

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

Кафедра вычислительной физики  
и моделирования физических процессов

Р. А. Хабибуллин, А. В. Мокшин

ВЫЧИСЛЕНИЕ ДИНАМИЧЕСКОГО  
СТРУКТУРНОГО ФАКТОРА НА ОСНОВЕ ДАННЫХ  
МОДЕЛИРОВАНИЯ КЛАССИЧЕСКОЙ  
МОЛЕКУЛЯРНОЙ ДИНАМИКИ

Учебное пособие

Казань – 2022

**УДК 538.9**  
**ББК 22.317**

Издается по решению Ученого совета Института физики  
Казанского федерального университета

*Работа поддержана грантом Фонда развития теоретической физики и  
математики «БАЗИС» (проект № 20–1–2–38)*

Рецензенты:

к.ф.-м.н. **Р. М. Хуснутдинов**

к.ф.-м.н. **Б. Н. Галимзянов**

Хабибуллин Р. А., Мокшин А. В. **Вычисление динамического структурного фактора на основе данных моделирования классической молекулярной динамики: учебное пособие** / Р. А. Хабибуллин, А. В. Мокшин. — Казань: Казанский федеральный университет, 2022. — 27 с.

© Р. А. Хабибуллин,  
А. В. Мокшин, 2022

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Неупругое рассеяние света и рентгеновских лучей . . . . .	4
1.2	Неупругое рассеяние медленных нейтронов . . . . .	6
<b>2</b>	<b>Детали моделирования</b>	<b>9</b>
<b>3</b>	<b>Флуктуации локальной плотности</b>	<b>12</b>
3.1	Чтение дамп-файла . . . . .	14
3.2	Усреднение по различным направлениям волнового вектора . . .	16
<b>4</b>	<b>Промежуточная функция рассеяния</b>	<b>18</b>
4.1	Метод интегрирования Филона . . . . .	21
<b>5</b>	<b>Применение теоремы Винера — Хинчина</b>	<b>23</b>

# 1 Введение

Пусть имеется система  $N$  взаимодействующих частиц, заключенных в объеме  $V$ , где  $T$  – есть температура системы. Очевидно, что микроскопическая динамика системы характеризуется набором различных переменных  $\mathbf{A}$ , в который могут входить, например, локальная плотность, смещение частицы в пространстве, скорость частицы, дипольный момент и т.д.. Тем не менее, вполне удобно рассматривать изначально те переменные, которые либо связаны с экспериментально измеряемой функцией отклика, либо характеризует некоторый интересующий релаксационный процесс.

Для получения информации о свойствах различных физических систем широко используется экспериментальная техника, в которой регистрируется отклик системы на внешние возмущения. В качестве возмущений могут выступать электромагнитное излучение (в диапазоне от рентгеновских длин волн до радиоволн) или поток «зондирующих частиц» (тепловых нейтронов, электронов и т.д.) [1; 2].

## 1.1 Неупругое рассеяние света и рентгеновских лучей

Рассмотрим случай неупругого рассеяния фотонов. В результате взаимодействия фотонов с системой их энергия изменяется от некоторого начального значения  $E_{in}$  до конечного значения  $E_{fin}$  на величину  $\hbar\omega = E_{in} - E_{fin}$ . Тогда с учетом закона сохранения энергии можно записать  $E_{in} + \varepsilon_{in} = E_{fin} + \varepsilon_{fin}$ , где  $\varepsilon_{in}$  и  $\varepsilon_{fin}$  – есть энергии начального и конечного состояний системы. Величина  $\hbar\omega = \varepsilon_{fin} - \varepsilon_{in}$  должна нести информацию об энергиях возбуждений, связанных, например, с движениями ионов или свободных электронов в изучаемой системе. С другой стороны, в результате процесса рассеяния импульс фотона меняется от некоторого начального значения  $\hbar\mathbf{k}_{in}$  до значения  $\hbar\mathbf{k}_{fin}$ , где изменение определяется волновым вектором  $\mathbf{k} = \mathbf{k}_{in} - \mathbf{k}_{fin}$ , где  $\mathbf{k}$  — волновой вектор.

Интенсивность рассеяния может быть выражена через дифференциальное

поперечное сечение рассеяния, пропорционального величине [3]

$$I_{\text{em}}(\mathbf{k}, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt e^{i\omega t} \langle A_k^\dagger(0) A_k(t) \rangle, \quad (1)$$

где конкретный вид  $A_k$  зависит от физической природы системы, а также от частоты падающей электромагнитной волны. Для системы, состоящей из одинаковых атомов с атомарным числом  $Z$ , при рассеянии света величина  $A_k$  принимает вид [4]:

$$A_k \approx k_{\text{in}}^2 \sum_i \alpha_0 e^{i\mathbf{k} \cdot \mathbf{r}_i}, \quad (2)$$

где  $\alpha_0$  — коэффициент атомной поляризации. В случае рассеяния рентгеновских лучей величина  $A_k$  имеет иной вид [5]:

$$A_k \approx r_0 \sum_i f(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{r}_i}. \quad (3)$$

Здесь  $r_0 = e^2/m_e c^2 = 2.82 \cdot 10^{-13}$  см — классический радиус электрона,  $f(\mathbf{k})$  — атомный формфактор, который является безразмерной убывающей функцией волнового вектора  $\mathbf{k}$  с начальным значением  $f(0) = Z$ .

