关于蒙特卡罗方法在高等数学中的应用

●简绍勇

(新余高等专科学校 数理系, 江西 新余 338000)

摘　要:运用蒙特卡罗方法并结合 Matlab软件,通过举例讨论了高数中常见的几类数学问题:数值积分、函数最值的求解、一元函数根的求解以及规划问题的求解。

关键词:蒙特卡罗方法;高等数学;Matlab

中图分类号:O242. 1　**文献标识码**:A　**文章编号**:1008 - 6765(2008)02 - 0072 - 04

1 蒙特卡罗方法简介

蒙特卡罗方法也称随机模拟方法,有时也称作随机抽样技术或统计实验方法。它的基本思想是:首先建立一个概率模型或随机过程,使它的参数等于问题的解;然后利用计算机模拟该随机现象,通过对大量模拟仿真试验的结果来分析计算所求参数,得出实际问题的近似解。

蒙特卡罗方法的特点可归纳成三个方面:(1)蒙特卡罗方法及其程序的结构简单。(2)蒙特卡罗方法的收敛性及收敛速度与问题的维数无关。(3)蒙特卡罗方法的适用性强,可用在求很多解析方法或常规数值方法难解问题的低精度解。

2 利用蒙特卡罗法求解数值积分

2.1定积分的蒙特卡罗算法

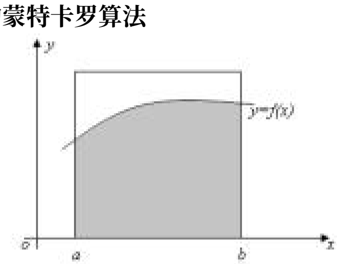

图 1

假定函数 $f(x)$ 在 $[a, b]$内有界连续且 $f(x) \geq 0$,对定积分 $I = \int_b^a f(x)\,dx$,为计算出定积分值,可构造一概率模型:取一个边长分别为 $b - a$和 M的矩形 D,使曲边梯形在矩形域之内(图 1),并在矩形内随机投点,假设随机点均匀地落在整个矩形之内,则落在图中灰色区域内的随机点数 k与投点总数 N之比 k/N,就近似地等于灰色区面积与矩形面积之比,从而得出定积 $I \approx \frac{k}{N}(b - a)M$。

例 1: 计算 $\int_{-1}^{1} e^{-x^2}\,dx$。

对 $f(x) = e^{-x^2}$,我们很难求出其原函数,所以用牛顿 - 莱布尼茨公式无法求解,可运用蒙特卡罗方法依如下步骤求出其近似值。(见图 2)

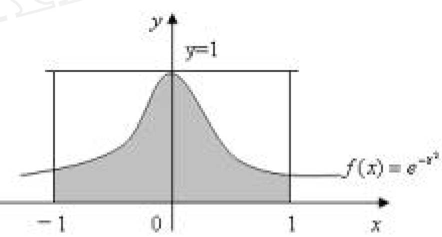

图 2

步骤 1:产生矩形域 D内的 N个均匀随机点 $P_i(x_i, y_i)$;

步骤 2:统计满足条件 $y_i \leq f(x)$的落在曲边梯形内随机点 P_i的数目 k;

步骤 3:(取 $M = 1, a = -1, b = 1, N = 100000$),则定积分的近似值 $I \approx \frac{k}{N}(b - a)M = \frac{k}{5000}$。

Matlab程序 1:

```
xi = unifrnd( - 1, 1, 100000, 1);
yi = rand(100000, 1);        %产生 100000个符合题意随机点的坐标 (xi, yi)
k = 0;
y = exp( - xi .^2);          %产生曲线上对应 xi的函数值 y
for i = 1: 100000
  if yi(i) < = y(i)
    k = k + 1;               %统计处在曲边梯形内的随机点数目
  end
end
I = k/50000
```

将以上程序利用 matlab软件重复执行 6次,所得计算结果如下

收稿日期:2007 - 11 - 28

作者简介:简绍勇 (1979 -),男,江西新余人,助教。

表 1

第 i次	1	2	3	4	5	6
积分近似值	1.4934	1.4928	1.4963	1.4936	1.4946	1.4941

注意:由于随机误差的影响,每次计算机模拟的结果可能不一样,但是各次计算结果总是在准确值附近作微小摆动。

2.2二重积分的蒙特卡罗算法

对二重积分 $I=\iint_A f(x,y)\,dxdy$,设 $f(x,y)$ 为区域 A上的有界函数且 $f(x,y)\geq 0$,据其几何意义,它是以 $f(x,y)$ 为曲面顶,A为底的曲顶柱体 C的体积。据此,用均匀随机数计算二重积分的蒙特卡罗方法基本思路为:

假设曲顶柱体 C包含在已知体积为 V_D的几何体 D的内部,在 D内产生 N个均匀随机点,统计出在 C内部的随机点数目 N_C,则 $I\approx \frac{V_D}{N}N_C$

