

# 计算物理总复习

导语：有错误欢迎指正，谢谢。可以随意修改 Word 文档给自己用，不要靠这个东西盈利就行了。事先说明，我根本不能保证覆盖所有考点，如果你想冲满分请参考老师给的所有资料。我就总结了容易考到的地方，但是书里的边边角角是没有的。如果有人像我一样完全不想看这种乱七八糟的 PPT 和不讲人话讲鬼话的课本（建议作者向伏拉夫学中文），只要求及格就行，可以参考一下这份资料。感谢支持。排版确实很难看，因为我也没太多时间，抱歉。

## 蒙特卡洛法计算一维积分

**平均值法：**这个完全没什么难度，对于  $\int_a^b f(x)dx$ ，积分估计值为  $I = (b-a)\frac{1}{N}\sum_{i=1}^N f(X_i)$ ，意思就是你在  $(a,b)$  区间内随机抽取  $N$  次  $X_i$ ，然后计算  $f(X_i)$ ，把这些结果求平均就行，最后乘区间长度  $b-a$ 。方差为  $V = \int_a^b (f(x) - I)^2 dx$ ，如果  $f(x)$  在  $(a,b)$  区间上比较平坦那么方差就小，如果起伏比较大那么方差就大（当然了）。

**重要抽样法：**这个可能不是那么好理解，但是直接记住结论就 OK。对于  $\int_a^b f(x)dx$ ，对他进行变换  $\int_a^b f(x)dx = \int_a^b \frac{f(x)}{g(x)} g(x)dx = \int_a^b f^*(x)g(x)dx$ 。 $g(x)$  需要满足分布函数条件，就是概率论那个，但是记住一定要让  $\int_a^b g(x)dx = 1$ 。然后在  $(a,b)$  区间内随机抽取  $N$  次  $X_i$ ，计算  $I = \frac{1}{N}(b-a)\sum_{i=1}^N \frac{f(X_i)}{g(X_i)}$ 。也就比平均值法多了一个步骤而已。给个例子吧。计算  $\int_0^1 \frac{e^{-x}}{1+x^2} dx$ ，很显然  $f(x) = \frac{e^{-x}}{1+x^2}$ 。随便找一个分布函数  $g(x) = \frac{4}{\pi(1+x^2)}$ ， $x \in (0,1)$  也是  $(0,1)$  定义域的。 $\frac{f(x)}{g(x)} = \frac{\pi e^{-x}}{4}$ ，对这个函数在  $(0,1)$  抽样求平均，然后乘区间长度 1，就 OK 了。方差为  $V = \int_a^b \left(\frac{f(x)}{g(x)} - I\right)^2 g(x)dx$ ， $I$  就是积分估计值。误差平方与方差成正比， $\sigma \propto \frac{1}{\sqrt{N}}$ （这个也记住，我也不知道为什么， $N$  就是抽样的次数）

**投点法：**首先要读一段话，接下来的东西很简单

对于一维积分  $\int_0^1 f(x)dx$ ， $0 \leq x \leq 1$ ， $0 \leq f(x) \leq 1$ ，（如果不满足这些条件，比如  $\int_a^b f(x)dx$ ， $L \leq f(x) \leq M$ ，你可以做变换， $x = \frac{z-a}{b-a}$ ， $g(z) = f\left(\frac{z-a}{b-a}\right)$ ， $g^*(z) = \frac{g(z)-L}{M-L}$ ，就可以变换到  $\int_0^1 g^*(z)dz$ 。当然不用这么麻烦，因为这个就是平移和拉伸，很简单的，只要让定义域和值域都在  $(0,1) \times (0,1)$  的正方形内就行，然后再把这个积分结果乘拉伸比例，就是原来的积分结果了）意思就是，要变换到  $(0,1) \times (0,1)$  区间内。

然后生成两个随机数  $X_1, X_2$ ，都在  $(0,1)$  区间内。如果  $X_1 \leq f(X_2)$  就试验成功，反过来就试验失败。如果  $N$  次试验中有  $m$  次成功，积分估计值就是  $I = m/N$ 。通俗地说就是如果你投

