КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

Кафедра вычислительной физики
и моделирования физических процессов

Р.М. ХУСНУТДИНОВ, А.В. МОКШИН

СБОРНИК ЗАДАЧ ПО КУРСУ «ВЫЧИСЛИТЕЛЬНАЯ ФИЗИКА»

Учебно-методическое пособие

Казань – 2021

УДК 53.072; 530.1; 530.145

ББК 22.311.4

Принято на заседании учебно-методической комиссии ИФ

Протокол № 5 от 4 марта 2021 года

Рецензенты:

кандидат физико-математических наук,
доцент кафедры вычислительной физики КФУ **Б.Н. Галимзянов;**
кандидат физико-математических наук,
доцент кафедры информационных систем КФУ **Ф.М. Гафаров**

Хуснутдинов Р.М.

Сборник задач по курсу «Вычислительная физика» / Р.М. Хуснутдинов, А.В. Мокшин. – Казань: Казан. ун-т, 2021. – 48 с.

В данном учебно-методическом пособии представлены задачи по курсу «Вычислительная физика», а также примеры с решениями типовых задач. Настоящее пособие предназначено для студентов физических специальностей высших учебных заведений при изучении дисциплин, связанных с программированием, информационными технологиями, численными методами и компьютерным моделированием. В сборнике представлены задания по физике, для решения которых предлагается использовать различные методы компьютерного моделирования и численного решения с применением вычислительного пакета Matlab.

© Хуснутдинов Р.М., 2021

© Казанский университет, 2021

Оглавление

Предисловие 4
§1. Метод конечных разностей 5
§2. Задачи механики 7
§3. Колебания в электрических цепях 12
§4. Задачи ядерной физики 14
§5. Метод наименьших квадратов 17
§6. Псевдослучайные числа 21
§7. Метод Монте-Карло 22
§8. Моделирование движения частиц 25
§9. Моделирование динамики частиц методом Монте-Карло 36
§10. Моделирование динамики частиц методом молекулярной динамики 41
§11. Задачи квантовой механики 45
Литература 48

Предисловие

В данном учебно-методическом пособии представлены задачи по курсу «Вычислительная физика», а также примеры с решениями типовых задач. Предлагаемое пособие предназначено для организации самостоятельной и аудиторной работы на практических занятиях по курсам «Информационные технологии» и «Программирование», а также на лабораторных занятиях по учебной практике «Практика по получению первичных профессиональных умений и навыков» со студентами Института физики КФУ. По каждой теме задачник-практикум содержит основные уравнения, блок-схемы, примеры решения задач, качественные и количественные задачи, рекомендованные для обсуждения и решения, как на занятиях, так и самостоятельно. Пособие содержит задачи, в основном, средней трудности, однако имеются и задачи повышенной сложности, которые предназначены для самостоятельного решения наиболее сильными студентами.

Учебное пособие предназначено для студентов физических специальностей высших учебных заведений. Пособие составлено с целью повышения эффективности организации самостоятельной работы и аудиторных занятий студентов очного и заочного отделений физического и физико-математического факультетов. Кроме того, пособие может быть рекомендовано для использования при проведении факультативных занятий студентов на других естественнонаучных факультетах.

§1. Метод конечных разностей

1. Методом конечных разностей численно решить следующее интегро-дифференциальное уравнение

- $\frac{dx}{dt} = -a\int_0^t x(t-\tau)x(\tau)d\tau$,
- $\begin{cases} \frac{dx}{dt} = -a\int_0^t x(t-\tau)y(\tau)d\tau, \\ \frac{dy}{dt} = -b\int_0^t y(t-\tau)y(\tau)d\tau. \end{cases}$
- $\frac{dx}{dt} = -a\int_0^t x(t-\tau)x(\tau)^2 d\tau$

Сравнить с результатами аналитического решения.

2. Методом конечных разностей численно решить следующее интегро-дифференциальное уравнение

$$\ddot{\Phi}(t) + \gamma\dot{\Phi}(t) + \Omega_0^2\Phi(t) + 4\lambda\Omega_0^2\int_0^t d\tau M(\tau)\dot{\Phi}(t-\tau) = 0,$$

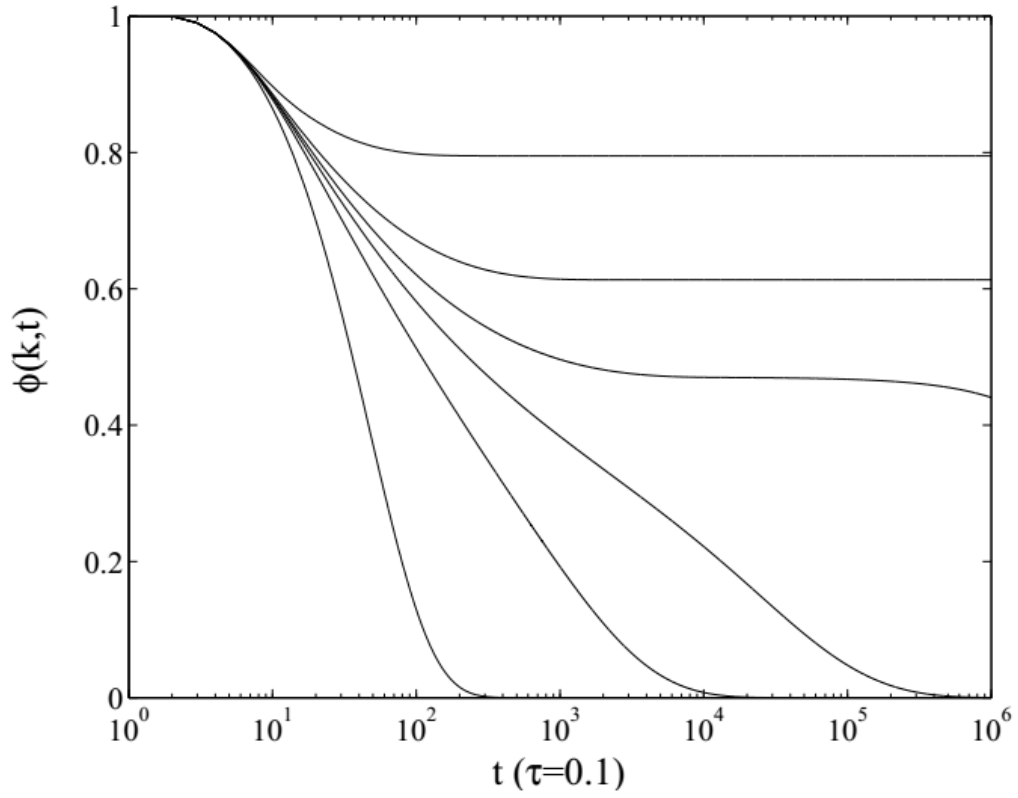
$$M(\tau) = a\Phi(\tau) + b\Phi^2(\tau),$$

с начальными условиями: $\Phi(t=0)=1$, $\dot{\Phi}(t=0)=0$, $\Omega_0^2=1$, $\gamma=1$, $0 \le \lambda \le 1.5$. Исследовать полученное решение при различных значениях параметров.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
g=1.0; om=1.9; lambda=1.3;
a1=0; a2=3.0;
x(1)=1; y(1)=0;
dt=0.1;
for i=1:1:50000
    p=0;
    for j=1:1:i
        p=p+(a1*x(j)+a2*x(j)^2)*y(i-j+1)*dt;
    end
    z(i)=-g*y(i)-om*x(i)-om*lambda*p;
    y(i+1)=z(i)*dt+y(i);
    x(i+1)=y(i+1)*dt+x(i);
end
semilogx((1:length(x)),x(1:length(x)),'-k')
ylim([0 1])
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§2. Задачи механики

1. Построить фигуры Лиссажу (зависимости *y=y(x)*), которые получаются при сложении двух взаимно перпендикулярных колебаний различной частоты

$$\begin{cases} x = X_m \cos(\omega_1 t), \\ y = Y_m \cos(\omega_2 t + \varphi) \end{cases}$$

для следующих случаев:

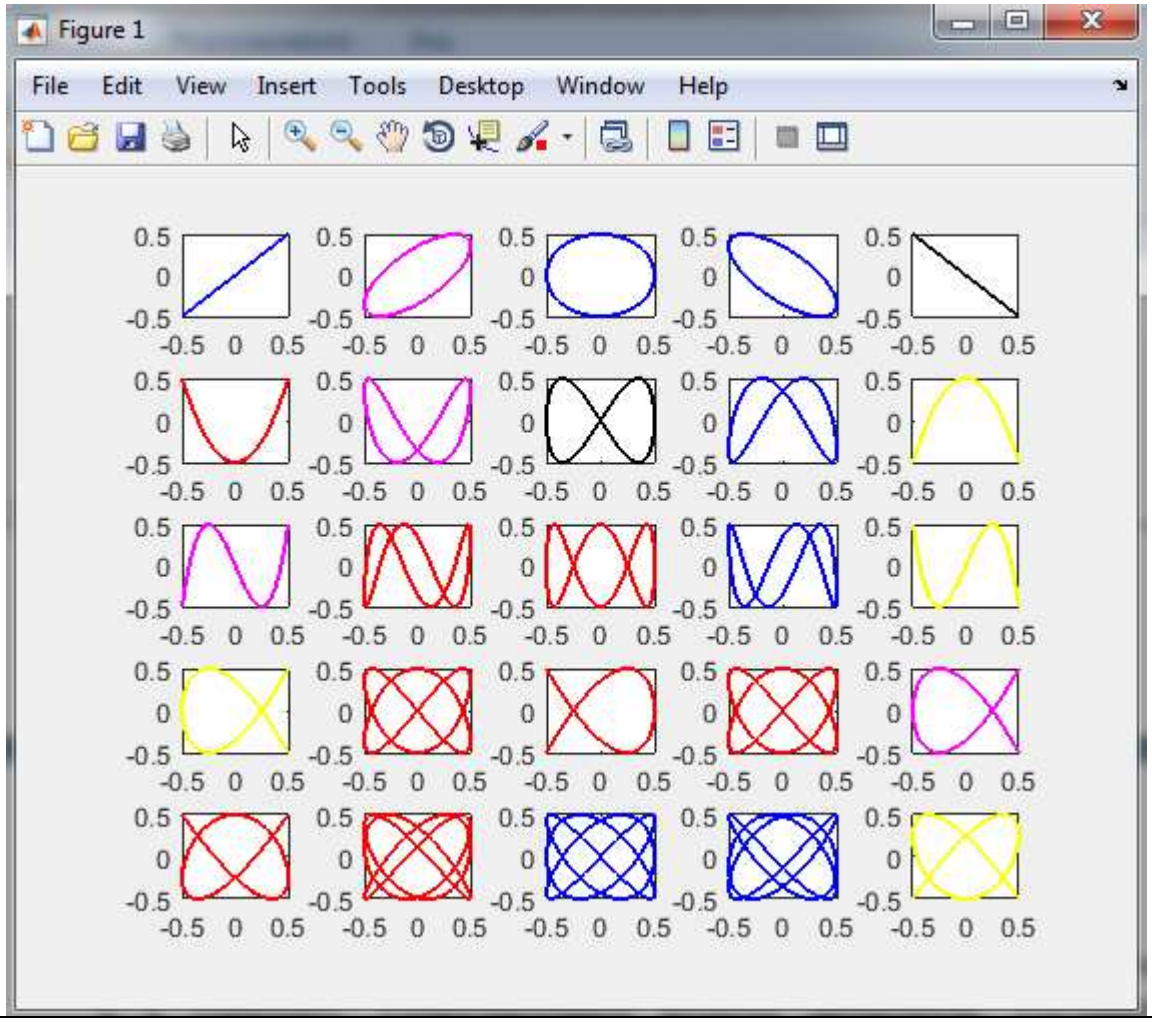
	$\varphi = 0$	$\varphi = \pi/4$	$\varphi = \pi/2$	$\varphi = 3\pi/4$	$\varphi = \pi$
$\omega_1 / \omega_2 = 1:1$					
$\omega_1 / \omega_2 = 1:2$					
$\omega_1 / \omega_2 = 1:3$					
$\omega_1 / \omega_2 = 2:3$					
$\omega_1 / \omega_2 = 3:4$					

Указание: при построении графиков использовать команду *subplot*

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
phi=[0, pi/4, pi/2, 3*pi/4, pi];
kw=[1, 2, 3, 3/2, 4/3];
color='gbmyrk';
Xm=0.5; Ym=0.5; w1=0.5; N=1000; dt=0.1;
t(1:N,1)=zeros(N,1);
X(1:N,1)=zeros(N,1);
Y(1:N,1)=zeros(N,1);
kol=0;
for i=1:1:5
    w2=kw(i)*w1;
    for j=1:1:5
        for k=1:1:N
            t(k)=dt*(k-1);
            X(k)=Xm*cos(w1*t(k));
            Y(k)=Ym*cos(w2*t(k)+phi(j));
        end
        kol=kol+1; pp=1+randi(5);
        subplot(5,5,kol);
        plot(X,Y,color(pp),'LineWidth',1.5);
    end
end
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

