

Содержание

1 Введение	3
2 Общее описание моделируемой системы	4
2.1 Модель идеальной стенки	4
2.2 Модель идеальной жёсткой сферы	4
2.3 Периодические граничные условия	6
3 Расчёт столкновения идеальных жёстких сфер	10
3.1 Расчёт времени до соударения двух идеальных жёстких сфер	10
3.2 Расчёт новых скоростей идеальных жёстких сфер после соударения	10
4 Начальное состояние системы. Посев	14
4.1 Посев	14
4.2 Расчёт коэффициента расширения системы β	16
5 Динамика моделируемой системы. События	17
5.1 Изменение положения частиц в системе	17
5.2 Типы событий	17
6 Собственные времёна частиц и глобальное время системы	18
6.1 Синхронизация частиц	18
6.2 Поиск ближайшего события для частицы	18
7 Бинарное дерево как структура данных	19
7.1 Бинарное дерево как структура данных	19
7.2 Добавление элемента в бинарное дерево	19
7.3 Удаление элемента из бинарного дерева	19
7.4 Сравнение алгоритмов на основе бинарного дерева и линейного массива	19
7.5 Бинарное дерево для сохранения информации о всех ближайших событиях в системе	19
8 Сохранение и загрузка состояния системы	20
9 Измерение относительной плотности	21
10 Получение данных о структуре системы	22
10.1 Профиль плотности	22
10.2 Исследование структуры выделенного слоя частиц. Разрезы .	22
11 Изменение объёма системы	23
12 Измерение давления	24
13 Измерение химического потенциала	25

14	Описание управляющих команд программы	26
14.1	new	26
14.2	save	27
14.3	load	27
14.4	step	28
14.5	compress	28
14.5.1	<i>compress_left_wall</i>	29
14.5.2	<i>compress_right_wall</i>	30
14.5.3	<i>compress_both_walls</i>	32
14.6	image	32
14.7	profile	32
15	Глоссарий	33
16	Список литературы	34

1 Введение

Данная статья посвящена рассмотрению алгоритма моделирования системы идеальных твёрдых сфер вблизи идеальных стенок для компьютерных экспериментов по изучению процесса кристаллизации жидкости твёрдых сфер вблизи идеальной стенки. В этой статье описывается модель жёстких сфер, модель идеальной стенки и модель используемых периодических граничных условий.

Используемый нами ранее алгоритм [1] был существенно переработан, в частности, был применён ряд оптимизаций [2] [3] [4], а так же были изменены граничные условия.

Описание, приведённое в данной статье, соответствует программе, написанной для моделирования исследуемой нами системы жёстких сфер, и может быть использовано для анализа алгоритма, реализованного в написанной программе.

Программа, реализующая описанный в этой статье алгоритм, написана на языке C++. Существует несколько версий программы, в которых реализация периодических граничных условий и некоторых функций может отличаться от описанного в этой статье алгоритма, в данной статье описывается только последний вариант реализации алгоритма, актуальный на данный момент, а так же приводится краткий обзор используемых ранее алгоритмов и периодических граничных условий с целью исторического обзора пути развития применяемых нами алгоритмов, а так же в целях объяснения причин, по которым мы выбрали именно данные граничные условия и определённую реализацию некоторых алгоритмов (выбор используемых алгоритмов, в основном, продиктован их оптимальностью - предпочтение отдавалось алгоритмам, требующим наименьшего времени для получения точного решения).

В разработке физической модели моделируемой системы и алгоритма, описанного в данной статье и реализованного в программе на языке C++, принимали активное участие:

Вешнев Владимир Петрович

Гераськин Алексей Сергеевич

Нурлыгаянова Марина Николаевна

Нурлыгаянов Тимур Артурович

Данная статья и любая информация, описанная в ней, а так же исходный код программы, реализующий данный алгоритм, может быть использован, изменён или опубликован только с согласия всех описанных лиц.

Во время работы над этой статьёй авторы использовали различную литературу, ссылки на которую будут даны в тексте статьи.

2 Общее описание моделируемой системы

2.1 Модель идеальной стенки

Модель идеальной жёсткой стенки используется для симуляции границы системы, через которую частицы не могут покинуть рассматриваемый объём. Модель идеальной жёсткой стенки широко применяется в различных компьютерных экспериментах и теоретических исследованиях.

Идеальная жёсткая стенка имеет следующие свойства:

- Имеет большую массу, стремящуюся к бесконечности:

$$M_{ideal\ wall} \rightarrow \infty, \quad (1)$$

- Является абсолютно гладкой для всех частиц в системе, т.е. при касании стенки любой частицей в системе возникающая между этой частицей и идеальной стенкой сила трения равна нулю,

- Является атермичной, т.е. не изменяет кинетическую энергию частиц в системе,

- Имеет импульс, равный нулю, т.е. не движется:

$$p_{ideal\ wall} = 0, \quad (2)$$

2.2 Модель идеальной жёсткой сферы

В данном эксперименте мы используем модель идеальной жёсткой сферы для моделирования частиц. Эта модель применяется во многих компьютерных экспериментах [5] [6] [7].

Идеальная жёсткая сфера обладает следующими свойствами:

- Имеет форму идеального равномерного шара, центр масс которого находится в его геометрическом центре [8],

- Не совершает вращательного движения,

- Является абсолютно гладкой для других идеальных жёстких сфер и идеальной стенки (т.е. при соприкосновении с другими жёсткими сферами или идеальной стенкой сила трения F , возникающая между двумя идеальными жёсткими сферами или идеальной жёсткой сферой и идеальной стенкой, равна нулю),

- Сталкивается с другими идеальными жёсткими сферами или идеальной стенкой мгновенно, т.е. промежуток времени Δt , на протяжении которого происходит соударение, стремится к нулю,

- В любой момент времени, если идеальная жёсткая сфера не сталкивается с другой идеальной жёсткой сферой или с идеальной стенкой, эта идеальная жёсткая сфера движется равномерно и прямолинейно,

- Имеет ненулевой размер и занимает объём системы, равный $\frac{4}{3}\pi R^3$, где R - радиус идеальной сферы.

- Сталкивается с другими идеальными жёсткими сферами и идеальной стенкой абсолютно упруго. Потенциал взаимодействия между двумя идеальными жёсткими сферами описывается следующей системой уравнений:

$$\begin{cases} p = 0, r > 2R, \\ p = \infty, r \leq 2R, \end{cases} \quad (3)$$

где p - потенциал взаимодействия между двумя идеальными сферами, R - радиус идеальной жёсткой сферы, r - расстояние между центрами идеальных жёстких сфер.