的点在这个函数下方与  $x$  轴上方，那就算进去，如果不是就不算。画个图就完全明白了。投点法的方差比平均值法的方差要大，投点法的方差是  $I - I^2$ （至于为什么他没讲清楚，如果想知道原理可能只能靠你们自己了），平均值法的方差是  $\int_0^1 (f(x) - I)^2 dx$ ，两者相减

$$\begin{aligned} I - I^2 - \int_0^1 (f(x) - I)^2 dx &= I - I^2 - \int_0^1 f(x)^2 dx + 2I \int_0^1 f(x) dx - I^2 \\ &= \int_0^1 f(x)(1 - f(x)) dx \geq 0 \quad (\text{反复使用 } I = \int_0^1 f(x) dx \text{ 就行}) \end{aligned}$$

**分层抽样法：**和重要抽样法一样，你要找到  $f(x)$  和  $g(x)$ ，区间还是在  $(a, b)$  上。具体要求就看重要抽样法那一段落。但是你这回需要分段，把区间分为  $(x_0, x_1), (x_1, x_2) \dots (x_{n-1}, x_n)$  这  $n$  小段，其中  $a = x_0, b = x_n$ 。然后在这  $n$  段上计算  $S_p = \int_{x_{p-1}}^{x_p} g(x) dx$ ， $S_p$  是第  $p$  段上  $g(x)$  对应的面积。（注意  $g(x)$  是已知的，不需要抽样计算  $g(x)$  积分，直接算就行）然后对于  $(x_{p-1}, x_p)$  区间，抽样  $N_p$  次，计算第  $p$  个区间上的积分  $I_p = \frac{1}{N_p} (x_p - x_{p-1}) \sum_{i=1}^{N_p} \frac{f(x_i)}{g(x_i)}$ ，再累加起来  $I = \sum_{p=1}^n I_p$ 。但是这回这个  $N_p$  就是有讲究的了， $N_p$  需要与  $S_p$  成正比，这样才能保证方差最小（其实也能理解， $S_p$  大那么就表明  $g(x)$  在第  $p$  个区间占到的比例就大，那肯定需要更多次抽样了。

## 一维随机变量抽样方法

（这段建议看看）伪随机数序列不满足随机数相互独立的要求，并且存在周期性，可以用数学方法生成。真随机数序列除了统计规律外完全无法预计，是不可重复产生的，需要借助物理随机过程产生，例如不稳定粒子的衰变，计算机信号的噪音。

**线性同余法：**给定  $a, c, M, x_0$  四个数，进行以下的计算

$$\begin{cases} x_n = (ax_{n-1} + c) \bmod M \\ \xi_n = x_n/M \end{cases}$$

就可以产生在  $(0, 1)$  上均匀分布的伪随机数序列（还是希望记忆一下，这次考了，这个分数最好不要丢，反正也不难记忆是不是）

**直接抽样法（如果你觉得废话太多就看这一部分最后一段）：**这里请记住一些概率论里面就应该掌握的公式，不过我早就忘掉了，如果你忘了你一定要看一下。这里面  $\int_{-\infty}^t f_X(x) dx = P(X \leq t)$ ，就是概率密度。对于随机变量  $Y = g(X)$ ，它的概率密度满足：

$$f_Y(y) = \begin{cases} \sum_{i=1}^n \frac{f_X(x_i)}{|g'(x_i)|}, & \text{when } g(x_1) = g(x_2) = \dots = g(x_n) = y \\ 0, & \text{when } g(x) = y \text{ has no solution} \end{cases}$$

当然了光灌输这种东西是没用的，我还不如给个例子，如果

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

然后有  $Y = X^2$ ，计算  $f_Y(y)$ 。

解答：此时  $y = g(x) = x^2$ 。有两个解  $x_1 = \sqrt{y}, x_2 = -\sqrt{y}$  满足  $g(x_1) = g(x_2) = y$ 。

$$f_Y(y) = \frac{f_X(x_1)}{|g'(x_1)|} + \frac{f_X(x_2)}{|g'(x_2)|} = \frac{f_X(\sqrt{y})}{|2\sqrt{y}|} + \frac{f_X(-\sqrt{y})}{|-2\sqrt{y}|} = \frac{1}{\sqrt{2\pi y}} e^{-y/2}, y \in (0, \infty)$$