Из вышеприведенных примеров видно, что независимо от конкретного вида величины  $A_k$  дифференциальное поперечное сечение рассеяния содержит временную корреляционную функцию — так называемую функцию рассеяния

$$F_r(\mathbf{k}, t) = \frac{1}{N} \left\langle \sum_i e^{-i\mathbf{k} \cdot \mathbf{r}_i(0)} \sum_j e^{i\mathbf{k} \cdot \mathbf{r}_j(t)} \right\rangle = \frac{1}{N} \langle \rho(\mathbf{k}, 0) \rho^*(\mathbf{k}, t) \rangle, \quad (4)$$

частотный спектр которой определяет динамический структурный фактор

$$\begin{aligned} S(\mathbf{k}, \omega) &= \frac{1}{2\pi N} \int_{-\infty}^{\infty} dt e^{-i\omega t} \left\langle \sum_{i,j} e^{-i\mathbf{k} \cdot (\mathbf{r}_i(0) - \mathbf{r}_j(t))} \right\rangle = \\ &= \frac{1}{2\pi N} \int_{-\infty}^{\infty} dt e^{-i\omega t} F_r(\mathbf{k}, t). \end{aligned} \quad (5)$$

В результате, основная неупругая часть поперечного сечения рассеяния в выражении 5 содержит информацию о временном поведении корреляций флуктуаций локальной плотности в рассматриваемой системе

$$\rho(\mathbf{k}) = \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{i\mathbf{k}\cdot\mathbf{r}_j}. \quad (6)$$

Таким образом, анализируя соответствующее излучение, рассеянное на образце, можно непосредственно извлекать информацию о динамике флуктуаций коллективной плотности в системе. Поскольку эти свет и рентгеновские лучи характеризуются различной длиной волны, в опытах по рассеянию будут проявляться процессы, происходящие на различных пространственных масштабах. С помощью законов сохранения энергии и импульса можно получить следующее соотношение:

$$|\mathbf{k}| = k = 2k_{\text{in}} \sin(\Theta/2), \quad (7)$$

где  $\Theta$  - угол рассеяния (между  $\mathbf{k}_{\text{in}}$  и  $\mathbf{k}_{\text{fin}}$ ). Как видно из этого соотношения, максимальное значение волнового вектора, которое может быть получено в эксперименте по рассеянию, равно  $2k_{\text{in}}$ . Это означает, что минимальная доступная пространственная область в системе должна превышать  $\lambda_{\text{in}}/2$ . Для рентгеновских лучей ( $\lambda_{\text{in}} \approx 1\text{\AA}$ ) это соответствует области атомарных размеров и межчастичных расстояний. При оптическом излучении с  $\lambda_{\text{in}} \sim 10^3\text{\AA}$  наблюдению доступны лишь большие пространственные флуктуации, соответствующие большим (гидродинамическим) пространственным масштабам.

## 1.2 Неупругое рассеяние медленных нейтронов

Тепловые нейтроны обладают энергией порядка  $10^{-2}$  эВ и длиной волны порядка  $1\text{\AA}$  [6]. Следовательно, пространственно-частотные характеристики возбуждений, наблюдаемые с помощью рассеяния тепловых нейтронов, также должны быть сопоставимы с этими значениями. Основное нейтрон-ядерное взаимодействие является короткодействующим и может быть описано с помощью эффективного

потенциала вида

$$(2\pi\hbar^2/m_N)b\delta(\mathbf{R}_N - \mathbf{r}),$$

где  $m_N$  — масса нейтрона,  $\mathbf{R}_N$  — его положение, и  $\mathbf{r}$  — положение ядра [4]. Величина  $b$  — есть так называемая длина рассеяния, которая зависит от атомного числа ядра (то есть его химических свойств) и массы ядра. Фактически,  $b$  является оператором с комплексными собственными значениями. Следовательно, даже в самом простом случае одноатомной системы с химически идентичными атомами (например, жидкие щелочные металлы) возникает необходимость в усреднении величины  $b$  по различным ядерным переменным. Тогда дифференциальное поперечное сечение рассеяния будет пропорционально выражению

$$I_n(\mathbf{k}, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt \exp^{-i\omega t} \sum_{i,j} \overline{b_i^+ b_j} \left\langle e^{-i\mathbf{k}\cdot\mathbf{r}_i(0)} e^{i\mathbf{k}\cdot\mathbf{r}_j(t)} \right\rangle, \quad (8)$$

где верхняя черта обозначает вышеупомянутое ядерное усреднение. Определив среднее как  $\overline{b_i^+ b_j} = (|\overline{b}|^2 + \delta_{i,j} [|\overline{b}|^2 - |\overline{b}|^2]) = (1/4\pi) [\sigma_{\text{coh}} + \sigma_{\text{incoh}}]$ , выражение 8 можно записать в виде суммы двух различных вкладов:

$$I_n(\mathbf{k}, \omega) = \frac{N}{4\pi} [\sigma_{\text{coh}} S(\mathbf{k}, \omega) + \sigma_{\text{incoh}} S_s(\mathbf{k}, \omega)], \quad (9)$$

где величины  $\sigma_{\text{coh}}$  и  $\sigma_{\text{incoh}}$  представляют собой весовые вклады когерентного и некогерентного рассеяния, соответственно. Тогда удобно ввести следующие факторы:

$$S(\mathbf{k}, \omega) = \frac{1}{2\pi N} \int_{-\infty}^{\infty} dt e^{-i\omega t} \sum_{i,j} \left\langle e^{-i\mathbf{k}\cdot\mathbf{r}_i(0)} e^{i\mathbf{k}\cdot\mathbf{r}_j(t)} \right\rangle, \quad (10)$$

$$S_s(\mathbf{k}, \omega) = \frac{1}{2\pi N} \int_{-\infty}^{\infty} dt e^{-i\omega t} \sum_i \left\langle e^{-i\mathbf{k}\cdot\mathbf{r}_i(0)} e^{i\mathbf{k}\cdot\mathbf{r}_i(t)} \right\rangle, \quad (11)$$