Потенциал взаимодействия идеальной жёсткой сферы с идеальными стенками, которые ограничивают рассматриваемую нами систему, можно описать системой уравнений

$$\begin{cases} p = 0, r > R, \\ p = \infty, r \leq R, \end{cases} \quad (4)$$

где p - потенциал взаимодействия между идеальной жёсткой сферой и идеальной твёрдой стенкой, R - радиус идеальной жёсткой сферы, r - расстояние между центром идеальной жёсткой сферы и идеальной твёрдой стенкой.

Потенциал взаимодействия между двумя жёсткими сферами или идеальной стенкой и жёсткой сферой, равный ∞ , накладывает условие невозможности такого состояния системы, в котором расстояние между центрами двух идеальных жёстких сфер было бы меньше $2R$, а расстояние центра идеальной жёсткой сферы и идеальной стенкой было бы меньше R .

Идеальная жёсткая сфера соударяется с идеальной стенкой абсолютно упруго, при этом энергия частицы не изменяется, изменяется лишь проекция скорости частицы на ось OX :

$$v'_x = -v_x, \quad (5)$$

где v_x - проекция скорости частицы на ось OX до соударения с идеальной стенкой, v'_x - проекция скорости частицы на ось OX после соударения с идеальной стенкой.

Условие соударения идеальной жёсткой сферы с идеальными стенками в рассматриваемой системе можно записать так:

$$\begin{cases} x = R, v_x < 0.0 \\ or \\ x = L - R, v_x > 0.0, \end{cases} \quad (6)$$

где x - это значение координаты частицы вдоль оси OX , v_x - значение проекции скорости частицы на ось OX , R - радиус идеальной сферы, L - расстояние между двумя идеальными стенками в системе.

Частицы не могут находиться на расстоянии менее R от идеальной стенки и координаты всех частиц в системе в любой момент времени удовлетворяют следующим правилам:

$$\begin{cases} x \geq R \\ x \leq L - R. \end{cases} \quad (7)$$

2.3 Периодические граничные условия

Периодические граничные условия не дают частицам беспрепятственно покидать выделенный объём, позволяя тем самым моделировать бесконечную систему частиц, рассчитывая динамику системы, содержащую только небольшое количество частиц.

Условия, которые накладываются на частицы, находящиеся в моделируемой системе:

1. Центр каждой частицы всегда находится в системе или на её границе. Положения центров каждой частицы в системе в любой момент времени удовлетворяют следующей системе уравнений:

$$\begin{cases} x \geq R, \\ x \leq L - R, \\ 0 \leq y \leq A, \\ 0 \leq z \leq A, \end{cases} \quad (8)$$

где x , y и z - координаты частицы в трехмерной системе координат, R - радиус частицы, представляющей из себя идеальную жесткую сферу, L - расстояние между двумя идеальными стенками и A - это расстояние между двумя противоположными плоскостями, представляющими периодические границы.

2. Частицы могут пересекать границы системы, заданные уравнениями:

$$y = 0 \quad (9)$$

$$y = A \quad (10)$$

$$z = 0 \quad (11)$$

$$z = A \quad (12)$$

И не могут пересекать границы, заданные уравнением:

$$x = 0 \quad (13)$$

$$x = L \quad (14)$$

т.к. эти границы являются идеальными стенками.

При пересечении границы в момент пересечения одной из периодических границ системы центром частицы на одной из других периодических границ создаётся частица с энергией, равной энергии данной частицы и движущаяся согласно тому же уравнению движения, что и исходная частица. Исходная частица, пересекающая границу системы, при этом удаляется.

3. Если для координат центра частицы выполняется одно из следующих условий:

$$0 \leq y \leq 1 \quad (15)$$

$$A - 1 \leq y \leq A \quad (16)$$

$$0 \leq z \leq 1 \quad (17)$$

$$A - 1 \leq z \leq A \quad (18)$$

то существует один "образ"данной частицы, который движется согласно тому же уравнению движения, что и исходная частица. При пересечении центром частицы одной из границ системы "образ"становится обычной частицей, а сама частица уничтожается. При этом энергия частицы не изменяется, меняются лишь координаты центра частицы.

4. Центр "образа"всегда находится вне системы, что позволяет не учитывать "образ"как отдельную частицу. Любой частице в системе может соответствовать либо ни одного "образа"либо только один "образ".

5. При столкновении частицы или ее "образа"с другой частицей, "образом"другой частицей или идеальной стенкой скорость частицы изменяется, как и её уравнение движения. При этом "образ"соответствующий этой частице, уничтожается (если он существовал) и создаётся новый "образ"(если это необходимо - см. пункт №3), который будет двигаться согласно новому уравнению движения.

6. "Образ"может иметь скорость, отличную от скорости самой частицы, при этом он обязательно должен перемещаться согласно тому же уравнению движения, что и исходная частица (по той же линии скорости или по линии скорости, полученной параллельным переносом по ОХ, что будет рассмотрено отдельно). При столкновении "образа"с другой частицей или "образом"другой частицы учитывается скорость исходной частицы, а не её "образа"(при расчёте новых скоростей), что позволяет сохранять общую энергию системы постоянной.

7. При создании "образа"некоторой частицы позиция его центра выбирается так, чтобы он находился на линии скорости исходной частицы, в точке, из которой он достигнет одной из периодических границ системы в то же самое время, когда центр исходной частицы будет находиться в плоскости пересекаемой ею границы.

Рассмотрим подробнее вычисление координат центра создаваемого "образа". Уравнение движения центра сферы по прямолинейной траектории с постоянной скоростью в векторном виде можно записать так:

$$\vec{r} = \vec{r}_0 + \vec{v} * t \quad (19)$$

где \vec{r} - радиус вектор нового положения центра частицы, \vec{r}_0 - радиус вектор начального положения центра частицы, \vec{v} - скорость частицы и t - время движения.

Распишем это выражение в проекциях на оси координат:

$$\begin{cases} r_x = r_{0x} + v_x * t, \\ r_y = r_{0y} + v_y * t, \\ r_z = r_{0z} + v_z * t. \end{cases} \quad (20)$$

из этой системы уравнений получаем:

$$\begin{cases} t = \frac{r_x - r_{0x}}{v_x}, \\ t = \frac{r_y - r_{0y}}{v_y}, \\ t = \frac{r_z - r_{0z}}{v_z}. \end{cases} \quad (21)$$

Полученная система уравнений (21) позволяет рассчитывать время, требующееся для перемещения центра частицы из одной позиции в другую.

