信陽師範學院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

第4章 函数与数值积分

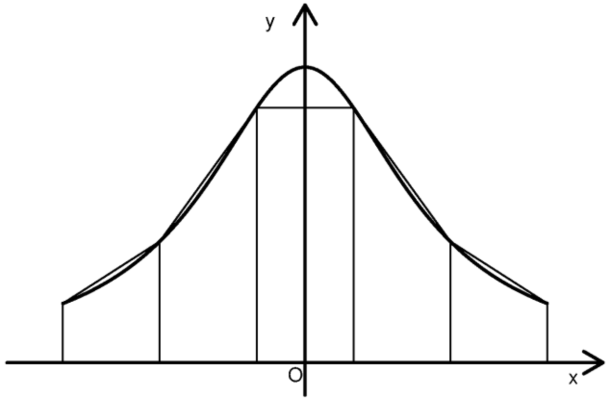
讲授人：牛言涛

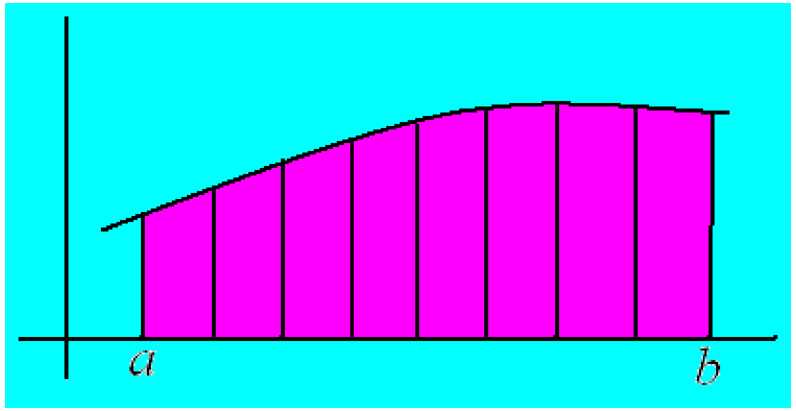
日期：2020年2月26日

目录
CONTENTS

数值积分原理

- 数值积分，用于求定积分的近似值。在数值分析中，数值积分是计算定积分数值的方法和理论。在数学分析中，给定函数的定积分的计算不总是可行的。许多定积分不能用已知的积分公式得到精确值。
- 求解定积分的数值方法多种多样，如简单的梯形法、辛普生(Simpson)法、牛顿 - 柯特斯(Newton-Cotes)法等都是经常采用的方法。它们的基本思想都是将整个积分区间$[a,b]$分成n个子区间$[x_i, x_{i+1}]$，$i=1,2,\dots,n$，其中$x_1=a$，$x_{n+1}=b$。这样求定积分问题就分解为求和问题。





1. 矩形区域积分

一重积分：(quad 、quadl不推荐使用) 、quadgk和integral函数格式：

- [q,fcnt]= quad(fun,a,b,tol,trace,p1,p2...)：simpson公式，精度低，被积函数平滑性较差。
- [q,fcnt] = quadl(fun,a,b,tol,trace,p1,p2...)：Lobatto公式，精度要求高，被积函数曲线比较平滑的数值积分。
- [q,errbnd] = quadgk(fun,a,b,param1,val1,param2,val2,...)：Gauss-Kronrod数值积分，适用于高精度和震荡数值积分，支持无穷区间，并且能够处理端点包含奇点的情况，同时还支持沿着不连续函数积分，复数域线性路径的围道积分法。
- q = integral(fun,xmin,xmax,Name,Value)：使用全局自适应积分和默认误差容限在 xmin 至 xmax 间以数值形式为函数 fun 求积分。指定具有一个或多个 Name,Value 对组参数的其他选项。例如，指定 'WayPoints'，后跟实数或复数向量，为要使用的积分器指示特定点。

矩形区域积分——案例分析

例：求积分$\int_0^{\frac{\pi}{2}} \sin(x)dx = 1$，分别用quad、quadl、quadgk和integral函数。

```
>> fh = @(x)sin(x);    %定义匿名函数
>> format long
>> I1 = quad(fh,0,pi/2)    %精度要求低，自适应simpson数值积分
I1 =
   0.999999997873119
>> I2 = quadl(fh,0,pi/2)   %精度要求高，自适应Lobatto数值积分
I2 =
   0.999999999991748
>> I3 = quadl(fh,0,pi/2,1e-12)    %给出精度要求
I3 =
   1.000000000000000
>> I4 = quadgk(fh,0,pi/2) %自适应Gauss-Kronrod数值积分，精度高
I4 =
   1.000000000000000
```

quad、quadl函数不再推荐使用，未来版本会删除这两个函数。

quad

(不推荐) 以自适应 Simpson 积分法计算数值积分

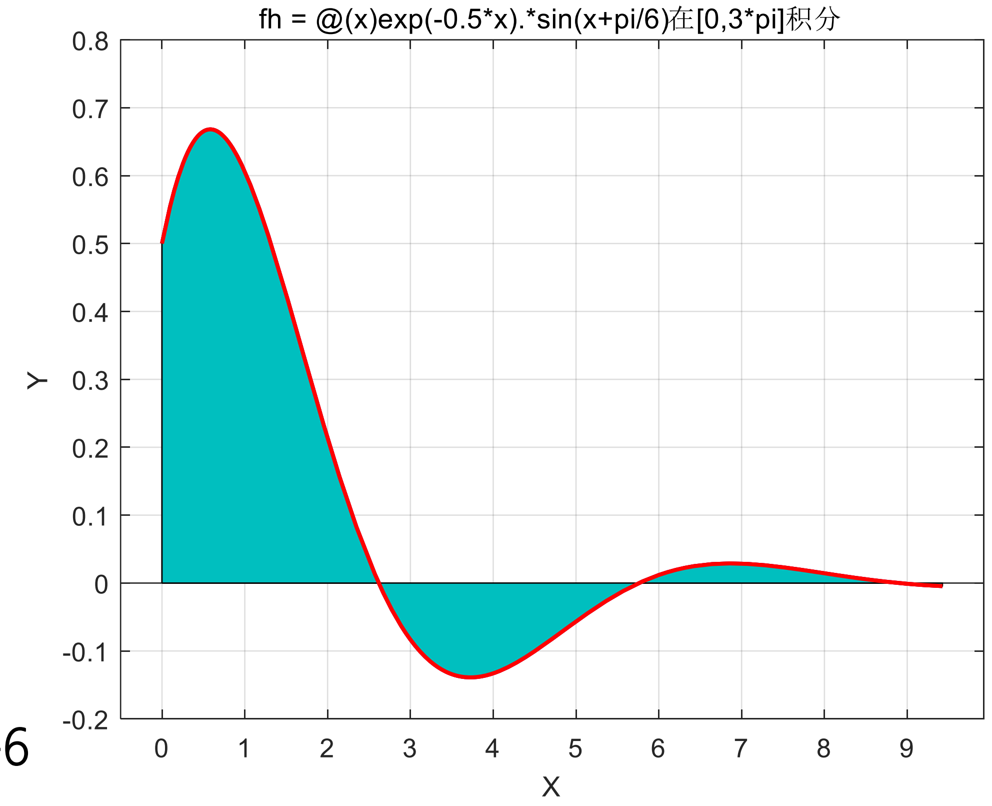
⚠ 不推荐使用 quad。请改用 integral。

```
>> fh = @(x)sin(x);
>> I5 = integral(fh,0,pi/2)
I5 =
   1.000000000000000
```

矩形区域积分——案例分析

例：求积分 $\int_0^{3\pi} e^{-0.5x}\sin(x+\frac{\pi}{6})dx$

```
>> fh = @(x)exp(-0.5*x).*sin(x+pi/6);
>> x = 0:pi/30:3*pi;   y = fh(x);
>> h = area(x,y);
>> h.FaceColor = [0 0.75 0.75];
>> hold on; grid on
>> fplot(fh,[0,3*pi],'r-','LineWidth',1.5)
>> axis([-0.5,3*pi+0.5,-0.2,0.8])
>> I1 = integral(fh,0,3*pi)  %使用默认误差容限，相对精度1e-6
I1 = 0.900840787818886
>> I2 = integral(fh,0,3*pi,'RelTol',1e-8,'AbsTol',1e-15)  %设置精度
I2 = 0.900840787818886
```



矩形区域积分——案例分析

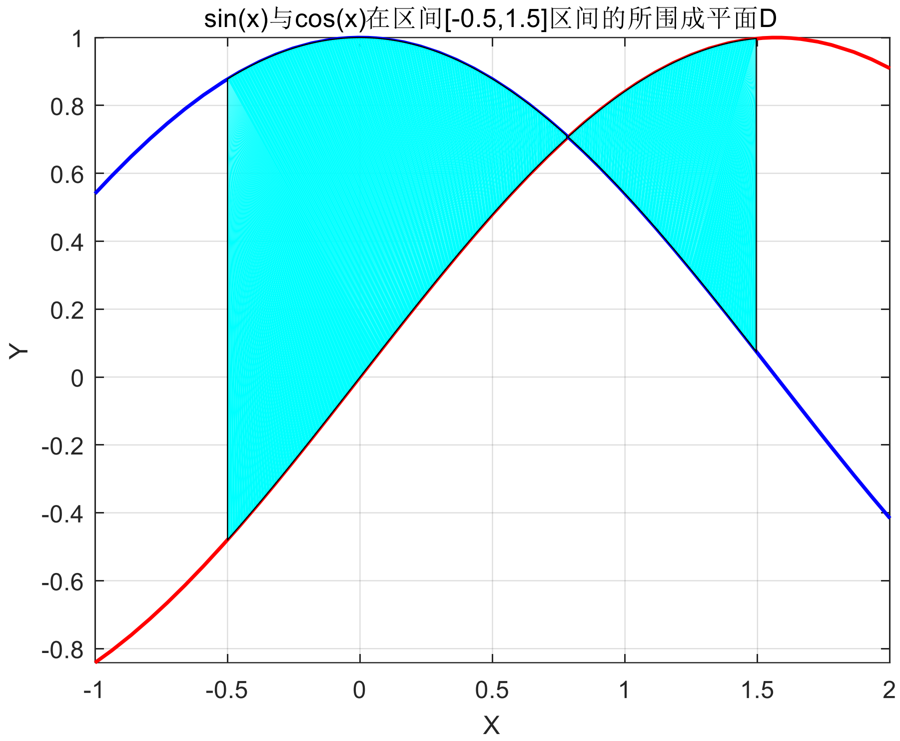
例：由 $y=\sin x$， $y=\cos x$， $x=-0.5$， $x=1.5$ 所围成的平面区域D，求平面区域D的面积。

exampleint1.m

```
y1 = @(x)sin(x);
y2 = @(x)cos(x);
fplot(y1,[-1,2],'r-','LineWidth',1.5)
hold on
fplot(y2,[-1,2],'b-','LineWidth',1.5)
[x0,y0]=fzero(@(x)sin(x) - cos(x),[-0.5,1.5]);  %求两条线的交点
x1 = -0.5:0.01:x0;
x2 = x0:0.01:1.5;
fill([x1,fliplr(x1)],[y1(x1),fliplr(y2(x1))],'c')  %填充颜色
fill([x2,fliplr(x2)],[y2(x2),fliplr(y1(x2))],'c')  %填充颜色
```

命令行窗口

```
>> fh = @(x)abs(cos(x) - sin(x));
>> S = integral(fh,-0.5,1.5)
S =
   1.362050622013523
```



2. 具有特殊形式的被积函数积分问题

quadgk函数

- 工程实际问题中，形如

$$S_1(m)=\int_a^b f(x)\sin mx dx, \quad S_2(m)=\int_a^b f(x)\cos mx dx$$

的积分，当m充分大时为高震荡函数积分。

- 对于高震荡函数积分，如果采用插值求积法进行积分，则在建立被积函数的插值多项式$P_n(x)$时，了使$P_n(x)$能够很好地逼近它们，就要$P_n(x)$也要震荡得很厉害，即要求$P_n(x)$的次数足够高。

