概率论与数理统计大作业报告——蒙特卡罗方法总结及应用

江昱峰 21009200038

一、 引言

蒙特卡罗法作为一种计算方法，是由美国数学家乌拉姆(Ulam ， S. M.)与美籍匈牙利数学家冯·诺伊曼(von Neumann,J.)在20世纪40年代中叶，为研制核武器的需要而首先提出来的。它是摩纳哥的著名赌城，该法为表明其随机抽样的本质而命名。实际上，该方法的基本思想早就被统计学家所采用了。例如，早在17世纪，人们就知道了依频数决定概率的方法。[1]

蒙特卡罗方法，对于我来说，早已不陌生。早在初中，我就接触到了它：《新思维》一书中讲到了蒲丰投针来测量π值；到了高中，生物中的标记重捕法也有蒙特卡洛法的影子；而到了大学，数学建模竞赛的方法学习中我又系统学习了它，虽然这种方法显得没那么高深，但有时不失为一种合理有效且简单易行的建模处理手段。

本文就将深钻这种方法，做出总结，应用到具体问题中并研究其准确性、合理性。

二、学习总结

虽然已经学习过，这次我还是又特意从文献、博客随笔的角度入手更深地学习了一遍这种方法，并做了如下的**思维导图**。具体内容图中已经剖析得十分清楚，不再文字赘述。[2][3]

图1：蒙特卡罗方法思维导图总结

蒙特卡罗方法的具体步骤在下文解决具体问题的过程中会给出,此处不再重复。

除了导图中提到的特点，蒙特卡罗方法还有很多优点，比如说能够比较逼真地描述具有随机性质的事物的特点及物理实验过程、对于连续性的问题不必进行离散化处理、受几何条件限制小、收敛速度和误差与问题的维数无关且误差容易确定、具有同时计算多个方案与多个未知量的能力等等。但也有一定的局限，比如对于确定性问题需要转化成随机性问题、收敛速度慢、误差具有概率性、通常需要较多的计算步数N等。[1]

我认为，蒙特卡罗方法就如同“天下武功，唯快不破”中的“快”，虽是大道至简，却胜过十八般兵器；就好比爱迪生让助手测量梨形灯泡容器时直接灌水的锦囊妙计，远胜过正面的复杂甚至无解的计算，欧亨利式的“意料外，情理中”。这就是它独特之处，亦是其胜过其他方法的地方。

三、方法应用与问题求解

蒙特卡罗方法应用之广，甚至就连高等数学里从数学角度难以解决的三重积分，它都可以轻松解决。下面，我将应用蒙特卡罗法来计算算式：

$$I = \iiint_{\Omega} (x + e^{y} + \sin z)\ dx\,dy\,dz \tag{1}$$

其中 Ω 由旋转面$z=8-x^2-y^2$, 圆柱面$x^2+y^2=4$和平面$z=0$围成。

所查文献中计算方法大都是选取一个包含不规则空间∑的规则空间（如立方体A），随机取大量样本点，数量为N，然后统计在∑内的样本点的数量n和这些样本点的 $f(x,y,z)$值的均值m，设空间A体积为V，则结果Ans为：

$$Ans = V * (n/N) * m \tag{2}$$

但个人认为，这样的计算方式中隐含了用蒙特卡罗方法去求解∑的体积，而这一步也会产生一定的误差。由于此问题中 Ω 的体积可以直接通过高等数学求解出无误差的理论值，所以我设计了更为准确的方法，大致算法如下：

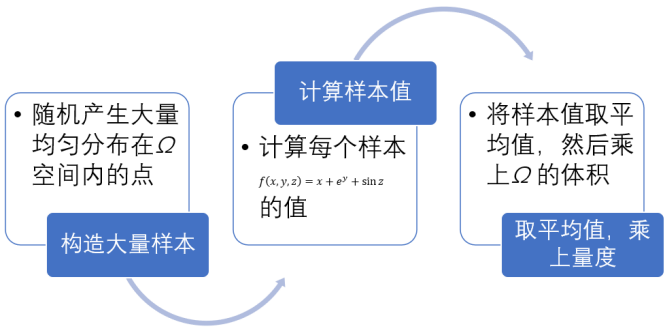

图2：蒙特卡罗**算法流程**图

此处蒙特卡罗算法的理论基础是概率统计中的**大数定律**，具体、确切来说是**辛钦大数定律**，即：

设$\{a_i,\ i \geq 1\}$为独立同分布的随机变量序列，若a_i的数学期望存在，则服从大数定律, 即对任意的 ε >0，有公式：

$$\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} a_i - \mu\right| < \varepsilon\right) = 1 \tag{3}$$

由此，我们可以看出，当样本数量n充分大时，样本的平均值就无限趋近于期望。在这个问题中期望的具体含义就是空间 Ω 中表达式

$$f(x,y,z)=x+e^{y}+\sin z \tag{4}$$

的平均值。然后只要将平均值乘上体积就是所求值。

具体的求解过程，本文用Python实现。其中**随机数的生成原理**就是x、y、z依次分别在它们的定义域中通过**Python的random库中的uniform函数**随机生成一个值，由此构成空间 Ω 中一个随机点的坐标。

