальные координаты и скорости всех частиц в системе. Положения частиц должны быть построены под структуру, что будет смоделирована. В любом случае, частицы не должны быть расположены в положениях, которые приводят к значительному перекрытию атомных или молекулярных ядер. Часто это достигается путем первоначального размещения частицы на кубической решетке.

Инициализация класса и генерация начальных условий представлена в листинге 2.1. Строки 6, 7 отвечают за инициализацию констант класса. Конструктор класса, определенная в строке 9, выделяет память под внутренние массивы и инициализирует встроенный генератор случайных чисел. Деструктор выполняет удаление массивов из памяти (определение в строке 22).

Метод *Init*, определенный в строке 27, выполняет инициализацию начальных условий исходя из переданных в него параметров. Так как используемый для интегрирования алгоритм Верле не является самостартующим, после инициализации происходит расчет ускорений частиц методом *calcAcel* (строка 46), описание которого будет рассмотрено позже.

Листинг 2.1. Init.cpp

```
1. #include "md.h"  
2. #include<stdlib.h>  
3. #include<stdio.h>  
4. #include<time.h>  
5.  
6. const TYPE md_System::R = 8.314462; // universal gas constant
```

```

7.  const TYPE md_System::X = 0.1; // correction factor
8.
9.  md_System::md_System(int _N){
10.     N = _N;      n = _N*_N*_N;
11.     srand(time(0)); for(inti = 0; i < n; i++) rand();
12.     x = (TYPE*)calloc(n,sizeof(TYPE));
13.     y = (TYPE*)calloc(n,sizeof(TYPE));
14.     z = (TYPE*)calloc(n,sizeof(TYPE));
15.     Vx = (TYPE*)calloc(n,sizeof(TYPE));
16.     Vy = (TYPE*)calloc(n,sizeof(TYPE));
17.     Vz = (TYPE*)calloc(n,sizeof(TYPE));
18.     ax = (TYPE*)calloc(n,sizeof(TYPE));
19.     ay = (TYPE*)calloc(n,sizeof(TYPE));
20.     az = (TYPE*)calloc(n,sizeof(TYPE));}
21.
22. md_System::~~md_System(){
23.     delete[] x; delete[] y; delete[] z;
24.     delete[] Vx; delete[] Vy; delete[] Vz;
25.     delete[] ax; delete[] ay; delete[] az;}
26.
27. void md_System::Init(TYPE _mass, TYPE _rVan, TYPE _E, TYPE
    _L, TYPE _T, TYPE _dt){

```

```

28.      dt = _dt; dt1 = 0.5*_dt; dt2 = 0.5*_dt*_dt;
29.      massA = _mass; rVan = _rVan; T = _T; E = _E;
30.      L = _L; rCut = 3*_rVan; L1 = _L/ - rCut;
31.      TYPE norm = 2.0/(TYPE)RAND_MAX;
32.      TYPE Vm = sqrt(8*R*T/(M_PI*massA));
33.      for (inti = 0; i < n; i++){
34.          Vx[i] = Vm*log(norm*rand() + 0.5);
35.          Vy[i] = Vm*log(norm*rand() + 0.5);
36.          Vz[i] = Vm*log(norm*rand() + 0.5);    }
37.      corrVeloc();
38.      TYPE s = L/N, s1 = s/2;
39.      for (inti = 0; i < N; i++)          {
40.          for (int j = 0; j < N; j++)      {
41.              for (int k = 0; k < N; k++)  {
42.                  int index = i*N*N + j*N + k;
43.                  x[index] = s1 + s*i;
44.                  y[index] = s1 + s*j;
45.                  z[index] = s1 + s*k;    }      }
46.      }
46.      calcAcel();}

```

Встроенные генераторы случайных чисел не дают идеального равномерного распределения на заданном интервале, тем более для небольшой выборки, поэтому для коррекции этой неидеальности необходимо перемасштабировать скорости частиц к заданной температуре, согласно формуле (1.21), вызовом метода *corrVeloc* (из листинга 2.2.) в строке 37.

В рассмотренном и нижеследующих листингах для простоты опущено множество перекрестных проверок переменных, которые разработчик программы вводит по своему усмотрению. Перед инициализацией переменных, полученных из графического или файлового интерфейса, обязательно производится проверка на совместимость их числовых значений. Кроме того, надо помнить, что они уже были инициализированы при входе в программу, и решить какой инициализации отдать приоритет.