Величина  $S(\mathbf{k}, \omega)$  — является динамическим структурным фактором и представляет собой когерентную часть спектра, которая является единственной в рассеянии фотонов. В отличие от рассеяния рентгеновских лучей и света, полный спектр рассеяния здесь содержит также некогерентный вклад  $S_s(\mathbf{k}, \omega)$ , который

описывает одночастичную динамику. Соответствующие им ВКФ будут иметь следующий вид:

$$F_r(\mathbf{k}, t) = \frac{1}{N} \sum_{i,j} \frac{\langle e^{-i\mathbf{k}\cdot\mathbf{r}_i(0)} e^{i\mathbf{k}\cdot\mathbf{r}_j(t)} \rangle}{\langle e^{-i\mathbf{k}\cdot[\mathbf{r}_i(0)-\mathbf{r}_j(0)]} \rangle}, \quad (12)$$

$$F_{\text{self}}(\mathbf{k}, t) = \frac{1}{N} \sum_i \langle e^{-i\mathbf{k}\cdot\mathbf{r}_i(0)} e^{i\mathbf{k}\cdot\mathbf{r}_i(t)} \rangle. \quad (13)$$



## 2 Детали моделирования

Для моделирования равновесной системы при некоторых заданных давлении и температуре могут быть использованы следующие ансамбли: изотермо-изобарический NPT-, канонический NVT- и микроканонический NVE-ансамбль. Рассмотрим сценарий молекулярно-динамического пакета LAMMPS [7; 8], в котором для моделирования применяются все три вышеописанных ансамбля:

```
variable      xx  equal 20
variable      yy  equal 20
variable      zz  equal 20

units         lj

variable      P    equal 6.6
variable      T    equal 2.8

variable      rho0 equal 0.75

lattice      fcc ${rho0}
region       1    block 0 ${xx} 0 ${yy} 0 ${zz} units lattice
create_box   1    1
create_atoms 1    box
mass         1    1.0

pair_style   lj/smooth/linear 5.00
pair_coeff   * * 1.00 1.00

neigh_modify delay 0 every 1 check yes

thermo       1000
thermo_style custom step temp press pe density cpuremain

variable     seed  equal 28597
velocity     all create 1.0 ${seed} rot yes dist gaussian

# S T A G E   1

fix          1    all nvt temp $T $T $(100 * dt)
run          100000
unfix       1

# S T A G E   2
```

```

fix          2  all npt temp $T $T $(100 * dt) iso $P $P $(250 * dt)
run          100000
unfix       2

# S T A G E   3

timestep    0.001

fix         3  all nve
run        100000
unfix      3

# S T A G E   4

reset_timestep 0

dump        1  all custom 5 dump/lj_nve_dump.lampstrj xs ys zs
dump_modify 1 format float %9.7f pbc yes

fix         extra all print 50 "$(temp) $(press) $(etotal)" &
           file thermo/lj_nve_thermo.dat screen no

fix         4  all nve
run        5000000
unfix      4

unfix       extra

undump      1

```

Здесь моделирование выполняется в 4 этапа:

1. плавление кристалла в NVT-ансамбле; (fix nvt).
2. приведение расплава к заданному давлению в NPT-ансамбле;
3. релаксация системы в микроканоническом ансамбле;
4. моделирование равновесной системы в NVE-ансамбле с сохранением траекторий частиц в файл.

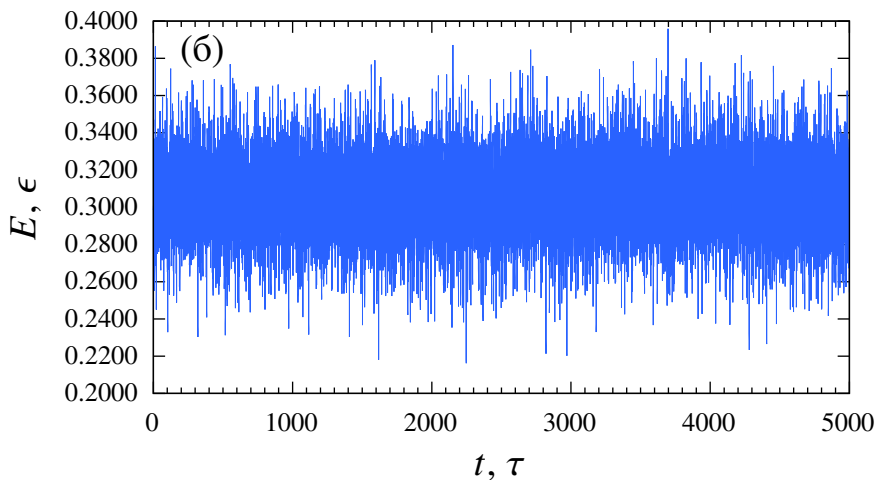
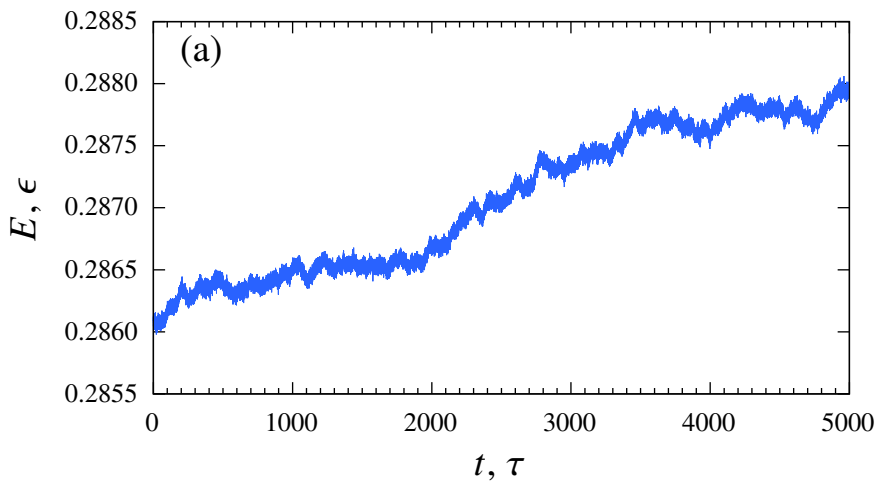


