第**8**章 数值积分

目录

8.1 插值型求积方法

一元函数定积分的数学表示为

$$I=\int_a^b f(x)\mathrm{dx} \qquad (8-1)$$

又有Newton-Leibniz公式

$$\int_a^b f(x)\mathrm{dx}=F(b)-F(a) \qquad (8-2)$$

用插值多项式$L_n(x)$替换$(8-1)$中的被积函数$f(x)$，然后计算$I=\int_a^b L_n(x)\mathrm{dx}$作为积分的近似值，这样建立的求积公式称为插值型求积公式。

设给定一组节点$a=x_0<x_1<\ldots<x_n=b$，且$f(x)$在这些节点上的值为$f(x_k)$，则可作n次插值多项式$L_n(x)=\sum_{k=0}^n f(x_k)l_k(x)$，其中$l_k(x)=\prod_{j=0,j\neq k}^n \frac{x-x_j}{x_k-x_j}$是插值基数。

用$L_n(x)$替换$(8-1)$中的$f(x)$，有

$$I_n(f)=\int_a^b L_n(x)\mathrm{dx}=\int_a^b \sum_{k=0}^n f(x_k)l_k(x)\mathrm{dx}\sum_{k=0}^n\left[\int_a^b l_k(x)\mathrm{dx}\right]f(x_k)=\sum_{k=0}^n A_k f(x_k) \qquad (8-3)$$

其中$A_k=\int_a^b l_k(x)\mathrm{dx}$。

下面讨论A_k的计算，把积分区间$[a,b]$分为n等份，令步长$h=\frac{b-a}{n}$，则$x_k=x_0+\mathrm{kh}(k=0,1,2,\ldots,n)$，作变换$t=\frac{x-x_0}{h}$，代入$A_k$中得

$$A_k=\int_a^b l_k(x)\mathrm{dx}=h\int_0^n \prod_{j=0,j\neq k}^n \frac{t-j}{k-j}\mathrm{dt}=\frac{(-1)^{n-k}h}{(n-k)!k!}\int_0^n \prod_{j=0,j\neq k}^n (t-j)\mathrm{dt}\ \ (k=0,1,2,\ldots,n) \qquad (8-4)$$

这种等距节点的插值型求积公式通常称为Newton-Cotes公式。

8.1.1 梯形求积公式

在Newton-Cotes公式中取$n=1, x_0=a, x_1=\mathrm{b}$，则

有$A_0=\frac{(-1)^1(b-a)}{1!0!}\int_0^1(t-1)\mathrm{dt}\ =\frac{b-a}{2}$，$A_1=\frac{(-1)^0(b-a)}{1!0!}\int_0^1(t-0)\mathrm{dt}\ =\frac{b-a}{2}$，于是得到如下两个求积节点的插值型求积公式 $I_1(f)=\frac{b-a}{2}f(x_0)+\frac{b-a}{2}f(x_1)=\frac{b-a}{2}[f(x_0)+f(x_1)]$，称

$$T\left(f\right)=\frac{b-a}{2}[f(a)+f(b)] \qquad (8-5)$$

为梯形公式。容易证明，梯形公式的误差为$R_T(f)=-\frac{(b-a)^3}{12}f''(\eta)\ (a<\eta<b)$
$(8-6)$

由$(8-6)$知，梯形公式的代数精度仅为1，比较低，所以实际应用中往往不采用梯形公式而采用复化梯形公式。

把积分区间$[a,b]$分为n等份，分点为$x_k=a+\mathrm{kh}(k=0,1,2,\ldots,n)$，其中$h=\frac{b-a}{n}$，在每个小区间$[x_k,x_{k+1}]$上利用梯形公式，则有$\int_a^b f(x)\mathrm{dx}=\sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}}f(x)\mathrm{dx}\approx\sum_{k=0}^{n-1}\frac{h}{2}[f(x_k)+f(x_{k+1})]$，记

$$T_n(f)=\sum_{k=0}^{n-1}\frac{h}{2}[f(x_k)+f(x_{k+1})]=\frac{h}{2}\left[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_k)\right] \qquad (8-7)$$

公式$(8-7)$称为复化梯形公式，对应程序trape.m如下

```
function I=trape(fun,a,b,n,varargin)
% 复化梯形公式求解数值积分
% 输入参数:
%     ---fun: 被积函数
%     ---a,b: 积分区间的端点
%     ---n: 区间等间距分割区间的数目
%     ---: p1,p2,⋯: 函数fun的附加参数
% 输出参数:
%     ---I: 求得的积分值
if nargin<4|isempty(n)
n=1e4;
end
h=(b-a)/n;  % 步长
x=a+h*(0:n);  % 节点
fx=feval(fun,x,varargin{:});  % 求取x处的函数值
```

I=h*[(fx(1)+fx(end))+2*sum(fx(2:end-1))]/2; % 公式8-9

end

该函数的调用格式为：I=trape(fun,a,b,n,p1,p2,...)

【例8-1】已知蹦极过程可以由该微分方程模型$\frac{\mathrm{dv}}{\mathrm{dt}}=g-\frac{c_d}{m}v^2$描述，其中$v$为垂直速度$(m/s)$，$g$是重力加速度$(\approx 9.81m/s^2)$，$c_d$是二阶阻尼系数$(=0.25\mathrm{kg}/m)$，$m$为蹦极运动员的质量$(68.1\mathrm{kg})$，试利用复化梯形公式求出自由落体蹦极者在第$3s$时的下降速度以及最初$3s$内下降的高度。

```
v=simplify(dsolve('Dv=g-c_d/m*v^2','v(0)=0','t'))  % 利用函数dsolve求微分方程解析解
f=eval(['@(t,g,m,c_d)',char(v)])  % 将符号表达式写成匿名函数
g=9.81;m=68.1;c_d=0.25;
v_3=subs(v,{'t','g','m','c_d'},{3,g,m,c_d})  % 第3秒的下降速度
x=trape(f,0,3,[],g,m,c_d)  % 最初3秒内下降的高度
```

trape.m是根据已知函数原型编写的，而实际问题中更常见的是已知一组数据而不知道函数表达式，这时候trape()就无能为力了，下面仍根据式$(8-7)$编写复化梯形公式求解已知离散数据点的数值积分问题的程序trape_quad.m，具体内容如下：

function I=trape_quad(x,y)

% 复化梯形公式求解数值积分(已知离散数据点)

