

## ЛАБОРАТОРНАЯ РАБОТА № 3 (ПИ)

### Метод молекулярной динамики

**Цель работы:** моделирование динамики системы взаимодействующих молекул

**Задачи:** 1) Изучение метода молекулярной динамики 2) реализация алгоритма моделирования динамики молекулярной системы с использованием алгоритма Верле и потенциала Ленарда-Джонса; 3) отладка программы и проведение вычислительного эксперимента; 4) оформление отчета.

#### Краткое описание метода молекулярной динамики.

Метод молекулярной динамики применяется для изучения свойств жидкостей, газов и различных поверхностных явлений [1, 2]. Основа этого метода – численное решение классических уравнений Ньютона для рассматриваемой консервативной системы частиц. При этом предполагается выполнение законов сохранения импульса и энергии, причем потенциальная энергия определяется только суммой двухчастичных взаимодействий.

$$U = V(r_{12}) + V(r_{13}) + \dots + V(r_{23}) + \dots = \sum_{i < j-1}^N V(r_{ij}), \quad (1)$$

где  $V(r_{ij})$  – потенциал двухчастичных взаимодействий, а  $V(r_{ij})$  зависит только от абсолютной величины расстояния  $r_{ij}$  между частицами  $i$  и  $j$ . Парное взаимодействие вида (1) соответствует простым жидкостям, например жидкому аргону. Наиболее важными особенностями  $V(r)$  для простых жидкостей является сильное отталкивание для малых  $r$  и слабое притяжение на больших расстояниях. Одной из наиболее употребительных феноменологических формул для  $V(r)$  является

потенциал Леннарда-Джонса : 
$$V(r) = \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right], \quad (2)$$

где  $\sigma$  – нормированная единица расстояния,  $r$  – расстояние между частицами в единицах  $\sigma$ .

Силы и ускорения молекул определяются следующим образом:

$$F_{i,n+1} = -grad(U_{i,n}), \quad a_{i,n+1} = \frac{F_{i,n+1}}{m_i}$$

Для расчета траекторий каждой частицы используется алгоритм Верле в скоростной форме.

Расчет координаты: 
$$x_{n+1} = x_n + v_n \Delta t + a_n (\Delta t)^2. \quad (3)$$

Расчет скорости: 
$$v_{n+1} = v_n + \frac{1}{2}(a_n + a_{n+1})\Delta t. \quad (4)$$

Расчет ускорения: 
$$a_{n+1} = \frac{x_{n+1} - 2x_n + x_{n-1}}{(\Delta t)^2}. \quad (5)$$

где  $x_n, v_n, a_n$  – значение координаты, скорости и ускорения на  $n$ -ом шаге расчета,  $x_{n+1}, v_{n+1}, a_{n+1}$  – значение координаты, скорости и ускорения на  $n+1$ -ом шаге расчета,  $\Delta t$  – шаг по времени.

Далее приводятся примеры практической реализации этого метода с использованием алгоритма Верле для интегрирования уравнений движений молекул, причем взаимодействие между которыми описывается потенциалом Ленарда-Джонса [1, 2].

#### Структура программы:

- 1) Исходные данные: начальное состояния системы (размеры ящика  $L_x, L_y$ , количество молекул  $N$ , их координаты и скорость, а также временной интервал  $dt = \Delta t$ , используемый при интегрировании).
- 2) Вычисление потенциала и сил взаимодействия (функция `forc2`)
- 3) Вычисление ускорения, новых координат и скоростей, спустя интервал  $dt = \Delta t$

#### Пример 1. Исходные данные:

$N=2$ ;  $mas=[0.9 \ 1.1]$ ;

$Lx=0.5$ ;  $Ly=0.5$ ;  $vmax=0.35$ ;  $dt=0.???$ ;  $dt2=dt*dt$ ;  $dr=0$ ;  $nstep=55$ ;

**Пример 2.** Функция `sep2` для вычисления текущего расстояния между молекулами (MATLAB):

```
function r=sep2(xt,yt)
% m-m distance – расстояние между молекулами
dx=xt(2)-xt(1);
dy=yt(2)-yt(1);
dxy=dx.*dx+dy.*dy;
r=sqrt(dxy);
```

**Пример 3.** Реализация функции, описывающей потенциал Ленарда-Джонса и силу взаимодействия двух частиц:

```
function f=forc2(r)
% m-m force & Len-Jones potential
% used in mo22de.m // tested in tstforc2.m
ri=1./r;
ri3=ri.*ri.*ri;
ri6=ri3.*ri3;
g=24*ri.*ri6.*(2*ri6-1);
f(1,:)=g./r;
% 1/r canceled in Fx Fy by dx - dy
% f(2) - L-J potential
f(2,:)=4*ri6.*(ri6-1);
```

**Пример 4.** Реализация алгоритма Верле в среде MATLAB (фрагмент):

```
...
%===== B- Start force testing %f0=forc1(r0);
f0=forc2(r0);
ac(1)=acold(1)+f0(1,1)*dr./mas(1);
ac(2)=acold(2)-f0(1,1)*dr./mas(2);
%! minus from 3-rd Newton's Law
x=x0;
y=y0;
vt=v0;
%===== C- N particles dynamics: Verlet - k steps
for k=1:nstep
xt=x+vt.*dt+ac.*dt.*dt/2;
yt=y+vt.*dt+ac.*dt.*dt/2;
figure(2) % Screen ++
```

```
plot(xx,yy)
hold on
plot(x0,y0,'g+')
plot(xt(1),yt(1),'b*')
plot(xt(2),yt(2),'ro')
title(' molecules start+G and current*R positions')
hold off
% == D- Verlet //new: r, vt, f2=force2(r) and accelerat.
pause % strike ENTER to continue !
step=k
acold=ac
x=xt;
y=yt;
r=sep2(x,y)
dr=r-r0;
r0=r;
f=forc2(r);
ac(1)=acold(1)+f(1,1)*dr./mas(1);
ac(2)=acold(2)-f(1,1)*dr./mas(2);
v=vt
vt=v+(acold+ac).*dt/2;
end
```

#### Форма отчета

А) Листинг программы моделирования Б) Значение оптимального временного интервала  $dt$ , найденное экспериментально. В) Результаты моделирования изменения состояния системы молекул (координат и скоростей) при заданных начальных условиях и найденном интервале  $dt$ ; Г) зависимость  $r$  от  $k$ .

#### Литература.

1. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: В 2-х частях. Пер с англ. М.: Мир, 1990.
2. Allen M, Tildesley D. Computer simulation of liquids. 1991.
3. Rapaport D. C. The Art of Molecular Dynamics Simulation. Cambridge, 2004.
4. Ананасович В.В., Скакун В.В. MATLAB и SIMULINK 1997.