Рис. 1: Зависимость полной энергии системы от времени в случае: (а) моделирования в NVE-ансамбле, (б) моделирования в NVT-ансамбле. Величина шага интегрирования уравнений движения  $0.001 \tau$  и  $0.005 \tau$  соответственно

### 3 Флуктуации локальной плотности

Чтобы получить рабочую формулу для вычисления динамического структурного фактора  $S(\mathbf{k}, \omega)$ , запишем еще раз определение промежуточной функции рассеяния

$$F(\mathbf{k}, t) = \frac{1}{N} \langle \rho_{\mathbf{k}}(t) \rho_{-\mathbf{k}}(0) \rangle. \quad (14)$$

Поставляя (6) в это выражение, получаем

$$\begin{aligned} F(\mathbf{k}, t) &= \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{l=1}^N \exp[-i\mathbf{k} \cdot \mathbf{r}_j(t)] \exp[i\mathbf{k} \cdot \mathbf{r}_l(0)] \right\rangle = \\ &= \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{l=1}^N \exp[-i\mathbf{k} \cdot (\mathbf{r}_j(t) - \mathbf{r}_l(0))] \right\rangle. \end{aligned} \quad (15)$$

После этого остается только «надеть» на выражение (15) интеграл из преобразования Фурье:

$$S(\mathbf{k}, \omega) = \frac{1}{N} \int_{-\infty}^{\infty} dt \left\langle \sum_{j=1}^N \sum_{l=1}^N \exp[-i\mathbf{k} \cdot (\mathbf{r}_j(t) - \mathbf{r}_l(0))] \right\rangle \exp(i\omega t). \quad (16)$$

Данная формула хороша тем, что позволяет рассчитывать динамический структурный фактор непосредственно из координат частиц. Однако, такой подход является вычислительно неэффективным из-за наличия в формуле (15) двойного суммирования — для расчета каждого отдельно взятого по времени значения  $F(\mathbf{k}, t)$  потребуется  $N^2$  операций без учета затрат на усреднение. Поэтому, с точки зрения производительности, более оптимальным будет оставить выражение (14) как есть, то есть вычислять промежуточную функцию рассеяния  $F(\mathbf{k}, t)$  на основе предварительно полученного временного ряда  $\rho_{\mathbf{k}}(t)$  — *флуктуаций локальной плотности*.

Используя формулу Эйлера, запишем по отдельности действительную и

мнимую части выражения (??):

$$\operatorname{Re} \rho_{\mathbf{k}}(t) = \sum_{j=1}^N \cos(\mathbf{k} \cdot \mathbf{r}_j(t)), \quad (17)$$

$$\operatorname{Im} \rho_{\mathbf{k}}(t) = - \sum_{j=1}^N \sin(\mathbf{k} \cdot \mathbf{r}_j(t)). \quad (18)$$

В таком виде формулы (17) и (18) прозрачно транслируются в программный код:

```
for (int i = 0; i < stepCount; i++)
{
    double re = 0.0;
    double im = 0.0;

    Span<double> step = coordinates.AsSpan().Slice(i * stepSize, stepSize);

    foreach (Vector3 position in MemoryMarshal.Cast<double, Vector3>(step))
    {
        Vector3 dot = Vector3.Dot(wavevector, position);

        re += Math.Cos(dot);
        im -= Math.Sin(dot);
    }

    rho[j].Add(new Complex(re, im) / Math.Sqrt(particleCount));
}
```

Рассмотрим подробно данный фрагмент. Здесь предполагается, что координаты частиц уже прочитаны из дамп-файла в массив `coordinates`. Это одномерный массив чисел типа `double`, в котором координаты атомов располагаются по порядку друг за другом

`x1, y1, z1, x2, y2, z2, ...`

В основном цикле `for` осуществляется проход по прочитанным из дамп-файла временным шагам; в переменную `stepCount` записано количество прочитанных шагов. Далее, из массива `coordinates` в переменную `step` выделяется фрагмент с координатами частиц на текущем временном шаге. После этого, во вложенном цикле `foreach` производится суммирование с последовательным

перебором радиус-векторов частиц. При этом радиус-вектора атомов (переменная `position` типа `Vector3`) генерируются «на лету» из координат атомов при помощи статического класса `MemoryMarshal` из пространства имен `System.Runtime.InteropServices`. Использование структуры `Vector3` избавляет нас от необходимости самостоятельно реализовывать скалярное умножение векторов внутри цикла, так как она уже содержит в себе метод `Dot`.

### 3.1 Чтение дамп-файла

Обработка дамп-файлов больших размеров (1 ТБ или больше), несмотря на невысокую вычислительную сложность выполняемых при этом операций, может занять продолжительное время. Простейшие замеры производительности (например, с использованием класса `Stopwatch`) показывают, что большая часть времени уходит не на обработку дамп-файлов, а на их чтение.

