

# 计算物理初步

Ryelin

2023 年 12 月 18 日

本篇 note 将覆盖初等计算物理中的四个话题，分别是 ODE 的数值求解，PDE 的有限差分求解，Module Dynamic 分子动力学模拟，Monte Carlo 模拟。Note 中对历史背景以及一些细节的东西会减少阐述，而尽可能强调这四个话题的主线问题以及关键推导思路。

## 1 ODE 的数值求解

我们给出一个最经典的 ODE 形式，在单变量情形下，即为

$$\begin{cases} \frac{dy}{dx} = f(x, y) & a \leq x \leq b \\ y(a) = y_0 \end{cases} \quad (1)$$

如果我们采用积分形式，则可以得到一个关于  $y$  的自洽方程

$$y(x) = y_0 + \int_0^x f(x', y) dx' \quad (2)$$

关于 ODE 方程的全部求解，都来自如何从两种形式下手，给出在尽可能多的格点  $x_k$  上的函数值  $y(x_k)$

作为数值近似，我们首先将待求解的区间进行分割，记作  $a = x_0 < x_1 < \dots < x_{N-1} < x_N = b$ ，我们需要想办法利用上面的两个方程，得到这个格点上的近似取值。在最开始的阶段，我们先认定划分的格点之间间隔均为  $h$

### 1.1 初值问题

#### 1.1.1 Euler 方法

我们使用两种方法来导出三种 Euler 方法。从 (1) 中，我们可以用差分来代替微分。如果我们认定

$$y'(x_k) \doteq \frac{y_{k+1} - y_k}{h}$$

我们可以导出向前 Euler 公式

$$y_{k+1} = y_k + hf(x_k, y_k) \quad (3)$$

如果我们认定

$$y'(x_k) \doteq \frac{y_k - y_{k-1}}{h}$$

我们可以导出向后 Euler 公式

$$y_{k+1} = y_k + hf(x_{k+1}, y_{k+1}) \quad (4)$$

将 (3) 和 (4) 两式相加，就可以得到改进 Euler 公式

$$y_{k+1} = y_k + \frac{h}{2}(f(x_k, y_k) + f(x_{k+1}, y_{k+1})) \quad (5)$$

上面的公式，我们也可以从积分形式 (2) 的数值积分中取得。其中，自洽项中，根据积分中值定理，将有

$$\int_{x_{k-1}}^{x_k} f(x, y) dx \doteq hf(\xi, y(\xi))$$

若令  $\xi = x_k$ ，则得到向前 Euler 公式 (3)，若令  $\xi = x_{k+1}$ ，则得到向后 Euler 公式 (4)，若采用梯形公式，即令  $\xi = \frac{1}{2}(x_k + x_{k+1})$ ，则得到改进 Euler 公式 (5)

向前 Euler 公式是显式公式，可以直接求解。向后 Euler 和中心差商公式都是隐式，因此需要往往用迭代求解才能得到较好的结果。我们后面会介绍针对隐式公式的预估校正方法。

**收敛性和稳定性** 作为数值求解，我们可以考虑各步近似所带来的截断误差。定义第  $k$  步的**总体截断误差**为

$$e_k = y(x_k) - y_k$$

这里只考察显式的向前 Euler 公式，由  $y_{k+1} = y_k + hf(x_k, y(x_k)) + O(h^2)$ ，可以定义局部截断误差为

$$T_{k+1} = y(x_{k+1}) - (y(x_k) + hf(x_k, y(x_k))) = O(h^2)$$

下面考虑总体截断误差和局部截断误差之间的关系，我们有

$$\begin{aligned} |e_{k+1}| &= |y(x_{k+1}) - y_{k+1}| = |y(x_k) + hf(x_k, y(x_k)) + T_{k+1} - y_k - hf(x_k, y_k)| \\ &\leq |y(x_k) - y_k| + h|f(x_k, y(x_k)) - f(x_k, y_k)| + |T_{k+1}| \leq |T_{k+1}| + (1 + hL)|e_k| \\ &\leq \sum_{j=0}^{k-1} (1 + hL)^j |T_{k-j}| \end{aligned}$$

上式中引用了光滑函数的 Lipschitz 条件。

因此，局部截断误差是

$$|e_{k+1}| \leq \sum_{j=0}^{k-1} (1 + hL)^j |T_{k-j}| = O(h^2) \sum_{j=0}^{k-1} (1 + hL)^j = O(h)$$

接下来，我们要考察数值结果的稳定性。这要求在各步数值计算中得到结果的误差  $\rho_k$  不会随着传播而增大。一般来讲误差方程满足数值迭代的相同方程，方便起见，我们讨论这样的一个微分方程

$$y' = \lambda y$$

则误差方程为

$$\rho_{k+1} = \rho_k + \lambda h \rho_k$$

如果要求数值计算的误差不增大，则必须有

$$\left| \frac{\rho_{k+1}}{\rho_k} \right| = 1 + \lambda h \leq 1$$

这给出了保证数值求解稳定对于迭代步长的限制。

如果是使用后退 Euler 方法，那么这个微分方程的稳定条件是

$$\left| \frac{\rho_{k+1}}{\rho_k} \right| = \frac{1}{|1 - \lambda h|} \leq 1$$

从而只需要  $|1 - \lambda h| \geq 1$ ，这个稳定区域包含了整个复平面的左半平面，我们称这个迭代方法是 A 稳定的。

### 1.1.2 Runge-Kutta 方法

我们重新考察 Euler 方法, 可以发现我们所做的, 无外乎是用了在  $[x_k, x_{k+1}]$  中一个点的  $f(x, y)$  的结果作为步进斜率。很自然的想法是, 我们完全可以试图使用在这个区间内的多个斜率预报  $K_1, \dots, K_m$ , 并将其加权平均为  $K = \lambda_1 K_1 + \lambda_2 K_2 + \dots + \lambda_m K_m$ , 得到一个步进斜率的预报值, 使得  $y_{k+1} = y_k + hK$ 。称这样的数值迭代求解 ODE 的方法为  $m$  阶 Runge-Kutta 方法。这里的加和权重将会和斜率预报的选取方式有关, 我们以一种三阶 R-K 公式为例