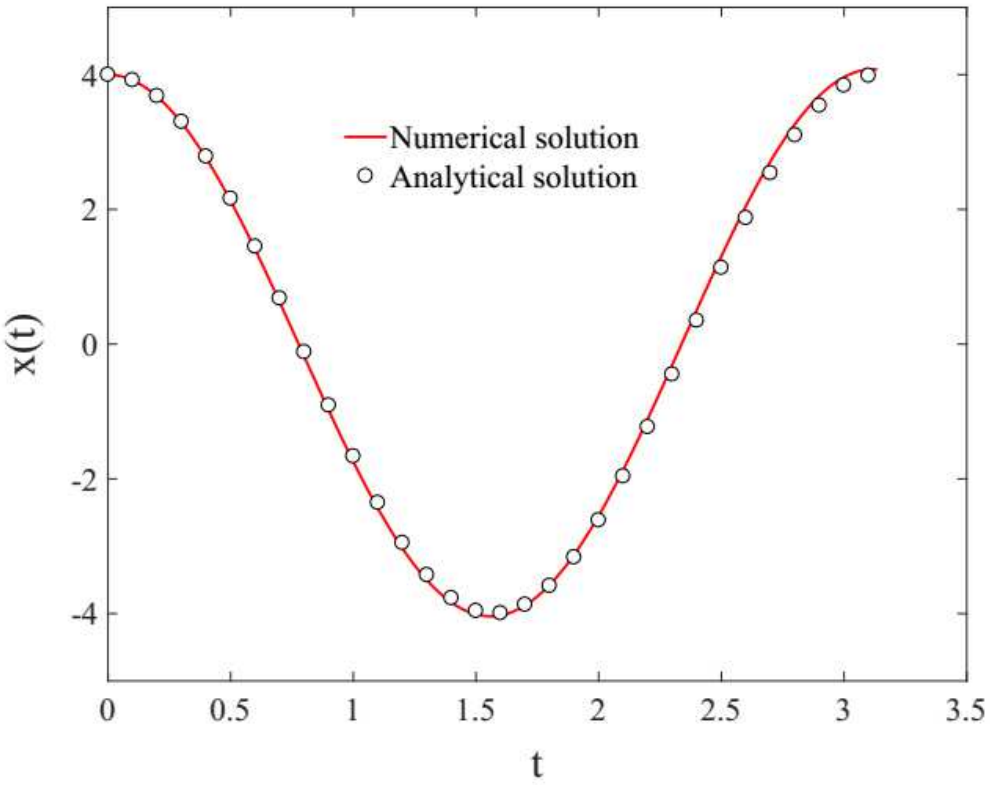


2. Написать программу моделирования свободных одномерных колебаний. Сравнить численное решение с аналитическим решением.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
a=4.0; om=2.0;
%% Numerical solution
t(1)=0; t(2)=(1/1000)*pi;
y(1)=a; y(2)=a;
d2t=0.00001;
for i=3:1000
    t(i)=(i-2)*pi/1000;
    y(i)=2*y(i-1)-y(i-2)*(1+om^2*d2t);
end
hold on; plot(t,y,'-r')
%% Analytical solution
x=0:0.1:pi; hold on; plot(x,a*cos(om*x),'ko')
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



3. Снаряд массой *m* имеет начальную скорость $\vec{\vartheta}_0$, направленную под углом α к горизонту. Сопротивлением воздуха пренебречь. Составить математическую модель задачи, разработать алгоритм и написать программу. Определить и проанализировать траекторию движения тела.
4. Тело массой *m* движется с начальной скоростью $\vec{\vartheta}_0$ под углом α к горизонту в поле силы тяжести и сопротивления воздуха $\vec{R} = -\gamma\vec{\vartheta}$, пропорциональной первой степени скорости. Составить математическую модель задачи, разработать алгоритм и написать программу. Определить и проанализировать траекторию движения тела.
5. Построить график изменения массы ракеты при вертикальном подъеме в однородном поле тяжести: а) при $\vec{\vartheta} = const$; б) при $\vec{a} = const$. Силой сопротивления воздуха пренебречь. Выполнить численный и теоретический расчет. Сравнить и проанализировать полученные результаты.

6. Ракета движется в однородном поле тяжести вверх с постоянной скоростью $\vec{\vartheta} = const$. Сила сопротивления воздуха $\vec{R} = -\gamma\vec{\vartheta}$, пропорциональна первой степени скорости. Определить закон изменения массы ракеты. Выполнить численный и теоретический расчет. Сравнить и проанализировать полученные результаты.
7. Анизотропный заряженный пространственный осциллятор находится в однородном магнитном поле H; требуется воспроизвести траектории осциллятора.

   Пространственный осциллятор описывается следующими уравнениями движения

$$\begin{cases} \ddot{x} = -\omega_x^2 x + \dfrac{eH}{mc}\dot{y}, \\ \ddot{y} = -\omega_y^2 y - \dfrac{eH}{mc}\dot{x}, \\ \ddot{z} = -\omega_z^2 z. \end{cases}$$

   Рассмотреть случаи, когда $\omega_x = \omega_y = \omega_y$, $\omega_x = \omega_y \neq \omega_y$ и $\omega_x \neq \omega_y \neq \omega_y$.
8. Решить предыдущую задачу при наличии диссипации энергии в системе

$$\begin{cases} \ddot{x} = -\omega_x^2 x + \dfrac{eH}{mc}\dot{y} - \alpha\dot{x}, \\ \ddot{y} = -\omega_y^2 y - \dfrac{eH}{mc}\dot{x} - \alpha\dot{y}, \\ \ddot{z} = -\omega_z^2 z. \end{cases}$$

   Рассмотреть случаи, когда $\omega_x = \omega_y = \omega_y$, $\omega_x = \omega_y \neq \omega_z$ и $\omega_x \neq \omega_y \neq \omega_z$ при различных значениях α.

9. Смоделировать задачу взаимодействия видов (система "хищник-жертва", "паразит-хозяин" и т.п.), описываемая моделью Лотки-Вольтерры:

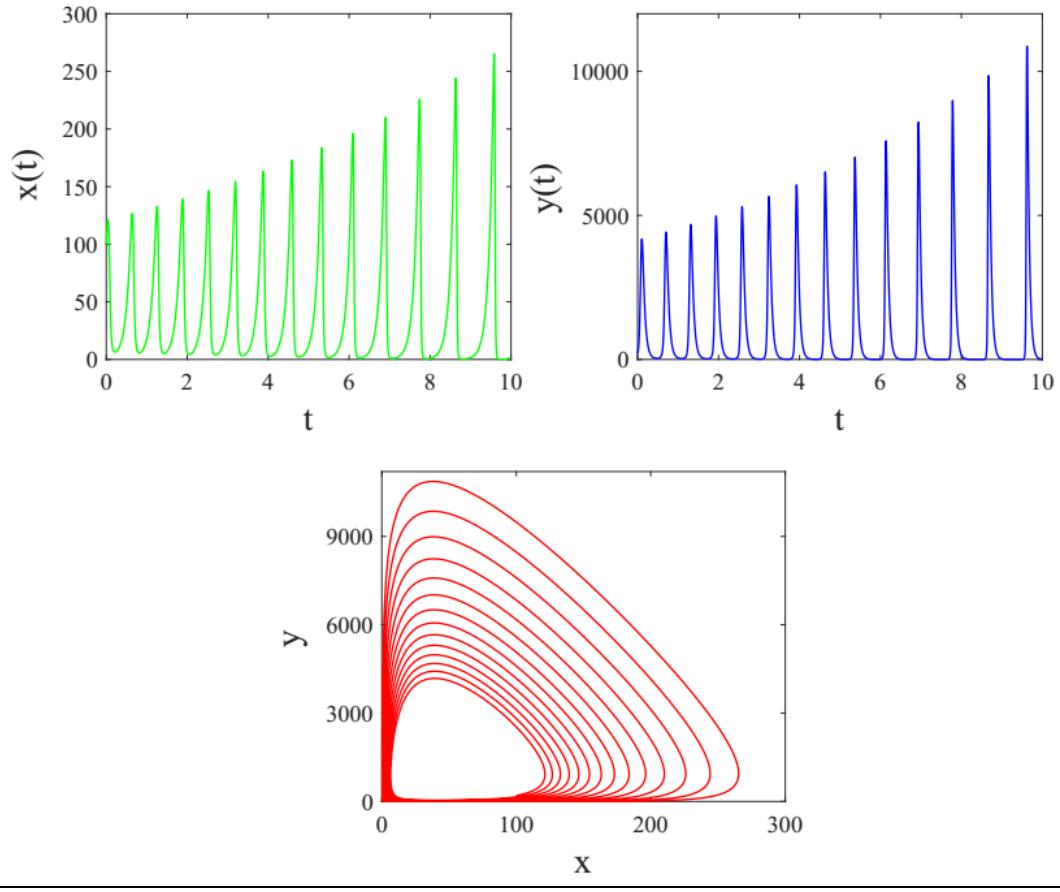
$$\begin{cases} \dfrac{dx}{dt} = (\alpha - \beta y)x, \\ \dfrac{dy}{dt} = (-\gamma + \delta x)y. \end{cases}$$

Здесь x - количество жертв, y - количество хищников, t - время, $\alpha, \beta, \gamma, \delta$ - коэффициенты, отражающие взаимодействия между видами. Построить зависимости *x(t), y(t)* и *y(x)*.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
alpha=9;   beta=0.01;
delta=0.5; gamma=20;
dt=0.001;  N=10000;
t(1:N,1)=zeros(N,1);
x(1:N,1)=zeros(N,1);
y(1:N,1)=zeros(N,1);
x(1)=100; y(1)=200;
for i=2:1:N
    t(i)=(i-1)*dt;
    x(i)=(alpha-beta*y(i-1))*dt*x(i-1)+x(i-1);
    y(i)=(-gamma+delta*x(i-1))*dt*y(i-1)+y(i-1);
end
subplot(2,2,1); plot(t,x,'-g')
subplot(2,2,2); plot(t,y,'-b')
subplot(2,2,3); plot(x,y,'-r')
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§3. Колебания в электрических цепях

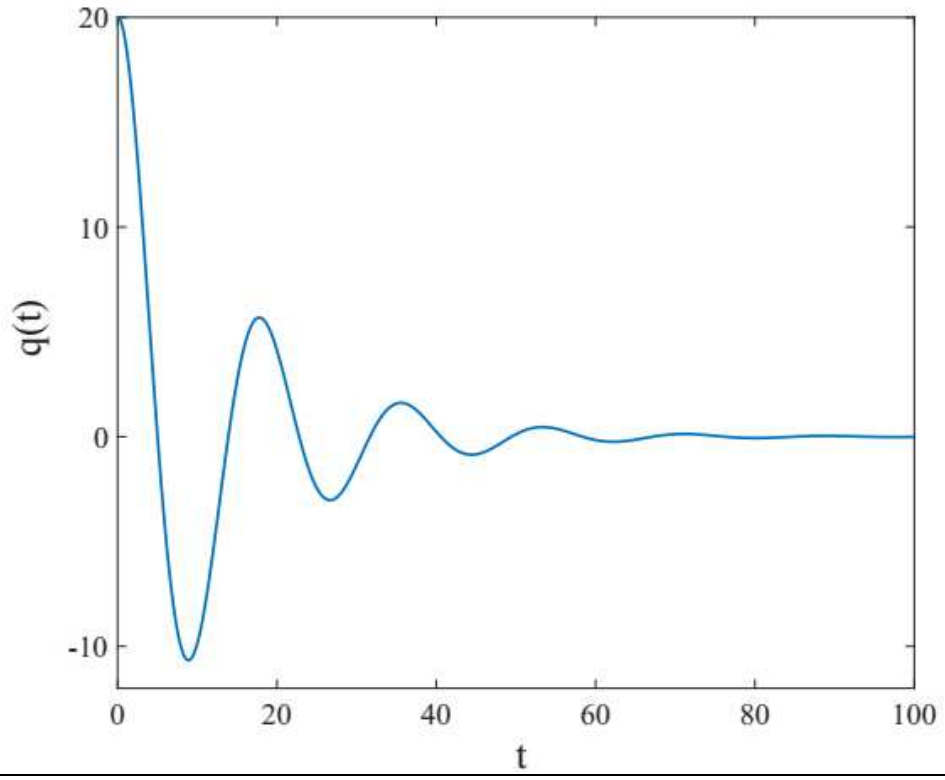
1. Разработать алгоритм и написать программу, реализующую решение дифференциальных уравнений, описывающих процессы в электрическом RLC-колебательном контуре (последовательное соединение) $\frac{d^2I(t)}{dt^2}+\frac{R}{L}\frac{dI(t)}{dt}+\frac{1}{LC}I(t)=0$. Здесь *I(t)* - есть закон изменения тока в цепи, R - сопротивление резистора, L - индуктивность катушки и C - емкость конденсатора. Выполнить сравнение численных и аналитических решений. Найти условия сходимости (расходимости) решений.