Когда центр частицы находится на границе системы, прямая, описывающая уравнение её движения, пересекает плоскость, которой описана эта граница. Рассчитав времена, которые потребуются частице для того, чтобы её центр оказался в плоскости каждой границы, мы можем вычислить границу, на которой центр выбранной частицы окажется быстрее всего. Для этого необходимо решить четыре системы уравнений (одна система уравнений для каждой периодической границы):

$$\begin{cases} y = 0, \\ t = \frac{r_y - r_{0y}}{v_y}. \end{cases} \quad (22)$$

$$\begin{cases} y = A, \\ t = \frac{r_y - r_{0y}}{v_y}. \end{cases} \quad (23)$$

$$\begin{cases} z = 0, \\ t = \frac{r_z - r_{0z}}{v_z}. \end{cases} \quad (24)$$

$$\begin{cases} z = A, \\ t = \frac{r_z - r_{0z}}{v_z}. \end{cases} \quad (25)$$

полагая при решении этих систем уравнений $r_y = y$ и $r_z = z$, мы получим 4 решения для t :

$$\begin{cases} t_1 = \frac{0 - r_{0y}}{v_y}, \\ t_2 = \frac{A - r_{0y}}{v_y}, \\ t_3 = \frac{0 - r_{0z}}{v_z}, \\ t_4 = \frac{A - r_{0z}}{v_z}. \end{cases} \quad (26)$$

где r_{0y} , r_{0z} - проекции радиус вектора текущего положения частицы на оси OY и OZ .

Необходимо найти t_{min} , удовлетворяющее условиям:

$$\begin{cases} t_{min} > 0, \\ t_{min} \leq t_1, \\ t_{min} \leq t_2, \\ t_{min} \leq t_3, \\ t_{min} \leq t_4, \end{cases} \quad (27)$$

где t_{min} есть время, за которое частица достигнет ближайшей границы системы, если будет двигаться с текущей скоростью. В соответствии с условием №7 'образ', соответствующий данной частице, должен за это же время достигнуть другой периодической границы системы.

Для того, чтобы вычислить координаты центра создаваемого "образа" возьмём новые значения v_x , v_y и v_z , равные:

$$\begin{cases} v'_x = -v_x, \\ v'_y = -v_y, \\ v'_z = -v_z. \end{cases} \quad (28)$$

и рассчитаем для них уравнения (26) и (27), подставив полученное в результате решения этих уравнений значение t'_{min} значения v'_x , v'_y и v'_z в систему уравнений (49), получим:

$$\begin{cases} r'_x = r_{0x} + v'_x * t'_{min}, \\ r'_y = r_{0y} + v'_y * t'_{min}, \\ r'_z = r_{0z} + v'_z * t'_{min}. \end{cases} \quad (29)$$

где r'_x , r'_y и r'_z есть значения новых координат центра частицы, которые она будет иметь после прохождения через периодическую границу.

Сложив t_{min} и t'_{min} и подставив получившееся значение t в систему уравнений (49) мы получим значения координат создаваемого образа, соответствующего данной частице:

$$\begin{cases} r''_x = r_{0x} + v'_x * (t_{min} + t'_{min}), \\ r''_y = r_{0y} + v'_y * (t_{min} + t'_{min}), \\ r''_z = r_{0z} + v'_z * (t_{min} + t'_{min}). \end{cases} \quad (30)$$

здесь r''_x , r''_y и r''_z - координаты создаваемого образа, r_{0x} , r_{0y} и r_{0z} - текущие координаты исходной частицы, для которой необходимо создать виртуальную частицу.

3 Расчёт столкновения идеальных жёстких сфер

Необходимо описать подробно.

3.1 Расчёт времени до соударения двух идеальных жёстких сфер

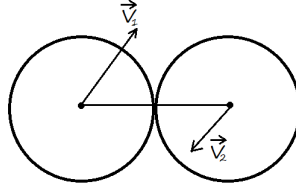
Необходимо описать подробно.

3.2 Расчёт новых скоростей идеальных жёстких сфер после соударения

Рассмотрим общий случай соударения двух частиц, представляющих из себя идеальные жёсткие сферы.

При любом столкновении двух идеальных жёстких сфер центры масс этих сфер, а так же точка соприкосновения поверхностей этих двух сфер лежат в одной плоскости.

Рассмотрим проекцию двух сфер и их скоростей на эту плоскость в момент соударения.



Здесь \vec{v}_1 - скорость первой частицы, \vec{v}_2 - скорость второй частицы.

Так как рассматриваемые сферы являются равномерными идеальными сферами и точка касания двух сфер находится на одной линии с центрами масс обеих сфер, то и силы, действующие со стороны первой частицы на вторую и со стороны второй частицы на первую будут направлены по линии, соединяющей центры масс сфер.

Идеальные сферы являются абсолютно гладкими и не могут совершать вращательное движение, поэтому силы, действующие на первую и вторую частицы, действуют только вдоль линии, соединяющей центры масс данных частиц.

Изобразим действующие на обе идеальные сферы силы:

Согласно второму закону Ньютона должно выполняться равенство:

$$\vec{F}_1 = -\vec{F}_2 \quad (31)$$

Распишем скорость \vec{v}_1 как сумму $\vec{v}_{1\perp}$ и $\vec{v}_{1\parallel}$ и скорость \vec{v}_2 как сумму $\vec{v}_{2\perp}$ и $\vec{v}_{2\parallel}$, где $\vec{v}_{1\perp}$ и $\vec{v}_{2\perp}$ составляющие скоростей \vec{v}_1 и \vec{v}_2 , перпендикулярные линии, соединяющей центры сталкивающихся частиц, $\vec{v}_{1\parallel}$ и $\vec{v}_{2\parallel}$ - параллельные

составляющие скоростей \vec{v}_1 и \vec{v}_2 относительно линии, соединяющей центры частиц.

Учитывая, что силы, действующие на первую и вторую частицы в момент их соударения, действуют только вдоль прямой, соединяющей центры масс данных частиц, можно утверждать, что составляющие скоростей движения частиц, перпендикулярные линии, соединяющей центры этих частиц, не влияют на величину и направление сил \vec{F}_1 и \vec{F}_2 .

