

计算物理A第一次作业

王铠泽 PB18020766

1 作业题目

- 用Schrage方法编写随机数子程序，用连续两个随机数作为点的坐标值绘出若干点的平面分布图。再用 $\langle x_k \rangle$ 测试均匀性。(取不同量级的N值，讨论偏差与N的关系)、 $C(l)$ 测试其2维独立性。(总点数 $N > 10^7$)。

2 实现方法

- Lehmer线性同余随机数产生器

$$I_{n+1} = (aI_n + b) \bmod m$$

$$x_n = I_n / m$$

在本次实验中，主要采用的是16807产生器(最低标准产生器)，即 $a = 16807, b = 0, m = 2^{31} - 1$

- Schrage方法

为了在计算过程中中间数据不溢出，使用Shrage方法来求取余数。

$$az \bmod m = \begin{cases} a(z \bmod q) - r[z/q] & a(z \bmod q) - r[z/q] \geq 0 \\ a(z \bmod q) - r[z/q] + m & \text{otherwise} \end{cases}$$

- 独立性检验

本次实验中，通过随机变量 x 的各阶矩和关联函数，以及检验统计量 χ^2 来验证独立性。理想情况下，

$$\langle x^n \rangle = \frac{1}{n+1}$$

$$C(l) = \frac{\langle x_n \rangle \langle x_{n+l} \rangle - \langle x_n \rangle^2}{\langle x_n^2 \rangle - \langle x_n \rangle^2} = 0$$

3 程式说明

- Schrage.c

该程式是使用基于16807产生器生成指定数目(N)随机数并计算各阶矩和关联系数的程式。

包含以下函数：

`int shrage (int a, int m, int In)`

返回值是 $aI_n \bmod m$

`int initial (int n)`

$n = 0$ 为默认种子 $I_0 = 1$, $n = 1$ 为时间种子生成 I_0

```
int main()
```

main函数分为三个模块，分别是生成随机数，计算各阶矩(1-4阶)和关联函数[C(0)-C(20)]。

- Correlation.c

这是后续用于生成大量的关联系数的程式。由此生成 $10^4, 10^5$ 个 $C(l)$ 探究规律。

- time_seed.txt

该文本文件显示的是最后一次调用时间种子时对应的原始数据。

- data.txt

这类文本文件包括了不同 N 下产生的随机数和各类相关物理量的详细数据。

4 计算结果

在调用时间种子情况下，得到如下结果。

4.1 散点分布情况

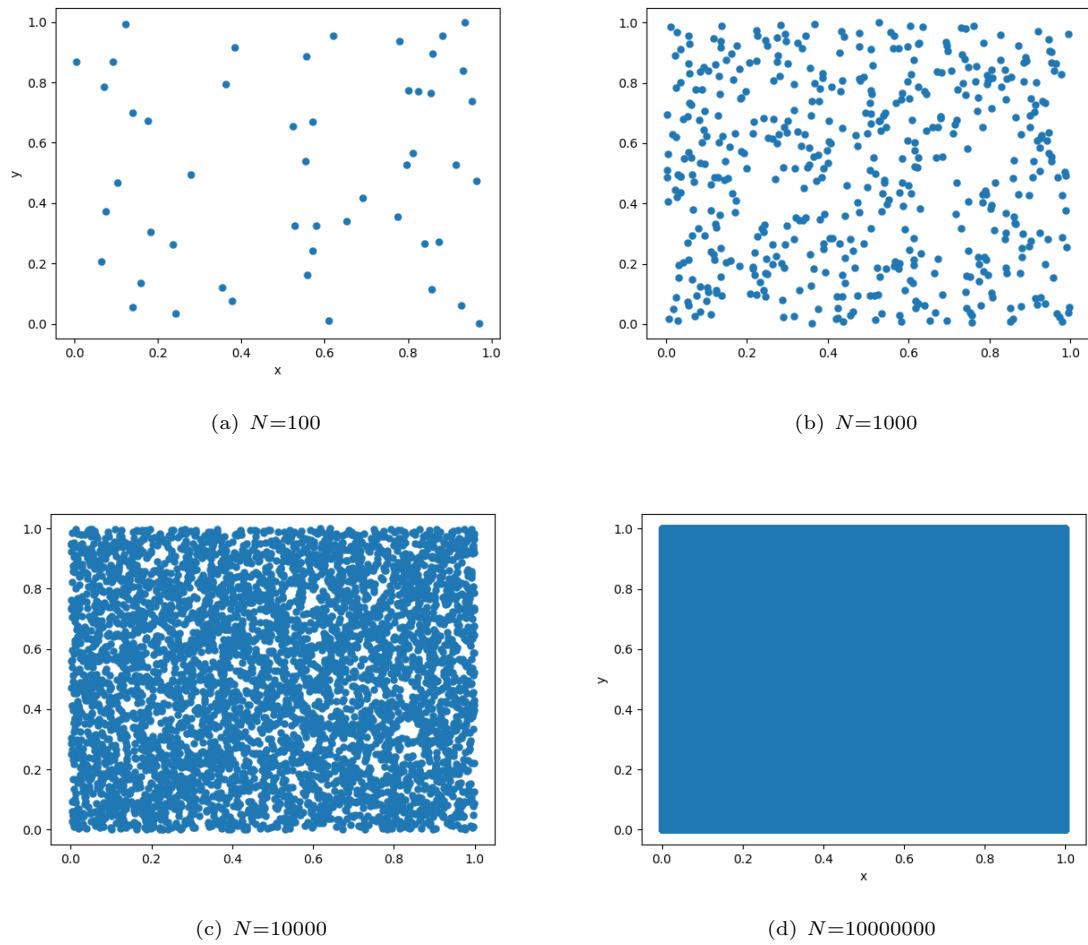


图 1: 不同 N 下的散点分布情况

当点数达到10000量级以后，可以认为是均匀铺满二维平面，说明16807产生器生成随机数比较理想。

4.2 不同抽样点数 N 下各阶矩和理想情况的比较

k	x^k	$x^k - \frac{1}{k+1}$	$\frac{1}{\sqrt{N}}$
1	0.511107	0.011107	0.100000
2	0.352723	0.019389	0.100000
3	0.268752	0.018752	0.100000
4	0.215926	0.015926	0.100000

表 1: $N=100$ 的 $\langle x^k \rangle$ 平均值和误差

k	x^k	$x^k - \frac{1}{k+1}$	$\frac{1}{\sqrt{N}}$
1	0.496715	-0.003285	0.010000
2	0.331324	-0.002009	0.010000
3	0.248791	-0.001209	0.010000
4	0.199217	-0.000783	0.010000

表 2: $N=1000$ 的 $\langle x^k \rangle$ 和误差

k	x^k	$x^k - \frac{1}{k+1}$	$\frac{1}{\sqrt{N}}$
1	0.499714	-0.000286	0.010000
2	0.334310	0.000976	0.010000
3	0.251595	0.001595	0.010000
4	0.201846	0.001846	0.010000

表 3: $N=10000$ 的 $\langle x^k \rangle$ 和误差

k	x^k	$x^k - \frac{1}{k+1}$	$\frac{1}{\sqrt{N}}$
1	0.499963	-0.000037	0.0003162
2	0.333304	-0.000029	0.0003162
3	0.249988	-0.000012	0.0003162
4	0.200005	0.000005	0.0003162

表 4: $N=10000000$ 的 $\langle x^k \rangle$ 和误差

通过上述分析可得到产生随机数的矩和理想值之误差满足 $\delta x = O(\frac{1}{\sqrt{N}})$, 甚至还略小一两个数量级。

4.3 用 $C(l)$ 测试其独立性

$$\frac{1}{\sqrt{N}} = 0.0003162$$

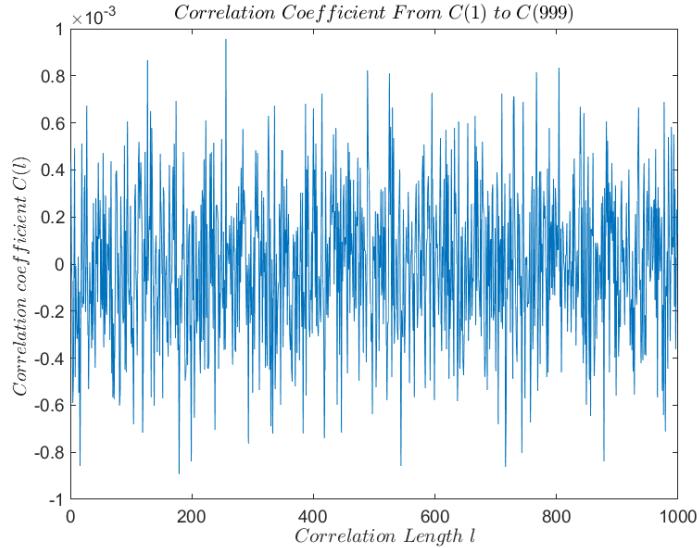
k	$C(k)$	k	$C(k)$
0	1.000000	1	0.000280
2	0.000218	3	-0.000389
4	-0.000154	5	-0.000148
6	-0.000488	7	-0.000273
8	-0.000010	9	0.000426
10	0.000516	11	-0.000012
12	-0.000010	13	0.000028
14	-0.000522	15	0.001037
16	0.000134	17	0.000220
18	0.000200	19	0.000340
20	0.000354		

表 5: $N = 10000000$ 时关联函数 $C(k)$ 的取值

N 比较大 ($N \geq 10^7$) 的情况下, 关联很弱, 基本上都在 $O(\frac{1}{\sqrt{N}})$ 量级。

为了探究 $C(l)$ 的分布, 分别生成了 $N = 10^4$ 和 $N = 10^5$ 个关联系数 $C(1) \sim C(N)$ 。下面给出一些计算结果。

4.3.1 $C(l) - l$ 关系

图 2: $C(l) - l$ 函数图线

分布较为杂乱, 考虑做出其频谱图:

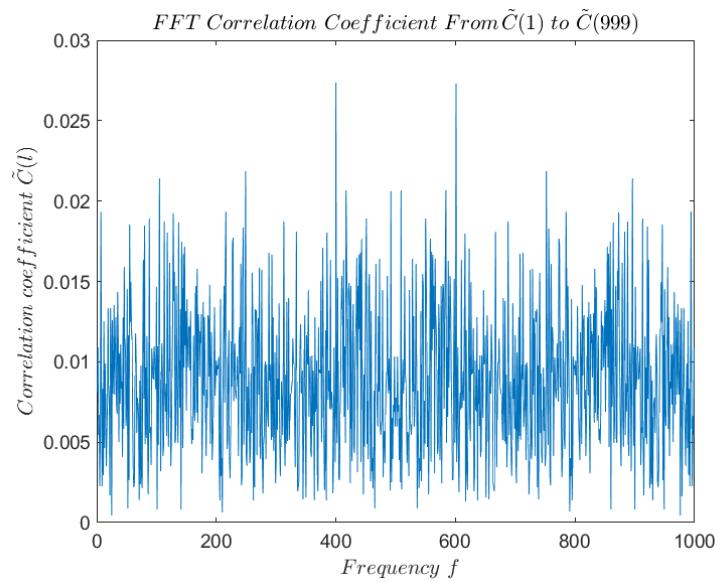


图 3: 频域 $\tilde{C}(l) - l$ 函数图线

频域上虽然也是一个剧烈震荡的分布，但是可以看出是左右对称并且可能有周期分布。目前暂时没有看出其他关于 $C(l) - l$ 关系的特别的性质。

4.3.2 误差分布

如果将各个 $C(l)$ 地位等同起来，只看误差的直方图分布，我们可以得到更清晰有趣的结论。看起来误差的分布像是一个正态分布。

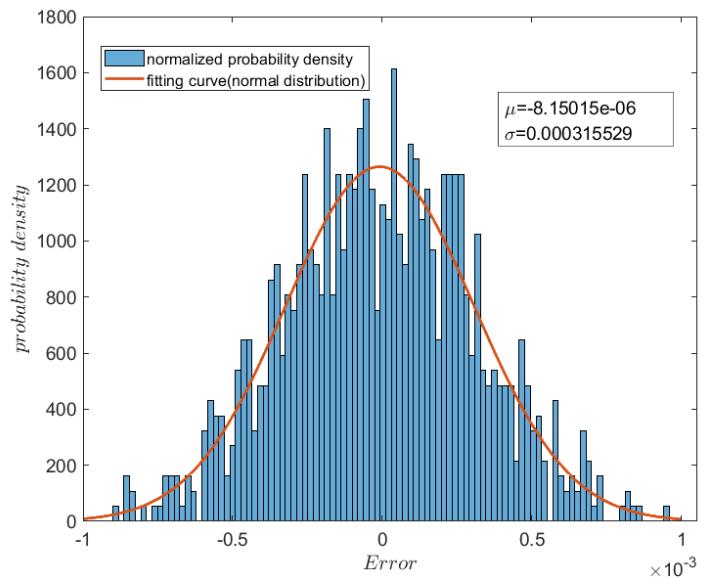
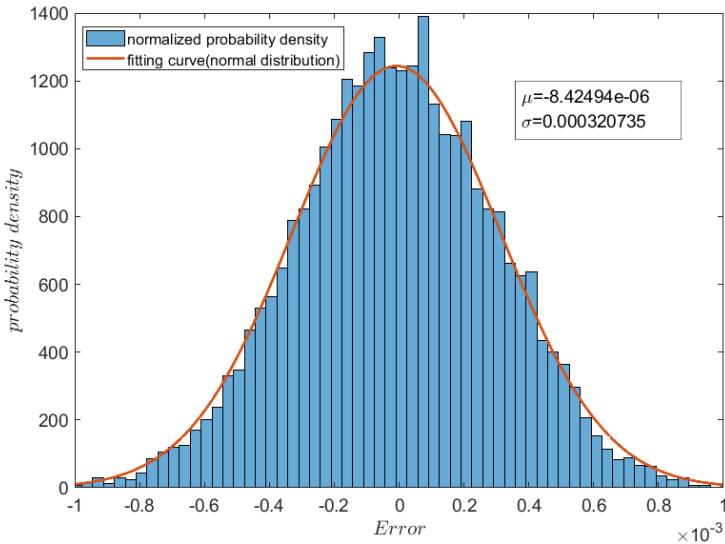
(a) 关系数总个数 = 10^4 (b) 关系数总个数 = 10^5

图 4: 误差分布

这个正态分布的参数值 $\mu \approx 0, \sigma \approx 0.00032$ 。这符合我们对多个随机序列检验的认知。具体的数学证明还有待推导。

4.4 χ^2 检验

对于默认种子 $I_0 = 1$, 生成 10^7 个随机数。将二维平面上 $[0,1] \times [0,1]$ 区域分成 $5 \times 5 = 25$ 个小正方形, 进行自由度为 24 的 χ^2 检验。得到的 $\chi^2 = 16.732340$ 。查阅资料得知:

$$\chi_{24}(0.995) = 9.886, \chi_{24}(0.99) = 10.856, \chi_{24}(0.975) = 12.401, \chi_{24}(0.95) = 12.848, \chi_{24}(0.90) = 15.659$$

选用不同时间种子统计出的 χ^2 也并不相同, 有时能达到 13.9000 这样的数值。比较可知, 虽然在其他检验中 16807 产生器还是算比较良好, 但是在 χ^2 检验下, 其只有大约 90% 的置信概率。

5 其他

选择好的线性同余器参数尤为重要，一组好的参数能大大提高重复周期。一般而言，不恰当的 a, b 取值会导致二维分布是在几条分立线上，而非几乎铺满整个平面。下面给出的是 $a = 7, b = 2$ 的结果。

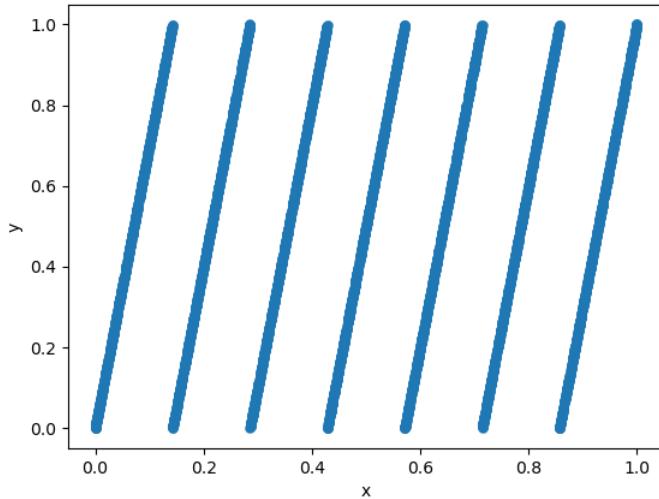


图 5: $a = 7, b = 2, N = 10000$

但是值得一提的是，这组随机数的各阶矩和关联系数都体现出较为良好的特征。

k	x^k	$\frac{1}{k+1} - x^k$	$\frac{1}{\sqrt{N}}$
1	0.498572	0.001428	0.010000
2	0.331881	0.001452	0.010000
3	0.248678	0.001322	0.010000
4	0.198830	0.001170	0.010000

表 6: $a = 7, b = 2, N = 10000$ 时的矩

k	$C(k)$	k	$C(k)$
0	1.000000	1	0.160635
2	0.034297	3	0.014033
4	0.001413	5	0.009139
6	0.008559	7	-0.007316
8	-0.000741	9	-0.016261
10	-0.004097	11	-0.013075
12	-0.000324	13	-0.007607
14	-0.013594	15	-0.016038
16	0.017166	17	-0.002381
18	-0.005216	19	-0.022513
20	-0.003828		

表 7: $N = 10000$ 时关联函数 $C(k)$ 的取值

$C(l)$ 表示的关联比较弱。

在进行 χ^2 检验时, $\chi^2 = 735.540000$, 明显这不是一组独立随机数。

6 总结

- 16807产生器生成的随机数序列随即独立性比较良好, 在置信概率大约90%下可以认为其是独立的。
- 良好随机数关联函数的误差分布可能是一个正态分布。
- 单独一种独立性检验都不能非常完美地刻画一组随机数的性质良好与否, 上面所举的就是一个很好的反例。

计算物理A第二次作业

王铠泽 PB18020766

1 作业题目

- 用16807产生器测试随机数序列中满足关系 $X_{n-1} < X_{n+1} < X_n$ 的比重。讨论Fibonacci延迟产生器中出现这种关系的比重。

2 实现方法

- Lehmer线性同余随机数产生器

$$I_{n+1} = (aI_n + b) \bmod m$$

$$x_n = I_n / m$$

在本次实验中，主要采用的是16807产生器(最低标准产生器)，即 $a = 16807, b = 0, m = 2^{31} - 1$

- Schrage方法

为了在计算过程中中间数据不溢出，使用Shrage方法来求取余数。

$$az \bmod m = \begin{cases} a(z \bmod q) - r[z/q] & a(z \bmod q) - r[z/q] \geq 0 \\ a(z \bmod q) - r[z/q] + m & \text{otherwise} \end{cases}$$

- Fibonacci产生器

这是一种延迟产生器，首先种下 $\max\{p, q\}$ 个种子，然后根据如下公式迭代产生随机序列：

$$I_n = I_{n-p} \otimes I_{n-q}$$

其中 \otimes 可以代表加法，减法或者XOR运算等。本次实验中，采用16807产生器和其默认种子 $I_0 = 1$ 产生 Fibonacci 产生器的前 $\max\{p, q\}$ 个数据，运算符采用加法运算符。

实验中， (p, q) 取如下建议经验值：(24, 55), (38, 89), (37, 100), (30, 127)

3 程式说明

- Fibonacci.c

该程式分别通过16807产生器和不同 (p, q) 组合下Fibonacci产生器产生100000个随机数并且统计题目要求关系比例。

包含以下函数：

`int shrage (int a, int m, int In)`

返回值是 $aI_n \bmod m$

`int initial (int n)`

$n = 0$ 为16807产生器默认种子 $I_0 = 1$, $n = 1$ 为时间种子生成 I_0 。本实验中采用默认种子。

```
int main()
```

main函数分为三个模块，分别是用16807产生器生成随机数，用不同 (p, q) 组合的Fibonacci产生器生成随机数，计算比例。

- ratio.txt

该文本文件记录了16807产生器和不同的Fibonacci产生器对应的关系比例。

4 计算结果

在理想条件下，得到 $X_{n-1} < X_{n+1} < X_n$ 等价于在 $[0, 1] \times [0, 1] \times [0, 1]$ 区域上找到 $x < z < y$ 区域的概率。其体积为一个四棱锥，为 $\frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$ 。如下图所示区域即为所求。

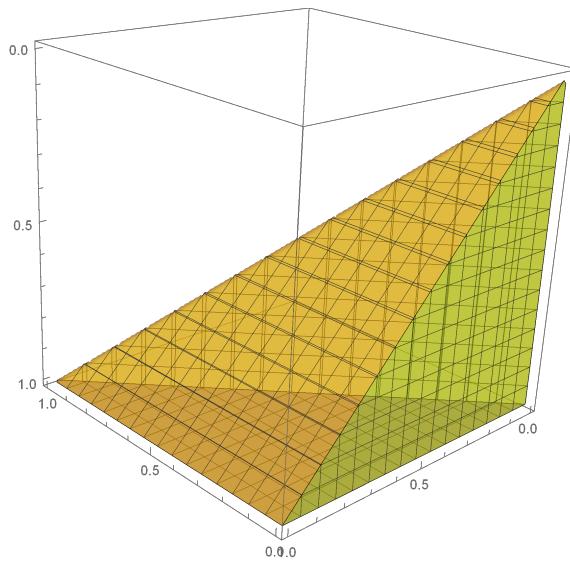


图 1: 理想期望值

4.1 比例关系比较

产生器	$X_{n-1} < X_{n+1} < X_n$ 比例	和理想比例之差绝对值
16807	0.165300	0.001367
Fibonacci, $(p, q) = (55, 24)$	0.165980	0.000687
Fibonacci, $(p, q) = (89, 38)$	0.165700	0.000967
Fibonacci, $(p, q) = (100, 37)$	0.165840	0.000827
Fibonacci, $(p, q) = (127, 30)$	0.168300	0.001633
Fibonacci, $(p, q) = (250, 103)$	0.166190	0.000477
Fibonacci, $(p, q) = (31, 3)$	0.168060	0.001393

表 1: 不同产生器 $X_{n-1} < X_{n+1} < X_n$ 比例

5 总结

- *Fibonacci*产生器的质量一定程度上依赖于 (p, q) 的选择。本次实验中， $(250, 103)$ 的产生器效果最理想。而 $(127, 30)$ 效果一般。相比之下，满足准则： $p^2 + q^2 + 1 = \text{prime number}$ 的 $(31, 3)$ 效果一般。很多建议的 (p, q) 数值都不满足该准则，所以*Fibonacci*参数选择可能是一个比较复杂的经验依赖性工作。
- 总体上看，*Fibonacci*产生器比16807产生器效果略好。
- *Fibonacci*由于需要的种子数量较多，其对随机数生成的质量有影响。本次实验采用16807产生种子。

计算物理A第三次作业

王铠泽 PB18020766

1 作业题目

- 在球坐标系 (r, θ, ϕ) 下产生球面上均匀分布的随机坐标点，给出其直接抽样方法。

2 实现方法

- 直接抽样法

对于分布 $f(\theta, \phi)$ 进行抽样，则有：

$$\int_0^\pi d\theta \int_0^{2\pi} d\phi f(\theta, \phi) \sin\theta = \int_0^\pi \Theta(\theta) \sin\theta d\theta \int_0^{2\pi} \Phi(\phi) d\phi = 1 \cdot 1 = 1$$

球面的均匀分布：

$$\int d\Omega f(\theta, \phi) = \int 2\pi d\theta d\phi = 1$$

其中 Ω 是立体角元

由上面可得：

$$f(\theta, \phi) = \frac{1}{4\pi}, \Theta = \frac{\sin\theta}{2}, \Phi(\phi) = \frac{1}{2\pi}$$

直接抽样法：

设 ξ_1, ξ_2 是 $[0, 1]$ 上生成的两组独立的伪随机数。则：

$$\xi_1 = \frac{\phi}{2\pi}, \xi_2 = \frac{1 - \cos\theta}{2}$$

即：

$$\phi = 2\pi\xi_1, \cos\theta = 1 - 2\xi_2$$

换算成直角坐标时：

$$\begin{cases} x = \sin\theta \cos\phi \\ y = \sin\theta \sin\phi \\ z = \cos\theta \end{cases}$$

- 算法思路

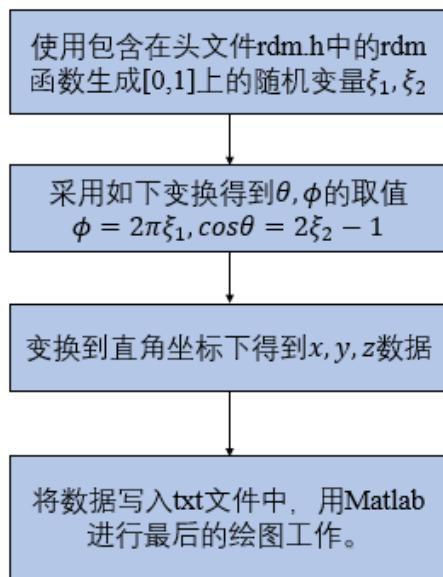


图 1: 算法框图

3 程式说明

- `direct_sample.c`

该程式是用于生成球面上均匀分布的直接抽样程序。

包含以下内容：

`rdm.h`

这是一个包含了使用时间种子/默认种子的16807产生器生成随机数的头文件。

```
void rdm(int N, double*x, int method)
```

method = 0 为默认种子, $I_0 = 1$; *method = 1* 为时间种子生成16807的首个数据 I_0 。本程序中为了保证独立性, ξ_1, ξ_2 两个数组各自分别用默认种子和时间种子生成。

```
int main()
```

main函数分为两个模块, 分别是生成 $[0, 1]$ 上随机序列 ξ_1, ξ_2 , 得到 $\cos\theta, \phi$ 序列, 得到直角坐标 x, y, z 数据并写入文件。

- `time_seed.txt`

该文本文件显示的是调用时间种子时对应的原始数据。每调用一次便写入一次。检查时以最后一次生成的时间为准。

- `coordinate.txt`

该文本文件包括了球面上均匀抽样的 (x, y, z) 数据。

4 计算结果

总点数 $N = 10000$,计算结果如下:

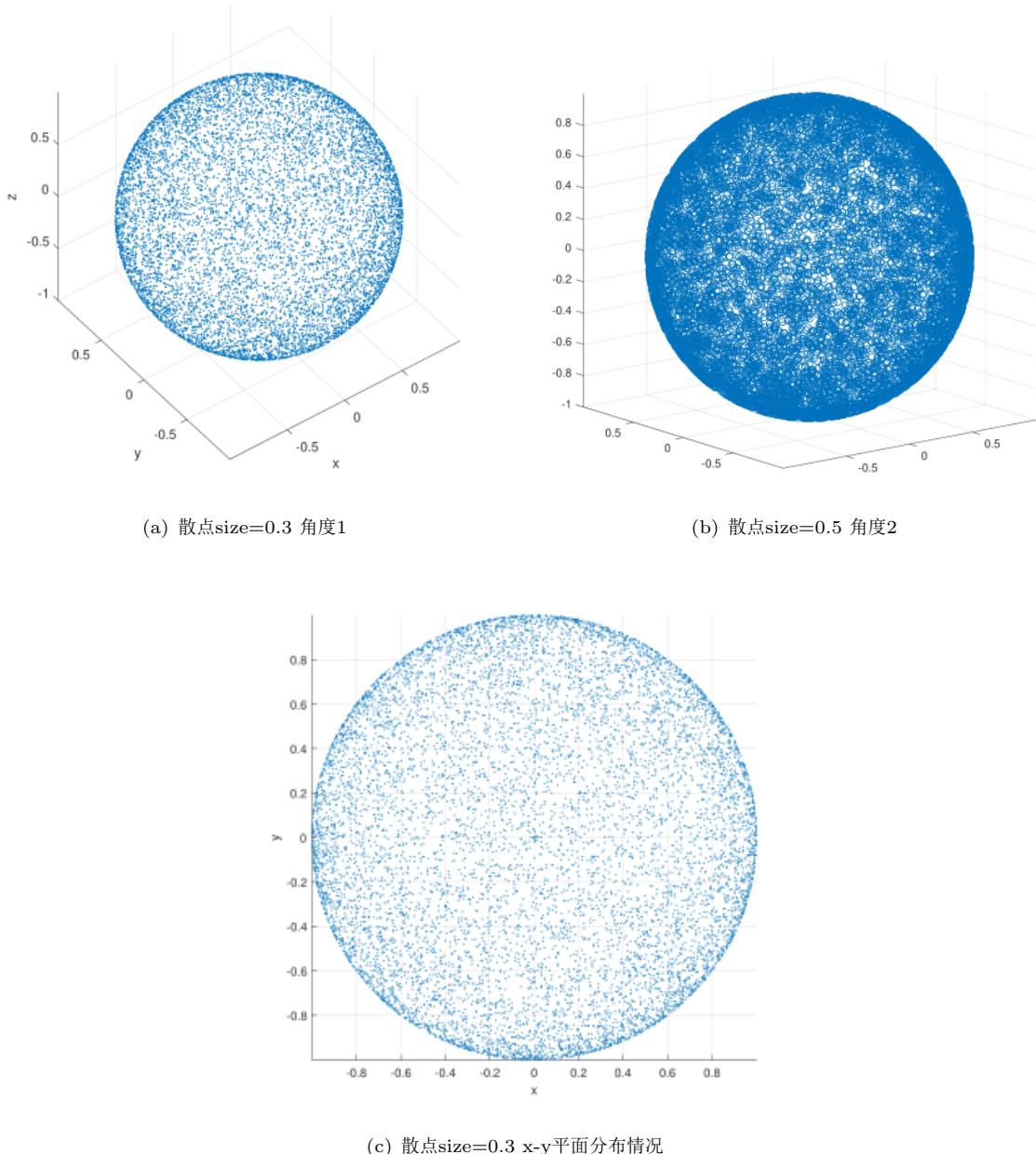
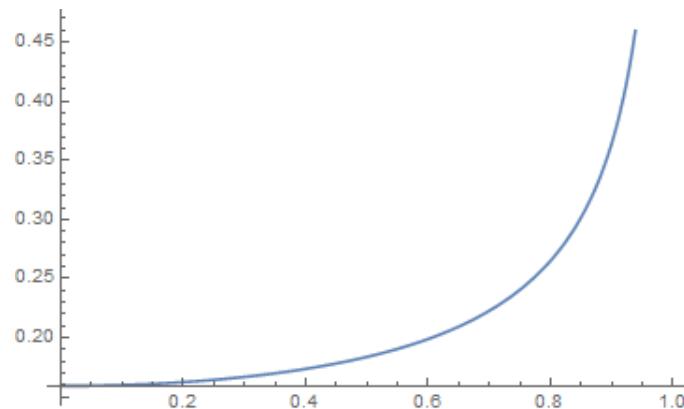


