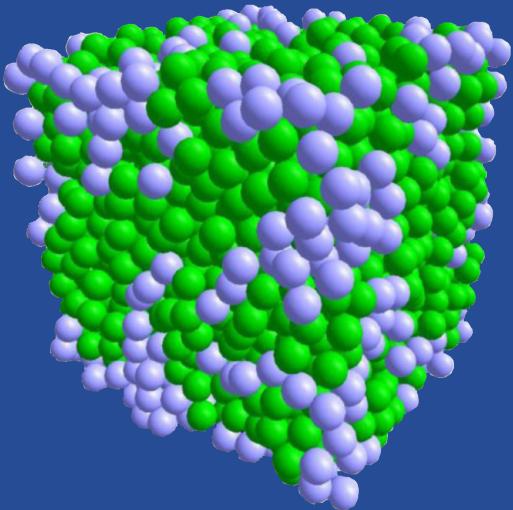


Б.Н. Галимзянов, А.В. Мокшин

Основы моделирования молекулярной динамики

Учебное пособие

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \frac{\vec{a}_i(t)\Delta t^2}{2} + \Omega(\Delta t^3)$$
$$\vec{r}_i(t - \Delta t) = \vec{r}_i(t) - \frac{\vec{a}_i(t)\Delta t^2}{2} - \Omega(\Delta t^3)$$


*Издается по решению учебно-методического совета
Института Физики
ФГАОУ ВО «Казанский (Приволжский) федеральный университет»*

Рецензенты:

д.ф.-м.н. **Ю.Д. Фомин** (ИФВД РАН, г. Москва, г. Троицк)

к.ф.-м.н. **Р.Е. Рыльцев** (УрО РАН, г. Екатеринбург)

к.ф.-м.н. **О.В. Недопекин** (КФУ, г. Казань)

Галимзянов Б.Н., Мокшин А.В. «Основы моделирования молекулярной динамики» Учебное пособие. – Казань: КФУ, 2016.
– 107 с.

В настоящем учебном пособии представлен базовый материал, необходимый для овладения знаниями и первичными навыками по компьютерному моделированию молекулярной динамики. Пособие включает в себя краткий лекционный материал, раскрывающий ключевые идеи и понятия, описание и примеры компьютерных программ, блок-схем основных процедур молекулярно-динамических расчетов, а также набор задач, связанных как с численными расчетами, так и визуализацией результатов моделирования. При работе с учебным пособием потребуется владение знаниями по классической механике, статистической физике и термодинамике, а также навыки в программировании. Работа с настоящим учебным пособием позволит овладеть навыками по составлению алгоритмов, связанных с моделированием молекулярной динамики; понять физические принципы, лежащие в основе современных вычислительных пакетов по компьютерному моделированию динамики многочастичных систем.

Учебное пособие предназначено для студентов бакалавриата и магистратуры физических специальностей высших учебных заведений, а также для аспирантов естественно-научных специальностей, интересующихся реализацией моделирования динамики многочастичных систем.

© Б.Н. Галимзянов,
А.В. Мокшин, 2016

Оглавление

Введение	5
Глава 1. Реализация метода молекулярной динамики	11
§ 1.1. Динамика атомов и молекул	11
§ 1.2. Уравнения движения	13
§ 1.3. Интегрирование уравнений движения	14
§ 1.4. Краевые (граничные) условия	17
§ 1.5. Статистические ансамбли	21
§ 1.6. Термостатирование и баростатирование	23
§ 1.7. Сила и ускорение частиц	31
§ 1.8. Оптимизация расчетов. Список Верле	33
Глава 2. Потенциалы взаимодействия	38
§ 2.1. Виды потенциалов	38
§ 2.2. Модели твердых и мягких сфер	41
§ 2.3. Парные потенциалы Леннард-Джонса и Джугутова	42
§ 2.4. Трехчастичный потенциал Стиллинжера-Вебера .	46
Глава 3. Обработка результатов моделирования	52
§ 3.1. Сохранение и загрузка конфигурации системы . . .	52
§ 3.2. Визуализация результатов	55
§ 3.3. Построение графиков в программном пакете Matlab	59
Глава 4. Моделирование молекулярной динамики твердых тел и жидкостей	65
§ 4.1. Этапы моделирования	65
§ 4.2. Начальные условия. Исходная конфигурации системы	68
§ 4.3. Функция радиального распределения частиц	76

§ 4.4. Моделирование фазовых переходов в модельной системе Леннард-Джонса	82
Задания для контроля знаний	90
Приложение	103
Литература	105

Введение

Методы компьютерного моделирования находят применение в решении самых разнообразных задач в таких научных областях как физика, химия, биология, материаловедение и т. д. Среди этих методов, включающих в себя моделирование сплошных сред, Монте-Карло моделирование, моделирование гидродинамики и др., особое место занимает метод моделирования классической динамики системы частиц (атомов, молекул), который обычно обозначают как *моделирование молекулярной динамики* (см. схему 1).



В основе метода лежит ключевая идея статистической физики о том, что любой материал представляет собой ансамбль большого числа взаимодействующих между собой атомов и/или молекул, которые, в свою очередь, в каждый определенный момент времени характеризуются координатами и импульсами (скоростями). При этом в настоящем пособии мы ограничиваемся рассмотрением таких систем, которые удовлетворяют классическому описанию: характеристическая длина волны де Броиля $\lambda_D = 2\pi\hbar/(mv)$ частицы с массой m и скоростью v , входящей в систему, является гораздо меньше среднего межчастичного расстояния $a \approx \sqrt[3]{V/N}$, где V – есть объем системы и N – есть количество частиц:

$$\lambda_D \ll a. \quad (1)$$

Учитывая то, что импульс частицы $p = mv$, определяющий значение λ_D , зависит от термодинамического состояния (равновесной) системы, простыми расчетами можно убедиться в том, что классическому описанию будут удовлетворять практически все жидкости (за исключением жидкого гелия и водорода), газы, многие кристаллы и аморфные материалы.

Более чем полувековая история становления и развития моделирования молекулярной динамики содержит много ярких моментов, перечисление и описание которых может составить содержание отдельной монографии. Приведем лишь некоторые из них, существенным образом повлиявшие на то, что из себя в настоящее время представляет моделирование молекулярной динамики, а также на то, какие научно-исследовательские задачи подвластны этому методу.

Так, первой работой по моделированию молекулярной динамики принято считать исследования 1957 года, выполненные Б. Олдером и Т. Вайнрайтом для перехода «жидкость-твердое тело» в системе, состоящей из твердых сфер [B.J. Alder and T.E. Wainwright, *Phase transition for a hard sphere system*, J. Chem. Phys. **27**, 1208 (1957)]. Примечательно, что, несмотря на несопоставимо низкую вычислительную производительность по сравнению с современными вычислительными возможностями, первый моделируемый образец представлял собой систему из 500 частиц, что уже в то время позволило выполнить статистическую обработку результатов.

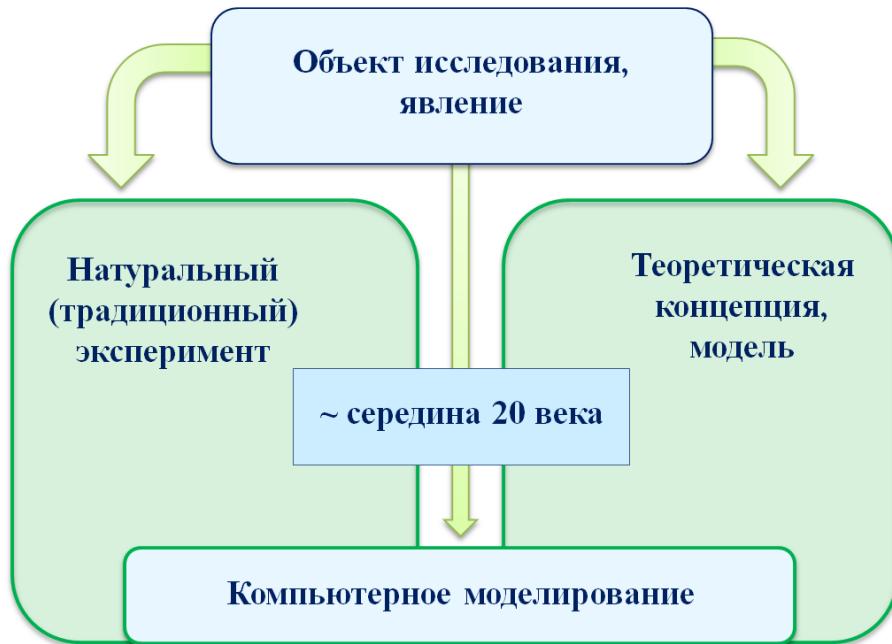
В 1964 А. Рахман публикует работу [A. Rahman, *Correlations in the motion of atoms in liquid argon*, Phys. Rev. A **136**, A405 (1964)],

в которой впервые были представлены результаты молекулярно-динамических расчетов, выполненных для системы из 864 частиц (атомов), чье взаимодействие реализовалось через потенциал Леннард-Джонса. Данная работа, в действительности, продемонстрировала, что результаты моделирования молекулярной динамики могут быть доведены до непосредственного сопоставления с экспериментальными данными. Методологический аппарат, основанный на расчете и последующем анализе корреляционных функций, который применил А. Рахман, активно используется исследователями вплоть до настоящего времени.

В работах А. Лиса и С. Эдвардса [A.W. Lees, S.F. Edwards, *The computer study of transport processes under extreme conditions*, J. Phys. C.: Solid State Phys. **5**, 1921 (1972)] было продемонстрировано применение метода моделирования молекулярной динамики к системам, находящихся в условиях устойчивого неравновесия. В работе было указано на возможность исследования обширного класса неравновесных процессов, что, в свою очередь, послужило стимулом к появлению значительного числа работ, посвященных проблемам реализации термостатирования и баростатирования в моделировании молекулярной динамики. В результате появилось новое научное направление, связанное с молекулярно-динамическим моделированием систем, испытывающих внешнее воздействие.

К концу 70-х годов XX века становится вполне понятным, что с увеличением числа научных задач, подвластных методу моделирования молекулярной динамики, появляется потребность в алгоритмах обработки и трактовки данных моделирования. Одной из ключевых в этом направлении является работа П. Стейнхардта, Д. Нельсона и М. Рончетти [P. Steinhardt, D. Nelson and M. Ronchetti, *Bond-orientational order in liquids and glasses*, Phys. Rev. B **28**, 784 (1983)], где впервые был представлен кластерный анализ, позволяющий исследовать локальные структурные особенности систем на основе данных моделирования.

Если в самых первых работах в середине XX века моделирование молекулярной динамики обозначалось как сугубо *теоретический метод*, то в настоящее время его выделяют в отдельное направление, выступающее связующим звеном между теорией и экспериментом, в направление, предполагающее проведение так называемых «численных экспериментов» (см. схему 2). Следует отметить, что в задачи традиционных молекулярно-динамических расчетов не вхо-



дит конструирование модельных потенциалов межчастичного взаимодействия. Этим занимается теоретическая физика. Здесь же силовые поля берутся в готовом виде и используются в молекулярно-динамических расчетах. Оперирование в моделировании данными на микроскопическом уровне позволяет получить исчерпывающую информацию о системе: знать положения, скорости и энергии всех атомов/молекул, формирующих систему. А именно на это нацелены многие экспериментальные методики.

К настоящему времени количество задач, решение которых может быть получено с помощью моделирования молекулярной динамики, столь велико, что перечислить их все не представляется возможным. Тем не менее, обозначим некоторые из них:

В физике:

- исследование релаксационных процессов в конденсированных средах;
- изучение одночастичной и коллективной атомарной/молекулярной динамики;
- исследование фазовых переходов и структурных трансформаций; исследование критических явлений;
- исследование жидкостей и стекол (микроскопическая дина-

мика, вязкоупругие свойства, теплопроводность и т. д.);

- свойства статистических ансамблей;
- размерные эффекты; структура и свойства кластеров;
- конструирование силовых полей и потенциалов межчастично-го взаимодействия;
- фазовые диаграммы физических систем;
- изучение равновесных и неравновесных процессов в многоча-стичных системах;
- и другие.

В материаловедении:

- создание (дизайн) новых устойчивых материалов и исследо-вание их свойств;
- материалы при экстремальных условиях (температурах, дав-лениях, нагрузках);
- микроскопические механизмы разрушения (например, фор-мирование пор и трещин); моделирование условий разрушения ма-териалов;
- точечные, линейные и плоские дефекты в кристаллах и их взаимодействие;
- исследование механических, реологических, упруго-пластиче-ских свойств твердых тел (кристаллов и аморфных материалов) на микроскопическом уровне;
- исследование трения;
- поверхностные и межфазные свойства (геометрия поверхно-сти, гидрофильность и гидрофобность и т. д.);
- свойства пленок, квазидвумерных и квазиодномерных струк-тур;
- и другие.

В химии и биологии:

- молекулярные структуры;
- протекание реакций;

- создание лекарственных препаратов;
- перенос энергии;
- структура мембран;
- динамика и взаимодействие крупных биомолекул;
- сворачиваемость белка;
- динамика ДНК;
- и другие.

Глава 1. Реализация метода молекулярной динамики

§1.1 Моделирование динамики атомов и молекул

В настоящее время для исследования структурных характеристик вещества широко применяются такие традиционные экспериментальные методы, как рентгеноструктурный анализ, нейтронная дифракция и т. д. В то же время активное развитие методов компьютерного моделирования, а также внедрение в практику вычислительных кластеров и суперкомпьютеров позволили исследовать разнообразные физические процессы, которые зачастую оказывались недоступными традиционным экспериментальным методам. Такое становится возможным из-за того, что в компьютерном моделировании траектории и скорости движения всех атомов или молекул вещества являются известными величинами.

Среди методов компьютерного моделирования широко применяются [1, 2]:

- метод *ab-initio* моделирования (моделирование из первых принципов), где силы межатомного/межмолекулярного взаимодействия определяются на основе решения квантово-механических уравнений [3];
- метод *молекулярной динамики* (МД), основанный на законах и положениях классической механики, статистической физики и термодинамики.

Из рисунка 1 видно, что метод *ab-initio* моделирования используется для описания квантово-механических процессов на пикосекундных временных масштабах в системах, размер которых составляет несколько нанометров. В настоящее время *ab-initio* моделирование позволяет исследовать динамику систем, состоящих из порядка 10^3 атомов.

Особый интерес представляет метод классического МД-моделирования, который дает возможность отслеживать эволюцию системы взаимодействующих частиц (т.е. атомов, молекул, ионов или других структурных единиц). Метод классического МД-моделирования позволяет рассматривать системы при некоторых заданных температуре T и давлении P с достаточно высокой расчетной скоростью. При этом моделирование систем, размер которых преимущественно составляет порядка нескольких десятков нанометров, возможно на временных масштабах, превышающих наносекунды.

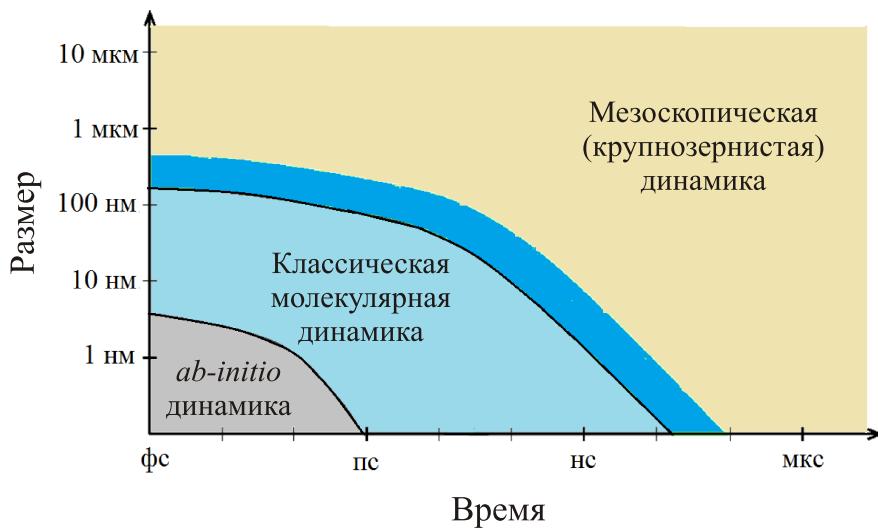


Рис. 1: Пространственно-временные области использования различных методов моделирования молекулярной динамики.

Преимущество метода классического МД-моделирования в том, что расчеты могут выполняться либо для одной заданной (P, T)-точки фазовой диаграммы рассматриваемой системы, либо для некоторой области фазовой диаграммы, например, при фазовом переходе из одного состояния в другое. При этом можно моделировать как равновесные, так и неравновесные процессы, происходящие в конденсированных системах [1, 2, 4]. Например, метод классического МД-моделирования позволяет рассматривать микроскопическую динамику перенасыщенного (переохлажденного) пара, жидкостей, аморфных материалов, перегретых и переохлажденных жидкостей, стекольных систем, кристаллов, а также квазикристаллов [5].

§1.2 Уравнения движения

Рассмотрим метод МД-моделирования на примере простой модельной системы. Под термином система подразумевается совокупность частиц (атомов, молекул, ионов и т. д.). В качестве примера рассмотрим трехмерную систему, состоящую из N идентичных частиц с одинаковой массой m . Частицы располагаются внутри моделируемой ячейки. В зависимости от условий решаемой задачи моделируемая ячейка может быть произвольной формы (например, кубической, сферической и т. д.). Преимущественно в МД-моделировании используется трехмерная прямоугольная ячейка с длинами ребер L_x , L_y , L_z и объемом $V = L_x \cdot L_y \cdot L_z$, где количественная плотность частиц системы определяется как $\rho = N/V$. На начальном этапе положения (координаты) частиц внутри этой ячейки задаются либо произвольным образом, например, с помощью генератора случайных чисел, либо задаются в виде некоторой кристаллической решетки (см. рисунки 2а и 2б). Начальные скорости и направления движения частиц также могут быть заданы случайным образом, при условии, что полная кинетическая энергия системы соответствует некоторой изначально заданной температуре [1, 2, 4, 5].

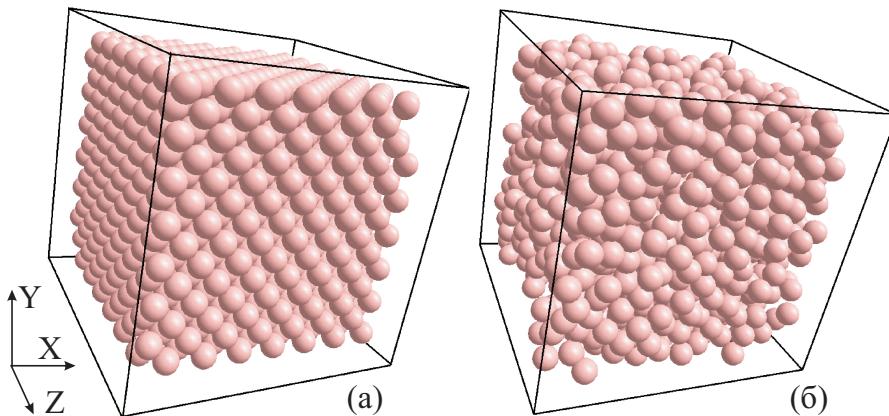


Рис. 2: (а) – вид кубической ячейки моделирования, где положения частиц заданы в узлах кристаллической решетки; (б) – координаты частиц заданы с помощью генератора случайных чисел.

В классическом равновесном случае моделирование молекулярной динамики сводится к решению уравнений движения Ньютона:

$$m \frac{d^2 \vec{r}_i(t)}{dt^2} = \vec{F}_i(t), \quad i = 1, 2, \dots, N, \quad (2)$$

где $\vec{r}_i = \vec{r}_i(x_i, y_i, z_i)$ определяет положение i -ой частицы, \vec{F}_i – есть сила, действующая на частицу i со стороны остальных частиц в мо-

мент времени t . Сила \vec{F}_i определяется через градиент от потенциала межчастичного взаимодействия $U(r_{ij})$:

$$\vec{F}_i = - \sum_{j \neq i}^N \frac{\partial U(r_{ij})}{\partial r_{ij}} \cdot \frac{\vec{r}_{ij}}{r_{ij}}, \quad j = 1, 2, \dots, N, \quad (3)$$

где $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$, $r_{ij} \equiv |\vec{r}_{ij}|$ – есть расстояние между i -ой и j -ой частицами. С математической точки зрения, при классическом МД-моделировании трехмерной системы необходимо решать $6N$ уравнений для того, чтобы определять положение $\vec{r}_i(x_i, y_i, z_i)$ и импульс $\vec{p}_i(p_{xi}, p_{yi}, p_{zi})$ каждой частицы в каждый момент времени t :

$$\begin{cases} v_{xi}(t) = v_{xi}(0) + a_{xi}(t)t \\ v_{yi}(t) = v_{yi}(0) + a_{yi}(t)t \\ v_{zi}(t) = v_{zi}(0) + a_{zi}(t)t \end{cases} \quad (4)$$

$$\begin{cases} x_i(t) = x_i(0) + v_{xi}(t)t + \frac{1}{2}a_{xi}(t)t^2 \\ y_i(t) = y_i(0) + v_{yi}(t)t + \frac{1}{2}a_{yi}(t)t^2 \\ z_i(t) = z_i(0) + v_{zi}(t)t + \frac{1}{2}a_{zi}(t)t^2 \end{cases} \quad (5)$$

Здесь $v_{xi} = p_{xi}/m$, $v_{yi} = p_{yi}/m$ и $v_{zi} = p_{zi}/m$ – компоненты вектора скорости i -ой частицы. При численном решении уравнений (22) и (5) рассматриваемый временной интервал делится на малые отрезки, на которых движение можно считать равноускоренным.

§1.3 Интегрирование уравнений движения

Разобьем временной масштаб моделирования t на малые отрезки размера Δt (см. рисунок 3). Здесь важным является правильный выбор размера временного шага Δt . С одной стороны, величина Δt должна быть очень малой. С другой стороны, слишком малое значение не позволит охватить достаточный временной интервал, на котором система эволюционирует. Преимущественно величина Δt выбирается таким образом, чтобы за один временной шаг не происходило взаимного перекрытия частиц. При моделировании физических процессов с достаточно высокими температурами, где частицы двигаются с большими скоростями, возникает необходимость использования малых временных шагов (порядка $\Delta t \leq 10^{-14}$ с). Для низкотемпературной области можно использовать относительно большие

временные шаги (порядка $\Delta t > 10^{-14}$ с). Временной шаг Δt – это ключевая величина, от которой зависит размер доступного времени наблюдения за системой. Очевидно, что при малых значениях величины Δt потребуется значительный объем вычислительных ресурсов и большое количество временных шагов M .



Рис. 3: Схематичный рисунок, показывающий разбиение временного масштаба на малые шаги Δt .

Для расчета координат x_i, y_i, z_i и скоростей v_{xi}, v_{yi}, v_{zi} i -ой частицы в различные моменты времени используются алгоритмы интегрирования уравнений движения. Наиболее известным является классический алгоритм численного интегрирования уравнений движения, введенный в 60-е годы XX века Л. Верле [4, 6, 7]. Согласно этому алгоритму для расчета положений частиц в момент времени $t + \Delta t$ используются координаты и скорости частиц в момент времени t , а также на предыдущем временном шаге $t - \Delta t$. В алгоритме Верле положение частицы определяется в виде разложения

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{\vec{a}_i(t)\Delta t^2}{2} + \Omega(\Delta t^3), \quad (6)$$

$$\vec{r}_i(t - \Delta t) = \vec{r}_i(t) - \vec{v}_i(t)\Delta t + \frac{\vec{a}_i(t)\Delta t^2}{2} - \Omega(\Delta t^3). \quad (7)$$

Из уравнений (6) и (7) можно получить обобщенное уравнение вида

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \vec{a}_i(t)\Delta t^2 + \Omega(\Delta t^4). \quad (8)$$

Здесь $\vec{a}_i = \vec{a}_i(a_x, a_y, a_z)$ – есть ускорение частицы ($i = 1, 2, \dots, N$), определяемое выражением

$$\vec{a}_i = \frac{\vec{F}_i}{m} = -\frac{1}{m} \sum_{j \neq i}^N \frac{\partial U(r_{ij})}{\partial r_{ij}} \frac{\vec{r}_{ij}}{r_{ij}}. \quad (9)$$

Более эффективным методом численного интегрирования уравнений движения является алгоритм Верле в скоростной форме [2]. В этом алгоритме координаты и скорости движения частиц рассчитываются с помощью уравнений вида

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \vec{a}_i(t)\frac{\Delta t^2}{2}, \quad (10)$$

$$\vec{v}_i(t + \Delta t/2) = \vec{v}_i(t) + \vec{a}_i(t) \frac{\Delta t}{2}, \quad (11)$$

$$\vec{a}_i(t + \Delta t) = \frac{\vec{F}_i(t)}{m}, \quad (12)$$

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t + \Delta t/2) + \vec{a}_i(t + \Delta t) \frac{\Delta t}{2}, \quad (13)$$

где m – есть масса частицы (здесь предполагается, что масса всех частиц одинакова). Координаты частиц определяются на временном шаге $t + \Delta t$ (10), а скорости движения частиц рассчитываются на половинном шаге $t + \Delta t/2$ (11). Здесь изначально ускорение частиц $\vec{a}_i(t)$ считается уже известным. Далее определяются силы взаимодействия частиц и их ускорения $\vec{a}_i(t + \Delta t)$ (12), а также скорости движения частиц на целом временном шаге $t + \Delta t$ (13).

Реализация процедуры интегрирования классических уравнений движения с помощью скоростного алгоритма Верле на языке программирования C# приведена в коде 1.1.

Код 1.1

```

for (int i = 0; i < N; i++)
{
    pos_x[i] = pos_x[i] + vel_x[i] * dt + acc_x[i] * dt * dt / 2.0f;
    pos_y[i] = pos_y[i] + vel_y[i] * dt + acc_y[i] * dt * dt / 2.0f;
    pos_z[i] = pos_z[i] + vel_z[i] * dt + acc_z[i] * dt * dt / 2.0f;
}
for (int i = 0; i < N; i++)
{
    vel_x[i] = vel_x[i] + acc_x[i] * dt / 2.0f;
    vel_y[i] = vel_y[i] + acc_y[i] * dt / 2.0f;
    vel_z[i] = vel_z[i] + acc_z[i] * dt / 2.0f;
}
load_forces()
for (int i = 0; i < N; i++)
{
    vel_x[i] = vel_x[i] + acc_x[i] * dt / 2.0f;
    vel_y[i] = vel_y[i] + acc_y[i] * dt / 2.0f;
    vel_z[i] = vel_z[i] + acc_z[i] * dt / 2.0f;
}

```

В программном коде 1.1 `pos_x`, `pos_y` и `pos_z` – есть массив координат частиц с проекциями на оси X, Y и Z соответственно; `vel_x`, `vel_y`, `vel_z` и `acc_x`, `acc_y`, `acc_z` – есть массивы, где хранятся текущие скорости и ускорения частиц соответственно. Эти массивы могут задаваться в виде переменных `float[]` или `double[]` с размером N . Через `dt` обозначается временной шаг интегрирования уравнений движения. Процедура `load_forces` используется для расчета ускорений частиц на временном шаге $t + \Delta t$ (реализация данной процедуры будет представлена во второй главе).

§1.4 Краевые (граничные) условия

Предположим, что имеется некоторая микроскопическая система. Пусть *однородность* в свойствах этой системы с уменьшением ее размера сохраняется вплоть до величины l . Так, в случае равновесного газа $l \geq$ длины свободного пробега молекул газа; в случае равновесной простой однокомпонентной жидкости $l \geq$ обратной величины, характеризующей удвоенный размер псевдозоны Бриллюэна и т. д. Это означает, что по свойствам произвольного *фрагмента* этой системы с размером, превышающим l , можно получать информацию о свойствах всей макросистемы. Именно эти рассуждения дают основания заменять макроскопическую систему, состоящую из бесконечно большого числа атомов/молекул, ее фрагментами, содержащей конечное (хотя и большое) число атомов/молекул. Моделируемый фрагмент системы обычно называется *ячейкой моделирования*.

Для того, чтобы ячейка моделирования представляла собой часть системы, а не изолированный образец некоторого конечного размера, необходимо применять соответствующие *краевые (граничные) условия*.

Обычно используются две разновидности граничных условий. В первом случае ячейка моделирования может являться «изолированной». Здесь частицы, достигшие границы, испытывают упругое (зеркальное) соударение с воображаемой поверхностью ячейки – стенкой (см. рисунок 4а). В другом случае, при моделировании условно «бесконечной» системы применяются так называемые периодические граничные условия [2, 4]. Для реализации таких условий используются следующие правила:

- ячейка моделирования транслируется по всем направлениям. В

результате этого движение «образов» частиц, расположенных в соседних ячейках, происходит аналогичным образом, как и в основной ячейке (см. рисунок 4б);

- частица, покидающая основную ячейку с одной стороны, попадает в нее в виде ее «образа» с противоположной грани ячейки. Таким образом, количество частиц в системе сохраняется.