还有一个就是如果知道了随机变量  $X$ 、 $Y$  的概率密度，那么对于新的随机变量  $Z = X + Y$ ，可以得出

$$f_Z(z) = f_X(z) * f_Y(z) = \int_{-\infty}^{\infty} f_X(w) f_Y(z-w) dw = \int_{-\infty}^{\infty} f_Y(w) f_X(z-w) dw$$

知道以上两个公式可以求解这道题了，如果  $\xi_1, \xi_2, \dots, \xi_n$  是一组在  $(0,1)$  上均匀并且独立同分布的随机变量，而  $\eta = -\frac{1}{a} \ln(\xi_1 \xi_2 \dots \xi_n)$ ，证明  $\eta$  的概率密度函数为

$$f(x) = \frac{a^n}{(n-1)!} x^{n-1} e^{-ax}, x \geq 0$$

解答： $\eta = -\frac{1}{a} \ln(\xi_1) - \frac{1}{a} \ln(\xi_2) - \dots - \frac{1}{a} \ln(\xi_n) = Y_1 + Y_2 + \dots + Y_n$ ，而且  $\xi_k$  的概率密度已经给定，是均匀分布，即  $f_{\xi_k}(\xi_k) = 1, \xi_k \in (0,1)$ 。你需要求出  $Y_k$  的概率密度。（如果后续过程看着别扭，你就把下标  $k$  全部忽略） $y_k = g(\xi_k) = -\frac{1}{a} \ln(\xi_k)$ ，有唯一解  $\xi_k = e^{-ay_k}$ 。所以

$$f_{Y_k}(y_k) = \frac{f_{\xi_k}(\xi_k)}{|g'(\xi_k)|} = \frac{1}{|-1/(a\xi_k)|} = |a\xi_k| = ae^{-ay_k}, y_k \geq 0$$

然后你要卷积  $n$  次，但是没有必要，这里你可以使用数学归纳法。假设  $n = k$  时  $\eta_k = Y_1 + Y_2 + \dots + Y_k$ ，概率密度  $f_{N_k}(x) = \frac{a^k}{(k-1)!} x^{k-1} e^{-ax}, x \geq 0$ ，那么  $n = k + 1$  时  $\eta_{k+1} = (Y_1 + Y_2 + \dots + Y_k) + Y_{k+1} = \eta_k + Y_{k+1}$ ，那么

$$\text{对于概率密度，需要使用第二个公式 } f_{N_{k+1}}(x) = f_{N_k}(x) * f_{Y_{k+1}}(x) = \int_{-\infty}^{\infty} f_{N_k}(w) f_{Y_{k+1}}(x-w) dw = \int_0^x \frac{a^k}{(k-1)!} w^{k-1} e^{-aw} a e^{-a(x-w)} dw = \frac{a^{k+1}}{k!} x^k e^{-ax}, x \geq 0$$

（这里积分区域从  $\mathbb{R}$  变到  $(0, x)$  的原因是  $f_{N_k}(x), f_{Y_k}(y_k)$  只有在自变量大于等于 0 的时候才是上面这种形式，如果自变量小于 0 那么概率密度就为 0 了，所以这个无穷积分里面的被积函数不为 0 的充分条件是  $w >= 0, x - w >= 0$ ，因此  $0 <= w <= x$ ，就可以把积分区域缩短到  $(0, x)$  里）

直接抽样法要解决的问题就在于如果你已经知道一个随机变量的概率密度函数了，那你怎么保证用一个在  $(0,1)$  上生成的随机数去产生满足这个随机变量概率密度函数的一组变量呢？比如用  $(0,1)$  上生成的随机数产生指数分布  $f_Y(y) = ae^{-ay}, y \in (0, \infty)$ 。你可以对  $f_Y(y)$  进行积分  $F_Y(y) = \int_{-\infty}^y f_Y(t) dt = 1 - e^{-ay}$ ，然后求反函数  $F_Y^{-1}(y) = -\frac{1}{a} \ln(1 - y)$ 。于是对于  $(0, 1)$  上均匀分布的随机变量  $f_X(x) = 1, x \in (0,1)$ ， $y = F_Y^{-1}(x)$  与  $y = F_Y^{-1}(1 - x)$ ，即  $Y = -\frac{1}{a} \ln(1 - X)$  与  $Y = -\frac{1}{a} \ln(X)$  都满足  $f_Y(y) = ae^{-ay}$  的条件。证明：此时  $y = g(x) =$