图 2: 不同角度下的散点分布情况

可以看出, 球面上的均匀分布投影到二维平面上体现出外围密集(边缘发散), 中间稀疏的特征。这是因为在二维平面上, 考虑到上下球面投影一致, 有:

$$\begin{aligned} g(x, y) dx dy &= f(\theta, \phi) d\theta d\phi = \frac{1}{4\pi} \sin\theta d\theta d\phi \\ \Rightarrow g(x, y) &= \frac{1}{2\pi} \left| \frac{\partial(\theta, \phi)}{\partial(x, y)} \right| = \frac{1}{2\pi} \frac{1}{\sqrt{1 - r^2}} \end{aligned}$$

所以在二维平面上的外围会出现发散的特征。密度随着 r 增大而逐渐增大。

图 3: $g(r)$ 函数图像

5 总结

- 直接抽样法简单直观，对于简单的密度分布函数是行之有效的方法。
- 不同空间的“均匀”差别很大，根本原因在于度量不一样，体现在Jacobi行列式带来的变换因子。因此球面上的均匀嵌入到三维空间看起来不像是“均匀的”。

计算物理A第四次作业

王铠泽 PB18020766

1 作业题目

- 设概率密度函数满足关系式:

$$p(x) = \frac{dp(x)}{dx} \frac{x - d}{ax^2 + bx + c}$$

请找到其中的一种函数，讨论其性质，并给出抽样方法。

- 尝试两个不同组合 $(a, b, c, d) = (1, 1, 1, 1) / (-1, 1, 1, 1)$ ，得到一般的图像大概如下:

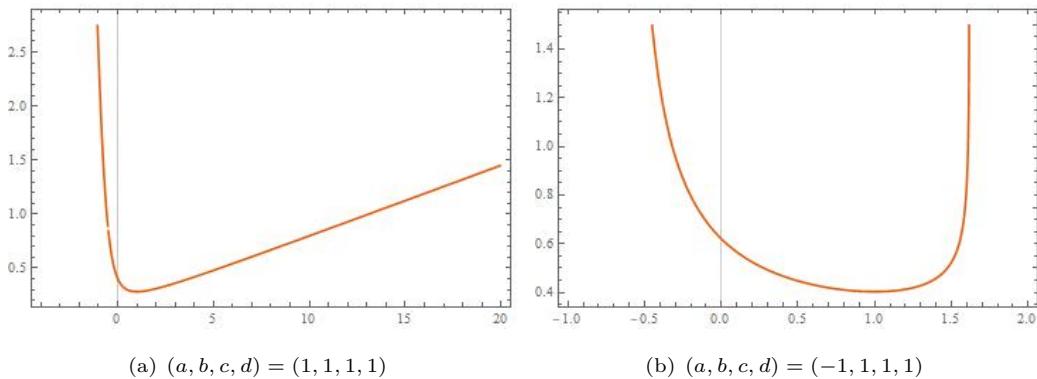


图 1: 函数形态

2 实现方法

本实验着重讨论两种情况，分别提取出分母平方项和一次方项的特征，令

$$(a, b, c, d) = (-1, 0, 1, 0), (a, b, c, d) = (0, -2, 1, 0)$$

即:

$$\begin{aligned} p_1(x) &= \frac{1}{\pi} \frac{1}{\sqrt{1-x^2}} \cdot I(-1 \leq x \leq 1) \\ p_2(x) &= \frac{1}{A} \frac{e^{-x}}{(x-2)^2} \cdot I(-1 \leq x \leq 1) \end{aligned}$$

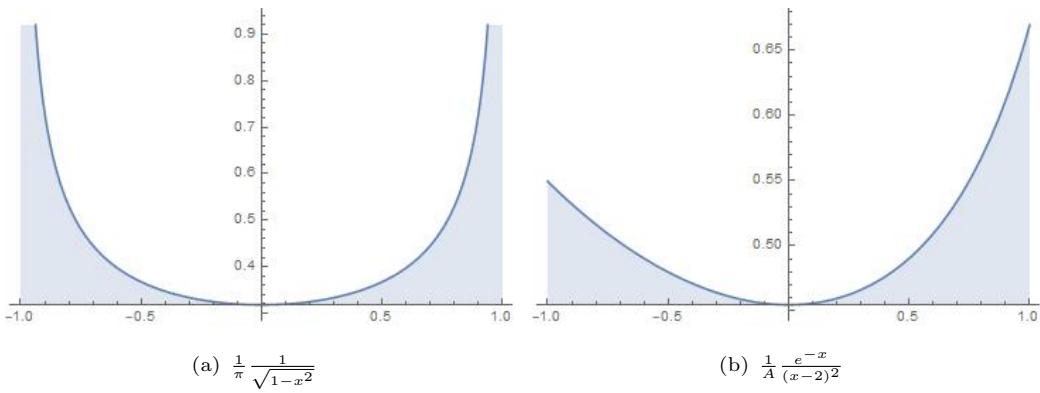
其中 $I(-1 \leq x \leq 1)$ 为示性函数，表示对 x 取值的限制。其取值为：

$$I(-1 \leq x \leq 1) = \begin{cases} 1 & -1 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

后面将自动略去示性函数，默认 $p(x)$ 定义在 $[-1, 1]$ 上。

使用 *Mathematica* 的数值积分功能得到 $A = 0.549707$, $\frac{1}{A} \approx 1.81915$ 。

函数图像如下：

图 2: $p_1(x), p_2(x)$ 函数图像

在本次实验中，采用的是16807产生器(最低标准产生器)，即 $a = 16807, b = 0, m = 2^{31} - 1$ 。

- 直接抽样法

对于 $p_1(x) = \frac{1}{\pi} \frac{1}{\sqrt{1-x^2}}$ 采用简单抽样。先求累计函数 $F(x)$:

$$F(x) = \int_{-1}^x p_1(\xi) d\xi = \frac{1}{\pi} \arcsin(x) + \frac{1}{2}$$

采用16807产生器生成随机序列 ξ , 则目标抽样为

$$x = \sin[\pi(\xi - \frac{1}{2})]$$

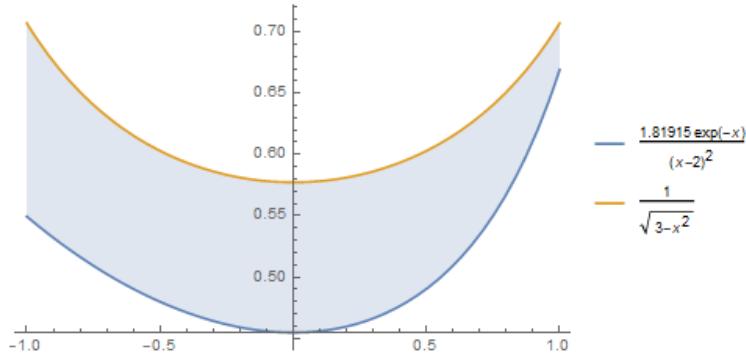
这和 $x = \sin(2\pi\xi)$ 的抽样等价。

- 舍选抽样法

由于对于 $p_2(x) = \frac{1}{A} \frac{e^{-x}}{(x-2)^2}$ 没有解析的累计函数，另一方面若想采用变换抽样法比较难以找到合适的变换使得 $|J| = p_2(x)$ 。所以只能舍弃一些效率选用舍选抽样法。

采用 $F(x) = \frac{1}{\sqrt{3-x^2}}$ 作为覆盖 $p_2(x)$ 的比较函数。 $I = \int_{-1}^1 F(x) dx = 2\sin^{-1}(\frac{1}{\sqrt{3}})$ 。该舍选效率为:

$$\text{Area}[p_2(x)]/\text{Area}[F(x)] \approx 0.812374$$

图 3: $F(x), p_2(x)$ 函数图像

抽样方法: 生成两个 $[0, 1]$ 上随机序列 ξ_1, ξ_2 。在 x 方向上, 按 $F(x)$ 分布抽样:

$$\xi_x = \sqrt{3} \sin[2 \sin^{-1}\left(\frac{1}{\sqrt{3}}\right)(\xi_1 - \frac{1}{2})]$$

在 y 方向上, 按 $\frac{1}{F(\xi_x)}$ 的均匀分布抽样:

$$\xi_y = F(\xi_x)\xi_2$$

比较关系:

$$\begin{cases} \xi_y < p_2(\xi_x) & accept \\ \xi_y \geq p_2(\xi_x) & reject \end{cases}$$

3 程式说明

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N, double *x, int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。

本次实验中, 在生成 ξ_1, ξ_2 时, 为了保证独立性, 分别采用默认种子和时间种子。

- pdf_sampling_1.c

该程式产生按 $p_1(x)$ 分布进行抽样的一组指定个数(N)的随机数。

- pdf_sampling_2.c

该程式产生按 $p_2(x)$ 分布进行抽样的一组指定个数(N)的随机数。

- time_seed.txt

该文本文件显示的是调用时间种子时对应的原始数据。

- p_1(x).txt p_2(x).txt

分别记录根据 $p_1(x), p_2(x)$ 为密度函数生成的随机序列。

4 计算结果

下面根据数据使用Python程序做出频数分布直方图, 检验生成随机序列。预设 $bin = 500$, $p_1(x)$ 总点

数 $N = 10000$; $p_2(x)$ 未舍选之前总点数为 $N = 10000, N = 100000, N = 1000000$ 。

实际效率为: 0.809300, 0.811870, 0.812796。

4.1 直方图检验

可见, $p_1(x)$ 序列采用直接抽样法, 在点数较少的情况下收敛情况良好。而 $p_2(x)$ 序列由于采用的是舍选法, 需要更多的点数来得到良好收敛的概率密度函数。

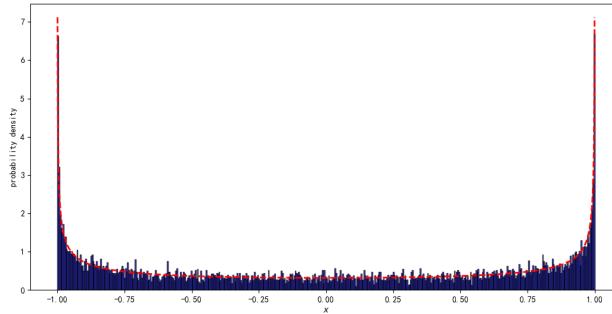
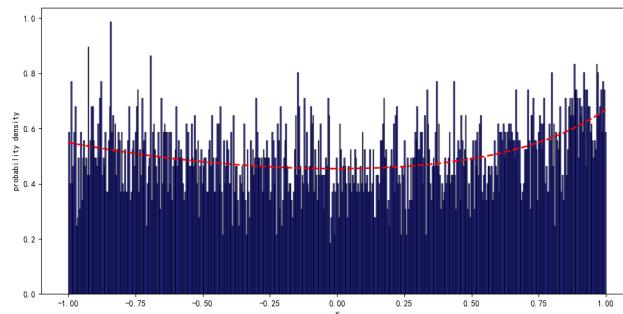
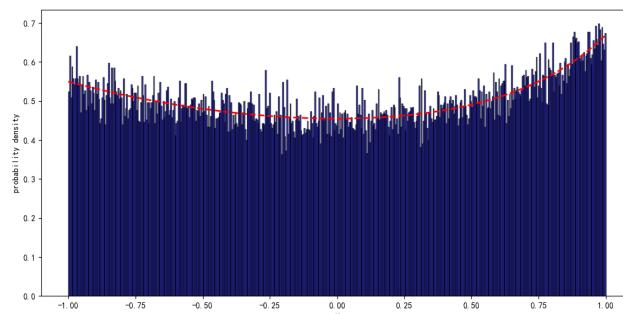
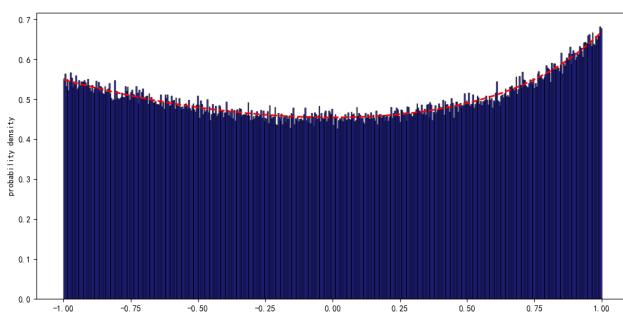
(a) $p_1(x)$ 序列, $N = 10000$ (b) $p_2(x)$ 序列, $N = 10000$ (c) $p_2(x)$ 序列, $N = 10000$ (d) $p_2(x)$ 序列, $N = 1000000$

图 4: 直方图检验(红色虚线为理想分布)

5 总结

- 当累计函数难以求得反函数时，舍选法是一个有效得到抽样的方法，但是其效率会降低，需要仔细地选择合适的比较函数。选择比较函数，一方面要尽可能和抽样函数形状，趋势一致，另一方面，还要容易求得反函数。本题中的比较函数 $F(x)$ 其实和 $p_1(x)$ 是同一类型函数，因而满足上述要求。
- 满足微分方程 $p(x) = \frac{dp(x)}{dx} \frac{x-d}{ax^2+bx+c}$ 的函数往往在某点会发散，因而只能将其定义在某个区间上，甚至密度函数出现不连续点。

计算物理A第五次作业

王铠泽 PB18020766

1 作业题目

- 对于球面上均匀分布的随机坐标点，给出它们在 (x, y) 平面上投影的几率分布函数。并由此验证Marsaglia抽样方法确为球面上均匀分布的随机抽样。

- Marsaglia抽样

$$\begin{cases} x = 2u\sqrt{1 - r^2} \\ y = 2v\sqrt{1 - r^2} \\ z = 1 - 2r^2 \end{cases}$$

其中, u, v 为 $[-1, 1]$ 上均匀分布随机数。

2 实现方法

- 变换抽样法

球坐标和直角坐标转化:

$$\begin{cases} x = \sin\theta \cos\phi \\ y = \sin\theta \sin\phi \\ z = \cos\theta \end{cases}$$

当球面上均匀分布 $f(\theta, \phi) = \frac{1}{4\pi} \sin\theta$ 要变换到二维 $x - y$ 上时:

$$f(\theta, \phi) d\theta d\phi = g(x, y) dx dy$$

从而有(考虑到两个上下半球在 $x - y$ 平面投影一致):

$$g(x, y) = \frac{1}{2\pi} \left| \frac{\partial(\theta, \phi)}{\partial(x, y)} \right| = \frac{1}{2\pi} \frac{1}{\sqrt{1 - s^2}}$$

其中 $s = \sqrt{x^2 + y^2}$

而对于 Marsaglia 抽样的变换, 有:

$$\left| \frac{\partial(x, y)}{\partial(u, v)} \right| = 4|1 - 2(u^2 + v^2)| = 4|1 - 2r^2| = 4\sqrt{1 - s^2}$$

但这里有个小细节, 当 $r^2 = u^2 + v^2 = \frac{1}{2}$ 时 Jacobi 行列式为 0, 有间断。而 $r^2 = \frac{1}{2}$ 刚好可以将圆盘分为等面积的两个部分, 将这两部分分开讨论, 将会导致最后的概率密度乘上因子 2。另一方面, 由于舍选导致 $p(u, v) = \frac{1}{\pi}$, 而不是 $\frac{1}{4}$ 。

$$\Rightarrow g'(x, y) = 2p(u, v) \cdot \frac{\partial(u, v)}{\partial(x, y)} = \frac{1}{2\pi} \frac{1}{\sqrt{1 - s^2}} = g(x, y)$$

这就验证了 $x - y$ 平面上的分布。

接下来验证 $x - z$ 平面分布($y - z$ 分布和这个类似, 不再证明)

$$f(\theta, \phi) = h(x, z) dx dz$$

$$\Rightarrow h(x, z) = \frac{1}{2\pi} \left| \frac{\partial(\theta, \phi)}{\partial(x, z)} \right| = \frac{1}{2\pi} \frac{1}{\sqrt{1 - z^2 - x^2}}$$

而*Marsalia*对应的密度函数 h' 满足：

$$h'(x, y) = 2p(u, v) \cdot \frac{\partial(u, v)}{\partial(x, z)} = \frac{1}{4\pi} \frac{1}{v\sqrt{1 - u^2 - v^2}}$$

另一方面：

$$\frac{1}{\sqrt{1 - z^2 - x^2}} = \frac{1}{2} \frac{1}{v\sqrt{1 - u^2 - v^2}}$$

$$\Rightarrow h'(x, y) = h(x, y)$$

至此，证毕。事实上 x, y, z 三个方面应当是等价的，所以 $x - y, y - z, z - x$ 平面上的分布应当是一样的。下面给出随机数在二维平面上的结果

3 程式说明

- Marsaglia.c

该程式使用*Marsaglia*抽样方法抽取三维球面上均匀分布随机数。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N, double *x, int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。

- dimensions.c

该程式是后续探究其他维度球面均匀分布时给出随机数的函数(二维, 四维)。

- time_seed0.txt

*Marsagli.c*抽样时对应的时间种子数据(1个种子)。

- time_seed.txt

*dimensions.c*抽样时对应的时间种子数据(4个种子)。

- sphere_surface.txt, ring.txt, 4-sphere.txt

这些都是数据文件。

4 计算结果

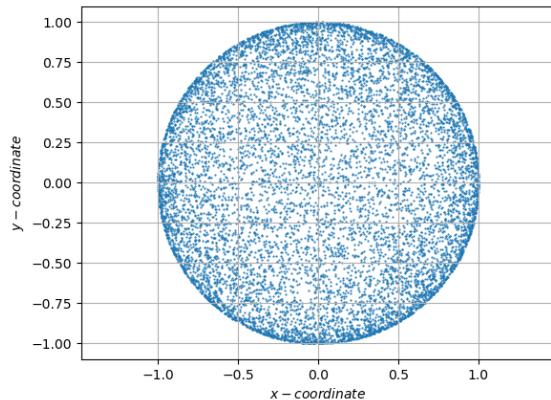
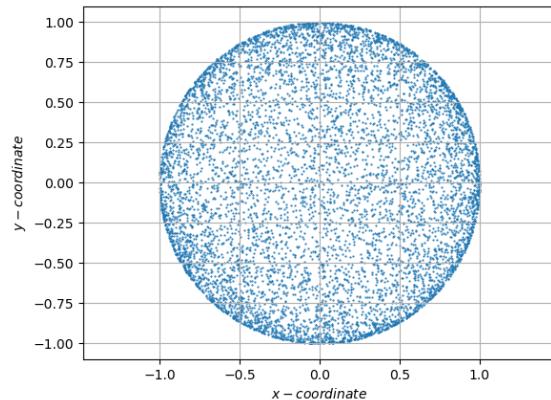
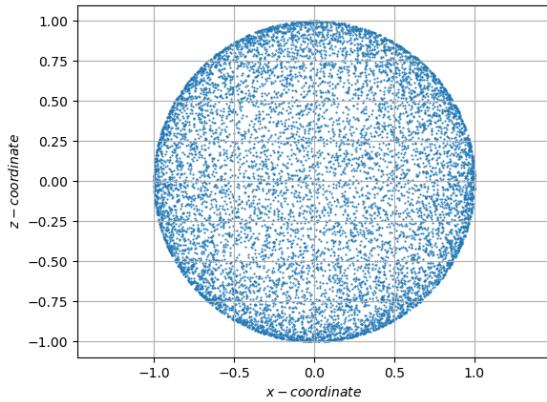
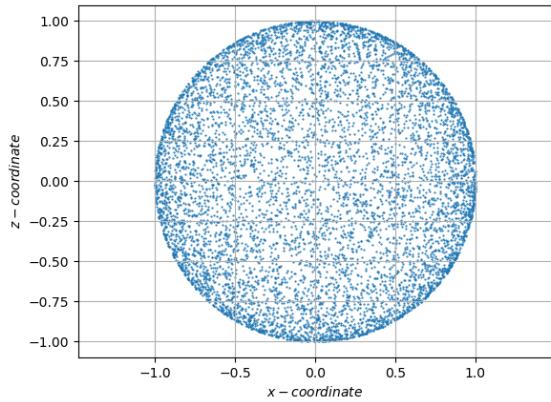
(a) 直接抽样 $x - y$ 平面分布(b) Marsaglia抽样 $x - y$ 平面分布(c) 直接抽样 $x - z$ 平面分布(d) Marsaglia抽样 $x - z$ 平面分布

图 1: 抽样分布

可以看出，球面上的均匀分布投影到二维平面上体现出外围密集(边缘发散)，中间稀疏的特征。这是和二维平面上有： $g(x, y) = \frac{1}{2\pi} \frac{1}{\sqrt{1-s^2}}$ 是吻合的

另一方面，这也直观上验证了Marsaglia抽样和直接抽样效果是一样的，只是牺牲了效率。

5 其他

可以考虑其他维度的*Marsaglia*抽样：

二维：

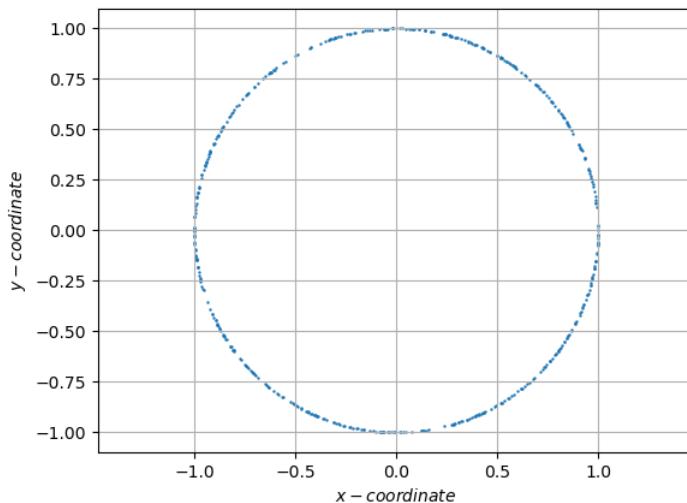


图 2: 二维球面均匀分布

和意料之中差别不大。

四维：

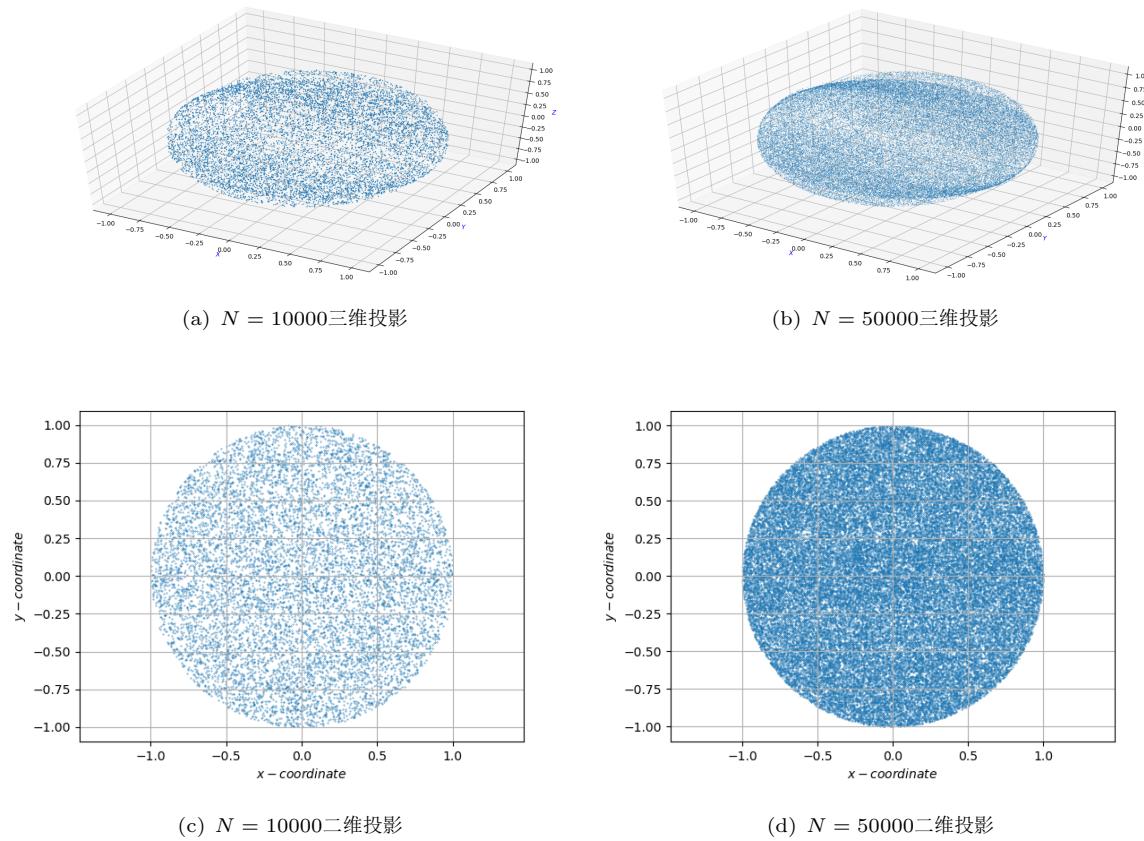


图 3: 四维投影分布

可见，三维投影更像是两个球体交叠在一起，在二维上投影均匀性比之前三维球面分布要高很多。

尽管使用 *Masarglia* 抽样可以不断得到更高维的球面上均匀分布，但是效率在逐步下降，直观上因为高维球体占据外接正方体体积比逐步下降，使得舍选效率降低。

<i>dimension</i>	<i>ratio</i>
$1d$	1.000000
$2d$	0.785398
$3d$	0.523599
$4d$	0.308425

表 1: 高维球体体积占据比

6 总结

- *Marsaglia*抽样法在难以通过直接抽样和变换抽样法时不失为一种有效的抽样方法。但是它的缺点也是所以舍选抽样具有的，效率问题。对于特别高维的球面均匀分布如果还使用*Marsaglia*抽样很多时候会抽到舍弃点，效率低。

计算物理A第六次作业

王铠泽 PB18020766

1 作业题目

- 对两个函数线型(*Gauss*分布和类*Lorentz*型分布), 设其一为 $p(x)$, 另一为 $F(x)$, 用舍选法对 $p(x)$ 抽样。将计算得到的归一化频数分布直方图与理论曲线 $p(x)$ 进行比较, 讨论差异。讨论抽样效率。

本实验中, 取 $p(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$, $F(x) = \frac{1}{\sqrt{\pi}} \frac{1}{x^4+1}$ 。理想效率 $\eta = 1/Area[F(x)] = \sqrt{\frac{2}{\pi}} \approx 0.797885$

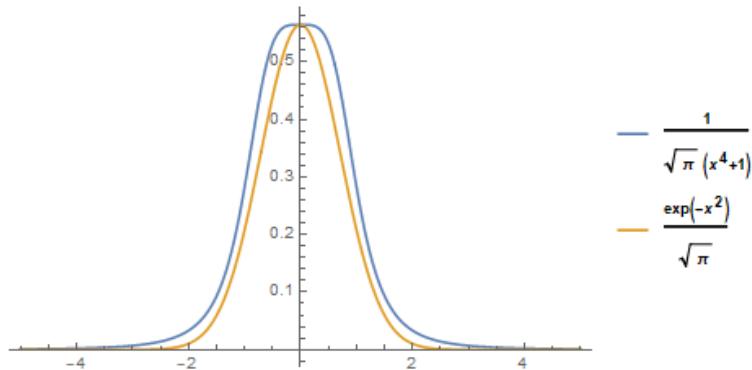


图 1: $p(x), F(x)$ 图例

2 实现方法

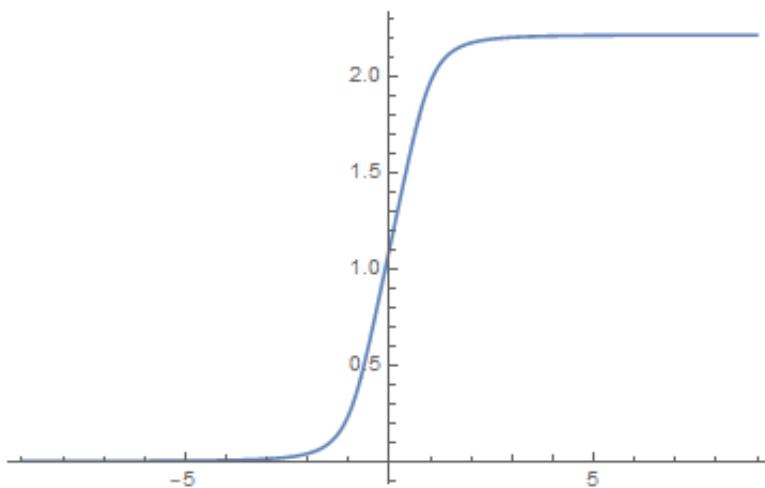
- 舍选法抽样

$F(x)$ 对应的归一化累计函数为:

$$H(x) = G(x) - G(-\infty)$$

其中 $G(x) = \frac{1}{4\pi} [2\tan^{-1}(1 + \sqrt{2}x) - 2\tan^{-1}(1 - \sqrt{2}x) + \ln(\frac{x^2 + \sqrt{2}x + 1}{x^2 - \sqrt{2}x + 1})]$, $\lim_{x \rightarrow \pm\infty} G(x) = \pm\frac{1}{2}$ 。

这个函数不易求出反函数, 本次实验中采用二分法求得对应 ξ_x 。选定的二分初始区间为 $[-10, 10]$ 。这个区间对应 $H(-10) = 0.00015$, $H(10) = 0.99985$, 覆盖99.97% $[0, 1]$ 上随机数。

图 2: $H(x)$ 图例

抽样方法: 生成两个 $[0, 1]$ 上随机序列 ξ_1, ξ_2 。在 x 方向上, 按 $F(x)$ 分布抽样:

$$\xi_1 = H(\xi_x) \Rightarrow \xi_x = H^{-1}(\xi_1)$$

在 y 方向上, 按 $\frac{1}{F(\xi_x)}$ 的均匀分布抽样:

$$\xi_y = F(\xi_x)\xi_2$$

比较关系:

$$\begin{cases} \xi_y < p_2(\xi_x) & \Rightarrow accept \\ \xi_y \geq p_2(\xi_x) & \Rightarrow reject \end{cases}$$

3 程式说明

- Rejection.c

该程式使用舍选抽样方法抽取正态分布 $\frac{1}{\sqrt{\pi}}e^{-x^2}$ 分布随机数。包含以下函数:

```
double p/F/H (double x)
```

这三个函数就是前述的 $p(x), F(x), H(x)$ 。

```
double root (double xi)
```

这个函数是通过二分法求解 $\xi = H(x)$ 对应的 x 。其中求解精度 ε 在函数中为 $erf = 10^{-4}$ 。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。