При использовании периодических граничных условий применительно к прямоугольной системе с длинами ребер L_x, L_y, L_z , координаты частиц преобразуются следующим образом:

$$\begin{aligned} x_i(t) \geq L_x &\Rightarrow x_i(t + \Delta t) = x_i(t) - L_x, \\ x_i(t) < 0 &\Rightarrow x_i(t + \Delta t) = x_i(t) + L_x, \\ y_i(t) \geq L_y &\Rightarrow y_i(t + \Delta t) = y_i(t) - L_y, \\ y_i(t) < 0 &\Rightarrow y_i(t + \Delta t) = y_i(t) + L_y, \\ z_i(t) \geq L_z &\Rightarrow z_i(t + \Delta t) = z_i(t) - L_z, \\ z_i(t) < 0 &\Rightarrow z_i(t + \Delta t) = z_i(t) + L_z. \end{aligned} \tag{14}$$

В случае, когда начало координат совпадает с геометрическим центром системы периодические граничные условия можно представить в следующем виде:

$$\begin{aligned} x_i(t) \geq L_x/2 &\Rightarrow x_i(t + \Delta t) = x_i(t) - L_x, \\ x_i(t) < -L_x/2 &\Rightarrow x_i(t + \Delta t) = x_i(t) + L_x, \\ y_i(t) \geq L_y/2 &\Rightarrow y_i(t + \Delta t) = y_i(t) - L_y, \\ y_i(t) < -L_y/2 &\Rightarrow y_i(t + \Delta t) = y_i(t) + L_y, \\ z_i(t) \geq L_z/2 &\Rightarrow z_i(t + \Delta t) = z_i(t) - L_z, \\ z_i(t) < -L_z/2 &\Rightarrow z_i(t + \Delta t) = z_i(t) + L_z. \end{aligned} \tag{15}$$

Аналогичным образом возникает необходимость в преобразовании расстояний между любыми двумя частицами $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ и их проекциями r_{xij}, r_{yij} и r_{zij} на координатные оси X, Y и Z соответственно (где $i = 1, 2, \dots, N$ и $j = 1, 2, \dots, N$). Для этого можно

воспользоваться условиями следующего вида

$$\begin{aligned} |r_{xij}| > L_x/2 \Rightarrow r'_{xij} &= r_{xij} - (\text{int})[r_{xij}/L_x]L_x, \\ |r_{yij}| > L_y/2 \Rightarrow r'_{yij} &= r_{yij} - (\text{int})[r_{yij}/L_y]L_y, \\ |r_{zij}| > L_z/2 \Rightarrow r'_{zij} &= r_{zij} - (\text{int})[r_{zij}/L_z]L_z, \end{aligned} \quad (16)$$

где операция $(\text{int})[\dots]$ возвращает целую часть от деления. С учетом краевых условий координаты частиц меняются от $-L_x/2, -L_y/2, -L_z/2$ до $L_x/2, L_y/2, L_z/2$, полагая, что начало координат совпадает с геометрическим центром системы. Периодические граничные условия применяются на каждом временнóм шаге МД-моделирования.

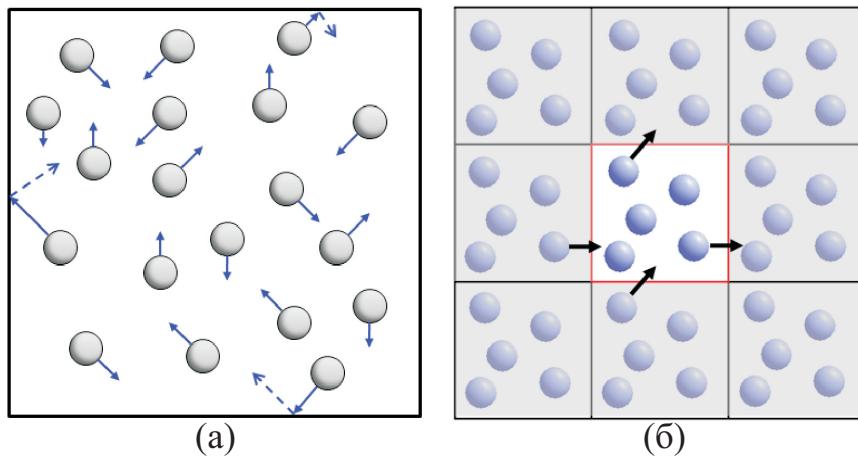


Рис. 4: (а) – ячейка моделирования в случае зеркальных граничных условий; (б) – ячейка моделирования с периодическими граничными условиями.

Процедура реализации периодических граничных условий приведена в программном коде 1.2:

Код 1.2

```
public void boundary_conditions()
{
    for (int i = 0; i < N; i++)
    {
        if (pos_x[i] >= L_x / 2.0) pos_x[i] = pos_x[i] - L_x;
        if (pos_x[i] < -L_x / 2.0) pos_x[i] = pos_x[i] + L_x;
        if (pos_y[i] >= L_y / 2.0) pos_y[i] = pos_y[i] - L_y;
        if (pos_y[i] < -L_y / 2.0) pos_y[i] = pos_y[i] + L_y;
        if (pos_z[i] >= L_z / 2.0) pos_z[i] = pos_z[i] - L_z;
        if (pos_z[i] < -L_z / 2.0) pos_z[i] = pos_z[i] + L_z;
    }
}
```

```

    }
}
```

Здесь через L_x , L_y и L_z обозначены длины ребер прямоугольной системы, где геометрический центр системы совпадает с началом системы координат. В случае преобразования расстояний между частицами используется процедура `distance_refold`, представленная в программном коде 1.3.

Код 1.3

```

public void distance_refold()
{
    for (int i = 0; i < N; i++)
    {
        rij_x = rij_x - Convert.ToInt32(rij_x / L_x) * L_x;
        rij_y = rij_y - Convert.ToInt32(rij_y / L_y) * L_y;
        rij_z = rij_z - Convert.ToInt32(rij_z / L_z) * L_z;
    }
}
```

Здесь rij_x , rij_y и rij_z – есть проекции вектора \vec{r}_{ij} на координатные оси. Также достаточно удобным представляется моделирование системы, у которой геометрический центр совпадает с началом системы координат. Например, при моделировании сферически-симметричных систем, при моделировании фазовых переходов, инициируемых внешними воздействиями, сдвиговой деформацией, при исследовании процессов зародышебразования, где зародыши новой формирующейся фазы (кристаллической, жидкой) имеют форму, близкую к сферической. Для того, чтобы геометрический центр системы совпадал с началом системы координат можно воспользоваться программным кодом 1.4.

Код 1.4

```

public void load_system_center()
{
    system_center = new float[3];
    system_center[0] = 0;
    system_center[1] = 0;
    system_center[2] = 0;
```

```

system_center[0] = 0;
system_center[1] = 0;
system_center[2] = 0;
for (int i = 0; i < N; i++)
{
    system_center[0] += pos_x[i];
    system_center[1] += pos_y[i];
    system_center[2] += pos_z[i];
}
system_center[0] = system_center[0] / N;
system_center[1] = system_center[1] / N;
system_center[2] = system_center[2] / N;
for (int j = 0; j < N; j++)
{
    pos_x[j] = pos_x[j] - system_center[0];
    pos_y[j] = pos_y[j] - system_center[1];
    pos_z[j] = pos_z[j] - system_center[2];
}
}

```

§1.5 Статистические ансамбли

В процессе моделирования на многочастичную систему могут накладываться дополнительные условия, которые позволяют учитывать влияние окружающей среды. В большинстве случаев МД-расчеты реализуются через следующие статистические ансамбли [1, 2, 4]:

1. Микроканонический «NVE-ансамбль». Здесь количество частиц N , объем V и полная энергия E системы остаются постоянными. Данный ансамбль соответствует адиабатическому процессу без теплообмена с окружающей средой. При этом обмен происходит между потенциальной и кинетической энергиами системы.
2. Канонический «NVT-ансамбль». Здесь помимо числа частиц и объема, постоянной величиной является температура T системы. В данном случае происходят эндо- и экзотермические процессы обмена энергии с терmostатом.
3. Изобарический-изотермический «NPT-ансамбль». Здесь температура T и давление P системы являются постоянными. Дан-

ный ансамбль наиболее приближен к реальным условиям, реализуемым в экспериментах, где кроме температуры учитывается внешнее давление. Здесь давление контролируется с помощью баростата.

В случае NVE-ансамбля мгновенное значение полной энергии системы определяется выражением

$$E(t) = \sum_{i=1}^N \frac{\vec{p}_i^2(t)}{2m} + \frac{1}{2} \sum_{i=1}^N \phi_i(t), \quad (17)$$

где ϕ – есть удельная потенциальная энергия (потенциальная энергия одной частицы). Согласно третьему закону Ньютона $F_{ij} = -F_{ji}$, следовательно, при расчете потенциальной энергии в выражении (17) каждую атомную пару нужно рассмотреть только один раз. В случае NVT- и NPT-ансамблей на каждом временном шаге МД-моделирования (или через определенный промежуток времени) необходимо определять мгновенные значения температуры $T(t)$ и давления $P(t)$ системы. Мгновенное значение температуры системы $T(t)$ характеризует среднюю кинетическую энергию теплового движения частиц, приходящуюся на одну степень свободы равновесной системы, и определяется выражением [4]:

$$T(t) = \frac{2}{3Nk_B} \sum_{i=1}^N \frac{\vec{p}_i^2(t)}{2m}. \quad (18)$$

Для определения мгновенного значения давления системы $P(t)$ преимущественно используется вириальное выражение следующего вида [4]:

$$P(t) = \frac{1}{3V} \sum_{i=1}^N \left(\frac{\vec{p}_i^2(t)}{m} + \sum_{j \neq i}^N \vec{F}_i(t) \cdot \vec{r}_{ij} \right). \quad (19)$$

Ниже приведен программный код 1.5, который можно использовать для расчета мгновенной температуры и мгновенного давления системы.

Код 1.5

```
kinetic_energy = 0;
for (int i = 0; i < N; i++)
```

```

{
    rv_x = vel_x[i];
    rv_y = vel_y[i];
    rv_z = vel_z[i];
    kinetic_energy += rv_x * rv_x + rv_y * rv_y + rv_z * rv_z;
}
kinetic_energy = kinetic_energy / (2.0f * N);
temperature = 2.0f * kinetic_energy / 3.0f;
volume = L_x * L_y * L_z;
density = N / volume;
pressure = density * temperature + virial / volume;

```

Следует обратить внимание на то, что на каждом временному шаге вместе с температурой `temperature` и давлением `pressure` рассчитываются объем `volume` и количественная плотность системы `density`. Переменная `virial` – есть вклад $\sum_{j \neq i}^N \vec{F}_i \cdot \vec{r}_{ij}$ в выражении (19).

§1.6 Термостатирование и баростатирование

Для удержания температуры и давления системы на определенном заданном значении или в диапазоне значений используются термостаты и баростаты. Реализация термостатов и баростатов как правило, осуществляется через корректировку скоростей и траекторий частиц системы, а также размера ячейки моделирования.

Рассмотрим некоторые виды термостатов и баростатов применяемых при моделировании МД.

Метод масштабирования скоростей, предложенный в 1971 году Л.В. Вудкоком [2, 8], является одним из наиболее простых алгоритмов термостатирования. В случае, когда величина $T(t)$ определяет температуру многочастичной системы в момент времени t , разность между заданным и мгновенным значениями температуры $\Delta T(t) = T_{ext} - T(t)$ будет определяться следующим образом

$$\Delta T(t) = \frac{m}{3Nk_B} \left[\sum_{i=1}^N (\lambda_{sc} \vec{v}_i(t))^2 - \sum_{i=1}^N \vec{v}_i^2(t) \right] \quad (20)$$

или

$$\Delta T(t) = [\lambda_{sc}^2 - 1]T(t). \quad (21)$$

Здесь λ_{sc} – есть масштабирующий параметр, позволяющий корректировать скорости движения частиц на каждом временнóм шаге Δt :

$$\begin{cases} v_{xi}(t + \Delta t) = \lambda_{sc} \cdot v_{xi}(t) + a_{xi}(t)\Delta t, \\ v_{yi}(t + \Delta t) = \lambda_{sc} \cdot v_{yi}(t) + a_{yi}(t)\Delta t, \\ v_{zi}(t + \Delta t) = \lambda_{sc} \cdot v_{zi}(t) + a_{zi}(t)\Delta t, \end{cases} \quad (22)$$

где

$$\lambda_{sc} = \sqrt{\frac{T_{ext}}{T(t)}}. \quad (23)$$

В соотношении (23) мгновенное значение температуры не должно быть равным нулю. Поэтому перед расчетом коэффициента λ_{sc} скорости движения частиц должны быть заданы.

Термостат Берендсена, предложенный в 1984 году, воспроизводит теплообмен с внешним источником тепловой энергии [4, 9, 10]. Здесь скорости движения частиц масштабируются на каждом временнóм шаге через параметр

$$\lambda_B = \sqrt{1 + \frac{\Delta t}{Q_T^{(B)}} \left(\frac{T_{ext}}{T(t)} - 1 \right)}, \quad (24)$$

где $Q_T^{(B)}$ определяет интенсивность взаимодействия системы с термостатом. С ростом значения параметра $Q_T^{(B)}$ флуктуации температуры вблизи заданного значения T_{ext} уменьшаются. В противном случае взаимодействие системы с термостатом уменьшается, что приводит к существенным колебаниям температуры. Значение величины $Q_T^{(B)}$ преимущественно определяется с учетом размера системы (от количества частиц) и температуры термостата. Как правило, оно составляет $Q_T^{(B)} \in [1; 100]\Delta t$.

Термостат Нозе-Гувера реализуется через введение искусственных координат и импульсов частиц, которые с исследуемой системой через параметр Q_T [11, 12]. Этот термостат использует дополнительные степени свободы, которые интегрированы в уравнения движения. В результате происходит обмен энергией между системой и тепловым резервуаром (термостатом). Отклонение температуры от заданного значения T_{ext} корректируется через параметр

$$\dot{\lambda} = \frac{1}{Q_T} \left(\sum_{i=1}^N \frac{\vec{p}_i^2}{2m} - 3Nk_B T_{ext} \right), \quad (25)$$

где Q_T определяет интенсивность взаимодействия термостата Нозе-Гувера с системой. При использовании термостата Нозе-Гувера уравнения движения имеют следующий вид:

$$m \frac{d\vec{p}}{dt} = \vec{F} - \lambda \vec{p}, \quad m \frac{d\vec{r}}{dt} = \vec{p}. \quad (26)$$

В отличие от термостата Берендсена в термостате Нозе-Гувера энергия поступает в систему некоторыми порциями, что приводит к периодическим колебаниям температуры. При больших значениях параметра Q_T осцилляции температуры уменьшаются. Однако малые значения Q_T могут привести к выраженным температурным флуктуациям. Как правило, значение параметра Q_T выбирается из области от $0.1 \epsilon \tau^2$ до $10^3 \epsilon \tau^2$ (в леннард-джонсовских единицах). Оптимальное значение параметра Q_T для трехмерной системы можно оценить с помощью соотношения

$$Q_T \simeq 3Nk_B T_{ext} \tau_r^2. \quad (27)$$

Здесь τ_r – есть время структурной релаксации системы (например, для LJ системы $\tau_r^2 \approx 88 \cdot 10^{-4} \tau^2 \approx 127 \cdot 10^{-28} c^2$).

При использовании термостата Нозе-Гувера гамильтониан системы имеет вид [2, 4]

$$H = E_{kin} + E_{pot} + \frac{\xi^2 Q_T}{2} + 3Nk_B T S_f. \quad (28)$$

Величина S_f характеризует взаимосвязь системы с термостатом и определяется выражением

$$\dot{S}_f = \xi + \left(\sum_{i=1}^N \frac{\vec{p}_i^2}{2m} - 3Nk_B T \right) \frac{\Delta t}{Q_T}. \quad (29)$$

В качестве примера ниже приведен код 1.6 процедуры `system_dynamics`, используемой для интегрирования уравнений движения через скоростной алгоритм Верле, где термостатирование осуществляется методом масштабирования скоростей.

Код 1.6

```
public void system_dynamics(int stage_id)
{
    if (stage_id == 1)
```

```

{
    for (int i = 0; i < N; i++)
    {
        pos_x[i] = pos_x[i] + vel_x[i] * dt + acc_x[i] * dt * dt / 2.0f;
        pos_y[i] = pos_y[i] + vel_y[i] * dt + acc_y[i] * dt * dt / 2.0f;
        pos_z[i] = pos_z[i] + vel_z[i] * dt + acc_z[i] * dt * dt / 2.0f;
    }
    kinetic_energy = 0;
    for (int i = 0; i < N; i++)
    {
        rv_x = vel_x[i];
        rv_y = vel_y[i];
        rv_z = vel_z[i];
        kinetic_energy += rv_x * rv_x + rv_y * rv_y + rv_z * rv_z;
    }
    kinetic_energy = kinetic_energy / 2.0f;
    temperature = 2.0f * kinetic_energy / (3.0f * N);
    T0 = T0 + (float)(Math.Sign(T_ext - T0) * vel_dT * dt);
    if (temperature == 0) temperature = T0;
    lambda = (float)Math.Sqrt(T0 / temperature);
    for (int i = 0; i < N; i++)
    {
        vel_x[i] = lambda * vel_x[i] + acc_x[i] * dt / 2.0f;
        vel_y[i] = lambda * vel_y[i] + acc_y[i] * dt / 2.0f;
        vel_z[i] = lambda * vel_z[i] + acc_z[i] * dt / 2.0f;
    }
}
if (stage_id == 2)
{
    volume = L_x * L_y * L_z;
    density = N / volume;
    pressure = density * temperature + virial / volume;
    for (int i = 0; i < N; i++)
    {
        vel_x[i] = vel_x[i] + acc_x[i] * dt / 2.0f;
        vel_y[i] = vel_y[i] + acc_y[i] * dt / 2.0f;
        vel_z[i] = vel_z[i] + acc_z[i] * dt / 2.0f;
    }
    H_energy = kinetic_energy + potential_energy;
}
}

```

Как видно из кода 1.6, масштабирование скорости движения частиц при интегрировании уравнений движения с помощью скоростного алгоритма Верле можно осуществить на временнóм шаге $t + \Delta t/2$.

Для нагрева или охлаждения системы используются следующие условия:

$$\begin{aligned} T_0(t) > T_{ext} &\Rightarrow T'_0(t + \Delta t) = T_0(t) - v_T \Delta t, \\ T_0(t) < T_{ext} &\Rightarrow T'_0(t + \Delta t) = T_0(t) + v_T \Delta t, \end{aligned} \quad (30)$$

где T_0 – есть температура системы в момент времени t , T_{ext} – целевая температура, v_T – скорость нагрева или охлаждения системы.

Моделирование многочастичной системы при внешнем заданном давлении осуществляется в NPT-ансамбле. Здесь давление системы может удерживаться вблизи некоторого заданного значения P_{ext} с помощью баростатов. В качестве примера ниже приведены процедуры реализации наиболее известных баростатов, таких как, баростаты Берендсена и Нозе-Гувера.

Баростат Берендсена использует принцип локального возмущения, где скорость изменения давления системы определяется через соотношение [9, 10]

$$\frac{dP(t)}{dt} = \frac{P_{ext} - P(t)}{Q_P^{(B)}}. \quad (31)$$

Здесь параметр $Q_P^{(B)}$ определяет интенсивность взаимодействия баростата с системой, величина P_{ext} – есть внешнее давление. Корректировка размеров (объема) системы и координат частиц осуществляется через масштабирующий параметр ξ_B , который связан с изотермической сжимаемостью β_T соотношением

$$\xi_B = \frac{\beta_T}{3Q_P^{(B)}} [P(t) - P_{ext}]. \quad (32)$$

В случае равновесной молекулярной динамики при использовании баростата Берендсена уравнения движения принимают вид:

$$m \frac{d\vec{r}}{dt} = \vec{p} - m\xi_B \vec{r}, \quad (33)$$

$$m \frac{d\vec{p}}{dt} = \vec{F}. \quad (34)$$

Объем системы корректируется следующим образом

$$\Delta V = V(t) - V = 3\xi_B V \Delta t, \quad (35)$$

где $V(t)$ – есть объем системы в момент времени t , V – предыдущий объем системы.

При использовании **баростата Нозе-Гувера** в уравнения движения вводится дополнительный множитель ξ , который определяется соотношением [11, 12]

$$\frac{d\xi}{dt} = \frac{V(t)}{Q_P} [P(t) - P_{ext}]. \quad (36)$$

Здесь

$$Q_P \simeq N k_B T_{ext} \tau_p^2, \quad (37)$$

где τ_p – характеризует период колебаний давления (например, для LJ системы $\tau_p^2 \approx 120 \tau^2$). Здесь уравнения движения принимают вид:

$$m \frac{d\vec{r}}{dt} = \vec{p} + m\xi \vec{r}, \quad (38)$$

$$m \frac{d\vec{p}}{dt} = \vec{F} - \xi \vec{p}. \quad (39)$$

Изменение объема системы при использовании баростата Нозе-Гувера определяется соотношением (35). Линейное расширение или сжатие прямоугольной системы корректируется следующим образом:

$$L_x(t + \Delta t) = L_x(t) + L_x(t)\xi \Delta t, \quad (40)$$

$$L_y(t + \Delta t) = L_y(t) + L_y(t)\xi \Delta t, \quad (41)$$

$$L_z(t + \Delta t) = L_z(t) + L_z(t)\xi \Delta t. \quad (42)$$

Термостат и баростат выбираются в зависимости от условий, наложиваемых на систему. Например, для того, чтобы привести систему в состояние термодинамического равновесия при постоянной температуре и/или давлении, как правило, используются метод масштабирования скоростей, термостат и баростат Берендсена. При моделировании фазовых переходов в конденсированных системах преимущественно используются термостат и баростат Нозе-Гувера.

Процедура реализации термостата и баростата Нозе-Гувера при МД-моделировании приведена в программном коде 1.7.

Код 1.7

```

public void system_dynamics(int stage_id)
{
    if (stage_id == 1)
    {
        for (int i = 0; i < N; i++)
        {
            pos_x[i] = pos_x[i] + vel_x[i] * dt + (acc_x[i] - (lambda + xi) *
vel_x[i]) * dt * dt / 2.0f;
            pos_y[i] = pos_y[i] + vel_y[i] * dt + (acc_y[i] - (lambda + xi) *
vel_y[i]) * dt * dt / 2.0f;
            pos_z[i] = pos_z[i] + vel_z[i] * dt + (acc_z[i] - (lambda + xi) *
vel_z[i]) * dt * dt / 2.0f;
        }
        kinetic_energy = 0;
        for (int i = 0; i < N; i++)
        {
            rv_x = vel_x[i];
            rv_y = vel_y[i];
            rv_z = vel_z[i];
            kinetic_energy += rv_x * rv_x + rv_y * rv_y + rv_z * rv_z;
        }
        temperature = kinetic_energy / (3.0f * N)
        T0 = T0 + (float)(Math.Sign(T_ext - T0) * vel_dT * dt);
        if (temperature == 0) temperature = T0;
        S_f += lambda*dt+(kinetic_energy-3.0f*N*T0)*dt*dt/Q_T;
        lambda += (kinetic_energy - 3.0f * N * T0) * dt / (2.0f * Q_T);
        for (int i = 0; i < N; i++)
        {
            vel_x[i] += (acc_x[i] - (lambda + xi) * vel_x[i]) * dt / 2.0f;
            vel_y[i] += (acc_y[i] - (lambda + xi) * vel_y[i]) * dt / 2.0f;
            vel_z[i] += (acc_z[i] - (lambda + xi) * vel_z[i]) * dt / 2.0f;
        }
    }
    if (stage_id == 2)
    {
        volume = L_x * L_y * L_z;
        density = N / volume;
        pressure = density * temperature + virial / volume;
        xi = xi + (pressure - P_ext) * dt / (Q_B * density);
        L_x = L_x * (1.0f + xi * dt);
        L_y = L_y * (1.0f + xi * dt);
        L_z = L_z * (1.0f + xi * dt);
    }
}

```

```

volume = L_x * L_y * L_z;
density = N / volume;
err = (float)(1.0E-5);
kinetic_energy = 0;
for (int i = 0; i < N; i++)
{
    v_x[i] = vel_x[i];
    v_y[i] = vel_y[i];
    v_z[i] = vel_z[i];
    kinetic_energy += v_x[i]*v_x[i]+v_y[i]*v_y[i]+v_z[i]*v_z[i];
}
p_s = lambda;
ready = false;
iter = 0;
while ((ready == false) && (iter < 50))
{
    iter = iter + 1;
    p_o = p_s;
    ds = 0;
    for (int i = 0; i < N; i++)
    {
        v_xo[i] = v_x[i];
        v_yo[i] = v_y[i];
        v_zo[i] = v_z[i];
        b_x[i] = -(dt/2.0f) * (acc_x[i]-p_o*v_x[i]) - (vel_x[i]-v_xo[i]);
        ds = ds + v_xo[i] * dt * b_x[i] / Q_T;
        b_y[i] = -(dt/2.0f) * (acc_y[i]-p_o*v_y[i]) - (vel_y[i]-v_yo[i]);
        ds = ds + v_yo[i] * dt * b_y[i] / Q_T;
        b_z[i] = -(dt/2.0f) * (acc_z[i]-p_o*v_z[i]) - (vel_z[i]-v_zo[i]);
        ds = ds + v_zo[i] * dt * b_z[i] / Q_T;
    }
    dd = -(p_o * (dt / 2.0f) + 1);
    ds = ds - dd * ((-kinetic_energy + 3.0f * N * T0) * (dt / 2.0f) /
Q_T - (lambda - p_o));
    ds = ds / (-dt * (dt / 2.0f) * kinetic_energy / Q_T + dd);
    kinetic_energy = 0;
    for (int i = 0; i < N; i++)
    {
        v_x[i] = v_x[i] + (b_x[i] + (dt / 2.0f) * v_xo[i] * ds) / dd;
        v_y[i] = v_y[i] + (b_y[i] + (dt / 2.0f) * v_yo[i] * ds) / dd;
        v_z[i] = v_z[i] + (b_z[i] + (dt / 2.0f) * v_zo[i] * ds) / dd;
        kinetic_energy += v_x[i]*v_x[i]+v_y[i]*v_y[i]+v_z[i]*v_z[i];
    }
}

```

```

p_s = p_o + ds;
ready = true;
int j = 0;
while ((j <= N) && (ready))
{
    if (j < N)
    {
        if (Math.Abs((v_x[j]-v_xo[j])/v_x[j])>err) ready=false;
        if (Math.Abs((v_y[j]-v_yo[j])/v_y[j])>err) ready=false;
        if (Math.Abs((v_z[j]-v_zo[j])/v_z[j])>err) ready=false;
    }
    else
    {
        if (Math.Abs((p_s-p_o)/p_s)>err) ready=false;
    }
    j++;
}
for (int i = 0; i < N; i++)
{
    vel_x[i] = v_x[i];
    vel_y[i] = v_y[i];
    vel_z[i] = v_z[i];
}
lambda = p_s;
kinetic_energy = kinetic_energy / 2.0f;
H_energy = kinetic_energy + potential_energy + (lambda * lambda *
Q_T) / 2.0f + 3.0f * N * T0 * S_f;
}
}

```

§1.7 Сила и ускорение частиц

На основе заданного потенциала межчастичного взаимодействия можно рассчитать потенциальную энергию (как удельную, так и всей системы), силу, действующую на частицу, а также ускорение частиц. Потенциальная энергия, приходящаяся на одну частицу системы есть [2, 4]

$$\phi_i = \sum_{j>i}^N U(r_{ij})\Theta(r_{cut} - r_{ij}), \quad i = 1, 2, \dots, N, \quad (43)$$

где Θ – есть функция Хевисайда:

$$\Theta = \begin{cases} 0 & r_{ij} > r_{cut} \\ 1 & r_{ij} \leq r_{cut} \end{cases}. \quad (44)$$

Здесь r_{cut} – есть некоторое граничное расстояние (расстояние усечения потенциала), на котором взаимодействие частиц исчезает и, следовательно, потенциальная энергия взаимодействия становится равной нулю. Полная энергия системы определяется через суммирование по всем частицам

$$E_{pot} = \sum_{i=1}^N \phi_i. \quad (45)$$

Здесь каждую атомную пару необходимо рассмотреть только один раз.