$$F_Y^{-1}(x), f_Y(y) = \frac{f_X(x_0)}{|g'(x_0)|}, \text{ 其中 } g(x_0) = F_Y^{-1}(x_0) = y, x_0 = F_Y(y) \text{。所以 } f_Y(y) = \frac{f_X(x_0)}{|g'(x_0)|} =$$

$$\frac{f_X(x_0)}{|F_Y^{-1}'(x_0)|} = \frac{1}{|F_Y^{-1}'(x_0)|} \text{。这里你要求反函数的导数，对于 } y = f^{-1}(x), \text{ 即 } x = f(y), f^{-1}'(x) =$$

$$\frac{dy}{dx} = \frac{1}{\frac{dx}{dy}} = \frac{1}{f'(y)} \text{。所以 } |F_Y^{-1}'(x_0)| = \left| \frac{1}{F_Y'(y)} \right| = \left| \frac{1}{f'(y)} \right| \text{，得证了（对于 } y = g(x) = F_Y^{-1}(1 - x) \text{ 也同理，但是不用证明了，我被绕晕了）。$$

**第一类舍选法(别的应该不考):**假设 $f(x)$ 在 $(a, b)$ 上有最大值 $L$ , 并且满足 $\int_a^b f(x)dx = 1$ , 就构造新的函数 $g(x) = f(x)/L$ , 让他落在 $(a, b) \times (0, 1)$ 区间内。然后生成两个随机数  $X_1, X_2$ , 对  $X_1$  变换:  $Y_1 = a + (b - a)X_1$ , 让  $Y_1$  落在 $(a, b)$ 区间内。如果  $X_2 < g(Y_1)$ 那么把这个  $Y_1$  (只需要记录  $Y_1$  即可,  $X_2$  不用记录) 记录到所需的样本里, 否则不要记录。在足够多次操作后, 所需样本经过统计后的概率密度函数就大约为 $f(x)$ 。也是画个图就能理解的事情, 如果  $g(a) = 1$ , 那么  $Y_1$  在  $a$  附近时,  $X_2 < g(Y_1)$ 就几乎成为必然。记录进去的  $Y_1$  在  $a$  附近分布就比较密集, 印证了 $f(x)$ 的概率密度。效率为 $\frac{\int_a^b f(x)dx}{L(b-a)} = \frac{1}{L(b-a)}$ (就是  $Y_1$  被记录进去的比例, 画个图就 OK),  $L$  如果比较大效率就很低, 因为 $\int_a^b f(x)dx = 1$ , 如果 $f(x)$ 的最大值比较大, 那么说明他有尖峰, 起伏较大, 效率也自然就低了。所以这个时候这个方法就不太好, 因此适用于 $f(x)$ 平缓的时候。