2. Разработать алгоритм и написать программу, реализующую решение дифференциальных уравнений, описывающих вынужденные колебания в RLC-контуре (последовательное соединение) $L\frac{d^2q(t)}{dt^2}+R\frac{dq(t)}{dt}+\frac{1}{C}q(t)=U_m\cos(\omega t)$. Здесь *q(t)* - есть закон изменения заряда в цепи, R - сопротивление резистора, L - индуктивность катушки и C - емкость конденсатора. Выполнить сравнение численных и аналитических решений. Найти условия сходимости (расходимости) решений. Определите условие возникновения резонанса токов в цепи. Рассмотреть случай параллельного соединения.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
R=0.5; L=3.5; C=2.2;
dt=0.01; N=10000;
q(1:N,1)=zeros(N,1);
q(1)=20; q(2)=20;
t(2)=dt;
for i=3:1:N
    t(i)=(i-1)*dt;
    q(i)=2*q(i-1)-q(i-2)-(R/L)*(q(i-1)
         -q(i-2))*dt-1/(C*L)*q(i-2)*dt^2;
end; plot(t,q)
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§4. Задачи ядерной физики

1. Численно решить уравнение радиоактивного распада:

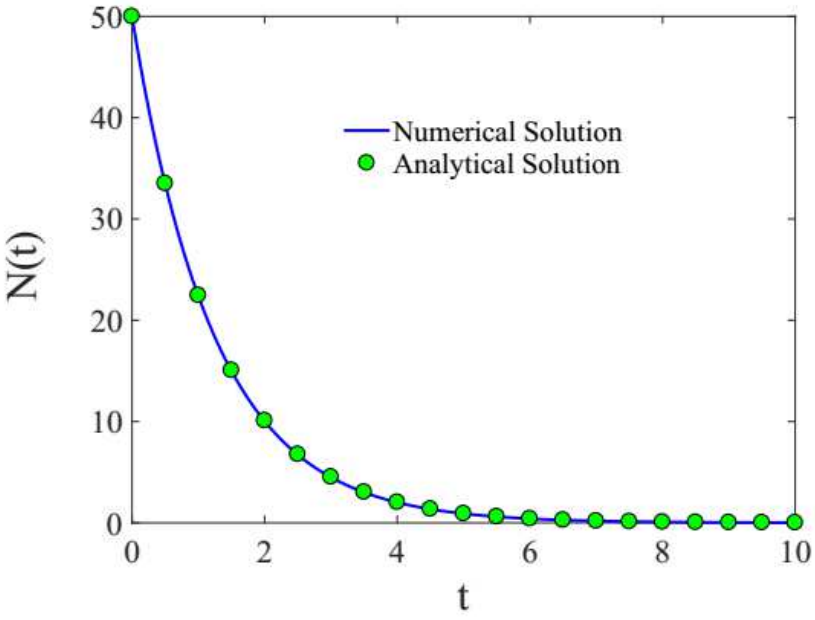
$$\frac{dN}{dt} = -\lambda N .$$

Разработать алгоритм решения задачи и написать программу на языке программирования MatLab. Сравнить численное решение с аналитическим решением. Определить условия сходимости.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
lambda=0.8; N0=50;
n=1000; Nt=10; dt=Nt/n;
t(1:n,1)=zeros(n,1);
N(1:n,1)=zeros(n,1);
N(1)=N0;
for i=2:1:n
    t(i)=(i-1)*dt;
    N(i)=N(i-1)*(1-lambda*dt);
end
hold on; plot(t, N,'-b')

t=0:50*dt:Nt;
N=N0.*exp(-lambda.*t);
hold on; plot(t, N,'ko'); box on;
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



2. Численно решить уравнение *разветвленного* радиоактивного распада:

$$\frac{dN}{dt} = -(\lambda_A - \lambda_B)N\,.$$

Разработать алгоритм решения задачи и написать программу на языке программирования MatLab. Определить временные зависимости количества изотопов $N_A(t)$ и $N_B(t)$.

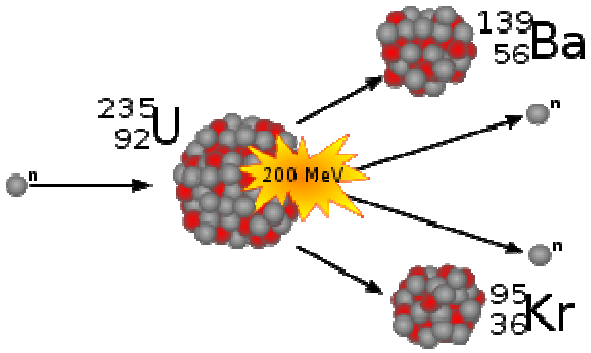
3. Численно решить уравнения последовательного распада (радиоактивное семейство):

$$\begin{cases} \dfrac{dN_A}{dt} = -\lambda_A N_A, \\ \dfrac{dN_B}{dt} = -\lambda_B N_B + \lambda_A N_A. \end{cases}$$

Разработать алгоритм решения задачи и написать программу на языке программирования MatLab. Построить зависимости $N_A(t)$ и $N_B(t)$. Рассмотреть случаи:

- $N_A(0) \neq 0$, $N_B(0) = 0$.
- $N_A(0) = 0$, $N_B(0) \neq 0$.
- $N_A(0) \neq 0$, $N_B(0) \neq 0$.

4. Смоделируйте процесс радиоактивного распада ядер урана в реакторе. Реактор имеет форму куба с конечными размерами. Реактор содержит несколько неподвижных радиоактивных ядер; несколько нуклонов (протонов и нейтронов), которые свободно двигаются по области реактора с постоянной скоростью. Нуклоны отражаются от поверхности реактора в соответствии с зеркальными граничными условиями. При неупругом взаимодействии нейтрона с радиоактивным ядром происходит распад с образованием двух нерадиоактивных осколков и 2-3 нейтронов (*примечание:* используйте датчик случайных чисел), которые способны вызвать новый радиоактивный распад. Нерадиоактивные осколки упруго взаимодействуют с нуклонами без дальнейшего распада ядер. При упругом взаимодействии протонов с ядром распад не наблюдается.



Рассчитайте за все время моделирования количество энергии, которое выделилось в процессе работы реактора.

§5. Метод наименьших квадратов

1. Разработать алгоритм и написать программу, реализующую подгонку экспериментальных данных степенными зависимостями. Написать процедуру определения параметров степенной зависимости по методу $\chi^2 = \sum_{i=1}^{N} \left(y(i) - y_{\exp}(i) \right)^2$ с точностью $\varepsilon = 0.01$. Здесь $y = \sum_{i=1}^{N} a_{i-1} x^{i-1}$ - полином N-1 порядка.

а)

x(i)	0	1	2	3	4	5
y_{exp}(i)	2.3500	13.5500	43.7500	92.9500	161.1500	248.3500

б)

x(i)	0	0.5	1.0	1.5	2.0	2.5	3.0
y_{exp}(i)	0	-2.4375	-9.75	-21.9375	-39.0	-60.9375	-87.7500

с)

x(i)	1	2	3	4	5	6	7
y_{exp}(i)	2.08	-2.29	-6.66	-11.03	-15.4	-19.77	-24.14

Решаем задачу методом наименьших квадратов

Пусть имеем набор экспериментальных данных:

x(i)	0	1	2	3	4	5
y_{exp}(i)	2.3500	13.5500	43.7500	92.9500	161.1500	248.3500

Экспериментальные данные будем аппроксимировать полиномиальной функцией вида:

$$y = \sum_{i=1}^{N} a_{i-1} x^{i-1} \tag{1}$$

Введем невязку

$$f = \sum_{i=1}^{N} \left(y_i^{\exp} - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3 - a_4 x_i^4 - .... \right)^2 \tag{2}$$

Коэффициенты полинома будем искать как

$$a_k : \quad \frac{\partial f}{\partial a_k} = 0 \tag{3}$$

тогда,

$$\begin{cases} \dfrac{\partial f}{\partial a_0} = -2\sum\limits_{i=1}^{N} \left(y_i^{\exp} - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3 - a_4 x_i^4 - .... \right) = 0 \\ \dfrac{\partial f}{\partial a_1} = -2\sum\limits_{i=1}^{N} \left(- x_i \right) \cdot \left(y_i^{\exp} - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3 - a_4 x_i^4 - .... \right) = 0 \\ \dfrac{\partial f}{\partial a_2} = -2\sum\limits_{i=1}^{N} \left(- x_i^2 \right) \cdot \left(y_i^{\exp} - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3 - a_4 x_i^4 - .... \right) = 0 \\ \dfrac{\partial f}{\partial a_3} = -2\sum\limits_{i=1}^{N} \left(- x_i^3 \right) \cdot \left(y_i^{\exp} - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3 - a_4 x_i^4 - .... \right) = 0 \\ ..... \end{cases}$$

Находим коэффициенты уравнения, решая систему уравнений методом Крамера:

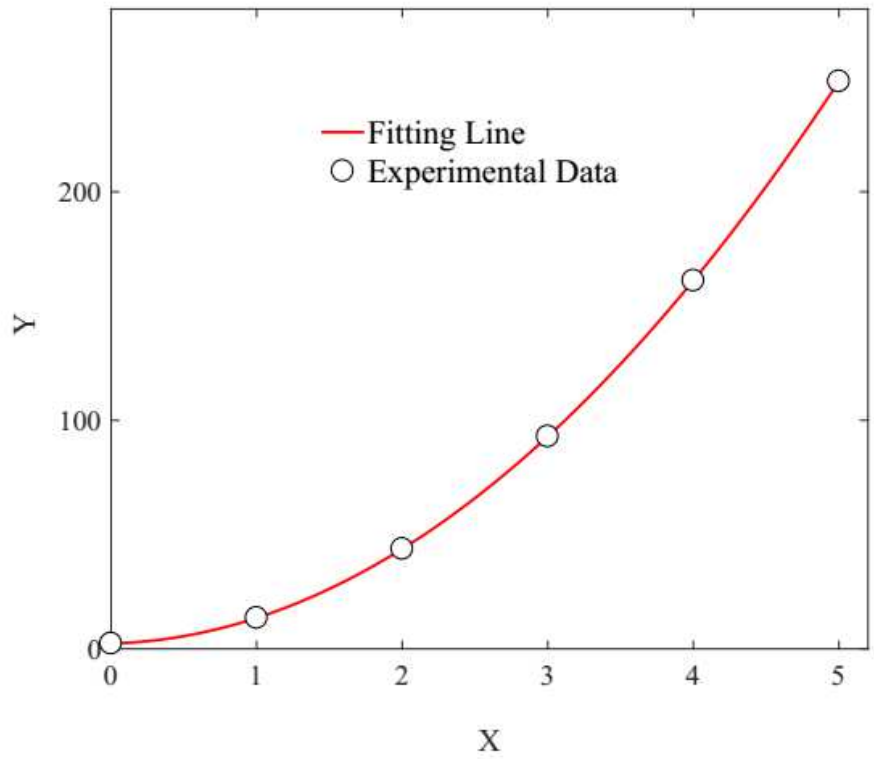
$A = \begin{pmatrix} \sum\limits_i^N x_i^0 & \sum\limits_i^N x_i^1 & \sum\limits_i^N x_i^2 \\ \sum\limits_i^N x_i^1 & \sum\limits_i^N x_i^2 & \sum\limits_i^N x_i^3 \\ \sum\limits_i^N x_i^2 & \sum\limits_i^N x_i^3 & \sum\limits_i^N x_i^4 \end{pmatrix}$	$B = \begin{pmatrix} \sum\limits_i^N y_i \\ \sum\limits_i^N x_i y_i \\ \sum\limits_i^N x_i^2 y_i \end{pmatrix}$

$$a_k = \Delta_k / \Delta$$

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
x=[0    1     2     3     4       5       ];
y=[2.35 13.55 43.75 92.95 161.15 248.35 ];
plot(x,y,'ko');

len=length(x); n=len-1;
A(1:n,1:n)=zeros(n,n);
B(1:n,1)=zeros(n,1);
for i=1:1:n
    for k=1:1:len
       B(i)=B(i)+y(k)*x(k)^(i-1);
    end
    for j=1:1:n
        for k=1:1:len
            A(i,j)=A(i,j)+x(k)^(i+j-2);
        end
    end
end
A0=A; A1=A; A2=A; A3=A; A4=A;

A0(:,1)=B; A1(:,2)=B; A2(:,3)=B; A3(:,4)=B;
A4(:,5)=B;
a0=det(A0)/det(A);
a1=det(A1)/det(A);
a2=det(A2)/det(A);
a3=det(A3)/det(A);
a4=det(A4)/det(A);

dx=(x(len)-x(1))/100;
p=x(1):dx:x(len);
z=a0+a1.*p+a2.*p.^2+a3.*p.^3+a4.*p.^4;
hold on; plot(p,z,'r-'); box on;
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§6. Псевдослучайные числа

1. Построить график функции распределения случайной величины в диапазоне [0, 1], генерируемой встроенным датчиком (функция ***rand()***).
2. Разработать алгоритм и написать программу, генерирующую последовательность случайных чисел. Построить ее функцию распределения. Написать программу, генерирующую последовательность случайных чисел:
   - линейным конгруэнтным методом $x_n = (ax_{n-1} + c) \bmod m$;
   - алгоритмом Блюма-Блюма-Шуба

     а) $x_n = (x_{n-1})^2 \bmod m$ б) $x_n = (x_0)^{2n \bmod (p-1)(q-1)} \bmod pq$
3. Разработать свой алгоритм и написать программу, генерирующую случайные числа в диапазоне [0, 1]. Выполнить сравнительный анализ разработанного метода с встроенным в компилятор ГСЧ.
4. Построить генератор случайных чисел, распределенных по следующему закону:

1) $p(x) = \begin{cases} 2 - 3x, & 0 \le x \le 1 \\ 0 \end{cases}$

2) $p(x) = \begin{cases} a^{-x}, & 0 \le x \le 1 \\ 0 \end{cases}$

3) $p(x) = \begin{cases} \dfrac{1}{2x+3}, & -1 \le x \le 1 \\ 0 \end{cases}$

4) $p(x) = \begin{cases} \dfrac{x^2+1}{3+x}, & 0 \le x \le 2 \\ 0 \end{cases}$

5) $p(x) = \begin{cases} xe^x, & 0 \le x \le \pi \\ 0 \end{cases}$

6) $p(x) = \begin{cases} \cos^2(x), & 0 \le x \le \pi \\ 0 \end{cases}$

7) $p(x) = \begin{cases} e^{x^2}, & 0 \le x \le 1 \\ 0 \end{cases}$

8) $p(x) = \begin{cases} erf(x), & -1 \le x \le 1 \\ 0 \end{cases}$

§7. Метод Монте-Карло

1. Вычислить значение числа π методом Монте-Карло с точностью 0.0001.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
N=0; M=0;
Pi=0; eps=0.00001;
a=1;
while (abs(Pi-pi)>eps)
    x=a*rand(); y=a*rand();
    N=N+1;
    R=sqrt(x^2+y^2);
    if (R<=a)
        M=M+1;
    end
    Pi=4.0*M/N;
    fprintf(1,'N=%d, M=%f, Pi=%f\n',N, M, Pi)
end
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

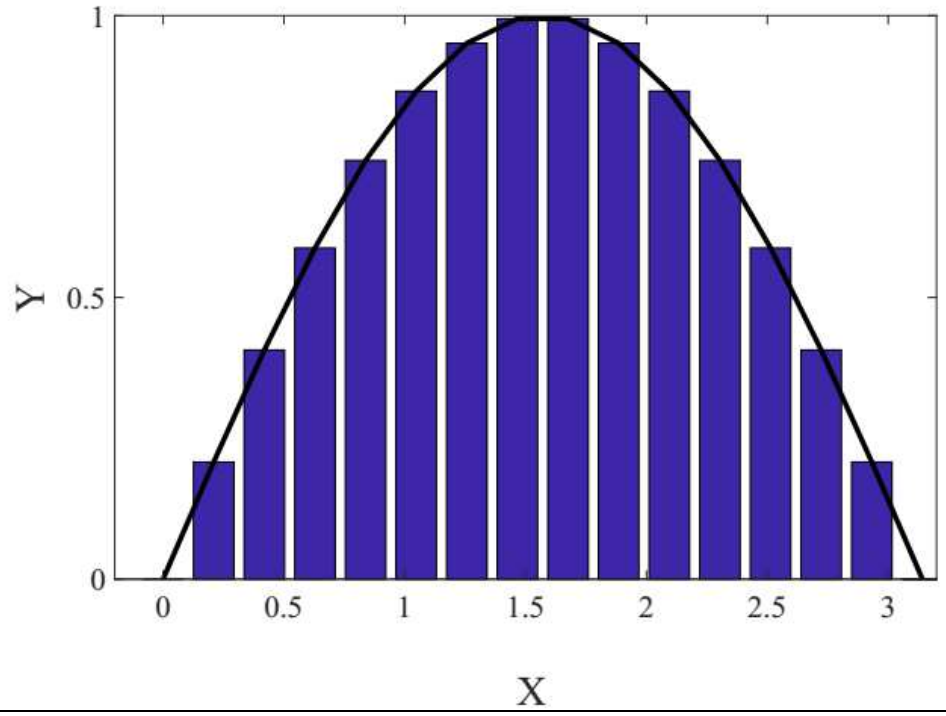
```
N=1, M=1.000000, Pi=4.000000
N=2, M=2.000000, Pi=4.000000
N=3, M=3.000000, Pi=4.000000
N=4, M=4.000000, Pi=4.000000
N=5, M=5.000000, Pi=4.000000
N=6, M=6.000000, Pi=4.000000
N=7, M=7.000000, Pi=4.000000
N=8, M=8.000000, Pi=4.000000
N=9, M=8.000000, Pi=3.555556
N=10, M=9.000000, Pi=3.600000
N=11, M=10.000000, Pi=3.636364
N=12, M=11.000000, Pi=3.666667
N=13, M=12.000000, Pi=3.692308
N=14, M=13.000000, Pi=3.714286
N=15, M=14.000000, Pi=3.733333
N=16, M=15.000000, Pi=3.750000
....
....
N=14736, M=11573.000000, Pi=3.141422
N=14737, M=11574.000000, Pi=3.141481
N=14738, M=11575.000000, Pi=3.141539
N=14739, M=11576.000000, Pi=3.141597
```

2. Вычислить методом Монте-Карло площадь фигуры, ограниченной половиной синусоиды.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
N=1000; M=0;
for i=1:1:N
    x=pi*rand();
    y=rand();
    if (y<sin(x))
        M=M+1;
    end
    S=pi*M/i;
    fprintf(1,'N=%d, M=%d, S=%d\n', i, M, S)
end

N=15; dx=pi/N; x=0:dx:pi;
y=sin(x); bar(x,y)
hold on; plot(x,y,'-k','LineWidth',4)
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



3. Написать программу для вычисления методом Монте-Карло площади S тела, ограниченного кривыми $xy = a$ и $x + y = \frac{5}{2}a$. Сравнить результат с точным значением.

4. Вычислить методом Монте-Карло площадь фигуры, ограниченной параболой $y = x^2$, осью абсцисс и прямой $x = 2$.
5. Методом Монте-Карло оценить определенный интеграл
   - $I = \int_1^3 (x+1)dx$.
   - $I = \int_0^{\pi/2} \cos(x)dx$.
   - $I = \int_0^1 e^x dx$.
   - $I = \int_0^2 (4 - x^2)dx$.
   - $I = \int_0^1 dx \int_x^1 (x + y)dy$.
   - $I = \int_0^2 dx \int_{x^2}^{2x} (x + y)dy$.
6. Вычислить определенный интеграл методом Монте-Карло для N=10000, 100000 и 1000000. Сравнить результат интегрирования с точным решением и с результатом интегрирования методом прямоугольников. Оценить погрешности интегрирования.

➢ $\int_2^3 \sqrt{\frac{3-2x}{2x-7}}dx$	➢ $\int_0^{\pi} \frac{\cos(x)}{2+\cos(x)}dx$	➢ $\int_{-4}^{-2} x\cos^2(x)dx$
➢ $\int_0^7 \frac{\sqrt{x+25}}{(x+25)^2\sqrt{x+1}}dx$	➢ $\int_1^4 \frac{1-\sqrt{x}}{\sqrt{x}(x+1)}dx$	➢ $\int_0^1 5\sqrt{\frac{6-x}{x-18}}dx$
➢ $\int_{-2}^{-1} (x^2+5x+6)\cos(2x)dx$	➢ $\int_1^e \frac{dx}{x\sqrt[4]{\ln(x)}}$	➢ $\int_0^{2\ln(3)} \frac{dx}{ch^2(x/2)}$
➢ $\int_{-3/2}^{-3/5} (x^2-4)\cos(3x)dx$	➢ $\int_1^2 (3-7x)\cos(2x)dx$	➢ $\int_1^e x\ln(x)dx$

7. *Игла Бюффона.* Стол разграфлен параллельными линиями на расстоянии *2a*, на стол случайным образом бросается игла длиной *2L*, *L<a*. Какова вероятность того, что игла пересечет какую-то линию?

§8. Моделирование движения частиц

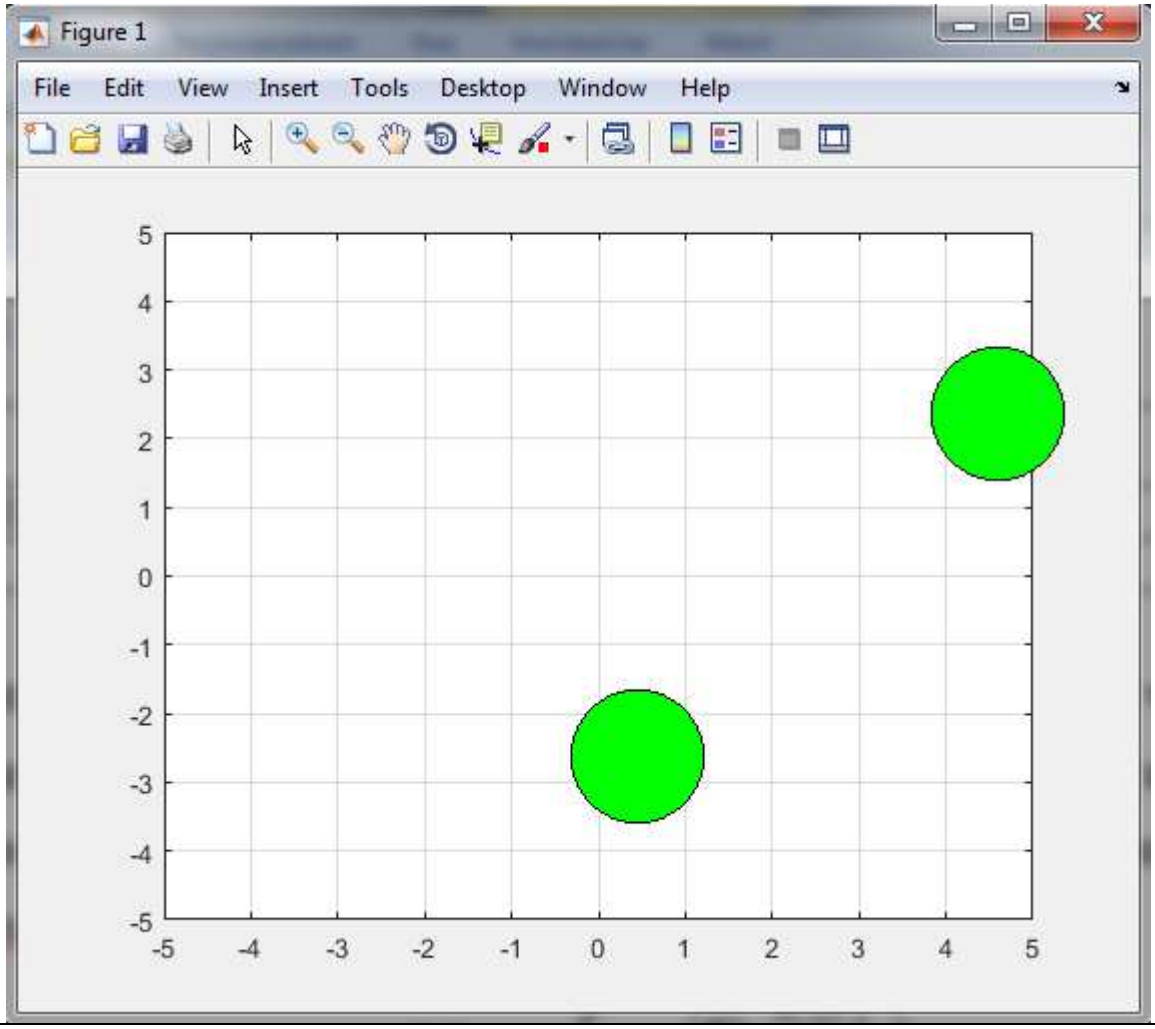
1. Смоделируйте движение броуновской частицы. Получите на экране траекторию броуновской частицы.
2. Смоделируйте динамику двух упругих шаров (дисков) с
   - периодическими граничными условиями;
   - с зеркальными граничными условиями.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
a=5; v=2.5; dt=0.1;
N=10000; M=10; eps=1.5;
dr=2*a/M;
r1(1:N,1:2)=zeros(N,2);
r2(1:N,1:2)=zeros(N,2);

r1(1,1)=a*(rand()-rand());
r1(1,2)=a*(rand()-rand());

r2(1,1)=a*(rand()-rand());
r2(1,2)=a*(rand()-rand());

vx1=v*(rand()-rand()); vy1=v*(rand()-rand());
vx2=v*(rand()-rand()); vy2=v*(rand()-rand());

for i=2:1:N
   r1(i,1)=r1(i-1,1)+vx1*dt;
   r1(i,2)=r1(i-1,2)+vy1*dt;
   r2(i,1)=r2(i-1,1)+vx2*dt;
   r2(i,2)=r2(i-1,2)+vy2*dt;
   if (abs(r1(i,1))>=a)
       vx1=-vx1;
   end
   if (abs(r1(i,2))>=a)
       vy1=-vy1;
   end
   if (abs(r2(i,1))>=a)
       vx2=-vx2;
   end
   if (abs(r2(i,2))>=a)
       vy2=-vy2;
   end
   if (abs(r1(i,1)-r2(i,1))<=eps)
      && (abs(r1(i,2)-r2(i,2))<=eps)
```

```
        t1=vx1;    t2=vy1;
        vx1=vx2;   vy1=vy2;
        vx2=t1;    vy2=t2;
    end
    plot(r1(i,1),r1(i,2),'ko',r2(i,1),r2(i,2),'ko',
            'MarkerFaceColor','g','MarkerSize',50);
    box on; grid on; xlim([-a a]); ylim([-a a]);
    pause(0.01);
end
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