Сила \vec{F}_i , действующая на i -ую частицу со стороны окружения, определяется через градиент от потенциала межчастичного взаимодействия, как это было показано в §1.2. Например, в случае потенциала Леннард-Джонса (65) величина \vec{F}_i имеет вид [2, 4, 17]

$$\vec{F}_i = - \sum_{i=1}^N \frac{dU(r_{ij})}{r_{ij}} \frac{\vec{r}_{ij}}{r_{ij}} = \frac{24\epsilon}{\sigma} \left[2 \left(\frac{\sigma}{r_{ij}} \right)^{13} - \left(\frac{\sigma}{r_{ij}} \right)^7 \right] \frac{\vec{r}_{ij}}{r_{ij}}. \quad (46)$$

В случае потенциала Джугутова величина \vec{F}_i определяется как

$$\vec{F}_i = - \sum_{i=1}^N \left[\phi_1(r_{ij}) \Theta \left(a - \frac{r_{ij}}{\sigma} \right) + \phi_2(r_{ij}) \Theta \left(b - \frac{r_{ij}}{\sigma} \right) \right] \frac{\vec{r}_{ij}}{r_{ij}}, \quad (47)$$

где

$$\phi_1(r_{ij}) = -A\epsilon \left\{ \frac{m}{\sigma} \left(\frac{\sigma}{r_{ij}} \right)^{m+1} + \left(\frac{\sigma}{r_{ij}} \right)^m - D \right\} \exp \left(\frac{\sigma c}{r_{ij} - \sigma a} \right), \quad (48)$$

$$\phi_2(r_{ij}) = -\frac{B\sigma d}{(r_{ij} - \sigma b)^2} \exp \left(\frac{\sigma d}{r_{ij} - \sigma b} \right). \quad (49)$$

Согласно первому закону Ньютона ускорения взаимодействующих частиц \vec{a}_i и \vec{a}_j определяются следующим образом:

$$\vec{a}_i = \frac{\vec{F}_i}{m}, \quad \vec{a}_j = \frac{\vec{F}_j}{m}. \quad (50)$$

§1.8 Оптимизация расчетов. Список Верле

Моделирование молекулярной динамики связано с большим числом вычислительных операций. При выполнении процедуры `load_forces` происходит порядка $(N - 1)N/2$ вычислительных операций для одного временного интервала Δt . Вполне понятно, что количество операций также определяется размером пространственной области действия потенциала. Большинство потенциалов межчастичного взаимодействия $U(r_{ij})$ являются короткодействующими с радиусом действия r_{cut} (т.е. при $r_{ij} > r_{cut} \Rightarrow U(r_{ij}) = 0$). Это означает, что расчет силы взаимодействия выполняется в ограниченной области, характеризуемой воображаемой сферой радиуса r_{cut} (для трехмерной системы). Для сокращения количества операций в МД-моделировании используются различные алгоритмы. Наиболее широко распространен так называемый алгоритм «список Верле» (Verlet list) [4, 18].

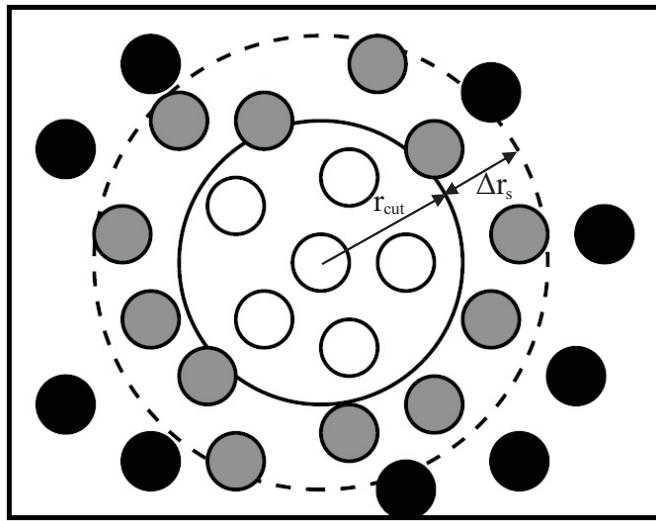


Рис. 5: Схематичный рисунок расположения частиц при использовании алгоритма «список Верле».

В алгоритме «список Верле» в пределах сферы радиусом r_{cut} определяется число «соседей» для каждой частицы. Далее строится воображаемая сфера радиусом r_s ($r_s > r_{cut}$). После этого формируется список частиц, которые располагаются в сферическом слое толщиной $\Delta r_s = r_s - r_{cut}$ [4, 18]. Схема представлена на рисунке 5. Список частиц внутри сферы радиусом r_s должен быть сформирован прежде, чем частицы, расположенные изначально за пределами r_s , попадут внутрь сферы. Процесс обновления списка может выполняться вручную или может быть автоматизирован с помощью до-

полнительных алгоритмов, при условии, что список частиц предыдущего обновления должен быть сохранен. Толщина сферического слоя Δr_s выбирается из соображений приемлемой частоты обновления списка. Необходимо выбирать оптимальную толщину Δr_s , поскольку чрезмерно частое обновление может замедлить расчеты. В зависимости от размера системы толщина слоя обычно составляет от $\Delta r_s = 0.2\sigma$ до $\Delta r_s = 0.5\sigma$ (σ – эффективный размер частицы) [4].

Ниже приведен программный код 1.8 для процедуры `load_forces`, адаптированной с использованием алгоритма «список Верле».

Код 1.8

```
public void load_forces()
{
    for (int i = 0; i < N; i++)
    {
        acc_x[i] = 0; acc_y[i] = 0; acc_z[i] = 0; e_pot[i] = 0;
    }
    virial = 0;
    // проверка на обновление «списка Верле»
    if (update_test)
    {
        update_list(r_cut + dr_s);
        update_test = false;
    }
    for (int i = 0; i < N - 1; i++)
    {
        for (int L = Marker1[i]; L <= Marker2[i]; L++)
        {
            J = List[L];
            rij_x = pos_x[i] - pos_x[J];
            rij_y = pos_y[i] - pos_y[J];
            rij_z = pos_z[i] - pos_z[J];
            rij_x = rij_x - Convert.ToInt32(rij_x / L_x) * L_x;
            rij_y = rij_y - Convert.ToInt32(rij_y / L_y) * L_y;
            rij_z = rij_z - Convert.ToInt32(rij_z / L_z) * L_z;
            rij2 = rij_x * rij_x + rij_y * rij_y + rij_z * rij_z;
            if (rij2 < (r_cut * r_cut))
            {
                ri = (int)((Math.Pow(rij2, 0.5) - 0.5f) / 0.0001);
                dP = table_pot[ri - 1, 0];
                rij_x = rij_x - rij2 * rij_x / rij2 * dP;
                rij_y = rij_y - rij2 * rij_y / rij2 * dP;
                rij_z = rij_z - rij2 * rij_z / rij2 * dP;
                pos_x[i] = pos_x[i] + rij_x;
                pos_y[i] = pos_y[i] + rij_y;
                pos_z[i] = pos_z[i] + rij_z;
            }
        }
    }
}
```

```
    e_pot[i] = e_pot[i] + (dP / 2.0f);
    e_pot[J] = e_pot[J] + (dP / 2.0f);
    dF = table_pot[ri - 1, 1];
    virial = virial + dF * rij2;
    acc_x[i] = acc_x[i] + dF * rij_x;
    acc_y[i] = acc_y[i] + dF * rij_y;
    acc_z[i] = acc_z[i] + dF * rij_z;
    acc_x[J] = acc_x[J] - dF * rij_x;
    acc_y[J] = acc_y[J] - dF * rij_y;
    acc_z[J] = acc_z[J] - dF * rij_z;
}
}

virial = virial / 3.0f;
potential_energy = 0;
for (int i = 0; i < N; i++) potential_energy += e_pot[i];
for (int i = 0; i < N; i++)
{
    // определяется смещение частиц
    disp_list_x[i] += vel_x[i] * dt + acc_x[i] * dt * dt / 2.0f;
    disp_list_y[i] += vel_y[i] * dt + acc_y[i] * dt * dt / 2.0f;
    disp_list_z[i] += vel_z[i] * dt + acc_z[i] * dt * dt / 2.0f;
}
load_move_test(dr_s);
update_test = move_test;
}

public void update_list(float rang)
{
    // обновление «списка Верле»
    max_list_size = N;
    float rng;
    rng = rang * rang;
    int L = 1;
    for (int i = 0; i < N - 1; i++)
    {
        for (int j = i + 1; j < N; j++)
        {
            rij_x = pos_x[i] - pos_x[j];
            rij_y = pos_y[i] - pos_y[j];
            rij_z = pos_z[i] - pos_z[j];
            rij_x = rij_x - Convert.ToInt32(rij_x / L_x) * L_x;
            rij_y = rij_y - Convert.ToInt32(rij_y / L_y) * L_y;
            rij_z = rij_z - Convert.ToInt32(rij_z / L_z) * L_z;
        }
    }
}
```

```

rij2 = rij_x * rij_x + rij_y * rij_y + rij_z * rij_z;
if (rij2 < rng)
{
    Advance[j] = 1;
}
else
{
    Advance[j] = 0;
}
}

Marker1[i] = L;
for (int j = i + 1; j < N; j++)
{
    List[L] = j;
    L = L + Advance[j];
}
Marker2[i] = L - 1;
}

list_size = L - 1;
for (int i = 0; i < N; i++)
{
    disp_list_x[i] = 0;
    disp_list_y[i] = 0;
    disp_list_z[i] = 0;
}
}

public void load_move_test(float skin)
{
    ds1 = 0; ds2 = 0;
    for (int i = 0; i < N; i++)
    {
        ds = (float)Math.Sqrt(disp_list_x[i] * disp_list_x[i] +
        disp_list_y[i] * disp_list_y[i] + disp_list_z[i] * disp_list_z[i]);
        if (ds >= ds1)
        {
            ds2 = ds1;
            ds1 = ds;
        }
        else
        if (ds >= ds2)
        {
            ds2 = ds;
        }
    }
}

```

```
    }
    move_test = (ds1 + ds2 > skin);
}
```

Процедура `update_list` обновляет список частиц `List`, расположенных внутри воображаемой сферической области с радиусом $r_{cut} + \Delta r_s$ (`r_cut + dr_s`). Обновление осуществляется автоматически при `move_test = true` после выполнения проверки с помощью процедуры `load_move_test`.

Глава 2. Потенциалы взаимодействия

§2.1 Виды потенциалов

С позиций статистической физики свойства конкретного материала (вещества) определяются непосредственно потенциалом межчастичного взаимодействия. Это может быть некоторая общая для всех молекул, формирующих систему, модель взаимодействия, или же набор различных моделей, воспроизводящих взаимодействия молекул, атомов, ионов различного сорта. Модельный потенциал должен как можно корректнее характеризовать физическую (химическую) природу межчастичного взаимодействия в веществе и, тем самым, верно отображать свойства рассматриваемой системы. При моделировании классической молекулярной динамики потенциал межчастичного взаимодействия, как правило, задается в готовом виде. В качестве основания для выбора конкретной модели потенциала могут служить соображения о наличие в многочастичной системе какого-то конкретного типа межчастичного взаимодействия: например, диполь-дипольного, ион-ионного, ковалентного и т. д. С другой стороны, модельный потенциал для какой-то конкретной системы может представлять собой результат квантово-механических расчетов; предварительных вычислений, связанных с так называемым первопринципным (*ab-initio*) моделированием. Далее, модельные потенциалы могут также конструироваться через решение *обратной задачи* – через рассмотрение различных структурных характеристик, представленных экспериментальными данными, например, по рентгеновской и/или конкретной дифракции. Такой подход реализуется, например, в методе *реверсивного Монте-Карло* моделирования [2, 4].

Изотропным (или сферическим) называется потенциал, в котором взаимодействие между частицами определяется лишь их взаим-

ным удалением. В этом случае потенциал зависит лишь от расстояния между частицами. В качестве примеров сферических потенциалов можно привести потенциал твердых сфер, потенциал Леннард-Джонса, потенциал Юкавы и другие. В *анизотропных* потенциалах кроме зависимости от расстояния присутствует также зависимость от направления. Примеры анизотропных потенциалов – потенциал Стиллинжера-Вебера и потенциал Терсофа.

Следует отметить, что конструирование потенциалов межчастичного взаимодействия применительно к какому-либо конкретному материалу (веществу) представляет собой отдельную, крайне важную задачу, решением которой занимаются многие современные научно-исследовательские группы. Примечательным является то, что решение этой задачи соотносится непосредственно с другой актуальной задачей – созданием новых перспективных материалов.

В настоящее время существует большое количество потенциалов, которые позволяют рассчитать энергию и силу взаимодействия частиц самых разнообразных систем: от простых атомарных, до сложных молекулярных. В случае классических одноатомных систем, взаимодействие частиц описывается через потенциалы, общий вид которых можно представить как

$$U = \sum_i u(r_i) + \sum_{i,j} u(r_i, r_j) + \sum_{i,j,k} u(r_{ij}, r_{ik}, \theta_{ijk}) + \dots, \quad (51)$$

где под r_{ij} подразумевается расстояние между *центрами частиц*. Так, например, модельный потенциал Морзе [13]

$$U(r_{ij}) = D \{ \exp [-2\alpha(r_{ij} - r_{cut})] - 2 \exp [-\alpha(r_{ij} - r_{cut})] \}, \quad (52)$$

используется для моделирования свойств таких металлов, как Cu, Au, Ag, Ni, Pb и т. д. В случае системы, состоящей из атомов Au, параметры потенциала: $D = 0.4826 \text{ eV}$, $\alpha = 16.16 \text{ нм}^{-1}$, $r_{cut} = 0.3 \text{ нм}$. Для моделирования металлических систем, таких как Al, Cu, используется модель «погруженного атома» (Embedded Atom Method potential) [14]:

$$U(r_{ij}) = \frac{1}{2} \sum_{i=1}^{N-1} \sum_{j \neq i}^N \phi_{ij} + \sum_{i=1}^N F(\rho_i), \quad (53)$$

где

$$\rho_i = \sum_{j \neq i}^N f_j(r_{ij}), \quad \phi_{ij} = A_{ij} \exp(-r_{ij}/B_{ij}), \quad (54)$$

$$f_i(r_{ij}) = c_i \exp[-b_i(r_{ij} - d_i)], \quad F(\rho_i) = -\rho_i^{1/2}. \quad (55)$$

Здесь $U(r_{ij})$ – есть потенциал парного взаимодействия атомов, $F(\rho_i)$ – многочастичная функция, ρ_i – электронная плотность вблизи i -го атома, значения параметров b_i , d_i и c_i задаются в зависимости от типа системы [14].

Потенциал непременно учитывает размер атомов/молекул и энергии связи между ними. Кроме этого, в сложных молекулярных системах необходимо учитывать взаимное расположение молекул друг относительно друга. В этом случае используется потенциал вида [2, 4]

$$U = U_{bond} + U_{angle} + U_{tors} + U_{pair} + U_{charge}, \quad (56)$$

где вклад

$$U_{bond} = \sum_{i,j} \kappa_r (r_{ij} - r_{0ij})^2 \quad (57)$$

характеризует энергию межмолекулярных связей некоторого равновесного радиуса действия r_{0ij} , где κ_r – некоторая постоянная, r_{ij} – есть расстояние между молекулами i и j . Вклад

$$U_{angle} = \sum_{i,j,k} \kappa_\theta (\theta_{ijk} - \theta_{0ijk})^2 \quad (58)$$

характеризует взаимную ориентацию молекул относительно некоторого равновесного угла θ_{0ijk} , где κ_θ – постоянная величина, θ_{ijk} – есть угол между двумя воображаемыми прямыми, соединяющими центры молекул i , j и k . Вклад вида

$$U_{tors} = \sum_{i,j} \kappa_\varphi [1 + \cos(\varphi_{ij})] \quad (59)$$

определяет энергию кручения молекул. Здесь коэффициент κ_φ определяется типом системы, φ_{ij} – есть торсионный угол между молекулами i и j . Энергия парного взаимодействия определяется потенциалом Леннард-Джонса [4, 15, 16]. В результате полная потенциальная энергия N -частичной системы:

$$U_{pair} = \sum_{i=1}^{N-1} \sum_{j>i}^N \left\{ 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \right\}. \quad (60)$$

Энергия системы N частиц-зарядов есть

$$U_{charge} = \sum_{i=1}^{N-1} \sum_{j>i}^N \left\{ \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}, \quad (61)$$

где q_i и q_j – есть заряды i -го и j -го атомов, ϵ_0 – электрическая постоянная ($\epsilon_0 \simeq 8.85 \cdot 10^{-12} \Phi/\text{м}$).

Для моделирования свойств ионных систем используются парные потенциалы, где учитывается кулоновское взаимодействие, например, потенциал следующего вида [2]

$$U(r_{ij}) = \sum_{i,j}^N \left[\frac{D_1}{r_{ij}^6} - \frac{D_2}{r_{ij}^8} + D_3 \frac{q_i q_j}{r_{ij}} \right], \quad (62)$$

где параметры D_1 , D_2 и D_3 задаются в зависимости от типа системы.

В методе моделирования молекулярной динамики в зависимости от системы и решаемой задачи могут использоваться различные способы представления единиц измерения физических величин (см. таблицу 1) [1, 2, 4].

Таблица 1: Единицы измерения некоторых физических величин, используемых в методе моделирования МД: (СИ) – международная система единиц, (LJ) – безразмерные леннард-джонсовские единицы измерения физических величин. Здесь m – масса частицы, σ – расстояние, на котором энергия взаимодействия становится равной нулю (эффективный размер частицы), ϵ – глубина потенциальной ямы, $k_B \simeq 1.38 \cdot 10^{-23}$ Дж/К – постоянная Больцмана.

Физические величины	СИ	LJ
масса	кг	m
расстояние	м	σ
время	с	$\tau = \sigma \sqrt{(m/\epsilon)}$
энергия	кг·м ² /с ²	ϵ
скорость	м/с	σ/τ
сила	кг·м/с ²	ϵ/σ
температура	К	ϵ/k_B
давление	кг·/(м·с ²)	ϵ/σ^3
плотность	кг/м ³	m/σ^3

§2.2 Модели твердых и мягких сфер

Потенциалы взаимодействия являются функциями расстояния между атомами, молекулами или ионами. Характер межчастичного взаимодействия в МД-моделировании задается через различные модели. К простейшим моделям можно отнести модели твердых и мягких сфер [17]. В модели твердых сфер столкновение атомов является абсолютно упругим. Здесь предполагается, что между частицами нет

ни притяжения, ни отталкивания, а соударение частиц является абсолютно упругим [4, 19].

$$U(r_{ij}) = \begin{cases} \infty & r_{ij} \leq \sigma \\ 0 & r_{ij} > \sigma \end{cases}. \quad (63)$$

Такой вид взаимодействия преимущественно используется в физической кинетике нейтральных газов, где рассеяние частиц происходит за счет отталкивателной части потенциала взаимодействия (см. рисунок 6а).

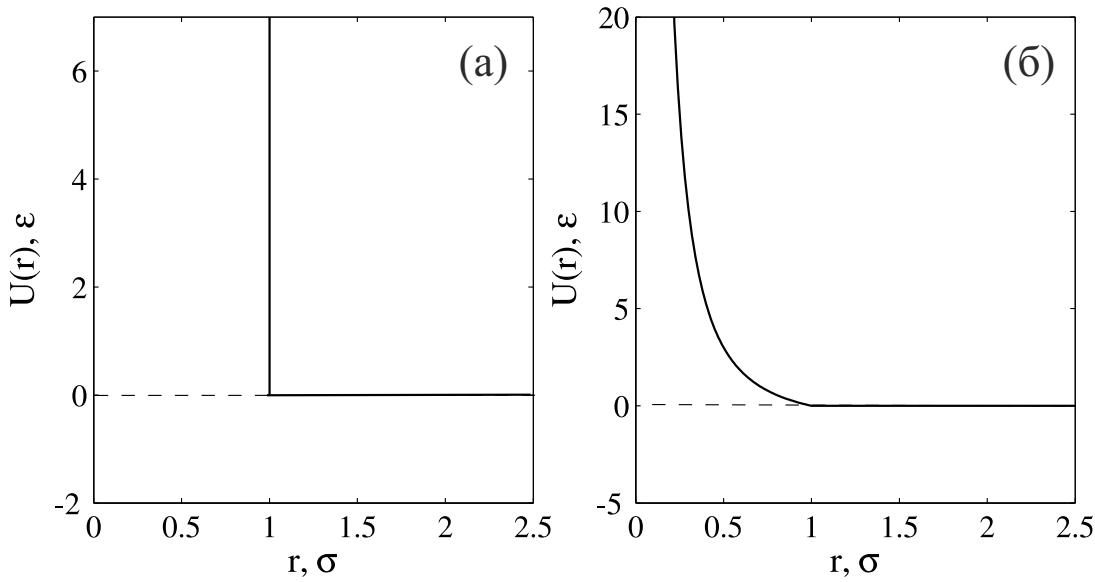


Рис. 6: (а) – потенциал взаимодействия в модели твердых сфер, определяемый выражением (63) при $\sigma = 1$; (б) – потенциал взаимодействия в модели мягких сфер, определяемый через выражение (64) при $n = 2$, $\epsilon = 1$ и $\sigma = 1$.

Потенциал мягких сфер характеризуется степенной зависимостью энергии взаимодействия частиц от расстояния между ними:

$$U(r_{ij}) = \begin{cases} \epsilon(\sigma/r_{ij})^n & r_{ij} \leq \sigma, \text{ где } n > 0 \\ 0 & r_{ij} > \sigma \end{cases}. \quad (64)$$

Очевидно, что в случае больших значений n потенциал (64) приближается к модели твердых сфер (63) (см. рисунок 6б).

§2.3 Парные потенциалы Леннард-Джонса и Джугутова

Потенциал Леннард-Джонса (LJ) – модель парного взаимодействия частиц, предложенная в 1924 году Д. Э. Леннард-Джонсом

[15, 16], которая достаточно точно воспроизводит взаимодействие атомов в таких системах как инертные газы (Ar, Kr, Xe и т. д.). Потенциал Леннард-Джонса записывается следующим образом

$$U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] - U(r_{cut}), \quad r \leq r_{cut}, \quad (65)$$

где r_{ij} есть расстояние между частицами i и j . В обобщенном виде (n-m Lennard-Jones potential) можно записать

$$U(r_{ij}) = \epsilon \left(\frac{n}{n-m} \right) \left(\frac{n}{m} \right)^{\left(\frac{m}{n-m} \right)} \left[\left(\frac{\sigma}{r_{ij}} \right)^n - \left(\frac{\sigma}{r_{ij}} \right)^m \right] - U(r_{cut}), \quad (66)$$

где $n > m$ (в случае потенциала LJ имеем $n = 12$ и $m = 6$) [20]. Таким образом, потенциал LJ имеет минимум в точке $r_{min} = \sigma\sqrt[6]{2}$ (см. рисунок 7). На расстоянии $r_{ij} \simeq 2.5\sigma$ энергия взаимодействия составляет малое значение, равное $\approx -0.063\epsilon$. Поэтому LJ-потенциал как правило «обрезается» на расстоянии $r_{cut} > 2.5\sigma$ и сдвигается на величину $-U(r_{cut})$. Параметры ϵ и σ потенциала определяются типом системы. Например, в таблице 2 приведены значения величин ϵ и σ для некоторых инертных газов.

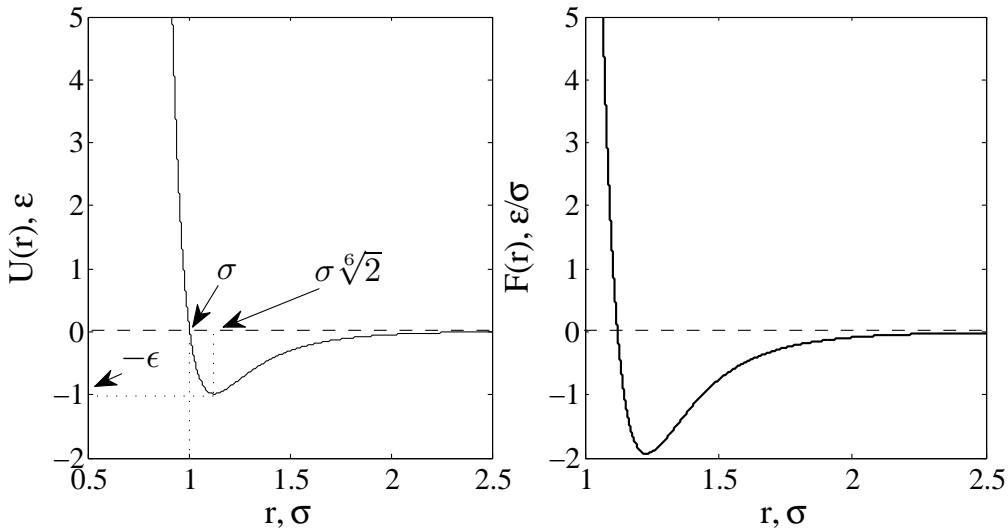


Рис. 7: Потенциал межчастичного взаимодействия Леннард-Джонса: (а) – вид потенциала; (б) – сила межчастичного взаимодействия, определяемая через градиент от потенциала.

В таблице 3 приводится соответствие между безразмерными леннард-джонсовскими единицами и с единицами измерения физических величин в системе СИ для случая аргона.

Таблица 2: Параметры потенциала Леннард-Джонса для гелия, неона, аргона и криптона.

Система	ϵ/k_B , К	σ , нм
He – гелий	10.2	0.228
Ne – неон	47.0	0.272
Ar – аргон	119.8	0.341
Kr – криптон	164.0	0.382

Таблица 3: Соответствие между безразмерными леннард-джонсовскими единицами и реальными физическими единицами в случае модели, описывающей жидкый аргон, где $\sigma = 0.341$ нм и $\epsilon = 119.8 K \cdot k_B$.

Физические величины	LJ	СИ
масса	m	$6.69 \cdot 10^{-26}$ кг
расстояние	σ	$3.4 \cdot 10^{-10}$ м
время	$\tau = \sigma \sqrt{m/\epsilon}$	$2.17 \cdot 10^{-12}$ с
энергия	ϵ	$1.65 \cdot 10^{-21}$ кг·м ² /с ² (Дж)
скорость	σ/τ	$1.57 \cdot 10^2$ м/с
сила	ϵ/σ	$4.85 \cdot 10^{-12}$ кг·м/с ² (Н)
температура	ϵ/k_B	119.8 К
давление	ϵ/σ^3	$4.2 \cdot 10^7$ кг/(м·с ²) (Па)

В случае двухкомпонентных систем, состоящих из сферически симметричных частиц α - и β -типов, LJ-потенциал принимает вид [21, 22, 23, 24, 25]

$$U_{\alpha\beta}(r_{ij}) = 4\epsilon_{\alpha\beta} \left[\left(\frac{\sigma_{\alpha\beta}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r_{ij}} \right)^6 \right] - U_{\alpha\beta}(r_{cut}). \quad (67)$$

Параметры $\epsilon_{\alpha\beta}$ и $\sigma_{\alpha\beta}$ характеризуют перекрестное взаимодействие между компонентами и определяются на основе так называемых *правил смешивания*. В качестве примера приведем полуэмпирические комбинационные правила смешивания Лоренца-Бертло [21, 26]:

$$\epsilon_{\alpha\beta} = \sqrt{\epsilon_{\alpha\alpha}\epsilon_{\beta\beta}}, \quad (68)$$

$$\sigma_{\alpha\beta} = \frac{\sigma_{\alpha\alpha} + \sigma_{\beta\beta}}{2}. \quad (69)$$

Правила смешивания, задаваемые выражениями (68) и (69), используются в случае таких двухкомпонентных систем, как смеси инертных газов Ar, Kr, Xe с различными углерод- и водородсодержащими

соединениями. Для так называемой системы Ванстрема (Wanstrom) потенциал взаимодействия частиц имеет вид выражения (67) при $\alpha = 50\%$ и $\beta = 50\%$ [23], где значения параметров $\sigma_{\alpha\beta}$ и $\epsilon_{\alpha\beta}$ определяются через правила смешивания (68) и (69). В случае бинарной системы Коба-Андерсена потенциал межчастичного взаимодействия также имеет вид (67), где $\alpha = 80\%$ и $\beta = 20\%$. В этой модели используются неполные правила смешивания Лоренца-Бертло (см. таблицу 4 и рисунок 8). Модель Коба-Андерсена была предложена в 1995 году для исследования свойств таких смесей газов, как, Ar-CH₄, Xe-CF₄, CO-C₂H₆ [22]. Системы Коб-Андерсена и Ванстрема хорошо аморфизуются, что позволяет их использовать для исследования процессов витрификации.