**Metropolis 方法:** (我看了 CSDN 资料) 对于已知概率密度  $f(x)$  的函数, 在  $f(x)$  定义域上随机抽取一个数  $r_n$ , 将  $r_n$  记录到所需的样本里, 并计算  $f(r_n)$ 。再随机抽取一个数  $r_{n+1}$ , 计算  $f(r_{n+1})$ 。计算  $\alpha = f(r_{n+1}) / f(r_n)$ 。如果  $\alpha$  大于 1, 直接将  $r_{n+1}$  记录到所需的样本里; 如果  $\alpha$  小于 1, 那么首先生成一个在  $(0, 1)$  上的随机变量, 如果  $\alpha$  大于该随机变量, 就将  $r_{n+1}$  记录到所需的样本里, 否则不要记录, 这个  $r_{n+1}$  作废, 需要重新抽取  $r_{n+1}$  重复上述步骤。在足够多次操作后, 所需样本经过统计后的概率密度函数就大约为 $f(x)$ 。可能你不一定能看懂什么意思, 这里直接给出 Python 代码:

```

import math

import random

import numpy as np

import matplotlib.pyplot as plt

def Gaussian(x, mu=0, sigma=1):
    """
    Calculate probability density of Gaussian distribution at point x.

    Args:

        x: The point for calculating probability density.

        mu: mean of Gaussian distribution.

        sigma: Stand deviation of Gaussian distribution.

    """
    return math.pow(math.e, -math.pow(x-mu, 2)/(2*math.pow(sigma, 2)))/(sigma * math.sqrt(2*math.pi))

def uniform_sampling(a, b):
    """ Sample from uniform distribution """
    return a + (b-a) * random.random()

num_samples = 500 # total number of samples to be generated

# Number of iterations to reach the stationary distribution (guessed value).

# You can test different value to see whether the sample mean and stand deviation are close to the true values.

skips = 500

samples = []

x = 0 # the initial value

counter = 0

while len(samples) < num_samples:
    accept = False

    while not accept:
        sample = uniform_sampling(-50, 50) # first get a sample from uniform distribution

        alpha = min(Gaussian(sample)/Gaussian(x), 1) # then calculate density ratio of the sample and current points

        # if the randomly generated number (probability) smaller than or equal to the alpha, accept the sample;

        # otherwise, re-sample a new point.

        p = random.random()

        if p <= alpha:
            x = sample

            accept = True

            if counter >= skips:
                samples.append(sample)

            counter += 1

print(samples)

print("mean: " + str(np.mean(samples)))

print("std: " + str(math.sqrt(np.var(samples))))

times = []

for i in range(1, num_samples + 1):
    times.append(i)

plt.plot(times, samples)

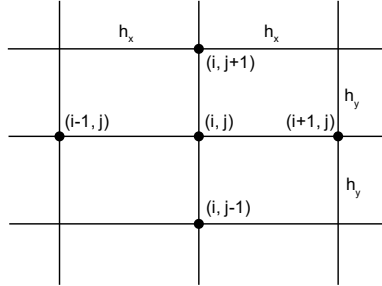
plt.show()

```

## 数值求解偏微分方程

五点差分法（或菱形格式，两个意思是一样的）求解下面的偏微分方程：

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + f\phi = q$$



$h_x$  是  $x$  方向的步长,  $h_y$  是  $y$  方向的步长, 下标  $(i, j)$  代表在格点  $(i, j)$  上函数的取值。

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{(i,j)} \rightarrow \frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h_x^2}$$

$$\left. \frac{\partial^2 \phi}{\partial y^2} \right|_{(i,j)} \rightarrow \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h_y^2}$$

所以

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h_x^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h_y^2} + f_{i,j}\phi_{i,j} = q_{i,j}$$

特殊情况  $h_x = h_y = h, f = 0$ , 那就转化成泊松方程了

$$\phi_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - h^2 q_{i,j})$$

如果题目问你的是拉普拉斯方程, 那么  $q_{i,j}$  也没了 (但愿你们没有忘记数理方程)

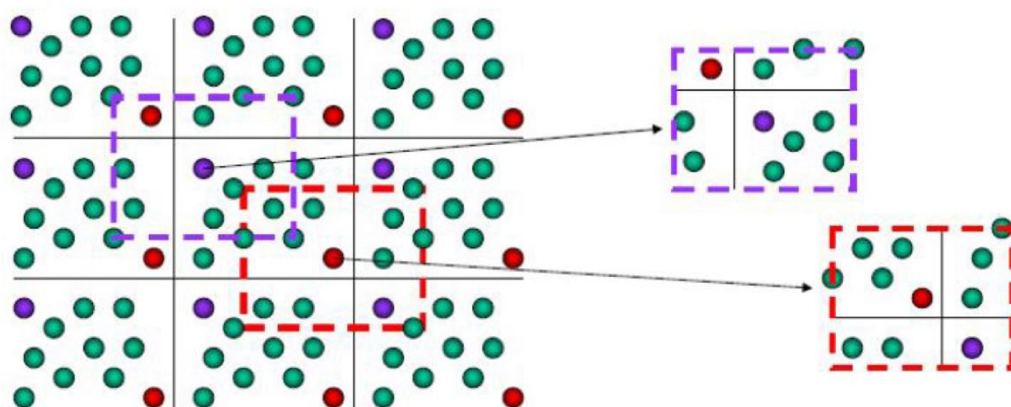
**有限元方法:** 2022 秋季期末考了有限元法的优缺点, 别的关于有限元法就没考。我这一章一点都没看懂, 关于优缺点, 你们自己点击这个链接看一下吧。[离散元与有限元优缺点 - 知乎 \(zhihu.com\)](https://www.zhihu.com/question/35411111)