Для ускорения чтения дамп-файлов можно дать следующие рекомендации:

1. Читать сразу несколько временных шагов из дамп-файла, сохранять их в буфер (массив `coordinates`), и только потом обрабатывать.
2. Использовать фиксированное (т. е. одинаковое для всех частиц и для всех временных шагов) количество знаков для записи координат атомов в дамп-файлах. В этом случае каждое число в строке с координатами будет иметь заранее известную позицию, выраженную в количестве символов. Это позволит сэкономить время на разбиении строки на подстроки по пробелам. Поскольку при такой форме записи «бюджет» символов, отведенный на запись числа, будет делиться между целой и дробной частями, возникает следующий побочный эффект: чем больше будет целая часть, тем меньше будет точность записи всего числа. Можно обойти это ограничение, если сохранять координаты частиц в приведенных единицах:

$$x^* = x/L_x, \quad y^* = y/L_y, \quad z^* = z/L_z.$$

Тогда целая часть всегда будет записываться одним символом (0 или

1) и точность координаты не будет зависеть от удаленности атома от начала координат. В заголовках временных шагов приведенные координаты обозначаются как ATOMS xs ys zs.

3. Явно указывать в каком формате записано число во входной строке при вызове метода `double.Parse()` или аналогичных ему. Например, если в качестве десятичного разделителя используется точка, число не может быть отрицательным и в его записи не используется экспоненциальная форма, — все это можно указать, чтобы метод не тратил время на поиск во входной строке знака минуса или символа экспоненты.

Ниже приведен пример реализации чтения дампа-файла с применением выше-описанных рекомендаций.

```
for (stepCount = 0; stepCount <= stepMax; i++)
{
    if (reader.EndOfStream)
    {
        break;
    }

    Span<double> step = coordinates.AsSpan().Slice(i * stepSize, stepSize);

    reader.ReadLine(); // ITEM: TIMESTEP
    reader.ReadLine();
    reader.ReadLine(); // ITEM: NUMBER OF ATOMS

    if (int.Parse(reader.ReadLine()) != particleCount)
    {
        throw new InvalidDataException();
    }

    reader.ReadLine(); // ITEM: BOX BOUNDS pp pp pp
    reader.ReadLine();
    reader.ReadLine();
    reader.ReadLine();

    if (reader.ReadLine() != "ITEM: ATOMS xs ys zs")
    {
        throw new InvalidDataException();
    }
}
```

```

NumberStyles style = NumberStyles.AllowDecimalPoint;
IFormatProvider provider = CultureInfo.InvariantCulture;

for (int j = 0; j < stepSize; j += 3)
{
    ReadOnlySpan<char> line = reader.ReadLine();

    step[j] = double.Parse(line.Slice(0, 9), style, provider);
    step[j + 1] = double.Parse(line.Slice(10, 9), style, provider);
    step[j + 2] = double.Parse(line.Slice(20, 9), style, provider);
}
}

```

Здесь, как и в предыдущем листинге, основной цикл `for` выполняет проход по временным шагам. Верхнее ограничение для цикла задается переменной `stepMax`, в которой хранится значение, равное размеру (в количестве шагов) буфера для чтения. Объект `reader` представляет собой экземпляр класса `StreamReader` из пространства имен `System.IO`.

Представление каждой прочитанной из файла строки с координатами атомов в виде `ReadOnlySpan<char>` полезно тем, что позволяет без значительных накладных расходов выделять из строки подстроки, в которых записаны координаты атомов. Более очевидный и менее эффективный способ разбиения строки на подстроки по пробелам при помощи метода `string.Split()`. В этом случае, на каждой итерации вложенного цикла `for` внутри метода `string.Split()` выполнялось посимвольное сканирование строки на наличие пробелов, создание массива для подстрок и копирование каждой подстроки в массив. Поскольку все координаты атомов в дамп-файле имеют фиксированную длину, мы можем не расходовать процессорное время на поиск пробелов и разделение строки по пробелам, а вместо этого выделять подстроки по их координатам. В заголовках временных шагов приведенные координаты обозначены как `xs ys zs`.

## 3.2 Усреднение по различным направлениям волнового вектора

В конденсированных неупорядоченных средах различные направления волнового вектора  $\mathbf{k}$  статистически равнозначны. Данное свойство имеет два важных для



нас следствия в контексте вычисления спектральных характеристик:

- 1) зависимость динамического структурного фактора от волнового вектора сводится к зависимости от волнового числа  $S(\mathbf{k}, \omega) \rightarrow S(k, \omega)$ ;
- 2) возможно повышение качества спектров через дополнительное усреднение спектров по всем возможным направлениям волнового вектора.

Также важно отметить, что допустимые значения волнового вектора  $\mathbf{k}$  в случае, когда траектории частиц получены из МД-моделирования в кубической симуляционной ячейке, связаны с длиной ребра ячейки  $L$  следующим соотношением:

$$\mathbf{k} = 2\pi/L(a, b, c) \tag{19}$$

где  $a, b, c$  — целые числа. Это соотношение накладывает ограничение на количество возможных направлений  $\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_n$  волнового вектора с фиксированной длиной  $k = |\mathbf{k}_1| = |\mathbf{k}_2| = \dots = |\mathbf{k}_n|$ , где  $n$  — число таких направлений. Максимальное количество возможных направлений  $n = 48$  реализуется при условии  $a \neq b \neq c \neq 0$ . Наглядно покажем это на примере. Если все компоненты вектора  $\mathbf{k}$  уникальны, то мы можем записать  $3! = 6$  перестановок компонент (левый столбец). Далее, если ни одна из компонент не равна нулю, то для каждой перестановки можно записать еще 8 вариантов, меняя знаки у компонент вектора (правый столбец).