Поэтому учитывая, что масса идеальных жёстких сфер постоянна, мы можем записать:

$$\left\{ \begin{array}{l} \vec{F}_1 = \frac{m_1 * d\vec{v}_{1\parallel}}{dt}, \\ \frac{m_1 * d\vec{v}_{1\perp}}{dt} = 0, \\ \vec{F}_2 = \frac{m_2 * d\vec{v}_{2\parallel}}{dt}, \\ \frac{m_2 * d\vec{v}_{2\perp}}{dt} = 0, \end{array} \right. \quad (32)$$

Учитывая второй закон Ньютона для данной системы двух идеальных твёрдых сфер (31), а так же то, что массы всех идеальных твёрдых сфер равны, мы можем записать (32) в виде:

$$\left\{ \begin{array}{l} \frac{d\vec{v}_{2\parallel}}{dt} = -\frac{d\vec{v}_{1\parallel}}{dt}, \\ \vec{v}_{1\perp} = const, \\ \vec{v}_{2\perp} = const, \end{array} \right. \quad (33)$$

Идеальные жёсткие сферы соударяются абсолютно упруго, а значит при соударении двух идеальных жёстких сфер должны выполняться закон сохранения импульса и закон сохранения энергии.

Закон сохранения импульса для рассматриваемой системы двух частиц может быть записан как

$$\left\{ \begin{array}{l} \vec{v}_{1\perp} + \vec{v}_{2\perp} = \vec{v}'_{1\perp} + \vec{v}'_{2\perp}, \\ \vec{v}_{1\parallel} + \vec{v}_{2\parallel} = \vec{v}'_{1\parallel} + \vec{v}'_{2\parallel}, \end{array} \right. \quad (34)$$

где $\vec{v}'_{1\perp}$, $\vec{v}'_{1\parallel}$, $\vec{v}'_{2\perp}$, $\vec{v}'_{2\parallel}$ - скорости частиц после соударения.

Запишем закон сохранения энергии для данной системы из двух сталкивающихся идеальных сфер:

$$\begin{cases} E_1 = \frac{mv_{1\perp}^2}{2} + \frac{mv_{1\parallel}^2}{2} + \frac{mv_{2\perp}^2}{2} + \frac{mv_{2\parallel}^2}{2}, \\ E_2 = \frac{mv_{1\perp}'^2}{2} + \frac{mv_{1\parallel}'^2}{2} + \frac{mv_{2\perp}'^2}{2} + \frac{mv_{2\parallel}'^2}{2}, \\ E_1 = E_2, \end{cases} \quad (35)$$

где E_1 - полная кинетическая энергия системы из двух частиц перед соударением, E_2 - полная кинетическая энергия системы из двух частиц после соударения. Потенциальная энергия частиц при соударении не изменяется, т.к. происходит абсолютно упругий удар.

Учитывая (33) мы можем записать, что $\vec{v}_{1\perp} = \vec{v}_{1\perp}'$ и $\vec{v}_{1\perp} = \vec{v}_{1\perp}'$, тогда учитывая закон сохранения импульса (34) и закон сохранения энергии (35) мы можем записать следующую систему уравнений:

$$\begin{cases} \vec{v}_{1\parallel} + \vec{v}_{2\parallel} = \vec{v}_{1\parallel}' + \vec{v}_{2\parallel}', \\ \frac{mv_{1\parallel}^2}{2} + \frac{mv_{2\parallel}^2}{2} = \frac{mv_{1\parallel}'^2}{2} + \frac{mv_{2\parallel}'^2}{2} \end{cases} \quad (36)$$

Сразу упростим второе уравнение, учтя равенство масс двух идеальных жёстких сфер:

$$\begin{cases} \vec{v}_{1\parallel} + \vec{v}_{2\parallel} = \vec{v}_{1\parallel}' + \vec{v}_{2\parallel}', \\ v_{1\parallel}^2 + v_{2\parallel}^2 = v_{1\parallel}'^2 + v_{2\parallel}'^2 \end{cases} \quad (37)$$

Преобразуем первое уравнение в (37), домножив обе части уравнения на $\vec{v}_{1\parallel}$:

$$\vec{v}_{1\parallel}^2 + \vec{v}_{2\parallel}\vec{v}_{1\parallel} = \vec{v}_{1\parallel}'\vec{v}_{1\parallel} + \vec{v}_{2\parallel}'\vec{v}_{1\parallel} \quad (38)$$

Теперь выразим из этого уравнения $\vec{v}_{1\parallel}^2$ и подставим получившееся выражение во второе уравнение системы уравнений (37), получим:

$$\vec{v}_{1\parallel}(\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}) + \vec{v}_{2\parallel}^2 = \vec{v}_{1\parallel}'^2 + \vec{v}_{2\parallel}'^2 \quad (39)$$

и выразим из полученного уравнения $\vec{v}_{1\parallel}$:

$$\vec{v}_{1\parallel} = \frac{\vec{v}_{1\parallel}'^2 + \vec{v}_{2\parallel}'^2 - \vec{v}_{2\parallel}^2}{\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}} \quad (40)$$

подставим результат в первое уравнение из (37), получим:

$$\frac{\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}}{\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}} + \vec{v}_{2\parallel} = \vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' \quad (41)$$

что можно записать как:

$$\vec{v}_{1\parallel}'^2 + \vec{v}_{2\parallel}'^2 - \vec{v}_{2\parallel}^2 = (\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}) * (\vec{v}_{1\parallel}' + \vec{v}_{2\parallel}' - \vec{v}_{2\parallel}) \quad (42)$$

после раскрытия скобок во второй части равенства и приведения подобных слагаемых мы получаем:

$$-2\vec{v}_{2\parallel}^2 = 2\vec{v}_{1\parallel}'\vec{v}_{2\parallel}' - 2\vec{v}_{1\parallel}'\vec{v}_{2\parallel} - 2\vec{v}_{2\parallel}'\vec{v}_{2\parallel} \quad (43)$$

сократив обе части равенства на 2 и вынеся за скобку $\vec{v}_{1\parallel}'$ мы получаем:

$$\vec{v}_{1\parallel}'(\vec{v}_{2\parallel}' - \vec{v}_{2\parallel}) - \vec{v}_{2\parallel}'\vec{v}_{2\parallel} = -\vec{v}_{2\parallel}^2 \quad (44)$$

Выразим $\vec{v}_{1\parallel}'$ из (44) мы можем записать:

$$\vec{v}_{1\parallel}' = \frac{\vec{v}_{2\parallel}'\vec{v}_{2\parallel}' - \vec{v}_{2\parallel}^2}{\vec{v}_{2\parallel}' - \vec{v}_{2\parallel}} \quad (45)$$