Таблица 4: Параметры потенциала Леннард-Джонса для двухкомпонентных систем Ванстрема (W) и Коб-Андерсена (KA).

Система	$\sigma_{\alpha\alpha}$	$\sigma_{\beta\beta}$	$\sigma_{\alpha\beta}$	$\epsilon_{\alpha\alpha}$	$\epsilon_{\beta\beta}$	$\epsilon_{\alpha\beta}$
W	1.0σ	0.8σ	0.9σ	1.0ϵ	1.0ϵ	1.0ϵ
KA	1.0σ	0.88σ	0.8σ	1.0ϵ	0.5ϵ	1.5ϵ

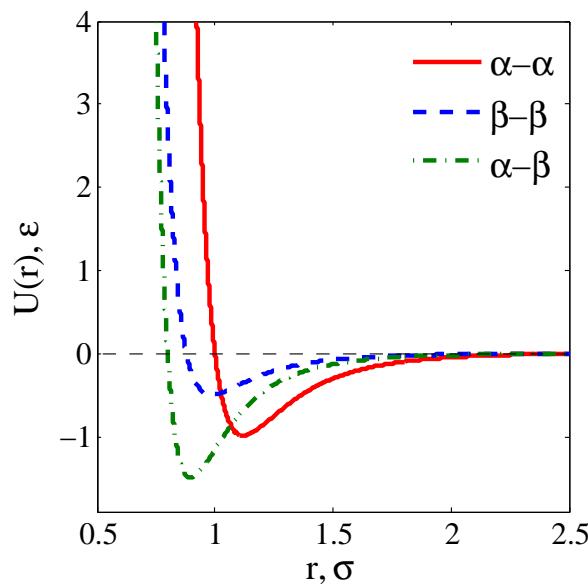


Рис. 8: Вид потенциала Леннард-Джонса для двухкомпонентной системы Коба-Андерсена. Отдельные кривые обозначают зависимость потенциальной энергии от расстояния при перекрестных взаимодействиях между частицами α и β -типа.

Потенциал Джугутова (Dz) был предложен в 90-е годы XX века для моделирования свойств металлических расплавов (свинца

Pb, V₃Ni₂, V₁₅Ni₁₀Si) [25, 27, 28, 29].

Потенциал Джугутова является парным и представляет собой комбинацию двух вкладов:

$$U(r_{ij}) = \Phi_1(r_{ij}) + \Phi_2(r_{ij}), \quad (70)$$

где

$$\Phi_1(r_{ij}) = A\epsilon \left[\left(\frac{r_{ij}}{\sigma} \right)^{-m} - B \right] \exp \left(\frac{\sigma c}{r_{ij} - \sigma a} \right) \Theta(a - r_{ij}/\sigma), \quad (71)$$

$$\Phi_2(r_{ij}) = B\epsilon \exp \left(\frac{\sigma d}{r_{ij} - \sigma b} \right) \Theta(b - r_{ij}/\sigma). \quad (72)$$

Численные значения параметров Dz-потенциала представлены в таблице 5. Величина Φ_1 характеризует энергию отталкивания на малых расстояниях с характерной потенциальной ямой при $r_{ij} \simeq 1.13\sigma$ и $U(r_{ij}) = -0.581\epsilon$. Вклад Φ_2 определяет энергию взаимодействия вблизи первой координационной сферы с максимумом потенциальной энергии на расстоянии $r_{ij} \simeq 1.63\sigma$, высота которого составляет $U(r_{ij}) \simeq 0.46\epsilon$. Вид потенциала Джугутова представлен на рисунке 9.

Таблица 5: Численные значения параметров Dz-потенциала, которые задают вид потенциала и являются едиными для различных систем.

	A	B	a	b	c	d	m
Dz	5.82	1.28	1.87	1.94	1.1	0.27	16

§2.4 Трехчастичный потенциал Стиллинжера-Вебера

Трехчастичный потенциал Стиллинжера-Вебера (SW) предложен в 1985 году для изучения свойств полупроводникового кремния, а также для численного моделирования процессов образования тетраэдрических структур в конденсированных средах [30]. Существует множество модификаций SW-потенциала применительно к германию (Ge), железу (Fe), никелю (Ni), соединениям Si-O-Cl и Si-O-F, а также для воды [30, 31, 32].

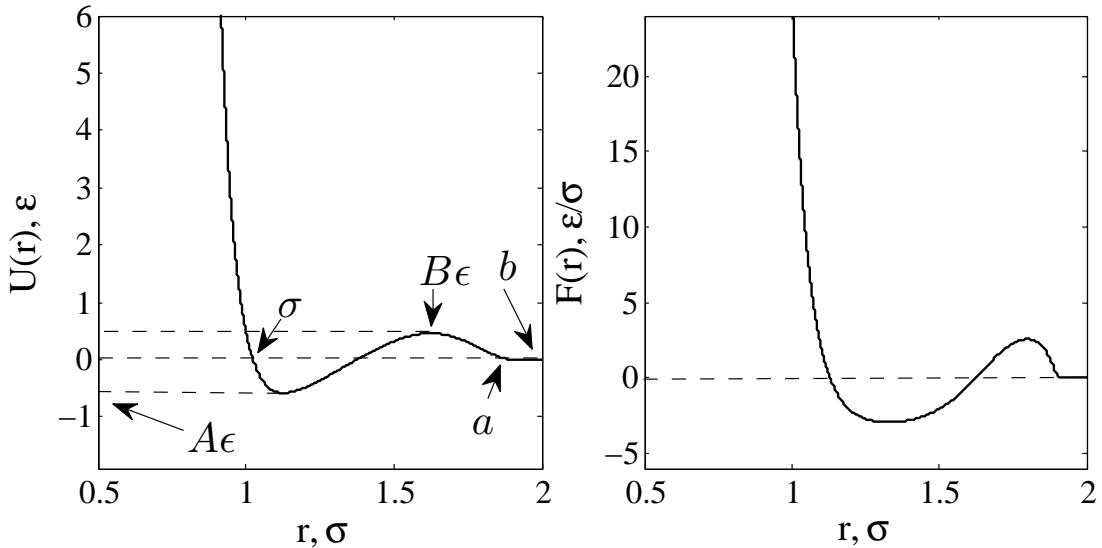


Рис. 9: (а) – вид потенциала Джугутова как функции от расстояния между частицами; (б) – сила межчастичного взаимодействия, определяемая через градиент от потенциала Джугутова.

Потенциал Стиллинжера-Вебера состоит из суммы двух вкладов – парного ϕ_2 и трехчастичного ϕ_3 :

$$U = \sum_{i,j} \phi_2(r_{ij}) + \sum_{i,j,k} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}), \quad (73)$$

где θ_{ijk} определяет угол между частицами i , j и k (см. рисунок 10). Выражение для парного вклада имеет вид

$$\phi_2(r_{ij}) = A\epsilon \left[B \left(\frac{\sigma}{r_{ij}} \right)^p - \left(\frac{\sigma}{r_{ij}} \right)^q \right] \exp \left(\frac{\sigma}{r_{ij} - r_{cut}} \right). \quad (74)$$

Трехчастичный вклад имеет вид

$$\begin{aligned} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) &= \\ &= \lambda\epsilon [\cos \theta_{ijk} - \cos \theta_{0ijk}]^2 \exp \left(\frac{\gamma\sigma}{r_{ij} - r_{cut}} \right) \exp \left(\frac{\gamma\sigma}{r_{ik} - r_{cut}} \right). \end{aligned} \quad (75)$$

Параметры A , B , p , q , λ , γ , θ_{0ijk} задаются таким образом, чтобы потенциал мог воспроизводить структурные и динамические свойства исследуемой системы максимально приближенно к реальным. Наличие угловой зависимости в выражении (75) позволяет воспроизводить тетраэдрическую структуру. В качестве примера в таблице 6 представлены численные значения параметров SW-потенциала для Si, Ge и воды [30, 31, 32].

Потенциал SW является короткодействующим, с радиусом действия $r_{cut} = 1.8\sigma$. Потенциал Стиллинжера-Вебера требует значи-

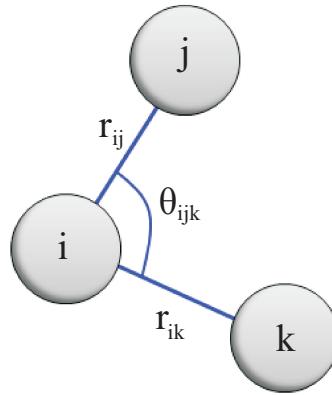


Рис. 10: Схематичный рисунок расположения трех частиц, где обозначен угол θ_{ijk} между воображаемыми прямыми, соединяющими центры частиц i , j и k , а также расстояния r_{ij} и r_{ik} между частицами ij и ik .

Таблица 6: Численные значения параметров потенциала Стиллинжера-Вебера для Si, Ge, а также для mW-модели воды.

Параметры	Si	Ge	mW
A	$\simeq 7.049$	$\simeq 7.049$	$\simeq 7.049$
B	$\simeq 0.602$	$\simeq 0.602$	$\simeq 0.602$
p	4	4	4
q	0	0	0
λ	21	31	23.15
γ	1.2	1.2	1.2
θ_{0ijk}	109.47°	109.47°	109.47°
σ	$\simeq 2.095 \text{\AA}$	$\simeq 2.18 \text{\AA}$	$\simeq 2.39 \text{\AA}$
ϵ	$\simeq 2.168 \text{eV}$	$\simeq 1.93 \text{eV}$	$\simeq 0.268 \text{eV}$

тельный объем вычислений, а следовательно, гораздо больше вычислительных ресурсов по сравнению с парными потенциалами.

Ниже приведен программный код 2.1 процедуры `load_forces`, используемой для расчета потенциальной энергии и ускорения частиц системы через потенциалы взаимодействия Леннард-Джонса и Джугутова.

Код 2.1

```
public void load_forces()
{
    for (int i = 0; i < N; i++)
    {

```

```

    acc_x[i] = 0; acc_y[i] = 0; acc_z[i] = 0; e_pot[i] = 0;
}
virial = 0;
for (int i = 0; i < N - 1; i++)
{
    for (J = 0; J < N; J++)
    {
        rij_x = pos_x[i] - pos_x[J];
        rij_y = pos_y[i] - pos_y[J];
        rij_z = pos_z[i] - pos_z[J];
        rij_x = rij_x - Convert.ToInt32(rij_x / L_x) * L_x;
        rij_y = rij_y - Convert.ToInt32(rij_y / L_y) * L_y;
        rij_z = rij_z - Convert.ToInt32(rij_z / L_z) * L_z;
        rij2 = rij_x * rij_x + rij_y * rij_y + rij_z * rij_z;
        if (rij2 < (r_cut * r_cut))
        {
            ri = (int)((Math.Pow(rij2, 0.5) - 0.5f) / 0.0001);
            dP = table_pot[ri - 1, 0];
            e_pot[i] = e_pot[i] + (dP / 2.0f);
            e_pot[J] = e_pot[J] + (dP / 2.0f);
            dF = table_pot[ri - 1, 1];
            virial = virial + dF * rij2;
            acc_x[i] = acc_x[i] + dF * rij_x;
            acc_y[i] = acc_y[i] + dF * rij_y;
            acc_z[i] = acc_z[i] + dF * rij_z;
            acc_x[J] = acc_x[J] - dF * rij_x;
            acc_y[J] = acc_y[J] - dF * rij_y;
            acc_z[J] = acc_z[J] - dF * rij_z;
        }
    }
}
virial = virial / 3.0f;
potential_energy = 0;
for (int i = 0; i < N; i++) potential_energy += e_pot[i];
}

```

Потенциальная энергия и сила межчастичного взаимодействия в процедуре `load_forces` извлекаются из заранее рассчитанного для системы массива `table_pot` длиной `table_size`. В этой же процедуре определяется вириальный вклад `virial` уравнения (19), используемого для расчета внутреннего давления системы.

Ниже приведены программные коды 2.2 и 2.3, позволяющие рас-

считать в табличном виде (`table_pot`) энергию и силу через потенциалы Леннард-Джонса и Джугутова.

Код 2.2

```
public void load_LJ_forces()
{
    float sigma = 1.0f, epsilon = 1.0f, displ = 0.0001f;
    double rij = 0.5f, energy, force, energy_rcut;
    r_cut = 2.5f;
    table_size = 25000;
    table_pot = new float[table_size, 2];
    energy_rcut = 4.0 * epsilon * (Math.Pow((sigma / r_cut), 12) -
    Math.Pow((sigma / r_cut), 6));
    for (int i = 0; i < table_size; i++)
    {
        rij = rij + displ;
        energy = 4.0 * epsilon * (Math.Pow((sigma / rij), 12) -
        Math.Pow((sigma / rij), 6)) - energy_rcut;
        force = 24.0 * epsilon * (2.0 * Math.Pow((sigma / rij), 13) -
        Math.Pow((sigma / rij), 7)) / sigma;
        table_pot[i, 0] = (float)energy;
        table_pot[i, 1] = (float)force;
    }
}
```

Код 2.3

```
public void load_Dz_forces()
{
    float A = 5.82f, D = 1.28f, a = 1.87f, c = 1.1f, d = 0.27f, m = 16,
    sigma = 1.0f, epsilon = 1.0f;
    double rij = 0.5f, V1 = 0.0, V2 = 0.0, energy = 0.0, F1 = 0.0, F2 =
    0.0, force = 0.0;
    float displ = 0.0001f;
    r_cut = 1.94f;
    table_size = 15000;
    table_pot = new float[table_size, 2];
    for (int i = 0; i < table_size; i++)
    {
```

```
rij = rij + displ;
if (rij / sigma < a)
{
    V1 = A * epsilon * (Math.Pow((sigma / rij), m) - D) *
Math.Exp(sigma * c / (rij - sigma * a));
    F1 = A * epsilon * (-(m / sigma) * Math.Pow((sigma / rij), (m +
1)) - (Math.Pow((sigma / rij), (m)) - D) * sigma * c * Math.Pow((rij -
sigma * a), (-2.0))) * Math.Exp(sigma * c / (rij - sigma * a));
}
else { V1 = 0.0; }
if (rij / sigma < r_cut)
{
    V2 = D * epsilon * Math.Exp(sigma * r_cut / (rij - sigma *
r_cut));
    F2 = -epsilon * sigma * d * D * Math.Exp(sigma * r_cut / (rij -
sigma * r_cut)) * Math.Pow((rij - sigma * r_cut), -2.0);
}
else { V2 = 0.0; }
energy = V1 + V2;
force = -1.0 * (F1 + F2);
table_pot[i, 0] = (float)energy;
table_pot[i, 1] = (float)force;
}
}
```

Глава 3. Обработка результатов моделирования

§3.1 Сохранение и загрузка конфигурации системы

В моделировании, как правило, требуется сохранять конфигурации системы в различные моменты времени с целью их использования в последующих расчетах. Ниже приведена процедура в виде программного кода 3.1, которая позволяет сохранять текущую конфигурацию системы в файл с именем «system_config.txt», с указанием числа частиц, размера системы и времени моделирования. В файл также записываются координаты и скорости всех частиц. В программном коде 3.2 представлена процедура чтения из файла с называнием «system_config.txt» ранее сохраненной конфигурации системы. Процедура `save_config`, как правило, запускается после процедуры `system_dynamics(2)`. Процедура `load_save_config` запускается после процедуры инициализации начальных условий.

Код 3.1

```
public void save_config()
{
    // создается файл для записи конфигурации системы
    FileStream fss = new FileStream("system_config.txt", FileMode.Create);
    StreamWriter sww = new StreamWriter(fss);
    // сохраняются параметры системы
    sww.WriteLine(L_x.ToString());
    sww.WriteLine(L_y.ToString());
    sww.WriteLine(L_z.ToString());
    sww.WriteLine(N.ToString());
    sww.WriteLine(time.ToString());
    for (int i = 0; i < N; i++)
```

```

{
    // сохраняются координаты частиц системы
    sww.WriteLine(pos_x[i].ToString() + " " + pos_y[i].ToString() + " " +
pos_z[i].ToString());
    // сохраняются скорости движения частиц
    sww.WriteLine(vel_x[i].ToString() + " " + vel_y[i].ToString() + " " +
vel_z[i].ToString());
}
sww.Close(); fss.Close();
}

```

Код 3.2

```

public void load_save_config()
{
    string st, sts;
    int idx;
    FileStream fss = new FileStream("system_config.txt", FileMode.Open);
    StreamReader str = new StreamReader(fss);
    //читываются параметры системы
    st = str.ReadLine(); L_x = Convert.ToSingle(st);
    st = str.ReadLine(); L_y = Convert.ToSingle(st);
    st = str.ReadLine(); L_z = Convert.ToSingle(st);
    st = str.ReadLine(); N = Convert.ToInt32(st);
    st = str.ReadLine(); time = Convert.ToSingle(st);
    pos_x = new float[N];
    pos_y = new float[N];
    pos_z = new float[N];
    vel_x = new float[N];
    vel_y = new float[N];
    vel_z = new float[N];
    acc_x = new float[N];
    acc_y = new float[N];
    acc_z = new float[N];
    for (int i = 0; i < N; i++)
    {
        // чтение координат частиц
        sts = str.ReadLine();
        idx = sts.IndexOf(" ");
        pos_x[i] = Convert.ToSingle(sts.Substring(0, idx));
        st = sts.Remove(0, idx + 1); idx = st.IndexOf(" ");

```

```

    pos_y[i] = Convert.ToSingle(st.Substring(0, idx));
    sts = st.Remove(0, idx + 1);
    pos_z[i] = Convert.ToSingle(sts);
    // чтение скорости движения частиц
    sts = str.ReadLine();
    idx = sts.IndexOf(" ");
    vel_x[i] = Convert.ToSingle(sts.Substring(0, idx));
    st = sts.Remove(0, idx + 1); idx = st.IndexOf(" ");
    vel_y[i] = Convert.ToSingle(st.Substring(0, idx));
    sts = st.Remove(0, idx + 1);
    vel_z[i] = Convert.ToSingle(sts);
}
str.Close(); fss.Close();
}

```

Кроме конфигурации системы иногда требуется сохранять мгновенные значения характеристик системы: например, текущую температуру, внутреннее давление, кинетическую, потенциальную и полную энергии системы. Для этого можно воспользоваться программным кодом 3.3 процедуры `save_system_parameters`, которая, как правило, запускается после процедуры `system_dynamics(2)`. Следует отметить, что указанная процедура `save_system_parameters` добавляет данные в ранее созданный файл. Поэтому перед новыми МД-расчетами требуется удаление предыдущих файлов. Для активации возможности чтения из файла и записи в файл информации необходимо подключить пакет `using System.IO`.

Код 3.3

```

public void save_system_parameters()
{
    // сохранение информации о температуре системы
    FileStream fs_T = new FileStream("system_temperature.txt",
    FileMode.Append);
    StreamWriter sw_T = new StreamWriter(fs_T);
    st = time.ToString() + " " + temperature.ToString();
    st = st.Replace(",", ".");
    sw_T.WriteLine(st);
    sw_T.Close(); fs_T.Close();
    // сохранение информации о внутреннем давлении системы
}

```

```
FileStream fs_P = new FileStream("system_pressure.txt",
 FileMode.Append);
StreamWriter sw_P = new StreamWriter(fs_P);
st = time.ToString() + " " + pressure.ToString();
st = st.Replace(",", ".");
sw_P.WriteLine(st);
sw_P.Close(); fs_P.Close();
// сохранение информации о кинетической энергии системы
FileStream fs_Ek = new FileStream("system_kinetic_energy.txt",
 FileMode.Append);
StreamWriter sw_Ek = new StreamWriter(fs_Ek);
st = time.ToString() + " " + (kinetic_energy * N).ToString();
st = st.Replace(",", ".");
sw_Ek.WriteLine(st);
sw_Ek.Close(); fs_Ek.Close();
// сохранение информации о потенциальной энергии системы
FileStream fs_Ep = new FileStream("system_potential_energy.txt",
 FileMode.Append);
StreamWriter sw_Ep = new StreamWriter(fs_Ep);
st = time.ToString() + " " + potential_energy.ToString();
st = st.Replace(",", ".");
sw_Ep.WriteLine(st);
sw_Ep.Close(); fs_Ep.Close();
// сохранение информации о полной энергии системы
FileStream fs_H = new FileStream("system_total_energy.txt",
 FileMode.Append);
StreamWriter sw_H = new StreamWriter(fs_H);
st = time.ToString() + " " + H_energy.ToString();
st = st.Replace(",", ".");
sw_H.WriteLine(st);
sw_H.Close(); fs_H.Close();
}
```

§3.2 Визуализация результатов

При обработке и анализе результатов компьютерного моделирования часто возникает необходимость визуализации конфигурации системы. Это можно сделать с помощью программы VMD (Visual Molecular Dynamics), разработанной американским университетом штата Иллиноис (University of Illinois at Urbana-Champaign) [33]. Данная программа позволяет визуализировать и анализировать физиче-

ские, химические, биологические системы, многокомпонентные атомарные и молекулярные системы, определить их структуру и т. д. Программа VMD отображает результаты компьютерного моделирования в виде трехмерной графики и анимаций. В качестве примера на рисунке 11 представлен внешний интерфейс программы VMD версии 1.9.

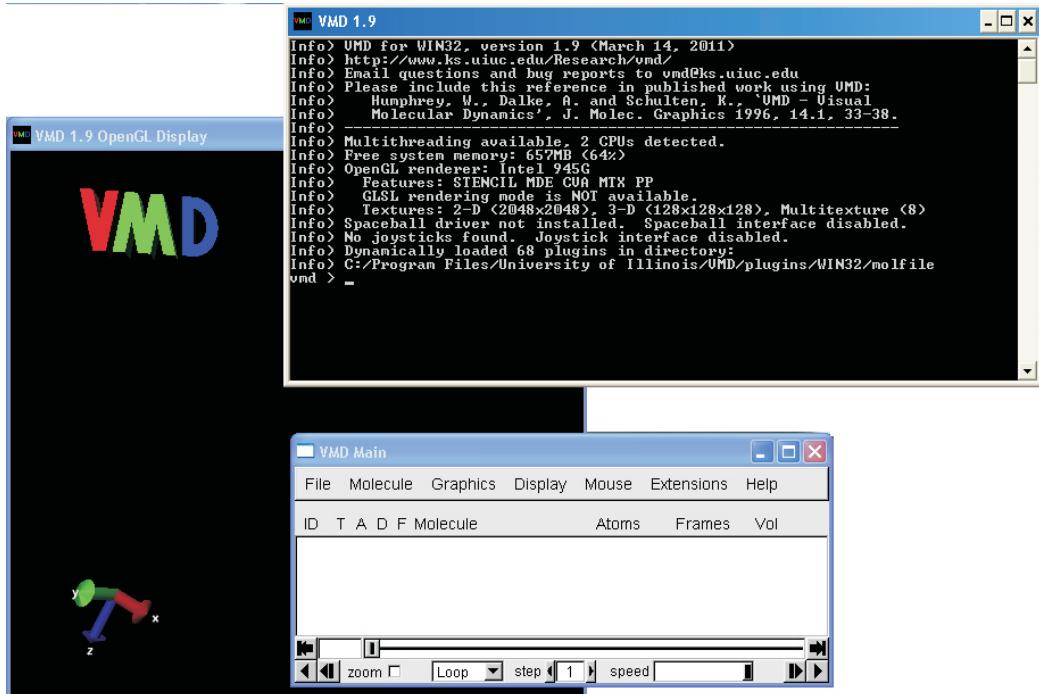


Рис. 11: Внешний интерфейс программы VMD 1.9.

Для визуализации результатов МД-моделирования в главном окне программы VMD выбирается пункт «New Molecule» (см. рисунок 12) и открывается диалоговое окно, где указывается файл с информацией о конфигурации моделируемой системы (см. рисунок 13). В меню «Determine file type» указывается тип загружаемого файла. Тип файла связан с известными вычислительными и расчетными пакетами, например, AMBER (программный пакет для моделирования биомолекул), CHARMM (пакет для моделирования биологических систем: протеинов, пептидов, липидов и других многочастичных систем), VASP (программный пакет для квантово-механического *ab initio* моделирования), LAMMPS (программа для моделирования атомарной и молекулярной динамики) и т. д.

В случае, если конфигурационный файл получен с помощью пакета LAMMPS, то в открывающемся меню «Determine file type» необходимо выбрать пункт «LAMMPS Trajectory» (см. рисунок 14). В общем случае, для визуализации с помощью VMD требуется файл,

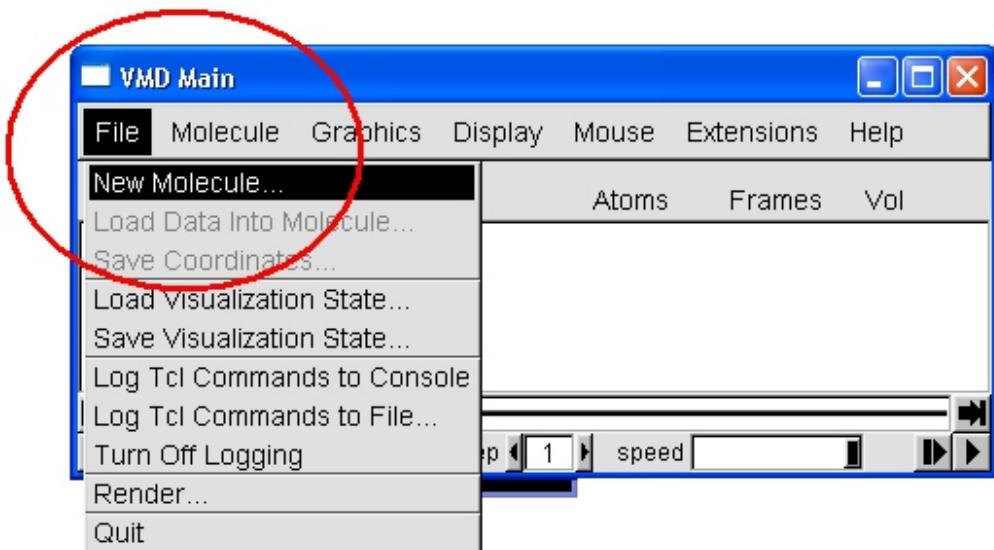


Рис. 12: Выбор пункта «New Molecule» в главном диалоговом окне программы VMD 1.9.

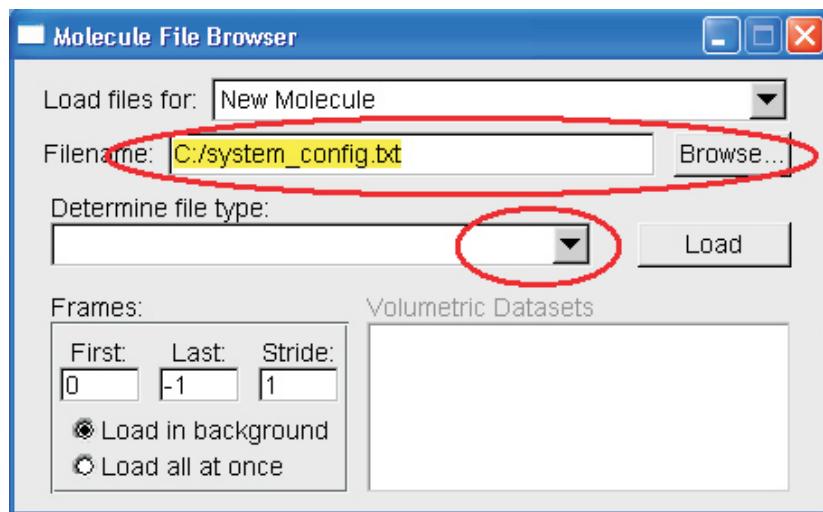


Рис. 13: Вид диалогового окна, где задаются детали загрузки конфигурации системы.

содержащий координаты всех частиц системы.

После загрузки конфигурации можно изменить настройки графического отображения результатов МД-моделирования (см. рисунки 15 и 16). Результат визуализации с помощью программы VMD, где частицы отображаются в виде шаров, представлен на рисунке 17.

Ниже приведен программный код 3.4, который можно использовать для визуализации результатов МД-моделирования с помощью программы VMD, выбрав пункт «LAMMPS Trajectory». Этот программный код используется вместо ранее приведенного кода 3.1.