Листинг 2.2. CorrVeloc.cpp

```

1. #include «md.h»
2.
3. voidmd_System::corrVeloc(){
4.     TYPE Vsx = 0, Vsy = 0, Vsz = 0, _n = 1/(TYPE)n;
5.     for (54nti = 0; i < n; i++)    {
6.         TYPE D = sqrt(Wo/(Vx[i]*Vx[i] + Vy[i]*Vy[i] +
7.         Vz[i]*Vz[i]));
8.         Vx[i] *= D; Vy[i] *= D; Vz[i] *= D;
9.         Vsx += Vx[i]; Vsy += Vy[i]; Vsz += Vz[i];    }
10.    Vsx *= _n; Vsy *= _n; Vsz *= _n;
11.    for (54nti = 0; i < n; i++)    {
12.        Vx[i] -= Vsx; Vy[i] -= Vsy; Vz[i] -= Vsz;    }
13.    }

```

В листинге 2.3. представлена функция расчета ускорений частиц. Строка 3 реализует расчет силы межчастичного взаимодействия. При расстоянии между частицами меньше $rVan$ происходит расчет абсолютно упругого удара (строки 22-28).

Листинг 2.3. calcAcel.cpp

```

1. #include «md.h»
2. #define sign(a) ( (a<0)?-1:1 )
3. #define A(r, sigma, E) (
   24*E*pow(sigma,6)*(2*pow(sigma,6)*pow(r,-13) - pow(r,-7)) )
4.
5. voidmd_System::calcAcel(){
6.     static TYPE a_rVan = A(rVan, rVan, E);
7.     for (inti = 0; i < n; i++){ ax[i] = 0; ay[i] = 0; az[i] = 0; }
8.     for (inti = 0; i < n-1; i++)      {
9.         for (int j = i+1; j < n; j++)  {
10.            TYPE dx = x[j] - x[i];
11.            if (fabs(dx) > L1) dx -= sign(dx)*L;
12.            if (fabs(dx) > rCut) continue;
13.            TYPE dy = y[j] - y[i];
14.            if (fabs(dy) > L1) dy -= sign(dy)*L;
15.            if (fabs(dy) > rCut) continue;
16.            TYPE dz = z[j] - z[i];

```

```

17.         if (fabs(dz) > L1) dz -= sign(dz)*L;
18.         if (fabs(dz) > rCut) continue;
19.         TYPE r = sqrt(dx*dx + dy*dy + dz*dz);
20.         if (r > rCut) continue;
21.         TYPE Nx = dx/r, Ny = dy/r, Nz = dz/r;
22.         if (r < rVan) {
23.             TYPE proj = Nx*(Vx[j] - Vx[i]) +
Ny*(Vy[j] - Vy[i]) + Nz*(Vz[j] - Vz[i]),
24.             dr = (rVan - r)/2;
25.             x[i] -= Nx*dr; y[i] -= Ny*dr; z[i] -=
Nz*dr;
26.             x[j] += Nx*dr; y[j] += Ny*dr; z[j]
+= Nz*dr;
27.             Vx[i] += Nx*proj; Vy[i] +=
Ny*proj; Vz[i] += Nz*proj;
28.             Vx[j] -= Nx*proj; Vy[j] -= Ny*proj;
Vz[j] -= Nz*proj; }
29.         else {
30.             TYPE a = A(r, rVan, E);
31.             ax[i] -= a*Nx; ay[i] -= a*Ny; az[i] -
= a*Nz;
32.             ax[j] += a*Nx; ay[j] += a*Ny; az[j]
+= a*Nz;
33.         } } } }

```


2.4. Алгоритм численного интегрирования уравнений движения на C++

Метод *Verle* листинга 2.4. представляет пример численного интегрирования уравнений движения с помощью скоростного алгоритма Верле (1.8 – 1.10).

Листинг 2.4. Verle.cpp

```
1. #include "md.h"
2.
3. void md_System::Verle(){
4.     for (inti = 0; i < n; i++){
5.         x[i] += Vx[i]*dt + ax[i]*dt2;
6.         y[i] += Vy[i]*dt + ay[i]*dt2;
7.         z[i] += Vz[i]*dt + az[i]*dt2;
8.         if (x[i] < 0) x[i] += L;
9.         if (x[i] > L) x[i] -= L;
10.        if (y[i] < 0) y[i] += L;
11.        if (y[i] > L) y[i] -= L;
12.        if (z[i] < 0) z[i] += L;
13.        if (z[i] > L) z[i] -= L;
14.        Vx[i] += ax[i]*dt1;
15.        Vy[i] += ay[i]*dt1;
16.        Vz[i] += az[i]*dt1;    }
```

```

17.     calcAcel();
18.     for (inti = 0; i < n; i++){
19.         Vx[i] += ax[i]*dt1;
20.         Vy[i] += ay[i]*dt1;
21.         Vz[i] += az[i]*dt1; } }