1	a	b	c	<<<<	1	a	b	c
2	a	c	b		2	a	-b	c
3	b	a	c		3	a	-b	-c
4	b	c	a		4	a	b	-c
5	c	a	b		5	-a	b	c
6	c	b	a		6	-a	-b	c
					7	-a	-b	-c
					8	-a	b	-c

Всего получается  $6 \cdot 8 = 48$  вариантов вектора  $\mathbf{k}$  с одинаковой длиной

$$k = 2\pi/L\sqrt{a^2 + b^2 + c^2} . \tag{20}$$

## 4 Промежуточная функция рассеяния

Промежуточная функция рассеяния (14) представляет собой временную корреляционную функцию, которая вычисляется методом скользящего окна. Главное условие применимости этого метода состоит в том, что исходный временной ряд (в данном случае  $\rho_{\mathbf{k}}(t)$ ) должен быть определен на интервале времени, значительно превышающем по продолжительности интервал, для которого нам нужно посчитать ВКФ. Изначально задается некоторой временной промежуток — т. н. окно, через которое нам будет видна лишь часть исходной последовательности. Для этой подпоследовательности вычисляется коррелятор. Далее, мы сдвигаем окно на один элемент исходной последовательности вперед, получая таким образом новую подпоследовательность такой же длины, и считаем коррелятор для нее. Данная операция повторяется до тех пор, пока в окно не попадет последний элемент исходной последовательности. В конце все посчитанные корреляторы усредняются. Иллюстрация работы метода приведена на рисунке 2.

Метод скользящего окна для вычисления промежуточной функции рассеяния реализуется достаточно просто:

```
var correlator = new Complex[timeWindowSize];

for (int i = 0; i <= density.Count - timeWindowSize; i++)
{
    Complex rho0 = Complex.Conjugate(density[j]);

    for (int j = 0; j < correlator.Length; j++)
        correlator[j] += rho[i + j] * rho0;
}

for (int i = 0; i < correlator.Length; i++)
{
    correlator[i] /= rho.Count - timeWindowSize + 1;
}
```

Здесь  $\rho$  — массив со значениями  $\rho_{\mathbf{k}}(t)$ . Расчет  $F(\mathbf{k}, t)$  для различных значений волнового вектора  $\mathbf{k}$  можно легко распараллелить, например, при помощи метода `Parallel.For` из пространства имен `System.Threading.Tasks`.

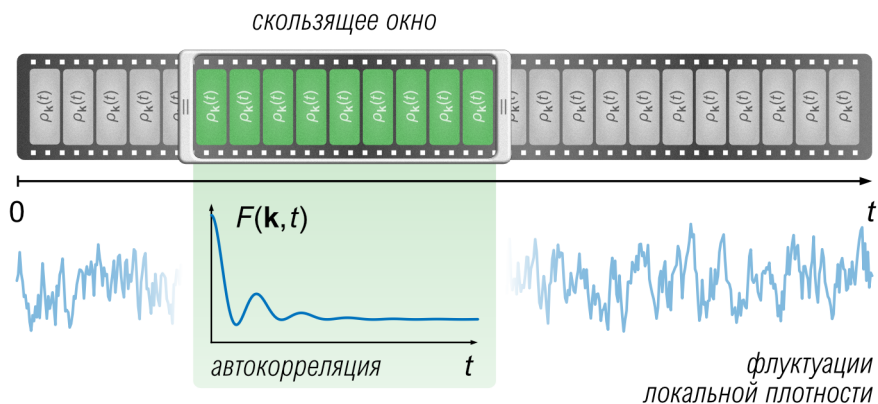


Рис. 2: Иллюстрация процесса вычисления автокорреляционной функции методом скользящего окна

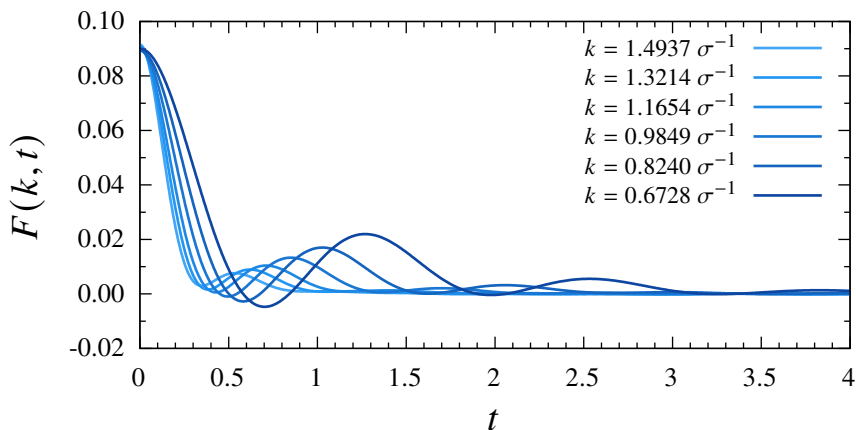


Рис. 3: Промежуточная функция рассеяния  $F(\mathbf{k}, t)$  для леннард-джонсовской системы в сверхкритическом состоянии при давлении  $P = 6.6 \epsilon/\sigma^3$  и температуре  $T = 2.8 \epsilon/k_B$