我们取第一个斜率预报为  $K_1 = f(x_k, y_k)$ , 第二个斜率预报为  $K_2 = f(x_k + ph, y_k + phK_1)$ , 第三个斜率预报为  $K_3 = f(x_k + qh, y_k + qhK_2)$  (这里我一直认为应该是  $K_3 = f(x_k + qh, y_k + qhK_1)$  或者  $K_3 = f(x_k + qh + ph, y_k + qhK_1 + phK_2)$ , 才具有更加正确的几何意义, 但是这个问题老师一直没正面回答我, 也似乎没 get 到我的意思, 为了正确性起见仍然以老师的思路编写)。于是应当有

$$\begin{cases} K_1 = f(x_k, y_k) \\ K_2 = f(x_k + ph, y_k + phK_1) \\ K_3 = f(x_k + qh, y_k + qhK_2) \\ y_{k+1} = y_k + h(\lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3) \end{cases}$$

为了得到权重因子, 我们将  $K_1, K_2, K_3$  在  $(x_k, y_k)$  处展开, 可以得到

$$\begin{aligned} K_1 &= f(x_k, y_k) = y'(x_k) \\ K_2 &= y'(x_k + ph) = y'(x_k) + phy''(x_k) + \frac{1}{2}(ph)^2 y'''(x_k) \\ K_3 &= y'(x_k + qh) = y'(x_k) + qhy''(x_k) + \frac{1}{2}(qh)^2 y'''(x_k) \end{aligned}$$

将这个结果代入到  $y_{k+1}$  的迭代表达式中, 合并同阶项将有

$$y(x_{k+1}) = y(x_k) + h(\lambda_1 + \lambda_2 + \lambda_3)y'(x_k) + h(p\lambda_2 + q\lambda_3)y''(x_k) + \frac{1}{2}(p^2\lambda_2 + q^2\lambda_3^2)y'''(x_k) \quad (6)$$

这个结果和  $y(x_{k+1}) = y(x_k + h)$  的标准结果对比, 最终得到非平庸的几个需要满足的等式为

$$\begin{cases} p\lambda_2 + q\lambda_3 = \frac{1}{2} \\ p^2\lambda_2^2 + q^2\lambda_3^2 = \frac{1}{3} \end{cases} \quad (7)$$

因此, 取点  $p, q$  和权重  $\lambda_{1,2,3}$  只要满足上面的式子, 即可构造出若干三阶 R-K 公式。典型的取法是取  $p = 1/2, q = 1$ , 可以得到  $y_{k+1} = y_k + \frac{1}{6}(K_1 + 4K_2 + K_3)$

### 1.1.3 线性多步法

前面几个方法共同的特性是, 如果我们要计算  $y_{k+1}$ , 我们只用到了  $y_k$  一个点, 而在这样的一个区间中找了很多的中间格点来辅助计算。这启发我们完全可以使用多个数据点来导出下一个数据点的取值。

我们考虑使用  $y_{k-3}, y_{k-2}, y_{k-1}, y_k$  四个数据点导出  $y_{k+1}$ 。在积分形式的迭代 (2) 中, 一个比较好的近似是认为  $[x_k, x_{k+1}]$  区间中,  $y(x)$  的行为接近由前面四个点插值出的三次 Lagrange 插值多项式  $p_k(x) = L_{k-3}y_{k-3} + L_{k-2}y_{k-2} + L_{k-1}y_{k-1} + L_k y_k$ ,  $L_i = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$ , 因此可以采用的迭代为

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} p_k(x) dx$$

当然, 我们也可以用  $y_{k-2}, y_{k-1}, y_k, y_{k+1}$  进行内插, 得到另一种线性多步法。两种迭代公式分别称为 Adams 外插和 Adams 内插

### 1.1.4 预估校正法

前面许多公式都用到了隐式形式，并且我们发现隐式形式普遍具有更高的稳定性。为了规避隐式形式需要的迭代求解，我们考虑使用预估校正方法。核心思路是，之所以需要进行迭代是因为我们需要方程的自洽项中有一个输入，迭代的思路是将这个思路选择为上一步迭代的计算结果。但事实上我们也可以采用一些显式形式得到一个输入。例如，如果我们希望引用改进 Euler 法，在每步迭代中我们可以先用向前 Euler 公式得到一个函数值作为预估，将这个预估值代入到改进 Euler 公式的自洽项中。又比如，如果我们希望引用四阶 Adams 内插法，我们可以先用外插法得到一个预估值，将这个预估值代入到内插法的自洽项中得到校正数值。

## 1.2 步长的选取

前面我们所提到的计算方法都是采取定步长的方案，事实上，如果  $f(x, y)$  的奇异性不均匀，那么定步长不再是一个好的策略。因此，可以在每一步迭代时，比较此时的步长和削减一半步长后的计算数值的差异，在小于一个设定好的允许容差值，采取当前的步长进行此步迭代。

我们考虑改变步长以后的截断误差。设步长为  $h$  和  $h/2$  时的截断误差分别为  $T_{k+1}^{(h)}, T_{k+1}^{(h/2)}$ ，则有

$$\frac{T_{k+1}^{(h)}}{T_{k+1}^{(h/2)}} = \frac{O(h^{r+1})}{2O((h/2)^{r+1})} = \frac{1}{2^r}$$