- 但是，高次插值实际的逼近性质很不好，实用价值不大。即使采用分段低次插值，效果也不会很理想。

2. 具有特殊形式的被积函数积分问题

[q,errbnd] = quadgk(fun,a,b,param1,val1,param2,val2,...)

q = integral(fun,xmin,xmax,Name,Value)

- 1. 积分限[a,b]可以是[-inf,inf]，但必须快速衰减；
- 2. 被积函数在端点可以有奇点，如果区间内部有奇点，将以奇点区划分成多个，也就是说奇点只能出现在端点上；
- 3. 被积函数可以剧烈震荡；
- 4. 可以计算不连续积分，此时需要用到'Waypoints'参数，‘Waypoints’中的点必须严格单调；
- 5. 可以计算围道积分，此时需要用到'Waypoints'参数，并且为复数，各点之间使用直线连接；
- 6. 可以计算广义积分问题。

特殊形式积分问题——案例分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例：求解下面的分段函数的积分问题：

$$I=\int_0^4 f(x)dx \quad 其中 \quad f(x)=\begin{cases} e^{x^2}, & 0\le x\le 2 \\ \dfrac{80}{4-\sin(16\pi x)}, & 2<x\le 4 \end{cases}$$

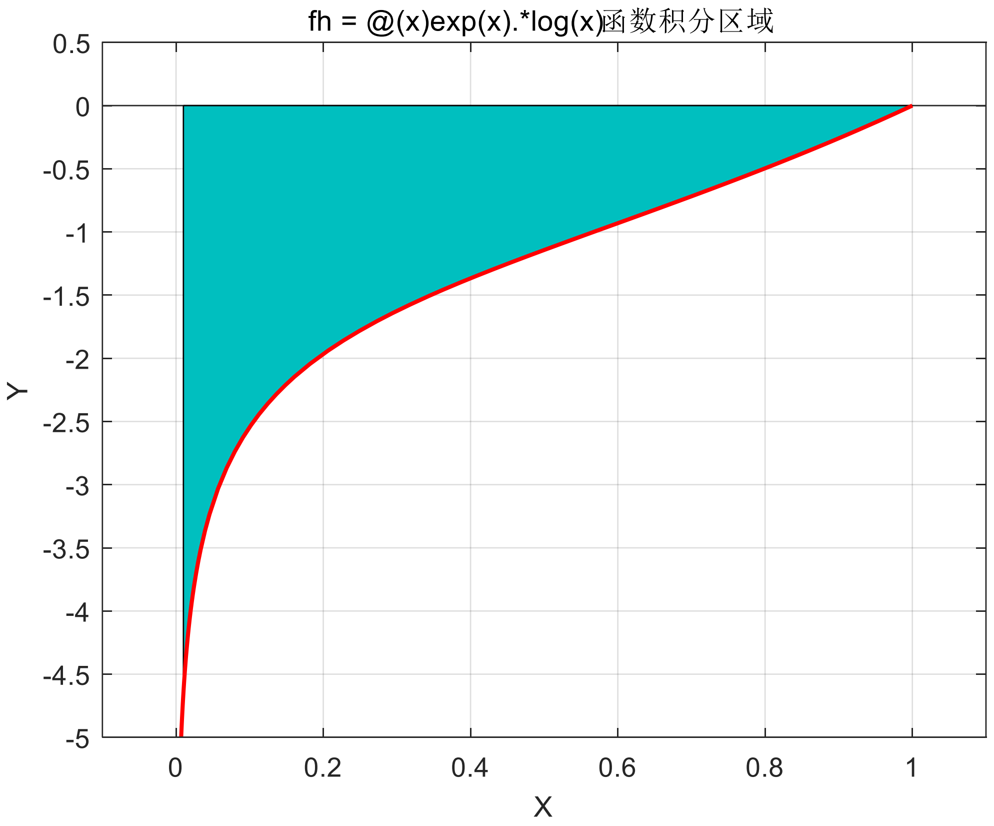
'Waypoints'通过指定不连续点的位置，在被积函数的不连续点附近高效求积分。

```
>> fh = @(x)exp(x.^2).*(x<=2)+80./(4-sin(16*pi*x)).*(x>2);
>> I1 = integral(fh,0,4,'AbsTol',1e-10,'Waypoints',[2])   %含奇点
I1 =
  57.764450125048512
>> I2 = quadgk(fh,0,4,'AbsTol',1e-10,'Waypoints',[2])   %含奇点
I2 =
  57.764450125048512
```

特殊形式积分问题——案例分析

例：计算有奇点积分 $\int_0^1 e^x \log(x)dx$

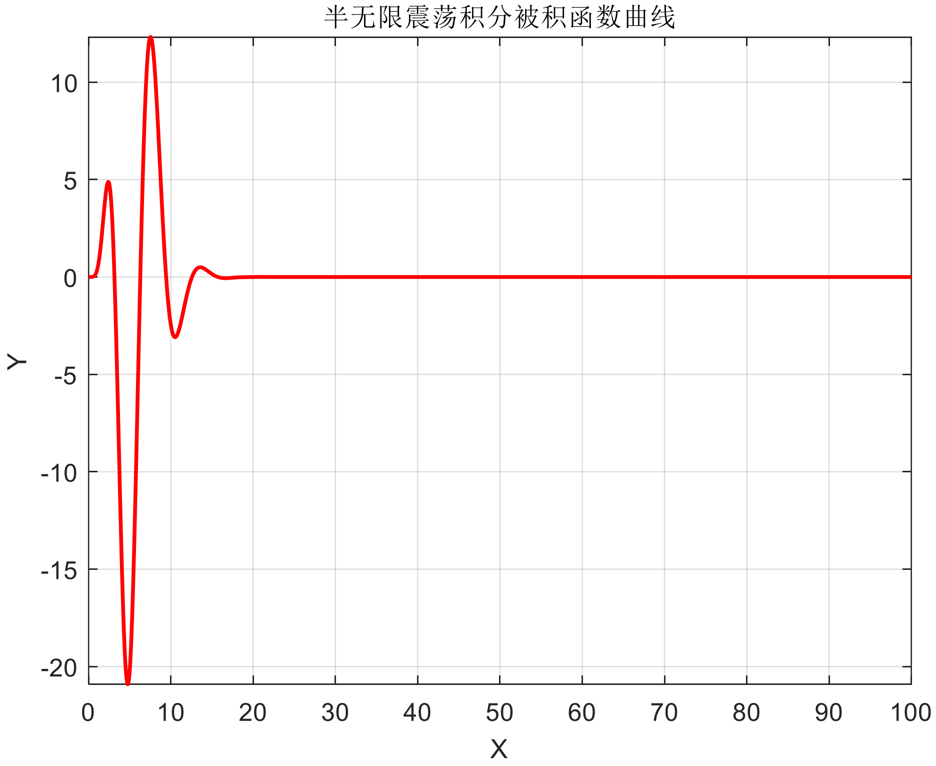
```
>> fh = @(x)exp(x).*log(x); %奇点必须在端点上，否则请先进行区间划分
>> [Q,errbnd] = quadgk(F,0,1)
Q =
  -1.317902162414081
errbnd =
     5.971115138767898e-07
>> I = integral(fh,0,1)
I =
  -1.317902162414081
```



特殊形式积分问题——案例分析

例：计算半无限(广义积分)震荡积分 $\int_0^{+\infty} x^5 e^{-x} \sin x dx$

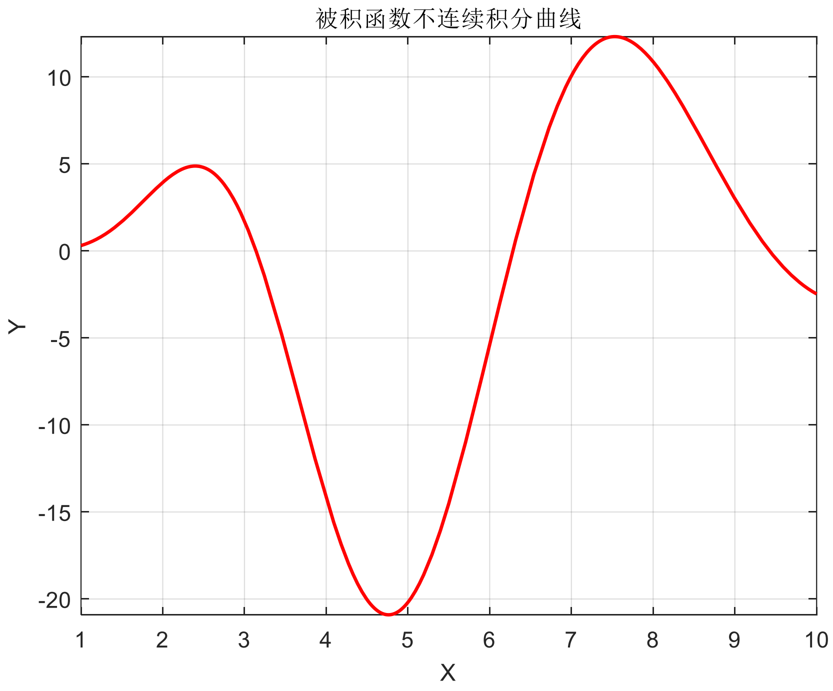
```
>> fh = @(x)x.^5.*exp(-x).*sin(x);
>> fplot(fh,[0,100],'r-','LineWidth',1.5)
>> %积分限中可以有inf，但必须快速收敛
>> [q,errbnd] = quadgk(fh,0,inf,'RelTol',1e-8,'AbsTol',1e-12)
q =
 -14.999999999998360
errbnd =
     9.438576354429664e-09
>> I = integral(fh,0,inf,'RelTol',1e-8,'AbsTol',1e-12)
I =
 -14.999999999998360
```



特殊形式积分问题——案例分析

例：计算不连续积分 $I=\int_1^{10} f(x)dx=\int_1^{10} x^5 e^{-x}\sin x dx$， $f(2)=1000$， $f(5)=-100$

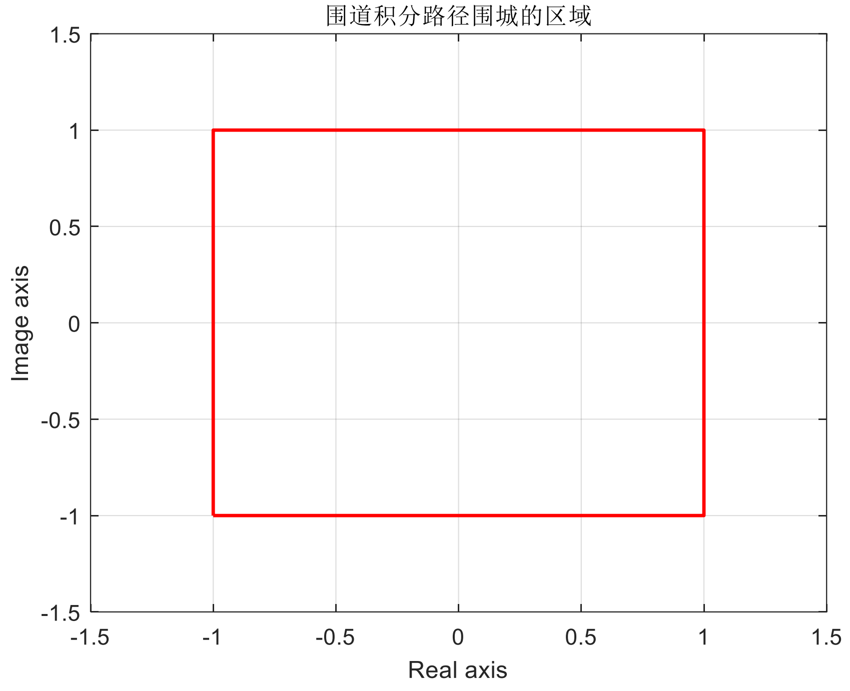
```
>> fh = @(x)x.^5.*exp(-x).*sin(x);
>> fplot(fh,[1,10],'r-','LineWidth',1.5)
>> [q,errbnd] = quadgk(fh,1,10,'Waypoints',[2 5])    %显然2，5为间断点
q =
 -10.940771682195068
errbnd =
     3.317415541455360e-14
>> I = integral(fh,1,10,'Waypoints',[2 5])    %显然2，5为间断点
I =
 -10.940771682195065
```