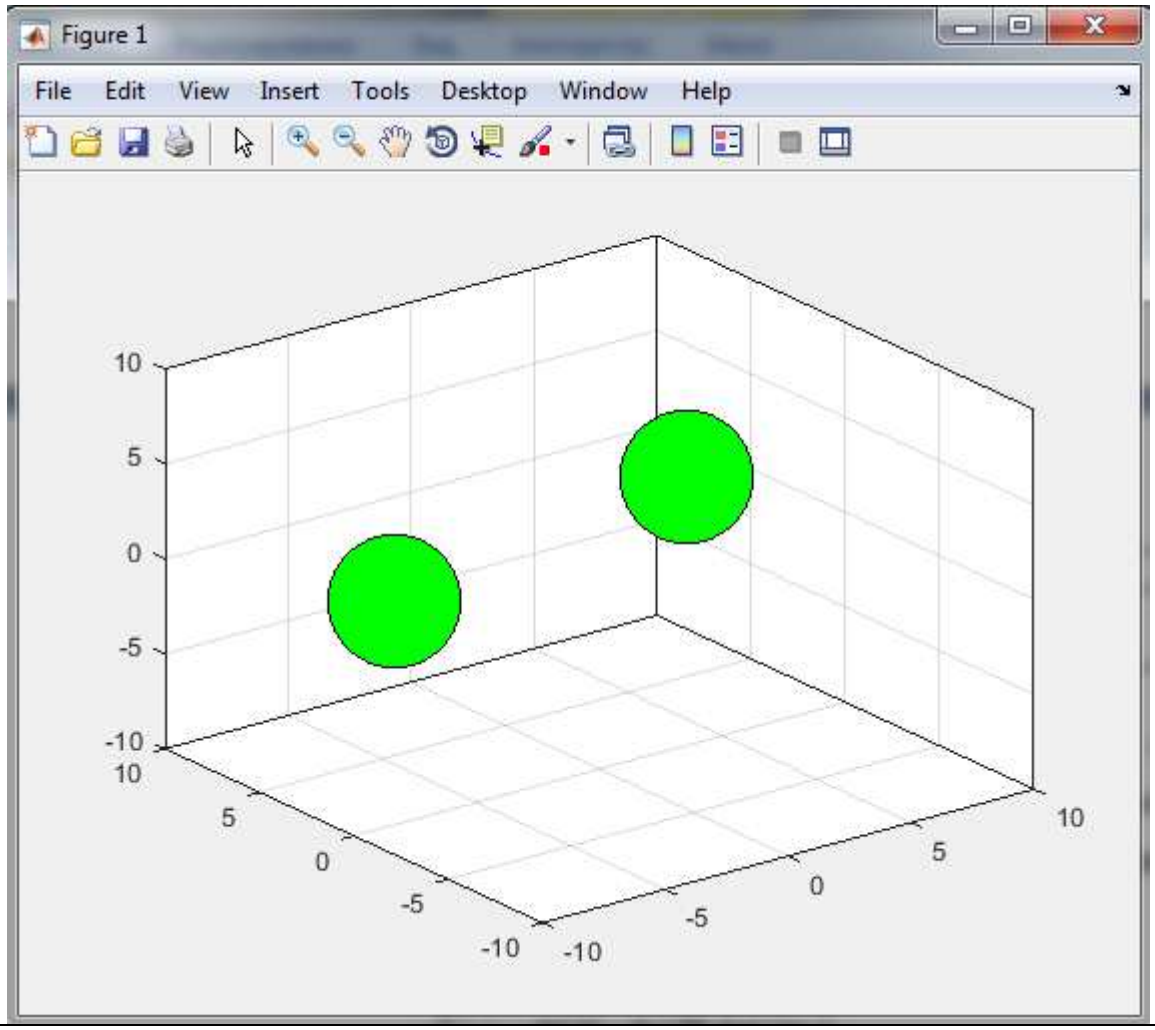


ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
a=10; v=2.5; dt=0.1;
N=10000; M=10; eps=1.5; dr=2*a/M;
r1(1:N,1:3)=zeros(N,3);
r2(1:N,1:3)=zeros(N,3);
```

```
v1(1:3,1)=zeros(3,1);
v2(1:3,1)=zeros(3,1);

for k=1:1:3
    r1(1,k)=a*(rand()-rand());
    v1(k)=v*(rand()-rand());
    r2(1,k)=a*(rand()-rand());
    v2(k)=v*(rand()-rand());
end

for i=2:1:N
    for k=1:1:3
        r1(i,k)=r1(i-1,k)+v1(k)*dt;
        r2(i,k)=r2(i-1,k)+v2(k)*dt;
        if (abs(r1(i,k))>=a)
            v1(k)=-v1(k);
        end
        if (abs(r2(i,k))>=a)
            v2(k)=-v2(k);
        end
    end
    if (abs(r1(i,1)-r2(i,1))<=eps)
        && (abs(r1(i,2)-r2(i,2))<=eps)
        && (abs(r1(i,3)-r2(i,3))<=eps)
        for k=1:1:3
            t=v1(k); v1(k)=v2(k); v2(k)=t;
        end
   end
   plot3(r1(i,1),r1(i,2),r1(i,3),'ko',
         r2(i,1),r2(i,2),r2(i,3),'ko',
         'MarkerFaceColor','g','MarkerSize',50);
   box on; xlim([-a a]); ylim([-a a]); zlim([-a a]);
   grid on; axis on; pause(0.01);
end
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



3. Написать программу, реализующую расположение частиц в моделируемой ячейке с ПК-структурой (структурой простой кубической решетки). Рассчитать для полученной конфигурации частиц двухчастичную радиальную функцию распределения частиц

$$g(r) = \frac{V}{N(N-1)} \left\langle \sum_{j=1}^{N} \frac{n_j(r)}{4\pi r^2 \Delta r} \right\rangle.$$

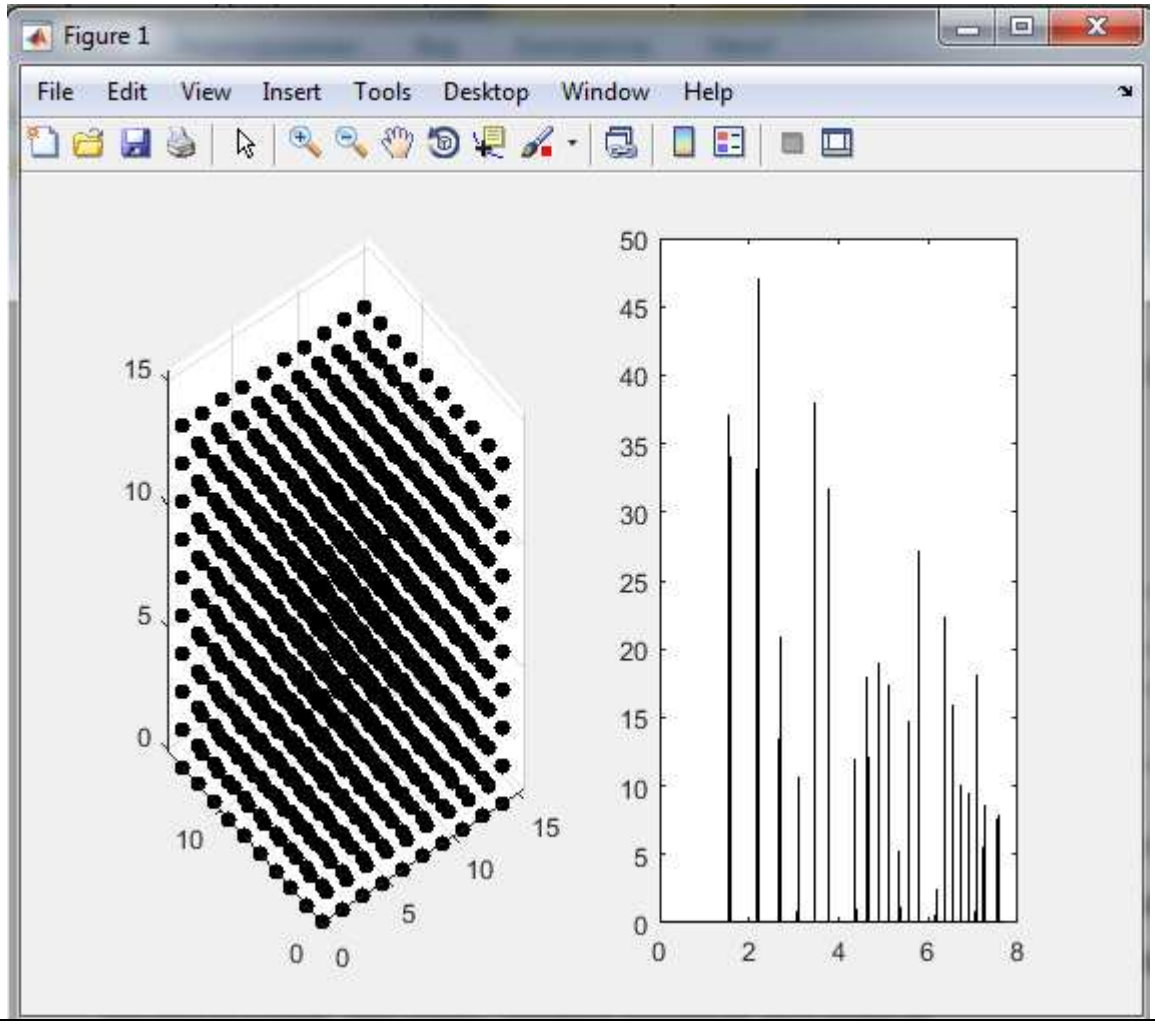
Здесь V - объем моделируемой системы, величина $n_j(r)$ определяет число частиц в сферическом слое толщиной Δr на расстоянии r друг от друга.