% 输入参数：

%   ---x：被积函数自变量的等距节点

%   ---y：被积函数在节点处的函数值

% 输出参数：

%   ---I：求得的积分值

m=length(x);n=length(y);

if m~=n

error('x和y的长度必须相等！')

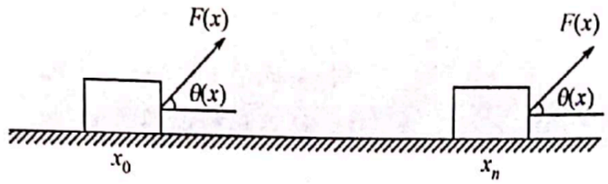
end

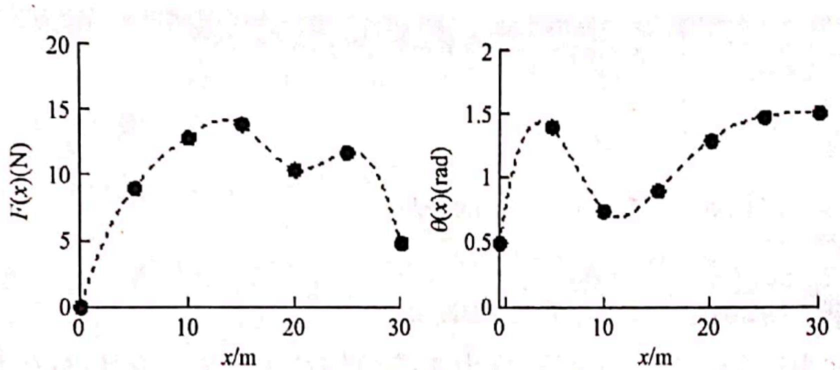
h=(x(m)-x(1))/(m-1); % 步长

I=h*[1,2*ones(1,m-2),1]*y'/2; % 公式8-9的向量形式

该函数的调用格式为I=trape_quad（x，y）

【例8-2】假设某物体在变力$F(x)$的作用下，由x_0运动到x_n，其中$F(x)$和$\theta(x)$都是连续函数，如图所示





这里只有x=5m的离散测量值，如表所示

x(m)	F(x)(N)	θ(x)(rad)
0	0.0	0.50
5	9.0	1.40
10	13.0	0.75
15	14.0	0.90
20	10.5	1.30
25	12.0	1.48
30	5.0	1.50

求解这一过程中$F(x)$做功W。

```
x=0:5:30;
theta=[0.50 1.40 0.75 0.90 1.30 1.48 1.50];
F=[0 9 13 14 10.5 12 5];
F_theta=F.*cos(theta);
xi=0:30;  % 将数据区间进行间距1m的采样以进行插值操作
thetai=spline(x,theta,xi);  % 对theta进行三次样条插值
Fi=spline(x,F,xi);  % 对F进行三次样条插值
F_thetai=Fi.*cos(thetai);
I_R=trape_quad(xi,F_thetai)  % 计算插值后的数值积分
n=[1 2 3 6];  % 区间等分数向量
for k=1:length(n)
    I(k)=trape_quad(x(1:n(k):end),F_theta(1:n(k):end));  % 求解各种情况下的积分值
end
I=fliplr(I)  % 将向量I反转
err=abs(I-I_R)/I_R  % 计算相对误差
```

8.1.2 辛普森求积公式

在Newton-Cotes公式中

取$n=2, x_0=a, x_1=\frac{a+b}{2}, x_2=b$, 则有 $A_0=\frac{(-1)^2 h}{2!0!}\int_0^2 (t-1)(t-2)\mathrm{d}t=\frac{b-a}{6}$, $A_1=\frac{(-1)^1 h}{1!1!}\int_0^2 t(t-2)\mathrm{d}t=\frac{4(b-a)}{6}$, $A_2=\frac{(-$

###3#############

$$I_2(f)=\frac{b-a}{6}f(x_0)+\frac{4(b-a)}{6}f(x_1)+\frac{b-a}{6}f(x_2)=\frac{b-a}{6}[f(x_0)+4f(x_1)+f(x_2)]$$

称

$$S(f)=\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right] \qquad (8-8)$$

#####Simpson####

##############$R_s(f)=-\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\eta)\ (a<\eta<b)$。 ######$(8-8)$############3#

############################

把积分区间$[a,b]$分为n等份，分点为$x_k=a+\mathrm{kh}(k=0,1,2,...,n)$，其中$h=\frac{b-a}{n}$。记区间$[x_k,x_{k+1}]$的中点为$x_{k+\frac{1}{2}}$, ######$[c,x_{k+1}]$###########$\int_a^b f(x)\mathrm{dx}=\sum_{k=0}^{n-1}\frac{x_{k+1}-x_k}{6}\left[f(x_k)+4f\left(x_{k+\frac{1}{2}}\right)+f(x_{k+1})\right]$，记

$$S_n(f)=\int_a^b f(x)\mathrm{dx}=\frac{h}{6}\left[f(a)+2\sum_{k=1}^{n-1}f(x_k)+4\sum_{k=0}^{n-1}f\left(x_{k+\frac{1}{2}}\right)+f(b)\right] \qquad (8-9)$$

##$(8-9)$##########

####$(8-9)$########$S_n(f)=\frac{h}{3}\left[\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1}f(x_k)\right]+\frac{2h}{3}\sum_{k=0}^{n-1}f\left(x_{k+\frac{1}{2}}\right)=\frac{1}{3}T_n+\frac{2}{3}H_n \qquad (8-10)$

其中$H_n=h\sum_{k=0}^{n-1}f\left(x_{k+\frac{1}{2}}\right)$。

根据公式可以编写复化梯形求积的程序simpson.m，具体内容如下：

```
function I=simpson(fun,a,b,n,varargin)
% 复化辛普森公式求解数值积分(已知函数原型)
% 输入参数：
%     ---fun：被积函数
%     ---a,b：积分区间的端点
%     ---n：区间等间距分割区间的数目
%     ---：p1,p2,···：函数fun的附加参数
% 输出参数：
```

%　---I：求得的积分值

if nargin<4|isempty(n)

n=1e4;

end

h=(b-a)/n;

x=a+h*(0:n);

fx=feval(fun,x,varargin{:});

I=h*[fx(1)+fx(end)+2*sum(fx(2:end-1))+...

4*sum(feval(fun,(x(1:end-1)+x(2:end))/2,varargin{:}))]/6; % 公式8-11

##########I=simpson(fun,a,b,p1,p2,...)