- time_seed.txt

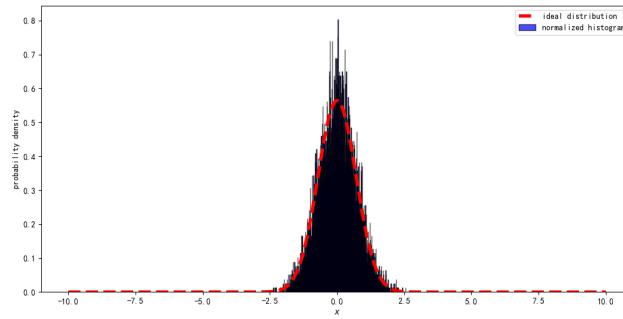
16807产生器抽样时对应的时间种子数据(1个种子)和最终舍选抽样效率(手动加的)。

- N=10000/100000/1000000.txt

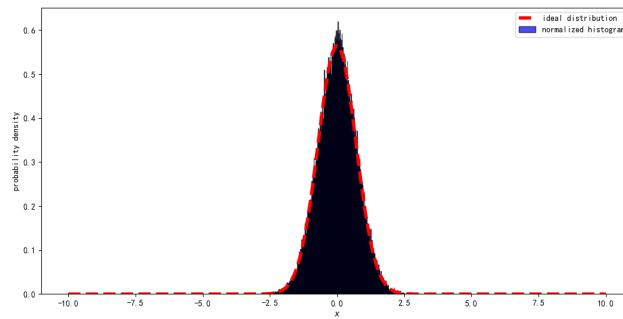
记录了生成的数据文件。

4 计算结果

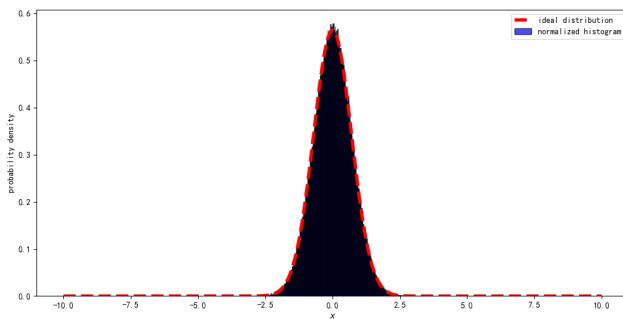
分别在总点数 $N = 10000, 100000, 1000000$ 下得到归一化直方图。默认参数 $bin = 500$, 红色虚线为理想分布。



(a) $N = 10000, \eta = 0.793200$



(b) $N = 100000, \eta = 0.798740$



(c) $N = 1000000, \eta = 0.798477$

图 3: 归一化直方图

可见，随着 N 增大，频数分布逐渐趋于理想分布。

5 总结

- 本次实验采用舍选法，利用和Guass函数形状类似的Lorentz型函数作为比较函数抽样，提供了又一种正态抽样的方法。
- 为了提高运算速度，应该考虑如何对 $F(x)$ 进行抽样，寻找比二分法求逆更有效率方式。

计算物理A第七次作业

王铠泽 PB18020766

1 作业题目

- 对一个实验谱数值曲线 $p(x)$ ，自设 $F(x)$ ，分别用直接抽样和舍选法对 $p(x)$ 抽样。比较原曲线和抽样得到的曲线以验证。讨论抽样效率

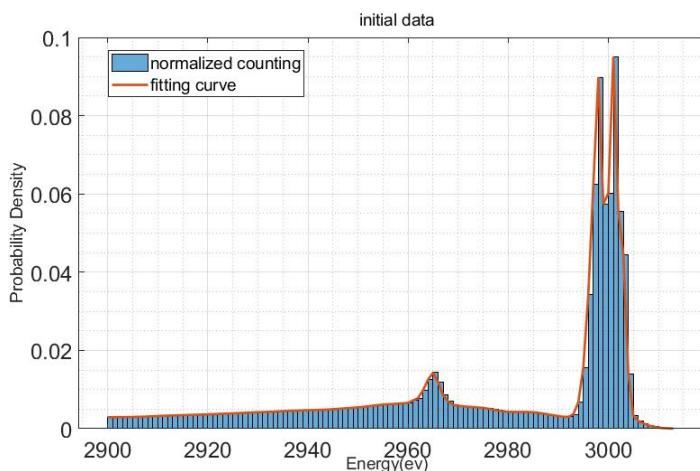


图 1: 数据直方图

2 实现方法

- 直接抽样

本实验中直接抽样采用离散化的分布模型(若要得到连续性分布可以将 $p(x)$ 进行线性插值得到相应的累计函数 $P(X)$,再通过二分法求逆)。 $p(k) = \frac{N_k}{\sum N_k}$
设 ξ 为 $[0,1]$ 上均匀抽样。若满足 $P(k-1) < \xi \leq P(k)$ ，则取 $E = k$ 作为抽样。

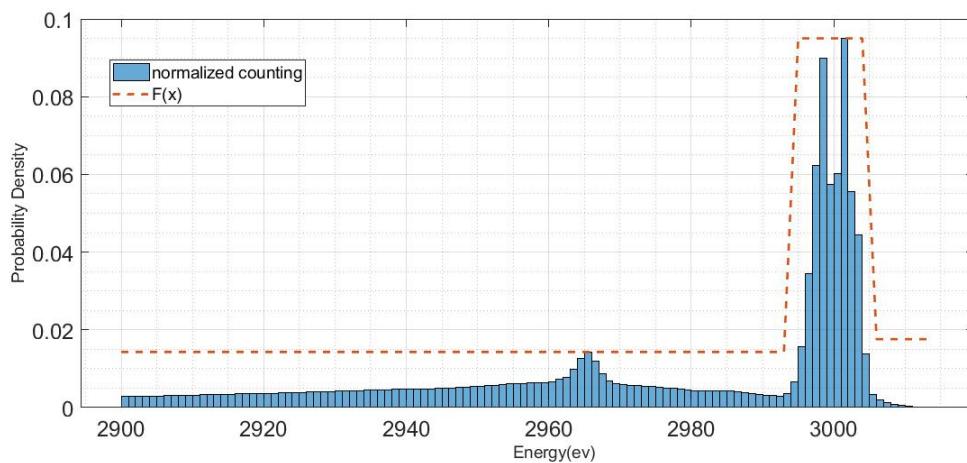
其中 $P(k) = \sum_{i=2899}^k p(i)$, $P(2899) = 0$, $P(3013) = 1$ 。

- 舍选法抽样

采用分段式的阶梯函数作为 $F(x)$ 。

$$\begin{cases} 0.0143 & 2900 \leq x < 2994 \\ 0.0950 & 2994 \leq x < 3005 \\ 0.0033 & 3005 \leq x \leq 3013 \end{cases}$$

示意图如下:

图 2: 阶跃型 $F(x)$

其效率为 $I = \frac{1}{\int_{2900}^{3013} F(x) dx} \approx 0.413976$ 。 $F(x)$ 归一化对应的累计函数为:

$$H(x) = \begin{cases} 0.00591986x - 17.1676 & 2900 \leq x < 2994 \\ 0.0393277x - 117.191 & 2994 \leq x < 3005 \\ 0.00136612x - 3.11612 & 3005 \leq x \leq 3013 \end{cases}$$

抽样方法: 生成两个 $[0, 1]$ 上随机序列 ξ_1, ξ_2 。在 x 方向上, 按 $F(x)$ 分布抽样:

$$\xi_x = H^{-1}(\xi_1) = \begin{cases} 168.923\xi_1 + 2900.00 & 0 \leq \xi_1 < 0.556466 \\ 25.4274\xi_1 + 2979.86 & 0.556466 \leq \xi_1 < 0.988799 \\ 732.00\xi_1 + 2281.00 & 0.988799 \leq \xi_1 \leq 1 \end{cases}$$

在 y 方向上, 按 $\frac{1}{F(\xi_x)}$ 的均匀分布抽样:

$$\xi_y = F(\xi_x)\xi_2$$

比较关系:

$$\begin{cases} \xi_y < p(\xi_x) & \text{accept} \\ \xi_y \geq p(\xi_x) & \text{reject} \end{cases}$$

在最后一步比较大小时, 我们采用了线性插值的办法: 对于 $\text{floor}(\xi_x) \leq \xi_x \leq \text{ceil}(\xi_x)$, 估计值为:

$$p(\xi_x) = p[\text{floor}(\xi_x)] \cdot \frac{\xi_x - \text{ceil}(\xi_x)}{\text{floor}(\xi_x) - \text{ceil}(\xi_x)} + p[\text{ceil}(\xi_x)] \cdot \frac{\xi_x - \text{floor}(\xi_x)}{\text{ceil}(\xi_x) - \text{floor}(\xi_x)}$$

其中 $\text{floor}, \text{ceil}$ 分别为向下, 向上取整函数。

3 程式说明

- sampling.c

该程式包含两大部分: 1. 使用直接抽样法抽取曲线对应数据; 2. 采用舍选抽样法抽取曲线对应数据。

包含以下函数:

double F (double x)

这个函数就是前述的 $F(x)$ 。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。程式中故意采用`sleep`函数就是为了得到不同的时间种子。

- time_seed.txt

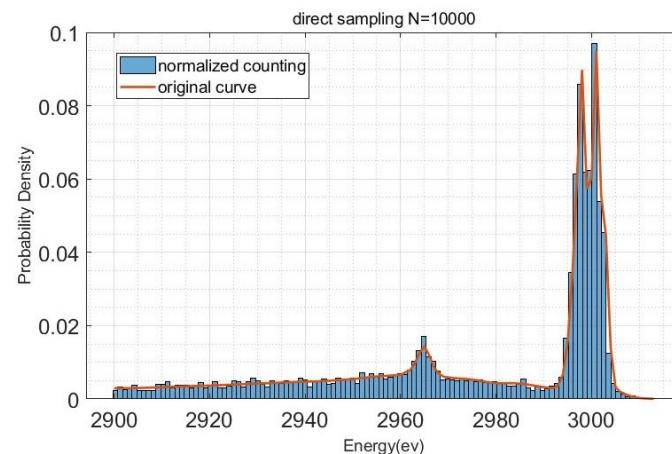
16807产生器抽样时对应的时间种子数据(1个种子)和对应抽样方法(后者是手动加入)。

- direct_sampling/reject_sampling.txt

记录了生成的随机数数据文件。

4 计算结果

在总点数为10000的条件下得到如下分布直方图。其中红色曲线是依据原始数据拟合而得。



(a) 直接抽样

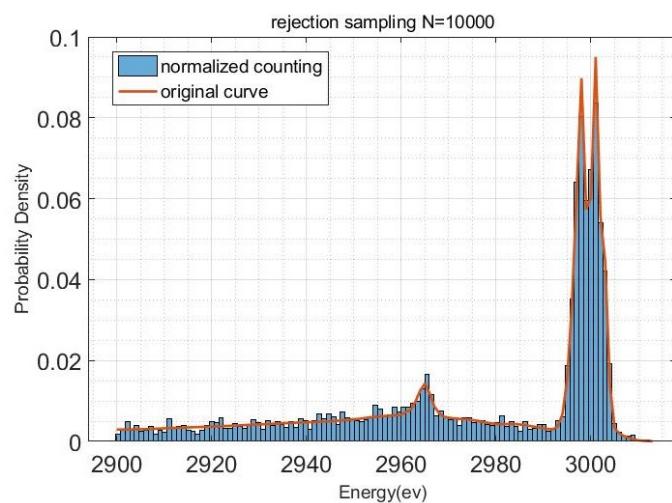
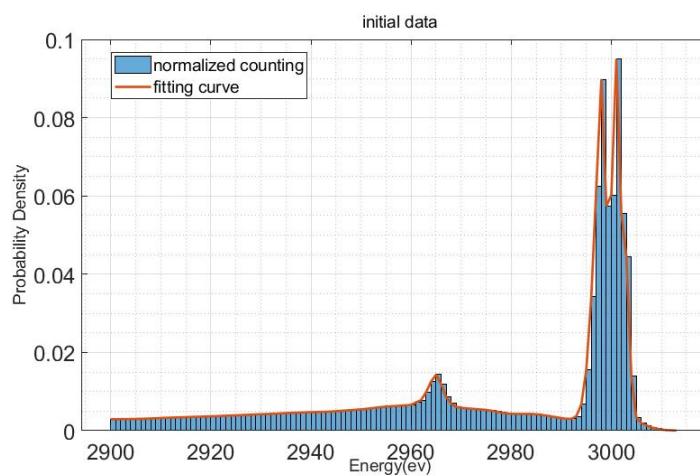
(b) 舍选抽样, $\eta = 0.4210$ 图 3: $N = 10000$ 时归一化直方图分布

图 4: 原图

可见两者都和实验曲线有良好拟合。

5 总结

- 本次实验分别采用直接抽样法和舍选抽样法对实验数据曲线进行抽样。两种抽样的效果差不多，舍选抽样在 $3000ev$ 左右峰处拟合更为理想，其原因可能是在舍选法中做了线性插值而直接抽样是直接视为离散分布。在峰处变化剧烈，采用线性插值能减小误差。
- 直接抽样法的抽样效率为%100，而舍选抽样在本实验中约为40%，明显偏低。为了提高抽样效率，可能更加精细的分出更多的阶梯。但这样做实在是大大增加了在代码层面的工作量，总效率也不见得提升。
- 实际应用中，应该根据数据的个数，数据分布情况，要求精度等来进行抽样方法选择。

计算物理A第八次作业

王铠泽 PB18020766

1 作业题目

- 用Monte Carlo方法计算如下定积分，并讨论有效数字位数。

$$I_1 = \int_0^2 dx \sqrt{x + \sqrt{x}}$$

$$I_2 = \int_0^{\frac{9}{10}} dx \int_0^{\frac{4}{5}} dy \int_0^{\frac{9}{10}} dz \int_0^2 du \int_0^{\frac{13}{10}} dv (6 - x^2 - y^2 - u^2 - v^2)$$

2 实现方法和原理

- Monte Carlo简单抽样求积分

单变量的情况下求 $\int_a^b f(x)dx$ ，可以按照以下方法：设 ξ 为 $[0, 1]$ 上均匀随机数。则 $\tilde{\xi} = (b-a)\xi + b$ 为 $[a, b]$ 上的均匀随机数。抽取 $\tilde{\xi}$ 序列(总点数为 N)，则 $\langle f \rangle = \frac{1}{N} \sum_{k=1}^N f(\tilde{\xi})$ 。

$$\Rightarrow \int_a^b f(x)dx = (b-a)\langle f \rangle = \frac{b-a}{N} \sum_{k=1}^N f(\tilde{\xi})$$

这个结论可以推广到多变量情形：

$$\int_R f(\vec{r})d\vec{r} = \langle f \rangle \cdot V(R)$$

其中 R 代表任意维度的矩形区域 $[a_1, b_1] \times [a_2, b_2] \times \dots$ ， $V(R)$ 表示矩形区域的体积， $\langle f \rangle = \frac{1}{N} \sum_{k=1}^N f(\tilde{\xi})$, $\tilde{\xi}$ 为 R 上均匀分布的随机变量。

- 大数定律和中心极限定理

假设 X_1, \dots, X_N 为服从同一分布的随机变量序列。设其期望为 μ ，标准差为 σ , $\langle X \rangle = \frac{1}{N} \sum_{k=1}^N X_k$ 。大数定律指出：

$$\frac{1}{N}(X_1 + \dots + X_N) \xrightarrow{N \rightarrow \infty} \mu$$

中心极限定理：

$$P\left(\frac{\langle X \rangle - \mu}{\sigma/\sqrt{N}} < x\right) \xrightarrow{N \rightarrow \infty} \Phi(x)$$

其中 $\Phi(x)$ 为标准正态分布 $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ 。

积分结果 $I = V(R)\langle f \rangle$

积分结果方差 $var(I) = V^2(R)var(\langle f \rangle)$

$$\text{积分结果标准差 } \sigma(I) = V(R)\sqrt{\langle f \rangle} = V(R)\frac{\sigma(f)}{\sqrt{N}}$$

当 N 充分大时，结果的误差是正比于 \sqrt{N} 的。

3 程式说明

- single.c

该程式计算第一个单变量积分。

- multi.c

该程式计算第二个多变量积分。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的[0, 1]上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针x对应的长度为N的数组用[0, 1]上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。程式中故意采用sleep函数就是为了得到不同的时间种子。

- time_seed_single/multi.txt

16807产生器抽样时对应的时间种子数据(1个种子)。调用多少次16807生成器就生成多少个数据记录。

- single_variable/multi_variable.txt

详细记录了不同N下的积分值I以及其和准确值之间的误差Err。

4 计算结果

4.1 单变量积分

对于积分:

$$I_1 = \int_0^2 dx \sqrt{x + \sqrt{x}}$$

使用Mathematica得到其准确值约为 $I \approx 2.689521304816752$ 。不同N下对应的情况如下表所示。

N	<i>Integral</i>	<i>Error</i>	$\frac{1}{\sqrt{N}}$
10	2.29035867e+000	3.99162639e-001	3.16227766e-001
10^2	2.64702903e+000	4.24922760e-002	1.00000000e-001
10^3	2.71176247e+000	2.22411622e-002	3.16227766e-002
10^4	2.67987859e+000	9.64271888e-003	1.00000000e-002
10^5	2.68859115e+000	9.30158296e-004	3.16227766e-003
10^6	2.68942442e+000	9.68839664e-005	1.00000000e-003
10^7	2.68944262e+000	7.86839022e-005	3.16227766e-004
10^8	2.68942564e+000	9.56666747e-005	1.00000000e-004

表 1: 不同N下单变量积分的情况(采用科学计数法)

可见, 当N越来越大时, 标准差和 $\frac{1}{\sqrt{N}}$ 逐渐接近。

4.2 多变量积分

对于积分:

$$I_2 = \int_0^{\frac{9}{10}} dx \int_0^{\frac{4}{5}} dy \int_0^{\frac{9}{10}} dz \int_0^2 du \int_0^{\frac{13}{10}} dv (6 - x^2 - y^2 - u^2 - v^2)$$

使用Mathematica得到其准确值约为 $I \approx 5.644079999999997$ 。不同 N 下对应的情况如下表所示。

N	<i>Integral</i>	<i>Error</i>	$\frac{1}{\sqrt{N}}$
10	6.17363991e+000	5.29559909e-001	3.16227766e-001
10^2	5.14681851e+000	4.97261489e-001	1.00000000e-001
10^3	5.69498165e+000	5.09016489e-002	3.16227766e-002
10^4	5.67040355e+000	2.63235535e-002	1.00000000e-002
10^5	5.64101095e+000	3.06904571e-003	3.16227766e-003
10^6	5.64586026e+000	1.78026028e-003	1.00000000e-003
10^7	5.64415403e+000	7.40250527e-005	3.16227766e-004

表 2: 不同 N 下多变量积分的情况(采用科学计数法)

可见, 当 N 越来越大时, 标准差和 $\frac{1}{\sqrt{N}}$ 逐渐接近。

4.3 误差分析

对单变量情况, 做出对数 $Error - 1/\sqrt{N}$ 拟合曲线如下:

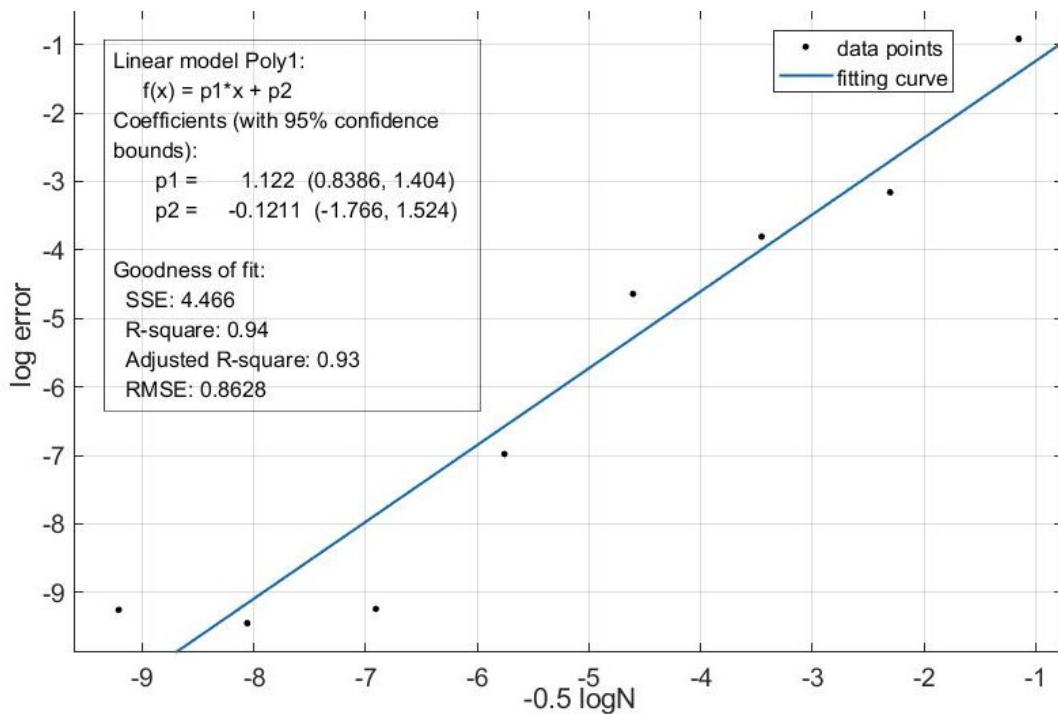


图 1: $\log(\epsilon) - \log(\frac{1}{\sqrt{N}})$

对多变量情况，做出对数 $Error - 1/\sqrt{N}$ 拟合曲线如下：

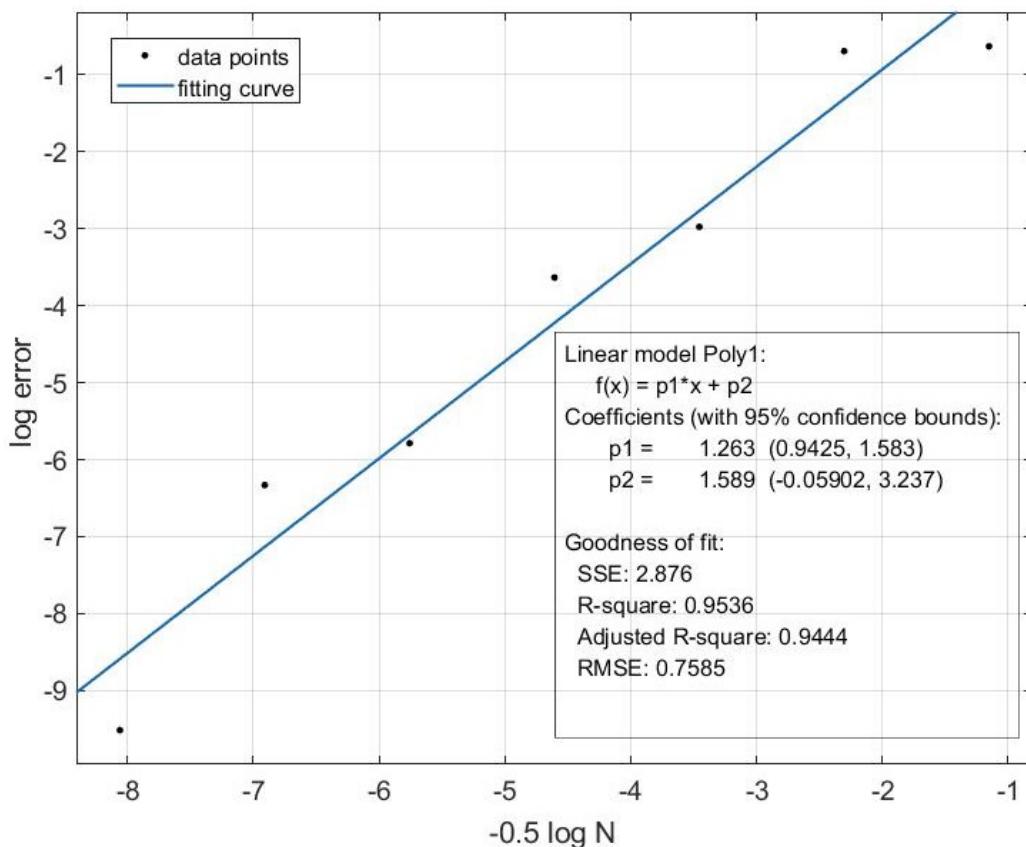


图 2: $\log(\epsilon) - \log(\frac{1}{\sqrt{N}})$

拟合结果可以看出误差 $\epsilon \sim 1/\sqrt{N}$ ，线性项系数都是比较接近1的

5 总结

- 本次实验使用*Monte Carlo*方法求简单区间上的积分值，在 N 较大的时候很接近准确值。
- 随着对精确度 ϵ 要求提高，对抽样点数 N 的增长大约是小数点后每精确一位就需要增长100倍，高精度计算只用简单抽样的*Monte Carlo*方法显然是不实际的，需要其他的优化。
- 由于点数有限和生成随机数的初始种子，生成方式等影响，误差并不严格满足中心极限定理语言的 $\frac{1}{\sqrt{N}}$ ，有时甚至会略小(见前述的单变量实验表格)。

计算物理A第九次作业

王铠泽 PB18020766

1 作业题目

- 自设若干个随机分布（相同或不同分布，它们有相同或不同的 μ 和 σ 通过 Monte Carlo 模拟，验证中心极限定理成立($N=2,5,10$)。

本次实验通过若干个相同分布的离散、连续随机变量序列来验证中心极限定理。

2 实现方法和原理

- 大数定律和中心极限定理

假设 X_1, \dots, X_N 为服从同一分布的随机变量序列。设其期望为 μ ，标准差为 σ ， $\langle X \rangle = \frac{1}{N} \sum_{k=1}^N X_k$ 。
大数定律指出：

$$\frac{1}{N} (X_1 + \dots + X_N) \xrightarrow{N \rightarrow \infty} \mu$$

中心极限定理：

$$P\left(\frac{\langle X \rangle - \mu}{\sigma/\sqrt{N}} < x\right) \xrightarrow{N \rightarrow \infty} \Phi(x)$$

其中 $\Phi(x)$ 为标准正态概率分布函数：

$$\int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} d\xi$$

- 本实验采用三组不同的随机序列来验证中心极限定理。

连续型分布：指数分布

$$\begin{aligned} p(x) &= \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) \cdot I(x \leq 0) \\ P(x) &= [1 - \exp\left(-\frac{x}{\lambda}\right)] \cdot I(x \leq 0) \\ \mu &= \lambda, \sigma = \lambda \end{aligned}$$

连续型分布：作业四中的自设分布

$$\begin{aligned} p(x) &= 1.81915 \frac{e^{-x}}{(x-2)^2} \cdot I(-1 \leq x \leq 1) \\ \mu &\approx 0.021, \sigma \approx 0.60 \end{aligned}$$

其中 $I(E)$ 为示性函数，表示对 x 取值的限制。其取值为：

$$I(E) = \begin{cases} 1 & \text{event } P \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

后面将自动略去示性函数

离散型分布:*Poisson* 分布

$$p(x = n) = \frac{\lambda^n}{n!} \exp(-\lambda) \quad (n = 0, 1, 2\dots)$$

$$P(n) = \sum_{k=0}^n \frac{\lambda^k}{k!} \exp(-\lambda) \quad (k = 0, 1, 2\dots)$$

$$\mu = \lambda, \sigma = \sqrt{\lambda}$$

- 抽样方法

直接抽样

对于指数分布和*Poisson*分布，由于反函数简单采用直接抽样法。

指数分布($\lambda = 1$):

$$x = \ln \xi$$

*Poisson*分布:

$$P(n) < \xi \leq P(n+1) \Rightarrow x = n$$

其中 ξ 是[0, 1]上的随机数。

舍选抽样法

对于自设分布，其反函数比较复杂，不易求逆，采用舍选抽样。其比较函数如下：

采用 $F(x) = \frac{1}{\sqrt{3-x^2}}$ 作为覆盖 $p(x)$ 的比较函数。 $I = \int_{-1}^1 F(x) dx = 2 \sin^{-1}(\frac{1}{\sqrt{3}})$ 。该舍选效率为：

$$\text{Area}[p(x)]/\text{Area}[F(x)] \approx 0.812374$$

更详细的说明参见压缩包中的文件*report04.pdf*。

3 程式说明

定义

$$Y = \frac{\langle X \rangle - \mu}{\sigma / \sqrt{N}}$$

其中 $\langle x \rangle = \frac{1}{N} \sum_{i=1}^N X_i$

- exp.c

该程式生成对应着指数分布时 $N = 2, 5, 10$ 的 Y 的分布， Y 的抽样总点数记为 M ,取 $M = 10^4$ 。

- self.c

该程式生成对应着自设分布时 $N = 2, 5, 10$ 的 Y 的分布， Y 的抽样总点数记为 M ,取 $M = 10^4$ 。

- poisson.c

该程式生成对应着*Poisson*分布时 $N = 2, 5, 10$ 的 Y 的分布， $\langle X \rangle$ 的抽样总点数记为 M ,取 $M = 10^4$ 。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。程式中故意采用sleep函数就是为了得到不同的时间种子。

- time_seed.txt

16807产生器抽样时对应的时间种子数据(每次1个种子)。调用多少次16807生成器就生成多少个数据记录。每一个分布对应的种子已经手动加上分布的对应了。

- p_self.txt

这是一类中间数据文件。在实现对自设函数抽样时，先把按自设密度函数 $p(x)$ 抽取的随机数存入此文件，后续计算 $\langle x \rangle$ 的时候再从此文件读入随机序列。