特殊形式积分问题——案例分析

例：计算围道积分，在复数域内，积分函数1/(2*z-1)，积分路径为由[-1-i 1-i 1+i -1+i -1-i]围成的矩形边框。

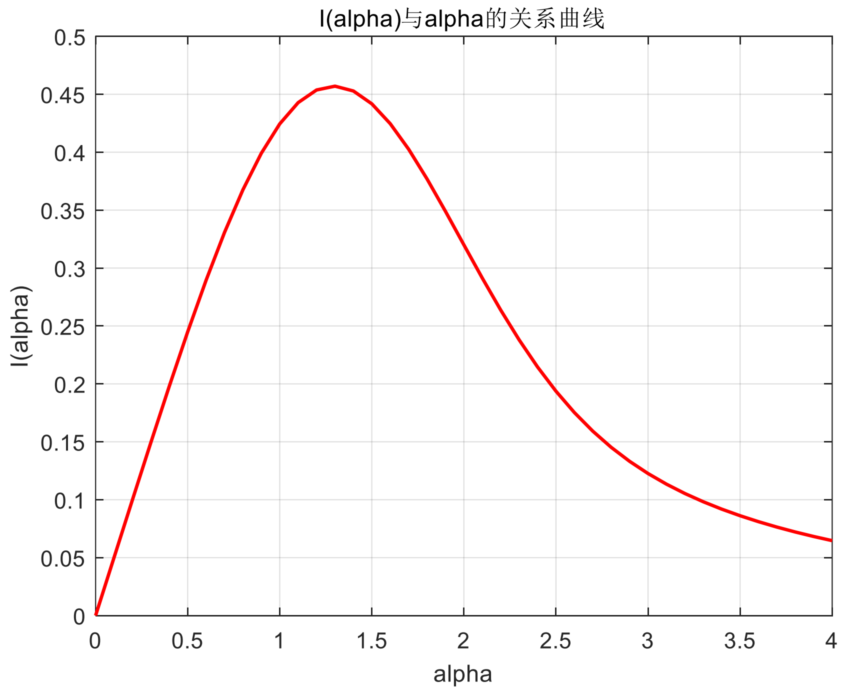
```
>> Waypoints = [-1-i 1-i 1+i -1+i -1-i];
>> plot(Waypoints);%绘制积分路径
>> xlabel('Real axis');ylabel('Image axis');axis([-1.5 1.5 -1.5 1.5]);grid on;
%注意各点间使用直线连接
>> [Q,errbnd] = quadgk(@(z)1./(2*z - 1),-1-i,-1-i,'Waypoints',[1-i,1+i,-1+i])
Q =  -0.0000 + 3.1416i
errbnd =  5.0509e-08
>> [Q2,errbnd2] = quadgk(@(z)1./(2*z - 1),-1-i,-1-i,'Waypoints',Waypoints)
%使用这个的效果也是一样的，就是说始末点可以随便包不包含在Waypoints中
Q2 = -0.0000 + 3.1416i
errbnd2 = 5.0509e-08
>> I = integral(@(z)1./(2*z - 1),-1-i,-1-i,'Waypoints',Waypoints)
I =
  0.000000000000000 + 3.141592653589802i
```



特殊形式积分问题——案例分析

例：已知 $I(\alpha)=\int_0^{\infty} e^{-\alpha x^2}\sin\left(\alpha^2 x\right)dx$， 试绘制出$I(\alpha)$与$\alpha$的关系曲线，其中$\alpha\in(0,4)$.

```
>> fh = @(x,alpha)exp(-alpha.*x.^2).*sin(alpha.^2.*x);
>> xi = 0:0.1:4;
>> I = integral(@(x)fh(x,xi),0,inf,'RelTol',1e-8,'AbsTol',1e-12,'ArrayValued',true);
>> plot(xi,I,'r-','LineWidth',1.5)
>> grid on
>> title('I(alpha)与alpha的关系曲线')
>> xlabel('alpha')
>> ylabel('I(alpha)')
>> grid on
```



3. 二重积分（矩形区域）

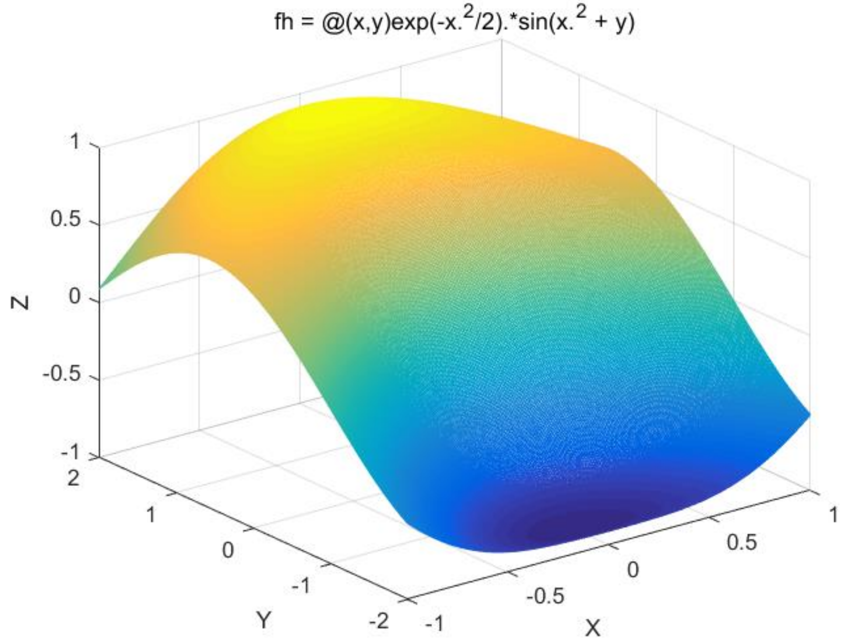
信陽師範學院
数学与统计学院

函数 dblquad（不推荐）、integral2：二重积分

- q = dblquad(fun,xmin,xmax,ymin,ymax,tol,method)，函数的参数分别为函数句柄、两个自变量的积分限，指定积分结果的精度，method 的取值可以是 @quadl，也可以是用户自定义的积分函数句柄，返回积分结果。
- q = integral2(fun,xmin,xmax,ymin,ymax) 在平面区域 xmin ⩽ x ⩽ xmax 和 ymin(x) ⩽ y ⩽ ymax(x) 上逼近函数 z = fun(x,y) 的积分。
- q = integral2(fun,xmin,xmax,ymin,ymax,Name,Value) 指定具有一个或多个 Name,Value 对组参数的其他选项。

二重积分（矩形区域）——案例分析

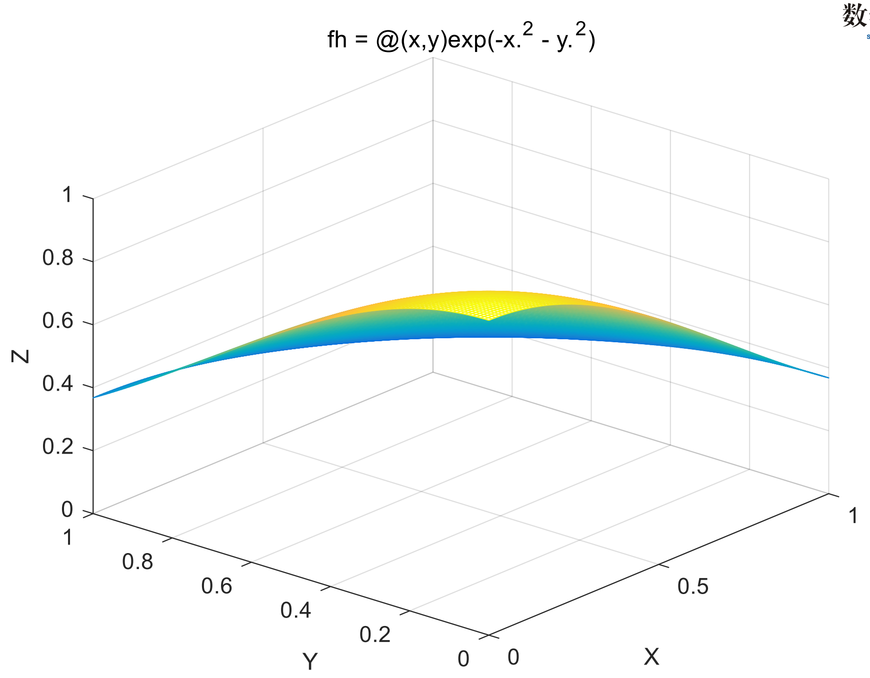
例：试求出二重定积分 $I=\int_{-1}^{1}\int_{-2}^{2}e^{-x^2/2}\sin(x^2+y)dxdy$




```
>> fh = @(x,y)exp(-x.^2/2).*sin(x.^2 + y);   %匿名函数

>> I2=dblquad(fh,-2,2,-1,1)

I2 =

   1.574493189744944

>> I3=dblquad(fh,-2,2,-1,1,1e-12,@quadl)

I3 =

    1.57449815921736

>> q = integral2(fh,-2,2,-1,1,'Method','iterated','AbsTol',1e-8,'RelTol',1e-12)

q =

   1.574498159217361
```

二重积分（矩形区域）——案例分析

例：计算 $I=\int_0^1\int_0^1 e^{-x^2-y^2}dxdy$ 的二重积分。




```
>> fh = @(x,y)exp(-x.^2 - y.^2);
>> [x,y] = meshgrid(0:0.01:1,0:0.01:1);
>> fhz = fh(x,y);
>> mesh(x,y,fhz)  %绘制三维网格面
>> q = integral2(fh,0,1,0,1,'Method','iterated','AbsTol',1e-8,'RelTol',1e-12)
q =
   0.557746285351034
```

4. 三重积分（矩形区域）

triplequad(不推荐)、integral3函数：三重积分

- q = triplequad(fun,xmin,xmax,ymin,ymax,zmin,zmax,tol,method)：类似dblquad函数使用方法。
- q = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax,Name,Value)在区域 xmin ⩽ x ⩽ xmax、ymin(x) ⩽ y ⩽ ymax(x) 和 zmin(x,y) ⩽ z ⩽ zmax(x,y) 逼近函数 z = fun(x,y,z) 的积分，且可以指定具有一个或多个 Name,Value 对组参数的其他选项。

例：用数值方程求解三重积分 $\int_0^1\int_0^\pi\int_0^\pi 4xze^{-x^2y-z^2}\,dzdydx$

```
>> fh = @(x,y,z)4*x.*z.*exp(-x.^2.*y-z.^2);
>> I1 = triplequad(fh,0,1,0,pi,0,pi,1e-10,@quadl)
I1 =
   1.732762223030904
```

```
>> I2 = integral3(fh,0,1,0,pi,0,pi,'AbsTol',1e-8,'RelTol',1e-12)
I2 =
   1.732762223028910
```

5. 向量化积分

- 矢量化自适应simpson数值积分：[Q,fcnt] = quadv(fun,a,b,tol,trace)
- integral，指定 'ArrayValued',true 以便计算数组值或向量值函数的积分。

矢量积分相当于多个一元函数积分。当被积函数中含有参数，需要对该参数的不同值计算该函数的积分时，可以使用一元函数的矢量积分。矢量积分返回一个向量，每个元素的值为一个一元函数的积分值。