## 1.3 边值问题

当方程在二阶以上时，边界条件可以以边值的形式给出，一般的形式为

$$\begin{cases} y'' = f(x, y, y') \\ a_0 y(a) - a_1 y'(a) = c_0 \\ b_0 y(b) - b_1 y'(b) = c_1 \end{cases}$$

一个最直接的想法是我们引入一个新的初值条件  $d_0 y(a) - d_1 y'(a) = s$ ，然后搜索适当的  $s$  使得此时初值问题的解能够满足原始的第二边值条件。适当选取  $d_0, d_1$  使得  $\begin{vmatrix} a_0 & -a_1 \\ d_0 & -d_1 \end{vmatrix} = 0$ ，则可以得到一个含有待定参数  $s$  的初值问题

$$\begin{cases} y'_0 = y_1 \\ y'_1 = f(x, y_0, y_1) \\ y_0(a) = a_1 s - d_1 c_0 \\ y_1(a) = a_0 s - d_0 c_0 \end{cases} \quad (8)$$

如果选取的  $s$  足够好，则应当有  $\phi(s) = b_0 y_0(b; s) - b_1 y_1(b; s) - c_1 = 0$ 。于是我们可以构造关于  $s$  的迭代系列为

$$s_{n+1} = s_n - \frac{\phi(s_n)}{\phi'(s_n)}$$

全部的问题变成了我们如何取得  $\phi'(s_n)$ 。

我们对有关  $s$  的初值问题 (8) 两边对  $s$  微分, 得到

$$\begin{cases} \frac{\partial y'_0}{\partial s} = \frac{\partial y_1}{\partial s} \\ \frac{\partial y'_1}{\partial s} = \frac{\partial f}{\partial y_0} \frac{\partial y_0}{\partial s} + \frac{\partial f}{\partial y_1} \frac{\partial y_1}{\partial s} \\ \left. \frac{\partial y_0}{\partial s} \right|_{y=a} = a_1 \\ \left. \frac{\partial y_1}{\partial s} \right|_{y=a} = a_0 \end{cases} \quad (9)$$

因此补充定义  $y_{s0} = \frac{\partial y_0}{\partial s}, y_{s1} = \frac{\partial y_1}{\partial s}$ , 得到补充了  $s$  的迭代方程为

$$\begin{cases} y_1 = y'_0 \\ y'_1 = f(x, y_0, y_1) \\ y'_{s0} = y_{s1} \\ y_{s1} = \frac{\partial f}{\partial y_0} y_{s0} + \frac{\partial f}{\partial y_1} y_{s1} \end{cases} \quad (10)$$

以及初值条件

$$y_0(a) = a_1 s - d_0 \quad y_1(a) = a_0 s - d_0 c_0 \quad y_{s0} = a_1 \quad y_{s1} = a_0$$

于是, 我们所期待的  $\phi'(s)$  为

$$\phi'(s) = \frac{\partial}{\partial s} (b_0 y_{s0}(b) - b_1 y_{s1}(b)) \quad (11)$$

从而可以实现迭代求解。迭代终点的  $s$  对应的初值问题的解, 即为边值问题的解。

这个方法被称为**打靶法**

对于边值问题的求解, 使用有限差分法将更加自然, 我们将下一章中详细介绍这种方法。

## 2 PDE 的有限差分求解

我们知道, 所谓的微分, 实际上是一种差分的极限。因此在数值上, 我们可以用差分来代替微分, 将一个微分方程的问题变成一个有限维度的线性代数问题。这一思路将贯穿我们一整章的内容。

### 2.1 对流方程的有限差分

#### 2.1.1 基本差分

在正式讨论之前, 我们首先将一些常见的基本差分格式列在这里。从这些差分格式上, 将间隔取极限, 就可以得到严格的微分定义。

$$\text{向前差分} \quad \left. \frac{\partial y}{\partial x} \right|_{x_k} = \frac{y_{k+1} - y_k}{h} \quad (12)$$

$$\text{向后差分} \quad \left. \frac{\partial y}{\partial x} \right|_{x_k} = \frac{y_k - y_{k-1}}{h} \quad (13)$$

$$\text{中心差分} \quad \left. \frac{\partial y}{\partial x} \right|_{x_k} = \frac{y_{k+1} - y_{k-1}}{2h} \quad (14)$$

**对流方程**是一种典型的双曲方程，其具体形式为  $\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, u|_{t=0} = \phi(x)$ 。在时空上，通过引用不同的差分方法，可以建立不同的差分格式。我们用  $k$  表示空间指标， $j$  表示时间指标，时空间隔分别取为  $\tau, h$ 。在这里我们列出几种常用的

- 逆风差分格式：时空均采用向前差分，则有  $\frac{u_{k,j+1} - u_{k,j}}{\tau} + a \frac{u_{k+1,j} - u_{k,j}}{h} = 0$ ，最后化简可得

$$u_{k,j+1} = (1 - ar)u_{k,j} + ar u_{k+1,j}, r = \frac{\tau}{h} \quad u_{k,0} = \phi_k \quad (15)$$

- FTCP 差分格式：时间向前差分，空间中心差分，最后化简可得

$$u_{k,j+1} = u_{k,j} + \frac{ar}{2}u_{k+1,j} - \frac{ar}{2}u_{k-1,j} \quad u_{k,0} = \phi_k \quad (16)$$