ЛИСТИНГ ПРОГРАММЫ

```
close all;
clear;
nx=10; ny=10; nz=10;
a=1.54;
nparts=nx*ny*nz;
pos(1:nparts,1:3)=zeros(nparts,3);
box(1:3,1)=zeros(3,1);
box(1)=nx*a; box(2)=ny*a; box(3)=nz*a;
kol=0;
for i=1:1:nx
    for j=1:1:ny
        for k=1:1:nz
            kol=kol+1;
            pos(kol,1)=(i-1)*a;
            pos(kol,2)=(j-1)*a;
            pos(kol,3)=(k-1)*a;
        end
    end
end
subplot(1,2,1);
scatter3(pos(:,1),pos(:,2),pos(:,3),'ko','filled');
xlim([0 box(1)]); ylim([0 box(2)]); zlim([0 box(3)]);

N=1000; dr=box(1)/N;
hist(1:N,1:2)=zeros(N,2);
dx(1:3,1)=zeros(3,1);
for i=1:1:nparts-1
    for j=i+1:1:nparts
        rij=0;
        for k=1:1:3
            dx(k)=pos(i,k)-pos(j,k);
            if abs(dx(k))>0.5*box(k)
                dx(k)=dx(k)*(1-box(k)/abs(dx(k)));
            end
            rij=rij+dx(k)^2;
        end
        rij=sqrt(rij);
        k=fix(rij/dr+1);
        hist(k,2)=hist(k,2)+1;
    end
end

kol=0;
for k=1:1:N
```

```
    if (k*dr+dr/2)<=box(1)/2
        kol=kol+1;
        hist(k,1)=k*dr+dr/2;

hist(k,2)=2.0*hist(k,2)*box(1)*box(2)*box(3)/(nparts*
(nparts-1)*4*pi*hist(k,1)^2*dr);
    end
end
subplot(1,2,2); plot(hist(1:kol,1),
                     hist(1:kol,2),'-k')
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



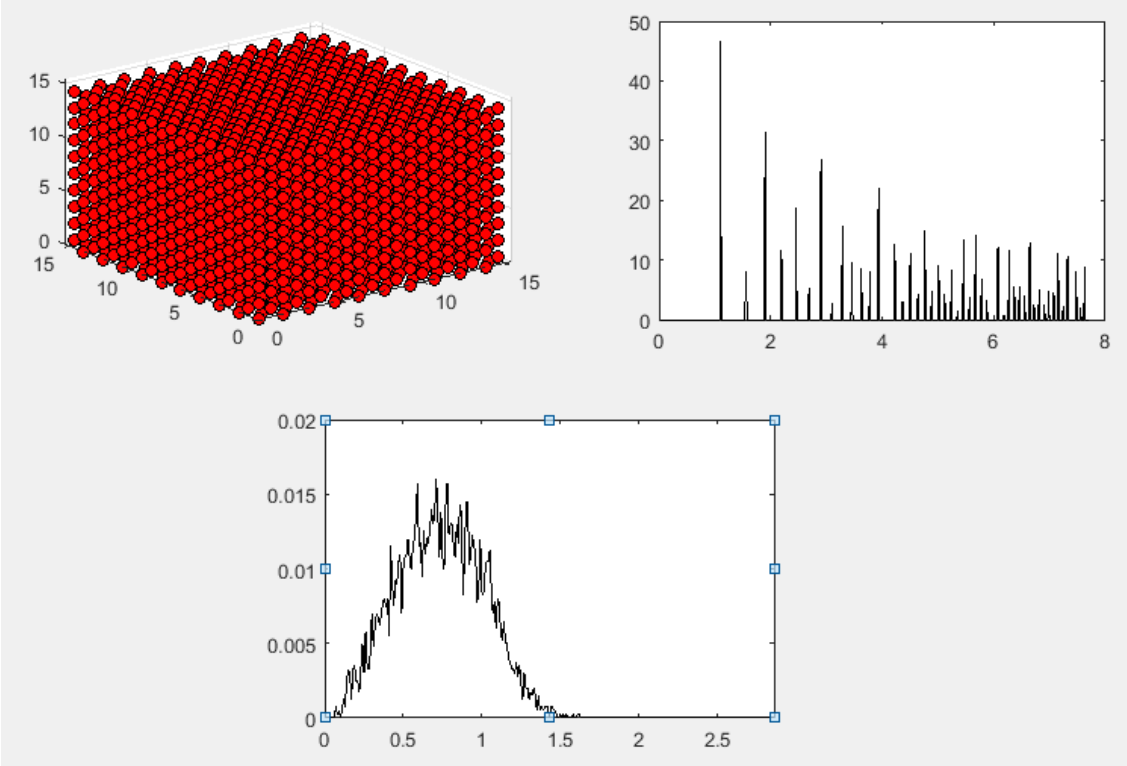
4. Написать программу инициализации скоростей частиц в соответствии с распределением Максвелла-Больцмана по скоростям и случайным распределением. Построить график функции соответствующих распределений. Численно оцените площадь под кривой.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
nx=10; ny=10; nz=10; a=1.54;
nparts=4*nx*ny*nz;
pos(1:nparts,1:3)=zeros(nparts,3);
box(1:3,1)=zeros(3,1);
box(1)=nx*a; box(2)=ny*a; box(3)=nz*a;
nbase=4;
rcell=[0.0 0.0 0.0;
       0.5 0.5 0.0;
       0.0 0.5 0.5;
       0.5 0.0 0.5];
kol=0;
for k=0:1:nz-1
    for j=0:1:ny-1
        for i=0:1:nx-1
            for L=1:1:nbase
                kol=kol+1;
                pos(kol,1)=a*(i+rcell(L,1));
                pos(kol,2)=a*(j+rcell(L,2));
                pos(kol,3)=a*(k+rcell(L,3));
            end
        end
    end
end
subplot(2,2,1);
scatter3(pos(:,1),pos(:,2),pos(:,3),'ko','filled');
xlim([0 box(1)]); ylim([0 box(2)]); zlim([0 box(3)]);

N=1000; dr=box(1)/N;
hist(1:N,1:2)=zeros(N,2);
dx(1:3,1)=zeros(3,1);
for i=1:1:nparts-1
    for j=i+1:1:nparts
        rij=0;
        for k=1:1:3
            dx(k)=pos(i,k)-pos(j,k);
            if abs(dx(k))>0.5*box(k)
                dx(k)=dx(k)*(1-box(k)/abs(dx(k)));
            end
            rij=rij+dx(k)^2;
        end
        rij=sqrt(rij);
        k=fix(rij/dr+1);
        hist(k,2)=hist(k,2)+1;
```

```
    end
end

kol=0;
for k=1:1:N
    if (k*dr+dr/2)<=box(1)/2
        kol=kol+1;
        hist(k,1)=k*dr+dr/2;

hist(k,2)=2.0*hist(k,2)*box(1)*box(2)*box(3)/(nparts*
(nparts-1)*4*pi*hist(k,1)^2*dr);
    end
end
subplot(2,2,2); plot(hist(1:kol,1),
                     hist(1:kol,2),'-k')

kB=1; T=1.2;  m=1; v=sqrt(3*kB*T/m)
vel(1:nparts,1:3)=sqrt(1/3)*(v*rand(nparts,3)
                  -v*rand(nparts,3));
Temp=0;
for i=1:1:nparts
    for k=1:1:3
        Temp=Temp+vel(i,k)^2;
    end
end
Temp=Temp*m/(3*kB*nparts);

hist=zeros(1000,1);
dv=0.01;
for i=1:1:nparts
    Temp=0;
    for k=1:1:3
        Temp=Temp+vel(i,k)^2;
    end
    Temp=sqrt(Temp);
    hist(fix(Temp/dv)+1)=hist(fix(Temp/dv)+1)+1;
end
subplot(2,2,3); plot((1:length(hist))*dv,
                hist(1:length(hist))./nparts,'-k')
xlim([0 3*Temp])
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



5. Разработайте алгоритм и напишите программу моделирования динамики системы упругих шаров (N=100) с периодическими граничными условиями. Скорости должны быть распределены в соответствии с распределением Максвелла-Больцмана по скоростям.
6. Написать программу, реализующую расчет среднеквадратического смещения частиц:

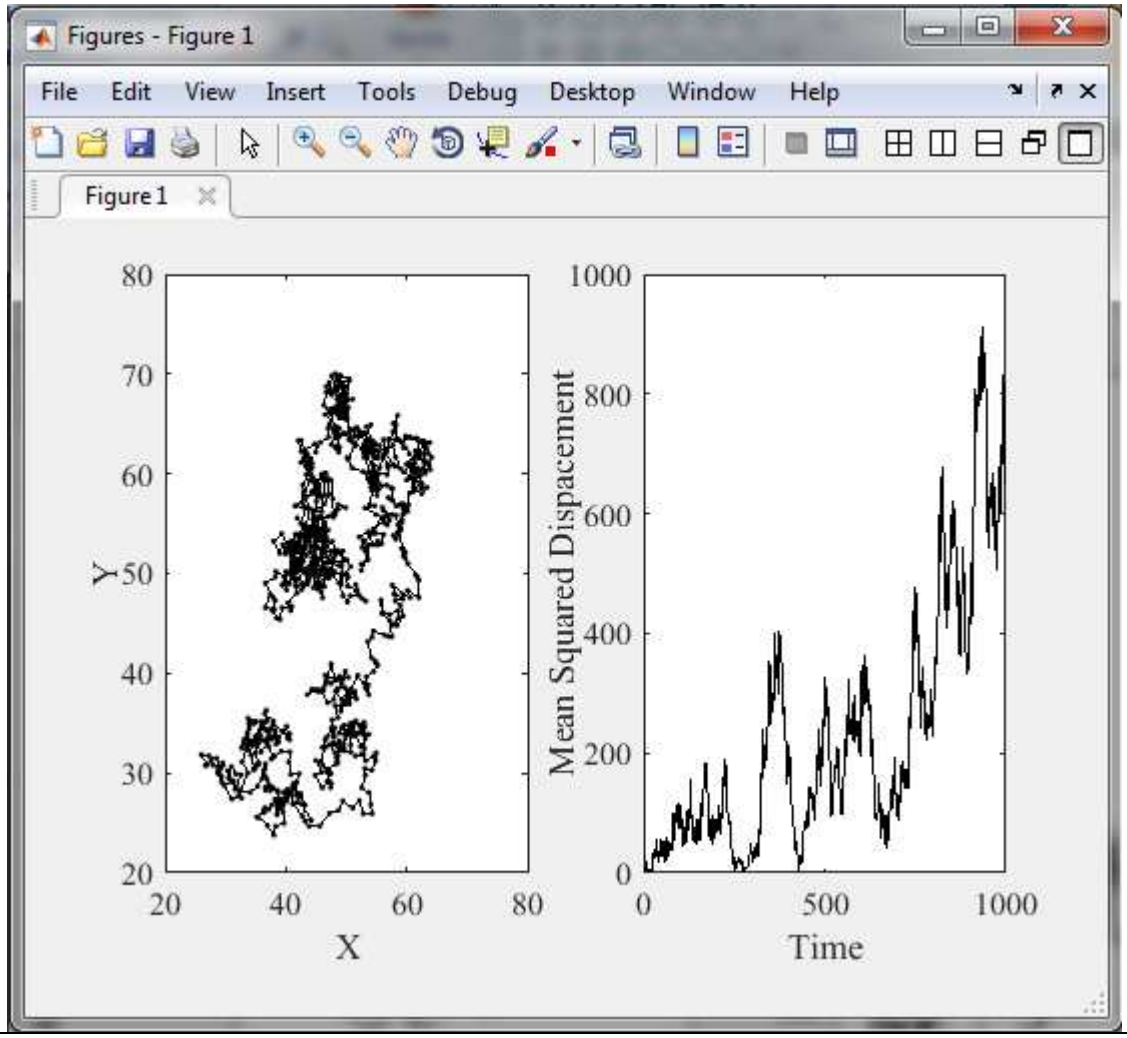
$$\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle = \frac{1}{N}\sum_{i=1}^{N}\left(\left[x_i(t)-x_i(0)\right]^2+\left[y_i(t)-y_i(0)\right]^2+\left[z_i(t)-z_i(0)\right]^2\right)$$

и коэффициента самодиффузии (по формуле А. Эйнштейна) $D=\lim_{t\to\infty}\frac{\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle}{6\cdot t}$. Проанализировать графики при различных термодинамических состояниях системы.

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
a=100;
rectangle('Position',[0 0 a a],'FaceColor',
    [0.95 .95 .98],'EdgeColor','k','LineWidth',2)
N=1000; dx=2;
pos(1:N,1:2)=zeros(N,2);
msd(1:N,1:2)=zeros(N,2);
pos(1,1)=50; pos(1,2)=50;
for i=2:1:N
    for k=1:1:2
        x=pos(i-1,k)+dx*(rand()-rand());
        if ((x>0) && (x<a))
            pos(i,k)=x;
        end
    end
    msd(i,1)=i;
    msd(i,2)=msd(i,2)+(pos(i,1)-
pos(1,1))^2+(pos(i,2)-pos(1,2))^2;
end
subplot(1,2,1); hold on;
plot(pos(:,1), pos(:,2),'k.-'); ; box on;
subplot(1,2,2); hold on;
plot(msd(:,1), msd(:,2),'k-'); box on;
xlabel('Time','FontName','Times New Roman',
                                    'FontSize', 14)
ylabel('Mean Squared Dispacement','Fontname',
       'Times New Roman','Fontsize',14)
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§9. Моделирование динамики частиц методом Монте-Карло

1. Разработать алгоритм и написать программу, реализующую моделирование жидкости с помощью потенциала Леннарда-Джонса

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

методом Монте-Карло по схеме Метрополиса (в NVT и NpT-ансамблях). Здесь σ - есть эффективный размер частицы, ε - глубина потенциальной ямы.

2. Для полученной конфигурации расположения частиц системы рассчитать двухчастичную радиальную функцию распределения частиц

$$g(r) = \frac{V}{N(N-1)}\left\langle \sum_{j=1}^{N} \frac{n_j(r)}{4\pi r^2 \Delta r} \right\rangle.$$

Здесь V - объем моделируемой системы, величина $n_j(r)$ определяет число частиц в сферическом слое толщиной Δr на расстоянии r друг от друга.

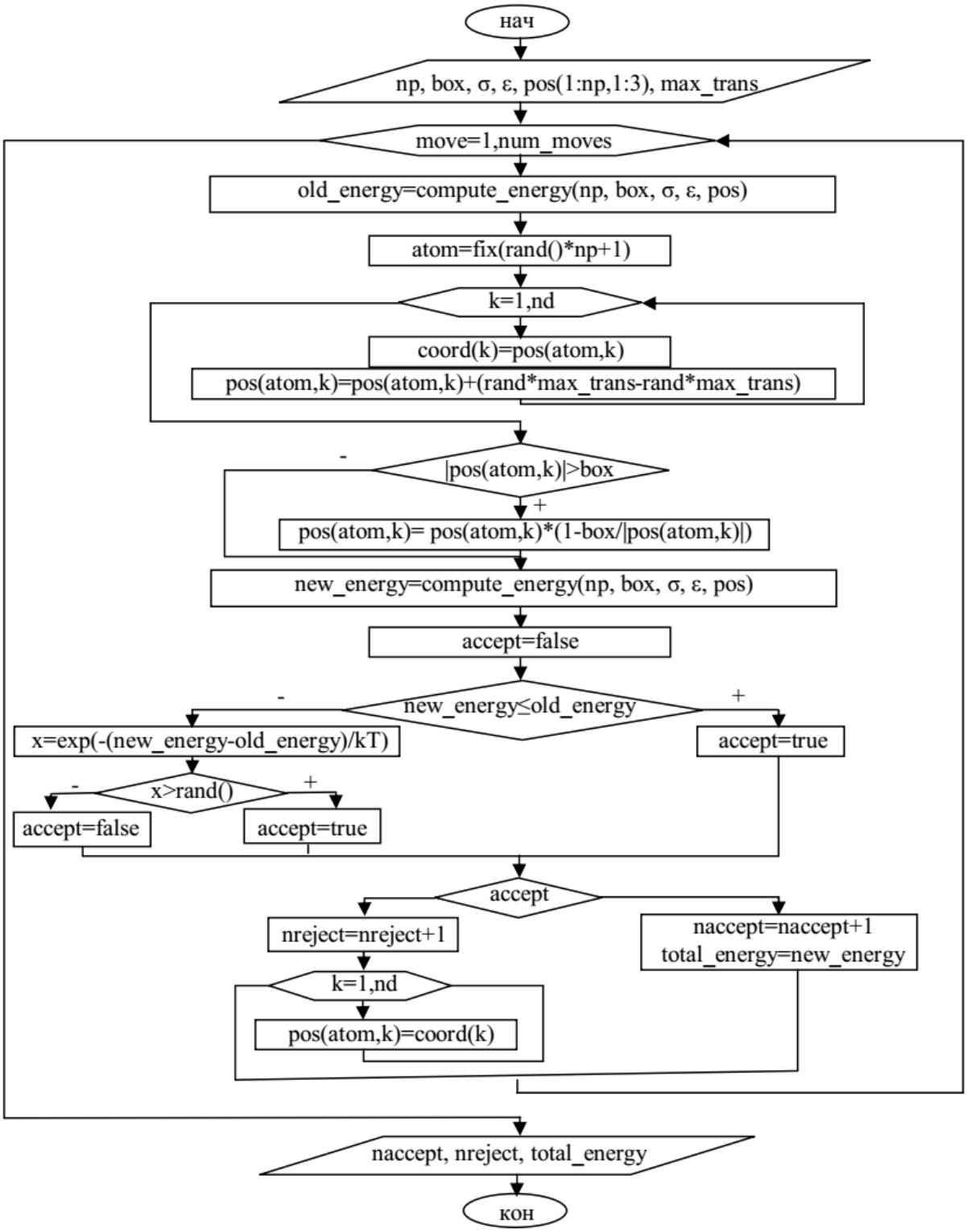
3. Постройте среднеквадратичное смещение частиц со временем, $\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle$.

С помощью формулы Эйнштейна оцените значение коэффициента самодиффузии

$$D = \lim_{t\to\infty} \frac{\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle}{6 \cdot t}.$$

Блок-схема алгоритм метода Монте-Карло

по схеме Метрополиса



ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
nd=3; a_lat=1.54; nbase=4;
nx=4; ny=4; nz=4;
np=4*nx*ny*nz;
boxsize(1)=a_lat*nx;
boxsize(2)=a_lat*ny;
boxsize(3)=a_lat*nz;
pos(1:np,1:nd)=zeros(np,nd);
rcell=[0.0 0.0 0.0,
       0.5 0.5 0.0,
       0.0 0.5 0.5,
       0.5 0.0 0.5];
kol=0;
for k=0:1:nz-1
    for j=0:1:ny-1
        for i=0:1:nx-1
            for L=1:1:nbase
                kol=kol+1;
                pos(kol,1)=a_lat*(i+rcell(L,1));
                pos(kol,2)=a_lat*(j+rcell(L,2));
                pos(kol,3)=a_lat*(k+rcell(L,3));
            end
        end
    end
end
subplot(1,2,1); scatter3(pos(:,1), pos(:,2),
                    pos(:,3), 200, 'bo', 'filled');
num_moves=500;
max_translate=0.5;
T=298.15;
sigma=3.405;
kB=1.38*10^(-23);
epsilon=120*kB;

naccept=0; nreject=0;
total_energy=0;
coord(1:nd)=zeros(nd,1);
for  move=1:1:num_moves
    old_energy=calculate_energy(pos, np, boxsize,
sigma, epsilon);
    atom=fix(rand*np+1);
    for k=1:1:nd
        coord(k)=pos(atom,k);
        pos(atom,k)=pos(atom,k)
```