##8-3##############8-1########

```
v=simplify(dsolve('Dv=g-c_d/m*v^2','v(0)=0','t'))  % 微分方程解析解
f=eval(['@(t,g,m,c_d)',char(v)])  % 将符号表达式写成匿名函数
g=9.81;m=68.1;c_d=0.25;
v_3=subs(v,{'t','g','m','c_d'},{3,g,m,c_d})  % 第3秒的下降速度
x=simpson(@(t)f(t,g,m,c_d),0,3)  % 最初3秒内下降的高度
% x=simpson(f,0,3,[],g,m,c_d)
```

#########$[x_k,x_{k+1}]$#########################2n+1#########$x_k=a+k\frac{h}{2}(k=0,1,2,\dots,2n)$，经推导得到

$$S_n(f)=\int_a^b f(x)\mathrm{d}x=\frac{h}{6}\left[f(a)+2\sum_{k=1}^{n-1}f(x_{2k})+4\sum_{k=0}^{n-1}f(x_{2k+1})+f(b)\right] \quad (8-11)$$

###$(8-11)$###############################simp_quad.m########

```
function I=simp_quad(x,y)

% #############(#######)

% #####

%       ---x#############

%       ---y#############

% #####

%       ---I#######

m=length(x);n=length(y);

if m~=n
```

```
error('x#y########')

end

if rem(n-1,2)~=0

% ##n-1###2##############

warning('########Simpson###########');

I=trape_quad(x,y);  % ##########

return;

end

N=(n-1)/2; h=(x(n)-x(1))/N;

a=[1,reshape([4*ones(1,N-1);2*ones(1,N-1)],1,[]),4,1];  % ##8-11#########

I=h/6*a*y';  % ##8-11#######

##########I=simp_quad(x,y)
```

##8-4############## $I=\int_{-1}^{1}e^{-x^2}\mathrm{d}x$ #####0.1#

#######

```
x=-1:0.1:1;
y=exp(-x.^2); %计算函数值
I=simp_quad(x,y) %调用函数simp_quad求解
```

8.1.3 Cotes公式

#Newton-Cotes#### $n=4, x_0=a, x_1=x_0+h, x_2=x_0+2h, x_3=x_0+3h, x_4=b$，则有 $A_0=A_4=\frac{7}{90}(b-a), A_1=A_3=\frac{32}{90}(b-a), A_2=\frac{12}{90}(b-a)$，故Cotes公式为

$$C(f)=\frac{b-a}{90}\left[7f(a)+32f\left(\frac{3a+4}{b}\right)+12f\left(\frac{a+b}{2}\right)+32f\left(\frac{a+3b}{4}\right)+7f(b)\right] \quad (8-12)$$

Cotes###### $R_C(f)=-\frac{8}{945}\left(\frac{b-a}{4}\right)^7 f^{(6)}(\eta)\ (a<\eta<b)$， #####Cotes########5#

###############################Cotes###

$$C_n(f)=\sum_{k=0}^{n-1}\frac{h}{90}\left[7f(x_k)+32f\left(x_{k+\frac{1}{4}}\right)+12f\left(x_{k+\frac{1}{2}}\right)+32f\left(x_{k+\frac{3}{4}}\right)+7f(x_{k+1})\right] \quad (8-13)$$

即 $C_n(f)=\frac{h}{90}\left[7f(a)+32\sum_{k=0}^{n-1}f\left(x_{k+\frac{1}{4}}\right)+12\sum_{k=0}^{n-1}f\left(x_{k+\frac{2}{4}}\right)+32\sum_{k=0}^{n-1}f\left(x_{k+\frac{3}{4}}\right)+14\sum_{k=1}^{n-1}f(x_k)+7f(b)\right]$

$(8-14)$

###$(8-12)$######Cotes############Cotes.m#######

```
function I=Cotes(fun,a,b,n,varargin)
% ##Cotes########(######)
% #####
%      ---fun#####
%      ---a,b########
%      ---n#############
%      ---p1,p2,...#fun#####
% #####
%      ---I#######
if nargin<4|isempty(n);
n=1e4;
end
h=(b-a)/n;
x=a+h*(0:n);
fx=feval(fun,x,varargin{:});
%I=h*[(7*fx(1)+7*fx(end))+32*sum(feval(fun,x(2:4:end),varargin{:}))+...
%12*sum(feval(fun,x(3:4:end),varargin{:}))+...
%32*sum(feval(fun,x(4:4:end),varargin{:}))+...
%14*sum(feval(fun,x(5:4:end),varargin{:}))]/90;
I=h*[(7*fx(1)+7*fx(end))+32*sum(feval(fun,x(1:end-1)+1/4*h,varargin{:}))+...
12*sum(feval(fun,x(1:end-1)+1/2*h,varargin{:}))+...
32*sum(feval(fun,x(1:end-1)+3/4*h,varargin{:}))+...
14*sum(feval(fun,x(1:end-1)+h,varargin{:}))]/90;  % ##8-14
```

##########I=Cotes(fun,a,b,n,p1,p2,...)

##8-5####Cotes######$I=\int_1^5 \frac{\sin x}{x}\mathrm{d}x$的值。

#######

```
f=@(x)sin(x)./x; %定义被积函数
I=Cotes(f,1,5) %Cotes法计算定积分
```

####$(8-12)$#########

$$C_n(f)=\frac{h}{90}\left[7f(a)+32\sum_{k=0}^{n-1}f(x_{4k+1})+12\sum_{k=0}^{n-1}f(x_{4k+2})+32\sum_{k=0}^{n-1}f(x_{4k+3})+14\sum_{k=1}^{n-1}f(x_{4k})+7f(b)\right]$$

$(8-15)$

######Cotes####################cots_quad.m# #######

```
function I=cotes_quad(x,y)

% ##cotes########(#######)

% #####

%       ---x#############

%       ---y#############

% #####

%       ---I#######

m=length(x);n=length(y);

if m~=n

error('x#y########')

end

if rem(n-1,4)~=0

% ##n-1###4############

warning('#######Cotes############');

I=trape_quad(x,y);

return;

end

N=(n-1)/4; h=(x(n)-x(1))/N;

a=[7,reshape([32*ones(1,N-1);12*ones(1,N-1);...

32*ones(1,N-1);14*ones(1,N-1)],1,n-5),32,12,32,7];
```

I=h/90*a*y';

##########I=cotes_quad(x,y)