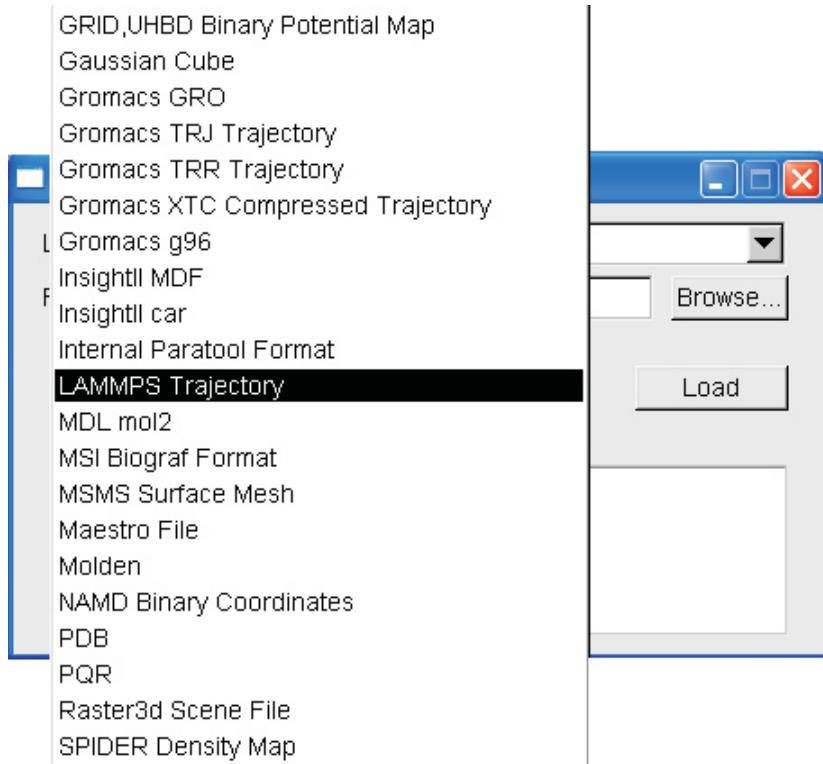


Рис. 14: Вид всплывающего меню, где указывается тип загружаемой конфигурации системы.

Код 3.4

```
public void load_lammps_trajectory()
{
    string stx, sty, stz, st;
    float bx, by, bz;
    // указывается адрес и открывается файл для записи
    FileStream fs = new FileStream("system_config.txt", FileMode.Append);
    StreamWriter wfs = new StreamWriter(fs);
    // записываются параметры для чтения через программу VMD
    wfs.WriteLine("ITEM: Timestep");
    wfs.WriteLine(step);
    wfs.WriteLine("ITEM: Number of Atoms");
    wfs.WriteLine(N);
    wfs.WriteLine("ITEM: Box Bounds pp pp pp");
    bx = L_x / 2.0f;
    by = L_y / 2.0f;
    bz = L_z / 2.0f;
    stx=(-bx).ToString() +" "+ bx.ToString(); stx=stx.Replace(",",".");
    sty=(-by).ToString() +" "+ by.ToString(); sty=sty.Replace(",",".");
    stz=(-bz).ToString() +" "+ bz.ToString(); stz=stz.Replace(",","");
    wfs.WriteLine(stx);
    wfs.WriteLine(sty);
```

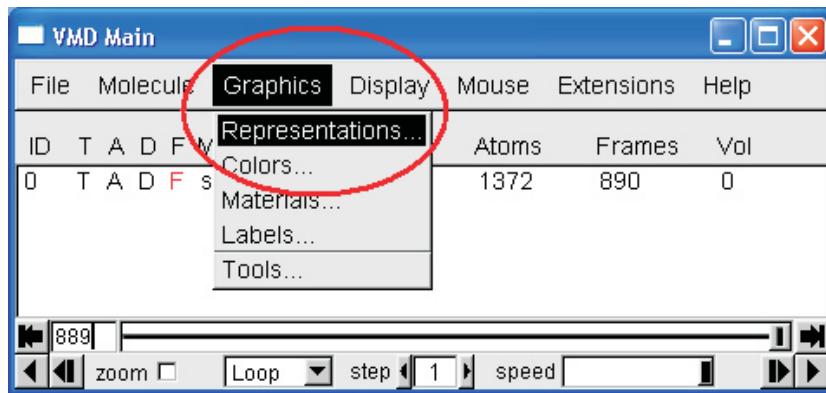


Рис. 15: Изменение параметров отображения конфигурации.

```
wfs.WriteLine(stz);
// записываются координаты частиц
wfs.WriteLine("ITEM: ATOMS id type x y z");
for (int i = 0; i < N; i++)
{
    st = (i + 1).ToString() + " 0 "+ pos_x[i].ToString() + " "+
pos_y[i].ToString() + "+" pos_z[i].ToString();
    st = st.Replace(",",".");
    wfs.WriteLine(st);
}
wfs.Close(); fs.Close();
}
```

Процедура `load_lammps_trajectory` встраивается после процедуры расчета координат частиц системы. Для активации возможности чтения из файла и записи в файл данных с помощью процедуры `load_lammps_trajectory` необходимо подключить пакет `using System.IO`.

§3.3 Построение графиков в программном пакете Матлаб

Представление результатов МД-моделирования графически (функции радиального распределения частиц, временные зависимости термодинамических параметров системы) удобно осуществлять с помощью пакета прикладных программ Матлаб (MATLAB – сокр. «Matrix Laboratory»). Пакет Матлаб разработан американской компанией MathWorks, специализирующейся на создании программного

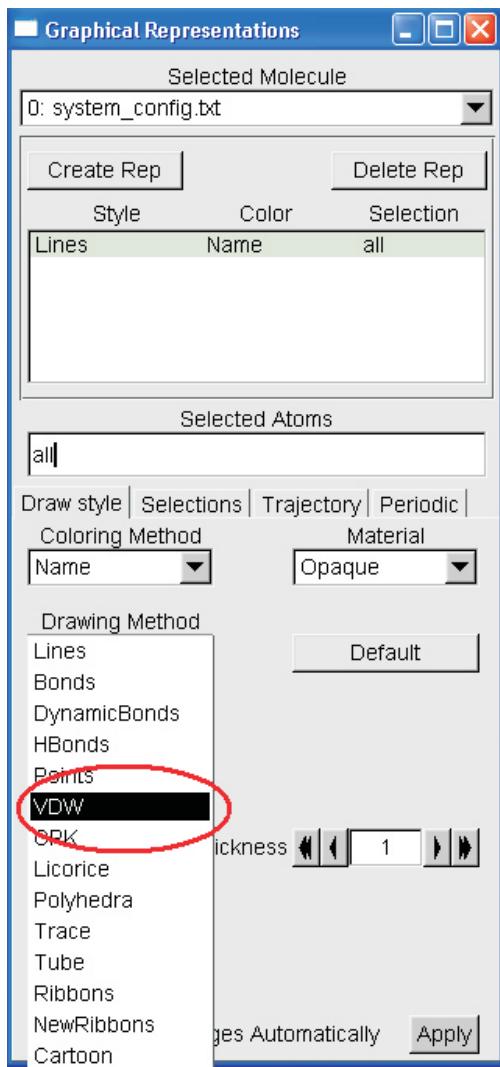


Рис. 16: Всплывающее окно, где можно настроить параметры отображения конфигурации.

обеспечения в области численных вычислений и компьютерного моделирования. Изначально пакет Матлаб был предназначен для проектирования систем управления, а также для работы в научных и инженерных областях. Матлаб использует оригинальный язык программирования, что позволяет применять этот пакет для выполнения сложных вычислительных операций. Внешний интерфейс пакета Матлаб представлен на рисунке 18. Рассмотрение всех возможностей пакета Матлаб не является целью настоящего учебного пособия, поэтому основное внимание будет уделено рассмотрению методов графического представления данных моделирования.

Рассмотрим один из способов, позволяющих выполнить чтение данных моделирования из текстового файла. Для того, чтобы написать соответствующую процедуру необходимо открыть окно Blank M-file, используемый для написания программ на оригинальном языке.

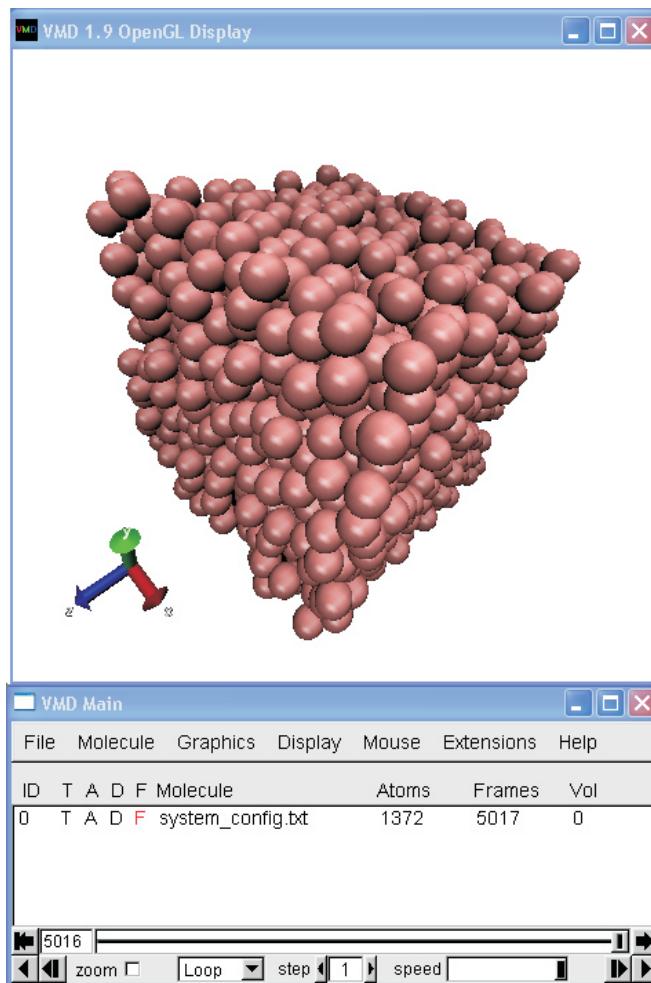


Рис. 17: Результат визуализации с помощью программы VMD, где показана конфигурация системы, состоящей из 1372 частиц, изображенных в виде шаров.

ке Матлаб. Для открытия файла Blank M-file необходимо запустить пакет Матлаб и нажать комбинацию клавиш **Ctrl+N**. Внешний вид диалогового окна бланка M-file представлен на рисунке 19.

Для считывания данных из файла можно использовать функцию `dlmread()`, которая задается в виде `f = dlmread(adr, frm);`, где `f` – массив данных, `adr` – адрес расположения файла, откуда считаются данные, `frm` – формат чтения данных из файла. Например, представим, что файл с названием `adr = 'MD_data.txt'` содержит следующие данные (в виде двух столбцов):

```

1 0.543
2 0.678
3 0.798
4 0.991
5 1.245

```

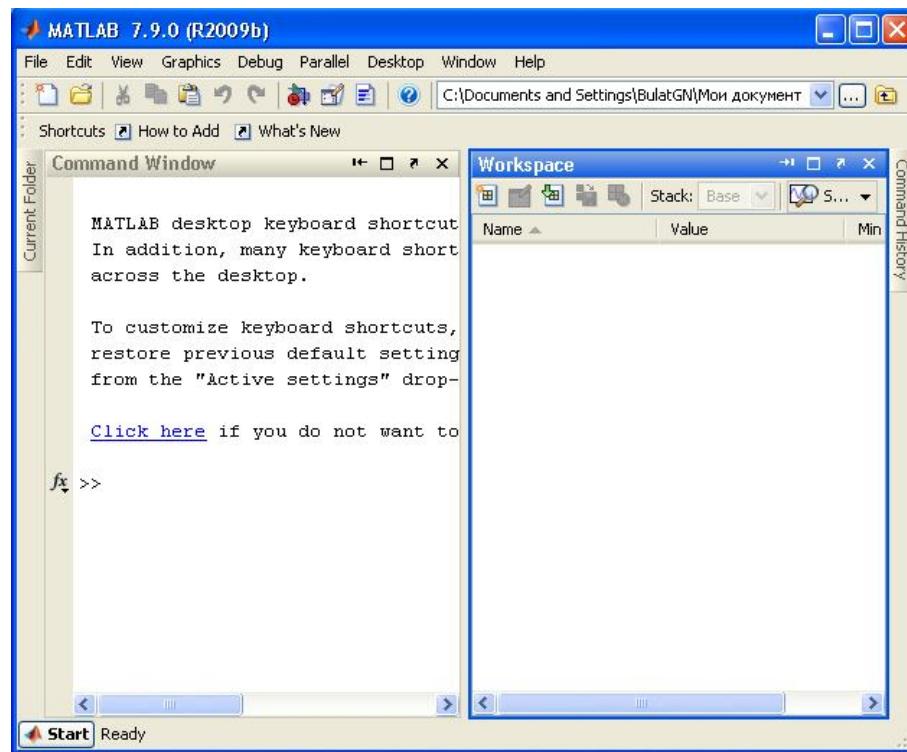


Рис. 18: Интерфейс программы Матлаб версии 7.9.0 (2009). На рисунке изображено главное окно программы, где видны панели инструментов, командное окно и рабочее пространство.

6 2.581

7 6.252

Для чтения данных из файла в бланке M-file необходимо выполнить следующую процедуру:

```
adr = 'MD_data.txt';
frm = ' ';
f = dlmread(adr, frm);
```

В итоге все данные из файла будут загружены в массив f:

```
» f =
   f =
      Column 1        Column 2
      1.0000        0.5430
      2.0000        0.6780
      3.0000        0.7980
      4.0000        0.9910
      5.0000        1.2450
      6.0000        2.5810
      7.0000        6.2520
```

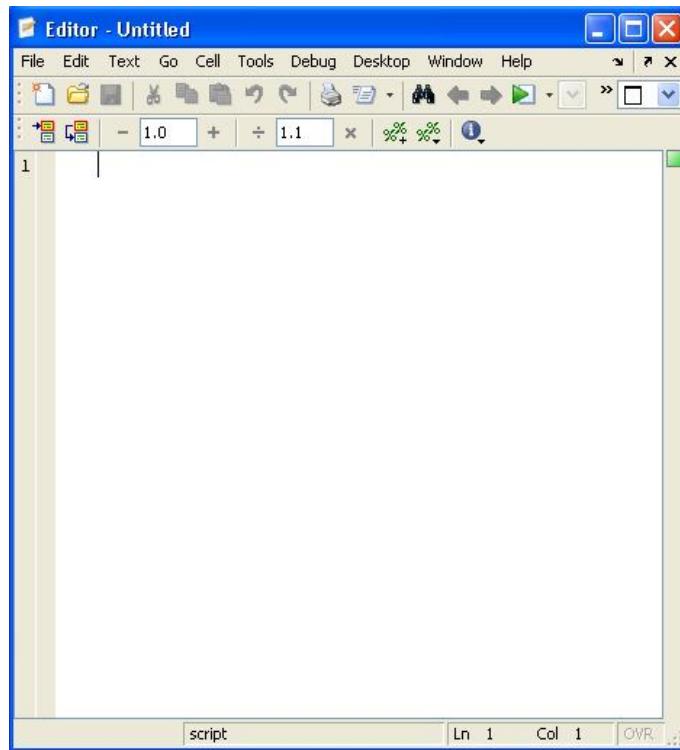


Рис. 19: Вид диалогового окна Blank M-file (открывается нажатием клавиш Ctrl+N), где задаются алгоритмы, выполняемые программой Матлаб.

Для графического отображения данных массива f можно воспользоваться функцией `plot()`:

```
sty = 'r-';
plot(f(:,1),f(:,2),sty);
```

Здесь при $f(:,1)$ данныечитываются из всех строк первого столбца, при $f(:,2)$ – из второго столбца. Величина `sty` задает параметры графического отображения данных. Важно отметить, что для корректного чтения данных из файла дробные числа должны быть указаны через точку, а не через запятую. Данные разных столбцов отделяются одним пробелом. Вид бланка M-file, используемого для чтения данных из файла «MD_data.txt», и результат работы программы представлены на рисунке 20. Для компиляции и запуска программы M-file необходимо нажать кнопку Run.

С помощью представленной на рисунке 20а программы можно достаточно быстро выполнить обработку и анализ результатов мо-

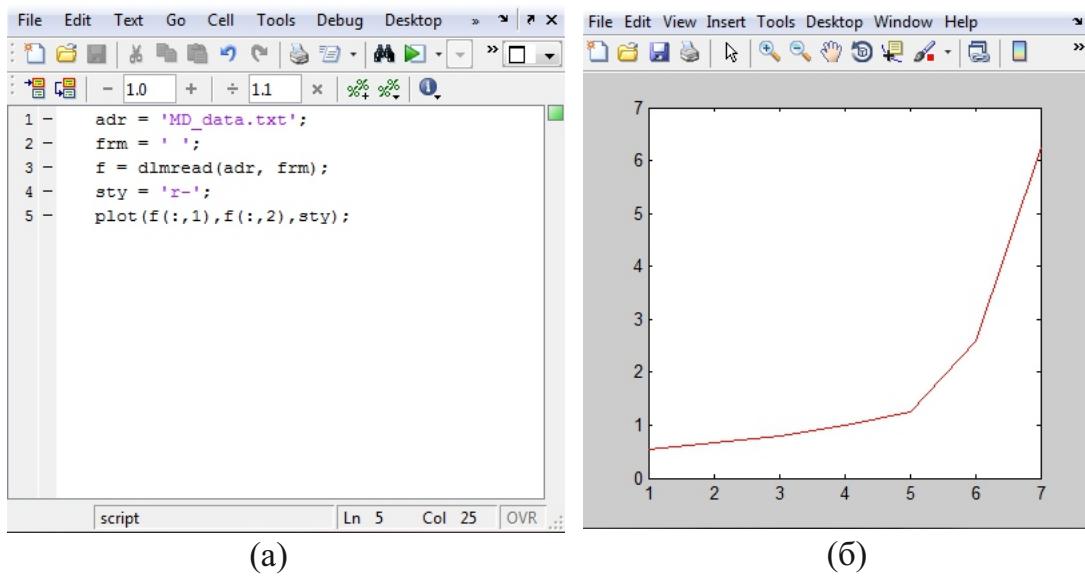


Рис. 20: (а) – вид бланка M-файла программы Матлаб, используемого для чтения данных из файла; (б) – результат компиляции и работы программы M-file.

делирования, в том числе, при решении задач, представленных в настоящем учебном пособии.

Глава 4. Моделирование молекулярной динамики твердых тел и жидкостей

§4.1 Этапы моделирования

Наиболее распространенный алгоритм реализации моделирования молекулярной динамики можно представить в виде следующих этапов [2, 4]:

1. Задаются начальная конфигурация и параметры системы (количество частиц, размеры моделируемой системы, количество временных шагов, потенциал межчастичного взаимодействия и т. д.);
2. Применяются подходящие граничные условия;
3. Интегрирование уравнений движения на временном шаге $t + \Delta t/2$, например, через скоростной алгоритм Верле. Определяются координаты и скорости движения частиц;
4. Определяются потенциальная энергия и силы межчастичного взаимодействия с помощью заданного потенциала;
5. На основе термостата и баростата (в зависимости от ансамбля) корректируются температура и давление системы;
6. Интегрирование уравнений движения на временном шаге $t + \Delta t$ (например, с помощью того же скоростного алгоритма Верле). Определяются скорости движения частиц;
7. Вычисление термодинамических и кинетических параметров, характеризующих состояние системы (например, температура, давление, потенциальная и кинетическая энергии);

8. Полученная конфигурация системы сохраняется и моделирование повторяется заданное число M раз, начиная со 2-го этапа. Моделирование может прерваться в любой момент в зависимости от заданных условий и решаемой задачи;
9. Статистическое усреднение и сохранение результатов моделирования.

Представленные выше этапы моделирования являются общими и применимыми практически во всех равновесных и неравновесных молекулярно-динамических расчетах. Однако в зависимости от условий моделирования и решаемой задачи этот алгоритм может быть упрощен или дополнен другими расчетными этапами. Реализация этого алгоритма представлена в программном коде 4.1. Блок-схема основных этапов моделирования молекулярной динамики приведена на рисунке 21.

Для корректной работы программы необходимо подключить следующие пакеты:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
```

Код 4.1

```
static void Main(string[] args)
{
    Program p = new Program();
    p.MD();
    Console.WriteLine("Расчет завершен");
    Console.ReadLine();
}

# region Глобальные переменные
bool load_syst, ready, testid, move_test, update_test = true;
string cryst_type;
Random rand = new Random();
int N, nx, ny, nz, init_type, M, step, table_size, nb, iter, ri;
int max_list_size, list_size, J;
int[] Advance, Marker1, Marker2, List;
```

```

float L_x, L_y, L_z, density, volume, alat, dt, r_cut, dr_s, time;
float rij2, dP, dF, ds, ds1, ds2;
float lambda, xi, kinetic_energy, T_ext, P_ext, temperature, S_f = 0, dd,
rv_x, rv_y, rv_z;
float Q_T, Q_B, pressure, virial, err;
float p_o, p_s, H_energy, potential_energy, rij_x, rij_y, rij_z, T0, T0,
vel_dT;
float[] pos_x, pos_y, pos_z, vel_x, vel_y, vel_z, acc_x, acc_y, acc_z,
e_pot;
float[] disp_list_x, disp_list_y, disp_list_z, system_center;
float[] v_x, v_y, v_z, v_xo, v_yo, v_zo, b_x, b_y, b_z;
float[,] r_cell, table_pot;
#endif
public void MD()
{
    load_initial_stat(); // задаются детали моделирования
    load_LJ_forces(); // задается потенциал межчастичного взаимодействия
    for (step = 1; step <= M; step++)
    {
        time = time + dt; // реальное время моделирования
        system_dynamics(1); // интегрирование уравнений движения на временном
шаге  $t + \Delta t/2$ 
        boundary_conditions(); // периодические граничные условия
        load_forces(); // расчет потенциальной энергии, силы и ускорения
частиц
        system_dynamics(2); // интегрирование уравнений движения на временном
шаге  $t + \Delta t$ 
        load_lammps_trajectory(); // запись полученной конфигурации в файл
        // на экран выводится информация о ходе выполнения расчетов
        Console.WriteLine("time = " + time.ToString() + " steps/M = "+
step.ToString() + "/" + M.ToString() + " T " + temperature.ToString() + " P
" + pressure.ToString());
    }
}

```

Здесь M – есть общее число МД-итераций (количество временных шагов), переменная `time` определяет реальное время моделирования.

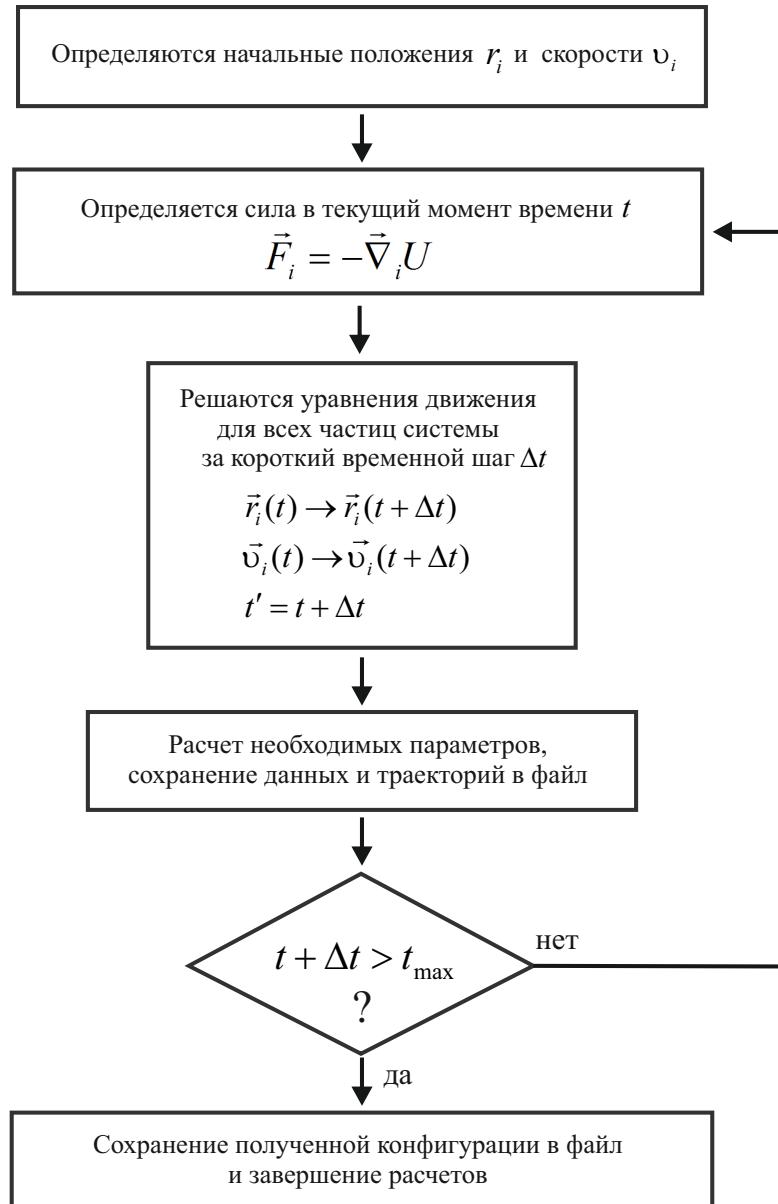


Рис. 21: Блок-схема реализации основных этапов МД-моделирования.

§4.2 Начальные условия. Исходная конфигурации системы

При моделировании молекулярной динамики какой-либо конкретной системы правильная постановка начальных условий является крайне важной [1, 2, 4, 17], поскольку от этого зависит корректность последующих МД-расчетов и получаемых результатов. Начальные условия моделирования задаются исходя из решаемой задачи, а также в зависимости от типа системы. Основными моментами, которые должны быть заданы на начальном этапе (до непосредственных расчетов), являются следующие:

-
1. Тип системы;
 2. Потенциал межчастичного взаимодействия. Параметры потенциала;
 3. Количество частиц (атомов, молекул, ионов);
 4. Размер ячейки моделирования. Количественная плотность системы;
 5. Начальная конфигурация системы;
 6. Статистический ансамбль (термостат и баростат);
 7. Температура системы. Внешнее давление. Параметры термостата и баростата;
 8. Уравнения движения. Временной шаг;
 9. Время моделирования или общее число временных шагов;
 10. Параметры дополнительных расчетных этапов;
 11. Адрес файла с выходными данными (при необходимости).

Рассмотрим каждый из пунктов более подробно.

1. *Тип системы.* На начальном этапе необходимо определиться с типом системы. Это означает, что необходимо выбрать модель некоторой реальной системы, физические свойства и структурные характеристики которой будут исследоваться. Модель задается соответствующим потенциалом межчастичного взаимодействия. Выбор системы зависит от решаемой задачи и конкретного интересуемого физического процесса.
2. Важным этапом является выбор *вида потенциала межчастичного взаимодействия*. Например межчастичное взаимодействие в инертных газах хорошо воспроизводится потенциалом Леннард-Джонса. Для металлических систем, например, состоящих из атомов алюминия, меди, приемлемыми являются потенциалы семейства ЕАМ. Межатомное взаимодействие кремния, германия воспроизводится трехчастичным потенциалом Стиллинжера-Вебера. Молекулярная динамика воды воспроизводится различными атомистическими модельными потенциалами SPC/E, TIP4P, или модифицированными эффект-

тивными крупнозернистыми модельными потенциалами. Существует большое разнообразие потенциалов межчастичного взаимодействия. Поэтому при моделировании той или иной системы следует заранее изучить соответствующую научную литературу по теме исследования, ознакомиться со свойствами исследуемого материала и фазовой диаграммой.

3. Задается *количество частиц* N (атомов или молекул). Величина N , в первую очередь, задается с учетом решаемой задачи, а также в зависимости от имеющихся в наличии вычислительных ресурсов. Для получения более корректных результатов, где усреднение выполняется по ансамблю частиц (например, при расчете радиальной функции распределения частиц, при структурном анализе) величина N должна быть достаточно большой. При моделировании фазовых переходов количество частиц должно быть порядка и более тысячи и десятка тысяч частиц. Если при моделировании не планируется выполнение статистической обработки, где рассчитываются только удельные термодинамические и кинетические параметры системы (например, потенциальная и кинетическая энергии, скорость движения одной частицы), то моделирование можно выполнить для небольшой системы ($N < 1000$). В случае, если конфигурация системы задается в виде кристаллической решетки, количество частиц $N = l_c \cdot n_x \cdot n_y \cdot n_z$ определяется с учетом размера элементарной ячейки l_c , а также количеством частиц n_x, n_y и n_z , располагающихся по направлению осей координатной системы. Здесь, например, для кристаллической ГЦК и ГПУ решетки $l_c = 4$, для простой кубической решетки $l_c = 1$.
4. После того, как указано количество частиц N задается *размер ячейки моделирования*. Например, в случае прямоугольной системы задаются длины ребер ячейки моделирования L_x, L_y и L_z . Определяется объем $V = L_x \cdot L_y \cdot L_z$ и количественная плотность $\rho = N/V$ системы. Для того, чтобы задать необходимую количественную плотность ρ достаточно подобрать соответствующие значения параметров L_x, L_y и L_z . В случае, если конфигурация системы задается в виде кристаллической решетки, длина ребер ячейки определяется с учетом постоянной решетки a_c и числа частиц, располагающихся по осям системы координат. Например, если по направлению осей X, Y и Z коли-

чество частиц составляет n_x , n_y и n_z , то длина сторон ячейки равна $L_x = a_c \cdot n_x$, $L_y = a_c \cdot n_y$ и $L_z = a_c \cdot n_z$.

5. *Начальная конфигурация системы* (т.е. начальные положения и скорости движения частиц) задается исходя из количества частиц и размера ячейки моделирования. Преимущественно используется два способа генерации начальных положений частиц внутри ячейки: 1) в узлах кристаллической решетки определенной структуры (например, ГЦК – гранецентрированной кубической, ГПУ – гексагональной плотно упакованной, ПК – простой кубической) (см. рисунок 22); 2) с помощью генератора случайных чисел [1, 4, 17]. Далее полученная конфигурация системы приводится в термодинамическое равновесие требуемой температурой и давлением. При генерации координат частиц случайнym образом частицы системы равномерно и хаотично распределяются по всему объему. При этом необходимо, чтобы не было взаимного перекрытия частицами, что может вызвать резкий скачок температуры и давления системы.
6. Выбор *статистического ансамбля*. Для моделирования свойств систем, находящихся в состоянии термодинамического равновесия, используется микроканонический ансамбль NVE. Для моделирования системы при заданной температуре можно использовать канонический NVT-ансамбль. В случае, когда помимо температуры задается постоянное внешнее давление можно использовать изобарно-изотермический NPT-ансамбль. Для контроля температуры и давления необходимо использовать термостат и баростат, соответственно. В случае, когда система приводится в состояние термодинамического равновесия, удобно применять метод масштабирования скоростей. При моделировании физических процессов (например, фазовых переходов, явлений переноса), наиболее приближенных к реальным условиям, можно использовать термостат и баростат Нозе-Гувера.
7. Задается *температура системы и внешнее давление* (при необходимости), определяемые из условий решаемой задачи. Корректируются параметры термостата и баростата.
8. *Уравнения движения и алгоритм их интегрирования*. Преимущественно используются классические уравнения движения Ньютона, которые приведены в первой главе. Как правило, ин-

тегрирование уравнений движения выполняется с помощью скользящего алгоритма Верле. Задается временной шаг Δt МД-моделирования.

9. На основе оцениваемого времени «эксперимента» определяется количество временных шагов M (МД-итераций). Например, если необходимо выполнить моделирование продолжительностью $t = 20$ нс (наносекунд) с временным шагом $\Delta t = 0.001$ нс, тогда общее число временных шагов составит $M = t/\Delta t = 20000$.
10. На следующем этапе задаются такие детали моделирования дополнительных расчетных этапов, как параметры расчета структурных и динамических характеристик системы.
11. В завершении указываются названия выходных файлов и адрес их расположения (если таковые имеются).

Представленный выше алгоритм реализации начальных условий в зависимости от решаемой задачи и исследуемого физического процесса может быть упрощен или дополнен другими этапами.

Ниже приведен программный код 4.2 процедуры, используемой для создания исходной конфигурации системы. Здесь частицы могут быть заданы в узлах кристаллической решетки или случайным образом внутри ячейки моделирования.

Код 4.2