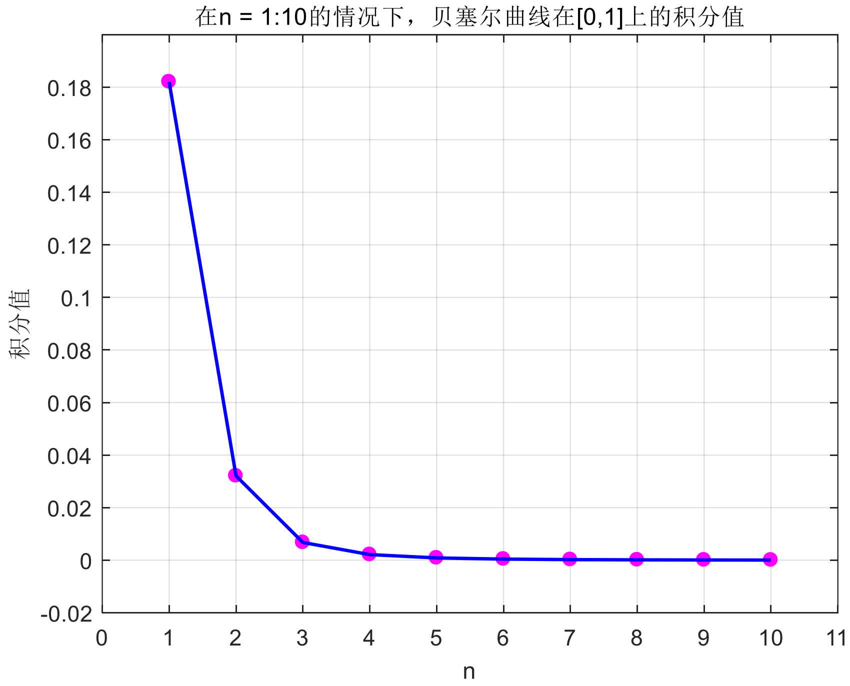
- 例：求贝塞尔函数 $f(x) = besselk(0, n^2\sqrt{x} + 1)$ 在n取1~10情况下，x从0~1的积分。

```
>> fh = @(x,n)besselk(0,(1:n).^2*x.^0.5+1);  %贝塞尔函数定义
>> sf = quadv(@(x)f(x,10),0,1)  %求解的同时先传递参数n的值10，进而求矢量积分
sf =
    0.1820    0.0321    0.0068    0.0021    0.0009    0.0004    0.0002    0.0001    0.0001    0.0001
```

5. 向量化积分

- 例：求贝塞尔函数 $f(x)=besselk(0,n^2\sqrt{x}+1)$ 在n取1~10情况下，x从0~1的积分。

```
>> fh2 = @(x)besselk(0,(1:10).^2*x.^0.5+1);
>> q = integral(fh2,0,1,'ArrayValued',true)
q =
   0.1820   0.0321   0.0068   0.0021   0.0009   0.0004   0.0002   0.0001   0.0001   0.0001
>> plot(1:10,q,'mo','MarkerFaceColor','m')
>> hold on
>> plot(1:10,q,'b-','LineWidth',1.5)
>> grid on
>> axis([0,11,-0.02,0.2])
>> title('在n = 1:10的情况下，贝塞尔曲线在[0,1]上的积分值')
```



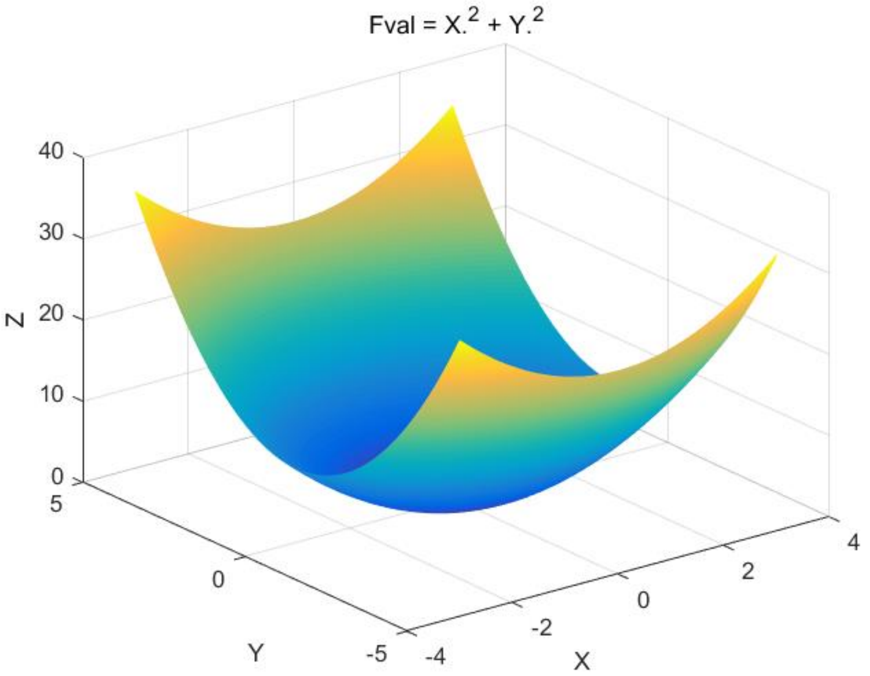
6. 离散数据积分

- **Z = trapz(X,Y,dim)**：梯形数值积分，通过已知参数x,y按dim维使用梯形公式进行积分。

```
>> x = 0:pi/100:pi/2;
>> y = sin(x);
>> S1 = trapz(x,y)   %利用离散数据积分
S1 = 0.999917751943722
```

$$S = \int_0^{\frac{\pi}{2}} \sin x dx$$

$$V = \int_{-5}^{5}\int_{-3}^{3} x^2 + y^2 dxdy$$




```
>> x = -3:0.01:3;  %积分区间
>> y = -5:0.01:5;  %积分区间
>> [X,Y] = meshgrid(x,y);  %生成网格点
>> Fval = X.^2 + Y.^2;
>> %trapz 先对 x 求积分以生成列向量。然后，y 上的积分可将列向量减少为单个标量。
>> V = trapz(y,trapz(x,Fval,2))  %trapz 稍微高估计确切答案 680，因为 f(x,y) 是向上凹的。
V = 680.0020
```

离散数据积分——案例分析

例：在桥梁的一端每隔一段时间记录1min有几辆车过桥，得到数据见表，试估计一天通过桥梁的车流量。

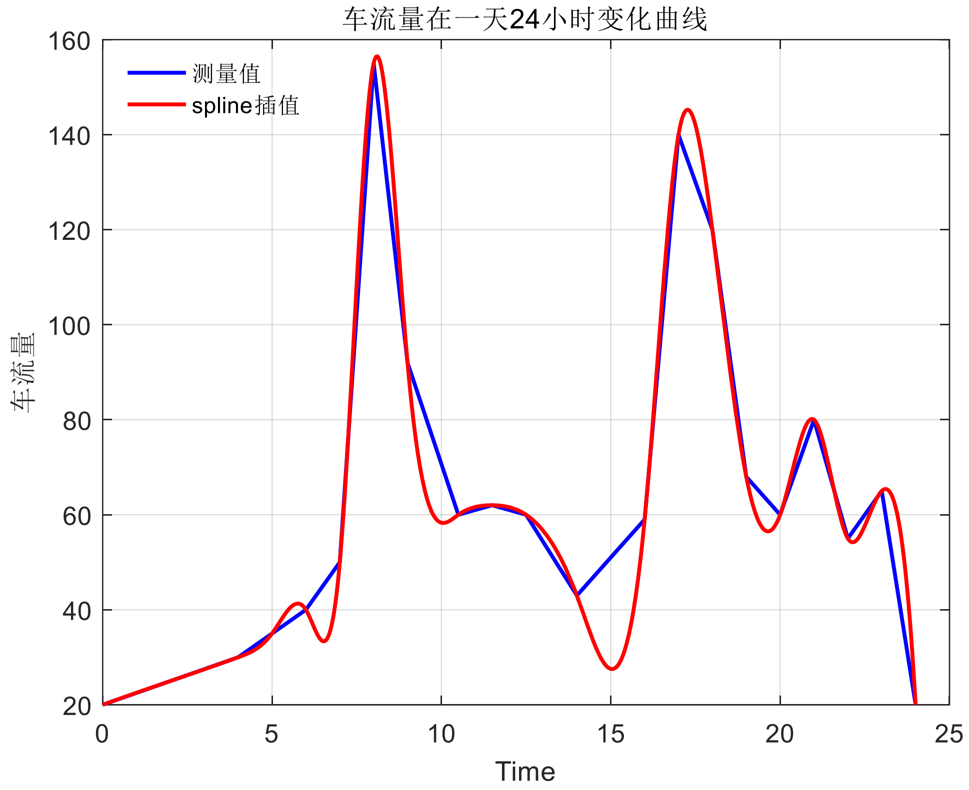
时间	0:00	2:00	4:00	5:00	6:00	7:00	8:00
车辆数/辆	20	25	30	35	40	50	155
时间	9:00	10:30	11:30	12:30	14:00	16:00	17:00
车辆数/辆	92	60	62	60	43	59	140
时间	18:00	19:00	20:00	21:00	22:00	23:00	24:00
车辆数/辆	120	68	60	80	55	65	20

解：记录时刻为t时，相应的车辆数为$x(t)$，则所求车流量即为计算积分$\int_0^{24} x(t)\mathrm{d}t$

```
>> x=[0,2,4,5:9,10.5,11.5,12.5,14,16:24];
>> y = [20,25,30,35,40,50,155,92,60,62,60,43,59,140,120,68,60,80,55,65,20];
```

离散数据积分——案例分析

```
>> s1 = trapz(x,y)  %利用梯形公式计算积分
s1 =
    1.483750000000000e+03
>> xi = 0:1/60:24;   %插值点
>> yi = spline(x,y,xi);  %通过三次样条插值求插值点函数值
>> tfi = trapz(xi,yi)  %梯形离散数据积分
tfi =
    1.463326739891108e+03
>> pp = spline(x,y);    %三次样条插值生成插值点系数矩阵
>> fh = @(x)fnval(pp,x);  %获得系数矩阵函数
>> tf = integral(fh,0,24)   %采用一重积分函数
tf =
    1.463329673879469e+03
```



离散数据积分——案例分析

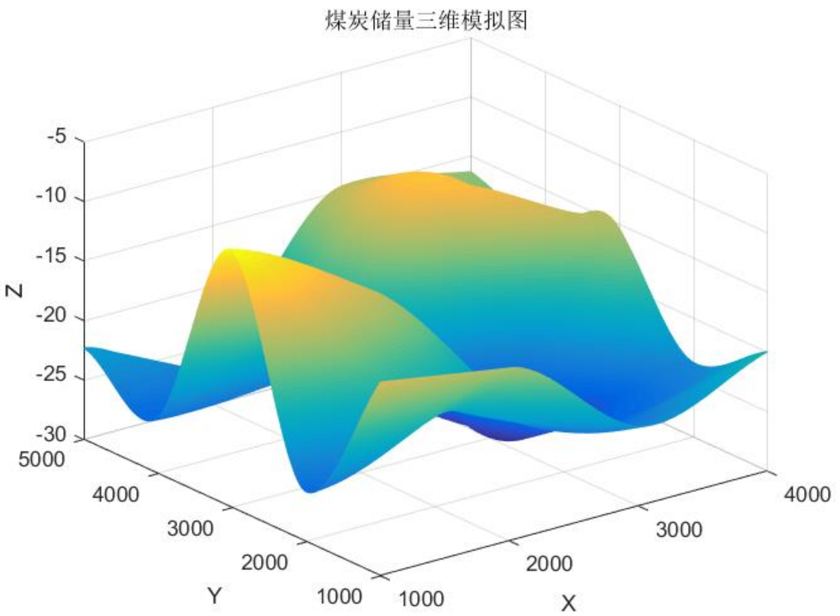
例：某煤矿为估计其煤炭的储量，在该矿区内进行勘探，得到数据如表。试估算出该矿区（$1\le x\le 4, 1\le y\le 5$）煤炭的储量。