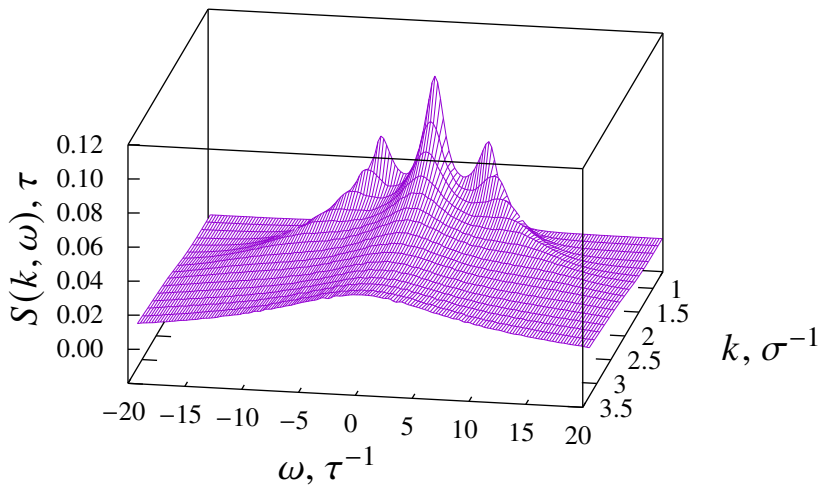


Рис. 4: Динамический структурный фактор, полученный из результатов МД-моделирования сверхкритического флюида с межчастичным потенциалом Леннард-Джонса при давлении  $P = 6.6 \epsilon/\sigma^3$  и температуре  $T = 2.8 \epsilon/k_B$

## 4.1 Метод интегрирования Филона

Промежуточная функция рассеяния является четной функцией времени, поэтому формула динамического структурного фактора может быть записана как

$$S(\mathbf{k}, \omega) = 2 \int_0^{\infty} dt F(\mathbf{k}, t) \cos(\omega t). \quad (21)$$

В этом выражении от комплексной величины  $F(\mathbf{k}, t)$  берется только действительная часть. В численных расчетах интегрирование до бесконечности заменяется интегрированием на конечном промежутке времени

$$\hat{C}(\omega) = 2 \int_0^{t_{\max}} dt C(t) \cos(\omega t) \quad (22)$$

Для вычисления таких интегралов применим метод численного интегрирования Филона [9]. Пусть

$$t_{\max} = 2n\delta t. \quad (23)$$

Далее, определим

$$\theta = \omega\delta t, \quad (24)$$

тогда для Фурье-образа запишем

$$\hat{C}(\omega) = 2\delta t(\alpha C(t_{\max}) \sin(\omega t_{\max}) + \beta C_e + \gamma C_o), \quad (25)$$

где

$$\begin{aligned} \alpha &= (1/\theta^3)(\theta^2 + \theta \sin \theta \cos \theta - 2 \sin^2 \theta), \\ \beta &= (2/\theta^3)(\theta(1 + \cos^2 \theta) - 2 \sin \theta \cos \theta), \\ \gamma &= (4/\theta^3)(\sin \theta - \theta \cos \theta). \end{aligned} \quad (26)$$

Ниже представлена реализация прямого преобразования Фурье на основе метода интегрирования Филона:

```
public static double FourierForward(double[] real, double dt, double omega)
{
    if (real.Length % 2 == 0)
```

```

{
    throw new ArgumentException($"Array length must be odd.",
        nameof(real));
}

double tmax = (real.Length - 1) * dt;

double theta = omega * dt;
double theta2 = theta * theta;
double theta3 = theta2 * theta;

double sin = Math.Sin(theta);
double sin2 = sin * sin;

double cos = Math.Cos(theta);
double cos2 = cos * cos;

double alpha = (1.0 / theta3) * (theta2 + theta * sin * cos - 2.0 * sin2);
double beta = (2.0 / theta3) * (theta * (1.0 + cos2) - 2.0 * sin * cos);
double gamma = (4.0 / theta3) * (sin - theta * cos);

double[] curve = real
    .Select((r, i) => r * Math.Cos(omega * i * dt))
    .ToArray();

curve[0] *= 0.5;
curve[^1] *= 0.5;

double evenSum = curve
    .Where((c, i) => i % 2 == 0)
    .Sum();

double oddSum = curve
    .Where((c, i) => i % 2 != 0)
    .Sum();

return 2.0 * dt * (alpha * real[^1] * Math.Sin(omega * tmax) + beta *
    evenSum
    + gamma * oddSum);
}

```

## 5 Применение теоремы Винера — Хинчина

Существует более простой в реализации подход для вычисления динамического структурного фактора на основе теоремы Винера — Хинчина. Данная теорема утверждает, что преобразование Фурье автокорреляции от функции равно квадрату модуля ее преобразования [10]. В этом случае выражение (1) для динамического структурного фактора приобретает следующий вид:

$$S(\mathbf{k}, \omega) \propto \lim_{t_M \rightarrow \infty} \frac{1}{t_M} \left| \int_0^{t_M} dt \rho_{\mathbf{k}}(t) \exp(i\omega t) \right|^2. \quad (27)$$

В выражении (27) динамический структурный фактор рассчитывается непосредственно из функции локальной плотности  $\rho_{\mathbf{k}}(t)$ , что позволяет избежать ошибок, связанных с вычислением промежуточной функции рассеяния  $F(\mathbf{k}, t)$  [11]. Недостатком такого подхода является сильная зашумленность получаемых спектров. Сгладить шум можно выполнив свертку спектра с некоторой оконной функцией, например, с гауссовой функцией

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{x - \mu}{\sigma} \right)^2 \right], \quad (28)$$

где параметр  $\mu$  определяет положение пика, а параметр  $\sigma$  — ширину «колокола» (гауссова функция имеет колоколообразный вид). В дискретном виде выражение (28) запишется как

$$w(n) = \exp \left[ -\frac{1}{2} \left( \frac{n - N/2}{\sigma N/2} \right)^2 \right]. \quad (29)$$

Здесь  $0 \leq n \leq N$  — индекс элемента окна,  $N$  — размер окна (количество элементов в массиве при этом равно  $N + 1$ ). На практике для выполнения пост-обработки спектров удобно использовать программный пакет MATLAB благодаря следующим возможностям пакета:

- быстрый импорт данных из текстовых файлов при помощи только графического интерфейса программы без необходимости ввода консольных команд;

- наличие готовых реализаций различных оконных функций, в том числе и гауссовой функции (`gausswin`);
- операции фильтрации скользящим окном для одномерных и двумерных массивов (матриц) уже реализованы в пакете и представлены функциями `filter` и `filter2` соответственно.

