

计算物理homework7

李明达 PB18020616^{1*}

摘要

这是计算物理第七题作业，作业题目是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。

关键词

直接抽样法，舍选抽样法

¹中国科学技术大学物理学院

*作者: dslmd@mail.ustc.edu.cn

这篇报告里，我首先阐述两种抽样的算法，介绍我所采用的原理，然后给出我自己的实验结果，并与原曲线对比验证，其次，我关于舍选法抽样效率进行讨论。最后，我对这篇实验进行总结。

我们先来看一下”data.txt”到底是一个什么样的图，如图1所示：

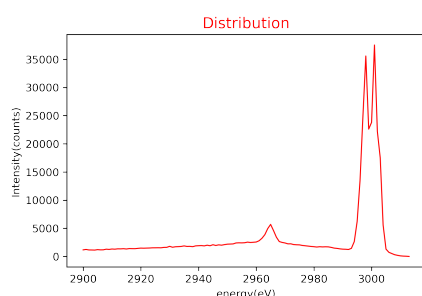


图 1. 原题中给出的分布图

通过我在C语言代码里写的read_data()函数我可以读出，总数据点的个数是396002个。我们后续的抽样的数目，会以这个为其中一个标准。

1. 算法以及公式

1.1 直接抽样

原则上我们可以用插值的办法处理数据，然后对连续函数进行抽样。但是为了更加直接，在此次实验我并没有处理离散的数据点，我直接采用离散抽样。离散抽样的算法是： if

$$\sum_{k=0}^{i-1} p[k] < r < \sum_{k=0}^i p[k]$$

then

$$x \rightarrow x[i]$$

即，如果随机数落在 $i-1$ 和 i 的分布函数之间，则这时候我认为抽样落在 $x[i]$ 点。于是该点

的sampling_number加一。最后sampling_number反映了某一个 x 处总共的随机数点数。

在这个实验中，我定义了read_data()来读取data.txt文件中的数据(注意，开头已被删除)，然后通过direct_sampling(int seed, int N)函数直接抽样N次并存储到抽样结果”direct_sampling.txt”，最后通过Python画图。

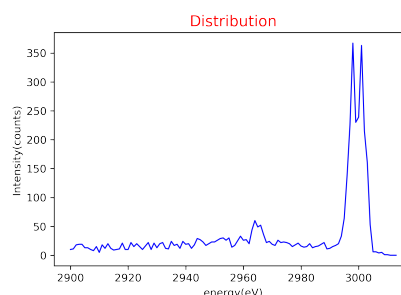


图 2. 直接抽样的结果（按照原文件3960个点）

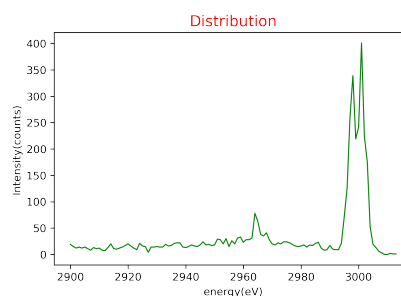


图 3. 舍选抽样的结果（按照原文件3960个点）

1.2 舍选法抽样及其效率

舍选法我同样采取离散抽样的办法，在 $x[i] < rand1 < x[i+1]$ 的范围内，如果 $rand2 > p[i]$ ，则舍去该点；相反，如果 $rand2 < p[i]$ ，则把这个点留下，于是该点的sampling_number加一。最

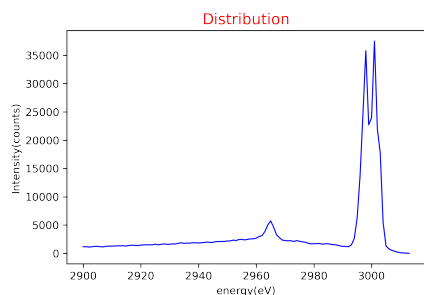


图 4. 直接抽样的结果（按照原文件396002个点）

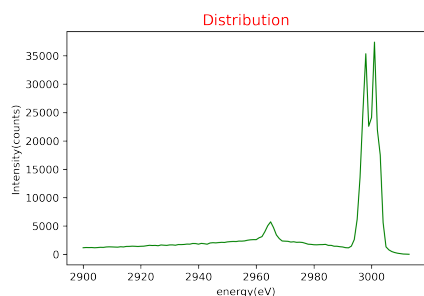


图 5. 舍选抽样的结果（按照原文件396002个点）

后sampling_number反映了某一个x处总共的随机数点数。但是为了更高的抽样效率，并且看到在 $x < 2990$ 范围内，最大值是5672，在 $x > 2990$ 的范围内，最大值是35560。我取了一个分段函数 $F(x)$ 来对概率抽样（所以要除以归一化因子396002）

$$F^x = \begin{cases} 5673/396002 & 2900 \leq x < 2990 \\ 38000/396002 & 2990 < x \leq 3013 \end{cases} \quad (1)$$