编号	1	2	3	4	5	6	7	8	9	10
坐标x(km)	1	1	1	1	1	2	2	2	2	2
坐标y(km)	1	2	3	4	5	1	2	3	4	5
煤炭厚度z(m)	13.72	25.80	8.47	25.27	22.32	15.47	21.33	14.49	24.83	26.19
编号	11	12	13	14	15	16	17	18	19	20
坐标x(km)	3	3	3	3	3	4	4	4	4	4
坐标y(km)	1	2	3	4	5	1	2	3	4	5
煤炭厚度z(m)	23.28	26.48	29.14	12.04	14.58	19.95	23.73	15.35	18.01	16.29

$$V \approx \frac{h_x h_y}{4}[-5\varphi(a,c)-5\varphi(a,d)-5\varphi(b,c)-5\varphi(b,d)+2\sum_{k=0}^{m}[\varphi(a,y_k)+\varphi(b,y_k)]+2\sum_{j=0}^{n}[\varphi(x_j,c)+\varphi(x_j,d)]+4\sum_{j=0}^{n}\sum_{k=0}^{m}\varphi(x_j,y_k)$$

离散数据积分——案例分析


```
x=[1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4]*1000;
y=[1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5]*1000;
z=-[13.72 25.80 8.47 25.27 22.32 15.47 21.33 14.49 24.83 26.19 23.28 26.48 ...
    29.14 12.04 14.58 19.95 23.73 15.35 18.01 16.29];
hx=1; hy=1;  %插值步长
cx=1000:hx:4000; cy=1000:hy:5000;  %插值点
[X,Y]= meshgrid(cx,cy);  %生成网格点
Z = griddata(x,y,z,X,Y,'cubic');   %离散数据插值
mesh(X,Y,Z)  %绘制三维网格图
n=length(cx);
m=length(cy);
V=-hx*hy*(-5*Z(1,1)-5*Z(1,n)-5*Z(m,1)-5*Z(m,n)+2*(sum(Z(1,:)+Z(m,:))+sum(Z(:,1)+Z(:,n)))+4*sum(sum(Z)))/4
V = 2.4722e+08
```

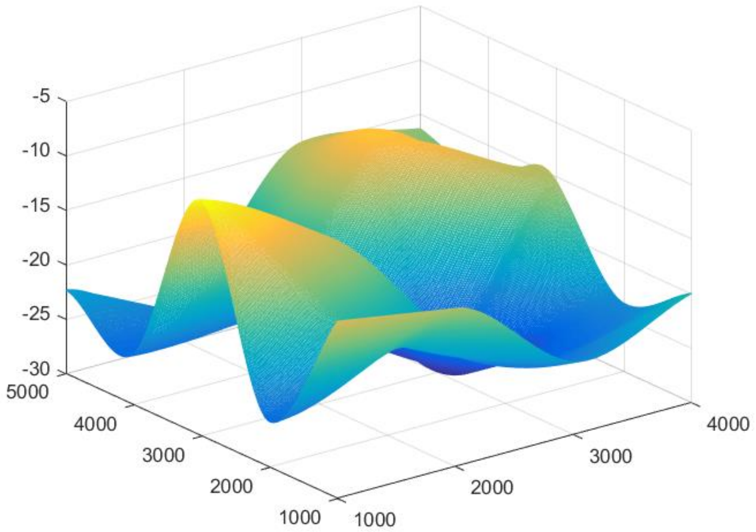


$$V \approx \frac{h_x h_y}{4}\left[-5\varphi(a,c)-5\varphi(a,d)-5\varphi(b,c)-5\varphi(b,d)+2\sum_{k=0}^{m}\left[\varphi(a,y_k)+\varphi(b,y_k)\right]+2\sum_{j=0}^{n}\left[\varphi(x_j,c)+\varphi(x_j,d)\right]+4\sum_{j=0}^{n}\sum_{k=0}^{m}\varphi(x_j,y_k)\right)$$

离散数据积分——案例分析


```
function [V,S] = trapz2(x,y,z)
% 求离散点曲面柱体体积V和面积S，x、y和z为原始测量值。
if nargin <= 2 || nargout <= 1
    error('输入或输出参数不足，格式：[V,S] = trapz2(x,y,z)，请重新输入！')
end
disp('-------------复合梯形公式求解此离散数据的积分start---------------');
format long;  %设置长精度
format compact;  %设置显示的紧凑格式
hx = input('请输入所求x的步长或插值精度：');
hy = input('请输入所求y的步长或插值精度：');
cx = min(x):hx:max(x);  %精细，用于插值
cy = min(y):hy:max(y);  %精细，用于插值
n = length(cx);
m = length(cy);
[X,Y] = meshgrid(cx,cy);  %生成网格数据
Z = griddata(x,y,z,X,Y,'cubic');  %散乱点插值
mesh(X,Y,Z);  %绘制曲面图形
%--------------以下求曲面体积------------------------
V = hx*hy*(-5*Z(1,1)-5*Z(1,n)-5*Z(m,1)-5*Z(m,n)+2*(sum(Z(1,:)+Z(m,:))...
    +sum(Z(:,1)+Z(:,n)))+4*sum(sum(Z)))/4;
%--------------以下求曲面面积------------------------
DZ = abs(diff(Z));  %求Z轴两点间距离差，即三角形的边b，a为hx
xy = sqrt(DZ.^2 + hx^2);  %求三角形的边c
S = sum(sum(xy.*hx)); %求面积之和
disp('-------------复合梯形公式求解此离散数据的积分结果为-------------');
end
```

```
>> [V,S] = trapz2(x,y,z)
-------------复合梯形公式求解此离散数据的积分start--
请输入所求x的步长或插值精度：10
请输入所求y的步长或插值精度：10
-------------复合梯形公式求解此离散数据的积分结果为-
V =
    -2.496465650563782e+08
S =
     1.204053337007178e+07
```



7. 一般区域二重积分

- quad2d(fun,a,b,c(x),d(x))：计算函数fun(x,y)在区域$a<x<b$, $c(x)<y<d(x)$的积分。
- integral2(fun,xmin,xmax,ymin,ymax) 在平面区域 xmin ⩽ x ⩽ xmax 和 ymin(x) ⩽ y ⩽ ymax(x) 上逼近函数 z = fun(x,y) 的积分。
- 例：求函数 $f(x,y)=\sqrt{10000-x^2}$， $x^2+y^2\le 10000$ 的积分。

```
>> tic,y1 = dblquad(@(x,y) sqrt(10^4-x.^2).*(x.^2+y.^2<=10^4),-100,100,-100,100 ),toc
     y1 =  2.666666666262098e+06    % 时间已过 6.380750 秒。
>> tic,y2 = quad2d(@(x,y) sqrt(10^4-x.^2),-100,100,@(x)-sqrt(10^4-x.^2),@(x)sqrt(10^4-x.^2)),toc
     y2 = 2.666666666666666e+06    % 时间已过 0.215190 秒。
>> tic,q = integral2(fh,-100,100,@(x)-sqrt(10^4-x.^2),@(x)sqrt(10^4-x.^2)),toc
     q = 2.666666666665794e+06   % 时间已过 0.085573 秒。
```

7. 一般区域二重积分

- 例：求二重积分 $\int_{10}^{20}\int_{5x}^{x^2} e^{\sin(x)}\ln(y)dydx$

```
>> y = @(x,y)exp(sin(x)).*log(y);
>> s1 = quad2d(y,10,20,@(x)5*x,@(x)x.^2,'AbsTol',1e-15,'RelTol',1e-10)
s1 =
    9.368671342676773e+03

>> q = integral2(y,10,20,@(x)5*x,@(x)x.^2,'Method','iterated','AbsTol',1e-15,'RelTol',1e-10)
q =
    9.368671342679692e+03
```

arrayfun用法介绍

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

arrayfun可以避免无谓的循环，大大提高代码的简介性。

1、A=arrayfun(FUN, B)

FUN是函数句柄，对B中的每一个元素调用FUN函数(计算顺序随意)，结果存放于A中，size(A)==size(B)，FUN函数可接受numeric, logical, char, struct, cell的自变量类型。

2、[A, B, ...] = arrayfun (FUN, C, ...)

FUN函数的返回值是一个向量，对B中的每一个元素调用FUN函数,计算结果放在A,B...中。

3、 A = arrayfun (FUN, B, C, ...)

FUN函数接受参数不唯一，分别调用B, C, ...中的元素，A(i,j...)=Fun(B(i,j...),C(i,j,...)...)，B,C...大小必须相等。

8. 一般区域三重积分

- quadl(@(x) arrayfun(@(xx) quad2d(被积函数f(xx,y,z)关于y,z变量的函数句柄,y积分下限y1(xx),y积分上限y2(xx),z积分下限z1(xx,y),z积分上限z2(xx,y)),x),x积分下限值,x积分上限值)。

- 例：求三重积分 $\int_1^2 \int_{\sqrt{x}}^{2x} \int_{xy}^{2x+y} x\sin(y+z)(x+z^2)dzdydx$

```
>> fh = @(x)arrayfun(@(x)integral2(@(y,z)x.*sin(y+z).*(x+z.^2),x^0.5,2*x,@(y)x*y,@(y)(2*x+y)),x);
>> I = integral(fh,1,2)
I =
  12.653057242613546
```

8. 一般区域三重积分

- q = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax) 在区域 xmin ≤ x ≤ xmax、ymin(x) ≤ y ≤ ymax(x) 和 zmin(x,y) ≤ z ≤ zmax(x,y) 逼近函数 z = fun(x,y,z) 的积分。
- 例：求三重积分 $\int_1^2\int_{\sqrt{x}}^{2x}\int_{xy}^{2x+y} x\sin(y+z)(x+z^2)dzdydx$

```
>> fh = @(x,y,z)x.*sin(y+z).*(x+z.^2);
>> xmin = 1;
>> xmax = 2;
>> ymin = @(x)sqrt(x); %y的下限
>> ymax = @(x)2*x; %y的上限
>> zmin = @(x,y)x.*y;  %z的下限，是关于x和y的函数
>> zmax = @(x,y)2*x + y; %z的上限，是关于x和y的函数
>> q = integral3(fh,xmin,xmax,ymin,ymax,zmin,zmax,'Method','tiled')
 q = 12.653057240716713
```

8. 一般区域三重积分

- 例：求二重积分 $\int_{0.2}^{1} 2ye^{-y^2}\left(\int_{-1}^{1}\frac{e^{-x^2}}{y^2+x^2}dx\right)^2 dy$

```
>> fh = @(y)2*y*exp(-y^2)*(integral(@(x)exp(-x.^2)./(y.^2+x.^2),-1,1))^2;
>> I = integral(fh,0.2,1,'ArrayValued',true)
     I = 10.213463733952988
>> I = integral(@(y) 2*y.*exp(-y.^2).*arrayfun(@(y)integral(@(x) exp(-x.^2)./(y.^2+x.^2),-1,1),y).^2,0.2,1)
     I = 10.213463733952986
%或者
>> fh = @(y) 2*y.*exp(-y.^2).*arrayfun(@(y)quadl(@(x) exp(-x.^2)./(y.^2+x.^2),-1,1),y).^2;
>> I = quadl(fh,0.2,1)
     I = 10.213463739161503
```

9. 一般区域的n重积分

一般区域n重积分函数：f = nIntegrate(fun,low,up)，自定义函数

- f为函数的返回值是n重积分积分结果。fun是被积函数字符串形式，不同的变量依次以x1,x2,...xn表示，（需要注意的是，必须以x1,x2,...xn这种形式表示，其余像y1,y2,...yn或是其他表示方法都不行）。
- low和up都是长度为n的cell数组，low存储从外到内各重积分的积分下限函数，up存储从外到内各重积分的积分上限函数（都是字符串形式）。low和up内的函数遵循原则：
  - 设积分重数为m，最内层到最外层的积分变量依次以xm,...x2,x1来表示（只能以这样顺序，其他顺序或者别的字母表示变量都不可以）；最内层的上下限函数不管是不是关于x(m-1)的函数都要求x(m-1)必须出现，不是其函数时可以写成“0*x(m-1)”等形式；
  - 依次类推次内层要求x(m-2)必须要出现等等；一直到次外层要求变量x1出现，最外层才是常数。