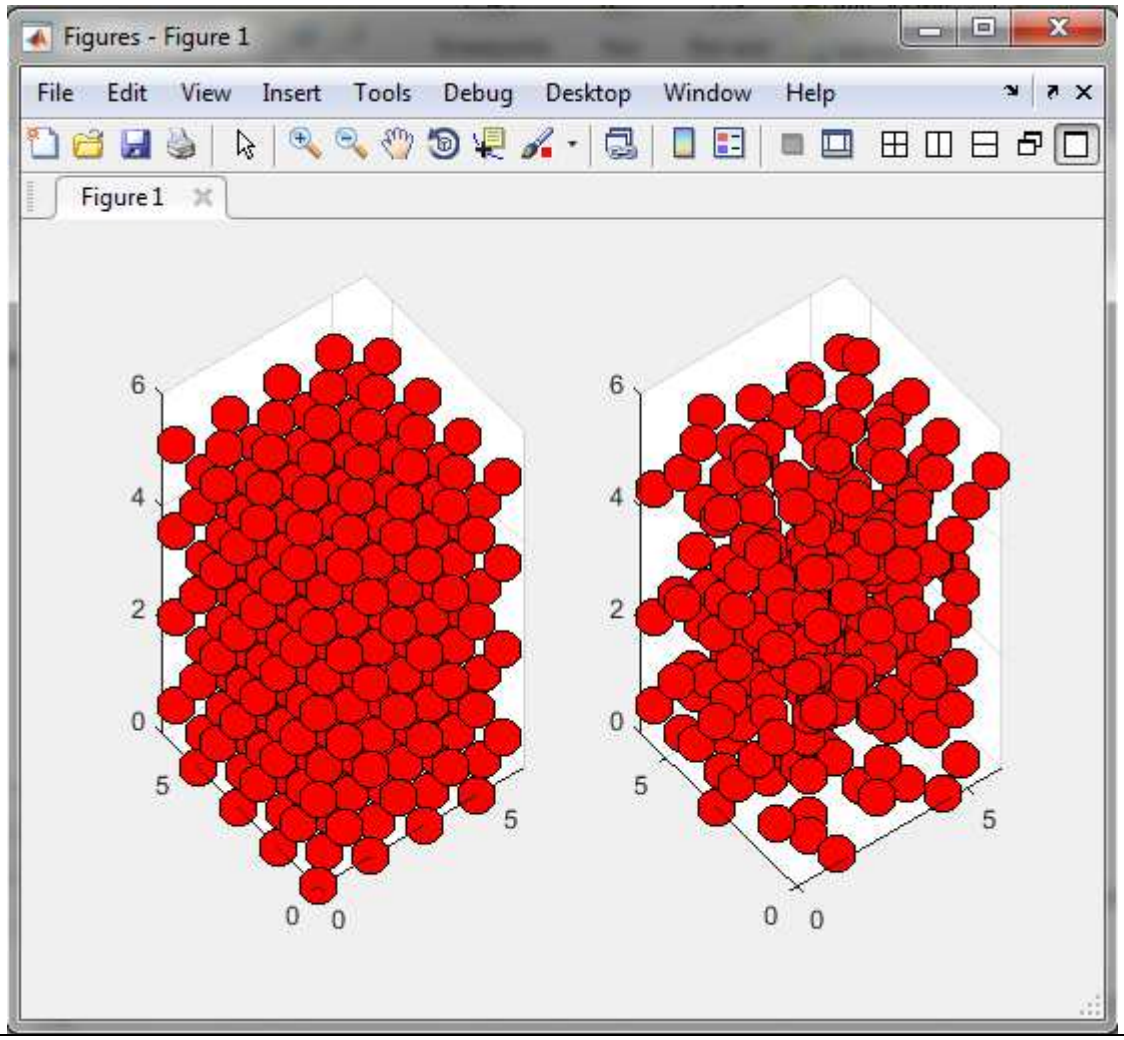
```
            +(rand*max_translate-rand*max_translate);
        if abs(pos(atom,k))>boxsize(k)
            pos(atom,k)=pos(atom,k)
                *(1-boxsize(k)/abs(pos(atom,k)));
        end
    end
    new_energy=calculate_energy(pos, np, boxsize,
                                sigma, epsilon);
    accept=false;
    if new_energy<=old_energy
        accept=true;
    else
        x=exp(-(new_energy-old_energy)/kB*T);
        if x>=rand
            accept=true;
        else
            accept=false;
        end
    end
    if accept
        naccept=naccept+1;
        total_energy=new_energy;
    else
        nreject=nreject+1;
        for k=1:1:nd
            pos(atom,k)=coord(k);
        end
    end
    fprintf( 1, '  Step = %d, Total Energy = %d.\n',
              move, total_energy);
end
fprintf ( 1, '  Naccept = %d, Nreject = %d.\n',
            naccept, nreject);
subplot(1,2,2); scatter3(pos(:,1), pos(:,2),
pos(:,3), 200, 'bo', 'filled');
```

ЛИСТИНГ ФУНКЦИИ РАСЧЕТА ПОТЕНЦИАЛЬНОЙ ЭНЕРГИИ

```
function [energy]=calculate_energy(pos, np, boxsize,
                                   sigma, epsilon)
    energy = 0;
    for i=1:1:np-1
        rij=0;
        for j=(i+1):1:np
            for k=1:1:3
```

```
                    dx(k)=pos(j,k)-pos(i,k);
                    if abs(dx(k))>0.5*boxsize(k)
                        dx(k)=dx(k)*
                                (1-boxsize(k)/abs(dx(k)));
                    end
                    rij=rij+dx(k)^2;
                end
                rij=sqrt(rij);
                energy=energy+4*epsilon*((sigma/rij)^(12)
                                        -(sigma/rij)^6);
            end
        end
        energy=0.5*energy;
end
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



§10. Моделирование динамики частиц методом молекулярной динамики

1. Написать программу, реализующую моделирование жидкости с помощью потенциала Леннарда-Джонса методом молекулярной динамики в NVT-ансамбле. Исследуйте, как быстро система приходит в состояние равновесия в зависимости от значений Nsteps, Δt и начальных значений скоростей.
2. Для некоторой *j*-той частицы постройте график функции $r_j(t)$. Нарисуйте траекторию движения этой частицы $y(x)$ и $z(x, y)$. Сравните результаты для жидкой и твердой фаз.
3. Разработайте алгоритм и напишите программу, реализующую расчет функции распределения по скоростям. Постройте график зависимости $f(\vartheta)$. Сравните с распределением Максвелла-Больцмана по скоростям. Численно оцените площадь под кривой.
4. Написать программу, реализующую расчет радиальной функции распределения частиц

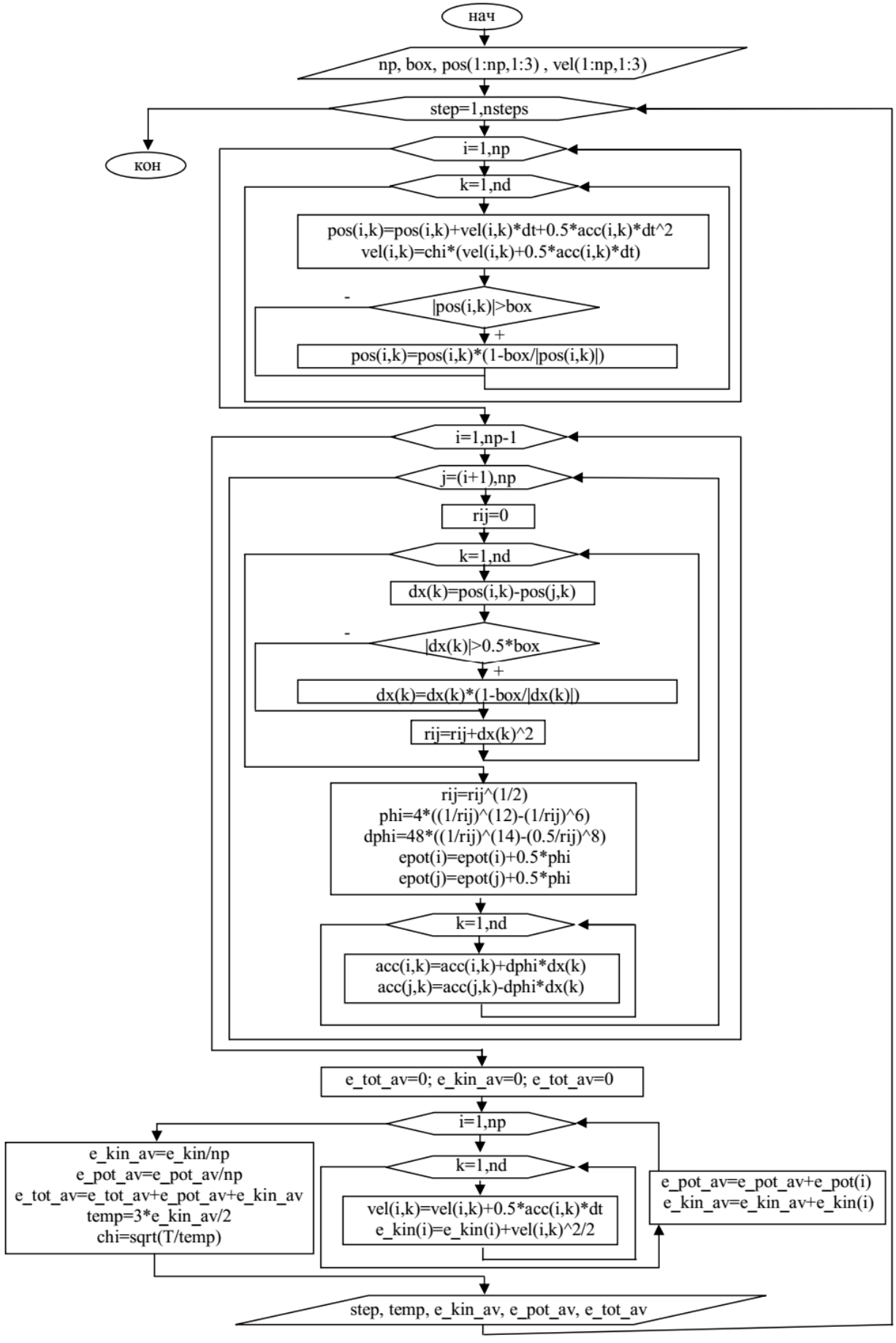
$$g(r) = \frac{V}{N(N-1)} \left\langle \sum_{j=1}^{N} \frac{n_j(r)}{4\pi r^2 \Delta r} \right\rangle.$$

Здесь V - объем моделируемой системы, величина $n_j(r)$ определяет число частиц в сферическом слое толщиной Δr на расстоянии r друг от друга. Проанализируйте графики при различных термодинамических состояниях системы.

5. Постройте среднеквадратичное смещение частиц со временем, $\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle$. Проанализируйте графики при различных термодинамических состояниях системы. С помощью формулы Эйнштейна оцените значение коэффициента самодиффузии

$$D = \lim_{t\to\infty} \frac{\left\langle \left|\Delta\vec{r}(t)\right|^2 \right\rangle}{6 \cdot t}.$$

Блок-схема алгоритм метода молекулярной динамики



ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
nx=10; ny=10; nz=10;
nd=3; np=nx*ny*nz;
alat=1.54; T=1.2;
box(1)=nx*alat; box(2)=ny*alat; box(3)=nz*alat;
density=np/(box(1)*box(2)*box(3));
fprintf(1,'************************************ \n');
fprintf(1,'Molecular Simulations of the LJ Liquid \n');
fprintf(1,'************************************ \n');

fprintf(1,'Number of atoms = %d \n',np);
fprintf(1,'Temperature of the system = %d \n',T);
fprintf(1,'Density of the system = %d \n',density);

pos(1:np,1:nd)=zeros(np,nd);
vel(1:np,1:nd)=zeros(np,nd);
acc(1:np,1:nd)=zeros(np,nd);

kol=0;
for i=1:1:nx
    for j=1:1:ny
        for k=1:1:nz
            kol=kol+1;
            pos(kol,1)=(i-1)*alat;
            vel(kol,1)=(rand-rand)*sqrt(3*T);
            pos(kol,2)=(j-1)*alat;
            vel(kol,2)=(rand-rand)*sqrt(3*T);
            pos(kol,3)=(k-1)*alat;
            vel(kol,3)=(rand-rand)*sqrt(3*T);
        end
    end
end
subplot(1,2,1);
scatter3(pos(:,1),pos(:,2),pos(:,3),100,'bo','filled')

nsteps=100; dt=0.003; temp=0; chi=1.0; rij=0;
e_pot_av=0; e_kin_av=0; e_tot_av=0; phi=0; dphi=0;
dx(1:nd)=zeros(nd,1);

% h = waitbar(0,'Please wait...');
for step=1:1:nsteps
    % computation here %
    % waitbar(step/nsteps,h)
    e_pot(1:np)=zeros(np,1);
    for i=1:1:np
        for k=1:1:nd
            pos(i,k)=pos(i,k)+vel(i,k)*dt+0.5*acc(i,k)*dt^2;
            vel(i,k)=chi*(vel(i,k)+0.5*acc(i,k)*dt);
            if pos(i,k)>box(k)
                pos(i,k)=pos(i,k)-box(k);
            end
            if pos(i,k)<0
```

```
                pos(i,k)=pos(i,k)+box(k);
            end
        end
    end
    % Evolve Forces and Energies
    for i=1:1:np-1
        for j=(i+1):1:np
            rij=0;
            for k=1:1:nd
                dx(k)=pos(i,k)-pos(j,k);
                if abs(dx(k))>0.5*box(k)
                    dx(k)=dx(k)*(1-box(k)/abs(dx(k)));
                end
                rij=rij+dx(k)^2;
            end
            rij=rij^(1/2);
            phi=4*((1/rij)^(12)-(1/rij)^6);
            dphi=48*((1/rij)^(14)-0.5*(1/rij)^8);
            e_pot(i)=e_pot(i)+0.5*phi;
            e_pot(j)=e_pot(j)+0.5*phi;
            for k=1:1:nd
                acc(i,k)=acc(i,k)+dphi*dx(k);
                acc(j,k)=acc(j,k)-dphi*dx(k);
            end
        end
    end
    % Velocity corrections
    e_pot_av=0; e_kin_av=0; e_tot_av=0;
    for i=1:1:np
        for k=1:1:nd
            vel(i,k)=vel(i,k)+0.5*acc(i,k)*dt;
            e_kin_av=e_kin_av+vel(i,k)^2;
        end
        e_pot_av=e_pot_av+e_pot(i);
    end
    e_kin_av=e_kin_av/np;
    e_pot_av=e_pot_av/np;
    e_tot_av=e_pot_av+e_kin_av;
    temp=3*e_kin_av;
    chi=sqrt(T/temp);
    fprintf(1,'Step= %d, Temp = %d, Ek = %d, Ep = %d, W= %d
              \n', step, temp, e_kin_av, e_pot_av, e_tot_av);
end
subplot(1,2,2);
scatter3(pos(:,1),pos(:,2),pos(:,3),100,'ro','filled')
xlim([0 box(1)])
ylim([0 box(2)])
zlim([0 box(3)])
```