- Frog-Leap 差分格式：时空中心差分，最后化简得到

$$u_{k,j+1} = u_{k,j-1} + ar u_{k+1,j} - ar u_{k-1,j}$$

### 2.1.2 利用特征线的差分格式

对于对流方程来说， $x - at = 0$  是其特征线，在这条线上， $u(x, t)$  的取值总是保证恒定。因此，过  $u_{k,j+1}$  的特征线交  $t = t_j$  于某点 A，则 A 点的函数值一定即为  $u_{k,j+1}$ 。而 A 点的位置，可以通过差值的方式来预报。

- Lax 差分格式：用  $(x_{k-1}, u_{k-1,j}), (x_{k+1}, u_{k+1,j})$  两点线性插值，然后代入 A 点的坐标得到 A 点的函数值。最终有

$$u_{k+1,j} = \frac{1}{2}(u_{k+1,j} + u_{k-1,j}) + \frac{ar}{2}(u_{k+1,j} - u_{k-1,j})$$

- Lax-Wendroff 差分格式：用  $(x_{k-1}, u_{k-1,j}), (x_k, u_{k,j}), (x_{k+1}, u_{k+1,j})$  三点做二次插值进行预报。

### 2.1.3 相容性、收敛性和稳定性

一个差分格式和对应的微分方程相容，是指差分格式在步长无穷小时能够回到微分方程形式。但是一般相容并不意味着这样的差分格式就是收敛的，我们首先讨论差分格式的收敛性

我们为了得到  $u_{k,j}$ ，总是会需要在  $t = t_{j-1}$  的若干个格点，这若干个格点又会需要在前一个时间切片上的若干个格点。最后总的来讲， $u_{k,j}$  总是依赖于不同时间层的若干格点，这些格点成为  $u_{k,j}$  的依赖区域。我们不加证明地指出，差分格式收敛的必要条件是，**差分格式的依赖区域必须包含微分方程的依赖区域**，这个条件被称为 Courant 条件。对于对流方程，这意味着特征线在时间层上的点必须在差分格式的依赖区域以内。

例如，如果  $a > 0$ ，那么方程的特征线总是在考察点的左侧，此时逆风差分格式就永远无法收敛，因为逆风差分格式的依赖区域在考察点的右侧。

接下来，我们讨论差分格式的稳定性，一种常用的方法是 Fourier 方法。它的思路是认定由差分格式给出的误差方程的解形式为

$$\varepsilon_{k,j} = G^j e^{inx_k}$$

那么差分格式稳定，意味着误差不会随着传播放大，此时就等价于要求  $|G| < 1$ 。

例如，对于逆风差分格式，误差方程为

$$\varepsilon_{k,j+1} = \varepsilon_{k,j} - ar(\varepsilon_{k+1,j} - \varepsilon_{k,j})$$

代入一般形式以后，消去一个  $G^j e^{inx_k}$  因子，可以得到

$$G = 1 + ar - are^{inh}$$

于是

$$|G|^2 = 1 + 4ar(1 + ar) \sin^2 \frac{nh}{2}$$

因此逆风差分格式的稳定性条件是  $a < 0, -1 < ar < 0$

对于一个适定的初值问题，一个与它相容的差分格式收敛的充要条件，即是这个格式的稳定性。这被称为 **Lax 等价定理**

## 2.2 抛物型方程的差分格式

典型的抛物型方程是热传导方程

$$\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2}$$

在空间上，我们需要引入二阶差分

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{k+1,j} - 2u_{k,j} + u_{k-1,j}}{h^2}$$

而在时间上，通过引用向前、向后和中心差分，可以得到显式、隐式和 Richardson 三种基本差分格式。

除此之外，我们可以引入一个时间的中间层来导出差分格式。我们考察  $t = t_{j+1/2}$ ，则

$$\left. \frac{\partial u}{\partial t} \right|_{k,j+1/2} = \frac{u_{k,j+1} - u_{k,j}}{\tau}$$

而我们用相邻两时间层上二阶空间偏导的平均值来表示这个中间层的空间项，则有

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{k,j+1/2} = \frac{1}{2} \left( \left. \frac{\partial^2 u}{\partial x^2} \right|_{k,j} + \left. \frac{\partial^2 u}{\partial x^2} \right|_{k,j+1} \right)$$

最后，在中间层  $j + 1/2$  上代入到原始微分方程中，可以得到

$$u_{k,j+1} = u_{k,j} + \frac{c\tau}{2h^2} (u_{k,j+1} + u_{k,j-1} - 2u_{k,j}) + \frac{c\tau}{2h^2} (u_{k+1,j+1} + u_{k-1,j+1} - 2u_{k,j-1})$$

这个方法被称为 **Crank-Nicholson 方法**

**边界条件的差分** 由于涉及到了二阶项，因此此时我们需要讨论一下边界条件如何导出差分形式。如果是第一类边界条件，那么直接在差分格点上取到边界条件的数值即可。但如果是第二类边界条件，就需要用差分来代替边界条件中的微分信息。

一种方法是在左端点使用向前差分，在右端点使用向后差分。另一种方法是假设所有网格点平移半个步长，则由  $u_{-1,0}, u_{0,0}$  给出在边界上的导数值。于是从边界条件出发即可得到迭代初始时的  $u_{0,j}$  取值。

**高维热传导方程的交替方向法** 如果是一个二维热传导方程，为了减少计算量，我们引入交替方向法。基本思想是，在时间层  $2j + 1$  上列出差分格式时，y 方向直接引用时间层  $2j$  上的结果，相当于在此时间层上是对  $(x, t)$  这个一维方程进行更新。而在时间层  $2j + 2$  上，列出的差分格式时，x 方向直接引用  $2j + 1$  的结果。这样的运算交替进行，每次时间层的更新都仅仅是一个一维的热传导迭代。