```

Для организации цикла вычислений и записи в файл результатов используется метод *Calc* (листинг 2.5.). Входными параметрами метода являются количество шагов интегрирования и частота вывода данных. В цикле, объявленном в 11 строке, осуществляется запись в файл “out.txt” координат и скоростей каждой частицы. Каждый раз значения последовательно записываются в одну строку с разделением знаком “;”.

Листинг 2.5. Calculation.cpp

```

1. #include "md.h"
2. #include <fstream>
3.
4. void md_System::Calc(int nSteps, int snap){
5.     inti, k; numStep = nSteps;
6.     std::fstream f("out.txt", std::ios_base::out);
7.     for (k = i = Step = 0; Step < nSteps; Step++, i++, k++){
8.         Verle();
9.         if (i == 99) { i = 0; corrVeloc(); }

```

```

10.             if (k == snap-1) { k = 0;
11.                 for (int j = 0; j < n; j++){
12.                     f<<x[j]<<" "<<y[j]<<" "<<z[j]<<" ";
13.
14.                     f<<Vx[j]<<" "<<Vy[j]<<" "<<Vz[j]<<" ";}
15.                 f<<std::endl; }
16.             f.close(); }

```

2.5. Заголовочный файл (объявление класса и методов)

Чтобы все вышеописанные методы класса были распознаны компилятором, необходимо добавить к ним, так называемые, заголовочные файлы компилятора, которые должны быть подключены в тексте программы до начала работы с классом.

Листинг 2.6. md.h

```

1. #pragma once
2. #define _USE_MATH_DEFINES
3. #include <math.h>
4.
5. #ifndef TYPE
6. #define TYPE float

```

```

7. #endif // !TYPE
8.
9. typedef TYPE* DynamicArray;
10.
11. typedef class md_System{
12.     md_System() {}
13.     static const TYPE X; // correction factor
14.     int N, // number of particles along base box
        side
15.     n, // general number of particles in the
        system = N*N*N
16.     numStep, // number of integration steps
17.     Step; // current integration step
18.     TYPE massA, // mass of the simple particle or atom
        "A" type
19.     rVan, // Van-der-Waalse diameter of the
        chain particles
20.     E, // chain's particles interaction en-
        ergy
21.     rCut, // cut off radius
22.     L, L1,
23.     dt, dt1, dt2, // time step and its parts for integra-
        tion

```

```

24.          Wo,          // kinetic energy of one particle
25.          WkA,        // kinetic energy of all atoms
26.          T;          // system temperature
27.          DynamicArray x, y, z,
28.          Vx, Vy, Vz,
29.          ax, ay, az;
30.          void corrVeloc();
31.          void calcAcel();
32.          void Verle();
33. public:
34.          static const TYPE R;
35.          md_System(int N);
36.          ~md_System();
37.          void Calc(intnSteps, int snap);
38.          void Init(TYPE _mass, TYPE _rVan, TYPE _E, TYPE _L,
TYPE _T, TYPE _dt);
39.          TYPE progress() { return ((TYPE)(Step+1))/((TYPE)num-
Step); };
40. }*md;

```

Заключение

В рамках данного пособия рассмотрены основы метода классической молекулярной динамики и его практической реализации.

На примере физической адсорбции показаны основные особенности разработки на базе классического метода МД моделей физико-химических процессов. Показан пример реализации теплообмена на границе газ-твердое тело. Рассмотрены некоторые алгоритмы численного решения уравнений движения. Приведены примеры кода основных функций, необходимых для программной реализации метода МД на языке C++.

Приведенные сведения не претендуют на полноту и предназначены, в первую очередь, для первоначального знакомства с данным методом. В частности, практически не затронуты вопросы, связанные с выбором потенциалов межчастичного взаимодействия. Тем не менее, информации, приведенной в пособии, вполне достаточно для реализации простейших моделей, которые могут использоваться для изучения поведения различных физико-химических систем, а также особенностей обработки результатов прямого имитационного моделирования.