##8-6#####Cotes######8-4##

#######

```
x=-1:0.1:1;
y=exp(-x.^2); %计算函数值
Icq=cotes_quad(x,y) %Cotes公式求解定积分

f=@(x)exp(-x.^2); %定义被积函数
Ic=Cotes(f,-1,1) %Cotes法计算定积分
```

8.2 自适应步长求积方法

##

8.2.1 自适应步长梯形公式

将区间$[a,b]$ n等分，步长$h_n=\dfrac{b-a}{n}$， #########

#$[x_k,x_{k+1}]$上， $T_{1k}=\dfrac{h_n}{2}(f(x_k)+f(x_{k+1}))$

在$[a,b]$上， $T(h_n)=T_n=\sum_{k=0}^{n-1}T_{1k}=\dfrac{h_n}{2}(f(a)+f(b))+h_n\sum_{k=1}^{n-1}f(x_k)$

##n###############$[x_k,x_{k+1}]$####$x_{k+\frac{1}{2}}$，则分成两个小区间$\left[x_k,x_{k+\frac{1}{2}}\right]$和$\left[x_{k+\frac{1}{2}},x_{k+1}\right]$，于是###$n$##$2n$###$h_n$##

#$h_{2n}=\dfrac{h_n}{2}$。

#$[x_k,x_{k+1}]$上， $T_{2k}=\dfrac{h_{2n}}{2}\left(f(x_k)+f\left(x_{k+\frac{1}{2}}\right)\right)+\dfrac{h_{2n}}{2}\left(f\left(x_{k+\frac{1}{2}}\right)+f(x_{k+1})\right)=\dfrac{1}{2}T_{1k}+\dfrac{h_n}{2}f\left(x_{k+\frac{1}{2}}\right)$

在$[a,b]$上， $T\left(\dfrac{h_n}{2}\right)=T_{2n}=\sum_{k=0}^{n-1}T_{2k}=\dfrac{1}{2}T_n+\dfrac{h_n}{2}\sum_{k=0}^{n-1}f\left(x_{k+\frac{1}{2}}\right)=\dfrac{1}{2}T_n+\dfrac{H_n}{2}$

##################$T_{2n}=\dfrac{1}{2}T_n+\dfrac{H_n}{2}$ $\qquad(8-16)$

###########

$$
\begin{cases} T_1 = \dfrac{b-a}{2}[f(a)+f(b)] \\ T_{2n} = \dfrac{1}{2}T_n + \dfrac{b-a}{2n}\sum_{j=0}^{n-1} f\left(a+(2j+1)\dfrac{b-a}{2n}\right) \quad (n=1,2,\dots) \end{cases} \tag{8-17}
$$

直到$|T_{2n}-T_n| \le \varepsilon$为止，$T_{2n}$作为积分的近似值。

```
##########################adapt_trape.m########

function I=adapt_trape(fun,a,b,eps,varargin)

% #############

% #####

%       ---fun#####

%       ---a,b########

%       ---eps#########1e-6

%       ---p1,p2,...#fun#####

% #####

%       ---I#######

if nargin<4|isempty(eps);

eps=1e-6;

end

N=1;h=b-a;

T=h/2*sum(feval(fun,[a,b],varargin{:}));

tol=1;

while tol>eps

h=h/2;I=T/2;

x=a+(2*(1:N)-1)*h;

fx = feval(fun,x,varargin{:}); % #####

I=I+h*sum(fx);

tol=abs(I-T);

N=N*2; T=I;

end

##########I=adapt_trape(fun,a,b,eps,pl,p2,...)
```

##8-7################8-5##

#######

```
f=@(x)sin(x)./x %定义被积函数
I=adapt_trape(f,1,5) %自适应梯形法求定积分
```

8.2.2 自适应步长辛普森公式

#(8-11)#(8-16)###$S_n = \frac{1}{3}T_n + \frac{2}{3}(2T_{2n} - T_n) = \frac{1}{4^1 - 1}(4^1 T_{2n} - T_n)$ $(8-18)$

#(8-18) ################### ######Simpson#######################

################MATLAB##adapt_simp.m######

```
function I=adapt_simp(fun,a,b,eps,varargin)

% ##############

% #####

%       ---fun#####

%       ---a,b########

%       ---eps#########1e-6

%       ---p1,p2,...#fun#####

% #####

%       ---I#######

if nargin<4|isempty(eps);

eps=1e-6;

end

N=1;h=b-a;

T=h/2*sum(feval(fun,[a,b],varargin{:}));

tol=1;S=T;

while tol>eps

h=h/2;T1=T/2;

x=a+(2*(1:N)-1)*h;

fx = feval(fun,x,varargin{:}); % #####

T1=T1+h*sum(fx);
```

I=(4*T1-T)/3;

tol=abs(I-S);

N=N*2;T=T1;S=I;

end

##########

I=adapt_simp(fun,a,b,eps,p1,p2,...)

##8-8################8-5########$\varepsilon = 10^{-5}$。

#######

```
f=@(x,a)sin(x)./x; %定义被积函数
I=adapt_simp(f,1,5)  %自适应步长辛普森法计算定积分
```

8.2.3 自适应步长Cotes公式

############################ $$C_n = \frac{4^2 S_{2n} - S_n}{4^2 - 1} \qquad (8-19)$$

#(8-19) #######Cotes##########Cotes###############

#########Cotes#######adapt_Cotes.m######

```
function I=adapt_Cotes(fun,a,b,eps,varargin)

% ###Cotes########

% #####

%       ---fun#####

%       ---a,b########

%       ---eps##########1e-6

%       ---p1,p2,...#fun#####

% #####

%       ---I#######

if nargin<4|isempty(eps);

eps=1e-6;

end

N=1;h=b-a;
```

```
T=h/2*sum(feval(fun,[a,b],varargin{:}));
tol=1;C=T;
while tol>eps
h=h/2;T1=T/2;
x=a+(2*(1:N)-1)*h;
fx = feval(fun,x,varargin{:}); % #####
T1=T1+h*sum(fx);
T2=(4*T1-T)/3;
I=(16*T2-T1)/15;
tol=abs(I-C);
N=N*2;T=T1;T1=T2;C=I;
end
end
```

##########I=adapt_Cotes(fun,a,b,eps,p1,p2,...)