当然，我们也可以引入一个中间层  $j + 1/2$  来替代前面  $2j + 1$  的中间步骤。

## 2.3 椭圆型方程的差分格式

椭圆型方程成为一个稳定问题，一般形式为

$$\nabla^2 u = f(\mathbf{r})$$

此时不再有初始条件，而只有边界条件。从边界条件出发所导出的结果则是一个稳定解。用差分代替微分的过程和前面所提的过程基本一致，唯一有所不同的是此时的边界条件变成了二维，边界条件变得复杂，因此对边界条件的差分和之前有所不同。

在 Poisson 方程中，我们需要区分内点和正则内点的不同。正则内点的周围四个格点近邻都仍然在求解区域以内，因此我们可以直接用一般的差分格式进行近似。然而非正则内点总是有与之紧邻的格点在边界以外，因此非正则内点的取值需要依赖于边界条件。

我们有两种方法确定非正则内点的取值，一种是直接转移法，即直接取与非正则内点距离最近的边界上的点作为取值。另一种方法是用边界上某点，及其和非正则内点连线上的一个近邻正则内点，这两点的线性插值来确定非正则内点的取值。

## 3 分子动力学

一言以蔽之，分子动力学是在试图以 Newton 力学作为第一性原理直接模拟分子的运动。这是一种确定性的对多体问题的模拟方法。

### 3.1 原胞与边界条件设置

在材料模拟中，一般人们都更感兴趣在热力学极限下的行为，但是由于算力的限制，人们显然不可能真的对无限大的体系进行模拟。因此我们必须选用一个较小的原胞进行研究。而为了用有限尺寸的原胞尽可能接近实际的效果，则会引入一系列的边界条件。

在这里，只阐述周期性边界条件。我们假设考察的原胞镶嵌在一块无穷大的物质中，而每当粒子以某个速度矢量穿过边界时，都将以相同的速度矢量从此边界的对面重新穿入。用这种方法，我们模拟了一个无穷大的系统。

### 3.2 运动方程与迭代算法

接下来，我们考察每一个粒子的运动。首先我们只考察一个单粒子的运动，则相当于一个粒子在一个势场  $V(\mathbf{r})$  中，在 Newton 方程的规律下不断更新自己的位置、速度和加速度。Newton 方程的结果是

$$m^{(i)} \ddot{\mathbf{r}}^{(i)} = -\frac{\partial U}{\partial \mathbf{r}^{(i)}}$$

从这个方程出发，我们将给出一些典型的算法，逐步计算在各个时间层下粒子的位置、速度和加速度。为了方便起见，接下来的一部分篇幅，我会忽略标记粒子的上指标。

#### 3.2.1 Verlet 算法

我们记时间间隔为  $\tau$ ，并用下指标  $i$  来标记各个时间层。在  $t = t_i$  时刻，位矢的向前和向后精确到二阶的 Taylor 展开为

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \tau \cdot \mathbf{v}_i + \frac{1}{2} \tau^2 \cdot \mathbf{a}_i^2 \quad (17)$$

$$\mathbf{r}_{i-1} = \mathbf{r}_i - \tau \cdot \mathbf{v}_i + \frac{1}{2} \tau^2 \cdot \mathbf{a}_i^2 \quad (18)$$



将 (17) 和 (18) 相加，即可得到

$$\mathbf{r}_{i+1} = -\mathbf{r}_{i-1} + 2\mathbf{r}_i + \tau^2 \cdot \mathbf{a}_i^2 \quad (19)$$

将 (17) 和 (18) 相减，可以得到

$$\mathbf{v}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2\tau} \quad (20)$$

因此我们得到了 Verlet 算法的更新策略

$$\begin{cases} \mathbf{r}_{i+1} = -\mathbf{r}_{i-1} + 2\mathbf{r}_i + \tau^2 \cdot \mathbf{a}_i^2 \\ \mathbf{v}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2\tau} \\ \mathbf{a}_i = -\frac{1}{m} \frac{\partial U}{\partial \mathbf{r}} \Big|_{i+1} \end{cases}$$

### 3.2.2 速度 Verlet 算法

和 Verlet 算法不同，速度 Verlet 算法期待先得到下一个时间层的加速度值，然后用改进 Euler 方法得到下一个时间层的速度，即

$$\begin{aligned} \mathbf{r}_{k+1} &= \mathbf{r}_i + \tau \cdot \mathbf{v}_i + \frac{1}{2} \mathbf{a}_i \cdot \tau^2 \\ \mathbf{a}_{i+1} &= -\frac{1}{m} \frac{\partial U}{\partial \mathbf{r}} \Big|_{i+1} \\ \mathbf{v}_{i+1} &= \mathbf{v}_i + \frac{1}{2} (\mathbf{a}_i + \mathbf{a}_{i+1}) \end{aligned}$$

在实际计算中，可以增加一个中间层

$$\begin{aligned} \mathbf{v}_{i+1/2} &= \mathbf{v}_i + \frac{1}{2} \mathbf{a}_i \tau \\ \mathbf{r}_{i+1} &= \mathbf{r}_i + \mathbf{v}_{i+1/2} \cdot \tau \\ \mathbf{a}_{i+1} &= -\frac{1}{m} \frac{\partial U}{\partial \mathbf{r}} \Big|_{i+1} \\ \mathbf{v}_{i+1} &= \mathbf{v}_{i+1/2} + \frac{1}{2} \mathbf{a}_{i+1} \cdot \tau \end{aligned}$$

### 3.2.3 Geer 预估校正法

我们换用一种思路。考虑位矢、速度、加速度和急动度的 Taylor 展开，则有

$$\begin{pmatrix} \mathbf{r}^p \\ \mathbf{v}^p \\ \mathbf{a}^p \\ \mathbf{b}^p \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{r}_i \\ \frac{d\mathbf{r}}{dt} \Big|_i \\ \frac{d^2\mathbf{r}}{dt^2} \Big|_i \\ \frac{d^3\mathbf{r}}{dt^3} \Big|_i \end{pmatrix}$$