и вынеся за скобку $\vec{v}_{2\parallel}'$, запишем это выражение как

$$\vec{v}_{1\parallel}' = \frac{\vec{v}_{2\parallel}'(\vec{v}_{2\parallel}' - \vec{v}_{2\parallel})}{\vec{v}_{2\parallel}' - \vec{v}_{2\parallel}} \quad (46)$$

Если мы сократим в (46) числитель и знаменатель дроби на $\vec{v}_{2\parallel}' - \vec{v}_{2\parallel}$ то получим конечное выражение для $\vec{v}_{1\parallel}'$:

$$\vec{v}_{1\parallel}' = \vec{v}_{2\parallel} \quad (47)$$

что с учётом (33) даёт нам право записать конечную систему уравнений, определяющую выражения для новых скоростей идеальных сфер после соударения:

$$\begin{cases} \vec{v}_{1\parallel}' = \vec{v}_{2\parallel}, \\ \vec{v}_{2\parallel}' = \vec{v}_{1\parallel}, \\ \vec{v}_{1\perp}' = \vec{v}_{1\perp}, \\ \vec{v}_{2\perp}' = \vec{v}_{2\perp}. \end{cases} \quad (48)$$

4 Начальное состояние системы. Посев

4.1 Посев

Для начала моделирования необходимо создать N жёстких сфер в моделируемой системе таким образом, чтобы центры всех частиц находились в исследуемом объёме, частицы не проникали друг в друга, а так же чтобы начальный импульс системы и момент импульса системы по осям OX , OY и OZ были равны нулю.

Процедура инициализации начальных параметров объёма, положений частиц и их скоростей называется "посев". В дальнейшем мы будем использовать этот термин для обозначения функции программы, которая позволяет произвести начальную инициализацию моделируемой системы.

Начальная плотность данной системы η_0 будет определяться задаваемым параметрами объёма исследуемой системы и количеством частиц в данном объёме (т.к. радиус идеальных жёстких сфер R по умолчанию равен 1.0, мы полагаем его постоянным и одинаковым для всех частиц).

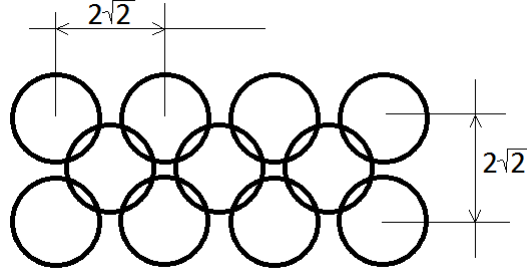
Изначально центры частиц размещаются в системе в "узлах" объёмноцентрированной кристаллической решётки, расстояние между частицами регулируется начальной плотностью системы, но при этом расстояние между двумя частицами не может быть меньше, чем два радиуса идеальных сфер (чтобы частицы не проникали друг в друга).

Расчёт начальных координат и скоростей идеальных сфер производится в несколько этапов, рассмотрим их более подробно:

1. Создаётся "кристалл" жёстких сфер с объёмноцентрированной кристаллической решёткой, содержащий M частиц. Частицы в данном кристалле касаются друг друга и образуют прямоугольный параллелепипед, рёбра которого по Y и Z равны между собой, при этом рёбра по X примерно в два раза больше рёбер по Y и Z . Это продиктовано особенностями моделируемой системы, т.к. нам необходимо задавать длину системы L в четыре раза больше, чем размер системы по Y и Z , чтобы иметь возможность исследовать свойства кристалла вблизи идеальной стенки, исключая при этом влияние второй, противоположной идеальной стенки, которая будет достаточно удалена от исследуемой части системы.

Один из углов данного параллелепипеда будет совпадать с началом координат.

Расстояние между двумя одинаковыми слоями объёмноцентрированного кристалла в случае плотного расположения слоёв равно $2 * \sqrt{2}$:



Координаты центра данного прямоугольного параллелепипеда можно рассчитать по формулам:

$$\begin{cases} y = \frac{r_b * (NN - 1.0) + 2.0}{2.0}, \\ z = \frac{r_b * (NN - 1.0) + 2.0}{2.0}, \\ x = \frac{r_b * (2.0 * NN - 1.0) + 2.0}{2.0}. \end{cases} \quad (49)$$

где r_b - это расстояние между одинаковыми слоями объёмноцентрированного кристалла, равное $2 * \sqrt{2}$, NN - количество частиц в ребре куба по Y и Z , это число задаётся при новом "посеве" частиц (см. главу "Описание управляющих команд программы").

2. Созданный кристалл копируется несколько раз так, чтобы увеличить кристалл вдвое по трём измерениям (x, y, z), таким образом, делается 7 копий созданного на шаге №1 кристалла, которые располагаются вплотную к первоначальному кристаллу так, чтобы частицы, находящиеся на границах параллелепипедов, касались друг друга.

В результате получается кристалл жёстких сфер в форме прямоугольного параллелепипеда, один угол которого совпадает с началом координат и одна сторона которого в два раза больше двух других сторон.

3. Для частиц, находящихся в кристалле, созданном на шаге №1, случайным образом задаются скорости v_x , v_y и v_z , при этом скорости остальных частиц в системе задаются так, чтобы суммарный импульс и момент импульса системы были равны нулю.

4. Основываясь на задаваемой плотности системы рассчитываются параметры моделируемого объёма A и L , а так же коэффициент расширения β (мы рассмотрим подробнее вывод коэффициента расширения системы чуть ниже), на который умножаются все координаты частиц, благодаря чему объёмноцентрированный кристалл жёстких сфер расширяется до размеров моделируемой системы и частицы перестают касаться друг друга. При этом суммарный импульс и момент импульса системы остаются равными нулю, т.к. импульсы одних частиц уравновешены импульсами других частиц, расположенных симметрично относительно центра моделируемой системы.

5. Все частицы копируются ещё раз и их координаты изменяются параллельным переносом вдоль оси OX так, чтобы в результате частицы заполнили весь объём системы, заданный из расчёта $L = 4 * A$. Скорости частиц не изменяются, поэтому общий импульс и момент импульса системы при этом так же не изменяются. Таким образом размер системы вдоль оси OX получается в четыре раза больше размера системы вдоль осей OY и OZ , центры идеальных сфер распределены по всему объёму моделируемой системы так, чтобы частицы не проникали друг в друга и суммарный импульс системы и момент импульса системы были равны нулю.