§11. Задачи квантовой механики

1. Численно решите стационарное уравнение Шредингера. Рассмотрите случай одномерного движения в отсутствии внешнего потенциального поля и случай, когда частица находится в прямоугольной потенциальной яме

$$V(x)=\begin{cases}0, & \text{при} \quad |x|\le a\\ V_0, & \text{при} \quad |x|> a\end{cases}$$

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
phi=0; dphi=1; d2phi=0;
k2=1; x=0.0; dx=0.01; a=20;
while x<=a
    x=x+dx;
    d2phi=-k2*phi;
    dphi=dphi+d2phi*dx;
    phi=phi+dphi*dx;
    hold on; plot(x,phi,'k.')
end
```

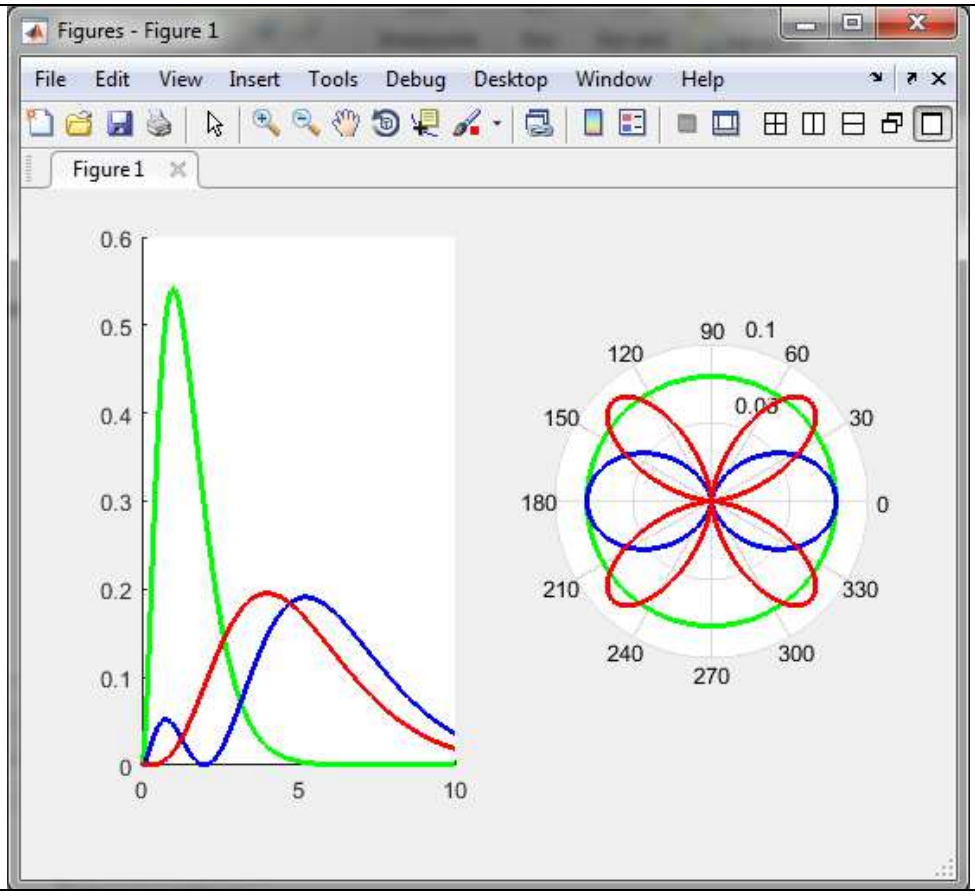
2. Постройте графики радиальной волновой функции R_{nl}, а также радиальные $P_{nl}(r)$ и угловые $P_{lm}(\theta)$ функции распределения для водородоподобного атома.

$R_{n,l}$	$Y_{l,m}$
$R_{1,0}=\sqrt{4Z^3}\exp(-Zr)$	$Y_{0,0}=\sqrt{\frac{1}{4\pi}}$
$R_{2,0}=\sqrt{\frac{Z^3}{8}}(2-Zr)\exp(-Zr/2)$	$Y_{1,0}=\sqrt{\frac{3}{4\pi}}\cos(\theta)$
$R_{2,1}=\sqrt{\frac{Z^3}{24}}Zr\exp(-Zr/2)$	$Y_{1,1}=\sqrt{\frac{3}{8\pi}}\cos(\theta)\sin(\theta)$
$R_{3,0}=\sqrt{\frac{4Z^3}{3^9}}\left[27-18Zr+2(Zr)^2\right]\exp(-Zr/3)$	$Y_{2,0}=\sqrt{\frac{5}{4\pi}}\left(\frac{3}{2}\cos^2(\theta)-\frac{1}{2}\right)$

ЛИСТИНГ ПРОГРАММЫ

```
close all; clear;
subplot(1,2,1);
r=0.02:0.01:10;
p10=4*r.^2.*exp(-2*r);
p20=(1/2)*r.^2.*exp(-r).*(1-0.5*r).^2;
p21=(1/24)*r.^4.*exp(-r);
hold on; plot(r,p10,'-g')
hold on; plot(r,p20,'-b')
hold on; plot(r,p21,'-r')

subplot(1,2,2);
theta=0:0.01:2*pi;
f00=1/(4*pi).*theta./theta;
f10=3/(4*pi).*cos(theta).^2;
f11=3/(8*pi).*sin(theta).^2.*cos(theta).^2;
polar(theta,f00,'-g');
hold on; polar(theta,f10./3,'-b');
hold on; polar(theta,f11.*3,'-r');
```

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ



3. Получите численные решения для нескольких первых собственных значений и собственных функций гармонического осциллятора с потенциальной энергией, $V(x)=\frac{1}{2}x^2$. Каким необходимо выбрать шаг *dx* для получения энергии основного состояния с точностью 0.1%?
4. Получите численное решение для ангармонического осциллятора с $V(x)=\frac{1}{2}x^2+bx^4$. Вычислите энергию основного состояния для *b=0.1*, *0.2* и 0.5. Как зависит энергия основного состояния от *b* при малых значениях *b*?
5. Получите численное решение для основного состояния линейного потенциала $V(x)=|x|$. Квантовомеханическое рассмотрение этого потенциала может быть использовано для изучения энергетического спектра связанной системы кварк-антикварк, называемой кварконием.
6. Для случая частицы, находящейся в прямоугольной потенциальной яме, рассчитайте возбужденные собственные состояния и собственные значения. Сколько всего имеется возбужденных состояний?
7. Численно решите одномерное нестационарное уравнение Шредингера. Рассмотрите случай отсутствия внешнего потенциального поля и случай, когда частица находится в прямоугольной потенциальной яме

$$V(x)=\begin{cases}0, & \text{при} \quad |x|\le a\\ V_0, & \text{при} \quad |x|>a\end{cases}.$$

Литература

1. Кунин, С. Вычислительная физика / С. Кунин. - М.: Мир, 1992. - 418 с.
2. Вычислительная физика. Практический курс: учебно-методическое пособие / Казан. гос. ун-т, Физ. фак. [авт.-сост.] Ю. Н. Прошин, И. М. Еремин. - Казань: Казанский государственный университет, 2009 .- 179 с.
3. Поршнев, С.В. Компьютерное моделирование физических процессов в пакете MATLAB: учебное пособие / С. В. Поршнев. - Издание 2-е, исправленное .- Санкт-Петербург: Лань, 2011. - 736 с.
4. Иродов, И. Е. Физика макросистем. Основные законы: учебное пособие / И. Е. Иродов. - 4-е изд. - М.: БИНОМ. Лаборатория знаний, 2009. - 207 с.
5. Основы статистической физики: Учебное пособие / А.Г. Браун, И.Г. Левитина. - 3-е изд. - М.: НИЦ ИНФРА-М, 2015. - 120 с.
6. Кузнецов, С. И. Молекулярная физика. Термодинамика: учебное пособие/ С. И. Кузнецов; Томский политехнический университет. - 2-е изд., перераб. и доп. - Томск: Изд-во ТПУ, 2007. - 126 с.
7. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. - Москва: Бином, 2001. - 636 с.
8. Ильина, В.А. Численные методы для физиков-теоретиков. т. 2 / В.А. Ильина, П.К. Силаев. - Москва-Ижевск: Институт компьютерных исследований, 2004. - 132 с.
9. Самарский, А.А. Методы решения сеточных уравнений / А.А. Самарский, Е.С. Николаев. - Москва: Наука, 1978. - 592 с.
10. Allen, M.P. Computer Simulation of Liquids / M.P. Allen and D.J. Tildesley. - Oxford: Clarendon Press, 1987. - 404 pp.
11. Гулд, Х. Компьютерное моделирование в физике / Х. Гулд, Я. Тобочник. - М.: Мир, 1990. - 350 с.
12. Метод молекулярной динамики в физической химии / Под ред. проф. Ю.К. Товбина. - М.: Наука, 1996. - 334 с.