

# Лекция 8: Молекулярная динамика

## 1 Идея молекулярной динамики

Молекулярная динамика (МД) – метод вычисления равновесных и кинетических свойств системы, которые подчиняются классическим законам физики. Любой процесс с характеристическим временем  $\tau > \tau_q \sim 2\text{пс}$  может быть описан при помощи классической физики. Поэтому, за исключением вибраций ковалентных связей, все движения в молекулярной динамике можно описывать при помощи классических законов движения. Основным преимуществом молекулярной динамики является её способность получить “реальную” микроскопическую динамику, подчиняющуюся ландшафту свободной энергии системы и меж-атомным взаимодействиям. Программная реализация молекулярной динамики включает в себя следующие шаги:

1. Задание параметров, описывающих условия молекулярных симуляций, таких как температура, количество атомов и так далее.
2. Инициализация, которая включает в себя чтение координат атомов и генерацию начальных скоростей атомов.
3. Вычисление сил.
4. Численное интегрирование уравнений движения.
5. Повтор шагов 3 и 4 пока не будет достигнут желаемый временной интервал.
6. Вычисление средних значений.

## 2 Начальная конформация

После того, как параметры молекулярной динамики установлены, читается начальная структура. Важно, чтобы данная конформация не включала в себя наложения атомов друг на друга, необычные локальные конформации, что может повлечь за собой слишком большие значения сил и нестабильность численного интегрирования.

## 3 Начальное распределение скоростей

Начальное распределение скоростей может быть получено при помощи распределения Максвелла-Больцмана:

$$p(v_{i,\alpha}) = \left( \frac{m}{2\pi k_B T} \right)^{\frac{1}{2}} \exp \left[ -\frac{mv_{i,\alpha}^2}{2k_B T} \right], \quad (1)$$

где  $v_{i,\alpha}$  – компонента ( $\alpha = x, y, z$ ) вектора скорости для атома  $i$ . Это распределение может быть использовано для получения температуры системы при помощи теоремы о равнораспределении:

$$\left\langle \frac{mv_{i,\alpha}^2}{2} \right\rangle = \frac{1}{2} k_B T. \quad (2)$$

## 4 Вычисление атомарных сил

Рассмотрим взаимодействия Ван-дер-Ваальса, описанные при помощи потенциала Леннарда-Джонса:

$$V_{LJ} = 4 \left( \frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^6} \right). \quad (3)$$

В уравнении 3 мы использовали параметры потенциала  $\sigma_{ij} = 1$  и  $\varepsilon_h = 1$  для чистоты описания. Таким образом, сила, действующая на атом  $i$  вдоль оси  $x$ :

$$f_{i,x}^{LJ} = -\frac{\partial V_{LJ}}{\partial x_i} = -\frac{\partial V_{LJ}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial x_i} = 48 \frac{x_i - x_j}{r_{ij}^2} \frac{1}{r_{ij}^6} \left[ \frac{1}{r_{ij}^6} - \frac{1}{2} \right]. \quad (4)$$

Стоит заметить, что  $f_{j,x} = -f_{i,x}$ . Программная реализация данного алгоритма приведена на рисунке 1.

```

for(i = 0; i < N; i++){
    for(j = 0; j < i; j++){
        dr = r[x] - r[y];
        dr2 = dr*dr;
        dr2i = 1/dr2;
        r6i = r2i*r2i*r2i;
        ff = 48*r2i*r6i*(r6i-0.5);
        f[i] = f[i] + ff*dr;
        f[j] = f[j] - ff*dr
    }
}

```

Рис. 1: Программная реализация расчёта потенциала Леннарда-Джонса на языке C.

## 5 Интегрирование уравнений движения

Используя разложение Тейлора, можно записать:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{r'''(t)}{6}\Delta t^3 + o(\Delta t^4) \quad (5)$$

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{r'''(t)}{6}\Delta t^3 + o(\Delta t^4) \quad (6)$$

Сложив эти уравнения, получим:

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{f(t)}{m}\Delta t^2 + o(\Delta t^4). \quad (7)$$

Скорости, необходимые для расчёта кинетической энергии и температуры, могут быть найдены при помощи центральной разностной схемы:

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + o(\Delta t^2). \quad (8)$$

Уравнения 7 и 8 описывают один из самых популярных в молекулярной динамике алгоритмов интегрирования уравнений движения - алгоритм Верле. Стоит обратить внимание, что точность в уравнении для расчёта смещения координат (Ур. 7) больше, чем в уравнении для скоростей (Ур. 8). Программная реализация данного алгоритма приведена на рисунке 2.

```
sumv = 0;
sumv2 = 0;
for(i = 0; i < N; i++){
    rr = 2.0*r[i]-rm[i]+f[i]*delta*delta;
    vi = (rr-rm[i])/(2.0*delta);
    sumv = sumv + vi;
    sumv2 = sumv2 + vi*vi;
    rm[i] = r[i];
    r[i] = rr;
}
temp = sum2/N;
etot = en +sum2/2.0;
```

Рис. 2: Программная реализация интегратора Верле на языке C. Масса атомов и постоянная Больцмана взяты равными единице.