通过预测的位置可以求出一个加速度，这个加速度和预测的加速度相比会有一个误差，定义此误差为

$$\delta \mathbf{a}_{i+1} = -\frac{1}{m} \frac{\partial U}{\partial \mathbf{r}} \Big|_p - \mathbf{a}^p$$

则最终将有

$$\mathbf{r}_{i+1} = \mathbf{r}_p + c_0 \delta \mathbf{a}_{i+1}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_p + c_1 \delta \mathbf{a}_{i+1}$$

$$\mathbf{a}_{i+1} = \mathbf{a}_p + c_2 \delta \mathbf{a}_{i+1}$$

$$\mathbf{b}_{i+1} = \mathbf{b}_p + c_3 \delta \mathbf{a}_{i+1}$$

这里  $c_i$  是 Geer 预测校正因子，可以查表。

### 3.3 势函数

在前面的讨论中，我们假设我们已经知道了势函数的形式。而在分子动力学模拟中，势函数的选取有多种方式。最经典的势函数是 Lennard-Jones 势函数，被写为

$$U(r_{ij}) = 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$

这里第一项代表短程排斥，第二项代表长程吸引。在  $r_{ij} = \sigma_{ij}$  时势能为零，势能最低点出现在  $2^{1/6}r_{ij}$ ，最小值为  $-\varepsilon_{ij}$ 。

除此之外，Morse 曾提出指数函数形式的势函数来处理双原子分子的分解，形式为

$$U(r_{ij}) = \varepsilon \sum_{i < j} [e^{-2a(r_{ij}/r_0-1)} - 2e^{r_{ij}/r_0-1}]$$

除此之外，为了考虑到多体相互作用以及描述原子间成键情况，引入半经验的嵌入原子势 EAM。中心想法是将原子周围的环境视为一种均匀正电荷背景下的均匀电子气。于是体系的总能量可以表示为嵌入能和二体排斥能之和，且体系的电荷分布近似为各个原子电荷的简单线性叠加。于是

$$U_{EAM} = \sum_i F_i(\rho_i) + \frac{1}{2} \sum_{i,j=1}^N \phi(r_{ij})$$

为了进一步减小计算量，考虑到在当粒子间距离很远时所有势函数形式都迅速衰减，因此我们只需要对部分处在设定好的有限力程以内的原子计算势函数就足够了。因此，我们可以设置截断距离为  $r_c$ ，势函数计算中的加和仅仅对  $r_{ij} < r_c$  的部分进行。同时由于势函数强行阶段，会导致原子间作用力在截断距离处不连续。为了解决这一问题，势函数可以变为如下形式

$$U^S(r_{ij}) = \begin{cases} U(r_{ij}) - U_c - \left( \frac{dU}{dr_{ij}} \right)_{r_{ij}=r_c} (r_{ij} - r_c) & r_{ij} \leq r_c \\ 0 & r_{ij} > r_c \end{cases}$$

如果这样，仍然无法摆脱每一次迭代都要对所有原子对计算距离的冗余计算量。为此，如果我们认为每一步模拟步长中，粒子的位置变化都足够小，即在一定的时间内，每个粒子的近邻粒子基本保持不变，我们就可以引入一个更新频率不需要很高的 Verlet 近邻表。以每个粒子为中心，构造一个半径为  $r_v > r_c$  的球，在球范围内的粒子被认为是该粒子的近邻，判断两粒子间距是否在截断距离以内只需要对在 Verlet 近邻表以内的粒子进行。Verlet 近邻表不用每步都更新，可以在经历了相当次数的迭代后再进行更新。

### 3.4 系综调节

一般的分子动力学模拟中, 无法保证系统某些外部条件, 这时需要对一些系统的物理参数进行调节。一般需要调节的主要是系统的温度和压强。

三种半经验地调节系统温度的方法分别是速度标定法, Anderson 热浴法和 Langevin 热池法。设我们需要系统维持在温度  $T_0$ , 在模拟的每一步, 我们都计算系统此时的温度  $T = \frac{2}{3(N - N_c)k} \sum_{i=1}^N \frac{1}{2} m_i v_i^2$ , 则在每一步, 我们都给所有粒子的速度乘上一个标度因子

$$\beta = \left[ \frac{3(N - N_c)kT_0}{\sum_i m_i v_i^2} \right]^{1/2}$$

这是速度标定法。

Anderson 热浴法是假想系统与一个具有目标温度的热池接触, 系统中的粒子与热池随机碰撞, 两次碰撞的时间满足 Poisson 分布。每次碰撞的效果, 都使得随机若干个粒子的速度重置, 重置的数值满足 Maxwell-Boltzmann 分布。

Langevin 热池法虚构了热池粒子, 在运动方程中引入了阻尼项和随机碰撞项。于是运动方程变为

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} \quad \dot{\mathbf{p}}_i = \mathbf{F}_i - \gamma \mathbf{p}_i + \sigma \xi_i$$

这里  $\xi_i \sim N(0, 2k_B T \gamma m_i)$

Nose—Hoover 则直接引入了一个额外的自由度  $s$  来表述恒温热库的自由度, 这个热库将带来  $\frac{Q}{2} \dot{s}^2$  的额外动能和  $Gk_B T \ln s$  的额外势能。于是系统的 Lagrangian 被改为

$$L_{Nose} = \sum_{i=1}^N \frac{m_i}{2} \dot{\mathbf{r}}_i^2 + \frac{Q}{2} \dot{s}^2 - U(\mathbf{r}_1 \cdots \mathbf{r}_N) - Gk_B T \ln s$$