##8-9########### $f(t,\xi)=\frac{e^{-\xi t}}{\cos\alpha}\cos\left(t\sqrt{1-\xi^2}+\alpha\right)$，其中 $\alpha=\arctan\frac{-\xi}{\sqrt{1-\xi^2}}$, #### $I=\int_0^{20}f(t,0.1)\mathrm{dt}$。

#############Dampedsinewave.m######

```
function f=Dampedsinewave(t,xi)
alpha=atan(-xi/sqrt(1-xi^2)); %##alpha
f=exp(-xi*t).*cos(t*sqrt(1-xi^2) +alpha)/cos(alpha) %####
end
```

########:

```
format long
xi=0.1;
I=adapt_Cotes(@Dampedsinewave,0,20,[],xi)  %数值解
```

8.2.4 Romberg求积公式

##(8-16)##(8-18)##(8-19)##
#Cotes### #Cotes##############Cotes##############

$$R_n=\frac{4^3C_{2n}-C_n}{4^3-1} \qquad (8-20)$$

##(8-20) #Romberg######## Romberg######Cotes### ########## ############## ###Romberg######

############
$$\begin{cases} T_1^{i+1} = \frac{1}{2}T_1^i + \frac{h}{2}\sum_{k=0}^{n-1} f\left(x_{k+\frac{1}{2}}\right) \\ T_{j+1}^{i+1} = \frac{4^j T_j^{i+1} - T_j^i}{4^j - 1} \end{cases} \qquad (8-21)$$

#(8-21) ###Richardson#######

Romberg#####################################Romberg########Romberg.m# #####

```
function [I,T]=Romberg(fun,a,b,eps,varargin)
% Romberg########
% #####
%       ---fun#####
%       ---a,b########
%       ---eps#########1e-6
%       ---p1,p2,...#fun#####
% #####
%       ---I#######
%       ---T#Romberg##########
if nargin<4|isempty(eps);
eps=1e-6;
end
N=1;h=b-a;
T(1,1)=h/2*sum(feval(fun,[a,b],varargin{:}));
tol=1;
while tol>eps
h=h/2;N=2*N;k=log2(N);
x=a+(2*(1:N/2)-1)*h;
fx = feval(fun,x,varargin{:}); % #####
T(k+1,1)=1/2*T(k,1)+h*sum(fx);
for j=1:k
T(k+1,j+1)=(4^j*T(k+1,j)-T(k,j))/(4^j-1);
```

```
end

tol=abs(T(k+1,k+1)-T(k,k));

end

I=T(k+1,k+1);

end
```

##########[I,T]=Romberg(fun,a,b,eps,p1,p2,...)

##8-10###Romberg########8-9##

#######

```
xi=0.1;
[I,T]=Romberg(@Dampedsinewave,0,20,[],xi)
```

测试:

```
2.03 * 4 / 3 - 10.79 / 3

0.384 * 16 / 15 + 0.894 / 15

0.3019 * 4 / 3 - 0.3012 / 3
```

8.3 Gauss求积方法

#Newton-Cotes##(8-3) #########################*n*##*n*###########*n*#######*n*###########$n+1$####

#############################$n \to \infty$##$\sum_{k=0}^{n} A_k f(x_k)$########$\int_a^b f(x)\mathrm{dx}$#############################

####?###############

8.3.1 Gauss求积公式的构造

#Newton-Cotes####$I_n(f) = \int_a^b f(x)\mathrm{dx} = \sum_{k=0}^{n} A_k f(x_k)$ $\qquad (8-22)$

#####$x_k = a + \mathrm{kh}(k = 0, 1, 2, \ldots, n)$############$A_k$###########$x_k$################(8-22)####2n+1######

$f(x) = 1, x, x^2, \ldots, x^{2n+1}$#######$A_k$###$x_k$###############

$$\int_{-1}^{1} f(x)dx = f(\frac{1}{\sqrt{3}}) + f(-\frac{1}{\sqrt{3}})$$

####Gauss###

#$P_n(x)(n = 0, 1, 2, \ldots)$#########$P_n(x)$######:

###*n*#$P_n(x)$#*n*#####

$\int_a^b \rho(x)P_i(x)P_j(x)\mathrm{dx} = 0, (i \neq j)$;

##########n-1#### $P(x)$ ## $\int_a^b \rho(x)P(x)P_n(x)\mathrm{dx} = 0, n \geq 1$

$P_n(x)$ # $[a,b]$ ##n######

############ $x_0, x_1, \ldots, x_n$ # $n+1$ ###### $P_{n+1}(x)$ # $n+1$ ############## $\int_a^b \rho(x)f(x)\mathrm{dx} \approx \sum_{k=0}^{n} A_k f(x_k)$ ##Gauss##

$A_k = \int_a^b \rho(x)\prod_{i=0,i\neq k}^{n}\frac{x-x_i}{x_k-x_i}\mathrm{dx}$

8.3.2 几个常用的Gauss求积公式

1.Gauss-Legendre####

###(Legendre) ### $P_{n+1}(x) = \frac{1}{2^{n+1}(n+1)!}\frac{d^{n+1}}{dx^{n+1}}(x^2-1)^{n+1}$ ###[-1,1]###### $\rho(x) = 1$ ######## $P_{n+1}(x)$ # $n+1$ ##

##########Gauss-Legendre#### $\int_{-1}^{1} f(x)\mathrm{dx} \approx \sum_{k=0}^{n} A_k f(x_k)$ (8 − 23)

其中 $A_k = \int_{-1}^{1}\frac{P_{n+1}(x)}{(x-x_k)P'_{n+1}(x_k)}\mathrm{dx} = \frac{2}{(1-x_k^2)[P'_{n+1}(x_k)]^2}$

n	Gauss 点 t_k	求积系数 A_k	n	Gauss 点 t_k	求积系数 A_k
0	0	2		±0.661 209 386 5	0.360 761 537 0
1	±0.577 350 269 2	1		±0.238 619 186 1	0.467 913 934 6
2	±0.774 596 669 2	0.555 555 555 6	6	±0.949 107 912 3	0.129 484 966 2
	0	0.888 888 888 9		±0.741 531 185 6	0.279 705 391 5
3	±0.861 136 311 6	0.347 854 845 1		±0.405 845 151 4	0.381 830 050 5
	±0.339 981 043 6	0.652 145 154 9		0	0.417 959 183 7
4	±0.906 179 845 9	0.236 926 885 1	7	±0.960 289 856 5	0.101 228 536 3
	±0.538 469 310 1	0.478 628 670 5		±0.796 666 477 4	0.222 381 034 5
	0	0.568 888 888 9		±0.525 535 409 9	0.313 706 645 9
5	±0.932 469 514 2	0.171 324 492 4		±0.183 434 642 5	0.362 683 783 4