一般区域的n重积分——案例分析

$$\int_0^1\left(\int_0^1\left(\int_0^1\left(\int_0^1 e^{x_1x_2x_3x_4}dx_4\right)dx_3\right)dx_2\right)dx_1$$

```
>> fun1 = 'exp(x1*x2*x3*x4)';
>> up = {'1','0*x1+1','0*x2+1','0*x3+1'};
>> low = {'0','0*x1','0*x2', '0*x3'};
>> f1 = nIntegrate(fun1,low,up)
f1 =
   1.069397608862895
>> fh = @(x1,x2,x3,x4)exp(x1.*x2.*x3.*x4);
>> %创建一个函数句柄，它使用 integral3 计算三个积分。
>> fhx = @(x1)integral3(@(x2,x3,x4)fh(x1,x2,x3,x4),0,1,0,1,0,1);
>> I = integral(fhx,0,1,'ArrayValued',true)
I =
   1.069397608860038
```

$$\int_1^2\left(\int_{x_1}^{2x_1}\left(\int_{x_1x_2}^{2x_1x_2} x_1x_2x_3dx_3\right)dx_2\right)dx_1$$

```
>> fun2 = 'x1*x2*x3';
>> up = {'2','2*x1','2*x1*x2'};
>> low = {'1','x1','x1*x2'};
>> f2 = nIntegrate(fun2,low,up)
f2 =
     1.792968750000302e+02
>> fh = @(x1,x2,x3)x1.*x2.*x3;
>> I =
integral3(fh,1,2,@(x1)x1,@(x1)2*x1,@(x1,x2)x1.*x2,@(x1,x2)
2*x1.*x2)
I =
     1.792968750000117e+02
```

一般区域的n重积分——案例分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

$$\int_1^2\left(\int_{x_1}^{3x_1}\left(\int_{x_1x_2}^{2x_1x_2}\left(\int_{x_1+x_1x_3}^{x_1+2x_1x_3}\left(\sqrt{x_1x_2}\ln x_3+\sin\left(\frac{x_4}{x_2}\right)\right)dx_4\right)dx_3\right)dx_2\right)dx_1$$

```
fun4 = 'sqrt(x1*x2)*log(x3)+sin(x4/x2)';    %构造被积函数字符表达式
up4 = {'2','3*x1','2*x1*x2','x1+2*x1*x3+0*x2'} ;    %积分上限函数字符表达式
low4 = {'1','x1','x1*x2','x1+x1*x3+0*x2'} ;    %积分下限函数字符表达式
f = nIntegrate(fun4,low4,up4)
f = 1.502515543166793e+03
```

```
fh = @(x1)arrayfun(@(x1)integral3(@(x2,x3,x4)sqrt(x1*x2).*log(x3) + sin(x4./x2),...
        x1,3*x1,@(x2)x1*x2,@(x2)2*x1*x2,@(x2,x3)x1+x1*x3,@(x2,x3)x1+2*x1*x3),x1)
>> I = integral(fh,1,2)
I =
    1.502515542840579e+03
```

一般区域的n重积分——案例分析

$$\int_0^1 dx_1 \int_{\exp(x_1)/2}^{\exp(x_1)} dx_2 \int_{(x_1+\sin(x_2))/2}^{x_1+\sin(x_2)} dx_3 \int_{(x_1+x_3)/2}^{x_1+x_3} dx_4 \int_{x_4}^{2x_4} (\sin(x_1 \exp(x_2\sqrt{x_3})) + x_4^{x_5})dx_5$$

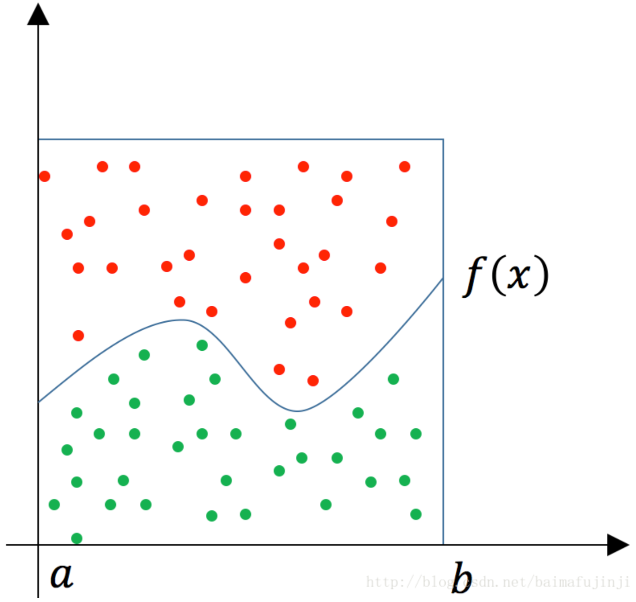
```
fun5 = 'sin(x1*exp(x2*sqrt(x3)))+x4^x5';    %构造被积函数字符表达式
up5 = {'1','exp(x1)','x1+sin(x2)','x1+x3','2*x4'};   %积分上限函数字符表达式
low5 = {'0','exp(x1)/2','x1/2+sin(x2)/2','x1/2+x3/2','x4'};  %积分下限函数字符表达式
f = nIntegrate(fun5,low5,up5)
f = 5.798135377232081
```

```
fh = @(x1)arrayfun(@(x1)integral(@(x2)arrayfun(...
    @(x2)integral3(@(x3,x4,x5)sin(x1*exp(x2*sqrt(x3)))+x4.^x5,...
    (x1+sin(x2))/2,x1+sin(x2),@(x3)(x1+x3)/2,@(x3)x1+x3,@(x3,x4)x4,@(x3,x4)2*x4),x2),...
    exp(x1)/2,exp(x1)),x1);   %计算效率较低
I = integral(fh,0,1)
I = 5.798135365534838
```

10. 蒙特卡洛计算n重积分

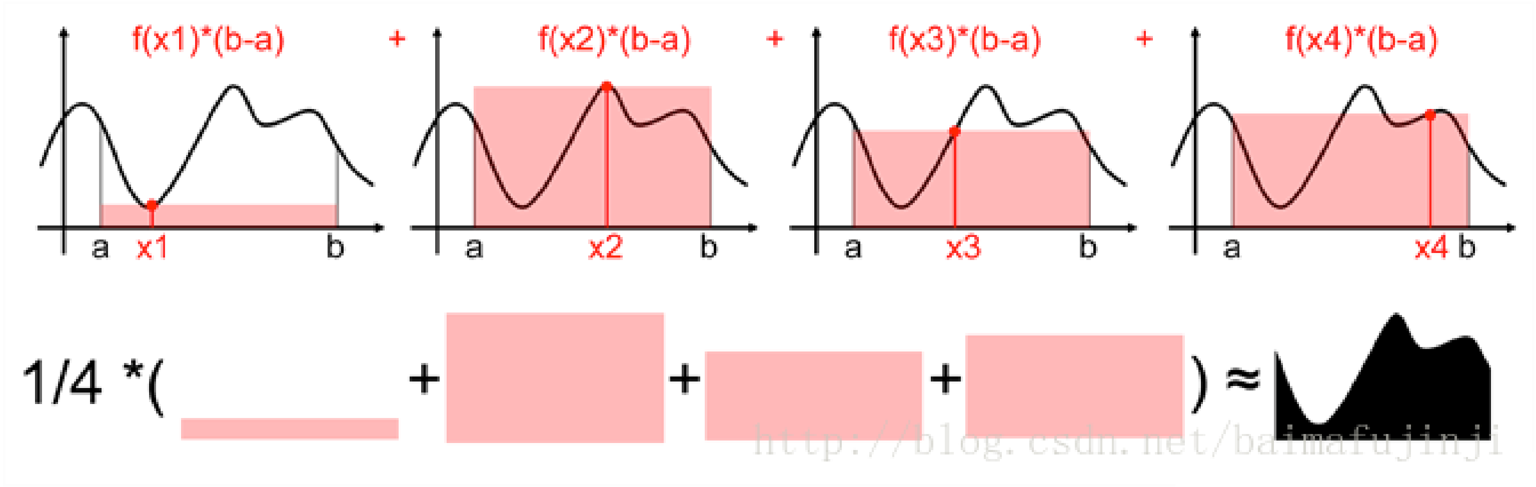
- 计算定积分 $S=\int_a^b f(x)dx$

一、蒙特卡洛求定积分：投点法



$$S=\int_a^b f(x)dx \approx \max\big(f(x)\big)\times(b-a)\times\frac{blue}{blue+red}$$

二、蒙特卡洛求定积分：期望法



$$S=\int_a^b f(x)dx \approx (b-a)\frac{1}{N}\sum_{i=0}^{N-1} f\left(X_i\right)$$

10. 蒙特卡洛计算n重积分

- 蒙特卡洛方法又称随机抽样法或统计试验方法，它用于求积分时，与积分重数无关，这点非常重要。
- 虽然四维以下的积分用蒙特卡洛法效率可能不如传统的一些数值积分方法，但是维数高的时候，蒙特卡洛法比传统方法要有效的多，而且实现起来也非常容易。
- 可以说，计算高维积分是蒙特卡洛方法最成功和典型的应用。
- 实际应用中，有多种蒙特卡洛方法可以计算n重积分，比较常用的基本蒙特卡洛法和等分布序列的蒙特卡洛法。

(1) 基本的蒙特卡洛积分法

例：用蒙特卡洛积分法求解 $\int_2^5\int_0^{x^2-2}\left(x^4-8xy+2\sin(y)-3\right)dydx$

```
>> fhm = @(x)(x(:,1).^4 - 8*x(:,1).*x(:,2) + 2*sin(x(:,2)) - 3);
>> n = 1000000;
>> x(:,1) = unifrnd(2,5,1,n);
>> x(:,2) = unifrnd(0,23,1,n);
>> ind = (x(:,2) >= 0 & (x(:,2) <= x(:,1).^2 - 2));
>> fhIm = (5-2)*(23-0)*sum(fhm(x(ind,:)))/n
fhIm =
   1.7130e+03
```

```
>> fh = @(x,y)(x.^4 - 8*x.*y + 2*sin(y) - 3);
>> fhI = quad2d(fh,2,5,0,@(x)(x.^2-2))
fhI =
   1.7069e+03
```

(1) 基本的蒙特卡洛积分法

例：用蒙特卡洛积分法求解 $\int_{10}^{20}\int_{5x}^{x^2} e^{\sin(x)}\ln(y)dydx$

```
>> fh = @(x)exp(sin(x(:,1))).*log(x(:,2));
>> n = 10000000;
>> x(:,1) = unifrnd(10,20,1,n);
>> x(:,2) = unifrnd(50,400,1,n);
>> ind = (x(:,2)>=5*x(:,1) & x(:,2)<=x(:,1).^2);
>> In = (20-10)*(400-50)*sum(fh(x(ind,:)))/n
In =
  9.3756e+03
```

```
>> y = @(x,y)exp(sin(x)).*log(y);
>> s1 = quad2d(y,10,20,@(x)5*x,@(x)x.^2)
s1 =
    9.368671342614414e+03
```

(1) 基本的蒙特卡洛积分法

例：用蒙特卡洛积分法求解（见一般区域的n重积分）

$$\int_1^2\left(\int_{x_1}^{3x_1}\left(\int_{x_1x_2}^{2x_1x_2}\left(\int_{x_1+x_1x_3}^{x_1+2x_1x_3}\left(\sqrt{x_1x_2}\ln x_3+\sin\left(\frac{x_4}{x_2}\right)\right)dx_4\right)dx_3\right)dx_2\right)dx_1$$