- exp/self/poisson(N=2/5/10/100/1000/10000).txt

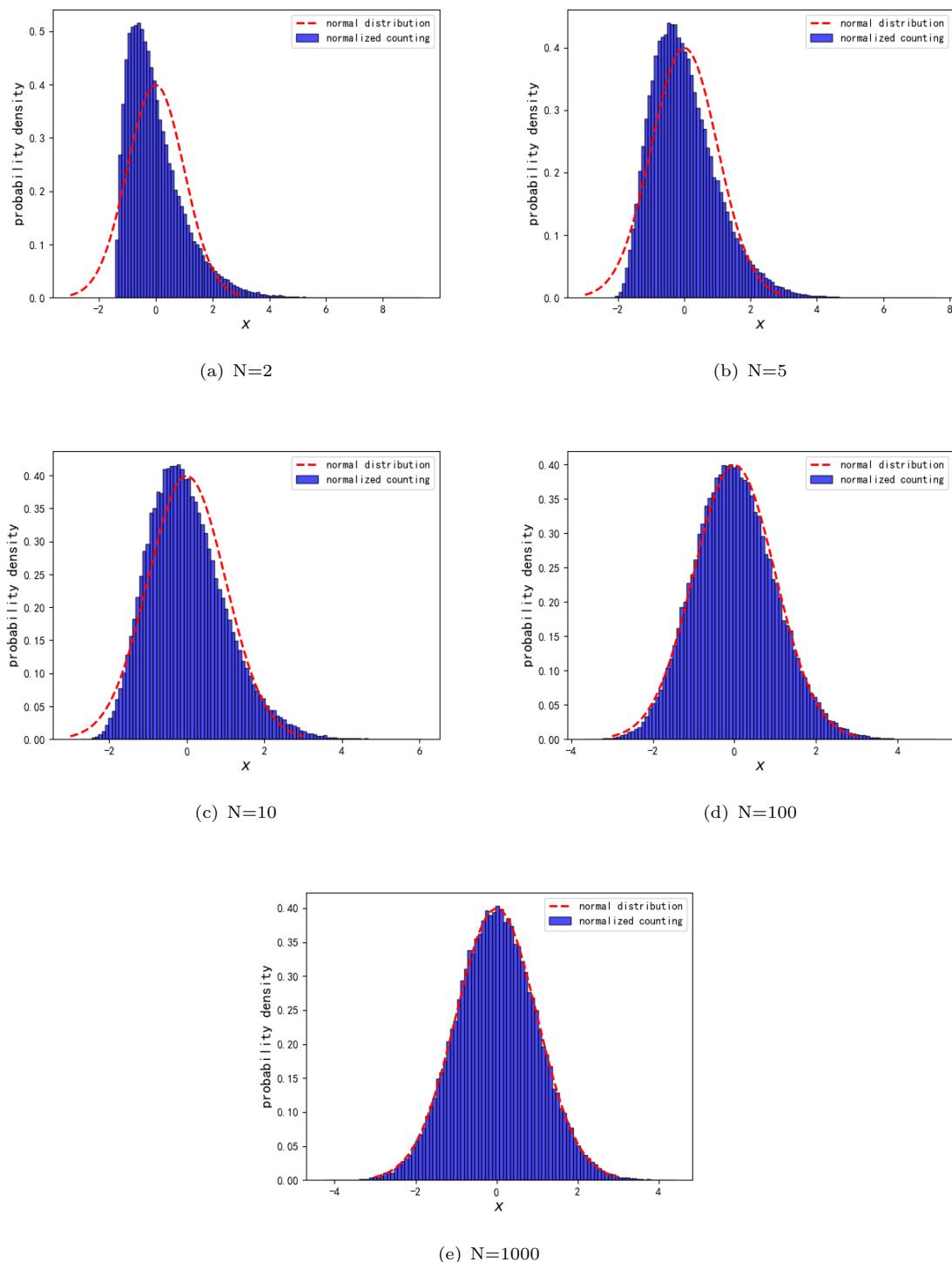
不同分布下不同 N 值对应的数据文件。

4 计算结果

以下计算通过计算 $Y = \frac{\langle X \rangle - \mu}{\sigma / \sqrt{N}}$, 对 Y 抽样点数设为 $M = 10000$ 。下面红线是标准正态分布曲线。

4.1 指数分布

$$p(x) = e^{-x} \quad (x > 0)$$

图 1: 不同 N 下的 Y 分布情况

可见，当 $N = 100$ 时已经和正态分布曲线非常吻合了。

4.2 Poisson分布

$$p(x = k) = \frac{1}{k!} e^{-1}$$

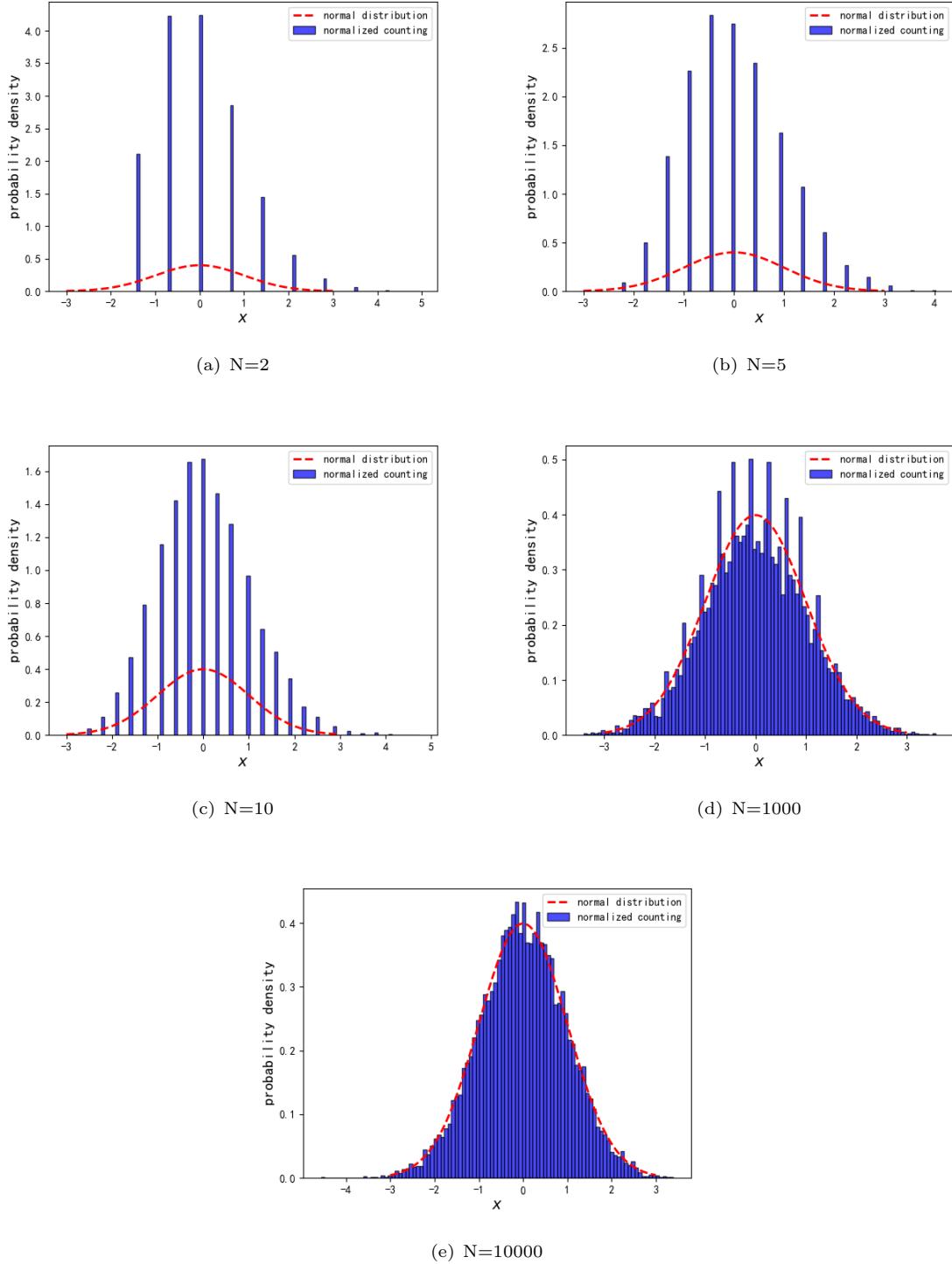


图 2: 不同 N 下的 Y 分布情况

对于离散型的 $poisson$ 分布，可见 Y 收敛到正态分布的速度要比连续性的指数分布慢得多，直到 $N = 10000$ 时在中心处仍有少许“毛刺”。

4.3 自设分布

$$p(x) = 1.81915 \frac{e^{-x}}{(x-2)^2} \quad (-1 \leq x \leq 1)$$

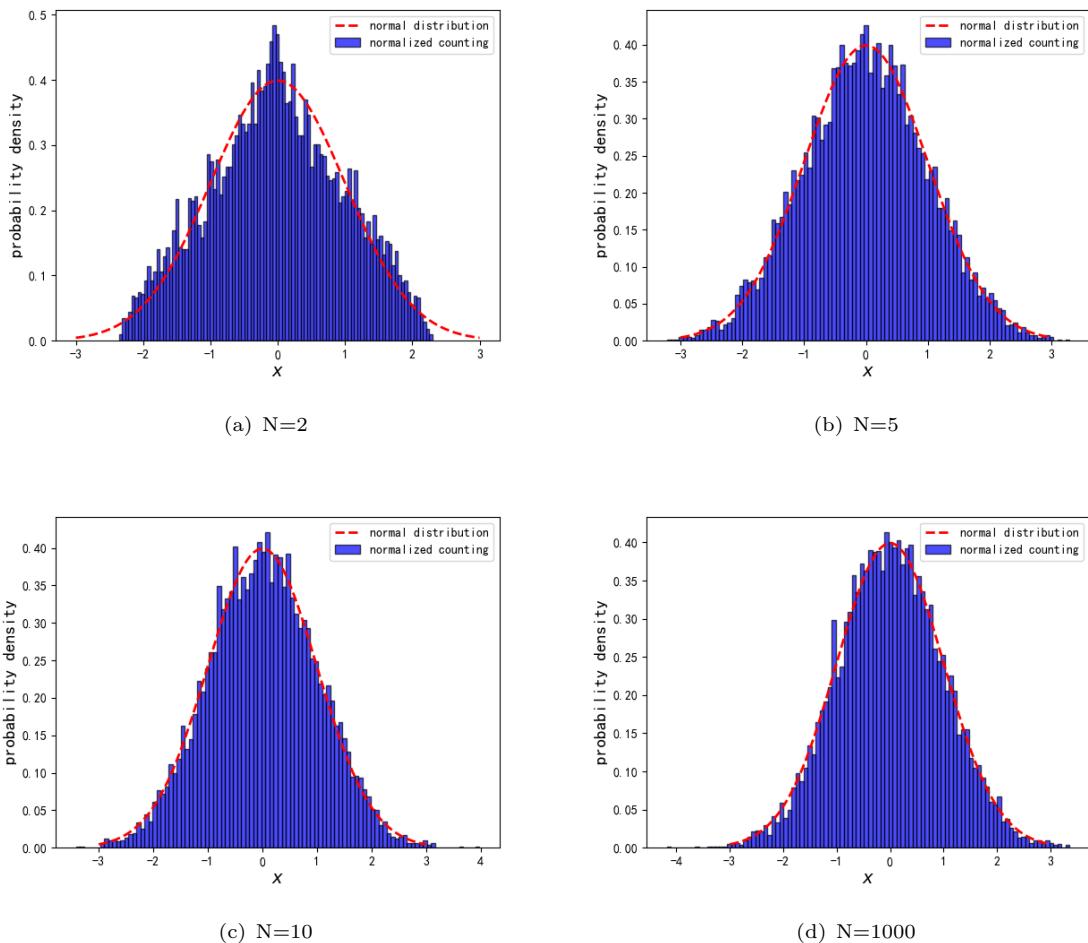


图 3: 不同 N 下面的 Y 分布情况

对于这个自设分布，当 $N = 1000$ 时有比较好的拟合效果，但是相比指数分布的还是收敛略慢。不过相比之下，当 $N = 5$ 时也已经比较靠近正态分布函数了，其原因可能是本来的分布函数在 $(-1, 1)$ 上的对称性比指数分布的要高。

5 总结

- 本次实验通过不同分布下的随机序列验证了中心极限定理成立。
- 中心极限定理给出了计算物理中估计误差非常重要的一个公式：

$$\langle x \rangle \sim N\left(\mu, \frac{\sigma_x}{\sqrt{N}}\right)$$

这表示当 N 足够大时，误差 $\sigma \sim \sigma_x / \sqrt{N}$

计算物理A第十次作业

王铠泽 PB18020766

1 作业题目

- 研究有取向的布朗粒子(如纳米棒)的随机行走，计算取向的自关联函数：

$$C(t) = \langle u_x(t)u_x(0) \rangle$$

其中 u_x 为取向单位向量在 x 轴上的投影。

2 实现方法和原理

- 模型建立

为了简单起见，我们考虑一个形状为椭圆的二维纳米棒的随机游走，并假设其长轴长度为 a ，短轴长度为 b 。以椭圆长轴为 x' 轴，椭圆短轴为 y' 轴，原点在质心的惯性系作为我们讨论的出发点，通过坐标变换可以得到原实验室系的坐标 x, y 。写出Langevin方程：

$$\begin{cases} \ddot{x}' = -\frac{1}{B_1} \dot{x}' + f_{x'} \\ \ddot{y}' = -\frac{1}{B_2} \dot{y}' + f_{y'} \\ \ddot{\theta} = -\frac{1}{C} \dot{\theta} + \tau \end{cases}$$

在这个最简单的假设下， x', y', θ 之间是完全没有任何耦合的。因为 $a > b$ ，可以认为 $B_1 < B_2$ 。本文再加上一条假设： θ 方向转动远远慢于平动, $C \ll a \cdot B_1 / B_2$ 。[1]

最后通过坐标变换到实验室系：

$$\begin{cases} dx = \cos\theta dx' - \sin\theta dy' \\ dy = \sin\theta dx' + \cos\theta dy' \end{cases}$$

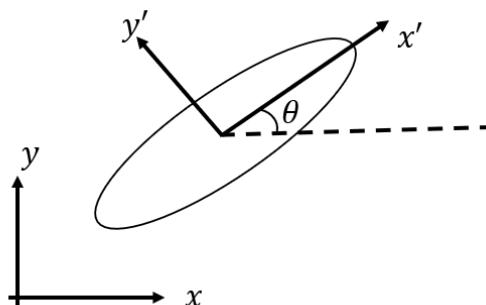


图 1: 模型示意图

由于三个方向解耦，可以算出各物理量二阶矩如下：

$$\langle x'^2 \rangle = 2D_x t$$

$$\langle y'^2 \rangle = 2D_y t$$

$$\langle \theta^2 \rangle = 2D_\theta t$$

其中 $D_{x'} = kTB_1$, $D_{y'} = kTB_2$, $D_\theta = kTC$

这是由 $x(t)$ 服从 $N(x_0, \sqrt{2D_x t})$ 得到的。在下面的计算以及讨论中，为了方便起见，令 D_1, D_2, D_t 为三个解耦变量的正态分布的标准差。

- 理论结果

对于自关联投影 $C(t)$ 的计算：

$\langle u(t)u(0) \rangle = \langle \cos(\theta)\cos(\theta_0) \rangle$, 系综平均理解为对初始不同角度 θ_0 和对大量随机运动的二重平均。而初始角度为 θ_0 的纳米棒在 t 时间后角度的分布应该是 $N(\theta_0, D_t^2 t)$ 。

$$\Rightarrow \langle \cos(\theta)\cos(\theta_0) \rangle = \int_0^{2\pi} d\theta_0 \int_{-\infty}^{\infty} d\theta \frac{1}{\sqrt{2\pi D_t t}} \cos(\theta)\cos(\theta + \theta_0) \exp\left[-\frac{\theta^2}{2D_t^2 t}\right] = \frac{1}{2} \exp\left(-\frac{D_t^2 t}{2}\right)$$

在关于 x, y 的讨论上，虽然在 $x' - y'$ 系中 x', y' 是可以视为互相独立的两个不同扩散系数的布朗运动，但由于要转换到实验室系中，出现了和角度之间的耦合，这时它们的分布就不再是简单的正态分布。一个较为量化地描述和正态分布偏移的方法是计算位移的四阶累积量(cumulant)**ref1**：

$$C_{\theta_0}^{(4)}(t) = \langle [x(t)]^4 \rangle_{\theta_0} - 3\langle [x(t)]^2 \rangle_{\theta_0}^2$$

正态分布的四阶累积量为零。

无量纲归一化后的指标是：

$$p(t, \theta_0) = \frac{C_{\theta_0}^{(4)}(t)}{3\langle [x(t)]^2 \rangle_{\theta_0}^2}$$

- 计算实现方法

计算自关联函数

进行两层循环：外层对系综进行循环(用 i 标记系统数)，内层对时间进行循环(用 j 标记步长)。每次进行一个系统的循环时，初始化初始的夹角 $\theta(0) = \theta_0$ ，为 $[0, 2\pi]$ 上的均匀分布。使用数组 T 来记录每一个时间下对应的 $C(t)$ 的数值，并写入文件。

计算扩散系数 D_x, D_y

基本思路和计算自关联函数差不多，只是每对一个系统进行计算时， θ 不需要随机初始化， $\theta(0) = 0$ 。注意对 x', y' 叠加时，每一步都是在原基础上叠加一个服从正态分布 $N(0, D_{1/2}^2)$ 的长度。最后用数组 Dx, Dy 记录数据，并写入文件。

计算四阶累积量

和上述思路差不多，仅是统计量换成 x^2, x^4 ，进行记录。

3 程式说明

- rw1.c

该程式计算在解耦情况下自相关函数 $u(t)$ 的情况。

- rw2.c

该程式计算在实验室系中扩散系数 $D_x = \langle x^2(t) \rangle / t$ 和 $D_y = \langle y^2(t) \rangle / t$ 的数据。

- cumulant.c

该程式计算在该模型下的四阶统计量 $C(t)$ 的数据。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的[0, 1]上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用[0, 1]上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。程式中故意采用sleep函数就是为了得到不同的时间种子。

- time_seed.txt

16807产生器抽样时对应的时间种子数据(每次1个种子)。调用多少次16807生成器就生成多少个数据记录。每一个分布对应的种子已经手动加上对应的实验了。种子产生公式如下:

年: $0 \leq i_y \leq 99$, 月: $1 \leq i_m \leq 12$, 日: $1 \leq i_d \leq 31$
 时: $0 \leq i_h \leq 23$, 分: $0 \leq i_n \leq 59$, 秒: $0 \leq i_s \leq 59$
 则可设种子值为: $I_0 = i_y + 70(i_m + 12(i_d + 31(i_h + 23(i_n + 59i_s))))$, 它的值约在区间
 $[0, 2^{31}-1]$ 内, 第二部分的括号在100年内不会重复。

Tip:程序中多处使用了sleep函数是为了换时间种子, 因而可能运行时间较久一点。

- ensemble.txt

这是取完系综平均的 D_x, D_y 数据文件, 后面括号内为详细的实验条件。

- one_ellipsoid.txt

这是针对一个特殊的纳米棒的随机游走记录下其位移, 转角随时间关系的数据文件, 后面括号为详细的实验条件。

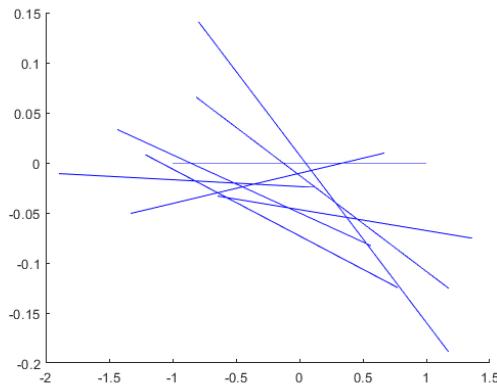
- cumulant.txt

这是记录了四阶累计量数据的文件。

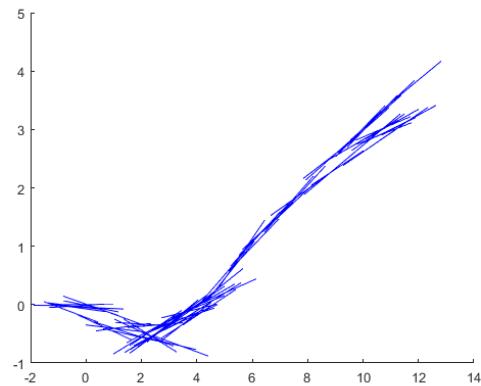
4 计算结果

选择正态分布的三个标准差分别为 $D_1 = 1.0, D_2 = 0.01, D_t = 0.05/0.08/0.10$ 做下面的计算并讨论。

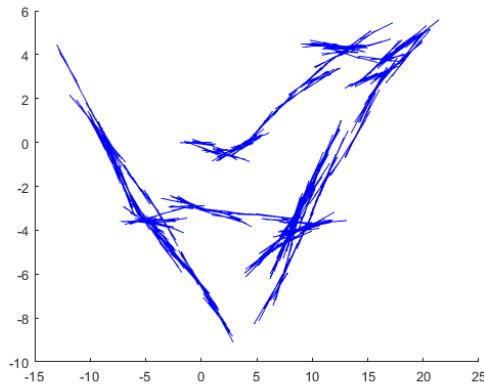
4.1 单个纳米棒随机行走展示



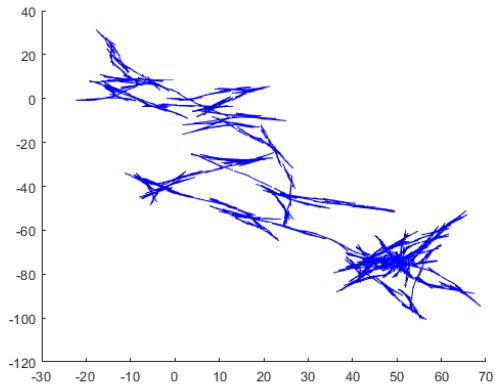
(a) step=8



(b) step=80



(c) step=800



(d) step=8000

图 2: 纳米棒运动随着时间演化

上面几个图将不同时间步长下的纳米棒运动形态做了一个展示，若将此过程进行下去，局部放大可以观察到自相似性。注意上图中纳米棒的形状简化为一根带取向(长轴方向)的直线。更直观的运动演示请参见压缩包中的 *rw.gif* 文件(置于 *reslut* 文件夹下)，记录了在前 800 步的纳米棒运动。

4.2 自关联函数计算

选取 $D_t = 0.5$ ，读入每一个时间步长对应的关联函数 $c(t)$ 。用 Matlab 自带的拟合工具箱拟合结果如下：

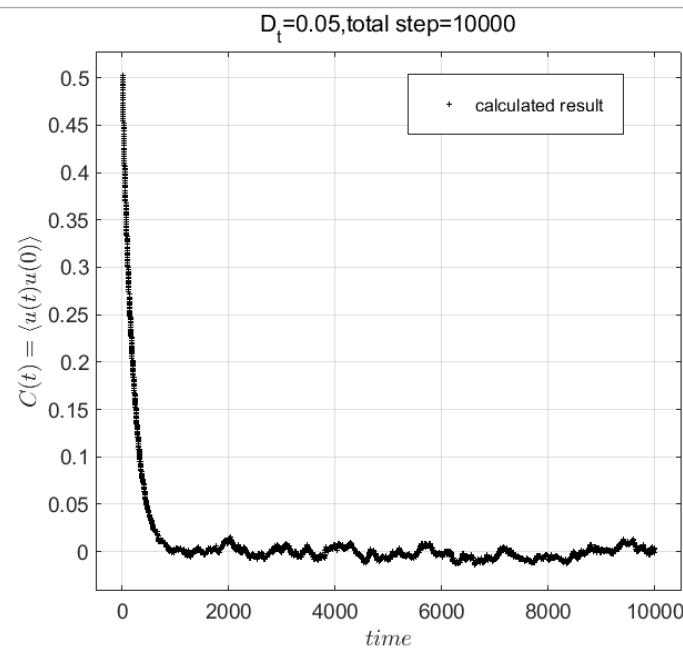


图 3: 实验数据

将下面的拟合结果和理论预言的 $\frac{1}{2} \exp(-\frac{D_t^2 t}{2})$ 对比。可以发现非常接近理论语言，偏差的原因在于系统平均理想情况下应该是对无数多个系统平均，而此处仅取了10000个，会有毛刺状出现。

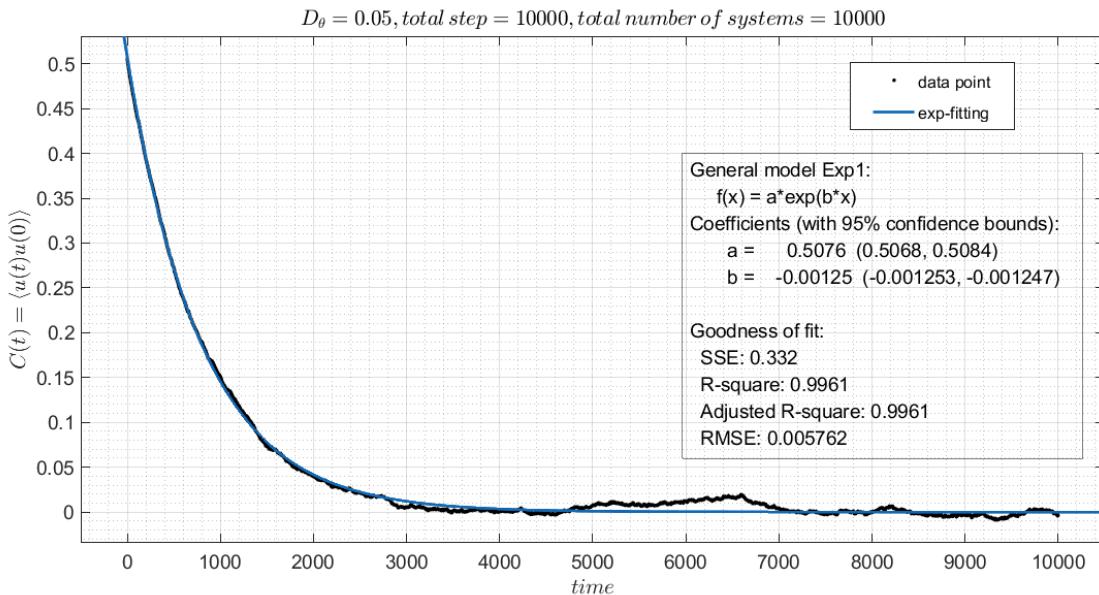


图 4: 拟合结果

4.3 扩散系数 D_x, D_y 计算

下面给出在实验室系观察到的 D_x, D_y 的数值结果。通过 $\langle x^2(y^2) \rangle / t$ 来计算这两个物理量。调整不同的 D_t , 即 θ 方向的转动速度来观察变化。

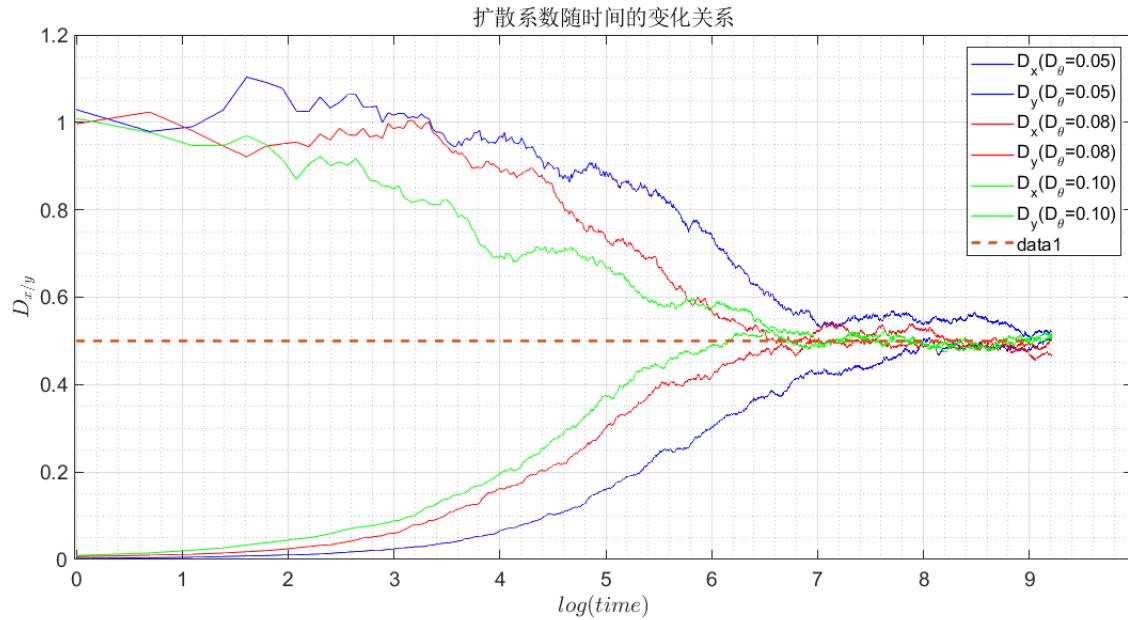


图 5: 不同 D_t 下数值计算结果(图中的 D_θ 即为 D_t)

图中虚线为 $D = 0.5001$ 。从图上可以明显看出, x, y 方向上的扩散一开始差别非常大, 这是因为我设定了初始 $\theta_0 = 0$, 一开始沿着 x 方向阻力很小, 很容易扩散, 所以体现出 $D_x \approx D_1^2 = 1 > D_y \approx D_2^2 = 10^{-4}$ 。但是由于 D_x, D_y 并非是完全解耦的, 它们通过“随机游走”的 θ 联系起来, 最后应当变得各向同性, 体现就在: $D_x = D_y = \frac{1}{2}(D_1^2 + D_2^2) \approx 0.5001$ 。同时, 可以看出随着 D_t 增大, 达到各向同性的时间减小, 这也符合物理直观。

4.4 四阶累计量的特征

如前所述, 为了和标准无 θ 耦合的情形对比, 采用10000个系统统计四阶累计量。

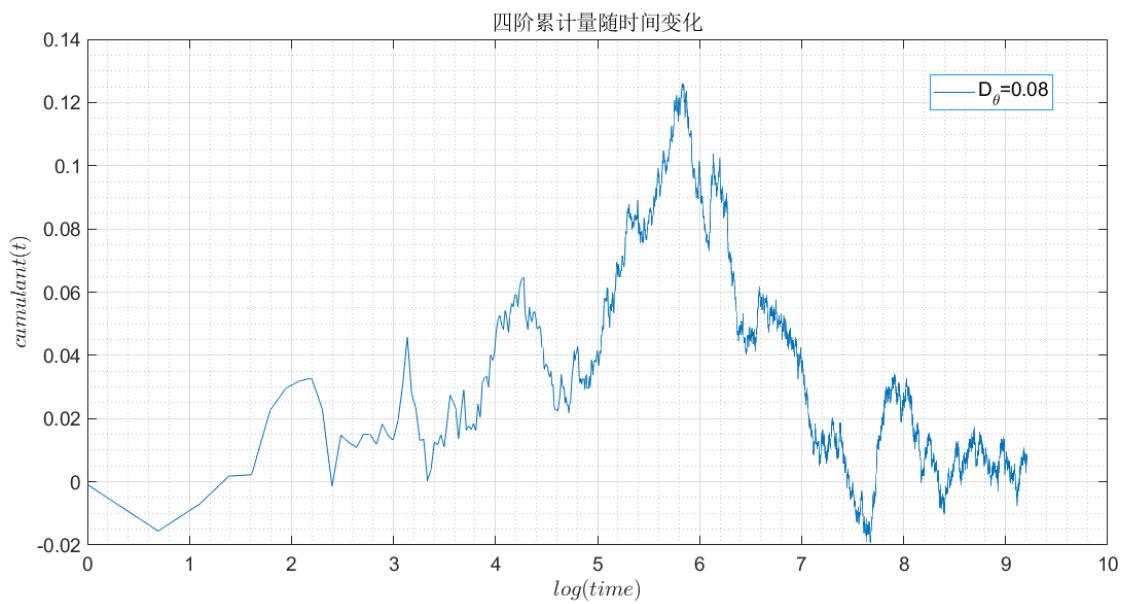


图 6: $D_t = 0.08$ 下数值计算结果(图中的 D_θ 即为 D_t)

四阶统计量体现出峰状的结构。从物理图像上来讲，一开始 θ 各向同性的平均作用不明显，尚可认为 x, y 解耦，因而统计量较小，和正态分布体现出相同特征。最后由于 x, y 扩散也已稳定，它们等价，也呈现出正态的特点。唯有在中间尚未达到各向同性时呈现出峰状的最大偏离特征。为了进一步得到平滑曲线，可以进一步增大系统数量，以期达到更好的效果。不过这里体现出的基本特征已经和参考文献[1]中的非常相似了。

5 总结

- 本次实验通过建立最简单的模型模拟了带取向纳米棒的随机游走。进一步考虑修正模型也许得到多聚体随机游走模型[2]。
- 多出来的 θ 自由度为 x, y 方向的随机游走提供了耦合，导致出一些与普通随机游走不同的有趣现象。
- 程序还需改进。有的地方动态分配多个大数组，明显减慢了运行速度。并且为了换不同的时间种子，用了几处`sleep`函数，也降低了运行速度。

参考文献

- [1] Yilong Han, Ahmed M Alsayed, Maurizio Nobili, Jian Zhang, Tom C Lubensky, and Arjun G Yodh. Brownian motion of an ellipsoid. *Science*, 314(5799):626–630, 2006.
- [2] Sethna J. Statistical mechanics: entropy, order parameters, and complexity[M]. Oxford University Press, 2006.
- [3] 中科大丁泽军计算物理讲义

计算物理A第十一次作业

王铠泽 PB18020766

1 作业题目

- 数值研究 $d(d = 1, 2, 3)$ 维空间中随机行走返回原点的几率 P_d , 讨论它随步数 N 的变化关系 $P_d(N)$, 能否定义相关的指数值?