并据此写出新的运动方程。

类似地, 对于压强的调节, 也是通过引入一个有效质量为  $M$ , 体积  $\Omega$  的有效活塞实现的, 引入以后系统的 Lagrangian 变为

$$L_{Anderson} = \sum_{i=1}^N \frac{m_i}{2} (\Omega^{1/3} \dot{\mathbf{x}}_i)^2 - \sum_{i>j} U(\Omega^{1/3} x_{ij}) + \frac{M}{2} \dot{\Omega}^2 - P_0 \Omega$$

### 3.5 关联函数

我们所关心的系统的物理性质, 大多可以通过计算相关物理量的关联函数得到。在这里我们介绍几种分子尺度下常用的关联函数以及相应的导出物理量。

首先, 在原子尺度上  $N$  粒子系统的粒子密度为

$$\rho(\mathbf{r}) = \sum_{i=1}^N \delta(\mathbf{r} - \mathbf{r}^{(i)})$$

则定义相对位置为  $\mathbf{r}$  的两个粒子之间的关联函数为

$$C(\mathbf{r}) = \langle \rho(\mathbf{r}_i) \rho(\mathbf{r}_i + \mathbf{r}) \rangle = \frac{1}{N} \sum_{i,j=1, i \neq j}^N \delta(\mathbf{r} - \mathbf{r}_{ij})$$

用体系平均密度  $\rho = \frac{N}{V}$  进行归一, 则定义对关联函数

$$g(\mathbf{r}) = \frac{C(\mathbf{r})}{\rho} = \frac{V}{N^2} \sum_{i,j=1, i \neq j}^N \delta(\mathbf{r} - \mathbf{r}_{ij})$$

径向形式即为

$$g(r) = \frac{1}{2\pi N r^2 \rho} \sum_{j=1, i>j}^N \delta(r - r_{ij})$$

因此，如果粒子的分布是有序的，那么总是会在相同的  $r$  处叠加很多的 delta 函数，因此会出现许多分立的峰值。而结构越无序，这种峰值越不明显。

根据对关联函数的形式，半径为  $r$  处，厚度为  $\Delta r$  的球壳内平均粒子数为

$$n = \frac{N}{V} \int_r^{r+\Delta r} g(r) \cdot 4\pi r^2 dr$$

取  $r = 0$ ， $\Delta r$  为对关联函数第一谷的距离，就可以得到体系最近邻配位数。

对  $g(\mathbf{r})$  做 Fourier 变换，可以得到静态结构因子

$$S(\mathbf{k}) = 1 + \rho \int g(\mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}} dV$$

在静态关联函数中引入时间以来，可以得到动态关联函数

$$A(\mathbf{r}, t) = \sum_{i=1}^N a_i(t) \delta(\mathbf{r} - \mathbf{r}(t))$$

则时间依赖的粒子密度为

$$\rho(\mathbf{r}, t) = \sum_{i=1}^N \delta(\mathbf{r} - \mathbf{r}(t))$$

时空依赖的关联函数为 van Hove 关联函数

$$G(\mathbf{r}, t) = \left\langle \frac{1}{N} \int \rho(\mathbf{r}' + \mathbf{r}, t) \rho(\mathbf{r}', 0) dV' \right\rangle = \left\langle \frac{1}{N} \sum_{i,j=1}^N \delta[\mathbf{r} - \mathbf{r}_j(t) + \mathbf{r}_i(0)] \right\rangle$$

这个求和项中，可以拆成自关联和异关联两部分，分别对应求和项中  $i = j$  和  $i \neq j$  的两部分。详细写为

$$G_s(\mathbf{r}, t) = \left\langle \frac{1}{N} \sum_{i=1}^N \delta[\mathbf{r} - \mathbf{r}_i(t) + \mathbf{r}_i(0)] \right\rangle$$

$$G_d(\mathbf{r}, t) = \left\langle \frac{1}{N} \sum_{i=1, i \neq j}^N \delta[\mathbf{r} - \mathbf{r}_j(t) + \mathbf{r}_i(0)] \right\rangle$$

对空间部分做一次 Fourier 变换，得到中间散射函数

$$F(\mathbf{k}, t) = \int G(\mathbf{r}, t) e^{i\mathbf{k} \cdot \mathbf{r}} dV$$

再进行一次 Fourier 变换，得到动态结构因子

$$S(\mathbf{k}, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\mathbf{k}, t) e^{i\omega t} dt$$

也可以类似定义自关联中间散射函数以及自关联动态结构因子，定义方式是从自关联函数出发做 Fourier 变换。

## 4 Monte Carlo 模拟

Monte Carlo 模拟是假定某一个实验重复进行了相当多次以后, 对这些实验构成的“系综”进行平均的结果。这个系统之所以能够导出这些特定的结果, 是因为系统内禀的性质决定了当重复性实验构成系综时, 系综内的系统都有属于自己的概率分布。在这些概率分布下, 物理量的均值便会反映出这个系统本征的特质。因此使用 MC 进行模拟, 最核心的步骤, 是得到大量的满足这个系统所应该满足的分布的样本。

当我们所选的样本数足够多时, 根据大数定理, 物理量在样本中的算数平均将收敛到在系综内的期望值。

### 4.1 伪随机数的产生

Monte Carlo 抽样的基础是能够产生在  $[0, 1]$  区间内的均匀分布随机数。数学上, 最实用的方法是使用一些数学递推公式产生伪随机数, 只要保证伪随机数的周期足够长, 最大容量足够大, 就可以作为很好的随机数近似。