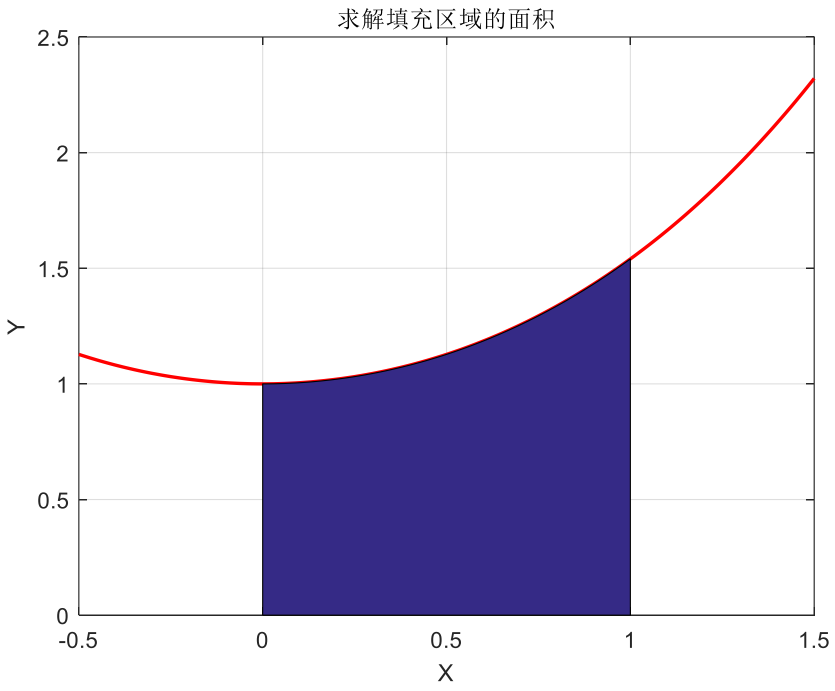
```
>> fh = @(x)(sqrt(x(:,1).*x(:,2)).*log(x(:,3)) + sin(x(:,4)./x(:,2)));
>> n = 1000000;
>> x(:,1) = unifrnd(1,2,1,n);   %x1上下限
>> x(:,2) = unifrnd(1,6,1,n);   %x2上下限
>> x(:,3) = unifrnd(1,24,1,n);   %x3上下限
>> x(:,4) = unifrnd(2,98,1,n);   %x4上下限
>> ind = (x(:,2)>=x(:,1)&x(:,2)<=3*x(:,1))&(x(:,3)>=x(:,1).*x(:,2)&x(:,3)<=2*x(:,1).*x(:,2))&...
          (x(:,4)>=(x(:,1)+x(:,1).*x(:,3))&x(:,4)<=(x(:,1)+2*x(:,1).*x(:,3)));
>> fnl = (2-1)*(6-1)*(24-1)*(98-2)*sum(fh(x(ind,:)))/n
fnl =
   1.5106e+03
```

信陽師範學院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

(1) 基本的蒙特卡洛积分法

例：计算 $f(x)=\int_0^1 (x^2+\cos(x))dx$

```
fh = @(x)(x.^2 + cos(x));
fplot(fh,[-0.5,2.5],'r-','LineWidth',1.5)
axis([-0.5,1.5,0,2.5])
hold on
x = 0:0.01:1;    y = fh(x);
area(x,y);  grid on
%蒙特卡洛积分法
n = 1000000; %生成点数
x = unifrnd(0,1,1,n); %随机生成服从均匀分布的点
I = (1-0)*sum(fh(x))/n %计算积分值
% I = 1.174778034654582
```

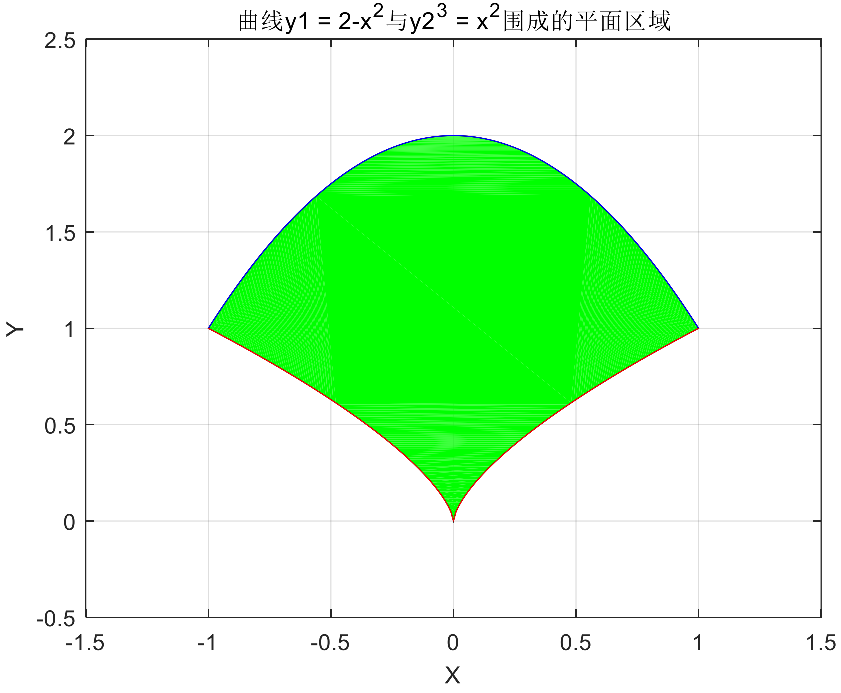



```
%使用函数integral求解
>> fh = @(x)(x.^2 + cos(x));
>> I = integral(fh,0,1)
% I = 1.174804318141230
```

(1) 基本的蒙特卡洛积分法

例：给定曲线$y = 2 - x^2$和曲线$y^3 = x^2$，曲线的交点为：$P_1(-1，1)$、$P_2(1，1)$。曲线围成平面有限区域，用蒙特卡罗方法计算区域面积。

```
x = -1:0.01:1;
y1 = 2 - x.^2;  y2 = (x.^2).^(1/3);
fill([x,fliplr(x)],[y1,y2],'g')
hold on; grid on
plot(x,y1,'b-',x,y2,'r-');
axis([-1.5,1.5,-0.5,2.5])
fh = @(x)(2-x.^2 - (x.^2).^(1/3));
I = quadgk(fh,-1,1)
I =
   2.13333302183321
```

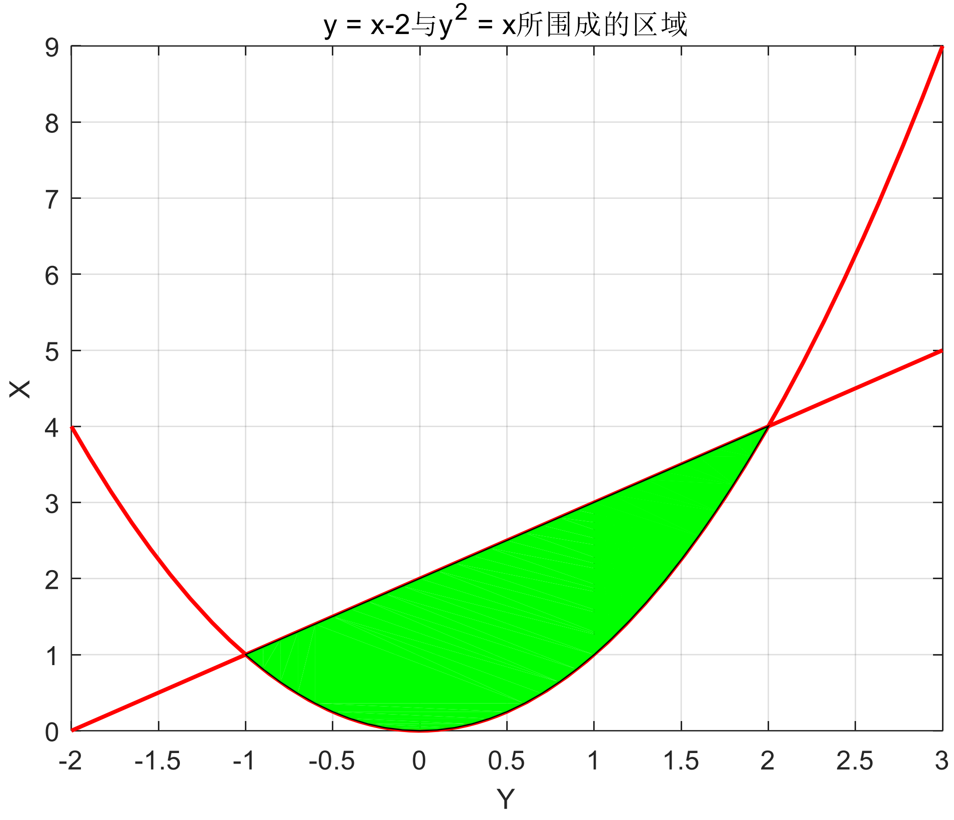



```
>> n = 1000000;
>> x = unifrnd(-1,1,1,n);
>> fnI = (1-(-1))*sum(fh(x))/n
fnI =
   2.132739371454548
>> n = 10000000; %增加10倍
>> x = unifrnd(-1,1,1,n);
>> fnI2 = (1-(-1))*sum(fh(x))/n
fnI2 =
   2.133221854062748
```

(1) 基本的蒙特卡洛积分法

- 例：计算 $\iint\limits_D xy^2\,dxdy$，其中$D$为$y = x - 2$与$y^2 = x$所围$D$的边界曲线交点为：$(1,-1)$、$(4, 2)$。

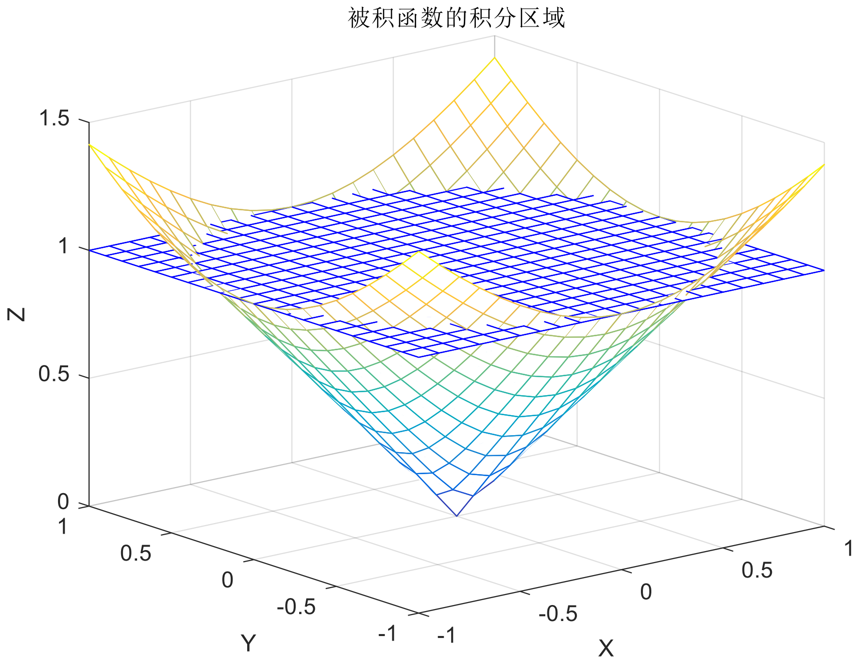
```
>> I = quad2d(@(x,y)x.*y.^2,0,4,@(x)x-2,@(x)sqrt(x))
I =
   7.9238
>> fh = @(x)x(:,1).*x(:,2).^2;
>> n = 1000000;
>> x(:,1) = unifrnd(0,4,1,n);
>> x(:,2) = unifrnd(-2,2,1,n);
>> ind = (x(:,2) >= x(:,1)-2) & (x(:,2) <= sqrt(x(:,1)));
>> fhI = (4-0)*(2-(-2))*sum(fh(x(ind,:)))/n
fhI =
   7.8933
```



(1) 基本的蒙特卡洛积分法

例：用蒙特卡罗方法计算 $\iiint\limits_{\Omega}(x^2+y^2+z^2)dxdydz$ ，其中，积分区域是由$z=\sqrt{x^2+y^2}$和 z = 1 所围成。被积函数在求积区域上的最大值为2。所以有四维超立方体$-1\le x\le 1$，$-1\le y\le 1$，$0\le z\le 1$，$0\le u\le 2$ 。

```
% 绘制被积区域
[X,Y] = meshgrid(-1:0.1:1,-1:0.1:1);
Z = sqrt(X.^2 + Y.^2);
mesh(X,Y,Z)
hold on
zz = ones(length(X));
h = mesh(X,Y,zz);
set(h,'EdgeColor','blue')
```