```
public void load_initial_stat()
{
    # region Детали моделирования
    N = 1372; // количество частиц (задается при init_type = 2)
    M = 10000; // количество временных шагов
    alat = 1.75f; // постоянная решетки
    nx = 7; ny = 7; nz = 7; // количество частиц (задается при init_type =
    1)
    dt = 0.001f; // временной шаг
    T_ext = 2.5f; // температура термостата
    T0 = 0; // начальная температура системы
    P_ext = 1.5f; // давление баростата
    Q_T = 100.0f; // параметр термостата Нозе-Гувера
    Q_B = 500.0f; // параметр баростата Нозе-Гувера
```

```
vel_dT = 0.002f; // скорость нагревания или охлаждения системы
dr_s = 0.3f; // параметр потенциала
init_type = 1; // вид загрузки начальной конфигурации системы
cryst_type = "гцк"; // тип кристаллической решетки: гцк, пк
#endifregion
// размер прямоугольной ячейки моделирования
L_x = alat * nx;
L_y = alat * ny;
L_z = alat * nz;
time = 0;
if (init_type == 1) gen_order_stat(cryst_type); // конфигурация
генерируется в виде кристаллической решетки
if (init_type == 2) gen_random_stat(); // координаты частиц задаются с
помощью генератора случайных чисел
#region Подготовка массивов
vel_x = new float[N];
vel_y = new float[N];
vel_z = new float[N];
acc_x = new float[N];
acc_y = new float[N];
acc_z = new float[N];
v_x = new float[N];
v_y = new float[N];
v_z = new float[N];
v_xo = new float[N];
v_yo = new float[N];
v_zo = new float[N];
b_x = new float[N];
b_y = new float[N];
b_z = new float[N];
disp_list_x = new float[N];
disp_list_y = new float[N];
disp_list_z = new float[N];
e_pot = new float[N];
Advance = new int[N];
Marker1 = new int[N];
Marker2 = new int[N];
List = new int[100 * N];
#endregion
// файл с выходными данными
FileStream fs = new FileStream("system_config.txt", FileMode.Create);
fs.Close();
}
```

```

// координаты частиц задаются в узлах кристаллической решетки
public void gen_order_stat(string ctype)
{
    if (ctype == "гцк")
    {
        // параметры элементарной ячейки гцк
        nb = 4;
        r_cell = new float[3, nb];
        r_cell[0, 0] = 0; r_cell[1, 0] = 0; r_cell[2, 0] = 0;
        r_cell[0, 1] = 0.5f; r_cell[1, 1] = 0.5f; r_cell[2, 1] = 0;
        r_cell[0, 2] = 0; r_cell[1, 2] = 0.5f; r_cell[2, 2] = 0.5f;
        r_cell[0, 3] = 0.5f; r_cell[1, 3] = 0; r_cell[2, 3] = 0.5f;
        N = 4 * nx * ny * nz;
        pos_x = new float[N];
        pos_y = new float[N];
        pos_z = new float[N];
    }
    if (ctype == "пк")
    {
        // параметры элементарной ячейки пк
        nb = 1;
        r_cell = new float[3, nb];
        r_cell[0, 0] = 0; r_cell[1, 0] = 0; r_cell[2, 0] = 0;
        N = nx * ny * nz;
        pos_x = new float[N];
        pos_y = new float[N];
        pos_z = new float[N];
    }
    int n = 0;
    // генерируется конфигурация системы в виде кристаллической решетки
    for (int k = 0; k < nz; k++)
    {
        for (int j = 0; j < ny; j++)
        {
            for (int i = 0; i < nx; i++)
            {
                for (int L = 0; L < nb; L++)
                {
                    // задаются координаты каждой частицы
                    n++;
                    pos_x[n - 1] = alat * (i + r_cell[0, L]);
                    pos_y[n - 1] = alat * (j + r_cell[1, L]);
                    pos_z[n - 1] = alat * (k + r_cell[2, L]);
                }
            }
        }
    }
}

```

```
        }
    }
}
}

load_system_center();

}

// координаты частиц задаются с помощью генератора случайных чисел
public void gen_random_stat()
{
    // задаются параметры генерации координат частиц
    float r2, rx, ry, rz, min_dist, sigma = 1.0f;
    load_syst = true;
    int n = 0;
    pos_x = new float[N];
    pos_y = new float[N];
    pos_z = new float[N];
    volume = L_x * L_y * L_z;
    density = N / volume;
    min_dist = 1.1f * sigma;
    while (load_syst)
    {
        // генерируются координаты частиц (случайным образом)
        pos_x[n] = (float)(rand.NextDouble() * L_x);
        pos_y[n] = (float)(rand.NextDouble() * L_y);
        pos_z[n] = (float)(rand.NextDouble() * L_z);
        testid = true;
        // проверка на взаимное перекрытие частицами
        for (int i = 0; i < n; i++)
        {
            rx = pos_x[i] - pos_x[n];
            ry = pos_y[i] - pos_y[n];
            rz = pos_z[i] - pos_z[n];
            rx = rx - Convert.ToInt32(rx / L_x) * L_x;
            ry = ry - Convert.ToInt32(ry / L_y) * L_y;
            rz = rz - Convert.ToInt32(rz / L_z) * L_z;
            r2 =(float)Math.Sqrt(rx * rx + ry * ry + rz * rz);
            if (r2 < min_dist)
            {
                // если имеются перекрытия, то генерируются новые координаты
                testid = false;
                break;
            }
        }
    }
}
```

```

    if (testid) n++; // при отсутствии перекрывания генерируются
    координаты следующей частицы
    if (n >= N) load_syst = false; // генерация завершается после
    размещения всех частиц внутри ячейки моделирования
}
load_system_center();
}

```

Результат, получаемый в результате инициализации начальной конфигурации процедурой `gen_order_stat`, представлен на рисунке 22, где изображены конфигурации системы в виде кристаллических ГЦК и ПК решеток.

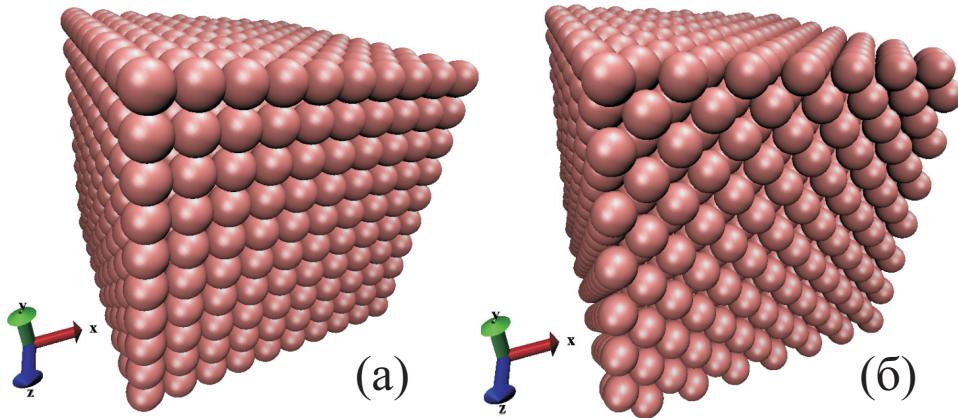


Рис. 22: Конфигурации системы, полученные с помощью процедуры `gen_order_stat`: (а) – кристалл с простой кубической решеткой (ПК) с количеством частиц $N = 1000$; (б) – кристалл с гранецентрированной кубической решеткой (ГЦК) с количеством частиц $N = 1372$.

§4.3 Функция радиального распределения частиц

Структура многочастичной системы может быть охарактеризована с помощью функции распределения положений частиц. С помощью этой функции можно получить информацию о структурных особенностях системы и осуществить сопоставление результатов моделирования с экспериментом, что важно в компьютерном дизайне новых материалов. Исследование структуры, исследование ближнего и дальнего порядка, можно реализовать через парную функцию радиального распределения частиц [4, 17, 34]:

$$g(\vec{r}) = \frac{V}{N^2} \left\langle \sum_{i=1}^{N-1} \sum_{j>i}^N \delta(\vec{r} - \vec{r}_{ij}) \right\rangle. \quad (76)$$

Эта функция определяет вероятность нахождения двух частиц в системе на расстоянии $r = |\vec{r}|$ друг от друга. В численных расчетах выражение (76), как правило, заменяется соотношением [4, 17]

$$g(r) = \frac{V}{4\pi r^2 N} \left\langle \sum_{i=1}^N \frac{\Delta n_i(r)}{\Delta r} \right\rangle. \quad (77)$$

Здесь $\Delta n_i(r)$ – количество частиц в некотором сферическом слое толщиной Δr (преимущественно $\Delta r \leq 0.01\sigma$) на расстоянии r от i -той частицы (см. рисунок 23). Угловые скобки $\langle \dots \rangle$ означают статистическое усреднение по ансамблю частиц.

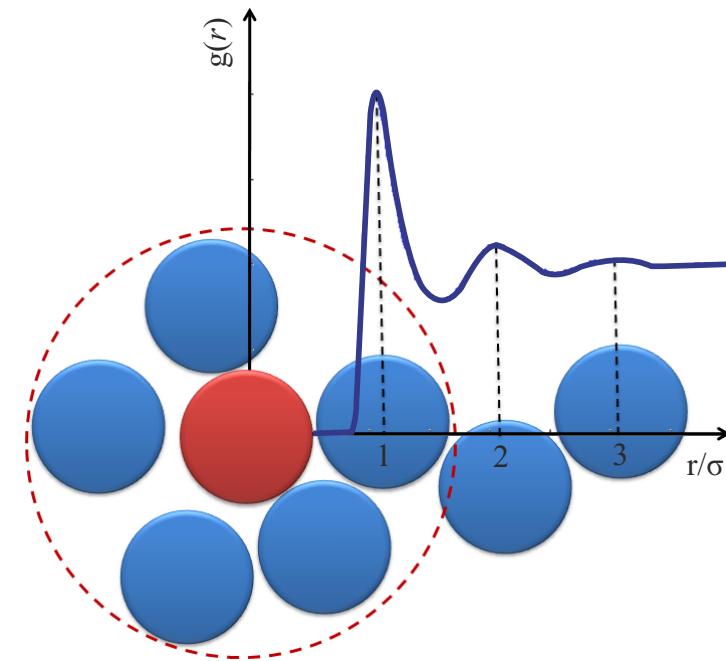


Рис. 23: Схематичный рисунок конфигурации двумерной системы, поясняющий расчет функции радиального распределения частиц. Кружками обозначены частицы.

В случае идеального кристалла функция $g(r)$ характеризуется последовательностью узких обособленных пиков. Положения этих пиков однозначно определяются типом кристаллической решетки. Таким образом, функция $g(r)$ может использоваться для расшифровки атомарной/молекулярной структуры кристаллов, т.е. $g(r)$ может выступить в качестве анализируемой величины в структурном анализе. Далее, ширины пиков определяются возможными смещениями атомов/молекул кристалла от своих равновесных положений, что подразумевает их непосредственную зависимость от температуры системы. Следует отметить, что в случае кристалла положения пиков соответствуют радиусам соответствующих координационных сфер.

Функция радиального распределения атомов/молекул жидкости, $g(r)$, имеет характерный вид: наличие нескольких максимумов и минимумов на расстояниях, соответствующих первой, второй и т. д. псевдокоординационным сферам; с увеличением расстояния r функция радиального распределения частиц в жидкости выходит на асимптоту $g(r) \rightarrow 1$. Такая форма функции $g(r)$ связывается с наличием в жидкостях так называемого ближнего порядка [4, 34].

В случае разреженных газов функция $g(r)$, как правило, имеет вид «ступеньки», положение которой соотносится с размером атомов/молекул газа. Такая форма $g(r)$ обусловлена отсутствием каких-либо корреляций в расположении частиц.

Функции радиального распределения частиц для некоторых фаз представлены на рисунке 24.

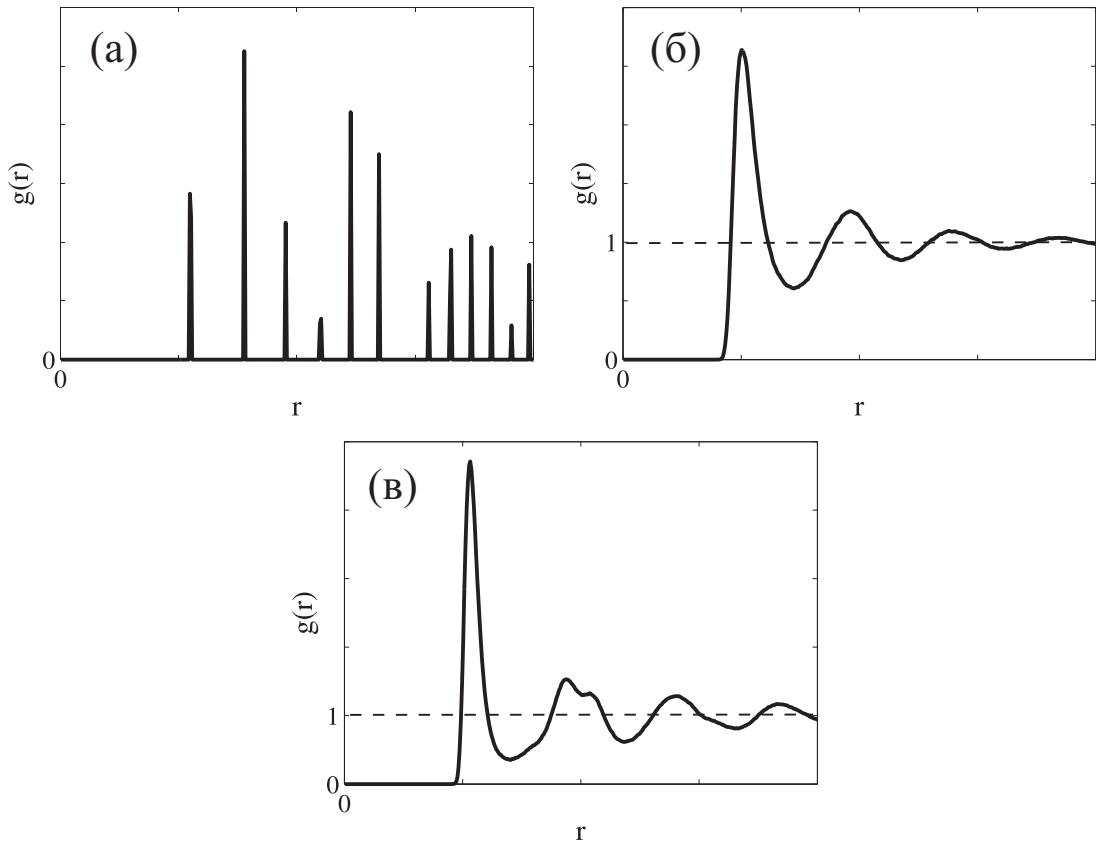


Рис. 24: Схематичный рисунок функции радиального распределения частиц: (а) – кристалл, (б) – жидкость, (в) – аморфное тело.

Ниже приведен программный код 4.3 для расчета функции радиального распределения частиц, где конфигурация системы считывается из выходного файла `system_config.txt`.

Код 4.3

```
static void Main(string[] args)
{
    Program p = new Program();
    p.load_RDF();
    Console.WriteLine("Расчет окончен");
    Console.ReadLine();
}

# region Глобальные переменные
float[] pos_x, pos_y, pos_z, gsum, gs;
int N, r;
float L_x, L_y, L_z, dr;
int Max, end_step, begin_step;
float r1, rx, ry, rz, rij;
#endregion
public void load_RDF()
{
    // задаются детали расчетов
    string st, sts;
    int idx;
    string id1, id2;
    N = 1372;
    L_x = 12.25f;
    L_y = 12.25f;
    L_z = 12.25f;
    dr = 0.01f;
    begin_step = 1;
    end_step = 1000;
    pos_x = new float[N];
    pos_y = new float[N];
    pos_z = new float[N];
    gsum = new float[10 * N];
    gs = new float[10 * N];
    // указывается адрес файла с входными данными - координатами частиц
    FileStream fs = new FileStream("system_config.txt", FileMode.Open);
    StreamReader str = new StreamReader(fs);

    for (int i = 1; i < begin_step; i++)
    {
        st = str.ReadLine();
        st = str.ReadLine();
        st = str.ReadLine();
        st = str.ReadLine();
```

```

st = str.ReadLine();
for (int j = 0; j < N; j++)
{
    st = str.ReadLine();
}
}

// выполняются расчеты end_step-begin_step+1 число раз
for (int step = begin_step; step <= end_step; step++)
{
    st = str.ReadLine();
    for (int i = 0; i < N; i++)
    {
        // чтение координат из файла
        st = str.ReadLine();
        sts = st.Replace(".", ",");
        idx = sts.IndexOf(",");
        id1 = sts.Substring(0, idx);
        st = sts.Remove(0, idx + 1);
        idx = st.IndexOf(" ");
        id2 = st.Substring(0, idx);
        sts = st.Remove(0, idx + 1);
        pos_x[i] = Convert.ToSingle(sts.Substring(0, idx));
        st = sts.Remove(0, idx + 1);
        idx = st.IndexOf(" ");
        pos_y[i] = Convert.ToSingle(st.Substring(0, idx));
        sts = st.Remove(0, idx + 1);
        pos_z[i] = Convert.ToSingle(sts);
    }
}

// расчет функции радиального распределения частиц
Max = 0;
for (int i = 0; i < N - 1; i++)
{
    for (int j = i + 1; j < N; j++)
    {

```

```
    rx = pos_x[i] - pos_x[j];
    ry = pos_y[i] - pos_y[j];
    rz = pos_z[i] - pos_z[j];
    rx = rx - Convert.ToInt32(rx / L_x) * L_x;
    ry = ry - Convert.ToInt32(ry / L_y) * L_y;
    rz = rz - Convert.ToInt32(rz / L_z) * L_z;
    rij = (float)Math.Sqrt(rx * rx + ry * ry + rz * rz);
    r = (int)(rij / dr);
    if (r > Max) Max = r;
    gsum[r] = gsum[r] + 1;
}
}

// нормировка функции радиального распределения частиц
for (int i = 0; i <= Max; i++)
{
    r1 = i * dr + dr / 2.0f;
    gs[i] = gs[i] + 2.0f * gsum[i] * L_x * L_y * L_z / (N * N * 4.0f *
(float)Math.PI * r1 * r1 * dr);
    gsum[i] = 0;
}
// на экран выводится текущая информация о выполнении расчетов
Console.WriteLine("step / end step = " + step.ToString() + "/" +
end_step.ToString());
}

str.Close(); fs.Close();
// результат записывается в файл
FileStream fsw = new FileStream("rdf_file.txt", FileMode.Create);
StreamWriter stw = new StreamWriter(fsw);
for (int i = 0; i <= Max; i++)
{
    r1 = i * dr + dr / 2.0f;
    gs[i] = gs[i] / (end_step - begin_step + 1);
    if (r1 <= L_x / 2.0)
    {
        st = r1.ToString() + " " + gs[i].ToString();
        st = st.Replace(",", ".");
        stw.WriteLine(st);
    }
}
stw.Close(); fsw.Close();
}
```

§4.4 Моделирование фазовых переходов в модельной системе Леннард-Джонса

В качестве примера рассмотрим моделирование фазовых переходов в однокомпонентной системе Леннард-Джонса.

Этап I. *Плавление кристаллической леннард-джонсовской системы* при температуре $T = 2.8 \epsilon/k_B$ (ϵ – есть глубина ямы в потенциале межчастичного взаимодействия) и при отсутствии внешнего давления, где система переходит из кристаллической фазы в жидкую фазу. Моделирование реализуется в NVT-ансамбле, где температура контролируется через масштабирование скоростей частиц. Исходная конфигурация системы генерируется в виде кристаллической ГЦК решетки, состоящей из $N = 1372$ частиц (т.е. $n_x = 7$, $n_y = 7$, $n_z = 7$). Частицы располагаются внутри кубической ячейки с длинами ребер $L_x = L_y = L_z \simeq 12.25\sigma$ и объемом $V \simeq 1838\sigma^3$. На ячейку моделирования накладываются периодические граничные условия. Система характеризуется постоянной количественной плотностью $\rho \simeq 0.75\sigma^{-3}$. Интегрирование уравнений движения осуществляется с помощью скоростного алгоритма Верле с временным шагом $\Delta t = 0.005\tau$. Для ускорения вычислительных операций используется «список Верле» с параметром $\Delta r_s = 0.3\sigma$. Нагревание системы осуществляется со скоростью $v_T = 0.02\epsilon/(k_B\tau)$ на протяжении временного окна в $M = T_{ext}/(v_T\Delta t) \simeq 25000$ временных шагов. Дополнительно 15000 временных шагов используется для того, чтобы привести систему в состояние термодинамического равновесия. Всего выполняется $M = 40000$ МД-итераций, что соответствует временному интервалу $t = M\Delta t = 200\tau$. Будем рассчитывать все величины в параметрах потенциала Леннард-Джонса.

Детали моделирования на этапе I задаются следующим образом:

```
# region Детали моделирования
N = 1372;
M = 40000;
alat = 1.75f; nx = 7; ny = 7; nz = 7;
dt = 0.005f;
T_ext = 2.8f;
vel_dT = 0.02f;
dr_s = 0.3f;
init_type = 1;
cryst_type = "гцк";
time = 0;
```

```
#endregion
```

На основе полученной конфигурации системы выполняется визуализация с помощью программы VMD. Также выполняется расчет функции радиального распределения частиц, которое представляется в графическом виде в вычислительном пакете Матлаб. Отслеживается временная зависимость термодинамических параметров для равновесной системы; данные сохраняются во внешнем файле и затем отображаются в графическом виде с помощью пакета Матлаб.

Этап II. Увеличение плотности системы от $\rho \simeq 0.75 \sigma^{-3}$ до $\rho \simeq 1.05 \sigma^{-3}$ при постоянной температуре $T = 2.8 \epsilon/k_B$. Здесь используется NPT-ансамбль, где температура и давление системы контролируются с помощью термостата и баростата Нозе-Гувера с параметрами $Q_T = 100 \epsilon \tau^2$ и $Q_B = 200 \epsilon \tau^2$ соответственно. Количественная плотность системы увеличивается в результате уменьшения объема системы при внешнем давлении $P = 22 \epsilon/\sigma^3$. Исходной конфигурацией системы является термодинамически равновесная жидкость, «приготовленная» на предыдущем этапе. Система приводится в термодинамически равновесное состояние при температуре $T = 2.8 \epsilon/k_B$ и давлении $P = 22 \epsilon/\sigma^3$ на временном интервале в $M = 40000$ временных шагов, что соответствует временному масштабу $t = 200 \tau$ ($\Delta t = 0.005 \tau$).

Детали моделирования на этапе II задаются следующим образом:

```
# region Детали моделирования
N = 1372;
M = 40000;
dt = 0.005f;
T_ext = 2.8f; T0 = 2.8f;
Q_T = 100.0f;
P_ext = 22.0f;
Q_B = 200.0f;
vel_dT = 0.0f;
dr_s = 0.3f;
time = 0;
#endregion
```

После выполнения моделирования на основе полученной конфигурации системы выполняется визуализация с помощью программы VMD. Также выполняется расчет функции радиального распределения частиц при различных плотностях системы. Полученные функции $g(r)$ представляются в графическом виде с помощью пакета

Матлаб.

Этап III. Охлаждение равновесной жидкости до температуры $T_{ext} = 0.4 \epsilon/k_B$. Используется NVT-ансамбль с термостатом Нозе-Гувера (параметр $Q_T = 100 \epsilon\tau^2$). Плотность системы является постоянной и равной $\rho \simeq 1.05 \sigma^{-3}$. Исходной конфигурацией системы является термодинамически равновесная жидкость, приготовленная на предыдущем этапе. Охлаждение системы происходит со скоростью $v_T = 0.02 \epsilon/(k_B\tau)$ на протяжении времени в $M = 24000$ временных шагов. Дополнительные 10000 временных шагов позволяют привести систему в термодинамически равновесное состояние. Весь расчетный временной интервал составляет $M = 34000$ временных шагов, т.е. $t = 170\tau$ ($\Delta t = 0.005\tau$).

Детали моделирования на этапе III задаются следующим образом:

```
# region
N = 1372;
M = 34000;
dt = 0.005f;
T_ext = 0.4f; T0 = 2.8f;
Q_T = 100.0f;
vel_dT = 0.0f;
dr_s = 0.3f;
time = 0;
#endregion
```

После завершения охлаждения на основе полученной конфигурации выполняется визуализация системы с помощью программы VMD. Далее при различных температурах рассчитывается функция радиального распределения частиц, кривые которой представляются в графическом виде с помощью пакета Матлаб. Также отслеживается временная эволюция термодинамических параметров системы. Полученные данные сохраняются в файл и в графическом виде представляются с помощью пакета Матлаб.

Каждый описанный выше этап моделирования представлен на фазовой диаграмме на рисунке 25, где стрелкой обозначены переходы между этапами I, II и III.

Результаты моделирования на этапе I. На рисунке 26а изображена конфигурация жидкой системы при температуре $T = 2.8 \epsilon/k_B$ и при плотности $\rho \simeq 0.75 \sigma^{-3}$. Структурные изменения, происходящие в системе в результате плавления кристалла с ГЦК решеткой,

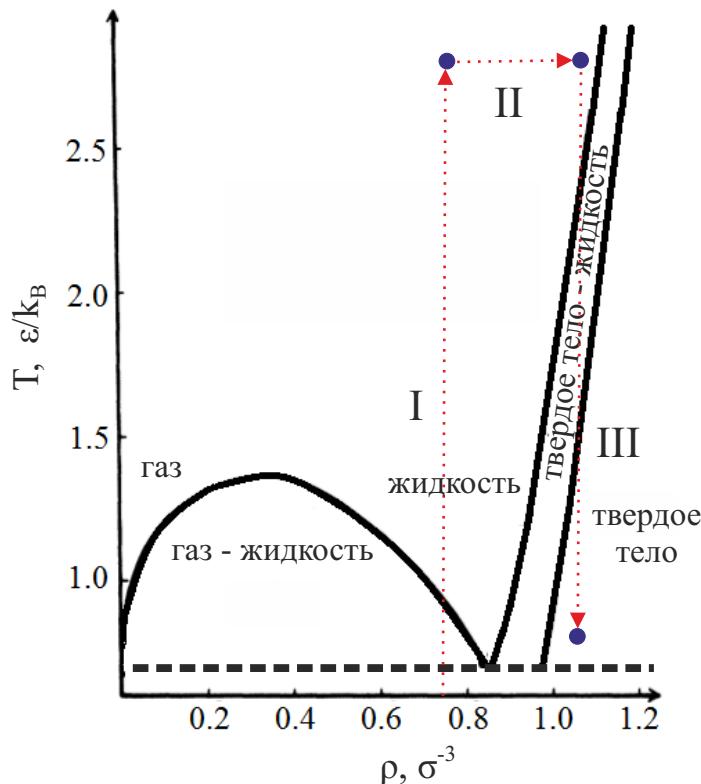


Рис. 25: (T, ρ) -диаграмма однокомпонентной системы Леннард-Джонса [16]. Стрелками обозначены переходы между этапами I, II и III.

анализируются с помощью функции радиального распределения частиц $g(r)$. На рисунке 26б представлены кривые $g(r)$, рассчитанные при различных температурах системы. Из рисунка видно, что пики, характерные ГЦК структуре, постепенно сглаживаются. Для жидкости характерно наличие нескольких чередующихся максимумов и минимумов различной интенсивности.

После перехода жидкой системы в состояние термодинамического равновесия значения таких величин, как температура, внутреннее давление, потенциальная и кинетическая энергии начинают осциллировать вблизи некоторого равновесного значения. На рисунке 27 изображена временная зависимость полной энергии системы (как сумма потенциальной E_{pot} и кинетической E_{kin} энергий), потенциальной энергии E_{pot} , внутреннего давления P и температуры T . Рассматривается временной отрезок $t \in [200; 250] \tau$. Из рисунка 27 видно, что при температуре $T = 2.8 \epsilon/k_B$ внутреннее давление системы, рассчитанное с помощью выражения (19), составляет $P \simeq 2.3 \epsilon/\sigma^3$.

Полученные конфигурации можно использовать для исследования фазовых переходов: плавления, испарения, аморфизации, кри-

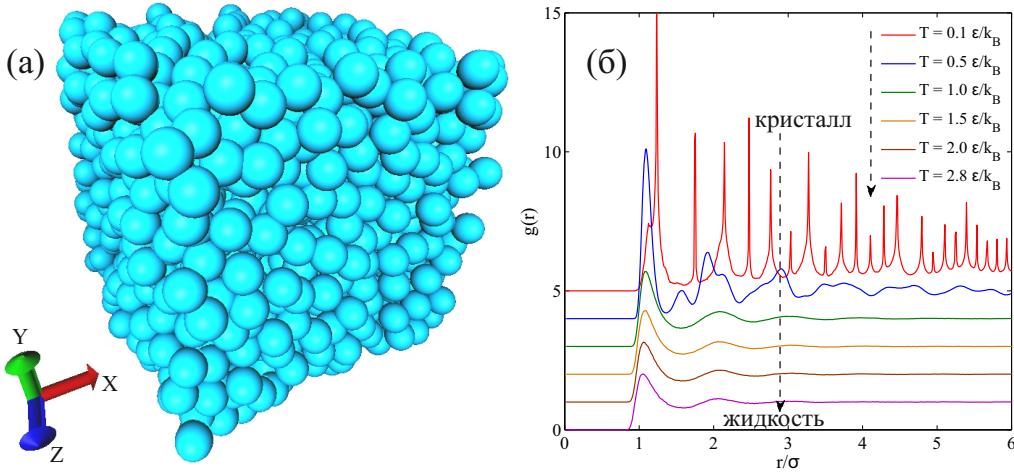


Рис. 26: (а) – конфигурация жидкой системы при температуре $T = 2.8 \epsilon/k_B$ и при плотности $\rho \simeq 0.75 \sigma^{-3}$; (б) – функции радиального распределения частиц, рассчитанные при различных температурах системы.

стализации.

Результаты моделирования на этапе II. Приготовленный на первом этапе жидкий образец испытывает внешнее давление $P = 22\epsilon/\sigma^3$. Количественная плотность системы достигает значения $\rho \simeq 1.05 \sigma^{-3}$. Изменение объема системы в этом случае составляет $\Delta V \simeq 523 \sigma^3$, что соответствует уменьшению объема до $V \simeq 1301 \sigma^3$. На рисунке 28а изображена конфигурация при температуре $T = 2.8 \epsilon/k_B$ и внешнем давлении $P = 22 \epsilon/\sigma^3$. С помощью функции радиального распределения частиц $g(r)$ фиксируются структурные изменения при увеличении плотности системы от $\rho \simeq 0.75 \sigma^{-3}$ до $\rho \simeq 1.05 \sigma^{-3}$ (см. рисунок 28б). На рисунке видно смещение пиков $g(r)$ в область малых расстояний, что обусловлено увеличением плотности. При этом система по прежнему находится в жидком состоянии.

Результаты моделирования на этапе III. На рисунке 29а изображена конфигурация системы, полученная при охлаждении со скоростью $v_T = 0.02 \epsilon/(k_B \tau)$ до температуры $T = 0.4 \epsilon/k_B$. На функции радиального распределения частиц $g(r)$ наблюдается расщепление второго пика, что связывается с переходом жидкости в аморфную фазу (см. рисунок 29б). При охлаждении жидкости до температуры $T = 0.4 \epsilon/k_B$ потенциальная энергия системы возрастает, а кинетическая энергия уменьшается. Внутреннее давление системы составляет $P \simeq 4.4 \epsilon/\sigma^3$. В состоянии термодинамического равновесия температура, внутреннее давление, потенциальная и полная энергии аморфной системы осцилируют вблизи некоторого равновесного значения (см. рисунок 30). На рисунке 30а видно, что по-

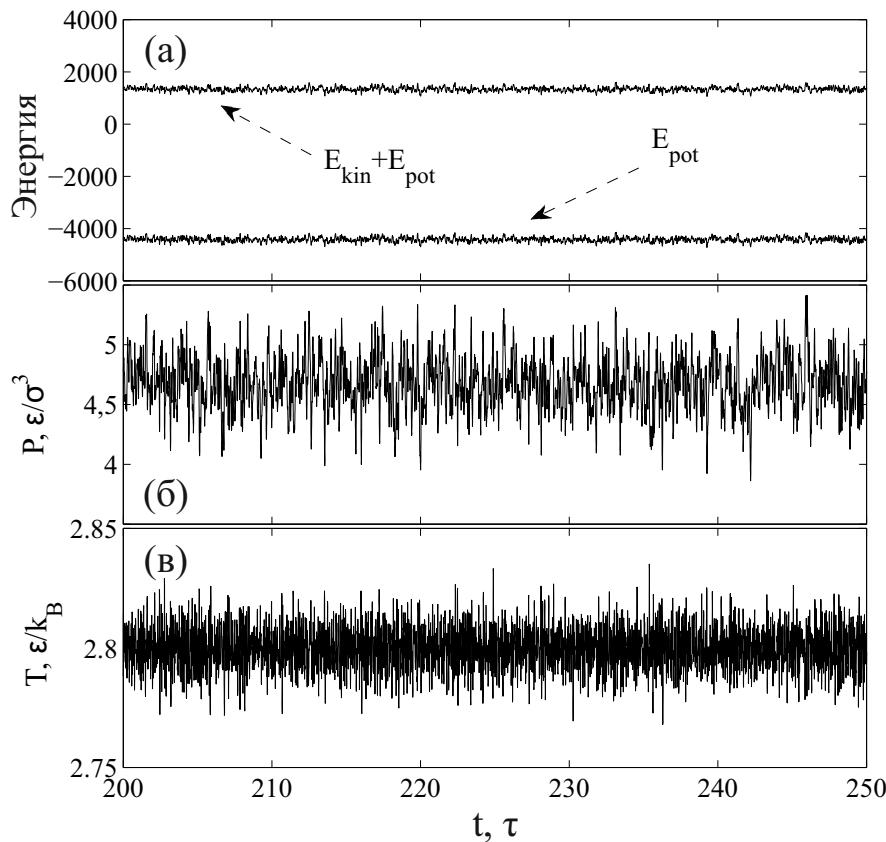


Рис. 27: (а) – временная зависимость полной энергии системы, представленной как сумма потенциальной E_{pot} и кинетической E_{kin} энергий; (б) – временная зависимость внутреннего давления системы, рассчитанная с помощью выражения (19); (в) – временная зависимость температуры системы.

тенциальная энергия системы со временем начинает возрастать, при этом внутреннее давление системы со временем быстро уменьшается (см. рисунок 30б). Это обусловлено слабыми стеклоформирующими свойствами леннард-ジョンсовской системы.

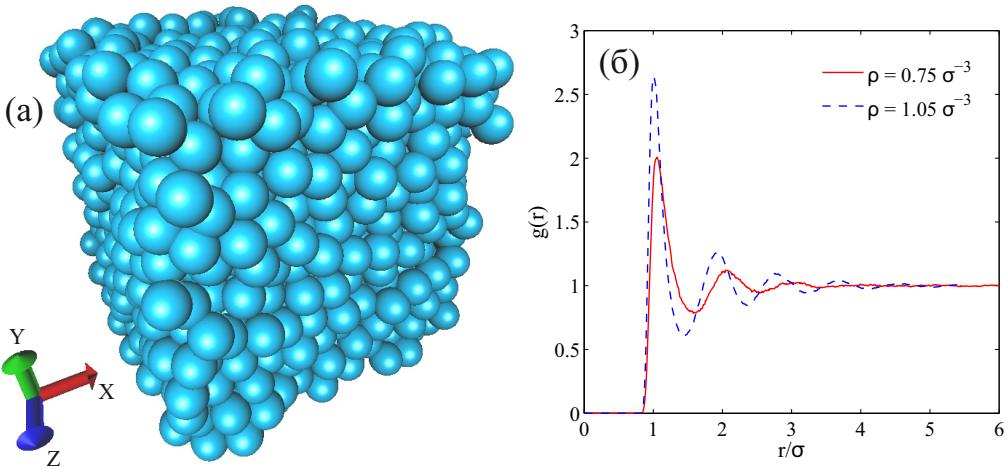


Рис. 28: (а) – конфигурация жидкой системы при температуре $T = 2.8 \epsilon/k_B$ и при плотности $\rho \simeq 1.05 \sigma^{-3}$; (б) – кривые функции радиального распределения частиц, рассчитанные для жидкой системы при плотностях $\rho \simeq 0.75 \sigma^{-3}$ и $\rho \simeq 1.05 \sigma^{-3}$.

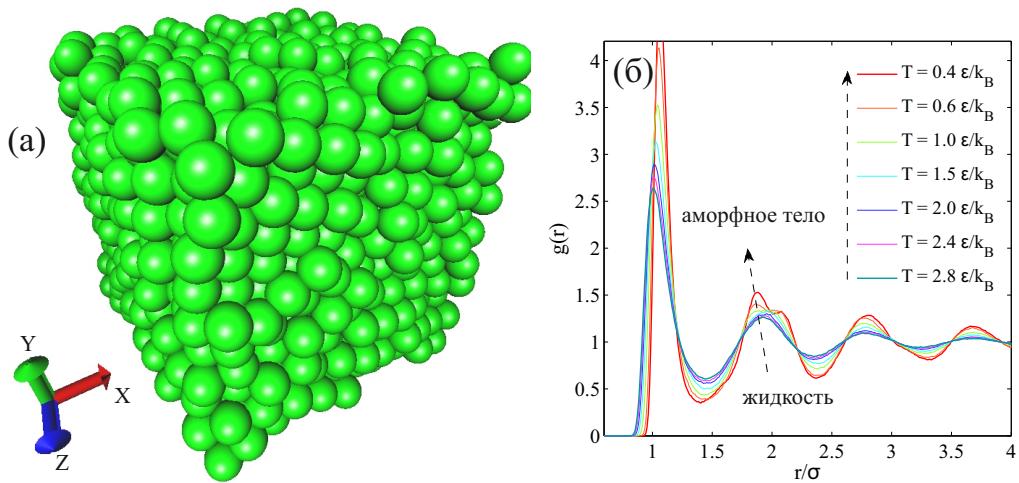


Рис. 29: (а) – конфигурация аморфной системы при температуре $T = 0.4 \epsilon/k_B$; (б) – кривые функции радиального распределения частиц при различной температуре, рассчитанные при быстром охлаждении системы.

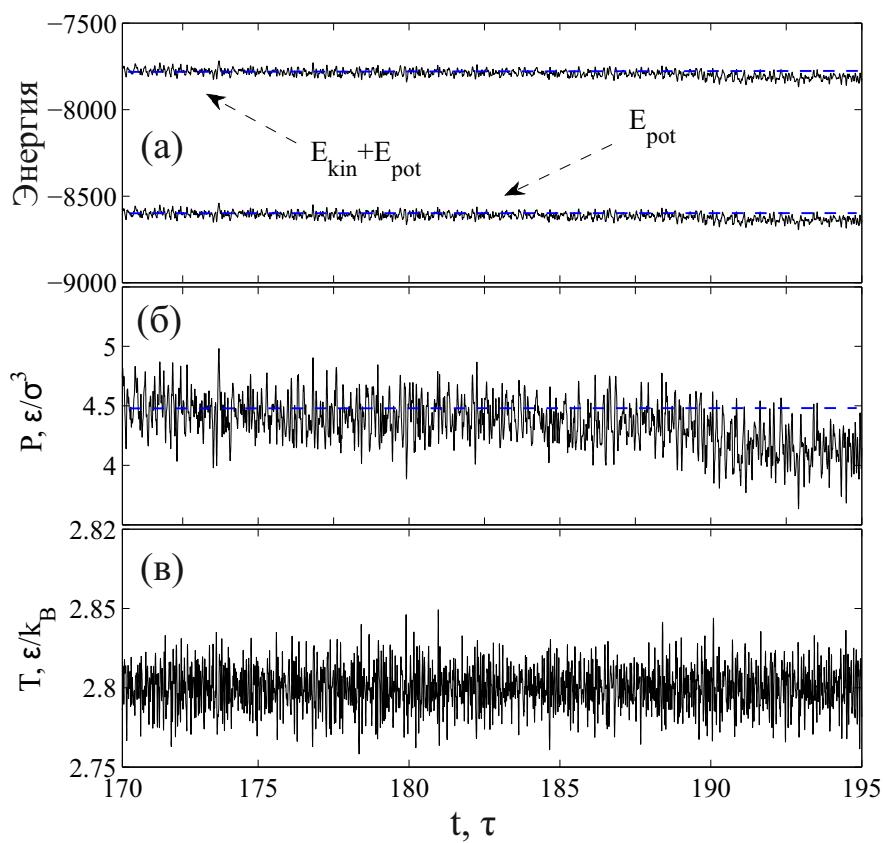


Рис. 30: (а) – временная зависимость полной энергии аморфной системы, представленной как сумма потенциальной E_{pot} и кинетической E_{kin} энергий; (б) – временная зависимость внутреннего давления, рассчитанная с помощью выражения (19); (в) – временная зависимость температуры аморфной системы.

Задания для контроля знаний

1. В чем заключаются сходства и различия в методах моделирования молекулярной динамики? Опишите преимущества и недостатки этих методов.
2. Назовите причину использования классического метода моделирования молекулярной динамики преимущественно для изучения жидкого состояния, по сравнению с твердым телом и газом?
3. Перейти от леннард-джонсовских единиц измерения в международную систему единиц (СИ): $T = 2.5 \epsilon/k_B$, $P = 1.8 \epsilon/\sigma^3$, $E = 50 \epsilon$. Здесь $\epsilon = 150 K \cdot k_B$, $\sigma = 0.29 \text{ нм}$, $k_B = 1.38 \cdot 10^{-23} \text{ Дж}/\text{К}$.
4. Рассчитать скорость v , ускорение a и силу F , в единицах системы СИ, если $v = 3.7 \sigma/\tau$, $a = 1.5 \sigma/\tau^2$, $F = 20 \epsilon/\sigma$. Здесь $\epsilon = 210 K \cdot k_B$, $\sigma = 0.32 \text{ нм}$, $m = 25.2 \cdot 10^{-27} \text{ кг}$.
5. Представить величины в леннард-джонсовских единицах:
 - а) температура: $T = 2000 \text{ К}$, $T = 273 \text{ К}$, $T = 50 \text{ К}$;
 - б) давление: $P = 20000 \text{ Па}$, $P = 50 \text{ атм}$, $P = 600 \text{ МН}/\text{м}^2$;
 - в) энергия: $E = 0.5 \cdot 10^{-20} \text{ Дж}$, $E = 0.26 \text{ эВ}$, $E = 10 \cdot 10^{-18} \text{ Дж}$;
 - г) скорость: $v = 200 \text{ м}/\text{с}$, $v = 120 \text{ см}/\text{с}$, $v = 1000 \text{ м}/\text{с}$.Здесь $\epsilon = 100 K \cdot k_B$, $\sigma = 0.25 \text{ нм}$, $m = 2.5 \cdot 10^{-27} \text{ кг}$.
6. Какова связь метода молекулярной динамики со статистической механикой? Как определяется средняя по ансамблю?
7. Определить положение частицы в моменты времени $t_1 = 1 \text{ пс}$, $t_2 = 1.2 \text{ пс}$ и $t_3 = 1.4 \text{ пс}$, если она движется внутри ячейки моделирования с постоянной скоростью, компоненты которой составляют $v_x = 50 \text{ м}/\text{с}$, $v_y = 100 \text{ м}/\text{с}$ и $v_z = 0$. Начальные координаты частицы $x_0 = 2 \text{ нм}$, $y_0 = 4.5 \text{ нм}$ и $z_0 = 1.0 \text{ нм}$.
8. Сколько потребуется временных шагов M при моделировании на протяжении времени $t = 10 \text{ нс}$, если временной шаг интегрирования уравнений движения составляет $\Delta t = 1 \text{ фс}$?

- 9.** Сколько временных шагов потребуется для того, чтобы нагреть систему от $T = 250\text{ K}$ до $T = 400\text{ K}$ со скоростью $v_T = 10^{10}\text{ K/c}$, если расчеты выполняются с временным шагом $\Delta t = 10\text{ фс}$?
- 10.** Определить количество временных шагов, достаточных для охлаждения системы от $T = 360\text{ K}$ до $T = 273\text{ K}$ со скоростью $v_T = 10^{11}\text{ K/c}$, если временный шаг составляет $\Delta t = 1.5\text{ фс}$.
- 11.** Рассчитать количество временных шагов, необходимых для охлаждения системы от $T = 2.5\epsilon/k_B$ до $T = 0.4\epsilon/k_B$ со скоростью $v_T = 0.005\epsilon/(k_B\tau)$, если временный шаг составляет $\Delta t = 0.002\tau$.
- 12.** Определить временный шаг Δt , если для нагревания системы от $T = 0.2\epsilon/k_B$ до $T = 3.2\epsilon/k_B$ со скоростью $v_T = 0.01\epsilon/(k_B\tau)$ требуется $M = 10^5$ временных шагов.
- 13.** Определить скорость охлаждения системы v_T в единицах СИ, если для охлаждения от $T = 2.8\epsilon/k_B$ до $T = 0.6\epsilon/k_B$ требуется $M = 10^5$ временных шагов при $\Delta t = 5 \cdot 10^{-3}\tau$. Здесь $\epsilon = 110K \cdot k_B$, $m = 24 \cdot 10^{-27}\text{ кг}$, $\sigma = 0.35\text{ нм}$.
- 14.** Каким образом в классическом методе моделирования молекулярной динамики учитываются эффекты конечного размера системы? Как эти эффекты влияют на результат молекулярно-динамических расчетов?
- 15.** Построить алгоритм периодических граничных условий для сферической ячейки моделирования радиуса R . Представить блок-схему.
- 16.** Выполняется моделирование двумерной системы с помощью ячейки моделирования круглой формы. Радиус ячейки составляет R . Какие периодические условия должны использоваться в этом случае?
- 17.** Имеется некоторая плоская система, представленная ячейкой моделирования шестиугранной формы. Длины ребер r . Записать периодические граничные условия.
- 18.** Исследуйте применимость периодических граничных условий в случае NVT- и NPT-ансамблей.
- 19.** Построить алгоритм периодических граничных условий для прямоугольной системы, если ребра ячейки $L_x = 10\sigma$, $L_y = 2.0L_x$ и $L_z = 0.5L_y$. Представить блок-схему.

20. Определить координаты частицы $r'(x', y', z')$, покинувшей ячейку моделирования с длинами сторон от 0 до $L_x = 10\sigma$, $L_y = 15\sigma$, $L_z = 10\sigma$, при использовании периодических граничных условий. Рассмотреть случай, при котором предыдущее положение частицы составляет:

- а) $x = 11.2\sigma$, $y = 5.6\sigma$, $z = 10.1\sigma$;
- б) $x = 2.3\sigma$, $y = 10.0\sigma$, $z = 10.3\sigma$;
- в) $x = 11.2\sigma$, $y = 11.0\sigma$, $z = 10.5\sigma$;
- г) $x = 0.5\sigma$, $y = -0.6\sigma$, $z = -1.2\sigma$;
- д) $x = -2.5\sigma$, $y = 2.6\sigma$, $z = 10.2\sigma$;
- е) $x = -1.5\sigma$, $y = -0.1\sigma$, $z = 11.2\sigma$.

Для каждого случая представить схематичный рисунок, показывающий изменение положения частицы относительно ячейки моделирования.

21. Рассчитать изменение расстояния между частицами r'_{xij} , r'_{yij} и r'_{zij} с учетом периодических граничных условий, если:

- а) $r_{xij} = 4\sigma$, $r_{yij} = 8\sigma$, $r_{zij} = -4\sigma$;
- б) $r_{xij} = 5\sigma$, $r_{yij} = -9\sigma$, $r_{zij} = 14\sigma$;
- в) $r_{xij} = -4.5\sigma$, $r_{yij} = 9.5\sigma$, $r_{zij} = 6.2\sigma$;
- г) $r_{xij} = -5\sigma$, $r_{yij} = 3\sigma$, $r_{zij} = -14.5\sigma$;
- д) $r_{xij} = 3.5\sigma$, $r_{yij} = 8.2\sigma$, $r_{zij} = 14\sigma$;
- е) $r_{xij} = 2\sigma$, $r_{yij} = -5.5\sigma$, $r_{zij} = 10.2\sigma$.

Размер ячейки моделирования составляет: $-4\sigma \leq L_x \leq 4\sigma$, $-6\sigma \leq L_y \leq 6\sigma$ и $-10\sigma \leq L_z \leq 10\sigma$.

22. Оценить среднеквадратичное отклонение Δr для координат атомов при интегрировании уравнений движения с помощью скоростного алгоритма Верле. Примечание: получить выражение для расчета Δr .

23. Определить температуру системы, если среднеквадратичная скорость движения частиц составляет:

- а) $\bar{v} = 250$ м/с;
- б) $\bar{v} = 600$ нм/нс;
- в) $\bar{v} = 10^5$ см/с;
- г) $\bar{v} = 5 \cdot 10^{-10}$ м/пс.

Масса частиц $m = 24.5 \cdot 10^{-27}$ кг.

24. Определить среднеквадратичную скорость движения частиц одинаковой массы $m = 32 \cdot 10^{-27}$ кг при температуре:

- а) $T = 50 K$;
- б) $T = 273 K$;
- в) $T = 600 ^\circ C$;
- г) $T = 200 ^\circ C$.

25. Оценить внутреннее давление системы, состоящей из 10^5 частиц, расположенных внутри ячейки моделирования объема $V = 2 \cdot 10^5 \sigma^3$, где $\sigma = 0.28$ нм, если температура системы составляет $T = 300 K$. Вириальным вкладом уравнения (19) пренебречь.

26. Оценить внутреннее давление системы P , состоящей из 10^3 частиц одинаковой массы $m = 41 \cdot 10^{-27}$ кг, движущихся со среднеквадратичной скоростью $\bar{v} = 3\sigma/\tau$. Объем системы составляет $V = 3400 \sigma^3$ ($\sigma = 0.35$ нм, $\epsilon = 90K \cdot k_B$). Вириальным вкладом уравнения (19) пренебречь.

27. Определить температуру системы, если внутреннее давление системы при количественной плотности $\rho = 0.45 \sigma^{-3}$ составляет $P = 12 \epsilon/\sigma^3$. Здесь $\epsilon = 110K \cdot k_B$.

28. Рассчитать температуру системы T и терmostата T_{ext} при терmostатировании методом масштабирования скоростей, если:

- а) $\Delta T = 10 K$, $\lambda_{sc} = 1.02$;
- б) $\Delta T = -25 K$, $\lambda_{sc} = 0.81$;
- в) $\Delta T = -12 K$, $\lambda_{sc} = 0.99$;
- г) $\Delta T = 80 K$, $\lambda_{sc} = 1.2$.

Здесь $\Delta T = T_{ext} - T(t)$.

29. Из предыдущей задачи оценить изменение скорости движения частицы, если ее масса составляет $m = 35 \cdot 10^{-27}$ кг.

30. Определить температуру терmostата T_{ext} , если масштабирование скорости движения частиц происходит при значениях:

- а) $\lambda_{sc} = 0.45$, $T = 100 K$;
- б) $\lambda_{sc} = 1.2$, $T = 2500 K$;
- в) $\lambda_{sc} = 0.98$, $T = 1.5 \epsilon/k_B$;
- г) $\lambda_{sc} = 1.15$, $T = 0.9 \epsilon/k_B$.

31. Рассчитать температуру терmostата Беренсена T_{ext} , если мгновенная температура системы $T = 250 K$. Здесь моделирование мо-

лекулярной динамики выполняется с временным шагом $\Delta t = 10$ фс. Рассмотреть следующие случаи:

- а) $\lambda_B = 1.2$, $Q_T^{(B)} = 100$ фс;
- б) $\lambda_B = 0.85$, $Q_T^{(B)} = 10^{-15}$ с;
- в) $\lambda_B = 0.98$, $Q_T^{(B)} = 10^{-2}$ пс.

32. Определить изменение температуры системы ΔT при использовании термостата Берендсена с параметрами $\lambda_B = 1.05$ и $Q_T^{(B)} = 10$ фс, если температура термостата составляет $T_{ext} = 900 K$. Временной шаг составляет $\Delta t = 0.5$ фс.

33. Для задачи 32 определить изменение скорости движения частицы Δv , если масса частицы составляет $m = 25 \cdot 10^{-27}$ кг.

34. Определить оптимальное значение параметра Q_T термостата Нозе-Гувера, если моделируемая система состоит из 10^{23} частиц. Термостат имеет температуру $350 K$. Время структурной релаксации системы составляет 10 фс.

35. Рассчитать внутреннее давление системы P , если давление баростата Берендсена составляет $P_{ext} = 100$ атм. Рассмотреть следующие случаи:

- а) $\xi_B = 1.2 c^{-1}$, $Q_P^{(B)} / \beta_T = 50$ атм·с;
- б) $\xi_B = 0.85 c^{-1}$, $Q_P^{(B)} / \beta_T = 500$ атм·с;
- в) $\xi_B = 1.05 c^{-1}$, $Q_P^{(B)} / \beta_T = 100$ атм·с.

36. Определить изменение объема системы $\Delta V = V - V_0$ при использовании баростата Нозе-Гувера, если:

- а) $\xi = 1.1 \tau^{-1}$, $\Delta t = 0.01 \tau$;
- б) $\xi = 0.5 c^{-1}$, $\Delta t = 10$ фс;
- в) $\xi = 1.45 \tau^{-1}$, $\Delta t = 0.002 \tau$;
- г) $\xi = 0.95 c^{-1}$, $\Delta t = 0.5$ фс.

Начальный объем системы составляет $V_0 = 2000 \sigma^3$, где $\sigma = 0.25$ нм ($\tau \simeq 10^{-12}$ с).

37. Оценить линейные размеры L'_x , L'_y и L'_z и объем V системы при использовании баростата Нозе-Гувера с параметром: а) $\xi = 0.95 \text{ фс}^{-1}$; б) $\xi = 1.05 \text{ фс}^{-1}$; в) $\xi = 0.86 \text{ фс}^{-1}$ при временному шаге $\Delta t = 0.5$ фс. Начальный размер ячейки моделирования составляет $L_x = 10$ нм, $L_y = 12$ нм и $L_z = 11$ нм.

38. Определить изменение линейных размеров L'_x , L'_y и L'_z и объема V системы при использовании баростата Берендсена с параметром: а) $\xi_B = 1.45 \tau^{-1}$; б) $\xi_B = 0.65 \tau^{-1}$ при временном шаге $\Delta t = 0.002 \tau$. Начальный размер ячейки моделирования $L_x = 20\sigma$, $L_y = 25\sigma$ и $L_z = 10\sigma$.

39. Рассчитать ускорение частицы массой $m = 26 \cdot 10^{-27}$ кг, если со стороны остальных частиц на нее действует сила, равная $F = 500 \cdot 10^{-14}$ Н.

40. Рассчитать скорость движения и ускорение частицы массой $m = 25 \cdot 10^{-27}$ кг, если на нее действует, сила равная $F = 350 \cdot 10^{-10}$ Н. Изменение скорости происходит на протяжении времени $t = 0.5$ фс.

41. Каким образом выполняются законы сохранения импульса и энергии при моделировании молекулярной динамики? В каком случае не выполняется закон сохранения момента импульса?

42. Рассчитать импульсы двух частиц одинаковой массы $m = 26.8 \cdot 10^{-27}$ кг после их абсолютно упругого соударения, если до столкновения они двигались со скоростями $v_1 = 200$ м/с и $v_2 = 350$ м/с. Отношение скоростей частиц после их столкновения составляет $v'_1/v'_2 = 0.8$. Движение частиц является одномерным.

43. Определить скорость совместного движения двух частиц после их абсолютно неупругого соударения, если частицы двигались со скоростями $v_1 = 460$ м/с и $v_2 = 250$ м/с. Масса частиц составляет $m_1 = 24.8 \cdot 10^{-27}$ кг и $m_2 = 28.2 \cdot 10^{-27}$ кг. Движение частиц является одномерным.

44. Определить значение гамильтониана системы, если потенциальная энергия всей системы составляет $E_{pot} = -0.025$ эВ. Частицы системы имеют одинаковую массу $m = 28.5 \cdot 10^{-27}$ кг, среднеквадратичная скорость движения которых составляет $\bar{v} = 150$ м/с. Определить, в какой фазе может находиться система.

45. Определить кинетические и потенциальные энергии в расчете на один атом. Вывести формулы для расчета соответствующих величин.

46. Рассчитать кинетическую энергию частицы массой $m = 25.5 \cdot 10^{-27}$ кг, если компоненты скорости ее движения составляют $v_x = 50$ м/с, $v_y = -43$ м/с и $v_z = 1.5$ м/с.

47. Определить суммарную кинетическую энергию трех частиц системы одинаковой массы $m = 42 \cdot 10^{-27}$ кг, если их импульсы составляют $p_1 = 3.2 \cdot 10^{-14}$ кг·м/с, $p_2 = 5.7 \cdot 10^{-14}$ кг·м/с и $p_3 = 10.5 \cdot 10^{-14}$ кг·м/с.

48. Определить полную энергию системы, состоящей из 500 атомов, если удельная кинетическая и потенциальная энергии системы составляют $E_{kin} = 0.002$ эВ и $E_{pot} = -0.008$ эВ соответственно.

49. Определить удельную потенциальную энергию системы, состоящей из 2500 атомов, если полная и кинетическая энергии системы составляют $E = 60$ эВ и $E_{kin} = 80$ эВ соответственно.

50. Какими способами можно оценить переход системы в состояние термодинамического равновесия?

51. Рассчитать координаты частицы в двумерной системе в моменты времени $t_1 = 1.5\tau$ и $t_2 = 2.5\tau$, если частица движется с ускорением $a_x = 10\sigma/\tau^2$ и $a_y = -5\sigma/\tau^2$. Начальные координаты частицы $x_0 = 0.5\sigma$ и $y_0 = 1.2\sigma$. Компоненты скорости составляют $v_{x0} = 0.5\sigma/\tau$ и $v_{y0} = 2\sigma/\tau$.

52. Получить выражение для расчета силы межчастичного взаимодействия F в случае потенциала Морзе, определяемого выражением (52).

53. Получить выражение для силы межчастичного взаимодействия F в случае потенциала вида $U = U_{pair} + U_{charge}$, где U_{pair} и U_{charge} определяются выражениями (60) и (61).

54. Найти выражение для силы $F(r_{ij})$ парного взаимодействия частиц в случае потенциалов следующего вида:

$$\begin{aligned} \text{а)} \quad & U(r_{ij}) = \epsilon \left(\frac{\sigma}{r_{ij}} \right)^{12} \exp \left(\frac{\sigma}{r_c - r_{ij}} \right); \\ \text{б)} \quad & U(r_{ij}) = \epsilon r_{ij}^{-m} \sigma^m - A \exp \left[\frac{\sigma - r_{ij}}{\delta} \right] + B \cos \left(\frac{2\pi r_{ij}}{a} \right); \\ \text{в)} \quad & U(r_{ij}) = \epsilon \left(\frac{\sigma}{r_{ij}} \right)^p \exp \left[- \left(\frac{r_{ij}}{\sigma} \right)^3 \right] - \epsilon \ln \left(\frac{\sigma}{r_{ij}} \right); \\ \text{г)} \quad & U(r_{ij}) = \epsilon \left(\frac{\sigma}{r_{ij}} - B(r_{ij}) \right)^n - A \left(\frac{\sigma}{r_{ij}} \right)^m \exp \left(\frac{\gamma}{r_{ij} - r_c} \right); \\ & B(r_{ij}) = D [\cos \theta_{ijk} - \cos \theta_{0ijk}]^2 \left(\frac{\sigma}{r_{ij}} \right)^p. \end{aligned}$$