2 实现方法和原理

- d 维随机游走的实现

本次实验采用离散化模型, 即一维链, 二维正方形网格, 三维正方体网络来模拟 d 维的随机游走。具体到算法上, 采用16807产生器每一次产生一个 $[0, 1]$ 之间的随机数 ξ , 若 $\xi > 0.5$, 则朝正方向前进一步; 若 $\xi < 0.5$, 则朝负方向前进一步。使用计数器 cnt 来计数在第 N 步返回原点的次数, 除以总系统数量 M , 就可以得到第 N 步返回原点的概率 $p(N)$ 。

- d 随机游走理论和常返性

对于网格模型, 显然只有在偶数步 N 时才能返回原点(奇数步必然在正负两个方向中有一个是多出一部分步长的)。

一维情形

$$p(N) = \frac{N!}{(\frac{N}{2}!)^2} \left(\frac{1}{2}\right)^N$$

二维情形

$$p(N) = \sum_{k=0}^{\frac{N}{2}} \frac{N!}{[k! \cdot (\frac{N}{2} - k)!]^2} \left(\frac{1}{4}\right)^N$$

三维情形

$$p(N) = \sum_{k=0}^{\frac{N}{2}} \sum_{j=0}^{\frac{N}{2}-k} \frac{N!}{[k! \cdot j! \cdot (\frac{N}{2} - k - j)!]^2} \left(\frac{1}{6}\right)^N$$

当步数 N 足够大的时候, 做Stirling近似展开得到 d 维随机游走标度律为[1]:

$$p^{2N}(0, 0) = C_d N^{-\frac{d}{2}}$$

常返性可以通过 $\sum_{N=0}^{\infty} p^{2N}(0, 0)$ 的收敛性来判断。一, 二维上该级数发散, 所以是常返的。三维以上发散, 不再常返[1]。

3 程式说明

- random_walk.c

该程式输出在 $d = 1, 2, 3$ 维情况下的 $P_1(N), P_2(N), P_3(N)$ 。

- ideal.c

该程式输出在 $d = 1, 2, 3$ 维情况下的理论 $P_1(N), P_2(N), P_3(N)$ 。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用 $[0, 1]$ 上的随机数填满。method 是关于初始种子的选择。method=0: 默认种子; method=1, 时间种子。程式中故意采用 *sleep* 函数就是为了得到不同的时间种子。

- time_seed.txt

16807 产生器抽样时对应的时间种子数据(每次1个种子)。调用多少次16807生成器就生成多少个数据记录。每一个分布对应的种子已经手动加上对应的实验了。种子产生公式如下:

$$\begin{array}{lll} \text{年: } & 0 \leq i_y \leq 99, & \text{月: } 1 \leq i_m \leq 12, \\ \text{时: } & 0 \leq i_h \leq 23, & \text{分: } 0 \leq i_n \leq 59, \\ & & \text{秒: } 0 \leq i_s \leq 59 \end{array}$$

则可设种子值为: $I_0 = i_y + 70(i_m + 12\{i_d + 31[i_h + 23(i_n + 59i_s)]\})$, 它的值约在区间 $[0, 2^{31}-1]$ 内, 第二部分的括号在 100 年内不会重复。

Tip: 程序中多处使用了 *sleep* 函数是为了换时间种子, 因而可能运行时间较久一点。

- 1/2/3d_random_walk.txt

记录了 $p(N)$ 的 txt 文件

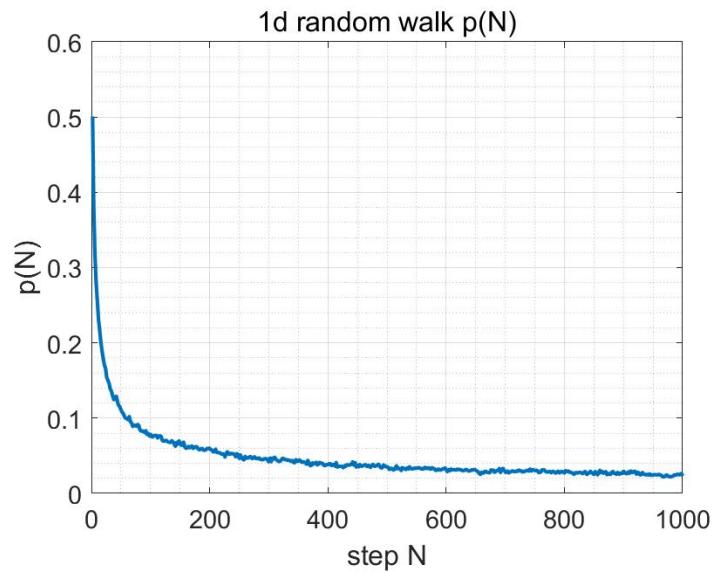
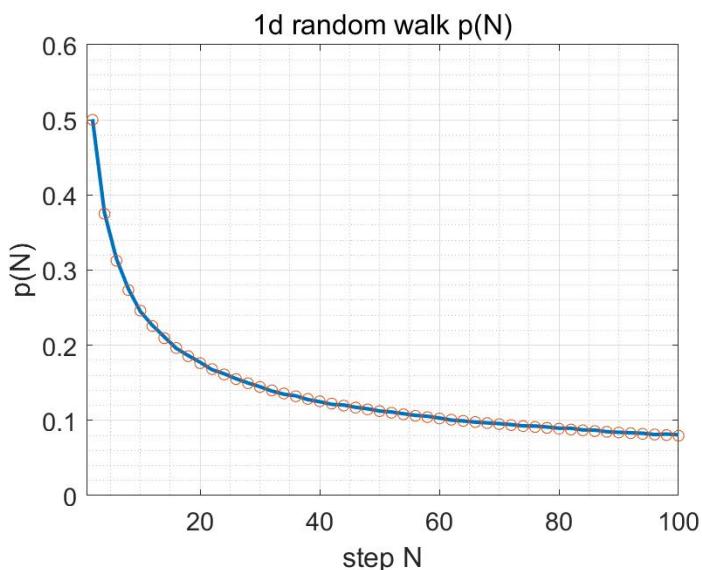
- 1/2/3d_ideal.txt

记录了理论上从 $N = 1 \sim 200$ 的 $p(N)$ 数值, 后续作图和实验值作比较。

4 计算结果

选择总步数为 $N = 1000$, 系统包含的系统数为 $M = 200000$ 个。以网格模型模拟随机游走, 得到的结果如下:

4.1 不同维度下 $p(N)$ 和理想结果的对比

(a) $p_{1d}(N)$ 

(b) 与理想值的比较(圆圈为理想值)

图 1: 一维情形

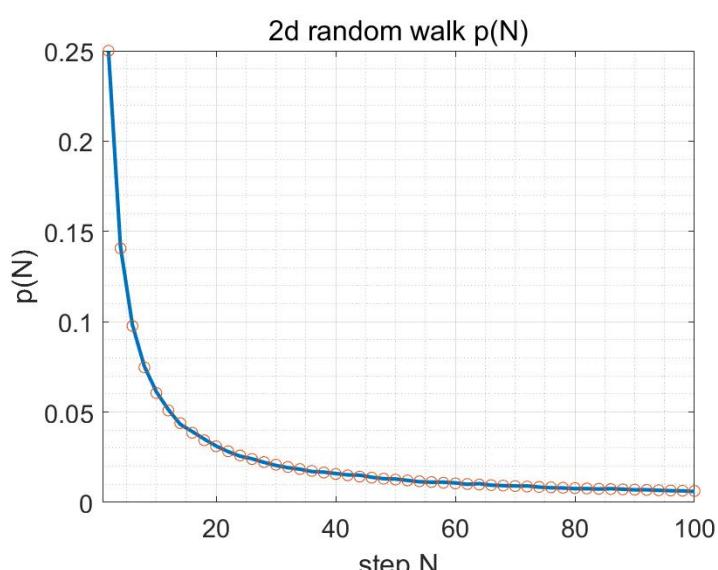
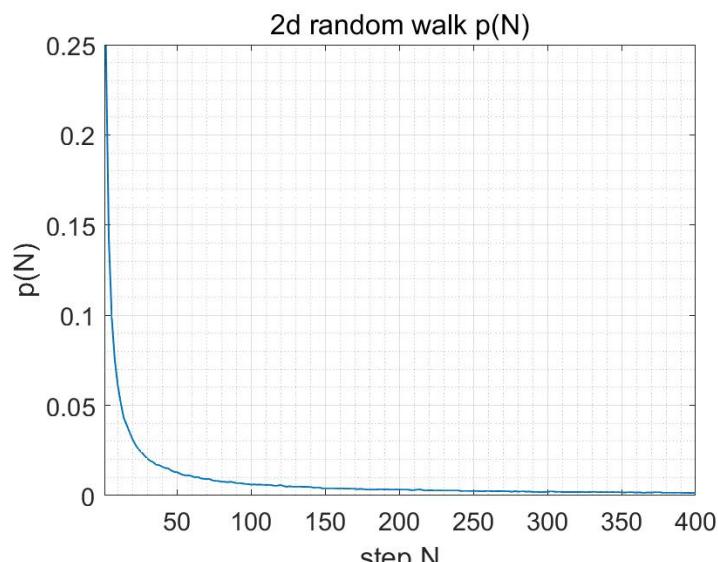


图 2: 二维情形

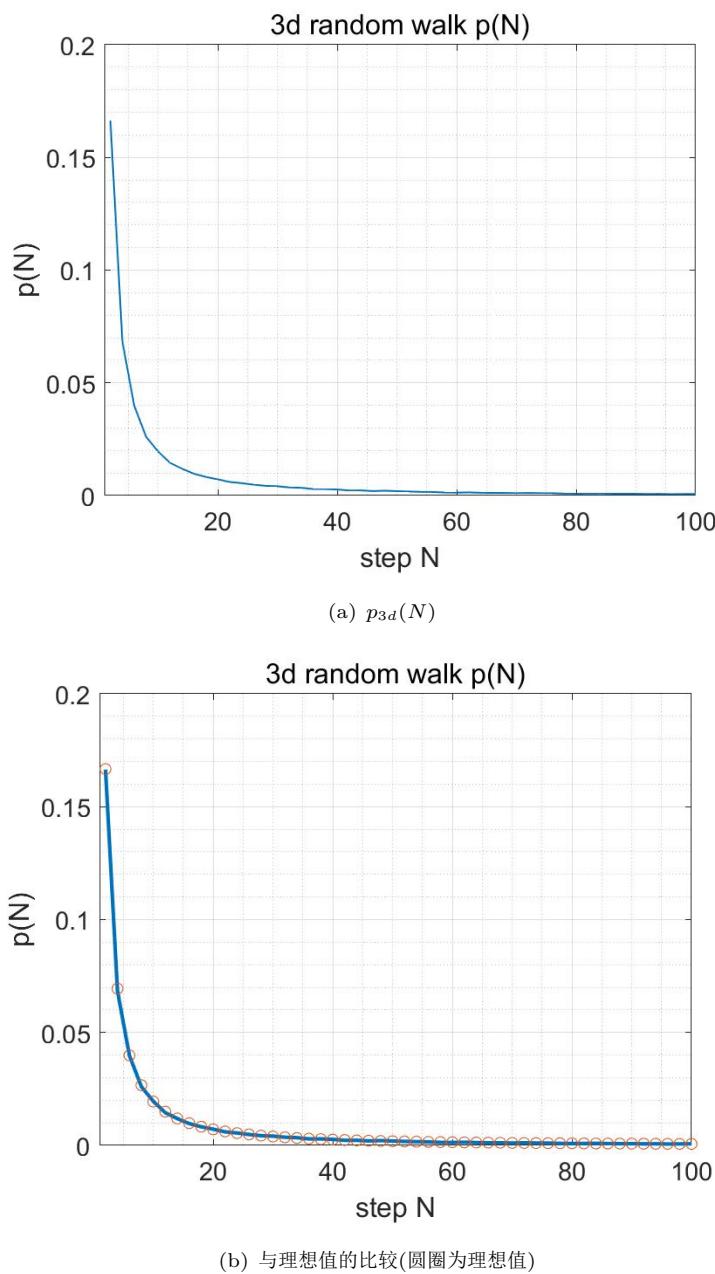


图 3: 三维情形

从上面的计算结果可以看出，我们的计算模拟和理论计算值非常吻合。除了少许的涨落，得到的曲线基本满足前面的理论推导。

4.2 标度律指数的计算

对 $d = 1, 2, 3$ 维分别做出双对数图，计算指数标度律如下

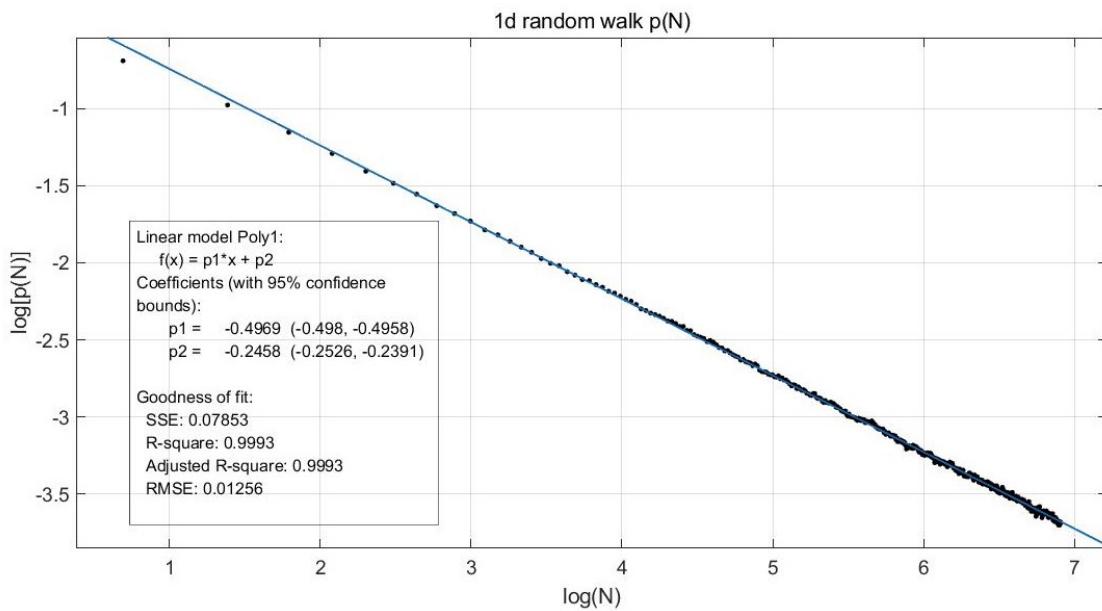


图 4: 1d随机游走

从拟合结果来看，一维的指数大约为 $\nu = -0.4969$ 。这和理论近似的-0.5非常接近，且随着 N 的增大，吻合程度增大，这也符合对大 N 值近似展开的前提。

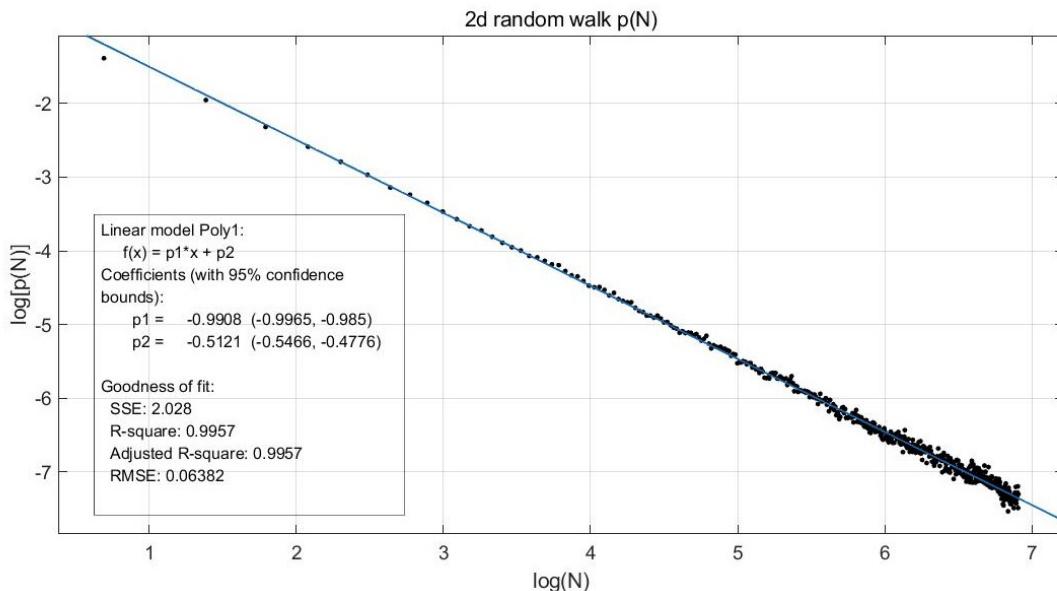


图 5: 2d随机游走

从拟合结果来看，一维的指数大约为 $\nu = -0.9908$ 。这和理论近似的-1.0非常接近，且随着 N 的增大，吻合程度增大，这也符合对大 N 值近似展开的前提。

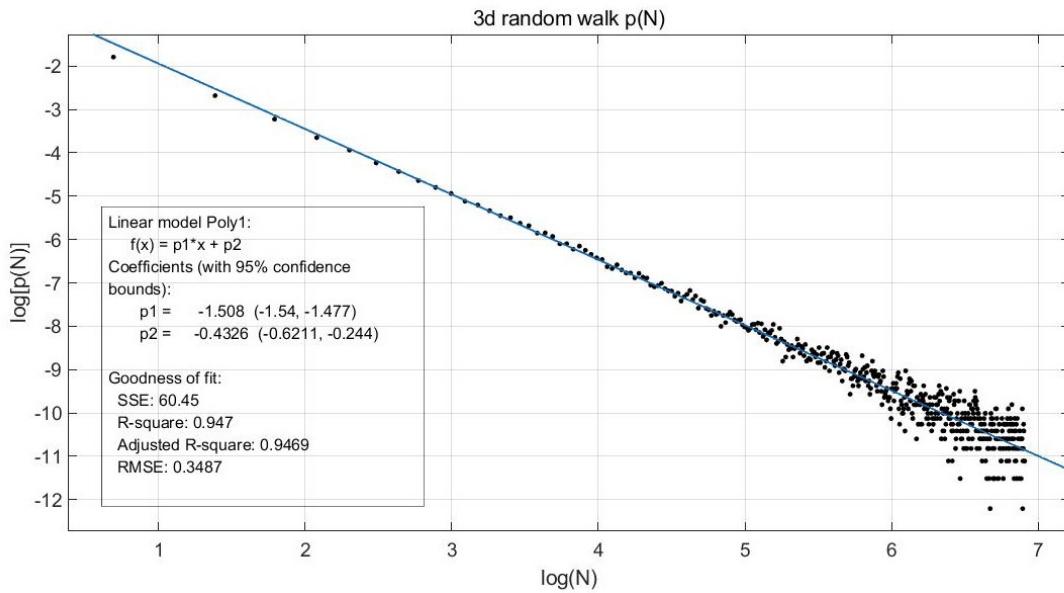


图 6: 3d随机游走

从拟合结果来看，一维的指数大约为 $\nu = -1.508$ 。这和理论近似的-1.5非常接近，且一开始随着 N 的增大，吻合程度增大，这也符合对大 N 值近似展开的前提。但是需要注意的是，当 N 继续增大时，可能是由于笔者选取的系综数量还不够多，在大步长的时候会出现 $\log(p)$ 无穷大的点，因为没有粒子经过原点。另一个值得注意的是，由于系综取得不够大，在后面很大的 N 时出现的统计涨落会变大，体现在图上就是最后模糊的一片点集，这是由系综中系统数 M 不够大引起的，而非步长 N 的原因。

4.3 三维和二维的随机游走比较

做出系综中一个粒子的随机游走轨迹。在二，三维的轨迹如下：

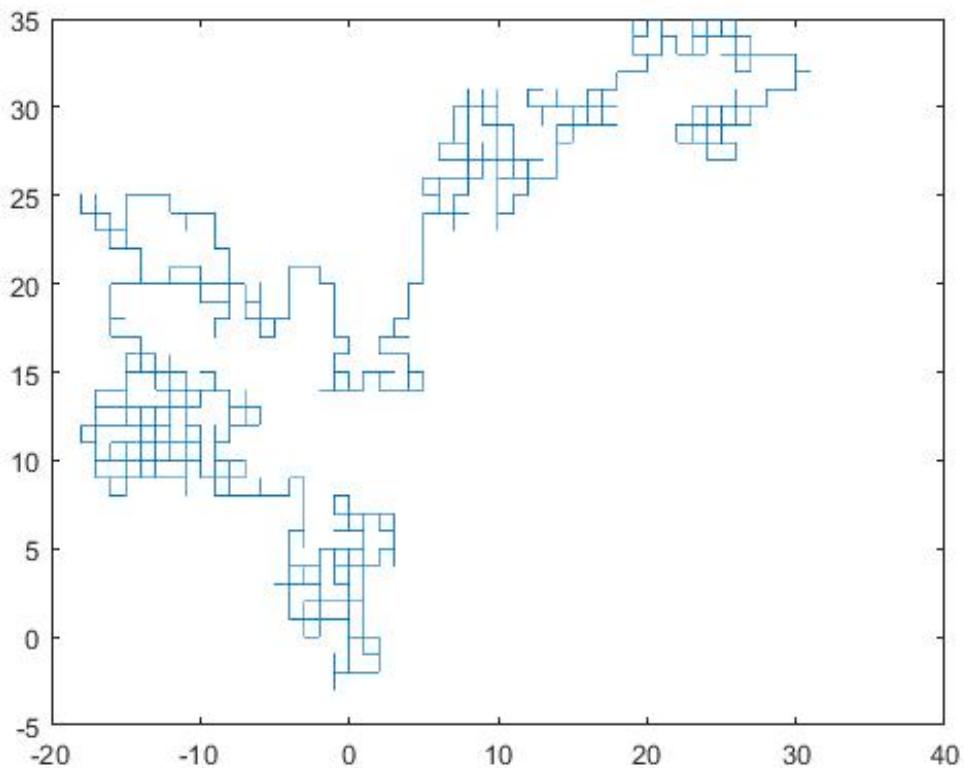


图 7: 二维随机游走

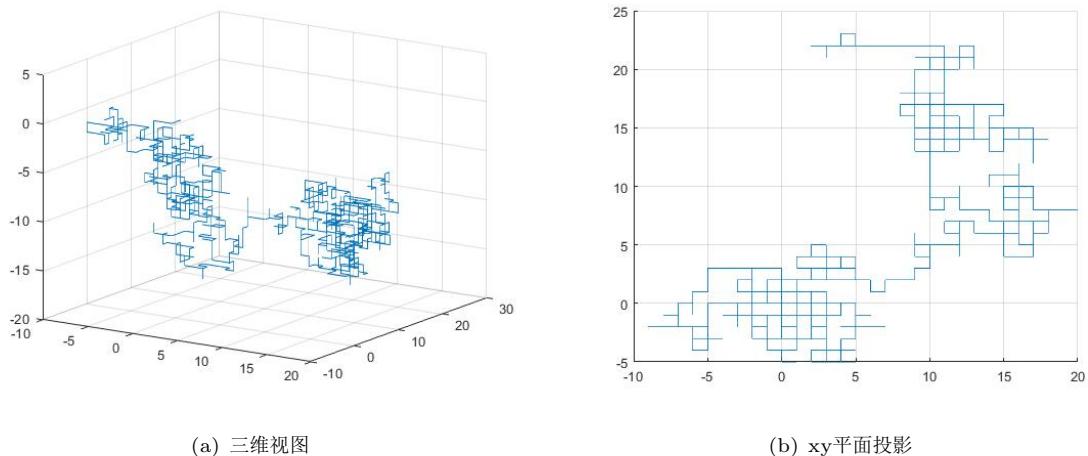


图 8: 三维随机游走

5 总结

- 本实验数值模拟了 $d = 1, 2, 3$ 维随机游走的标度律，得到了和理论自洽的结果。
- 程序还需改进。有的地方动态分配多个大数组，明显减慢了运行速度。并且为了换不同的时间种子，用了几处`sleep`函数，也降低了运行速度。

参考文献

- [1] Woess, Wolfgang. Random walks on infinite graphs and groups. Vol. 138. Cambridge university press, 2000.