Библиографический список

1. *Frenkel D., Smit B.* Understanding molecular simulation. 2nd edition. San Diego; San Francisco; New York; Boston; London; Tokyo: Academic Press, 2002. 638 p.
2. Моделирование массопереноса в микропористых системах с учетом процессов физической адсорбции методом классической молекулярной динамики / А.Н. Агафонов, В.И. Платонов, А.В. Еремин [и др.] // Вестник Самарского государственного технического университета. Сер. Технические науки. 2015. № 3 (47). С. 7–11.
3. *Аксенова Е.В., Кшевецкий М.С.* Вычислительные методы исследования молекулярной динамики. СПб.: СПбГУ, 2009. 50 с.
4. *Левич В.Г., Вдовин Ю.Я., Мямлин А.А.* Курс теоретической физики. В 2 т. М.: Наука, 1971.
5. *Валуев А.А., Каклюгин А.С., Норманн Г.Э.* Молекулярное моделирование химического взаимодействия с поверхностью // Успехи химии. 1995. Т. 64. № 7. С. 643–671.
6. *Валуев А.А., Каклюгин А.С., Норманн Г.Э.* Области применимости адиабатического приближения в методе молекулярной динамики // Химическая физика 1996. Т. 15. № 2. С. 18–31.
7. *Marx D., Hutter J.* Ab Initio Molecular Dynamics: Theory and implementation // NIC Series 2000. Vol. 3. P. 329-477. URL: <http://www.fz-juelich.de/nicseries/volume3>.
8. *Бараш Ю.С.* Силы Ван-дер-Ваальса. М.: Наука. 1988. 344 с.
9. *Weeks J.D., Chandler D., Andersen H.C.* Role of Repulsive Forces in Determining the Equilibrium Structure in Simple Liquids // J. Chem. Phys. 1971. Vol. 54. № 12. P. 5237–5247.
10. *Кривцов А.М., Кривцова Н.В.* Метод частиц и его использование в механике деформируемого твердого тела // Дальневосточный математический журнал. 2002. Т. 3, № 2. С. 254.
11. *Еремин А.В., Горovenko Т.А.* Алгоритмы интегрирования уравнений движения в приложении метода молекулярной динамики // Теория и практика современной науки № 1(7) 2016. URL: http://modern-j.ru/osnovnoy_razdel__1_7__2016.
12. *Каплан И.Г.* Введение в теорию межмолекулярных взаимодействий. М.: Наука, 1982. 312 с.

13. Stringlike Cooperative Motion in a Supercooled Liquid / C. Donati, J.F. Douglas, W. Kob [et al.] // *Phys. Rev. Lett.* 1998. Vol. 30. № 11. P. 2338(3).

14. Growing range of correlated motion in polymer melt on cooling towards the glass transition / C. Benneman, C. Donati, J. Baschnagel [et al.] // *Nature* 1999. Vol. 399. P. 246–249.

15. Spatially correlated dynamics in a simulated glass-forming polymer melt: Analysis of clustering phenomena / Y. Gebremichael, T.B. Schroder, F.W. Starr [et al.] // *Phys. Rev. E* 2001. Vol. 64. № 5. P. 051503(13).

16. Particle rearrangements during transition between local minima of the potential energy landscape of a binary Lennard-Jones liquid / M. Vogel, B. Doliwa, A. Heuer [et al.] // *J. Chem. Phys.* 2004. Vol. 120. № 9. P. 4404–4414.

17. *Dunweg B., Kremer K.* Microscopic Verification of Dynamic Scaling in Dilute Polymer Solutions: A Molecular-Dynamics Simulation // *Phys. Rev. Lett.* 1991. Vol. 66. № 23. P. 2996–2999.

18. Finite size effects in simulation of glass dynamics / J. Horbach, W. Kob, K. Binder [et al.] // *Phys. Rev. E* 1996. Vol. 54. № 1. P. R5897–R5900.

19. *Baumketner A., Chushak Ya.* Correction for finite-size effects in molecular dynamics simulation of liquid alloy // *J. Non-Cryst. Solids.* 1999. Vol. 250-252. Part 1. № 8. P. 354–359.

20. *Doliwa B., Heuer A.* Finite-Size Effects in a Supercooled Liquid // *J. Phys.: Condens. Matter* 2003. Vol. 15. № 11. P. 849-857. URL: arXiv:cond-matt/0210121v1.

21. *Yeh I-C., Hummer G.* System-Size Dependence of Diffusion Coefficients and Viscosities from Molecular Dynamics Simulations with Periodic Boundary Conditions // *J. Phys. Chem. B* 2004. Vol. 108. № 40. P. 15873–15879.