После проведения "посева" состояние системы сохраняется во временный файл и загружается из него. Во время загрузки сохранённого состояния системы мы рассчитываем ближайшие события для каждой частицы и начинаем расчёт динамики системы. Более подробно это описано в главе "Сохранение и загрузка состояния системы".

4.2 Расчёт коэффициента расширения системы β

Рассмотрим подробнее вывод коэффициента расширения системы β .

Необходимо описать подробно.

5 Динамика моделируемой системы. События

Необходимо описать подробно.

5.1 Изменение положения частиц в системе

Необходимо описать подробно.

5.2 Типы событий

В рассматриваемой нами системе могут происходить следующие события:

1. Соударение частицы или её "образа" с идеальной стенкой.
2. Соударение частицы или её "образа" с другой частицей или "образом" другой частицы.
3. Создание "образа" для частицы, находящейся в области, близкой к периодическим граничным условиям.
4. Переход частицы из одной "ячейки" в другую.
5. Замена "образа" частицей - "перерождение" частицы при прохождении через периодические граничные условия.
6. Удаление "образа" частицы.

Необходимо описать подробно.

6 Собственные времёна частиц и глобальное время системы

Необходимо описать подробно.

6.1 Синхронизация частиц

Необходимо описать подробно.

6.2 Поиск ближайшего события для частицы

Необходимо описать подробно.

7 Бинарное дерево как структура данных

Бинарное (двоичное) дерево - структура данных, хорошо известная в информатике. Позволяет сохранять упорядоченные по ключу данные так, что операции добавления и удаления элементов из такой структуры данных имеют алгоритмическую сложность $\log_2 N$, в то время как удаление произвольного элемента из упорядоченного списка имеет алгоритмическую сложность N , а добавление нового элемента в упорядоченный список имеет алгоритмическую сложность $N * \log_2 N$ (поиск позиции и вставка элемента с перестановкой всех элементов "вправо").

Необходимо описать подробно.

7.1 Бинарное дерево как структура данных

Необходимо описать подробно.

7.2 Добавление элемента в бинарное дерево

Необходимо описать подробно.

7.3 Удаление элемента из бинарного дерева

Необходимо описать подробно.

7.4 Сравнение алгоритмов на основе бинарного дерева и линейного массива

Необходимо описать подробно.

7.5 Бинарное дерево для сохранения информации о всех ближайших событиях в системе

Необходимо описать подробно.

8 Сохранение и загрузка состояния системы

Необходимо описать подробно.

9 Измерение относительной плотности

Необходимо описать подробно.

10 Получение данных о структуре системы

Необходимо описать подробно.

10.1 Профиль плотности

Необходимо описать подробно.

10.2 Исследование структуры выделенного слоя частиц. Разрезы

Необходимо описать подробно.

11 Изменение объёма системы

Необходимо описать подробно.

12 Измерение давления

Данная часть алгоритмов и программы ещё не реализовано, необходимо добавить описание того, как мы будем производить измерения данных и описать реализацию алгоритма в программе.

13 Измерение химического потенциала

Данная часть алгоритмов и программы ещё не реализовано, необходимо добавить описание того, как мы будем производить измерения данных и описать реализацию алгоритма в программе.

14 Описание управляющих команд программы

Для управления ходом компьютерного эксперимента необходимо использовать управляющие команды, которые должны быть последовательно описаны в текстовом файле *program.txt* в директории с запускаемой программой. Программа последовательно считывает управляющие команды и файла *program.txt* и так же последовательно выполняет их, выводя информацию о текущей задаче на экран.

Все команды имеют строго определённый синтаксис, опечатки и неправильная последовательность аргументов, пустые строки и некорректные символы в файле описания хода эксперимента могут привести к аварийному завершению программы, сохранению данных в неправильные файлы и даже перезапись уже существующих экспериментальных данных (если в программе эксперимента будет указано, что нам необходимо сохранить некоторые данные в уже существующий файл). Необходимо аккуратно заполнять файл описания эксперимента и проверять его содержимое до запуска эксперимента.

Все функции, которые вызываются для выполнения команд, детально описаны отдельно в различных главах данной статьи.

Рассмотрим подробнее каждую управляющую команду, параметры для различных команд и примеры использования.

14.1 new

new - команда, позволяющая запустить процедуру инициализации начального состояния системы (см. главу Посев).

Аргументы:

- Количество частиц в ребре куба объёмноцентрированного кристалла, который будет использоваться во время посева. От числа частиц в ребре этого куба зависит конечное число частиц в моделируемой системе. Например, если число частиц в ребре равно 2, то всего частиц в системе будет 64, если число частиц в ребре равно 8, то частиц в системе будет 6976.

- Относительная плотность частиц в системе, которую необходимо задать при инициализации начального состояния. Данная плотность не может быть больше 0.68 (максимальная относительная плотность кристалла с объёмноцентрированной кристаллической решёткой).

Пример:

new 8 0.4

Данная команда создаст новую систему из 6976 частиц, начальная относительная плотность частиц в системе η будет равна 0.4. После инициализации системы так же будут рассчитаны времена ближайших событий для каждой частицы.

После выполнения этой команды мы можем использовать другие управляющие команды, такие, как *step*, *compress*, *profile* и т.д.

14.2 save

Для сохранения состояния системы используется команда *save*. При сохранении состояния системы в указанный нами текстовый файл будут сохранены число частиц в системе, число "образов" частиц, существующих на данный момент в системе, параметры объёма A и L моделируемой системы а так же координаты и скорости каждой частицы и всех существующих на данный момент времени в системе "образов" частиц.

В дальнейшем сохранённое состояние системы можно будет загрузить с помощью команды *load*.

Аргументы:

- Имя текстового файла, в который необходимо сохранить данные о состоянии системы. Если этот файл не существует, он будет создан в директории с программой или по указанному пути, если указать имя файла с полным путём к нему. Если такой файл уже существует, то существующий файл будет перезаписан новыми данными.

Пример:

```
save save_file_0.40000_10000.txt
```

Данная команда создаст файл *save_file_0.40000_10000.txt* в папке с программой и запишет в этот файл информацию о текущем состоянии системы.

Рекомендуем в имени файла, содержащего информацию о системе, добавлять информацию о текущей плотности системы а так же о количестве соударений, прошедших в систем с момента последнего изменения плотности системы. Это поможет в дальнейшем быстро найти необходимое состояние в системе и вспомнить как именно данное состояние было получено.