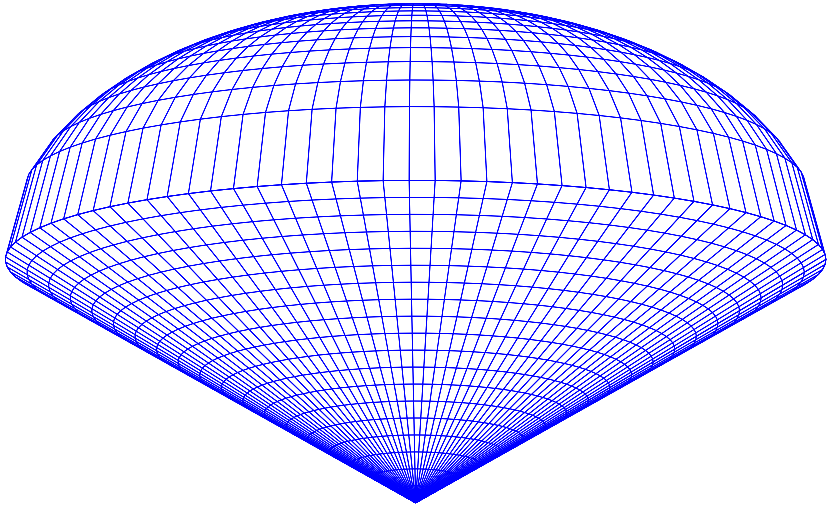

```
% 基本蒙特卡洛积分法求解
>> data = rand(100000,4);
>> x = -1+2*data(:,1);
>> y = -1+2*data(:,2);
>> z = data(:,3);
>> u = 2*data(:,4);
>> xyzu = find(z>sqrt(x.^2+y.^2)&...
   z<=1&u<=x.^2+y.^2+z.^2);
>> n = length(xyzu);
>> V = 8*n/100000
   V = 0.9376
```

(1) 基本的蒙特卡洛积分法

例：蒙特卡罗方法计算体积，冰淇淋锥含于体积 = 8 的六面体

$$\Omega=\{(x,y,z)\,|\,-1\le x\le 1,-1\le y\le 1,0\le z\le 2\}\quad z\ge\sqrt{x^2+y^2}\ \ \&\ \ z\le 1+\sqrt{1-x^2-y^2}$$

```
m=20;
n=100;
t=linspace(0,2*pi,n);
r=linspace(0,1,m);
x=r'*cos(t);
y=r'*sin(t);
z1=sqrt(x.^2+y.^2);
z2=1+sqrt(1+eps-x.^2-y.^2);
X=[x;x];
Y=[y;y];
Z=[z1;z2];
mesh(X,Y,Z)
view(0,-18)
colormap([0 0 1])
axis off
```



(1) 基本的蒙特卡洛积分法

```
function [q,error] = MonteC(ts)
if nargin == 0
    ts = 7;
end
N = input('请输入生成的随机个数N：');
for k = 1:ts
   data = rand(N,3);
   x = 2*data(:,1) - 1;   y = 2*data(:,2) - 1;
   z = 2*data(:,3);
   R2 = x.^2+y.^2;
   R = sqrt(R2);
   xyz = find(z >= R & z <= 1+sqrt(1-R2));
   q(k) = 8 * length(xyz)/ N;
end
error = q - pi;
```

```
%在命令窗口调用函数
>> [q,error] = MonteC(8)
请输入生成的随机个数N：1000000
q =
    3.1348    3.1422    3.1375    3.1356    3.1472
3.1463    3.1418    3.1398
error =
   -0.0068    0.0006   -0.0041   -0.0059
0.0056    0.0047    0.0002   -0.0018
>> I = mean(q)   % 8次蒙特卡洛积分均值
I =
   3.143730000000000
```

(2) 等分布序列的蒙特卡洛积分法

一般的蒙特卡洛法具有计算不可重复性等缺点，相比而言，等序列的蒙特卡洛法在选定产生等序列的无理数后，计算结果唯一确定，比采用随机序列的蒙特卡洛法误差阶要好。

- 三重积分
  - x3 = bsxfun(@times, repmat(1:n,3,1),[sqrt(2);sqrt(3);sqrt(6)/3]);
- 四重积分
  - x4 = bsxfun(@times, repmat(1:n,4,1), [sqrt(2); sqrt(3); sqrt(6)/3; sqrt(10)]);
- 五重积分
  - x5 = bsxfun(@times, repmat(1:n,5,1),[sqrt(2); sqrt(3); sqrt(6)/3; sqrt(10); sqrt(19)]);

(2) 等分布序列的蒙特卡洛积分法

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例：计算 $f(x)=\int_0^1 (x^2+\cos(x))dx$

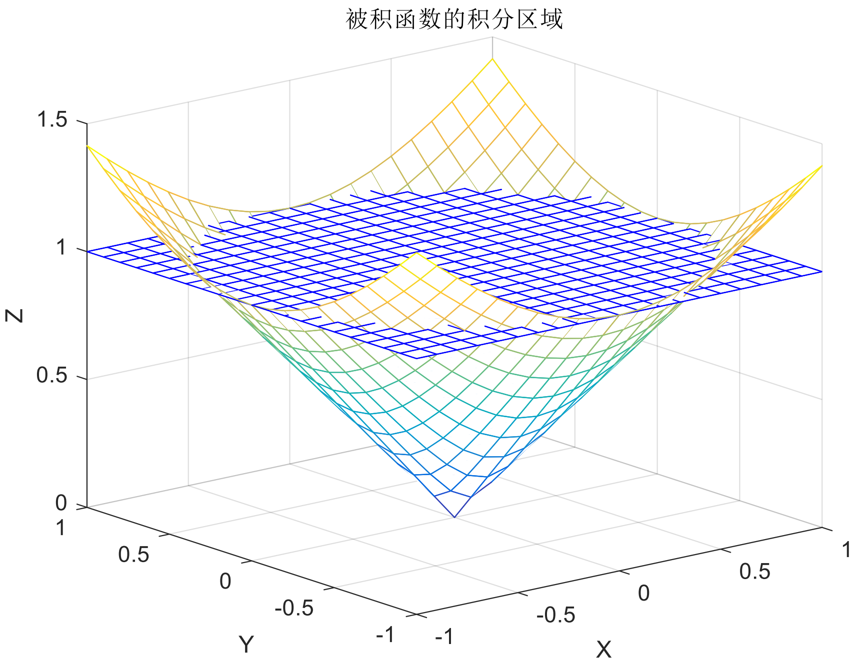
```
%等分布序列的蒙特卡洛积分法
>> x = bsxfun(@times,repmat(1:200000,1,1),[sqrt(2);]);
>> x = mod(x,1);
>> y = x.^2 + cos(x);
>> s = sum(y) * (1 - 0)/200000
s = 1.174803469
%基本的蒙特卡洛积分法
>> t = rand(1,200000);
>> xrnd = 0 + (1 - 0) * t;
>> yrnd = xrnd.^2 + cos(xrnd);
>> s = sum(yrnd) /200000
s = 1.1743
```

```
%符号求解
>> syms x
>> f = x^2 + cos(x);
>> s = int(f,0,1)   %精确解
s =
   sin(1) + 1/3
>> vpa(s,10)
ans =
1.174804318
```

(2) 等分布序列的蒙特卡洛积分法

例：用蒙特卡罗方法计算 $\iiint\limits_{\Omega}(x^2+y^2+z^2)dxdydz$ ，其中，积分区域是由$z=\sqrt{x^2+y^2}$和 z = 1 所围成。被积函数在求积区域上的最大值为 2。所以有四维超立方体−1≤ x ≤1，−1≤ y ≤1，0≤ z ≤1，0≤ u ≤2 。

```
>> data = bsxfun(@times,repmat(1:100000,3,1),[sqrt(2);sqrt(3);sqrt(6)/3]);
>> data = mod(data,1);
>> x=4*data(1,:);
>> y=-1+3*data(2,:);
>> z=16*data(3,:);
>> xyz = find(x>=y.^2&x<=y+2&z<=x.*(y.^2));
>> n = length(xyz);
>> V = (4-0)*(2-(-1))*(16-0)*n/100000
V =
    7.6454
```



(2) 等分布序列的蒙特卡洛积分法

信阳师范学院 数学与统计学院

例：求解 $\int_1^2\int_{\sqrt{x}}^{2x}\int_{xy}^{2x+y} x\sin(y+z)(x+z^2)dzdydx$

```
fh = @(x)x(:,1).*sin(x(:,2)+x(:,3)).*(x(:,1)+x(:,3).^2);
n = 100000000;
data = bsxfun(@times, repmat(1:n,3,1),[sqrt(2);sqrt(3);sqrt(6)/3]);
data = mod(data,1);
x(:,1) = data(1,:)+1;
x(:,2) = 3*data(2,:)+1;
x(:,3) = 7*data(3,:)+1;
ind = (x(:,2)>=sqrt(x(:,1))&x(:,2)<=2*x(:,1)&x(:,3)>=x(:,1).*x(:,2)&x(:,3)<=2*x(:,1)+x(:,2));
In = (2-1)*(4-1)*(8-1)*sum(fh(x(ind,:)))/n
In =
  12.651219558395249
```

函数integral3求解结果为：

12.653057240716713

%基本蒙特卡洛积分法求解近似解为12.6182，不如等分布序列的蒙特卡洛积分法

(2) 等分布序列的蒙特卡洛积分法

- 例：求解 $\int_0^1 \int_{e^{x_1}/2}^{e^{x_1}} \int_{(x_1+\sin x_2)/2}^{x_1+\sin x_2} \int_{(x_1+x_3)/2}^{x_1+x_3} \int_{x_4}^{2x_4} \sin\left(x_1 e^{x_2\sqrt{x_3}}\right) + x_4^{x_5} dx_5 dx_4 dx_3 dx_2 dx_1$

$$0 \le x_1 \le 1, \frac{1}{2} \le x_2 \le e, \frac{\sin(e)}{2} \le x_3 \le 2, \frac{\sin(e)}{4} \le x_4 \le 3, \frac{\sin(e)}{4} \le x_5 \le 6,$$

```
fh = @(x)sin(x(1,:).*exp(x(2,:).*sqrt(x(3,:))))+x(4,:).^x(5,:);
n = 1000000;
data = bsxfun(@times, repmat(1:n,5,1),[sqrt(2); sqrt(3); sqrt(6)/3; sqrt(10); sqrt(19)]);
x = mod(data,1);
BminusA = diff([0.5 sin(exp(1))/2 sin(exp(1))/4 sin(exp(1))/4;exp(1) 2 3 6])';
x(2:end,:) = bsxfun(@times,x(2:end,:),BminusA);
x(2:end,:) = bsxfun(@plus,x(2:end,:),[0.5;sin(exp(1))/4*[2;1;1]]);
ind = (x(2,:)>=exp(x(1,:))/2&x(2,:)<=exp(x(1,:))&x(3,:)>=(x(1,:)+sin(x(2,:)))/2&x(3,:)<=(x(1,:)+sin(x(2,:)))...
    &x(4,:)>=(x(1,:)+x(3,:))/2&x(4,:)<=x(1,:)+x(3,:)&x(5,:)>=x(4,:)&x(5,:)<=2*x(4,:));
I = (exp(1)-0.5)*(2-sin(exp(1))/2)*(3-sin(exp(1))/4)*(6-sin(exp(1))/4)*sum(fh(x(:,ind)))/n
% I = 5.898580893966090
```

信阳师范学院 数学与统计学院 SCHOOL OF MATHEMATICS AND STATISTICS

EDUCATION
BUSINESS
01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
PHOTO BOOK

感谢聆听