Gauss-Legendre#######Gauss_legendre.m######

```
function I=Gauss_legendre(fun,a,b,m,n,varargin)

% Gauss########

% #####

%       ---fun#####

%       ---a,b########

%       ---m###Gauss##

%       ---n######
```

```
%      ---p1,p2,...#fun#####
% #####
%      ---I#######
if nargin<5|isempty(n);
n=10;
end
switch m
case 0
t=0;A=2;
case 1
t=[-0.5773502692,0.5773502692];A=[1,1];
case 2
t=[-0.7745966692,0,0.7745966692];
A=[0.5555555556,0.8888888889,0.5555555556];
case 3
t=[-0.8611363116,-0.3399810436,0.3399810436,0.8611363116];
A=[0.3478548451,0.6521451549,0.6521451549,0.3478548451];
case 4
t=[-0.9061798459,-0.5384693101,0,0.5384693101,0.9061798459];
A=[0.2369268851,0.4786286705,0.5688888889,0.4786286705,0.2369268851];
case 5
t=[-0.9324695142,-0.6612093865,-0.2386191861,...
0.2386191861,0.6612093865,0.9324695142];
A=[0.1713244924,0.3607615370,0.4679139346...
0.4679139346,0.3607615370,0.1713244924];
case 6
t=[-0.9491079123,-0.7415311856,-0.4058451514,0,...
0.4058451514,0.7415311856,0.9491079123];
A=[0.1294849662,0.2797053915,0.3818300505,0.4179591837,...
```

```
0.3818300505,0.2797053915,0.1294849662];

case 7

t=[-0.9602898565,-0.7966664774,-0.5255354099,0.1834346425,...

0.1834346425,0.5255354099,0.7966664774,0.9602898565];

A=[0.1012285363,0.2223810345,0.3137066459,0.3626837834...

0.3626837834,0.3137066459,0.2223810345,0.1012285363,];

otherwise

error('########m=7#')

end

h=(b-a)/n;I=0;

for k=1:n

F=feval(fun,h/2*t+a+(k-1/2)*h,varargin{:});

I=I+A*F';

end

I=h/2*I;

end
```

##########I=Gauss_legendre(fun,a,b,m,n,p1,p2,...)

##8-11# ##Gauss-Legendre########8-5#

```
syms x;
format short e
f=@(x) sin(x)./x; %定义被积函数
for m=1:6
I=Gauss_legendre(f,1,5,m); %Gauss-Legendre公式求积分
err(m) =abs(I-double(int(sin(x)/x,1,5))); %计算绝对误差
end
err %显示误差结果
```

2.Gauss-Lobatto####

Gauss-Lobatto#####Gauss-Legendre###########################$n-1$#############Gauss-Lobatto#
#####$2n-1$######

#$f(x)\in C[-1,1]$, #Gauss-Lobatto##### $\int_{-1}^{1} f(x)\mathrm{d}x \approx \dfrac{2}{n(n+1)}[f(1)+f(-1)]+\sum_{k=1}^{n-1} A_k f(x_k)$ $\quad(8-24)$

式中x_k为$P'_{n-1}(x)=0$的根， $P_{n-1}(x)$#$n-1$#########$A_k = \dfrac{2}{n(n-1)[P_{n-1}(x_k)]^2}$

n	x_k	求积系数 A_k	n	x_k	求积系数 A_k
0	± 1	1	3	± 1	1/10
1	± 1	1/3		0	32/45
	0	4/3		$\pm 0.654\,654$	49/90
2	± 1	1/6	4	± 1	0.066 667
	$\pm 1/\sqrt{5}$	5/6		$\pm 0.765\,055$	0.378 475
				$\pm 0.285\,232$	0.554 858

Gauss-Lobatto#######Gauss_Lobatto.m#####:

```
function I=Gauss_Lobatto(fun,a,b,m,n,varargin)
% Gauss_Lobatto########
% #####
%       ---fun#####
%       ---a,b########
%       ---m###Gauss##
%       ---n######
%       ---p1,p2,...#fun#####
% #####
%       ---I#######
if nargin<5|isempty(n);
n=10;
end
switch m
case 0
t=[-1,1];A=[1,1];
case 1
t=[-1,0,1];A=[1/3,4/3,1/3];
case 2
```

```
t=[-1,-1/sqrt(5),1/sqrt(5),1];

A=[1/6,5/6,5/6,1/6];

case 3

t=[-1,-0.654654,0,0.654654,1];

A=[1/10,49/90,32/45,49/90,1/10];

case 4

t=[-1,-0.765055,-0.285232,0.285232,0.765055,1];

A=[0.066667,0.378475,0.554858,0.554858,0.378475,0.066667];

otherwise

error('########m=4#')

end

h=(b-a)/n;I=0;

for k=1:n

F=feval(fun,h/2*t+a+(k-1/2)*h,varargin{:});

I=I+A*F';

end

I=h/2*I;

##########I=Gauss_Lobatto(fun,a,b,m,n,p1,p2,...)
```

##8-12###Gauss-Lobatto########8-5##

```
syms x;
format short e
f=@(x)sin(x) ./x; %定义被积函数
for m=1:4
I=Gauss_Lobatto(f,1,5,m); %Gauss_Lob at to公式求积
err(m)=abs(I-double(int(sin(x)/x,1,5))); %绝对误差
end
err   %显示误差结果
```

8.4 特殊函数的积分

8.4.1 震荡函数的积分

##################

$I_C(f) = \int_a^b f(x)\cos\omega x dx$ 和 $I_S(f) = \int_a^b f(x)\sin\omega x dx$，其中 $a = 0, b = 2\pi$。当 ω 很大时，$\cos\omega x$ 和 $\sin\omega x$ 在区间 (a,b) 内与X轴有很多个交点。#### #ω#### $f(x)\cos\omega x$ # $f(x)\cos\omega x$ 在区间 (a,b) 内与X轴也有很多个交点##############

##8-13############# $I(f) = \int_0^{\pi} e^x\cos 1000x dx$

```
format long
f=@(x)exp(x).*cos(1000*x); %定义被积函数
I_trape=trape(f,0,pi,1000) %复化梯形求积
```

```
I_trape =
     5.462992443166432e-05
```

```
I_simpson=simpson(f,0,pi,1000) %复化辛普森求积
```

```
I_simpson =
     1.820997484319636e-05
```

```
I_legendre=Gauss_legendre(f,0,pi,ceil(6*rand),1000) %高斯-勒让德求积
```

```
I_legendre =
     2.214067050389734e-05
```

```
syms x;
expr = exp(x) * cos(1000 * x);
F = int(expr,[0  pi])
```