14.3 load

Для загрузки некоторого ранее сохранённого состояния моделируемой системы из текстового файла используется команда *load*. Данная команда обнуляет текущее состояние системы, загружает данные о состоянии системы из текстового файла и запускает перерасчёт времён ближайших событий для каждой частицы и каждого "образа" в системе.

Для корректной работы данной команды необходимо существование файла с информацией о некотором состоянии моделируемой системы, синтаксис такого файла строго определён и не должен подвергаться ручному изменению, такие файлы создаются автоматически при сохранении некоторого состояния системы с помощью команды *save*.

Аргументы:

- Имя текстового файла, из которого необходимо загрузить информацию о некотором, сохранённом ранее, состоянии системы. Данный файл должен существовать и быть доступным для чтения запускаемой программой, он должен иметь строго определённую структуру и синтаксис. В случае, если указанный файл не существует, программа будет аварийно завершена.

Пример:

```
load save_file_0.40000_10000.txt
```

Данная команда загрузит состояние системы из файла *save_file_0.40000_10000.txt* и произведёт расчёт времен ближайших событий для всех частиц и "образов" в системе.

После выполнения этой команды мы можем использовать другие управляющие команды, такие, как *step*, *compress*, *profile* и т.д.

14.4 step

Управляющая команда *step* позволяет запустить моделирование динамики системы частиц.

Аргументы:

- Количество соударений в системе на одну частицу, в течении которых необходимо рассчитать динамику моделируемой системы. Этот параметр должен быть натуральным числом.

Пример:

```
step 1000
```

Данная команда просчитает динамику системы начиная от последнего состояния системы и до тех пор, пока в системе не произойдёт $500 * N$ соударений между частицами, где N - количество частиц в системе. Так как в каждом соударении частиц участвует одновременно две частицы, можно считать что при этом в среднем в системе произошло 1000 соударений для каждой частицы, хотя на самом деле для каждой отдельной частицы число соударений будет разным.

В логику расчёта динамики системы частиц входит множество различных операций, все они подробно описаны в данной статье, а в этой главе мы ограничимся описанием управляющей команды *step*, которая позволяет управлять продолжительностью моделирования динамики этой системы.

Примечание:

Мы можем использовать команду *step* несколько раз подряд, при этом состояние системы каждый раз не будет сбрасываться и возвращаться в некоторое исходное состояние, команда *step* будет использовать текущее состояние системы как исходное. Например, если мы напишем такой алгоритм:

```
step 1000
```

```
step 10000
```

то программа рассчитает сначала динамику системы в течении тысячи соударений на каждую частицу в системе, а затем рассчитает динамику системы в течении ещё десяти тысяч соударений, начиная с того состояния системы, на котором мы остановились после расчёта тысячи соударений в системе.

14.5 compress

Управляющая команда *compress* позволяет изменять параметры объёма моделируемой системы, за счёт чего изменяется относительная плотность

частиц в системе.

Изменение параметров объёма системы происходит вследствие изменения L (длины системы по оси OX), изменяя длину моделируемой системы мы симулируем движение идеальных стенок, которые ограничивают моделируемую систему в плоскостях $X = 0$ и $X = L$.

Существует три типа изменения плотности системы: *compress_left_wall*, *compress_right_wall*, *compress_both_walls*, которые симулируют, соответственно, движение только левой стенки, движение только правой стенки и движение одновременно двух стенок.

Все три типа процедур изменения параметров системы действуют по примерно одному и тому же алгоритму, который коротко можно описать так:

1. Найти ближайшую к идеальной стенке частицу и определить расстояние Δr от поверхности сферы, ограничивающей частицу, до идеальной стенки.
2. Сдвинуть стенку на расстояние, меньшее Δr и меньшее некоторой заранее определённой ΔL . Важно отметить, что при смещении одной или нескольких идеальных стенок в системе перерасчёт позиций "образов" частиц произведён не будет, т.к. мы можем перемещать образы вдоль оси OX не изменяя при этом момент импульса L_x моделируемой системы.
3. Рассчитать время ближайших событий для всех частиц и образов.
5. Рассчитать динамику в системе на протяжении некоторого, заранее заданного, числа соударений на каждую частицу.
6. Если относительная плотность частиц в системе не соответствует указанной пользователем в параметрах плотности, то вернуться на шаг №1.

Рассмотрим каждый тип изменения параметров системы в отдельности.

14.5.1 *compress_left_wall*

Управляющая команда *compress_left_wall* позволяет изменять параметры моделируемой системы, симулируя движение "левой" идеальной стенки (данная стенка условно называется левой, т.к. на профиле плотности, где представлены данные о значениях относительной плотности в системе, данная граница системы расположена слева, в начале координат).

При этом сама "левая" идеальная стенка не движется, программа производит смещение всех частиц и образов на определённое ΔL в сторону этой стенки вдоль оси OX , после чего уменьшает параметр системы L на ту же величину ΔL , в результате чего идеальная стенка, ограничивающая моделируемую нами систему в плоскости $x = 0$, становится ближе (или дальше, это зависит от того, хотим ли мы увеличить относительную плотность частиц в системе или хотим уменьшить её) ко всем частицам в системе на расстояние ΔL , расстояние между частицами и второй идеальной стенкой, ограничивающей систему в плоскости $x = L$ не изменяется.

Таким образом мы моделируем изменение координаты "левой" идеальной стенки, приближая её ко всем частицам в системе или отдаляя эту границу от них.

Величина ΔL , на которую мы изменяем расстояние между частицами и стенкой не может быть больше расстояния между идеальной стенкой и самой ближней к ней частице, а так же ограничивается значением максимально допустимого смещения стенки, задаваемого в качестве аргумента для данной команды.

Аргументы:

- Относительная плотность системы, которую необходимо установить в системе с помощью движения идеальной стенки. Указываемая относительная плотность частиц в системе может быть как больше, так и меньше текущей плотности в системе, в зависимости от чего с помощью данной команды можно как сжимать моделируемую систему, уменьшая объём, в котором находятся частицы, так и наоборот, расширять данную системы, отодвигая идеальную стенку от частиц и уменьшая относительную плотность системы.

- Значение максимально допустимого смещения стенки ΔL_{max} . Если ближайшая к идеальной стенке частица находится на расстоянии, большем ΔL_{max} , то идеальная стенка может быть смещена только на расстояние ΔL_{max} .