Таким образом, код для фильтрации шумов в спектрах динамического структурного фактора может быть записан всего в 3 строки:

```
window = gausswin(50, 2.5);  
window = window / sum(window);  
  
S_ave = filter(window, 1.0, S);
```

Здесь  $S$  — массив с исходным спектром,  $S_{ave}$  — сглаженный спектр. Чтобы избежать появления нежелательных артефактов свертки вблизи  $\omega = 0$ , рекомендуем сглаживать двусторонний спектр, т. е. спектр, продолженный в область отрицательных частот.

В качестве примера на рисунке (5) показан динамический структурный фактор для жидкого галлия, полученный вышеизложенным методом из результатов молекулярно-динамического моделирования.



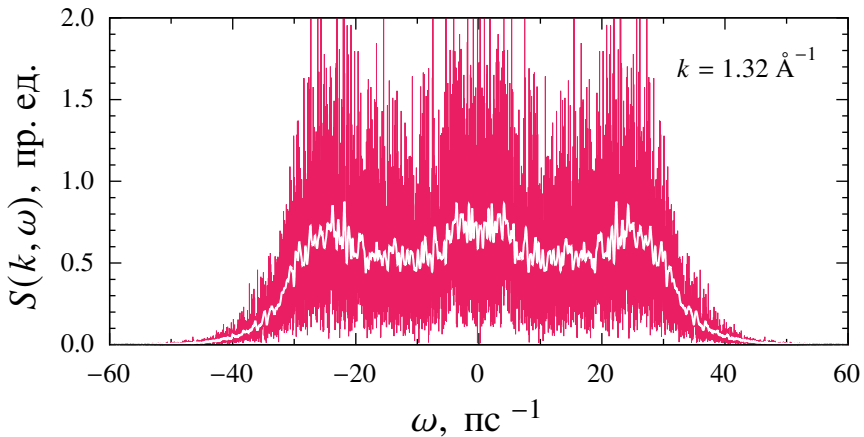


Рис. 5: Частотный спектр динамического структурного фактора (в произвольных единицах) для жидкого галлия при температуре  $T = 794$  К и давлении  $P = 1$  бар [12]. Цветом показан исходный спектр, белой линией — свертка спектра с гауссовой функцией

## Список литературы

1. Темперли Г., Роулинсон Д., Раибрук Д. Физика простых жидкостей. Статистическая теория. — М. : Мир, 1971. — С. 308.
2. Темперли Г., Роулинсон Д., Раибрук Д. Физика простых жидкостей. Экспериментальные исследования. — М. : Мир, 1973. — С. 400.
3. Balucani U., Lee M. H., Tognetti V. Dynamical correlations // Phys. Rep. — 2003. — Т. 373, № 6. — С. 409—492.
4. Wang C. H. Spectroscopy of condensed media: dynamics of molecular interactions. — New York : Academic Press, 1985. — С. 356.
5. Burkel E. Inelastic scattering of X-rays with very high energy resolution. — Berlin : Springer, 1991. — С. 356.
6. Egelstaff P. A. Thermal neutron scattering. — New York : Academic Press, 1965. — С. 540.

7. *Plimpton S.* Fast parallel algorithms for short-range molecular dynamics // Journal of computational physics. — 1995. — Т. 117, № 1. — С. 1—19.
8. Implementing molecular dynamics on hybrid high performance computers—short range forces / W. M. Brown [и др.] // Comput. Phys. Commun. — 2011. — Т. 182, № 4. — С. 898—911.
9. *Allen M. P., Tildesley D. J.* Computer simulation of liquids. — Oxford university press, 2017.
10. *Стюард И. Г.* Введение в фурье-оптику: Пер. с англ. — М. : Мир, 1985. — С. 182.
11. *Khusnutdinoff R. M.* Microscopic collective dynamics of water // Colloid J. — 2016. — Т. 78, № 2. — С. 225—234.
12. Collective modes and gapped momentum states in liquid Ga: Experiment, theory, and simulation / R. M. Khusnutdinoff [и др.] // Phys. Rev. B. — 2020. — Т. 101, вып. 21. — С. 214312.

## Ссылки на полезные ресурсы

- Легковесная виртуальная машина Linux для Windows 10 с поддержкой вычислений на видеокартах Nvidia  
<https://docs.microsoft.com/ru-ru/windows/wsl/install>
- Дистрибутив Arch Linux, адаптированный для работы в подсистеме WSL  
<https://github.com/yuk7/ArchWSL>
- Программный пакет молекулярно-динамического моделирования LAMMPS  
<https://www.lammps.org>
- Кроссплатформенный текстовый редактор Visual Studio Code  
<https://code.visualstudio.com>
- Расширение Lammps Syntax Highlighting для Visual Studio Code, добавляющее подсветку синтаксиса в сценариях LAMMPS  
<https://marketplace.visualstudio.com>
- Библиотека для научных вычислений MathNet.Numerics, которая может быть использована в качестве бесплатной замены пакету MATLAB
- Утилита, позволяющая использовать C# как язык сценариев  
<https://github.com/filipw/dotnet-script>
- Открытый кроссплатформенный программный пакет для построения графиков  
<http://www.gnuplot.info>