F =

$$\frac{e^{\pi}}{1000001} - \frac{1}{1000001}$$

```
vpa(F)
```

ans = $0.000022140670492108776896952189415759$

8.4.2 反常（广义）积分

1.#########

############### $f(x)$ ### $[a,b)$ ####b###### $\forall\varepsilon > 0$ 且 $b - \varepsilon > a$，称极限 $\lim_{\varepsilon\to 0^+}\int_a^{b-\varepsilon} f(x)dx$ (8 – 25)

为#### $f(x)$ # $[a,b)$ ######(####)### $\int_a^b f(x)dx$。###

##8-14###### $I = \int_0^1 \frac{1}{\sqrt{x}(e^x + 1)}dx$

```
f=@(x)1./(sqrt(x).*(exp(x) +1)); %定义被积函数
I=Gauss_legendre(f,0,pi,4,1e5) %利用高斯-勒让德公式求奇异积分
```

```
I =
```

```
1.050585007009430
```

2.##########

######a###b# $f(x)$ # $[a,b]$ ######### $\lim\limits_{b\to+\infty}\int_a^b f(x)\mathrm{dx}$ #f(x)#####[a,+∞)######### $\int_a^{+\infty} f(x)\mathrm{dx}$

#############################

#1#######

########### $\int_0^{+\infty} f(x)\mathrm{dx}$ ###### $\int_0^{+\infty} f(x)\mathrm{dx}$ $\lim\limits_{b\to+\infty}\int_0^b f(x)\mathrm{dx}$ ，因此

取 $0<b_0<b_1<\ldots<b_n<\ldots$ 且 $b_n\to+\infty(n\to+\infty)$，有

$$\int_0^{+\infty} f(x)\mathrm{dx}=\int_0^{b_0} f(x)\mathrm{dx}\int_{b_0}^{b_1} f(x)\mathrm{dx}+\ldots+\int_{b_n}^{b_{n+1}} f(x)\mathrm{dx}+\ldots \qquad (8-26)$$

#(8-26)########################## $\left|\int_{b_n}^{b_{n+1}} f(x)\mathrm{dx}\right|\le\varepsilon$ ######

####### $\left|\int_{b_n}^{b_{n+1}} f(x)\mathrm{dx}\right|\le\varepsilon$ ####################

###################quad_inf.m######

```
function I=quad_inf(fun,a,b,tol,eps)
% #######
% #####
%       ---fun#####
%       ---a,b###########a<b
%       ---tol################1e-6
%       ---eps################1e-5
% #####
%       ---I#######
if nargin<5|isempty(eps);eps=1e-5;end;
if nargin<4|isempty(tol);tol=1e-6;end;
N=2;I=0;T=1;
if isinf(a)&isinf(b)
I=quad_inf(fun,-inf,0)+quad_inf(fun,0,inf);  % ####
elseif isinf(b)
```

```
while T>eps
b=a+N;
T=quadl(fun,a,b,tol);
I=I+T;
a=b; N=2*N;
end
elseif isinf(a)
while T>eps
a=b-N;
T=quadl(fun,a,b,tol);
I=I+T;
b=a; N=2*N;
end
else
I=quadl(fun,a,b,tol);
end
```

##########I=quad_inf(fun,a,b,tol,eps)

##8-15####### $I=\int_0^{\infty} e^{-x^2}\mathrm{dx}$

```
f=@(x)exp(-x.^2); %定义被积函数
I=quad_inf(f,0,inf) %求无穷积分
```

```
I =
   0.886227674998196
```

（2）变量替换

##8-16###########8-15#

####### $t=\frac{x}{1+x}\left(x=\frac{t}{1-t},\mathrm{dx}=\frac{1}{(1-t)^2}\mathrm{dt}\right)$ ### $[0,+\infty)$ ##### $[0,1]$ ### $I=\int_0^1 e^{-\left(\frac{t}{1-t}\right)^2}\frac{1}{(1-t)^2}\mathrm{dt}$

```
f=@(x)exp(-(x./(1-x)).^2)./(1-x).^2; %定义被积函数
I=simpson(f,eps,1-eps) %复化辛普森公式求积分
```

############MATLAB##eps# #####2.2204e-016#########NaN#Inf##################################

8.4.3 重积分的近似计算

##

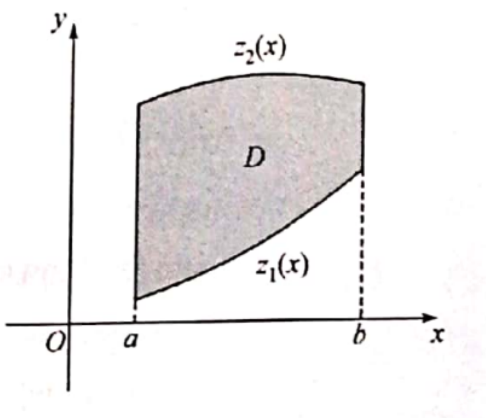


图 8-5　二重积分积分区域

$\int\int_D f(x,y)\,dx\,dy$，可化为累次积分 $\int_a^b \int_{z_1(x)}^{z_2(x)} f(x,y)\mathrm{dydx}$ (8 − 27)

###########$[a,b]$ m#####$h_x = \dfrac{b-a}{m}, x_i = a + \mathrm{ih}(i = 0,1,2,\ldots,m)$，则 $I \approx h_x\left(\dfrac{G(a)+G(b)}{2} + \sum_{i=1}^{m-1} G(x_i)\right)$，

其中 $G(x_i) = \int_{z_1(x_i)}^{z_2(x_i)} f(x_i,y)\mathrm{dy}$。再将 $[z_1(x_i), z_2(x_i)]$ 区间n等分，令 $h_y(i) = \dfrac{z_1(x_i) - z_2(x_i)}{n}$，因此 $y_{\mathrm{ij}} = z_1(x_i) + jh_y(i)(i = 0,1,\ldots,m; j = 0,1,\ldots,n)$

进而有 $G(x_i) \approx h_y(i)\left(\dfrac{f(x_i,z_1(x_i)) + f(x_i,z_2(x_i))}{2} + \sum_{j=1}^{n-1} f(x_i,y_{\mathrm{ij}})\right)$