## 分子动力学模拟

(最好背一下) 两个分子之间的相互势能为  $U = 4\epsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right]$ 。如果题目问你  $\epsilon, \sigma$  分别表达什么物理意义, 你就说  $\epsilon$  确定能量单位, 其中  $-\epsilon$  是势能的最小值;  $\sigma$  确定长度单位,  $r = \sigma$  时势能为 0,  $r = 2^{\frac{1}{6}}\sigma$  时势能取到最小 (可以现场推导, 将  $U$  对  $r$  进行微分求解函数最值)。

**微正则系综:** 2022 秋季期末考了这个概念, 我就添加一下。微正则系综广泛被应用在分子动力学模拟中。假定  $N$  个粒子处在体积为  $V$  的盒子内, 并固定总能量 ( $E$ )。此时, 系综的温度 ( $T$ ) 和系统压强 ( $P$ ) 可能在某一平均值附近起伏变化。平衡体系为孤立系统, 与外界即无能量交换, 也无粒子交换。

最小像力与周期性边界条件：直接看图



这里有一些排列成网格状的正方形，把它称为分子动力学元胞，它的边长为  $L$ 。分子动力学整个模拟就是在一个元胞内进行的，不可能模拟无穷大的系统，目的是维持恒定的密度。最中间的正方形就代表被模拟的分子的实际位置。但是你看周围的正方形是不是与中间一样都完全相等？没错，你需要在周围都进行复制粘贴操作（周期性边界条件）。然后以需要计算的分子（红色与紫色分子）为中心，画一个半径为  $r_c$  的圆形（图中是虚线的正方形），这样就可以完全忽略掉圆形（对应图中虚线正方形）以外的分子，只需要计算圆形（对应图中虚线正方形）以内的分子给他的作用力就 OK 了，换句话说  $r > r_c$  的分子就可以忽略。如果问你该分子与多少个分子相互作用，你只需要把圆形（对应图中虚线正方形）里面除自己以外的分子数一下就可以。

**Verlet 算法：**假设圆形（对应图中虚线正方形）内有  $N$  个分子。对于第  $i$  个分子位置和速度迭代公式为

$$\begin{cases} \mathbf{r}_i^{(n+1)} = \mathbf{r}_i^{(n)} + \mathbf{v}_i^{(n)}h + \mathbf{F}_i^{(n)}h^2/2m \\ \mathbf{v}_i^{(n+1)} = \mathbf{v}_i^{(n)} + (\mathbf{F}_i^{(n+1)} + \mathbf{F}_i^{(n)})h/2m \end{cases}$$

不用死记硬背，你就记住高中那个运动公式  $x = x_0 + v_0t + \frac{at^2}{2} = x_0 + v_0t + \frac{Ft^2}{2m}$ ,  $v = v_0 + at = v_0 + \frac{Ft}{m}$ 。其中  $h$  是时间步长， $\mathbf{r}_i^{(0)}, \mathbf{v}_i^{(0)}$  都是一开始给定的初始条件， $\mathbf{r}_i^{(n)}, \mathbf{v}_i^{(n)}$  是第  $n$  步的位置和速度（换句话说就是经过了  $n \times h$  的时间计算得到的位置和速度）。 $\mathbf{F}_i$  和  $\mathbf{r}_i, \mathbf{v}_i$  有关，公式如下（考场现推也行，我是根本记不住这个）

$$\mathbf{F}_i = -\nabla U = 48 \frac{\epsilon}{\sigma^2} \sum_{j=1, j \neq i}^N (\mathbf{r}_i - \mathbf{r}_j) \left[ \left( \frac{\sigma}{r_{ij}} \right)^{14} - \frac{1}{2} \left( \frac{\sigma}{r_{ij}} \right)^8 \right]$$

$r_{ij}$  是第  $i$  个与第  $j$  个分子之间的距离。

机器学习我就没看，这个很抱歉，只能靠你们自己了，我是准备完全放弃这里。**Mathematica** 的话你就多看看帮助文档，或者看一下作业自己总结一下，这种软件的东西多操作就 OK。高性能计算就看 PPT 吧，我不准备总结了，我根本不知道考哪里。