例 2:计算 $\iint_A [1+\sqrt{1-x^2-y^2}-\sqrt{x^2+y^2}]\,dxdy$,其中 $A=\{(x,y)\mid x^2+y^2\leq 1\}$。

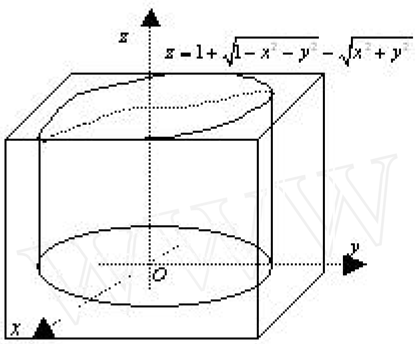

图 3

分析:该二重积分可看作以 $z=(1+\sqrt{1-x^2-y^2})-\sqrt{x^2+y^2}$为顶的曲顶柱体的体积,此曲顶柱体在一个边长为 2的立方体之内(见图 3),可计算出其精确值为 π。现利用蒙特卡罗算法计算其近似值,见以下程序:

Matlab程序 2:

```
n = 100000;
x = unifrnd(-1, 1, n, 1);
y = unifrnd(-1, 1, n, 1);
zi = unifrnd(0, 2, n, 1);
z = 1 + sqrt(1 - x.^2 - y.^2) - sqrt(x.^2 + y.^2);
Nc = 0;
for i = 1:n
  if x(i)^2 + y(i)^2 <= 1&zi(i) <= z(i)
    Nc = Nc + 1;
  end
end
I = 8 * Nc/n
```

下面是重复十次运算所得数据(见表 2)

表 2

第 k次	1	2	3	4	5
近似值	3.1450	3.1485	3.1420	3.1464	3.1468
第 k次	6	7	8	9	10
近似值	3.1386	3.1408	3.1498	3.1410	3.1436

从上面十个数据可以看出,每次计算结果在精确值 π的附近摆动。为提高精度,可选取其它随机数促进蒙特卡罗的收敛性,例如选取有利随机数来代替均匀随机数可在较少随机点的情况下使计算精度大大提高。

3 蒙特卡罗求一元函数的最值

高数中求一元函数 $f(x)$ 在闭区间 $[a,b]$的最值的方法为:找出函数 $f(x)$ 的所有驻点和不可导点并计算出其相应函数值,并将其与区间端点函数值比较得出函数的最大值最小值。但实际问题中方程 $f'(x)=0$很难求出其根,所以驻点往往很难求出。

若利用蒙特卡罗方法:在 $[a,b]$上产生 n个随机数,计算出这些数的函数值并作比较,则可得出最值近似值。

例 3: $f(x)=(1-x^3)\sin 3x$, $-2\pi\leq x\leq 2\pi$,求函数的最大值。

解法一:使用蒙特卡罗方法

matlab程序 3:

```
x = unifrnd(-2*pi, 2*pi, 1e4, 1);
fx = (1 - x.^3).*sin(3*x);
fmax = max(fx)
% 画函数图形观察
f = '(1 - x^3) * sin(3*x)';
fplot(f, [-2*pi, 2*pi]);
title('y = (1 - x^3) * sin(3*x)')
```

解法二:采用等距法搜索函数最值

matlab程序 4:

```
n = 1e4;
x = linspace(-2*pi, 2*pi, n);
fx = (1 - x.^3).*sin(3*x);
fmax = max(fx)
```

将 matlab程序 3重复运行 6次(见表 3),则可得出函数的最大值为 194.9062

表 3

第 i次	1	2	3	4	5	6
最大值	194.9008	194.9061	194.9062	194.9049	194.9038	194.9062

将 matlab程序 4运行后得函数最大值为 194.9061。根据图 4,可知两种方法的结果都很接近函数最大值。从结果上来看,可发现蒙特卡罗方法比等距法结果更优。

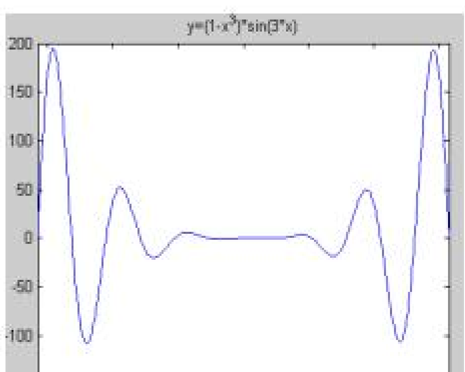

图 4

4 蒙特卡罗方法求一元函数的根

求一元函数 $f(x)=0$的根有很多方法,如一般迭代法,牛顿切线法等等,用这些方法求其根的收敛性与初始迭代值 x_0密切相关,但通常要给出一个合适的初始迭代值 x_0是比较