22. Self-Diffusion Coefficient of the Hard-Sphere Fluid: System-Size Dependence and Empirical Correlations / D.M. Heyes, M.J. Cass, J.G. Powles [et al.] // *J. Phys. Chem. B* 2007. Vol. 111. № 6. P. 1455–1464.

23. *Shirts R.B., Burt S.R., Johnson A.M.* Periodic boundary condition induced breakdown of the equipartition principle and other kinetic effects of finite sample size in classical hard-sphere molecular dynamics simulations // *J. Chem. Phys.* 2006. Vol. 125. № 16. P. 164102(9).

24. *Karmakar S., Dasgupta C., Sastry S.* Growing length and time scales in glass-forming liquids // *PNAS* 2009. V. 106. № 10. P. 3675–3679.

25. *Ландау Л.Д., Лифшиц Е.М.* Статистическая физика. Ч. 1. М.: Наука, 1995.

26. *Хеерман Д.В.* Методы компьютерного эксперимента в теоретической физике. М.: Наука, 1990.

27. *Frenkel D., Smit B.* Understanding molecular simulation. 2nd edition // San Diego; San Francisco; New York; Boston; London; Tokio: Academic Press, 2002. 638 p.
28. *Полухин В.А., Ухов В.Ф., Дзугутов М.М.* Компьютерное моделирование динамики и структуры жидких металлов // М.: Наука. 1981. 324 с.
29. *Хеерман Д.В.* Методы компьютерного эксперимента в теоретической физике / под ред. С.А. Ахманова; пер. с англ. М.: Наука. 1990, 176 с.
30. *Alder B.J., Wainwright T.E.* Phase Transition for a Hard Sphere System // J. Chem. Phys. 1957. Vol. 27. № 6. P. 1208–1209.
31. *Де Жен П.* Идеи скейлинга в физике полимеров / под ред. И.М. Лифшица; пер. с англ. М.: Мир. 1982. 368 с.
32. *Douglas J.F.* How Does Surface Roughness Affect Polymer-Surface interactions? // Macromolecules 1989. Vol. 22. № 9. P. 3707–3716.
33. *Павлов А.С., Халатур П.Г.* Молекулярно-динамическое моделирование твердого углеводородного монослоя // Коллоидный журнал. 1996. Т. 58. № 1. С. 73–81.
34. *Raut J.S., Fichtorn K.A.* Tracer-diffusion coefficients for both localized and non-localized adsorption: Theory and molecular dynamics simulation // J. Chem. Phys. 1995. Vol. 103. № 19. P. 8694–8704.
35. *Liu P., Harder E., Berne B.J.* On the Calculation of Diffusion Coefficients in Confined Fluids and Interfaces with an Application to the Liquid-Vapor Interface of Water // J. Phys. Chem. B 2004. Vol. 108. № 21. P. 6959–6602.
36. *Odijk T.* On the Statistics and Dynamics of Confined or Entangled Stiff polymers // Macromolecules. 1983. Vol. 16. № 8. P. 1340–1344.
37. *Milchev A., Binder K.* Dynamics of Polymer Chains Confined in Slit-Like Pores // J. Phys. II France. 1996. Vol. 6. № 1. P. 21–31.
38. *Gallo P., Rovere M., Spohr E.* Glass transition and layering effects in confined water: A computer simulation study // J. Chem. Phys. 2000. Vol. 1113. № 24. P. 11324–11335.
39. Segment diffusion and nuclear magnetic resonance spin-lattice relaxation of polymer chains confined in tubes: Analytical treatment and Monte Carlo simulation of the crossover from Rouse to reptation dynamics / A. Denisov, M. Kroutieva, N. Fatkullin [et al.] // J. Chem. Phys. 2002. Vol. 116. № 12. P. 5217–5230.
40. *Liu P., Harder E., Berne B.J.* On the Calculation of Diffusion Coefficients in Confined Fluids and Interfaces with an Application to the Liquid-Vapor Interface of Water // J. Phys. Chem. B 2004. Vol. 108. № 21. P. 6959–6602.