- Число соударений на частицу, которое необходимо рассчитать после каждого маленького изменения плотности моделируемой системы прежде, чем совершить следующее изменение координат идеальной стенки. Данный параметр позволяет задавать интервал изменения плотности системы, что даёт нам возможность изменять плотность на столько медленно, на сколько это требуется условиями эксперимента.

Пример:

```
compress_left_wall 0.503423 0.000001 10
```

Данная команда изменит относительную плотность частиц в системе до значения 0.503423, при этом максимальное расстояние, на которое будет меняться расстояние между частицами и идеальной стенкой будет не более $\Delta L_{max} = 0.000001$ и после каждого небольшого изменения плотности системы будет произведён расчёт динамики движения частиц в системе на протяжении десяти соударений на каждую частицу в этой системе.

Короткая запись команды:

Данная команда имеет вариант короткой записи, которая имеет те же аргументы и работает так же, как и полная запись команды:

```
compressl 0.49 0.503423 0.000001 10
```

14.5.2 *compress_right_wall*

Управляющая команда *compress_right_wall* позволяет изменять параметры моделируемой системы, симулируя движение "правой" идеальной стенки (данная стенка условно называется правой, т.к. на профиле плотности, где представлены данные о значениях относительной плотности в системе, данная граница системы расположена справа).

На каждом шаге изменения параметров моделируемой системы программа определяет наименьшее расстояние Δr поверхности частиц в систе-

ме до идеальной стенки, ограничивающей моделируемую систему в плоскости $x = L$ и изменяет координаты этой идеальной стенки на величину, меньшую Δr . Если расстояние Δr больше, чем задаваемое пользователем максимально допустимое смещение стенки ΔL_{max} , то стенка смещается на расстояние ΔL_{max} .

В результате изменения значения параметра системы L идеальная стенка, ограничивающая моделируемую нами систему в плоскости $x = L$, становится ближе (или дальше, это зависит от того, хотим ли мы увеличить относительную плотность частиц в системе или хотим уменьшить её) ко всем частицам в системе на расстояние ΔL , расстояние между частицами и второй идеальной стенкой, ограничивающей систему в плоскости $x = 0$ не изменяется.

Таким образом мы моделируем изменение координаты "правой" идеальной стенки, приближая её ко всем частицам в системе или отдаляя эту границу от них.

Величина ΔL , на которую мы изменяем расстояние между частицами и стенкой не может быть больше расстояния между идеальной стенкой и самой ближней к ней частице, а так же ограничивается значением максимально допустимого смещения стенки, задаваемого в качестве аргумента для данной команды.

Аргументы:

- Относительная плотность системы, которую необходимо установить в системе с помощью движения идеальной стенки. Указываемая относительная плотность частиц в системе может быть как больше, так и меньше текущей плотности в системе, в зависимости от чего с помощью данной команды можно как сжимать моделируемую систему, уменьшая объём, в котором находятся частицы, так и наоборот, расширять данную системы, отодвигая идеальную стенку от частиц и уменьшая относительную плотность системы.

- Значение максимально допустимого смещения стенки ΔL_{max} . Если ближайшая к идеальной стенке частица находится на расстоянии, большем ΔL_{max} , то идеальная стенка смещается только на расстояние ΔL_{max} .

- Число соударений на частицу, которое необходимо рассчитать после каждого маленького изменения плотности моделируемой системы прежде, чем совершить следующее изменение координат идеальной стенки. Данный параметр позволяет задавать интервал изменения плотности системы, что даёт нам возможность изменять плотность на столько медленно, на сколько это требуется условиями эксперимента.

Пример:

`compress_right_wall 0.49 0.0000001 1000`

Данная команда изменит относительную плотность частиц в системе до значения 0.49000, при этом максимальное расстояние, на которое будет меняться расстояние между частицами и идеальной стенкой, будет не более $\Delta L_{max} = 0.0000001$ и после каждого небольшого изменения плотности системы будет произведён расчёт динамики движения частиц в системе на протяжении тысячи соударений на каждую частицу в этой системе.

Короткая запись команды:

Данная команда имеет вариант короткой записи, которая имеет те же аргументы и работает так же, как и полная запись команды:

compress 0.49 0.0000001 1000

14.5.3 *compress_both_walls*

Необходимо описать подробно.

14.6 image

Управляющая команда *image* собирает информацию об относительной плотности частиц в системе и сохраняет эту информацию в текстовый файл.

Необходимо описать подробно.

14.7 profile

Управляющая команда *profile* сохраняет информацию о положениях центров частиц в некотором "слое" системы, что позволяет изучать структуру системы жёстких сфер в некоторой выделенной области моделируемой системы. Информация о положении центров частиц в некотором "слое" системы, ограниченным плоскостями $x_1 = const$ и $x_2 = const$, называется "разрезом т.к. позволяет нам увидеть положение центров частиц, находящихся в данном "слое" в проекции на плоскость OYZ .

Необходимо описать подробно.

15 Глоссарий

16 Список литературы

- [1] В.П. Вешнев Т.А. Нурлыгаянов. Кристаллизация твердых сфер вблизи стенки // Письма в ЖТФ. 2011. Т. 37, № 18.
- [2] S. Miller S. Luding. Event-driven molecular dynamics in parallel // Journal of Computational Physics. 2003. Т. 193. с. 306–316.
- [3] Rapaport Dennis C. The Event-Driven Approach to N-Body Simulation // Progress of Theoretical Physics Supplement. 2009. № 178. С. 5–14.
- [4] Белкин А. А. ОБ ОДНОЙ МОДИФИКАЦИИ МЕТОДА МОЛЕКУЛЯРНОЙ ДИНАМИКИ // Сибирский журнал индустриальной математики. 2006. Октябрь–декабрь. Т. IX, № 4(28). С. 27–32.
- [5] Rosenbluth Marshall N., Rosenbluth Arianna W. Further Results on Monte Carlo Equations of State // J. Chem. Phys. 22, 881 (1954); <http://dx.doi.org/10.1063/1.1740207>. 1954.
- [6] Wood W. W., Jacobson J. D. Preliminary Results from a Recalculation of the Monte Carlo Equation of State of Hard Spheres // J. Chem. Phys. 1957. Т. 27. с. 1207.
- [7] Alder B. J., Wainwright T. E. Phase Transition for a Hard Sphere System // J. Chem. Phys. 1957. Т. 27. с. 1208.
- [8] Джанколи Д. Физика. Т. I. Мир, 1989.