计算物理A第十二次作业

王铠泽 PB18020766

1 作业题目

- 推导三角格子点阵上座逾渗的重整化群变换表达式 $p' = R(p)$, 其中端一端连接的条件是3个格点中的2个是占据态, 求临界点 p_c 与临界指数 ν , 与正确值(表1.6.1.3-1)相比较。

2 实现方法和原理

- 重整化群方法

重整化的基本思想就是对体系的长度尺度连续不断地做变换, 将体系元胞尺度由 a 变换成 ba (ba 应小于体系的相关长度 ξ), 相继标度变换的结果产生出一个流向图, 空间流向场趋向于若干特殊的不动点, 这些点在标度变换下保持不变。

- 三角格子的重整化

对于二维的三角格子, $b = (N)^{1/d} = \sqrt{3}$ 。具体尺度变化如下图:

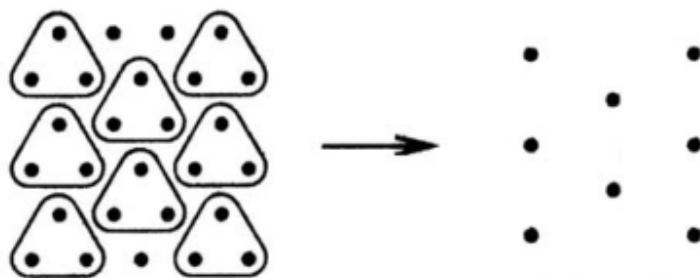


图 1: 重整化尺度变换

根据题目描述, 某个重整化后的个点导通对应的条件为至少两个原格点被占据, 如下图的四种情况:

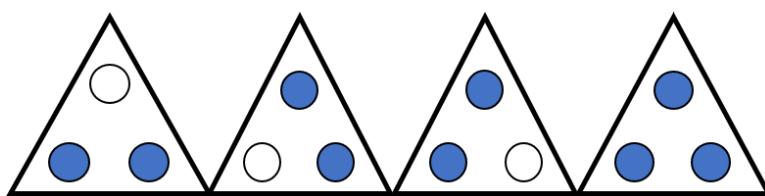


图 2: 导通条件(蓝色填充表示被占据)

- 临界点 p_c 计算

由上面导通条件, 可以得到重整化后每一个格点导通的概率为:

$$p' = R(p) = p^3 + 3p^2(1 - p) = -2p^3 + 3p^2$$

临界点(不动点) p_c 满足:

$$p_c = -2p_c^3 + 3p_c^2$$

求解这个方程, 得到:

$$p_c = 0.5, p_0 = 0, p_1 = 1$$

0和1是平凡解, 我们所关注的是 $p_c = 0.5$, 这才是临界点。

- 临界指数 ν 计算

重整化下的格子为了保持标度律不变, 应当选择重整化后的关联长度 $\xi' = \frac{\xi}{b}$ 。在接近 p_c 时, 有 $\xi(p) = |p - p_c|^{-\nu}$ 。所以:

$$|p' - p_c|^{-\nu} = \frac{1}{b} |p - p_c|^{-\nu}$$

而在 p_c 附近, 有:

$$p' - p_c = R(p) - R(p_c) = \lambda(p - p_c)$$

$$\text{其中 } \lambda = \frac{dR(p)}{dp} \Big|_{p=p_c}$$

$$\Rightarrow |p' - p_c|^{-\nu} = \lambda^{-\nu} |p - p_c|^{-\nu}$$

$$\Rightarrow b = \lambda^\nu$$

$$\Rightarrow \nu = \frac{\ln(b)}{\ln(\lambda)}$$

带入上面计算的数据 $p_c = 0.5$, 可以得到:

$$\nu = \frac{\ln\sqrt{3}}{\ln(\frac{3}{2})} \approx 1.354755646$$

最后, 将重整化计算得到的临界值和正确值进行比较:

图1.6.1.3-3 四种二维点阵的构型和配位数为3的 Bethe 点阵。

表1.6.1.3-1 各种点阵下座逾渗与键逾渗的逾渗阈值 p_c				
维数	点 阵	座逾渗 p_c	键逾渗 p_c	配位数
2	三角形	0.500000	0.34729	6
2	正方形	0.592746	0.50000	4
2	Kagome	0.6527	0.45	4
2	蜂房形	0.6962	0.65271	3
3	面心立方	0.198	0.119	12
3	体心立方	0.246	0.1803	8
3	简立方	0.3116	0.2488	6
3	金刚石	0.428	0.388	4
3	无规密堆积	0.27(实验值)		
4	简立方	0.197	0.160	8
5	简立方	0.141	0.118	10
6	简立方	0.107	0.094	12

图 3: 正确值表

可见 p_c 的座逾渗准确值为 0.5，和我们用重整化的方法做出来的完全一致。

3 总结

- 粗粒化近似，重整化群这一套方法是物理中重要的抓住本质的思想。对于相变理论，重整化往往能很好地计算出临界指数。本次作业就算一次很好的验证。
- 本次没有数值模拟环节，缺少对准确值的计算验证。等时间充裕，不失为一个很好的计算物理编程练习。

计算物理A第十三次作业

王铠泽 PB18020766

1 实现目标

用Metropolis – Hasting抽样方法计算积分：

$$I = \int_0^\infty (x - \alpha\beta)^2 f(x) dx = \alpha\beta^2$$

$$f(x) = \frac{1}{\beta\Gamma(\alpha)} \left(\frac{x}{\beta}\right)^{\alpha-1} \exp\left(-\frac{x}{\beta}\right)$$

设积分的权重函数为： $p(x) = f(x)$ 和 $p(x) = (x - \alpha\beta)^2 f(x)$ ，给定参数 α, β ，并用不同的 γ 值，分别计算积分，讨论计算精度和效率。

2 实现方法

按 $p(x)$ 为平稳分布来实现Metropolis – Hasting抽样。

设 T 与初态无关（即非对称的）： $T_{ij} = T(x \rightarrow x') = T(x') = 0.5 \exp^{-\frac{x'}{\gamma}}$

设 $x_0 = 1$ ，抽样 $x' = -\gamma \ln R$ ， R 为 $[0,1]$ 上均匀分布的随机数，由此抽取在 $(0, \infty)$ 上分布的 x' 。

$$\frac{p_j T_{ji}}{p_i T_{ij}} \equiv r = \left(\frac{x'}{x_i}\right)^{\alpha-1} \exp[-(x' - x_i)/\beta] \exp[(x' - x_i)/\gamma]$$

最后按照 r 的大小来决定接受概率，得到下一步的 x_{i+1}

$$x_{i+1} = \begin{cases} x' & R' < \min(1, r) \\ x_i & R' > \min(1, r) \end{cases}$$

其中 R' 为 $[0,1]$ 上均匀分布的随机数

最后只需要用求和近似积分：

$$I = \frac{1}{N-m} \sum_{i=m+1}^N (x_i - \alpha\beta)^2$$

其中 m 是为了去除前面热化阶段引入的参数，本次实验中，取 $m = \frac{N}{10}$ 。

本实验中，取 $\alpha = 2, \beta = 1$ ，调整不同 γ 来讨论精度和效率。

3 程式说明

- metropolis_1.c

这是一个对于生成对于 $N = 10^6$ 步数的Metropolis抽样计算积分的误差评估的程序。其输出为不同 γ 值下面的积分误差，并且输入到文件中。注意到，循环中的 γ 范围根据需要可以修改。

- metropolis_2.c

这是一个对于用于生成对于不同步数 $N = 10^1 \sim 10^7$ 步数的Metropolis抽样计算积分的误差评估的程序。其输出为不同 γ 值下面的积分误差，并且输入到文件中。注意到，循环中的 γ 范围根据需要可以修改。本次实验中取 γ 值为0.2, 1.4, 4.0。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的[0, 1]上均匀分布随机数函数的头文件。

```
void rdm(int N, double *x, int method)
```

该函数将输入的指针 x 对应的长度为 N 的数组用[0, 1]上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。

- time_seed(gamma range).txt

对于括号内标识的 γ 取值范围对应使用的时间种子文件。注意在程式中生成随机数时，一组随机数使用时间种子，另一组采用默认种子值($I=1$)。16807产生器抽样时对应的时间种子数据(每次1个种子)。调用多少次16807生成器就生成多少个数据记录。每一个分布对应的种子已经手动加上对应的实验了。种子产生公式如下：

$$\begin{aligned} \text{年: } & 0 \leq i_y \leq 99, \quad \text{月: } 1 \leq i_m \leq 12, \quad \text{日: } 1 \leq i_d \leq 31 \\ \text{时: } & 0 \leq i_h \leq 23, \quad \text{分: } 0 \leq i_n \leq 59, \quad \text{秒: } 0 \leq i_s \leq 59 \end{aligned}$$

则可设种子值为： $I_0 = i_y + 70\left(i_m + 12\left\{i_d + 31\left[i_h + 23\left(i_n + 59i_s\right)\right]\right\}\right)$ ，它的值约在区间 $[0, 2^{31}-1]$ 内，第二部分的括号在100年内不会重复。

4 计算结果

4.1 不同 γ 参数值对于积分的影响

取 $N = 10^6$ ，调节 γ 的取值，使得其取值落在 $0 \sim 10^4$ 范围，然后做出误差- $\log(\gamma)$ 图像如下：

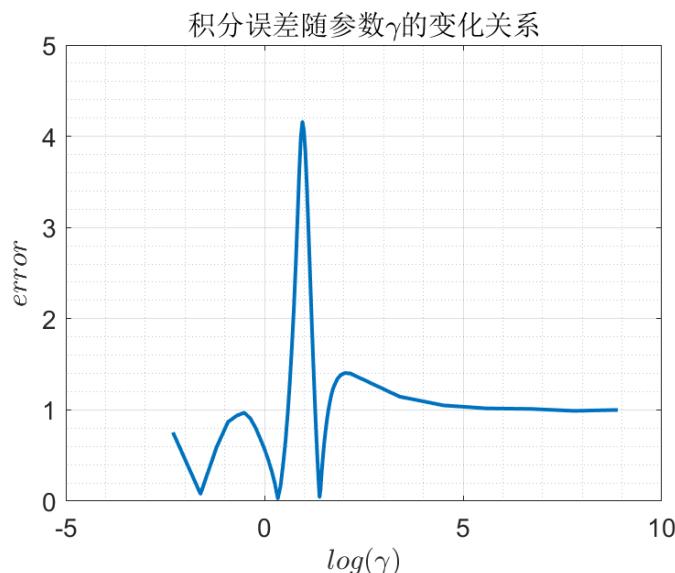


图 1: error- $\log(\gamma)$ 图线

可以观察到基本可以认为在 $\gamma > 0$ 的范围内，积分误差会出现三个极小值，分别是 0.2, 1.4, 4.0。在其附近更加精确的采点绘出的图像如下：

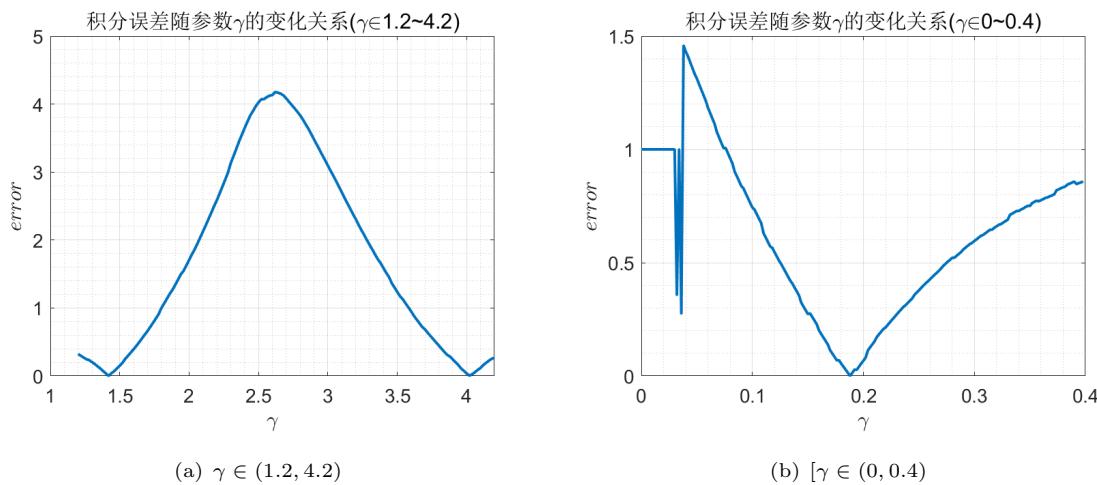


图 2: 误差随步数 N 变化曲线

可以非常明确的确定这就是三个极小值点，能够得到最理想的积分误差。从图线中可以看出，积分误差对于 γ 的依赖性非常明显，选择不合适的 γ 很可能会导致积分误差非常大(即使已经走了很大的步数)。有趣的是在很大的 γ 下，误差基本稳定在1左右。不过这也是可以理解的： γ 是 T_{ij} 抽样时对应的指数分布的参数，当 γ 大到一定程度时，指数分布下降得非常快，基本只剩下0附近的点被抽到，这等价于很小的步进距离，这当然是不合适的，要经过相当长的步长才能达到平稳分布。同理，太小的 γ 也是不合适的，因为这样每一步都走的很大，很难以到达合适的平衡位置，这在图上也有所体现。

4.2 不同步数 N 对积分误差的影响

取在上述分析中误差的极小值点来进行步数依赖的分析。

积分步数 N	绝对误差 ϵ
10	1.000000
100	0.192531
1000	0.137306
10000	0.028388
100000	0.099934
1000000	0.099932
10000000	0.087569

表 1: $\gamma = 0.2$ 时的积分误差表

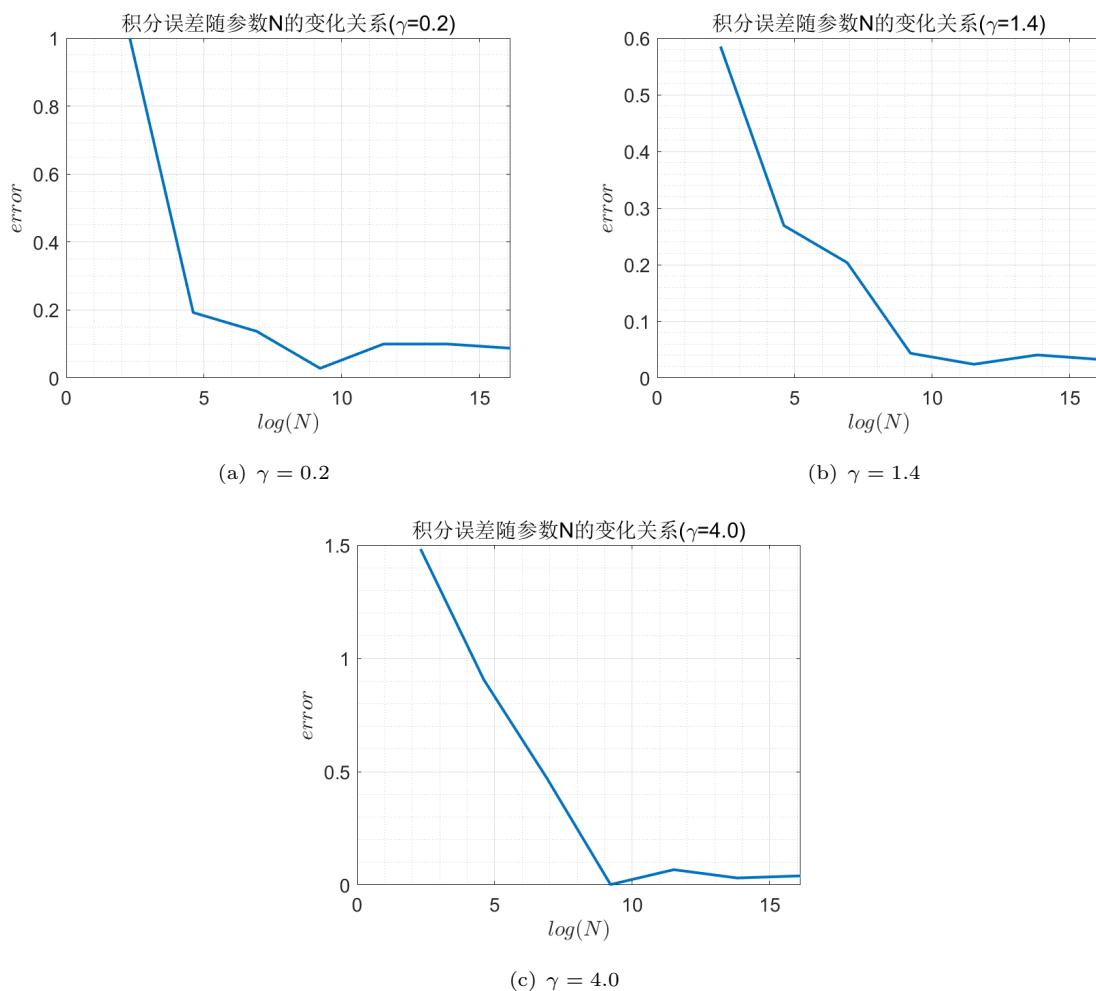
积分步数 N	绝对误差 ϵ
10	0.585176
100	0.269133
1000	0.203556
10000	0.043600
100000	0.024212
1000000	0.040619
10000000	0.032623

表 2: $\gamma = 1.4$ 时的积分误差表

积分步数 N	绝对误差 ϵ
10	1.482793
100	0.905431
1000	0.469053
10000	0.001988
100000	0.067546
1000000	0.030883
10000000	0.040191

表 3: $\gamma = 4.0$ 时的积分误差表

可以看出, 当 N 较小的时候, 误差很大, 随着 N 增大, 误差减小, 并且后面误差随着 N 的增大几乎不会出现量级上的变化, 比较稳定, 这时候想要加大精度, 可能要再走很多步, 说明增大步数并不能非常有效地提高精度。以步数的对数作为横坐标, 误差作为纵坐标, 做出的曲线图如下:

图 3: 误差随步数 N 变化曲线

从曲线上更加直观地看出，最后随着步数增加误差和步数之间并没有很明显的单调递减关系，而是变得逐渐平缓，控制在0.1的量级。

5 总结

- 本次实验通过Metropolis抽样实现积分的计算，选择合适的参数值，能够得到较为理想的模拟结果
- 可以发现Metropolis抽样对于步进矩阵 T 的选择有很高的要求，一个合适的 T 矩阵应该使得每一步的步长不能太长也不能太短，这样才能在尽可能少的步数到达热平衡态，得到我们所要的重要抽样分布。

计算物理A第十四次作业

王铠泽 PB18020766

1 作业题目

- 一篇应用MCMC方法研究聚乙烯小球自组装结构的研究论文“Formation of wafer-scale monolayer close packed polystyrene spheres template by thermally assisted self-assembly”在投稿某刊物后被审稿人拒稿，现作者欲以向刊物编辑申诉。请根据文章内容和审稿人评审意见，撰写申诉理由（你认为，作者在文中阐述的方法和概念以及审稿人的评论意见有哪些是合理的，哪些是需要修正的，或者哪些是需要进一步阐明的）。进一步，如果你是作者的话，你将如何进行该工作以及建立模型？

2 文章摘要

本文提出通过旋涂和热处理结合的方式实现高质量的密堆自组装聚苯乙烯小球阵列在晶片尺寸规模(wafer-scale)上的量产。作者声称经由蒙特卡洛抽样方法对传统自组装理论进行了补充，得到了实验中热处理的最佳温度，为制备过程提供了很大的便利。

3 评论和意见

1. 理论模型中相互作用只考虑入和最近邻核(nucleus)的作用不合理

在接近最终的构型过程中，球与球之间是存在着紧密排布的情形，这时最近邻核可能不止一个。再者，此时的液面形变导致的作用力(capillary force)由于曲率减小可能不是简单地随距离速降的一个函数，所以只做最近邻截断可能会出现问题。

再者，本文的模型本身是想讲多体问题(每个球和球之间相互作用都计入)完全由几个分布较为稀疏的随机选取的核来代替。这一点的合理性有待质疑，在物理上的对应不是十分清晰。我的想法和评审一样是no guarantee。

如果我是作者，可能会更加细致地对不同构型阶段采取更为精细的，并且更有物理意味(例如引入一个平均吸引场)来处理这个问题。

2. 相互作用具体形式的参数问题

评审也提到关于相互作用的形式假设本身就存在问题。而我觉得存在更大问题是这些参数视为温度函数是如何确定的问题。文中的相互作用中的 Q 由下式确定：

$$Q = U_0 = A e^{-\frac{\Delta}{k_B T}} e^{-\frac{s}{\tau_0(1-\frac{T}{T_c})}} \quad (4)$$

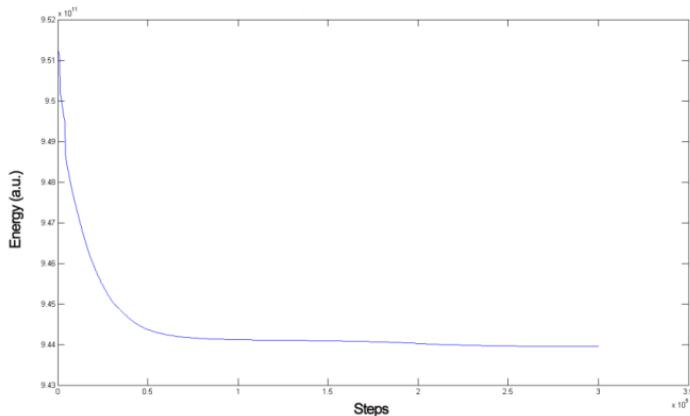
很显然，势垒 Δ 和弛豫时间 τ_0 应该是温度的函数，但是具体是什么函数，作者也没有指出。如果直接当成常数计算那是更加不合理。

另一个很让人费解的是出现在指数上的Metropolis抽样的步数 s 。我们知道，Metropolis抽样是对平衡构型的抽样，它的步数代表的并不是时间先后，仅仅是抽样的一个个中间过程，物理上是没有

意义的。所以将其置于指数上作为随时间衰减的意思来用是错误的，这可能也是评审让作者回去多看看书的原因。而作者最后在申诉信中回复的时候似乎避重就轻，没有意识到上述问题。

3. 系统能量变化令人费解

在补充材料中，可以找到系统能量随着步数变化如下：



按作者自设，势能 Q 应该是指数下降的，能量最终达到随着步数不变的稳态实在是令人费解。我觉得要么是算法笔误，要么是计算结果完全错误。

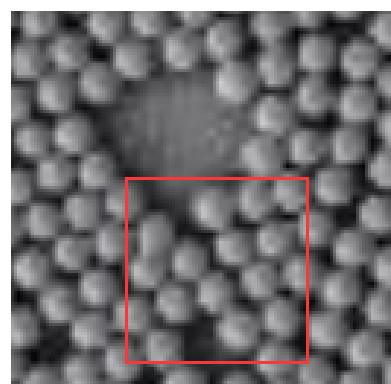
如果我是作者，应该好好检查一边算法和程序再发表。

4. 三角网格模型本身带来信息的缺失

在引入相互作用之后，作者讲小球的构型格点化，变为离散的三角网格。虽然结果上可以模拟得到密排的小球图像，但是存在对实际实验严重的信息丢失。

首先，网格模型中是体现不出球堆叠的情况，没有设置重复占据(over occupied)的情形。这对后续抽样时后的步进选择造成比较大的影响。要是我是作者，我会设置好网格上的格点被多重占据对应能量，参与到抽样过程中去，而不是每次只占据空格点。

再次，网格模型完全体现不出实验中自组装阵列畴与畴之间错位的情形。就算是作者得到的所谓缺陷，空隙始终也只是遵循着三角格子一定排布的空隙。详见下图这种情形：



上图中红框中这种错位是模拟所无法得到的。

如果我来处理这个问题，比较好的解决方法是牺牲简单性，直接放弃网格方法，采用二维平面上自由运动的方式来模拟。

5. 方法论可能存在根本问题

*Metropolis*抽样是对平衡构型的抽样。直接当作正则系综处理就是默认分布是*Boltzmann*分布，但是自组装是发生于远离平衡态的动力学过程，甚至是*Boltzmann*输运方程都无法准确描述的。如此抽样方法本身是否合理实在有待商榷。

如果我是作者，可能我会采用MD(分子动力学模拟)这种 bottom-up 的方法来更模拟远离平衡态的演化。

6. SEM 图像傅里叶变换处理存在局限

对SEM图像做图像处理(FT变换)得到几个谐波分量，测量随温度变化图像来表征球与球之间密堆质量在数学上是正确的，但是实际中是很容易受实验条件影响的方式。本身实际的结构一定是 $x - y$ 平面上各向异性的，所以拍摄角度不确定、图片像素排列方向不一样就会带来谐波计算的影响，在想要比较精确地确定有序度时可能存在问题。

我建议是采用其他方式来衡量密堆度，例如计算对于一个固定球，某一固定理论晶面上实际球和理论球的比较。

7. 一些文章处理上的问题

多次提到 45° 是最佳的温度，但是缺鲜有展示此时的构形，SEM图像等。

若我是作者，如果我坚信自己的计算没有问题，应该多放出在这个最佳温度附近的结果。

8. 其他

经查，这篇文章在删去全部的计算内容，改进了实验方法后发表在 *Langmuir* 杂志上 (*Langmuir* 2013, 29, 14017—14023)。侧面说明了这部分的工作是存在问题的。

计算物理A第十五次作业

王铠泽 PB18020766

1 作业题目

- 设体系的能量为 $H = x^2/2\sigma_x^2 + y^2/2\sigma_y^2$ (以 kT 为单位), 采用 *Metropolis* 抽样法计算 $\langle x^2 \rangle$, $\langle y^2 \rangle$, $\langle x^2 + y^2 \rangle$, 并与解析结果进行比较。抽样时在 2 维平面上依次标出 Markov 链点分布, 从而形象地理解 Markov 链。

2 实现方法和原理

- Metropolis* 抽样方法

本次实验中, 设 $\sigma_x = \sigma_y = 1$

理论上解析计算得到:

$$\langle x^2 \rangle = \frac{1}{Z} \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx \exp\left(-\frac{x^2}{2} - \frac{y^2}{2}\right) x^2$$

其中 Z 为配分函数, 其数值为:

$$Z = \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dx \exp\left(-\frac{x^2}{2} - \frac{y^2}{2}\right) = 2\pi$$

$$\Rightarrow \langle x^2 \rangle = 1$$

同理得到:

$$\langle y^2 \rangle = 1, \langle x^2 + y^2 \rangle = 2$$

采用正则系统的 *Boltzmann* 分布作为平衡构型时的分布。所以从 (x, y) 过渡到 (x', y') 的概率为:

$$p(\vec{x} \rightarrow \vec{x}') = \min\left\{1, \exp\left(\frac{x^2/2\sigma_x^2 + y^2/2\sigma_y^2}{x'^2/2\sigma_x^2 + y'^2/2\sigma_y^2}\right)\right\}$$

算法基本描述:

本次实验采用每次在原来位置上随机抽取 $[-A, A]$ 上的随机数作为 dx 、 dy , 进行概率判断。若能量降低, 则前进概率等于 1, 直接进入下一个状态。若概率小于 1, 则再次生成 $[0, 1]$ 中的随机数 r , 若 $r < \exp(\Delta E/kT)$, 则刷新状态, 进行下一步, 否则就维持原样。

最后注意: 采用 *Metropolis* 抽样计算系综平均时要去掉热化阶段:

$$\langle X \rangle = \frac{1}{N - M} \sum_{i=M+1}^N X_i$$

3 程式说明

- metropolis.c

这是一个对于用于生成对于 $N = 10^6$ 步数的Metropolis抽样计算积分的误差评估的程序。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的[0, 1]上均匀分布随机数函数的头文件。

```
void rdm(int N,double *x,int method)
```

该函数将输入的指针x对应的长度为N的数组用[0, 1]上的随机数填满。method是关于初始种子的选择。method=0:默认种子;method=1,时间种子。

- time_seed(gamma range).txt

对于括号内标识的 γ 取值范围对应使用的时间种子文件。注意在程式中生成随机数时，一组随机数使用时间种子，另一组采用默认种子值($I=1$)。16807产生器抽样时对应的时间种子数据(每次1个种子)。调用多少次16807生成器就生成多少个数据记录。每一个分布对应的种子已经手动加上对应的实验了。种子产生公式如下：

$$\begin{aligned} \text{年: } & 0 \leq i_y \leq 99, \quad \text{月: } 1 \leq i_m \leq 12, \quad \text{日: } 1 \leq i_d \leq 31 \\ \text{时: } & 0 \leq i_h \leq 23, \quad \text{分: } 0 \leq i_n \leq 59, \quad \text{秒: } 0 \leq i_s \leq 59 \end{aligned}$$

则可设种子值为： $I_0 = i_y + 70 \left(i_m + 12 \left(i_d + 31 \left[i_h + 23 \left(i_n + 59 i_s \right) \right] \right) \right)$ ，它的值约在区间
 $[0, 2^{31}-1]$ 内，第二部分的括号在100年内不会重复。

4 计算结果

4.1 不同步长下的平均值计算

本次实验中，采取总步长 $N = 10^6$ ，热化步长 $M = 10^4$ 。为了探究不同行走步长 A 对结果的影响，对其取值如下：

$$A = 0.001, 0.002, 0.005, 0.01, 0.04, 0.08, 0.1, 0.5, 1, 2, 4, 10, 100$$

由此得到的 $\langle x^2 \rangle, \langle y^2 \rangle, \langle x^2 + y^2 \rangle$ 列表如下：

系综平均 N	$A = 0.001$	$A = 0.002$	$A = 0.005$	$A = 0.01$	$A = 0.04$	$A = 0.08$
$\langle x^2 \rangle$	59.965622	42.503714	9.911649	3.336977	0.916165	0.983122
$\langle y^2 \rangle$	31.747700	21.228011	4.922366	1.654147	0.899795	0.930838
$\langle x^2 + y^2 \rangle$	91.713322	63.731725	14.834015	4.991124	1.815960	1.913960

表 1: $A = 0.01 \sim 0.08$ 系综平均计算表格

系综平均 N	$A = 0.1$	$A = 0.5$	$A = 1$	$A = 2$	$A = 4$	$A = 10$	$A = 100$
$\langle x^2 \rangle$	0.999358	1.002436	1.000011	0.998231	1.000731	1.005191	1.049830
$\langle y^2 \rangle$	0.950428	1.010010	1.000383	0.995312	1.001040	1.001623	1.045211
$\langle x^2 + y^2 \rangle$	1.949786	2.012446	2.000394	1.993542	2.001770	2.006814	2.095041

表 2: $A = 0.1 \sim 100$ 系综平均计算表格

从上面的表格可以看出，系综平均的计算值误差随步长并不是单调的关系。步长过小/过大都不能得到很好的结果，这将在稍后继续讨论。我们的模拟方法存在一个“最佳步长”，不至于太小，走不到理想分布；也不至于太大，每一步的涨落太大，这13个值中最佳的是 $A = 0.04$ 。下面给出 $\langle x^2 + y^2 \rangle$ 的误差表格以更加直观地表述上面的观点。

步长	误差 ϵ
$A = 0.001$	89.713322
$A = 0.002$	61.731725
$A = 0.005$	12.834015
$A = 0.01$	2.991124
$A = 0.04$	0.184040
$A = 0.08$	0.086040
$A = 0.1$	0.050214
$A = 0.5$	0.012446
$A = 1$	0.000394
$A = 2$	0.006458
$A = 4$	0.001770
$A = 10$	0.006814
$A = 100$	0.095041

表 3: $A = 0.01 \sim 100$ 系综平均误差表格

将误差-步长曲线绘制如下(由于取值范围跨度较大，取用对数坐标)：

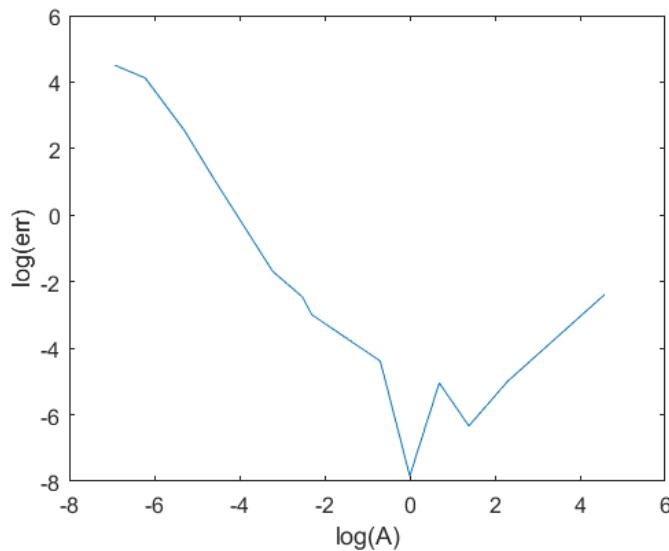


图 1: 误差曲线

显然在 $A = 1$ 附近会存在极小值，过大或过小步长误差会增大，特别是小步长。还可以从 $Markov$ 链抽样出来的 x 最终分布的形态来观察是否达到良好的平衡构型，得到准确的结果。理论上，平衡态时， x, y 的边缘分布都是 $\sigma = 1$ 的高斯分布。下面分别给出 $A = 0.001$ (小步长), $A = 1$ (恰当步长), $A = 10$ (大步长), $A = 100$ (巨大步长) 的 x 直方图统计情况。我们在下一小节中集中讨论这些问题。