此处需要说明的一点是，编程语言中的rand函数生成的随机数普遍是**伪随机数**，即通过某个固定的方法而不是完全随机的方法得到随机数。这样得到的伪随机数序列有抱团现象，分布不均匀。因此如果采用如**Halton序列**等更为随机的方法得到更均匀的随机数，结果会更为准确。但这种更准确的方法往往在拟蒙特卡洛法中采用，而本篇文章展示的是用蒙特卡罗方法求解的过程，所以随机数的产生依然用的是程序中的随机数生成函数，**过程代码详见附录**。

体积的计算方法通过高等数学的三重积分内容即可求出，结果为 24π，具体过程不在这里展示。由此，当**样本数量为1000000**时，程序计算得的 I **值为120.53129420066189**。

四、误差分析

当然，样本的数量总是有限的，不能达到真正的无穷，因此误差一直是存在的。**误差具体计算方式**可根据概率统计中的独立同分布的中心极限定理，又称**Lindeberg-Levy定理**：

设随机变量 $X_1, X_2, \cdots\cdots$ 独立同分布，并且具有有限的数学期望和方差：$E(X_i)=\mu$，$D(X_i)=\sigma^2$ $(i=1,2\cdots\cdots)$，则对任意x，分布函数

$$F_n(x)=P\left\{\frac{\sum_{i=1}^{n}X_i-n\mu}{\sigma\sqrt{n}}\le x\right\} \tag{5}$$

满足

$$\lim_{n\to\infty}F_n(x)=\lim_{n\to\infty}P\left\{\frac{\sum_{i=1}^{n}X_i-n\mu}{\sigma\sqrt{n}}\le x\right\}=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-\frac{t^2}{2}}dt=\emptyset(x) \tag{6}$$

当n充分大时，有如下的近似式：

$$\mathrm{P}\left(\left|\overline{\mathrm{X}}_{\mathrm{N}}-E(\mathrm{X})\right|<\frac{\lambda_\alpha\sigma}{\sqrt{\mathrm{N}}}\approx\frac{2}{\sqrt{2\pi}}\int_0^{\lambda_\alpha}\mathrm{e}^{-\mathrm{t}^2/2}\mathrm{dt}=1-\alpha\right) \tag{7}$$

其中 α 称为置信度，$1-\alpha$ 称为置信水平。这表明不等式

$$\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}} \tag{8}$$

近似地以概率1-α成立，且误差收敛速度阶为$O(N^{-1/2})$。通常蒙特卡罗方法的**样本均值的误差 ε 计算公式**为

$$\varepsilon = \frac{\lambda_\alpha \sigma}{\sqrt{N}} \tag{9}$$

上式中λ_α与置信度α是一一对应的，根据问题的要求确定出置信水平后，查标准正态分布表，就可以确定出λ_α。[4]

其中σ的计算公式如下：

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} X_i\right)^2} \tag{10}$$

基于上面的误差分析方法，现在对程序中的计算方法进行误差估计。程序中n值为1000000，σ值为1.9742170684751297，置信度我们取常见的α=0.05，此时查表知$\lambda_\alpha = 1.96$。这样，由式（9）就可得出**样本均值的误差ε为0.0038694654542112544**。

五、参考文献

[1]百度百科，蒙特卡罗法
https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%B3%95/1225057

[2]陈乐.利用蒙特卡洛方法计算一重和二重积分[J].玉林师范学院学报,2013,34(02):22-25.DOI:10.13792/j.cnki.cn45-1300/z.2013.02.004.

[3]知乎作者 少刷知乎多读书，蒙特卡罗方法详解
https://zhuanlan.zhihu.com/p/369099011

[4]CSDN作者 鬼道2022，蒙特卡洛方法介绍
https://blog.csdn.net/qq_38406029/article/details/120325657

[5]CSDN作者 u014430186，美赛数学模型（一）——蒙特卡罗法

https://blog.csdn.net/u014430186/article/details/112873222#:~:text=%E8%AF%AF%E5%B7%AE%20%E7%94%B1%E4%B8%AD%E5%BF%83%E6%9E%81%E9%99%90%E5%AE%9A%E7%90%86%E5%8F%AF%E7%9F%A5%EF%BC%8C%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%96%B9%E6%B3%95%E7%9A%84%E8%AF%AF%E5%B7%AE%20%CF%B5%20%E5%AE%9A%E4%B9%89%E4%B8%BA%20%CF%B5%20%3D%20N%20%CE%BB%CE%B1%CF%83,i%3D1%E2%88%91N%20X%20i2%20%E2%88%92%28N%201%20i%3D1%E2%88%91N%20X%20i%292

·

六、附录：

```python
# Python 实现蒙特卡罗算法
from random import uniform
from math import sqrt, sin, exp, pi

N = 1000000
cnt, square_cnt = 0, 0


def f(x, y, z):
    return x + exp(y) + sin(z)


for _ in range(N):
    x = uniform(-2, 2)
    y = uniform(-sqrt(4 - x ** 2), sqrt(4 - x ** 2))
    z = uniform(0, 8 - x ** 2 - y ** 2)
    cnt += f(x, y, z)
    square_cnt += f(x, y, z) ** 2
print("样本均值为", cnt / N)
print("标准差为", sqrt(square_cnt / N - (cnt / N) ** 2))
```