比较常见的伪随机数产生算法包括线性同余法和乘同余法。在线性同余法中, 采用递推公式

$$x_{i+1} = (ax_i + c) \pmod{m}$$

此方法产生随机序列的最大容量为  $m$ 。若  $c = 0$ , 则一般称之为乘同余法。

### 4.2 从概率分布抽样

设我们所考察的物理系统可以用随机变量  $x$  来标记, 相应的概率密度为  $f(x)$ , 现在我们要考察如何生成满足这样的概率分布的一系列样本  $x_n$ 。

#### 4.2.1 直接抽样

设  $y = F(x)$  是随机变量  $x$  的累积分布函数, 则我们可以通过生成满足均匀分布的  $y_n$ , 然后将其求反函数得到  $x_n = F^{-1}(y_n)$

#### 4.2.2 舍选抽样法

我们构造一个新的概率函数  $g(x)$  使得  $Cg(x) > f(x)$ , 然后我们按照概率分布  $g(x)$  依次生成随机数  $x_n$ 。对于这个随机数  $x_n$ , 我们生成一个在  $[0, Cg(x)]$  上的随机数  $y$ , 若  $y > g(x)$ , 则舍弃, 否则, 我们将这个  $x_n$  保留到随机序列中

#### 4.2.3 变换抽样

设我们需要生成一个分布为  $f(x)$  的随机序列  $x_n$ , 我们可以先生成一个分布为  $g(y)$  的随机序列  $y_n$ , 然后将  $y_n$  映射到  $x_n$ 。简单起见, 我们可以取  $g(y)$  为均匀分布, 根据概率密度的守恒, 我们有

$$g(y)dy = f(x)dx$$

于是

$$f(x) = \left| \frac{dy}{dx} \right|$$

因此,  $y(x)$  满足

$$\left| \frac{dy}{dx} \right| = f(x)$$

一旦找到  $y(x)$ , 则  $x(y)$  满足  $f(x)$  分布。

#### 4.2.4 重要性抽样

设我们需要以  $f(x)$  的分布生成一个随机序列  $x_n$ , 在后面物理问题的讨论中, 这些  $x_n$  所代表的系统在系综中占有相同的权重。我们可以这样做, 我们改用另一个更简单的分布  $g(x)$  生成一个随机序列  $x_n$ , 但是每个随机数都自带一个权重  $\frac{p(x)}{q(x)}$ , 在后面物理问题的讨论中, 所有的平均都要带上这个权重因子, 那么这样抽样的效果是等效的。

例如, 如果我需要算一个多重积分  $I = \int_D f(\mathbf{x}) dV$ , 这相当于以均匀分布  $p(\mathbf{x})$  对  $f(\mathbf{x})$  做平均, 即

$$I = \langle f(\mathbf{x}) \rangle_p = \int_D f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \sim \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i), \mathbf{x}_i \sim p(\mathbf{x})$$

我们可以换一个抽样方法, 在分布  $q(\mathbf{x})$  上抽取  $\mathbf{x}_n$ , 则

$$\langle f(\mathbf{x}) \rangle_p = \int_V f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = \left\langle f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} \right\rangle_q$$

### 4.3 随机游走抽样

#### 4.3.1 随机游走问题

一种最经典的随机游走问题是一维醉汉。认定醉汉每次行走只会经过一个格点, 向左还是向右完全随机, 我们总是可以模拟概率的方式决定每次迭代醉汉的行走方向。

如果进一步要求随机行走是自回避的, 那么需要在所有可能的行走方向上随机选择下一步, 如果遇到了自回避, 就放弃整个行走过程, 重新开始。最终经过多次试验后, 可以保证产生大量的等概率 SAW 构型。

随机游走的每一步, 都会产生一个新的位置。如果我们想象这是发生在相空间中, 这就相当于给我们的系统选取了一个新的随机构型, 而这个随机构型将满足这个系统所需要满足的分布。因此在相空间的随机行走在平衡后的每一步迭代都将给出一个处在平衡的样本。

#### 4.3.2 Metropolis 方法

前面的醉汉随机游走, 相当于选取了随机游走规则为  $w(x_i \rightarrow x_{i+1}) = 1/2, w(x_i \rightarrow x_{i-1}) = 1/2$ 。若设想随机变量  $x$  是在相空间中, 并且我们考虑  $x \rightarrow x'$  的任意过渡几率  $w(x \rightarrow x')$ , 我们需要它在行走中经过相空间的点  $x_1, x_2, \dots$  最终能够收敛到系统的平衡分布  $f(x)$ 。可以证明的是, 如果过渡几率能够满足细致平衡条件

$$f(x)w(x \rightarrow x') = f(x')w(x' \rightarrow x)$$

则随机游走可以达到平衡分布。这是一个充分条件, 因此过渡几率的选取具有任意性。

Metropolis 算法选取的过渡几率为

$$w(x, x') = \min \left[ 1, \frac{f(x')}{f(x)} \right]$$

。具体来讲, 先选取一个初始状态, 然后考察任意游走位点下的相对几率  $r = f(x')/f(x)$ 。若  $r \geq 1$ , 则让此游走发生。如果  $r < 1$ , 就生成一个标注随机数, 如果标准随机数小于相对几率  $r$ , 就让这一步游

走发生，否则不发生。在这样的游走下，产生了大量的相空间内的点，这些状态点最终将收敛到系统的平衡分布。

#### 4.4 随机生长过程——DLA 模型

DLA 模型用于研究分形生长。在一个晶格上，首先选取初始粒子作为种子，并以种子为圆心画半径为  $r$  的大圆。在晶格系统上产生一个粒子，向着各个方向随机游走，一旦游走到种子颗粒的近邻格点，就附着在种子颗粒上。而如果行走到圆的边界或者离开此圆，就被边界吸收，再产生第二个粒子重复上述过程。最后，种子集团会形成一个无规分叉图形。