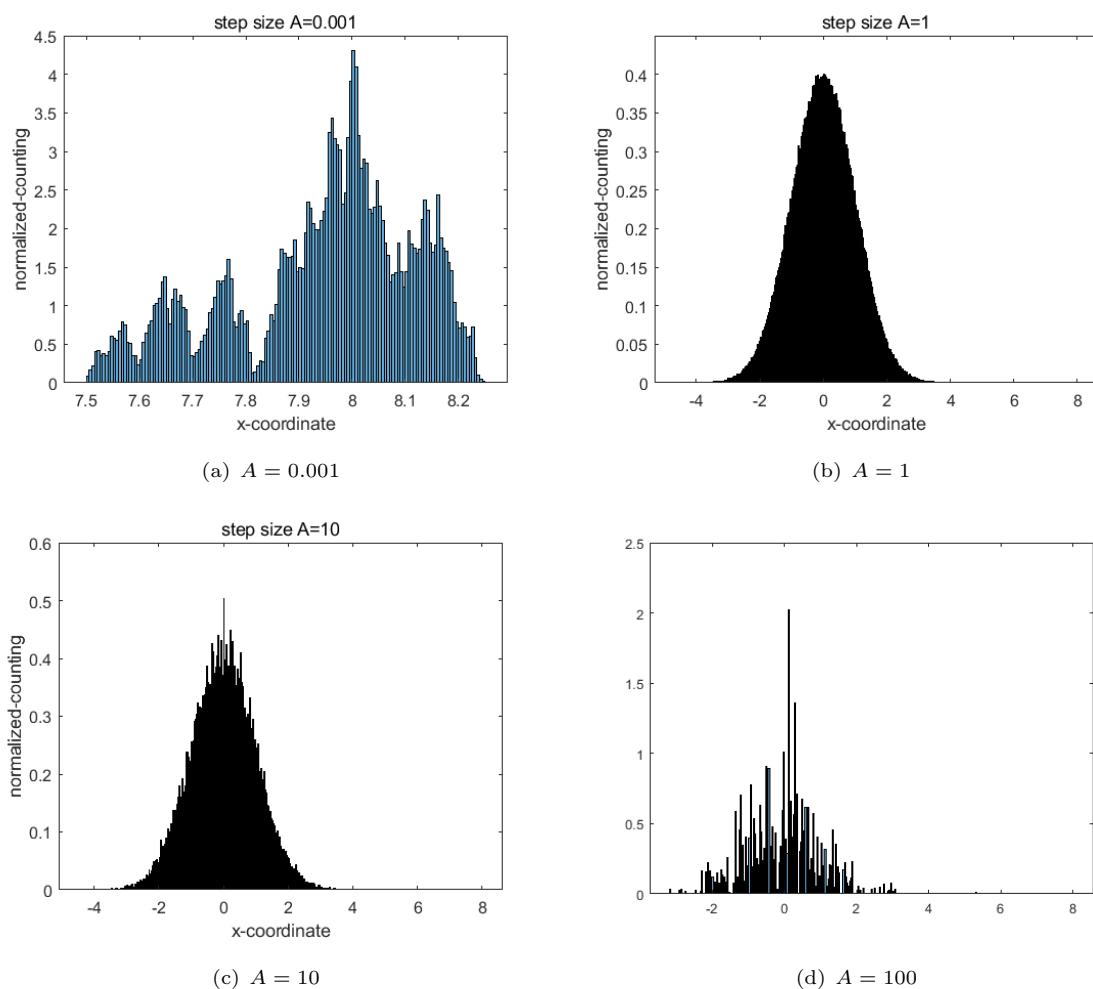
5 $Markov$ 链的进一步讨论

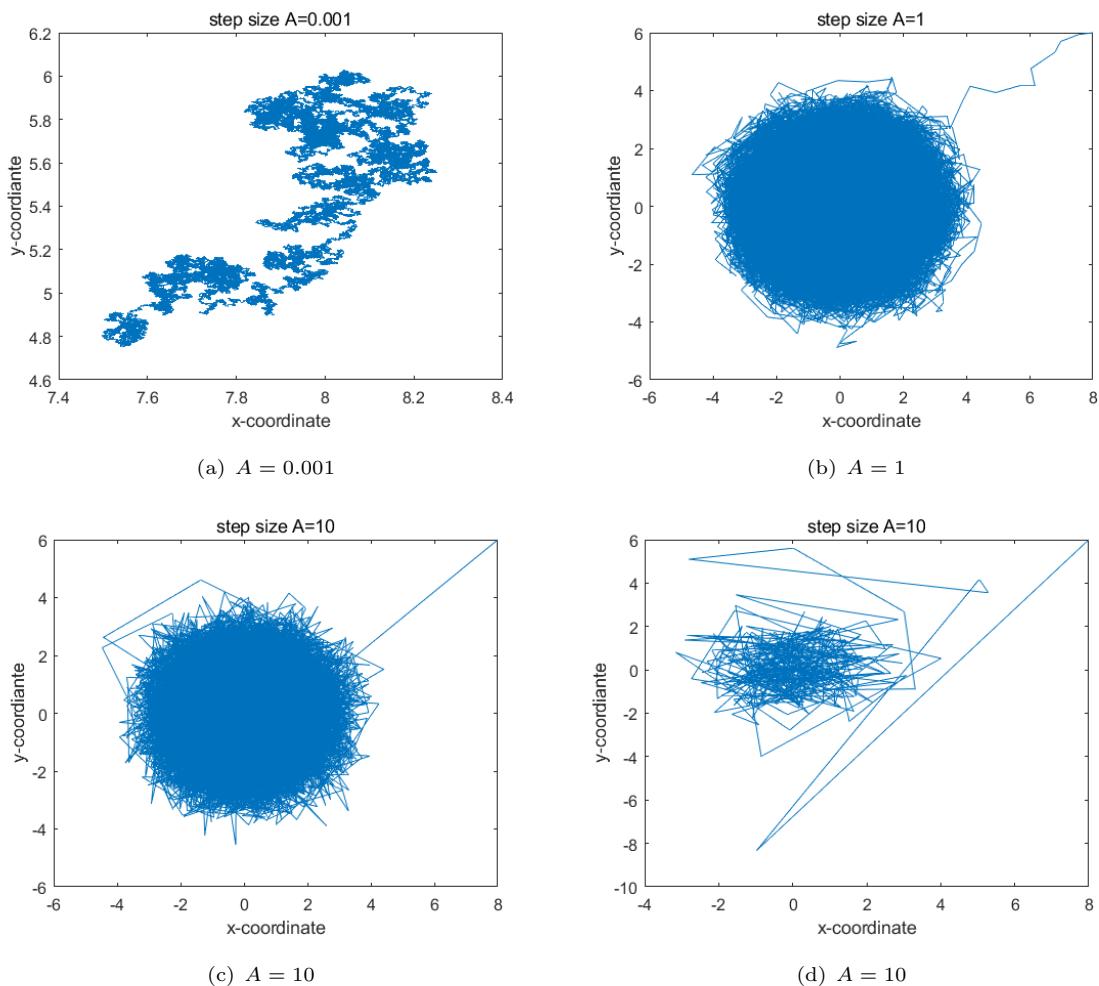
首先给出热化之后的 x 分布直方图：

从图上可以看出，当步长过小或者过大时分布都不是那么理想，特别是步长小的时候，就算用尽了 $N = 10^6$ 步，看起来似乎离平衡位置还很远。并且可以看到小步长的概率密度图上有很多个小峰状结构，这是由于步长太小，倾向于在一个地方“打转”的可能性变大了，所以出现一系列峰。

至于在步长很大($A = 100$)的时候，分布倒是已经体现出正态分布的雏形了，对称性也基本具备，但是由于步数不够多，还不够得到平衡时的理想构型。从(b), (c), (d)三图能看出分布被步长变大破坏的过程。

总的来说，步数越大越能得到理想效果；步长要选取合适的，一般判据是选取步长和数据方差在一个数量级，这样得到的收敛速度快，精度高。对于 $Markov$ 链直观的可视化抽样图如下给出：

图 2: 不同 A 情况的 x 分布概率密度

图 3: 不同 A 情况Markov链抽样情况

三个Markov链都是从 $(x_0, y_0)=(8, 6)$ 开始模拟。在小步长 $A = 0.001$ 情况下, 至循环结束时才行走到大约 $(7.4, 4.4)$ 处, 不能有效进入理想构型附近。此时的运动更加接近的是Brown运动, 带有随机游走的特征。我们可以粗略地估计一下, 如果是理想的布朗运动, $\Delta x \approx \sqrt{N}A \approx 1$, 这和(a)图吻合得很好。而大步长下的轨迹显然比“最佳步长”的要粗糙很多, 毛刺更明显, 在平衡位置附近涨落更大, 也不能得到理想的结果。

6 总结

- Metropolis抽样是模拟各种系综的一个简单直观的抽样方法, 有很大的实用性。
- Metropolis抽样的缺点也很明显, 非常依赖于步长(步进频率分布)的选取, 有一定的预实验或者计算前的估计非常重要。

计算物理A第十七次作业

王铠泽 PB18020766

1 作业题目

- 以 $x_{n+1} = \lambda \sin(\pi x_n)$ 为迭代方程进行迭代:

- (1) 画出系统状态随参数 λ 的变化图, 要求在图中体现出定值状态、倍周期分叉和混沌状态;
- (2) 列出各个倍周期分叉处的 λ 值, 求相应的 Feigenbaum 常数。

2 实现方法和原理

- 稳态、倍周期分岔、混沌

在动力系统迭代方程过程中, 不同 λ 参数选择会导致系统最终处于不同的状态, 比如稳态、周期态、混沌态(周期无穷)。下面是1维简单的 logistic 方程迭代的效果:

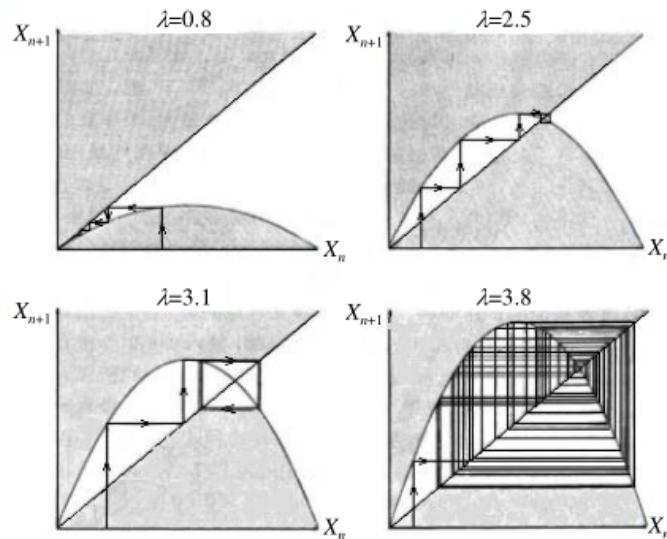


图 2.1 不同参数下的 Logistic 映射过程^[3]

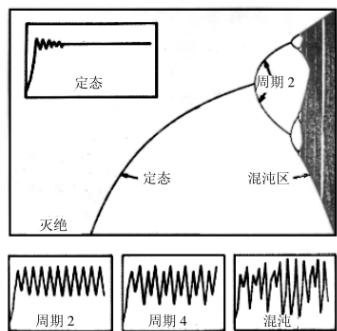


图 2.2 系统状态随参数 λ 的变化^[3]

- *Feigenbaum* 常数

倍周期分岔处的 λ 值满足如下关系：

$$\lim_{m \rightarrow \infty} \frac{\lambda_m - \lambda_{m-1}}{\lambda_{m+1} - \lambda_m} = \delta(\text{const})$$

其中 $\delta \approx 4.6692016091029906718532038$, 称为*Feigenbaum* 常数。另一个简单的等价表述是：

$$\lambda_\infty - \lambda_m = A\delta^{-m} (m \gg 1)$$

- 模拟实现

本次模拟算法很简单，直接利用迭代方程生成一组 $\{x\}$ 序列。在一定的步长 M 之后取出若干 x 作为此时系统的状态点，观察稳态、周期态、混沌态的行为。

关于*Feigenbaum*常数的计算，采用如下模式：

本次实验关注从2周期分岔到128周期分岔的情况。使用一个长度为128数组 $temp$ 来记录从 M 步之后的连续128个 x_k 的值，逐项和 $temp[0]$ 比较，以 cnt 来记录周期计数，将每一个周期分岔处的 λ 值写入文件*Feigenbaum.txt*中。比较麻烦的是对于两个数是否相等的误差判定，随着周期增大，两个不同迭代点之间越来越接近，这使得计算误差和真正的周期性容易混淆，对于周期128的情况尤为明显。

3 程式说明

- *chaos.c*

这是生成每一个 λ 对应系统最终状态的程式。将结果以文件的形式输出， λ 范围和步长可以在程式中更改。

- *Feigenbaum.c*

这是生成不同周期分岔处对应 λ 值的程式，结果以文件形式输出。计算 $\lambda > 0$ 和 $\lambda < 0$ 时步骤有少许不同，可在同一程式中更改(特别是两个数相等误差的判定有一些不同)。

- *lambda(range).txt*

这是不同 λ 范围和步长下的系统终态随 λ 变化的文件。

- *Feigenbaum+/-txt*

这分别对应 $\lambda > 0$ 和 $\lambda < 0$ 时周期分岔处的 λ 计算结果文件。

- 其他说明

本次程式中最终决定的总迭代步数为每个 x 迭代1000次，前100次舍去， λ 的步进步长取其取值范围的 $\frac{1}{1000000}$ ，运行时间可能较长，请耐心等待。或者自行改变步长。

4 计算结果

4.1 系统状态变化图

选取不同的 λ 范围得到系统状态图，必要的时候缩小 λ 范围得到更精细的结构。

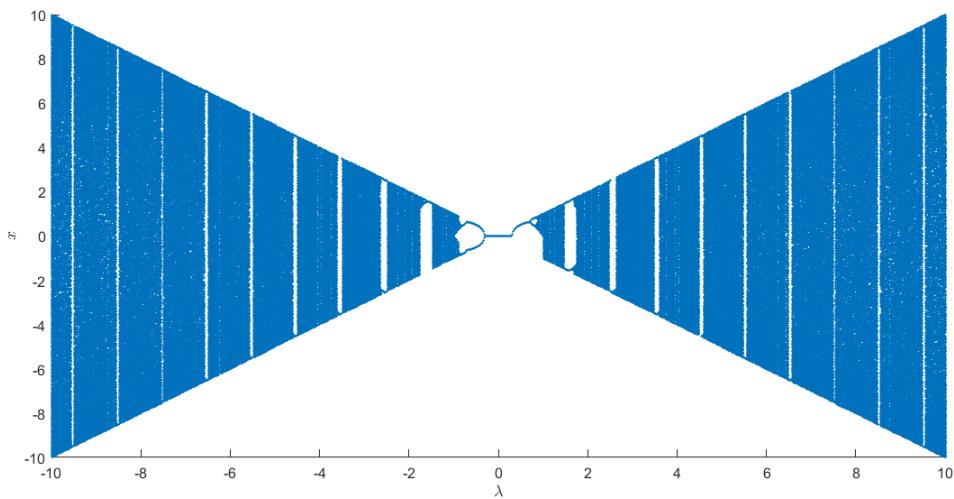
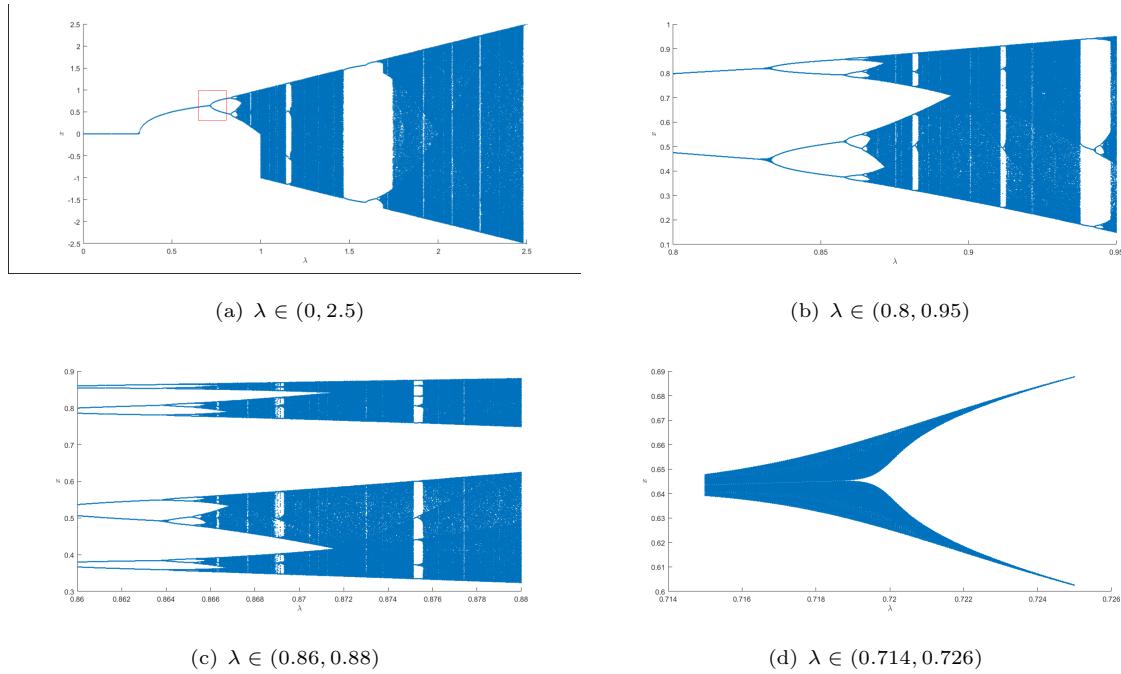


图 1: $\lambda \in (-10, 10)$

在比较大的尺度上，很明显地可以看到绝灭的稳态，呈现周期的态，以及大片大片的混沌态，还有混沌之中出现的周期窗口。

有趣的是无论 $\lambda > 0$ 还是 $\lambda < 0$ 都能看到相似的现象，不同的仅在开始的时候正值 $\lambda \approx 1$ 处似乎缺了一块状态。下面给出几张更加精细的图片。

图 2: 不同范围下的图像 $\lambda > 0$

值得注意的是(a)图中红框内部分放大后具有(d)图精细结构，不是简单的二值结构（这并非是计算误差导致的，(d)图已经将步长缩短10倍以上得到）。

更好的说明上述现象，可以看下图的结构，将 $0.8 \sim 0.95$ 的图放大：

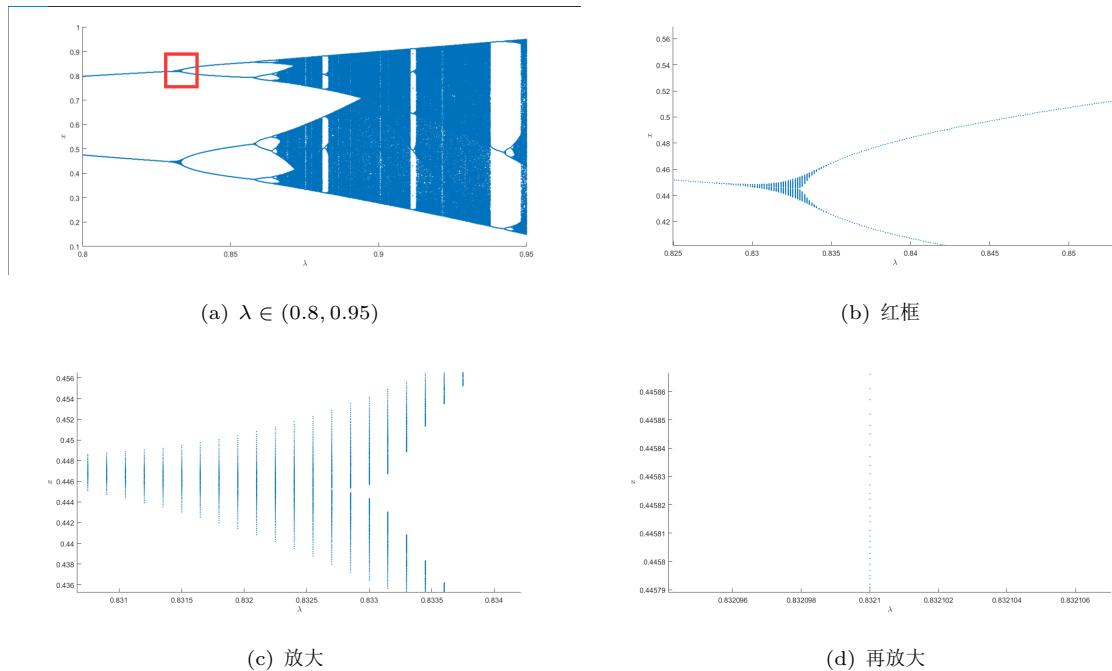


图 3: 分岔结构

可见这种分岔结构是一系列密集的点。初步怀疑是计算精度的问题，但是仔细考虑这个分岔结构的宽度

大概是有0.1的量级，不应该是由于计算误差导致的，可能是迭代方程本身带来的问题。

对于 $\lambda < 0$ ，同样可以给出不同范围的放大图像：

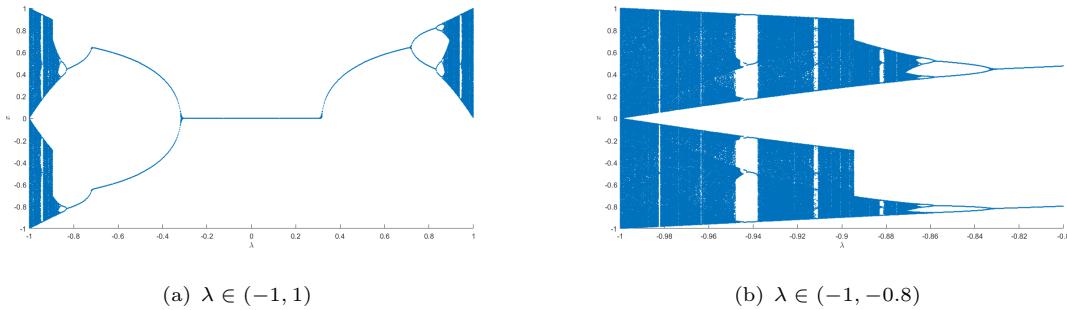


图 4: 分岔结构

4.2 Feigenbaum常数的计算

根据程式得到结果列表如下：

周期分岔	λ_m	$\frac{\lambda_m - \lambda_{m-1}}{\lambda_{m+1} - \lambda_m}$
1→2	0.7198504000	
2→4	0.8338179200	4.5902
4→8	0.8586462400	4.4857
8→16	0.8641812800	4.9941
16→32	0.8652896000	4.7510
32→64	0.8655228800	3.8984
64→128	0.8655827200	

表 1: $\lambda > 0$ case

周期分岔	λ_m	$\frac{\lambda_m - \lambda_{m-1}}{\lambda_{m+1} - \lambda_m}$
1→2	-0.3025601200	
2→4	-0.8302080800	18.5542
4→8	-0.8586462400	5.1378
8→16	-0.8641812800	4.9941
16→32	-0.8652896000	4.7510
32→64	-0.8655228800	3.8984
64→128	-0.8655827200	

表 2: $\lambda < 0$ case

可见 $\lambda > 0$ 和 $\lambda < 0$ 计算出来的Feigenbaum常数计算都不是非常准确（标为黑色加粗）。原因有很多，一个是之前提到的判定两数相等和周期分岔造成的麻烦，可以看到最后的 λ 相差应该只有 10^{-6} 量级。另

一个是在初始多少步长才判定 x 迭代稳定，太小了不精确，太大了计算时间长，本实验采取1000作为初始步长。

粗略地取平均值进行估计，Feigenbaum常数为 $4.5439(\lambda > 0)$ 和 $4.6953(\lambda < 0)$ 。和准确值 $\delta=4.669\dots$ 仍有差距。并且我们能观察到其实算出来的 $\Delta\lambda$ 比例对于 $\lambda > 0$ 和 $\lambda < 0$ 情况下出现很多一致，除了第一个点，这和我们之前对于图像的观察是一致的。两者的区别仅在于最开始缺了一块与否，这就是决定了第一个 λ 差别的地方，其他的值都是相反数，这从迭代方程就能很明显地看出，由于 \sin 是奇函数，所以 λ 基本上不会改变迭代方程的性态。

接下来继续计算 α 的值，根据定义：

$$\alpha = \lim_{m \rightarrow \infty} \frac{d_m}{d_{m+1}}$$

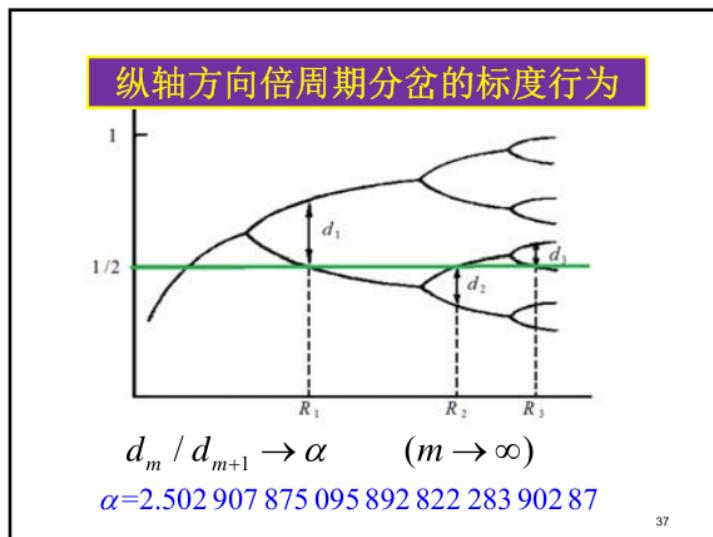
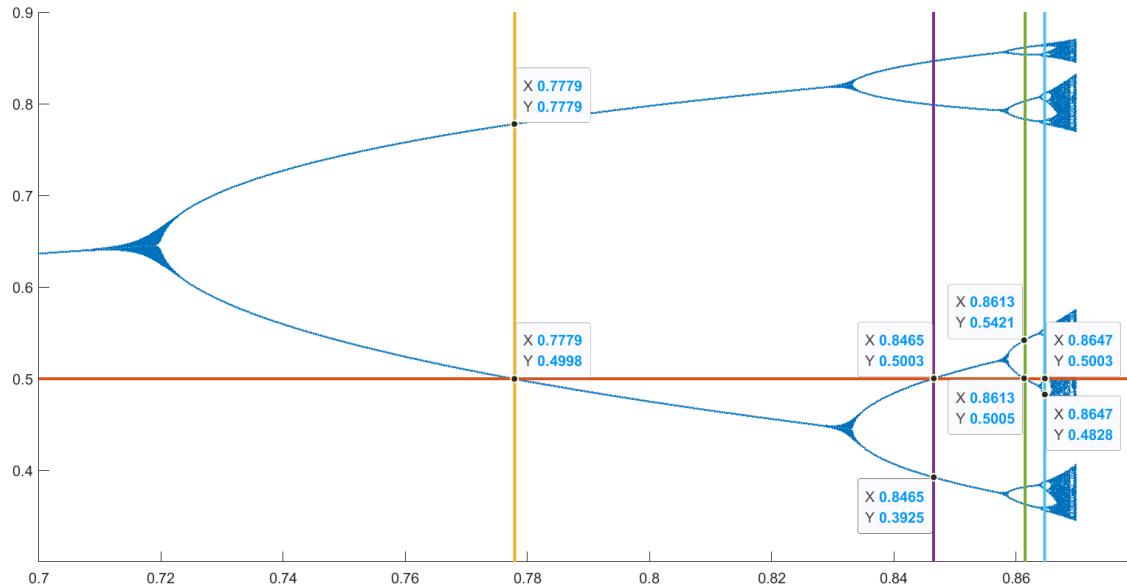


图 5: Feigenbaum α

计算得到的分叉图片如下，选取纵坐标都为0.5:

图 6: *Feigenbaum* α

周期分岔	d_m	$\frac{d_m}{d_{m+1}}$
1→2	0.2781	
2→4	0.1087	2.5584
4→8	0.0416	2.6123
8→16	0.0175	2.3771

表 3: $\lambda < 0$ case

α 计算值和准确值2.502有一定偏差，但仍算准确，可能的原因在于没有缩短 λ 步长，进行更为精确的计算。

5 总结

- 迭代与混沌是物理中有趣的现象，本质上是微分方程的非线性性带来的效应。本次实验通过具体的模拟计算得到了从稳态向周期态和混沌态的过渡模拟，计算了*Feigenbaum*常数，加深了对这一现象的理解。

计算物理A第十八次作业

王铠泽 PB18020766

1 作业题目

- 进行单中心DLA模型的模拟(可以用圆形边界，也可以用正方形边界)，并用两种方法计算模拟得到的DLA图形的分形维数，求分形维数时需要作出双对数图。

2 实现方法

- DLA生长模拟

DLA的想法是存在若干生长中心，外界的粒子从边界上生成并且进行随机游走，若触碰到生长中心，则停下游走形成团簇，后续的粒子若碰到团簇就被粘住，形成一个生长的图案。

本次实验中采取的算法如下：

建立 $(N + 1) \times (N + 1)$ 背景网格，使用二维数组 $cnt[N + 1][N + 1]$ 来记录生长情况。生长中心的对称中心定在 $cnt[N/2][N/2]$ 。程序初始化时， cnt 数组全部置零（表示未生长），然后给一个 $(2d + 1) \times (2d + 1)$ 大小的生长中心赋值为 1，表示此处是生长的核。接下来，不断随机地从边界生成粒子，让粒子随机游走，判断是否游走出边界，若走出边界，则开始新粒子的进入，若下一步就会和值为 1 的矩阵元素重合，则此处也将矩阵元素从 0 变为 1，表示粒子生长到此处。调节模拟中投入的粒子总数 NP 得到不同尺寸的 DLA 图案。本次实验设置 $N = 400, d = 1, NP = 10^6$ ，运行时长较长。

- sandbox计数法

sandbox 法是一种很好的计算单中心生长的分形维数的方法。其原理是计算一系列边长为 r 的盒子中的生长粒子个数 N 。理论上，对于分形图形，有 $N(r) \sim r^D$ 。做出双对数图，可以通过直线斜率求得分形维数 D 。

本次实验中采取的算法如下：

在得到生长矩阵 cnt 之后，采用从中心 $cnt[N/2][N/2]$ 开始往外计数的方法，以步长为 1 开始计数并将每一步对应的数据输入 txt 文件，直到将包裹这个图形的最小方形给计数完毕就退出循环。

- 回转半径法(R_g 法)

回转半径法也是一种计算自相似图形分形维数的方法。计算回转半径公式如下：

$$R_g^2 = \frac{\sum_{i=1}^N r_i^2}{N}$$

根据计数的个数（面积） N 和 R_g 满足 $N \sim R_g^D$ 来得到分形维数。

本次实验中采取的算法如下：

在得到生长矩阵之后，开始和 sandbox 法一样从中心开始一层一层计数，计数得到的个数为 N ，同时在计数过程中可以计算得到不同 N 下的回转半径 R_g 。最后将各个数据输出到 txt 文件中。值得一提的是为了避免生长的边缘效应造成的误差，我将输出控制在计数的粒子数为总生长的 0.7。

3 程式说明

- DLA.c

这是一个用于生成 DLA 点阵上的生长与否信息的文件的程式。同时具有输出两种计算分形维数方法相关的数据的功能。

- rdm.h

这是一个包含了使用16807产生器生成指定长度的 $[0, 1]$ 上均匀分布随机数函数的头文件。

```
double rdm()
```

该函数输出值为 $(0,1)$ 上的随机数，生成的方法是线性同余法，具体是16807生成器实现。使用的初始种子为 $I=46984$ ，然后每一次能采用静态变量的方法记录 I_n 的值，实现每一次输出值都不同的效果。