困难的。利用蒙特卡罗方法求根的近似值则克服了求根对初始值 x_0 的依赖性。下面介绍蒙特卡罗求函数根的方法:

要解方程 $f(x) = 0, x \in [a, b]$,其中函数 $f(x)$ 为连续函数,ξ为指定精度。令 $X_{down} = a, X_{up} = b, k = 1, F_{xmin}(1) = F_0$ 再进行以下步骤:

(1) 令 $k = k + 1$,;在 $[X_{down}, X_{up}]$ 内产生 n 个随机数 $x_i(i = 1, 2, .. n)$,计算并比较出这 n 个随机数的函数绝对值的最小值 $|f(x_{nk})| = \min\limits_{1 \le i \le n}\{f(x_i)\}$,$nk$为 的某个取值,令 $F_{xmin}(k) = \min\{F_{xmin}(k-1), f(x_{nk})\}$。

(2) 若 $F_{xmin}(k) < \xi$成立,则终止计算,令 $f_{xroot} = x_{nk}$,根就是 f_{xroot};

若 $F_{xmin}(k) > \xi$且 $F_{xmin}(k) = F_{xmin}(k-1)$,则令 $k = k - 1$,转至 (1);

若 $F_{xmin}(k) > \xi$且 $F_{xmin}(k) < F_{xmin}(k-1)$,则令 $f_{xroot} = x_{nk}$,转至 (3)。

(3) 令 $d = d_0 / k, X_{down} = f_{xroot} - d, X_{up} = f_{xroot} + d$,转至 (1)。

说明: I) F_0 是人为给定的一个很大的正数,$d_0 << b - a$ 且 $d_0 > 0$

I) $k = k + 1$ 表示重新赋值给 k,使 k的值增加 1;对 $k = k - 1$同理。

II) 对 $d = d_0 / k$是为了使搜索根的区间越来越小,以加快收敛速度。

III) 区间 [xdown, xup]一定在定义域 [a, b]之内。

此迭代步骤能使函数值序列 $F_{xmin}(1) > F_{xmin}(2) > \cdots F_{xmin}(k) > \cdots$,最终使 $F_{xmin}(k) < \xi$成立,得出函数 $f(x) = 0$达到精度要求的根 $\bar{f}_{xroot}$。

例 4: 求方程 $f(x) = e^{-x^3} - \tan x + 800 = 0$在 $(0, \pi/2)$ 的实根。(假定只要满足 $|f(x)| < 10^{-5}$,则认为 x为根)

分析:对此方程用任意给定的初值,如取初值为 0.5, 0.7, 0.9, 1.1, 1.3, 1.5,用熟知的牛顿迭代公式在区间内都找不出根,若用蒙特卡罗方法,则不需给定初值。

下为用蒙特卡罗法求方程根的 matlab程序

matlab程序 5

```
n = 10000; xup = pi/2; xdown = 0; k = 1; fxmin(1) = 1000; d0 = 1e - 4;
while(fxmin(k) > 1e - 5)
    k = k + 1;
.      x = unifrnd(xdown, xup, n, 1);   y = exp( - x ^3) - tan(x) + 800;
    fxmin(k) = fxmin(k - 1);
    for i = 1: n
        if abs(y(i)) < = fxmin(k)
          fxmin(k) = abs(y(i));
          xnum = i;
        end
    end
    if fxmin(k) = = fxmin(k - 1)
        k = k - 1; continue;
    end
    fxroot = x(xnum);
    d = d0/k; xup = fxroot + d; xdown = fxroot - d;
end
```

表 4 matlab程序 5运行后的某次结果

k	1	2	3	4	5	6	7	8	f_{xroot}
$f_{xmin(k)}$	1000	41.951	12.103	0.0043777	5.74e - 5	1.7079e - 5	1.27e - 5	2.695e - 6	1.56954636015342

从表 4可看出 fxmin(k)是一个逐步减小的序列,fxroot = 1.56954636015342时,相应的函数值 f(fxroot) = fxmin(8) = 2.695e - 6 < 1e - 5,因此方程的近似根为 fxroot = 1.56954636015342

从结果也可看出,以上利用蒙特卡罗求方程根的算法精度很高,而且其运算量与函数的复杂度是无关的,所以此求根算法具有很强的应用价值。

5 蒙特卡罗方法解规划问题

用蒙特卡罗方法可解约束规划问题:

$$\min \ f(X), X \in E^n$$
$$s.t \begin{cases} g_i(X) \ge 0, i = 1, 2, \cdots, m \\ a_j \le x_j \le b_j, j = 1, 2, \cdots, n \end{cases}$$

其基本思想是:在估计的区域 $\{(x_1, x_2, \cdots, x_n) | x_j \in [a_j, b_j], j = 1, 2, \cdots, n\}$ 内随机取若干个试验点,然后从试验点中找出可行点,再从可行点中选择出使函数值最小的点。

其具体作法为:

①符号假设和说明:

设试验点的第 j个分量 x_j 服从 $[a_j, b_j]$ 内的均匀分布;P: 试验点总数;$MAXP$: 最大试验点总数;K: 可行点总数;$MAXK$:最大可行点数;X^*:迭代产生的最优点;R:在 [0, 1]上的均匀随机数;Q:迭代产生的最小值 $f(X^*)$,其初始值为计算机所能表示的最大数.

