蒙特卡洛仿真实验报告

一、实验目的

验证蒙特卡洛仿真求解函数定积分的原理，并学会使用 MATLAB 实现功能。

二、实验原理

定积分的实质即为函数与坐标轴围成的面积，而蒙特卡洛计算积分的方式，就是在定积分区间内随机生成规定数量均匀分布的点，当函数值恒不为负时，统计落在曲线下方的点数，并与生成的总点数相比，以此来估计函数曲线下方面积与整个观测区域面积之比，这个比值与整个观测区域面积相乘即为所求的积分。函数值恒不为正时同理。

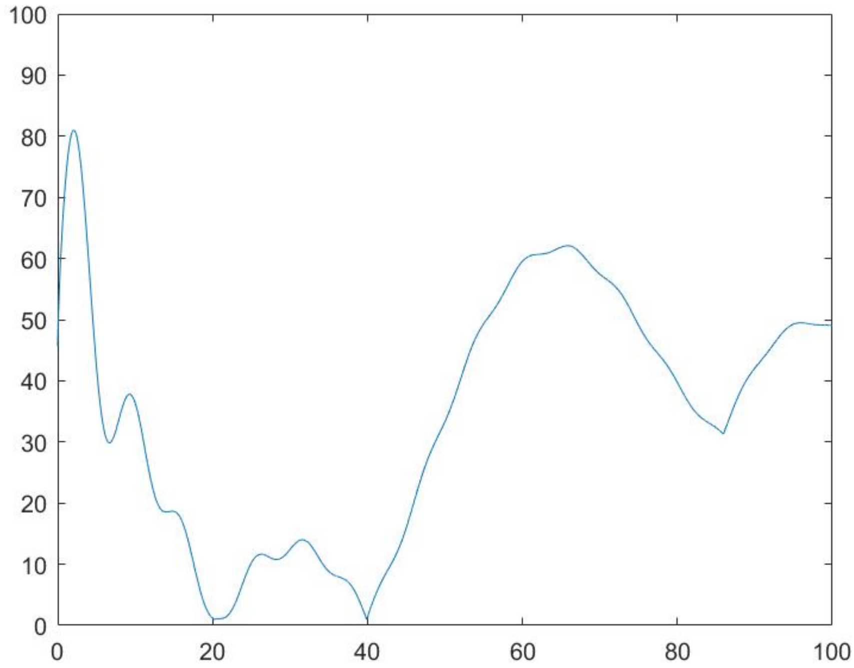
三、实验步骤

蒙特卡洛仿真代码如下

```
clc;
clear;
t=0:0.01:100;           %将仿真范围限定在 0~100,间隔为 0.01
m=input('m=');          %用 m 控制随机生成的点数
M=1000000*m;
I=0;
f=@(x) abs(30*cos(x/(pi)^2)+5*log(x+0.1)+40*sinc((x-2)/pi))+0.01*(x-30).^2;
for i=1:M
    x=100*rand;
    y=100*rand;
    fx=f(x);
    if y<fx
        I=I+1;          %如果随机数 y 比 x 的函数小，则自增 1
    end
end
I=I/M                   %I 可用于估计均匀分布点落在函数曲线内的概率
S=integral(f,0,100)     %对函数进行定积分，并在命令行输出结果
disp(['dev=',num2str(abs(I*100^2-S)/S*100),'%']); %在命令行输出仿真结果与实际积分的误差
ft=f(t);
plot(t,ft);
axis([0,100,0,100]);    %绘制函数图像，并限制坐标轴范围
```

实验开始前，我先选择了较为简单的函数 fx=100-x，以验证蒙特卡洛仿真理论以及代码的可行性。得到函数积分 S 为 5000，误差 dev 为 0.021%，认为代码可行。

更换为代码中的函数后，进一步仿真，绘制函数如图



先输入 m=1，进行两次仿真，结果分别为

I=0.3439，S=3433.6，dev=0.1462%

I=0.3428，S=3433.6，dev=0.1759%

接着增加生成的点数，令 m=10，进行两次仿真，结果分别为

I=0.3435，S=3433.6，dev=0.0509%

I=0.3434，S=3433.6，dev=0.0238%

继续增加生成的点数，令 m=100，进行两次仿真，结果分别为

I=0.3434，S=3433.6，dev=0.0029%

I=0.3434，S=3433.6，dev=0.0016%

四、实验结论

由以上数据可看出，由于软件本身保留位数的原因，当点数继续增加时，输出的概率几乎不再变化，停留在 0.3434，但误差明显随着点数的增多而减小。由此可知，生成的点数越多，蒙特卡洛仿真精度越高，仿真结果越接近真实值。