- size_(N+1)_(# of particles).txt

这是输出背景矩阵为 $(N+1) \times (N+1)$ 维，总共生长粒子数为(# of particles)的数据txt文件。

- sandbox.txt

这是输出盒子边长 r （第一列）和像素计数 N 的数据文件。

- Rg.txt

这是输出面积（第一列）和回转半径 R_g 的数据文件。

4 计算结果

4.1 DLA生长图示

采取背景网格为 401×401 ，不同生长阶段图解如下：

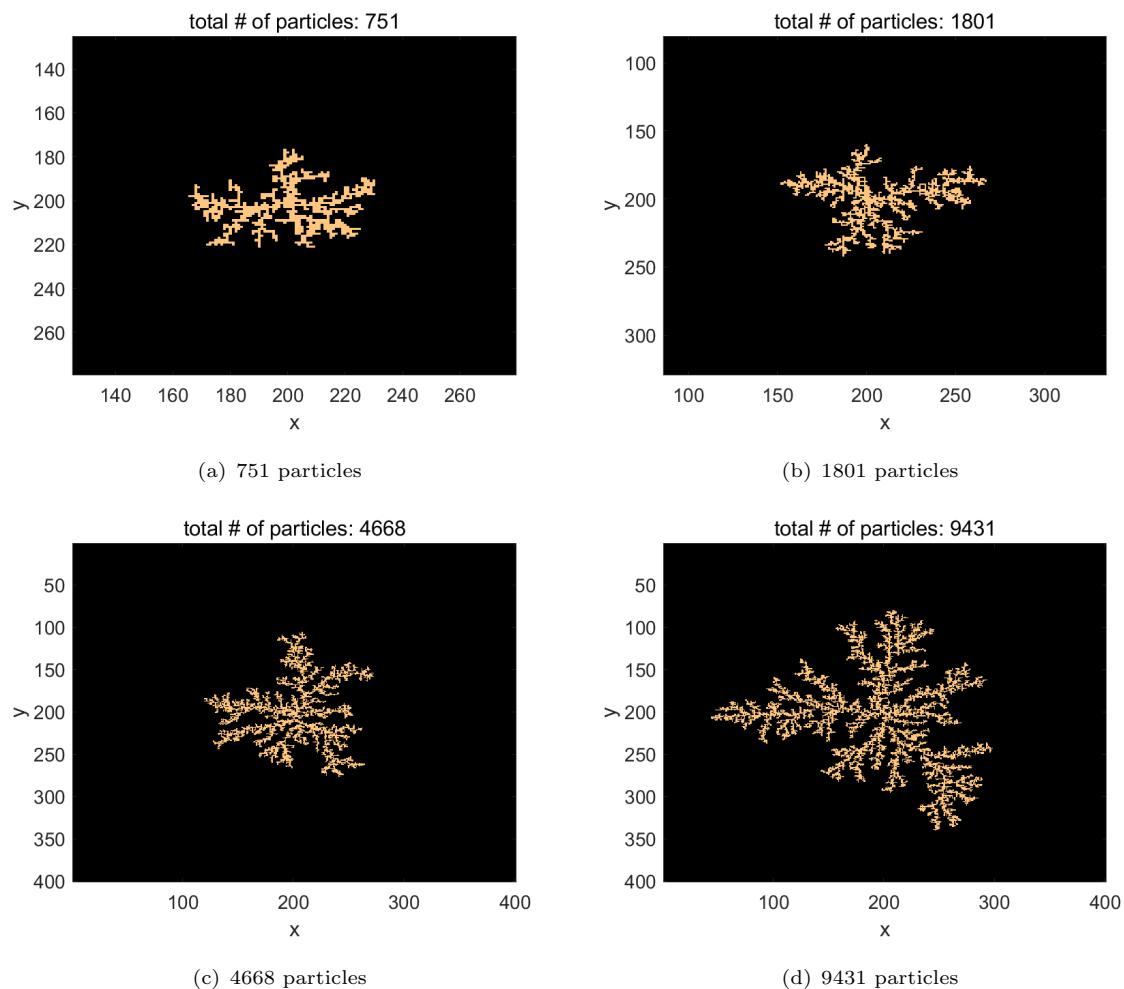


图 1: 不同阶段的DLA生长

可以看出粒子清晰地体现出延展方向，呈支状生长。

笔者还尝试了二中心的生长情况：

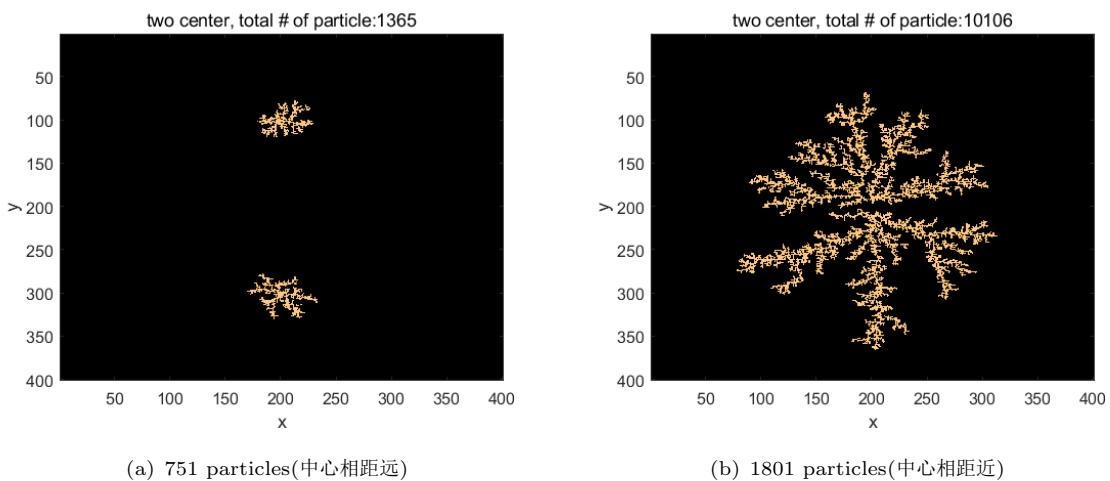


图 2: 不同阶段的双中心DLA生长

有趣的是发生了相规避生长的情况，两个团簇不会融合成一个，而是各自占据一半地盘向外延伸。

4.2 分形维数计算

4.2.1 sandbox法

先对上述的单中心生长进行计算，采用sandbox法得到结果如下：

$$D = 1.679$$

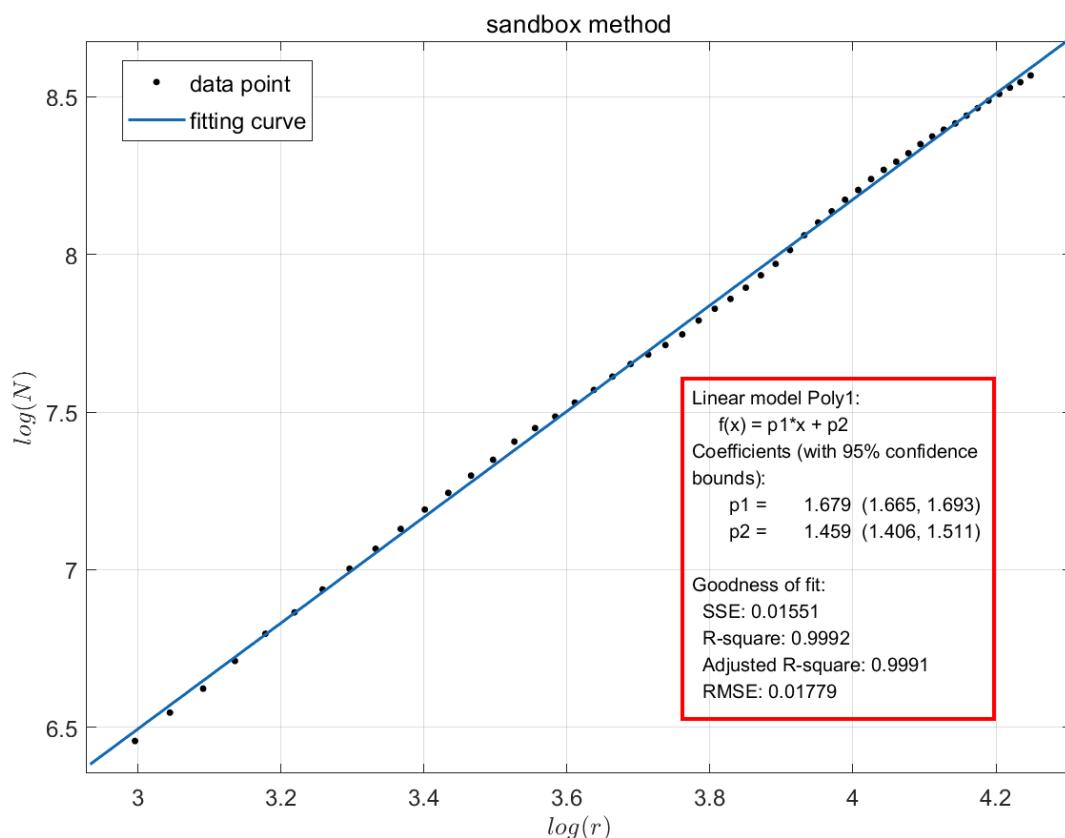


图 3: sandbox method

理论上DLA生长的分维数为1.60~1.70，我们的计算和这个符合。

接着对双中心(图2 (b))进行分维数计算：

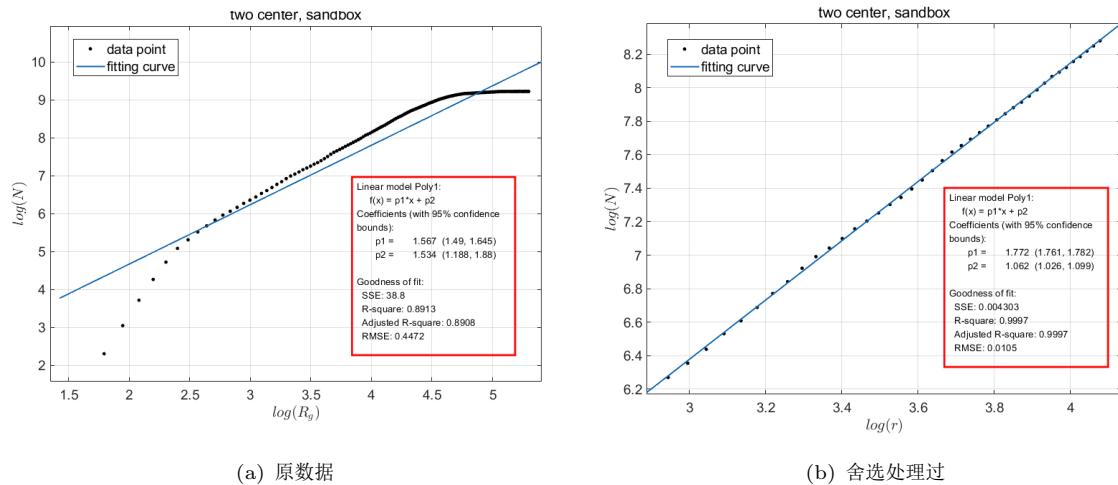


图 4: 双中心的分维数计算

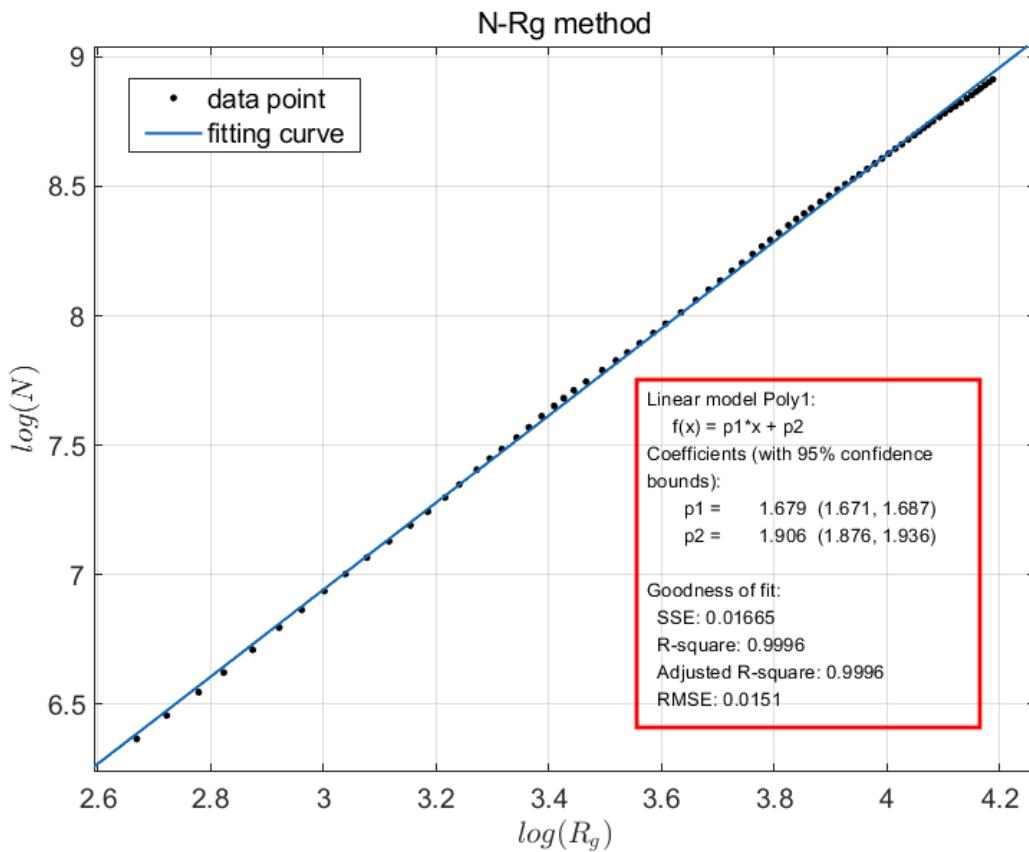
我此时选择的双中心分别为 $cnt[N/2 - 10][N/2]$ 和 $cnt[N/2 + 10][N/2]$, 所以仍然从 $cnt[N/2][N/2]$ 开始sandbox计数仍然是合理的。只是由于不再是单中心了, 线性区域变得更窄了, 经过挑选线性区域的图(b)得到的分维数为1.772, 略大于理论值, 这也一定程度上体现出sandbox法比较适合单中心生长图形。

4.2.2 回转半径法

先对上述的单中心生长进行计算, 采用回转半径法得到结果如下:

$$D = 1.679$$

这得到的分维数和sandbox法一样, 所以说明这个图形分维数约为1.679。

图 5: R_g method

接着对双中心(图2 (b))进行分维数计算:

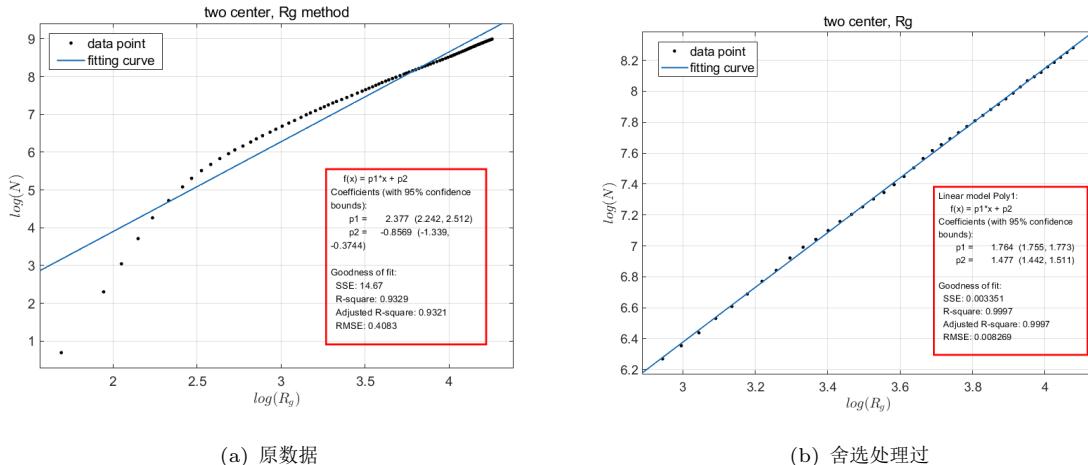


图 6: 双中心的分维数计算

和sandbox法出现类似的偏离线性情形, 原因可能是笔者在计算的时候吧($N/2, N/2$)当成生长中心计算, 实际上在多中心时采取下列的公式会更加准确一点:

$$R_g^2 = \frac{\sum_{i=1}^N (r_i - \bar{r})^2}{2N(N-1)}$$

此处计算得到的分维数为：

$$D = 1.764$$

略大于理论值范围。

5 总结

- 扩散限制凝聚是一个模拟薄膜生长，自然沉积过程的很好的模型，具有丰富的物理意义。本次实验通过二维DLA的模拟，得到了其分维数和基本图形性质，加深了理解。
- 本次实验采用固定边界，所以最开始生长过程粒子游走到触碰团簇的概率小，需要投入很多粒子，等待时间长。一个比较好的改进方法是将固定边界变成不断变化的生长边界，始终保持比最大团簇尺寸大上数圈，然后以此为边界生长。

计算物理A第十九次作业

王铠泽 PB18020766

1 作业题目

- 用 Numerov 法求解一维定态薛定谔方程在一个对称势阱（势能函数 $V(x)$ 可任意设置）中的基态和激发态的能量本征值。画出能量本征值及其附近的波函数。

$$\left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right) \psi(x) = E\psi(x)$$

2 实现方法和原理

- Numerov 方法

对于如下形式的微分方程：

$$\phi''(x) = F(\phi, x) = f(x)\phi(x)$$

做变量代换：

$$y(x) = \left[1 - \frac{h^2 f(x)}{12} \right] \phi(x)$$

得到如下迭代格式：

$$\Rightarrow y_{n+1} = 2y_n - y_{n-1} + h^2 f_n \phi_n + O(h^6)$$

$$\Rightarrow \phi_{n+1} = \frac{2y_n - y_{n-1} + h^2 f_n \phi_n}{1 - \frac{h^2 f_n}{12}}$$

- 打靶法

在求解边值问题时，常用的方法是先给出预估的能量 E 值，然后根据参数计算的结果进行调整。

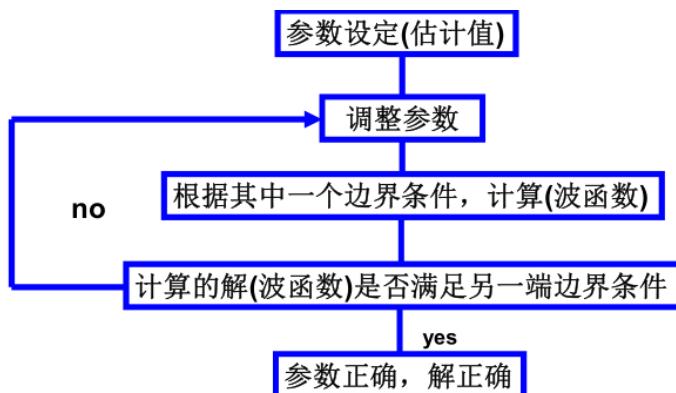


图 1：打靶法算法描述

在本次实验中，由于理论上边界是无穷大处为零，所以取足够大的 x_0 作为截断，边界条件变为 $\psi(\pm x_0) = 0$ 。本次实验中采取的对称势为谐振子势能加上四阶项微扰：

$$V(x) = \frac{1}{4}x^2 + \epsilon x^4 (\epsilon \ll 1)$$

薛定谔方程简化为：

$$\left(-\frac{\partial^2}{\partial x^2} + \frac{1}{4}x^2 + \epsilon x^4 \right) \psi(x) = E\psi(x)$$

对于基态和第一激发态的能量预估值 E_{G0}, E_{G1} 分别以理想无微扰的谐振子势能计算得到：

$$E_{G0} = 0.5, E_{G1} = 1.5$$

其零阶近似解为：

$$\psi_0^{(0)}(x) = \left(\frac{1}{2\pi} \right)^{1/4} \exp\left(-\frac{x^2}{4}\right)$$

$$\psi_1^{(0)}(x) = \left(\frac{1}{2\pi} \right)^{1/4} x \exp\left(-\frac{x^2}{4}\right)$$

可以取特征截断长度为 $x_0 = 6$ 。本次实验采取 $\epsilon = \frac{1}{1000}, \frac{2}{1000}, \frac{3}{1000}$, $h = 1.2 \times 10^{-3}$ ，参数每一次初始变化步长 $\Delta E = 0.01$ ，此后作为可调节变量不断缩小，当上一个循环的 $\psi(x_0)$ 和本循环的 $\psi(x_0)$ 异号时， $\Delta E = \Delta E/10$ 。

$$f(x) = \frac{1}{4}x^2 + \epsilon x^4 - E_G$$

• 理论结果

理论上能量的一阶微扰为：

$$\delta E = \langle \psi^{(0)} | \epsilon x^4 | \psi^{(0)} \rangle$$

$$E_0^{(1)} = E_0^{(0)} + \delta E_0 = 0.5 + 3\epsilon$$

$$E_1^{(1)} = E_1^{(0)} + \delta E_0 = 1.5 + 15\epsilon$$

当 $\epsilon = \frac{1}{100}$ 时还是比较接近微扰的，具体见下面两个势能项的对比：

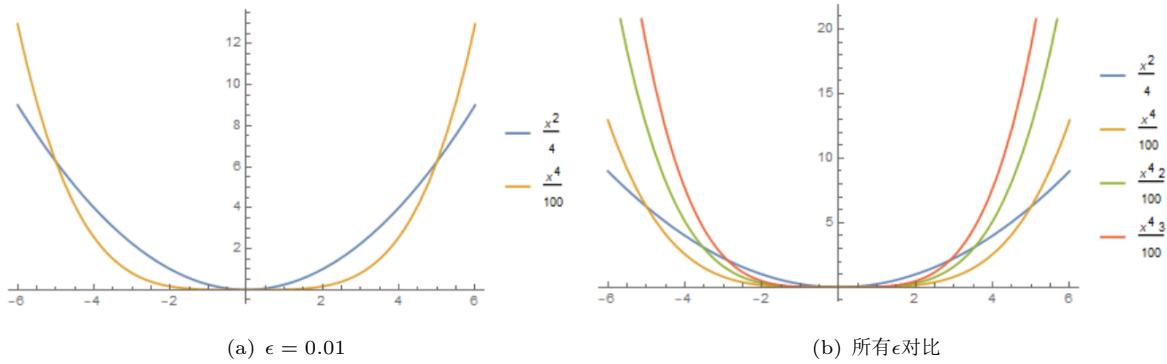


图 2: 势能曲线对比

当 $\epsilon = 0.01$, 四次方项在 $x = 4$ 处和二次方项以及能相比大小了, 此时大约是 2σ , 还尚能算做微扰, 其余的两个微扰处理出来的理论值应该误差会比较大, 这可以参见后续的计算结果讨论。

3 程式说明

- Numerov.c

这是一个能用于计算对称势能下的定态薛定谔方程能量本征值和波函数的程式, 结果以文件形式输出。运行结束时会有边界上 $\psi(x_0)$ 大小和能量本征值的提示。

- ground_state_1/2/3.txt

这三个文件分别对应着 $\epsilon = 0.01, 0.02, 0.03$ 的基态能量和波函数输出。左一列是位置 x , 右一列是波函数的值 $\psi(x)$ 。

- 1st_excited_state_1/2/3.txt

这三个文件分别对应着 $\epsilon = 0.01, 0.02, 0.03$ 的第一激发态能量和波函数输出。左一列是位置 x , 右一列是波函数的值 $\psi(x)$ 。

- 2nd_excited_state.txt

这三个文件分别对应着 $\epsilon = 0.03$ 的第二激发态能量和波函数输出。左一列是位置 x , 右一列是波函数的值 $\psi(x)$ 。

4 计算结果

4.1 能量计算

采用Numerov方法和打靶法得到的能量本征值如下:

ε	E_0	E_1	E_2
0.00	0.500000	1.500000	2.500000
0.01	0.526734	1.626748	—
0.02	0.549072	1.725986	—
0.03	0.568652	1.810017	3.228912

表 1: 计算结果

其中 $\epsilon = 0.00$ 是未加微扰的理想结果，不是计算得到的。

在一阶微扰近似下：

ε	E_0	E_1	E_2	$E_0^{(1)}$	$E_1^{(1)}$	$E_2^{(1)}$	$err_0(\%)$	$err_1(\%)$	$err_2(\%)$
0.01	0.526734	1.626748	—	0.53	1.65	—	0.62	1.41	—
0.02	0.549072	1.725986	—	0.56	1.80	—	2.13	4.11	—
0.03	0.568652	1.810017	3.228912	0.59	1.95	3.67	3.62	7.12	12.02

表 2: 和理想的非简并一阶微扰的比较

可以看出， $\varepsilon = 0.01$ 和微扰计算较为一致，而且随着能级升高，越来越偏离微扰计算。另一方面一阶微扰所计算得到的能量都偏高，这和我们在量子力学课上学习得到的结果是相互一致的。

还有值得一提的是 ε 增大时会越来越偏离微扰计算的范畴，当 $\varepsilon = 0.03$ 时已经出现很大的偏差，此时当作微扰计算很可能会有问题。笔者尝试计算这种情况下的二阶微扰能量，发现并不收敛。

4.2 波函数计算

采用Numerov方法计算得到的波函数如下：

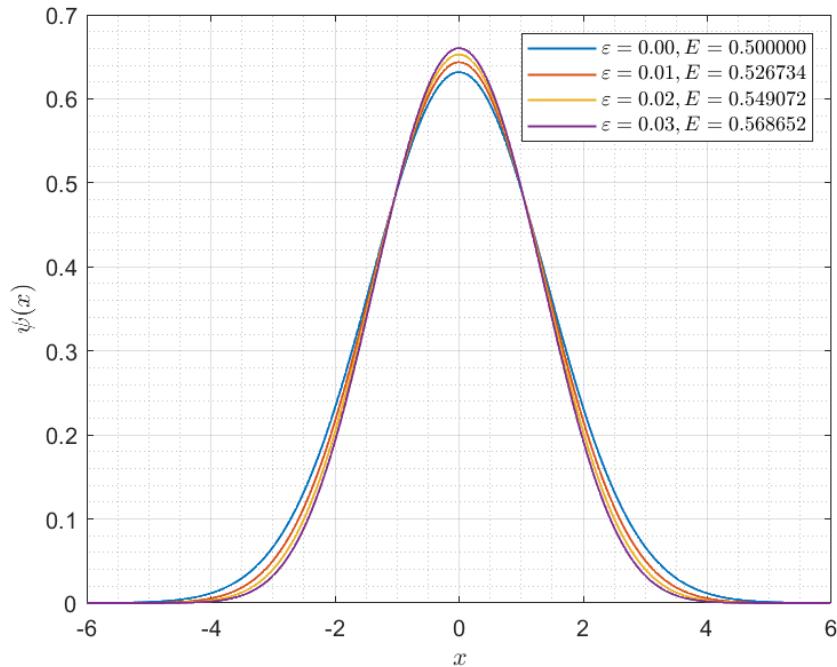
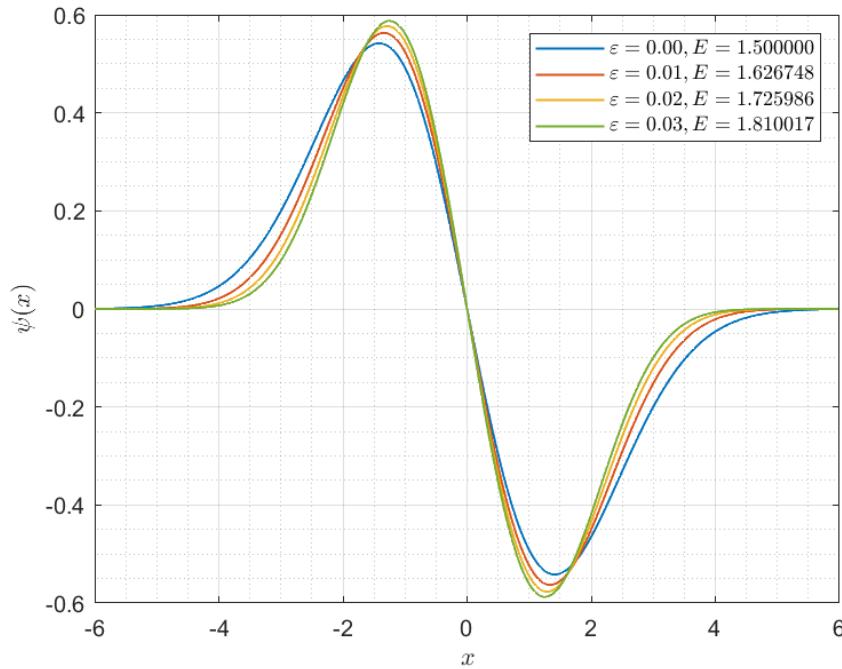
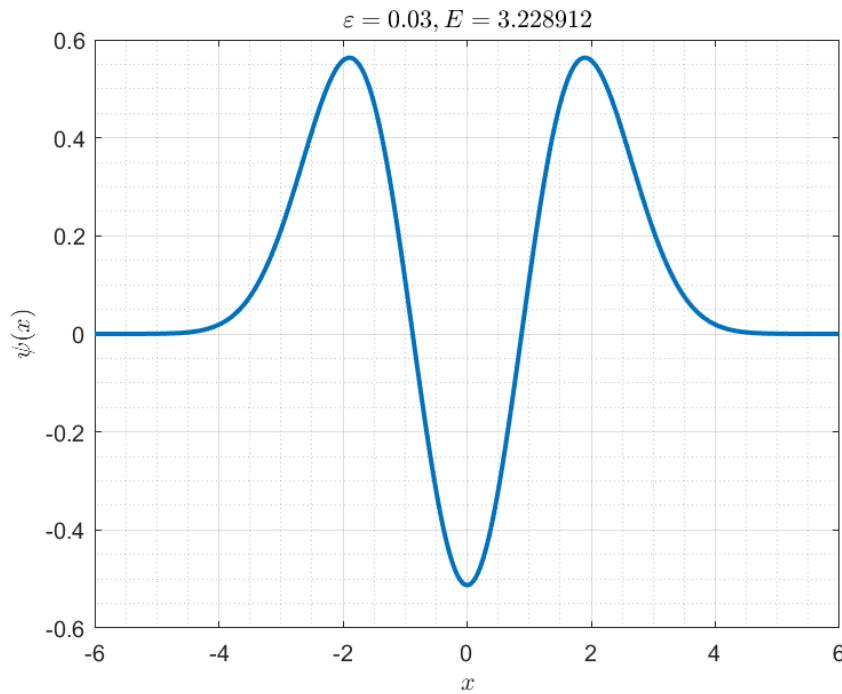


图 3: 不同 ε 下的基态波函数

可以看出随着 ε , 即微扰的增大, 波函数趋于更加局域化, 更加集中在 $x = 0$ 的附近, 展宽更小。这也是符合物理图像的, 当加入四阶势能, 等于再引入一个三次方正比的吸引力, 使得粒子运动更加趋于局域化。

第一, 第二激发态的波函数如下:

图 4: 不同 ε 下的第一激发态波函数图 5: $\varepsilon = 0.03$ 下的第二激发态波函数

激发态的波函数也体现出和基态类似的局域化的特征，符合物理直觉。

5 总结

- Numerov法和打靶法是解决边值问题，特别是薛定谔方程的基本方法，能够得到比较理想的结果。
- 打靶法的能量改变步长应该也要随着程式进行来调节，这样才能得到正确的能量本征值，不然可能会把第一激发态能计算成更高激发态能量，这是由于步长过长导致的。
- 打靶法的运行效率较低，应该有更加优化的数值方法。