②求解过程:

先产生一个随机数作为初始试验点,以后则将上一个试验点的第 j个分量随机产生,其它分量不变而产生一新的试验点.这样,每产生一个新试验点只需一个新的随机数分量.当 $K > MAXK$或 $P > MAXP$时停止迭代。

③流程图

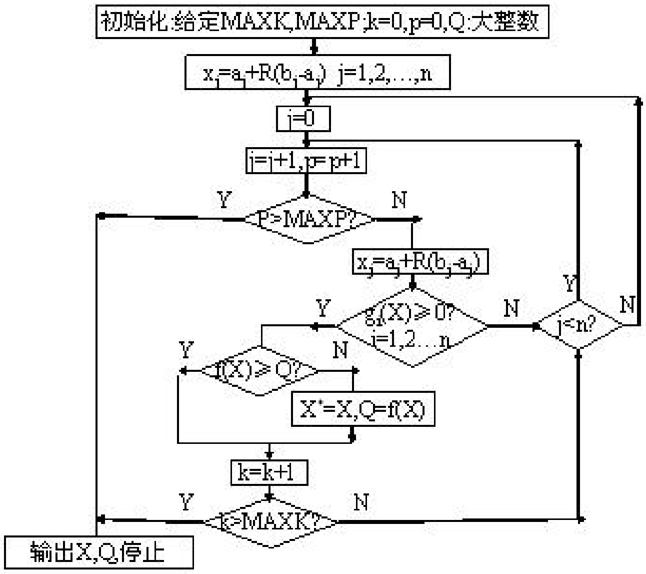

图 5 Matlab算法流程图

例 5:
$$\min \quad z = -2x_1^2 - x_2^2 + x_1x_2 + 8x_1 + 3x_1$$
$$s.t.\begin{cases} 3x_1 + x_2 = 10 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

在 *Matlab*中编程,共需三个 *M* - 文件: *randlp.m*, *mylp.m*, *lpconst.m*. 主程序为 *randlp.m*.

```
% mylp.m
function z = mylp (x)        %目标函数
z = 2 * x(1)^2 + x(2)^2 - x(1) * x(2) - 8 * x(1) - 3 * x(2);    %转化为求最小值问题
% lpconst.m
function lpc = lpconst(x)        %约束条件
if 3 * x(1) + x(2) - 10 <= 0.5 & 3 * x(1) + x(2) - 10 >= -0.5        %约束条件的误差为
   lpc = 1;
else
   lpc = 0;
end
% randlp.m
function [sol, r1, r2] = randlp (a, b, n)        %随机模拟解非线性规划
debug = 1;
a = 0;    %试验点下界
b = 10;    %试验点上界
n = 1000;    %试验点个数
r1 = unifrnd(a, b, n, 1);      % n(1阶的 [a, b]均匀分布随机数矩阵
r2 = unifrnd(a, b, n, 1);
sol = [r1(1) r2(1)];
z0 = inf;
for i = 1: n
    x1 = r1(i);
    x2 = r2(i);
    lpc = lpconst([x1 x2]);
        if lpc == 1
          z = mylp([x1 x2]);
          if z < z0
            z0 = z;
            sol = [x1 x2];
          end
        end
    end
```

参考文献:

[1] 胡良剑,孙晓君. Matlab数学实验[M]. 北京:高等教育出版社, 2006.

[2] 柴中林. 蒙特卡罗方法在无穷级数中的应用[J]. 中国计量学院学报. 2007, 18(3): 257 - 260.

[3] 赵静,但琦. 数学建模与数学实验(第 2版)[M]. 北京:高等教育出版社, 2003: 299 - 301.

(责任编校:任 华)

Application of Monte Carlo method in higher mathematics

JIAN Shao - yong

(Xinyu College, Xinyu 338000 China)

Abstract: By using Monte Carlo method in combination with Matlab software, this paper discusses several mathematic problems: numerical integration, solution of maximum and minimum of function, solution of root of function of one variable, solution of infinite series and planning problem.

Key words: Monte Carlo method; higher mathematics; Matlab

更 正

本刊 2008年第 1期《论受贿罪附加刑的完善》一文作者冯广涛的工作单位应为“浙江邮电职业技术学院”,特此更正。

本刊编辑部