41. Molecular dynamics simulation study of dynamics of fluids in thin films / R.G. Winkler, R.H. Schmid, A. Gerstmair [et al.] // J. Chem. Phys. 1996. Vol. 104. № 20. P. 8103–8111.
42. Molecular dynamics simulation of water between hydrophobic surfaces. Implication for the long-range hydrophobic force / M. Sakurai, H. Tamagawa, K. Ariga [et al.] // Chem. Phys. Lett. 1998. Vol. 289. P. 567–571.
43. *Yeh I.C., Berkowitz M.L.* Structure and dynamics of water at water / Pt interface as seen by molecular dynamics computer simulation // J. Electroanal. Chem. 1998. Vol. 450. P. 313–325.
44. *Пестряев Е.М.* Исследование ориентационных явлений стержнеобразных молекул в присутствии адсорбирующей поверхности методом молекулярной динамики // Коллоидный журнал. 2006. Т. 68. № 5. С. 653–663.
45. Interfacial Mobility of Polymers on Inorganic Solids / K. Tanaka, Y. Tateishi, Y. Okada [et al.] // J. Phys. Chem. B 2009. Vol. 113. № 2. P. 4571–4577.
46. *Kasturi L., Chopra.* McGraw-Hill Thin Film Phenomena // NewYork, 1969. xx, 844 pp.
47. *Брагин М.Д., Иванов А.В.* Локальноадаптивный выбор шага интегрирования в задачах молекулярной динамики // Препринты ИПМим. М.В. Келдыша. 2013. № 62. 39 с. URL: <http://library.keldysh.ru/pre-print.asp?id=2013-62>.
48. *Quinlan G.D., Tremaine S.* On the reliability of gravitational n-body integrations // Mon. Not. R. Astron. Soc., 1992. Vol. 259. P. 505–518.
49. *Hess W.* Tracer Diffusion in Polymeric Mixtures // Macromolecules. 1987. Vol. 20. № 10. P. 2587–2599.
50. *Kopf A., Dunweg B., Paul W.* Dynamics of polymer “isotope” mixtures: Molecular dynamics simulation and Rouse model analysis // J. Chem. Phys. 1997. Vol. 107. № 17. P. 6945–6955.
51. *Mukherjee A., Srinivas G., Bagchi B.* Reentrant Behavior of Relaxation Time with Viscosity at Varying Composition in Binary Mixtures // Phys. Rev. Lett. 2001. Vol. 86. № 26. P. 5926–5929. URL: [arXiv:physics/0012469v1](https://arxiv.org/abs/physics/0012469v1).

ПРИЛОЖЕНИЕ А. Таблица коэффициентов C_{mn}

n\m	2	3	4	5	6	7	8
3	6.75						
4	4	9.481481					
5	3.070026	5.379144	12.20703				
6	2.598076	4	6.75	14.92992			
7	2.310762	3.303852	4.920707	8.116861	17.65138		
8	2.116535	2.882048	4	5.836718	9.481481	20.372	
9	1.975953	2.598076	3.443646	4.691083	6.75	10.84472	23.09206
10	1.869186	2.393276	3.070026	4	5.379144	7.66155	12.20703
11	1.785152	2.238229	2.801186	3.536663	4.553155	6.065245	8.57193
12	1.717163	2.116535	2.598076	3.20378	4	5.104201	6.75
13	1.660937	2.01832	2.438959	2.952648	3.603024	4.461131	5.653778
14	1.613602	1.937277	2.310762	2.756174	3.303852	4	4.920707
15	1.573159	1.869186	2.205145	2.598076	3.070026	3.652753	4.395362
16	1.538172	1.811113	2.116535	2.467976	2.882048	3.381557	4

n\m	9	10	11	12	13	14	15
10	25.81175						
11	13.56871	28.53117					
12	9.481481	14.92992	31.25039				
13	7.433789	10.39042	16.29079	33.96946			
14	6.202289	8.116861	11.29891	17.65138	36.68841		
15	5.379144	6.75	8.799386	12.20703	19.01177	39.40727	
16	4.789531	5.836718	7.297096	9.481481	13.11487	20.372	42.12606

Учебное издание

*Агафонов Андрей Николаевич,
Еремин Анатолий Викторович*

**МЕТОД КЛАССИЧЕСКОЙ МОЛЕКУЛЯРНОЙ ДИНАМИКИ
В МОДЕЛИРОВАНИИ ФИЗИКО-ХИМИЧЕСКИХ ПРОЦЕССОВ**

Учебное пособие

Редактор *Т.В. Кондратьева*
Компьютерная верстка *Т.В. Кондратьева*

Подписано в печать 18.09.2017. Формат 60x84 1/16.

Бумага офсетная. Печ. л. 4,25.

Тираж 25 экз. Заказ . Арт. 26 /2017.

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(Самарский университет)
443086 САМАРА, МОСКОВСКОЕ ШОССЕ, 34.**

Изд-во Самарского университета.
443086 Самара, Московское шоссе, 34.