由离散的估算，该抽样效率的理论值应该在0.28左右。

在这个实验中，我定义了shexuan_sampling(int seed, int N)直接抽样N次并存到抽样结果”shexuan_sampling.txt”，最后通过Python画图。该函数还可以printf出此次抽样的效率，我们采取 $N=3960$ 和 $N=396002$ 来做这次实验。

2. 实验结果

2.1 直接抽样与舍选抽样法的结果

下面是我用直接抽样（图2，图4）和舍选抽样（图3，图5）分别画出的图。其中我总共分别取了3960个点和396002个点。

2.2 将上述结果与原曲线进行对比

下面我把直接抽样和舍选抽样的结果，与原曲线画到同一个图里，进行比较。直接抽样法（图6，图8），舍选抽样法（图7，图9）。

其中在这些实验图里我总共分别取了3960个点和396002个点。

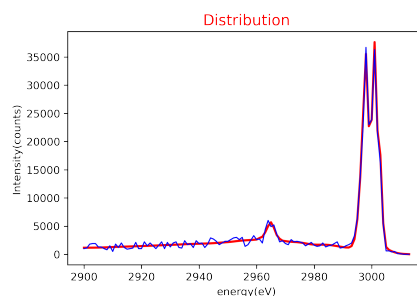


图 6. 直接抽样与原曲线对比的结果（按照原文件3960个点，但为了和原数据对比，我将图放大了100倍）

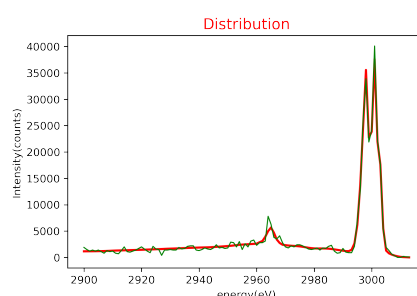


图 7. 舍选抽样与原曲线对比的结果（按照原文件3960个点，但为了和原数据对比，我将图放大了100倍）

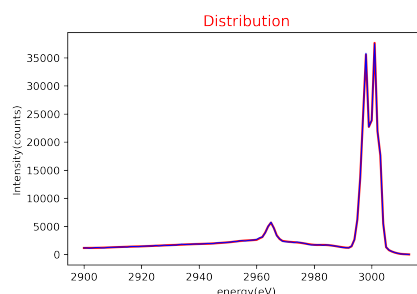


图 8. 直接抽样与原曲线对比的结果（按照原文件396002个点）

可以看出，当N足够大的时候（ $N=396002$ ），两种抽样的抽样结果都与原数据完美符合！

2.3 舍选法抽样效率的讨论

其中我总共分别取了3960个点和396002个点，计算了舍选法的效率，如图10和图11所示。

可以看出，这两次抽样的抽样效率稳定在0.278左右，这也与理论值0.28符合。

3. 总结

本次实验总体原理比较简单，也就是直接抽样

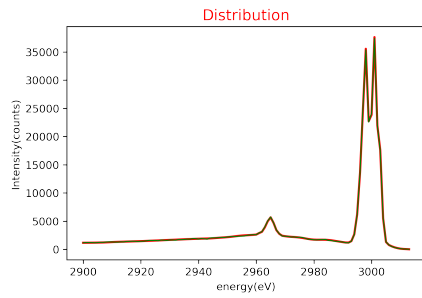


图 9. 舍选抽样与原曲线对比的结果（按照原文件396002个点）

```
PS C:\Users\dela\OneDrive\计算物理\04\蒙特卡罗\04\舍选抽样> cd "C:\Users\dela\OneDrive\计算物理\04\蒙特卡罗" ; if ($?) { gcc 第七题代码.c -o 第七题代码.exe }
total: the random number generator ran 14453 times, the efficient is 0.224026
```

图 10. 3960个点下，舍选抽样的效率

和在一个特定函数 $F(x)$ 下进行舍选抽样，但是画图和报告用了比较久的时间，一是图片数目较多，另一个是我想给读者展现一个比较完整的思路。总体来说，我达到了预期的效果，在抽样数目足够多的时候，直接抽样和舍选抽样与原数据图符合的非常好，而且在 N 比较大的时候，抽样效率也与理论值完美符合。综上所述，本次实验完美完成！

```
PS C:\Users\dela\OneDrive\计算物理\04\蒙特卡罗> cd "C:\Users\dela\OneDrive\计算物理\04\蒙特卡罗" ; if ($?) { gcc 第七题代码.c -o 第七题代码.exe }
total: the random number generator ran 1438329 times, the efficient is 0.228886
```

图 11. 396002个点下，舍选抽样的效率