55. Определить потенциальную энергию двух частиц, расположенных на расстоянии r_{ij} друг от друга, если они взаимодействуют через модельный потенциал Леннард-Джонса с параметрами системы

$\epsilon = 0.26 \text{ эВ}$ и $\sigma = 0.35 \text{ нм}$. Рассмотреть следующие случаи:

- а) $r_{ij} = 0.9\sigma$; г) $r_{ij} = 2.6\sigma$;
б) $r_{ij} = 1.0\sigma$; д) $r_{ij} = 1.2\sigma$;
в) $r_{ij} = 1.5\sigma$; е) $r_{ij} = 0.8\sigma$.

56. Для каждого случая задачи 55 рассчитать силу межчастичного взаимодействия F .

57. Какова суть метода «списка Верле»? Каким образом происходит расчет взаимодействия частиц при использовании «списка Верле»?

58. Каким образом можно уменьшить количество вычислительных операций при расчете взаимодействий атомов/молекул?

59. Применить метод «список Верле» для двумерной системы.

60. В чем преимущество использования табличного представления энергии и сил взаимодействия в методе моделирования молекулярной динамики?

61. Определить выигрыш в производительности при использовании табличного способа представления сил взаимодействия.

62. Как влияет правильная постановка начальных условий и организация расчетных этапов моделирования на скорость вычислений и на точность результатов?

63. Рассчитать значение потенциальной энергии и силы через потенциал Леннард-Джонса при различных значениях межчастичного расстояния. Представить результаты в табличном виде для каждой величины. Примечание: размер таблицы должен составить не менее 20000 строк.

64. Оценить возможные погрешности и определить их источники на различных этапах молекулярно-динамических расчетов.

65. С помощью программы Visual Studio C# создать проект в консольном режиме с названием Comp_MD_Sim.

66. Согласно блок-схеме из рисунка 21 написать процедуру реализации основных этапов моделирования молекулярной динамики с помощью проекта Comp_MD_Sim на языке программирования C#.

67. С учетом заданий 65 и 66 написать и внедрить в программу Comp_MD_Sim процедуру:

- а) периодических граничных условий: boundary_conditions;

б) интегрирования уравнений движения через скоростной алгоритм Верле: `system_dynamics`, где используется термостат, основанный на методе масштабирования скоростей;

в) расчета потенциальной энергии и силы межчастичного взаимодействия: `load_forces`;

г) сохранения конфигурации системы для визуализации с помощью программы VMD: `load_lammps_trajectory`.

68. Построить блок-схему, содержащую этапы моделирования молекулярной динамики согласно задачи 67.

69. Написать программу на языке C#, позволяющую создать конфигурацию системы в виде гранецентрированной кубической (ГЦК) решетки. Примечание: воспользоваться программным кодом 4.2.

70. Написать программу, позволяющую создать конфигурацию системы, где координаты частиц задаются с помощью генератора случайных чисел.

71. Для системы, состоящей из 1000 частиц, расположенных в прямоугольной ячейке с длинами сторон $L_x = 10\sigma$, $L_y = 15\sigma$ и $L_z = 12\sigma$, получить конфигурацию с помощью генератора случайных чисел. Сохранить полученную конфигурацию во внешнем файле и визуализировать с помощью программы VMD.

72. Для системы, состоящей из 2048 частиц, расположенных в кубической ячейке, получить конфигурацию в виде кристаллической ГЦК решетки, где постоянная решетки составляет 1.65σ . Определить количественную плотность системы. Полученный результат визуализировать с помощью программы VMD.

73. Внедрить процедуру генерации начальной конфигурации системы `load_initial_stat` в программу `Comp_MD_Sim`.

74. Внедрить в программу `Comp_MD_Sim` процедуру сохранения в файл текущей конфигурации системы: `save_config`.

75. Внедрить в программу `Comp_MD_Sim` процедуру чтения из файла конфигурации системы: `load_save_config`.

76. Выполнить моделирование молекулярной динамики процесса плавления модельной леннард-джонсовской системы, состоящей из $N = 1372$ частиц, расположенных внутри кубической ячейки. Плотность системы составляет $\rho = 0.86\sigma^{-3}$. Начальная конфигурация

задается в виде кристаллической ГЦК решетки. Моделирование выполняется в NVT-ансамбле, где температура поддерживается методом масштабирования скоростей. Температура термостата $T_{ext} = 2.0 \epsilon/k_B$. Интегрирование уравнений движения осуществляется с помощью скоростного алгоритма Верле с временным шагом $\Delta t = 0.005 \tau$. Рассчитать количество временных шагов, если нагревание системы происходит со скоростью $v_T = 0.02 \epsilon/(k_B\tau)$. Полученный результат визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

77. Смоделировать процесс плавления модельной системы Джугутова, состоящей из $N = 864$ частиц, расположенных внутри кубической ячейки. Плотность системы составляет $\rho = 0.75 \sigma^{-3}$. Начальная конфигурация задается в виде кристаллической ГЦК решетки. Моделирование выполняется в NVT-ансамбле, где температура поддерживается методом масштабирования скоростей. Температура термостата $T_{ext} = 2.4 \epsilon/k_B$. Интегрирование уравнений движения осуществляется с помощью скоростного алгоритма Верле с временным шагом $\Delta t = 0.002 \tau$. Рассчитать количество временных шагов, если нагревание системы происходит со скоростью $v_T = 0.005 \epsilon/(k_B\tau)$. Полученный результат визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

78. Смоделировать процесс плавления модельной системы Леннард-Джонса, состоящей из $N = 8000$ частиц, расположенных внутри кубической ячейки. Плотность системы составляет $\rho = 0.7 \sigma^{-3}$. Начальная конфигурация задается с помощью генератора случайных чисел. Моделирование выполняется в NVT-ансамбле, где температура поддерживается методом масштабирования скоростей. Температура термостата $T_{ext} = 1.5 \epsilon/k_B$. Интегрирование уравнений движения осуществляется с помощью скоростного алгоритма Верле с временным шагом $\Delta t = 0.002 \tau$. Рассчитать количество временных шагов, если нагревание системы происходит со скоростью $0.05 \epsilon/(k_B\tau)$. Полученный результат визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

79. Сохранить текущие конфигурации системы при моделировании процесса плавления, выполняемого в задаче 76, при следующих температурах: $T = 0.5 \epsilon/k_B$, $T = 1.0 \epsilon/k_B$, $T = 1.5 \epsilon/k_B$, $T = 2.0 \epsilon/k_B$ и $T = 2.5 \epsilon/k_B$. Написать соответствующую программу, которая позволит сохранять конфигурацию системы через процедуру

`save_config`, где в названии файла указывается текущая температура.

80. Задачу 79 рассмотреть для случая моделирования процесса плавления системы Джугутова. Сохранить конфигурацию моделируемой системы при температурах: $T = 0.2 \epsilon/k_B$, $T = 0.4 \epsilon/k_B$, $T = 0.6 \epsilon/k_B$, $T = 0.8 \epsilon/k_B$, $T = 1.2 \epsilon/k_B$, $T = 1.5 \epsilon/k_B$, $T = 2.0 \epsilon/k_B$ и $T = 2.5 \epsilon/k_B$.

81. Исследуйте, как изменится характер взаимодействия частиц, если потенциал Леннард-Джонса заменить потенциалом Джугутова. Примечание: рассмотреть трехмерный случай для однородной системы, внутри которой расположены сферические частицы.

82. Исследовать характер взаимодействия атомов, если в потенциале Леннард-Джонса расстояние «обрезания» потенциала увеличить от $r_{cut} = 2.5 \sigma$ до $r_{cut} = 3.0 \sigma$.

83. Смоделировать процесс охлаждения леннард-джонсовской жидкости со скоростью $v_T = 0.05 \epsilon/(k_B\tau)$, приготовленной по алгоритму задачи 76. Систему охладить до температур $T = 0.8 \epsilon/k_B$, $T = 0.4 \epsilon/k_B$ и $T = 0.2 \epsilon/k_B$. При каждой указанной температуре сохранить конфигурацию системы. Исходную конфигурацию загрузить из внешнего (ранее сохраненного) файла. Рассчитать количество временных шагов, необходимых для охлаждения системы. Полученный результат визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

84. Смоделировать процесс охлаждения жидкой системы Джугутова, приготовленной согласно деталям моделирования из задачи 77. Систему необходимо охладить до температур $T = 0.6 \epsilon/k_B$, $T = 0.5 \epsilon/k_B$, $T = 0.4 \epsilon/k_B$ и $T = 0.3 \epsilon/k_B$ со скоростью $v_T = 0.02 \epsilon/(k_B\tau)$. При каждой указанной температуре сохранить конфигурацию системы. Исходную конфигурацию загрузить из внешнего (ранее сохраненного) файла. Рассчитать количество временных шагов, необходимых для охлаждения системы. Полученный результат визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

85. Какими методами можно оценить наличие ближнего и дальнего порядка?

86. Рассчитать функцию радиального распределения частиц для жидкой системы из задачи 76, предварительно приводя систему в

состояние термодинамического равновесия на протяжении времени $t = 100\tau$ с временным шагом $\Delta t = 0.002\tau$. Выполнить анализ полученных результатов. Данные представить в графическом виде с помощью программы Матлаб.

87. Рассчитать функцию радиального распределения частиц для системы из задачи 78, предварительно приводя систему в состояние термодинамического равновесия на протяжении времени $t = 50\tau$ с временным шагом $\Delta t = 0.005\tau$. Выполнить анализ полученных результатов. Данные представить в графическом виде с помощью программы Матлаб.

88. Рассчитать функцию радиального распределения частиц для аморфной системы из задачи 84, предварительно приводя систему в состояние термодинамического равновесия на протяжении времени $t = 200\tau$ с временным шагом $\Delta t = 0.01\tau$. Выполнить анализ полученных результатов. Данные представить в графическом виде с помощью программы Матлаб.

89. Выполнить моделирование процесса плавления системы Леннард-Джонса, согласно условиям задачи 76, воспользовавшись алгоритмом «список Верле». Сравнить скорость вычислений с задачами, где алгоритм «список Верле» не используется.

90. Выполнить моделирование процесса охлаждения жидкой системы Джугутова, согласно условиям задачи 84. Воспользоватьсяся алгоритмом «список Верле» для ускорения вычислений при расчете сил межчастичного взаимодействия. Сравнить скорость вычислений со случаем, где алгоритм «список Верле» не используется.

91. Смоделировать процесс сжатия леннард-джонсовской системы, получаемой согласно условиям задачи 78, при постоянном внешнем давлении $P = 10\epsilon/\sigma^3$. Воспользоваться термостатом и баростатом Нозе-Гувера. Для полученной конфигурации рассчитать функцию радиального распределения частиц. Процесс сжатия системы визуализировать с помощью программы VMD. Выполнить анализ полученных результатов.

92. Выполнить моделирование процесса сжатия системы Джугутова при постоянном внешнем давлении $P = 3.5\epsilon/\sigma^3$ (согласно условиям задачи 77). Воспользоваться термостатом и баростатом Нозе-Гувера. Для полученной конфигурации рассчитать функцию радиального распределения частиц. Процесс сжатия системы визуализи-

ровать с помощью программы VMD. Выполнить анализ полученных результатов.

93. Выполнить моделирование леннард-джонсовской системы, состоящей из $N = 1372$ частиц, находящихся внутри кубической ячейки с периодическими граничными условиями. Использовать NPT-ансамбль с термостатом и баростатом Нозе-Гувера. Уравнения движения интегрируются с помощью скоростного алгоритма Верле с временным шагом $\Delta t = 0.005 \tau$. Рассмотреть следующие точки фазовой диаграммы:

- а) $T = 1.0 \epsilon/k_B$, $\rho = 0.2 \sigma^{-3}$;
- б) $T = 1.5 \epsilon/k_B$, $\rho = 0.8 \sigma^{-3}$;
- в) $T = 2.0 \epsilon/k_B$, $\rho = 0.4 \sigma^{-3}$;
- г) $T = 1.6 \epsilon/k_B$, $\rho = 1.2 \sigma^{-3}$.

94. Рассчитать эволюционные кривые потенциальной и кинетической энергий системы при плавлении леннард-джонсовской системы, состоящей из 2048 частиц, изначально заданной в виде кристаллической ГЦК решетки с количественной плотностью $\rho = 1.0 \sigma^{-3}$. Система плавится в NVT-ансамбле, при скорости нагревания $v_T = 0.04 \epsilon/(k_B \tau)$ до температуры $T_{ext} = 2.5 \epsilon/k_B$. Использовать временный шаг $\Delta t = 0.002 \tau$. Полученные результаты сохранить в текстовый файл и представить в графической виде с помощью программы Matlab.

95. Рассчитать временную зависимость температуры и внутреннего давления системы при охлаждении системы Джугутова с количеством частиц 1372, изначально заданной в виде равновесной жидкости при температуре $T = 2.5 \epsilon/k_B$ и при количественной плотности $\rho = 0.85 \sigma^{-3}$. Охлаждение происходит в NVT-ансамбле со скоростью $0.02 \epsilon/(k_B \tau)$ до температуры $0.5 \epsilon/k_B$. Использовать временный шаг $\Delta t = 0.002 \tau$. Полученные результаты сохранить во внешнем файле и представить в графической виде с помощью программы Matlab.

Приложение

Дополнительная литература

Рапапорт Д.К. *Искусство молекулярной динамики* / Д.К. Рапапорт; пер. с англ. А.Н. Дьяконовой под науч. ред. Р.Г. Ефремова. - Москва, Ижевск: НИЦ «Регулярная и хаотическая динамика», 2012. - 632 с.

Солонина А.И. Цифровая обработка сигналов. Моделирование в MATLAB / А.И. Солонина, С.М. Арбузов. — СПб.: БХВ-Петербург, 2008. — 814 с.

Иглин, С.П. Математические расчеты на базе MATLAB / С.П. Иглин. - СПб.: БХВ-Петербург, 2005. - 634 с.

Мироновский Л.А. *Введение в MATLAB*: учебное пособие / Л.А. Мироновский, К.Ю. Петрова. - СПб.: ГУАП, 2005. - 122 с.

Лазарев Ю.Ф. *Начала программирования в среде MatLAB*: Учебное пособие / Ю.Ф. Лазарев. - К.: НТУУ «КПИ», 2003. - 424 с.

Смоленцев Н.К. *MATLAB: Программирование на Visual C#, Borland C#, JBuilder, VBA*: учебный курс / Н.К. Смоленцев. - М.: ДМК Пресс; Спб.: Питер, 2008. - 464 с.

Френкель Д. *Принципы компьютерного моделирования молекулярных систем: от алгоритмов к приложениям* / Д. Френкель, Б. Смит; пер. с англ. и науч. ред. В.А. Иванова, М.Р. Стуканова. - М.: Научный мир, 2013. - 578 с.

Албахари Д. *C# 5.0. Справочник. Полное описание языка = C# 5.0 in a Nutshell: The Definitive Reference* / Д. Албахари, Б. Албахари. — М.: «Вильямс», 2013. — 1008 с.

Стиллмен Э. *Изучаем C#. 2-е издание = Head First C#, 2ed* / Э. Стиллмен, Дж. Грин. — СПб.: «Питер», 2012. — 704 с.

Хейлсберг А. *Язык программирования C#. Классика Computers Science. 4-е издание = C# Programming Language (Covering C# 4.0)*,

4th Ed / A. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд. — СПб.: «Питер», 2012. — 784 с.

Ландау Л.Д. *Статистическая физика. Часть 2. Теория конденсированного состояния.* / Л.Д. Ландау, П.П. Питаевский. - М.: Физматлит, 2004. - 496 с.

Ландау Л.Д. *Статистическая физика. Часть 1: Учебное пособие для вузов / Л.Д. Ландау, Е.М. Лифшиц.* — М.: Физматлит, 2010. — 616 с.

Полезные Интернет-ресурсы

Программа визуализации результатов молекулярно-динамического моделирования VMD (Visual Molecular Dynamics):

<http://www.ks.uiuc.edu/Research/vmd>

LAMMPS – программный пакет по моделированию молекулярной динамики:

<http://lammps.sandia.gov>

Интернет страница компании MathWorks по программному пакету MatLab и Simulink:

<http://matlab.ru>

Электронный сервис, где можно найти научную литературу по широким предметным областям:

<http://arxiv.org>

Электронно-библиотечная система Знаниум:

<http://znanium.com>

Литература

- [1] Allen M.P. *Computer simulation of liquids* / M.P. Allen and D.J. Tildesley. - Oxford: Clarendon Press, 1987. - 385 p.
- [2] Frenkel D. *Understanding Molecular Simulation* / D. Frenkel and B. Smit. - San Diego: Academic Press, 2001. - 638 p.
- [3] Car R. *Unified Approach for Molecular Dynamics and Density-Functional Theory* / R. Car, M. Parrinello // Phys. Rev. Lett. - 1985. - Vol. 55, P. 2471.
- [4] Рапапорт Д.К. *Искусство молекулярной динамики* / Д.К. Рапапорт; пер. с англ. А.Н. Дьяконовой под науч. ред. Р.Г. Ефремова. - Москва, Ижевск: НИЦ «Регулярная и хаотическая динамика», 2012. - 632 с.
- [5] Hansen J.P. *Theory of Simple Liquids* / J.P. Hansen, I.R. McDonald. - London: Academic Press, 2006. - 416 p.
- [6] Rahman A. *Correlations in the motion of atoms in liquid argon* / A. Rahman // Phys. Rev. A. - 1964. - Vol. 136, P. A405.
- [7] Verlet L. *Computer «experiments» on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules* / L. Verlet // Phys. Rev. - 1967. - Vol. 159, P. 98.
- [8] Woodcock L.V. *Isothermal Molecular Dynamics Calculations for Liquid Salts* / L.V. Woodcock // Chem. Phys. Lett. - 1971. - Vol. 10, P. 257.
- [9] Berendsen H.J.C. *Intermolecular forces* / H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, J. Hermans. - Dordrecht: Riedel, 1981. - 330 p.
- [10] Berendsen H.J.C. *Molecular Dynamics with Coupling to an External Bath* / H.J.C. Berendsen, J.P.M. Postma, W.F. van

- Gunsteren, A.D. Nola and J.R. Haak // *J. Chem. Phys.* - 1984. - Vol. 81, P. 3684.
- [11] Nose S. *A unified formulation of the constant temperature molecular dynamics methods* / S. Nose // *J. Chem. Phys.* - 1984. Vol. 81, P. 511.
- [12] Hoover W.G. *Canonical dynamics: Equilibrium phase-space distributions* / W.G. Hoover // *Phys. Rev. A.* - 1985. - Vol. 31, P. 1695.
- [13] Hinchliffe A. *Molecular Modelling for Beginner* / A. Hinchliffe. - UMIST, Manchester, UK, 2003. - 428 p.
- [14] Ercolessi F. *Interatomic potentials from first-principles calculations: the force-matching method* / F. Ercolessi and J.B. Adams // *Europhys. Lett.* - 1994. - Vol. 26, P. 583.
- [15] Lennard-Jones J.E. *On the Determination of Molecular Fields* / J.E. Lennard-Jones // *Proc. R. Soc. Lond. A.* - 1924. - Vol. 106, P. 463.
- [16] Hansen J.-P. *Phase transitions of the Lennard-Jones system* / J.-P. Hansen, L. Verlet // *Phys. Rev.* - 1969. - Vol. 184, P. 151.
- [17] Гулд Х. *Компьютерное моделирование в физике. Часть 1 и 2* / Х. Гулд, Я. Тобочник. - М.: Мир, 1990. - 350 и 400 с.
- [18] Allen M.P. *Introduction to Molecular Dynamics Simulation* / M.P. Allen. - University of Warwick, Coventry, UK, 2004.
- [19] Alder B.J. *Phase transition for a hard sphere system* / B.J. Alder and T.E. Wainwright // *J. Chem. Phys.* - 1957. - Vol. 27, P. 1208.
- [20] Mairhofer J. *Thermodynamic properties of supercritical n-m Lennard-Jones fluids and isochoric and isobaric heat capacity maxima and minima* / J. Mairhofer and R.J. Sadus // *J. Chem. Phys.* - 2013. - Vol. 139, P. 154503.
- [21] Rowlinson J.S. *Liquid and Liquid Mixtures* / J.S. Rowlinson. - Butterworths: London, 1982. - 328 p.
- [22] Kob W. *Testing mode-coupling theory for a supercooled binary Lennard-Jones mixture I: The van Hove correlation function* / W. Kob and H.C. Andersen // *Phys. Rev. E*.-1995.-Vol.51,P.4626.

- [23] Wahnström G. *Molecular-dynamics study of a supercooled two-component Lennard-Jones system* / G. Wahnström // Phys. Rev. A. - 1991. - Vol. 44, P. 3752.
- [24] Toxvaerd S. *Stability of supercooled binary liquid mixtures* / S. Toxvaerd, U.R. Pedersen, T.B. Schroder and J.C. Dyre // J. Chem. Phys. - 2009. - Vol. 130, P. 224501.
- [25] Mokshin A.V. *Scaling law for crystal nucleation time in glasses* / A.V. Mokshin and B.N. Galimzyanov // J. Chem. Phys. - 2015. - Vol. 142, P. 104502.
- [26] Kong J. *Combining rules for intermolecular potential parameters. II. Rules for the Lennard-Jones (12-6) potential and the Morse potential* / J. Kong // J. Chem. Phys. - 1973. - Vol. 59, P. 2464.
- [27] Dzugutov M. *Pair potential in liquid lead* / M. Dzugutov, K.-E. Larsson and I. Ebbsjö // Phys. Rev. A.-1988.-Vol.38,P.3609.
- [28] Denton A.R. *Stability of Colloidal Quasicrystals* / A.R. Denton and H. Löwen // Phys. Rev. Lett. - 1998. - Vol. 81, P. 469.
- [29] Roth J. *Shock waves in materials with Dzugutov-potential interactions* / J. Roth // Phys. Rev. B. - 2005. - Vol. 72, P. 014125.
- [30] Stillinger F. *Computer simulation of local order in condensed phases of silicon* / F. Stillinger and T.A. Weber // Phys. Rev. B. - 1985. - Vol. 31, P. 5262.
- [31] Molinero V. *Water Modeled As an Intermediate Element between Carbon and Silicon* / V. Molinero and E.B. Moore // J. Phys. Chem. B. - 2009. - Vol. 113, P. 4008.
- [32] Mokshin A.V. *Steady-State Homogeneous Nucleation and Growth of Water Droplets: Extended Numerical Treatment* / A.V. Mokshin, B.N. Galimzyanov // J. Phys. Chem. B. - 2012. - Vol. 116, P. 11959.
- [33] Humphrey W. *VMD: Visual molecular dynamics* / W. Humphrey, A. Dalke, K. Schulten // J. Mol. Graph. - 1996. - Vol. 14, P. 33.
- [34] March N.H. *Atomic dynamics in liquids* / N.H. March, M.P. Tosi. - Dover: New-York, 1991. - 296 p.