#######################double_integra.m########

```
function I=double_integral(fun,a,b,lowfun,upfun,m,n,varargin)

% #############

% #####

%        ---fun#####(#####)

%        ---a,b,lowfun,upfun######lowfun,upfun#x###

%        ---m#[a,b]######

%        ---n#[lowfun,upfun]#####

%        ---p1,p2,...#fun#lowfun#upfun#####
```

```
% #####
%       ---I####
if nargin<7|isempty(n),n=100;end
if nargin<6|isempty(m),m=100;end
hx=(b-a)/m;
x=a+(0:m)*hx;
for i=1:m+1
ylow= feval(lowfun,x(i));
yup=feval(upfun,x(i));
hy=(yup - ylow)/n;
y(i,:)=ylow+(0:n)*hy;
f(i,:)=feval(fun,x(i),y(i,:));
G(i)=trapz(y(i,:),f(i,:));
end
I=trapz(x,G);
end
```

##########I=double_integral(fun,a,b,lowfun,upfun,m,n,p1,p2,...)

##8-17################# $I = \int\int_D (x + y + x^3e^y)\,dx\,dy$ ##*D*##### $\{(x, y)|x^2 + y^2 \le 4, y \le 0\}$ #

##########*y*##########*x*# $x + x^3e^y$ #####*y*#####*x*#######

$$I = \int\int_D (x + y + x^3e^y)\,dx\,dy = \int\int_D y\,dx\,dy = 2\int_0^2 \mathrm{dx}\int_{-\sqrt{4-x^2}}^0 \mathrm{ydy}$$

```
fun=@(x,y)y*ones(size(x)); %被积函数
lowfun=@(x)-sqrt(4-x.^2); %内部积分积分下限
upfun=@(x)zeros(size(x)); %内部积分积分上限
I=2*double_integral(fun,0,2,lowfun,upfun) %计算二重积分
```

```
I =
  -5.333200000000000
```

8.5 数值积分的MATLAB函数求解

#########MATLAB##############################MATLAB######################trapz()###quad()###quadl()####

8.5.1 trapz()函数

MATLAB##trapz()##########################I=trpaz(x,y,dim)###x#y######x###########y####### ##y######x########dim##########dim=1(###)########dm=2#######

##8-18###trapz()###### $\int_0^{\frac{3\pi}{2}} \cos 15x dx$ ######0.0001#

```
h=0.0001;
x=[0:h:3*pi/2,3*pi/2];y=cos(15*x);
I=trapz(x,y) %梯形法求积分
```

8.5.2 quad()函数

########################MATLAB###quad()###########################

[q,fcnt] =quad(fun,a,b,tol,trace,p1,p2,...)

##fun############################;a,b##########tol############ 10^{-6}#trace######[fcnt ab-aq]##trace=[]##quad########pl#p2...###fun######q#########fcnt###########

##8-19###quad()#######8-13##

```
f=@(x)1./(sqrt(x).*(exp(x)+1)); %定义被积函数
I_quad=quad(f,0,pi) %quad()函数求解定积分
```

##########,##8-13#######-###################################MATLAB#######MATLAB########### ###############

##8-20# ##quad()########8-9##

###8-9################Dampedsinewave.m#######

function f=Dampedsinewave(t,xi)

alpha=atan(-xi/sqrt(1-xi^2)); %##a i pha

f=exp(-xi*t).*cos(t*sqrt(1-xi^2)+alpha)/cos(alpha); %####

#########

```
syms t xi alpha;
f=exp(-xi*t)*cos(t*sqrt(1-xi^2)+alpha)/cos(alpha);
y=subs(f,alpha,atan(-xi/sqrt(1-xi^2)));
I1=int(y,t,0,20); %解析解
xi=0.1;format long
I2=vpa(subs(I1),30) %精确解
I=quad(@Dampedsinewave,0,20,[],[],xi) %数值解
```

###MATLAB##########quadl()#######quad()################Lobatto############quad()######### ##############

8.5.3 quadgk()函数

quadgk()###MATLAB R2007b#######Gauss-Kronrod### ###[q,errbnd]=quadgk(fun,a,b,param1,val1,param2,val2,...)

##fun##############################a#b###############-inf#inf;parami,vali#############

##8-21###quadgk()######8-15#

```
f=@(x)exp(-x.^2); %定义被积函数
I_quadgk=quadgk(f,0,inf) %quad gk函数求解无穷积分
```

########quad()########NaN###quad()############NaN#

##8-22########$I=\int_1^{10} f(x)\mathrm{d}x$ ，其中 $f(x)=\begin{cases} 1000 & x=2 \\ -100 & x=5 \\ x^5e^{-}xsinx & \text{其他} \end{cases}$

```
F=@(x)x.^5.*exp(-x).*sin(x); %定义被积函数
[q,errbnd]=quadgk(F,1,10,'Waypoints',[2 5]) %其中2.5为间断点
```

##8-23#########$I=\int_2^{6-\mathrm{j}5} e^{-x^2-\mathrm{j}x}\sin(7+\mathrm{j}2)x\mathrm{d}x$

```
format long
i=sqrt(-1);f=@(x)exp(-x.^2-i*x).*sin((7+2i)*x); %定义被积函数
I-quadgk(f,2,6-5i) %调用quad gk函数求解复数积分问题
syms x;f=exp(-x^2-i*x)*sin((7+2i)*x);
I0=vpa(int(f,2,6-5i)) %求解析解
```

##8-24#############$I(f)=\int_0^{\pi} e^x\cos 1000x\mathrm{d}x$

####quad()###quadgk()####:

```
syms x;
I_accurate=vpa(int(exp(x)*cos(1000*x),x,0,pi)) %精确值
f=@(x)exp(x).*cos(1000*x);
I_quad=quad(f,0,pi) %quad函数求解
I_quadgk=quadgk(f,0,pi,'MaxIntervalCount',1000) %quad gk函数求解
```

##############################quad()##############quadgk() ###############

8.5.4 dblquad()函数

############$\int_{y_m}^{y_M}\int_{x_m}^{x_M} f(x,y)\mathrm{d}x\mathrm{d}y$ ，###MATLAB###dblquad()###############

q=dblquad(fun,xmin,xmax,ymin,ymax,tol#@quadl,pl,p2,...)########@quadl##################
#@quad############################quadl()################quadl()#

##8-25####### $\int_{-1}^{1}\int_{-2}^{2}e^{-\frac{x^2}{2}}\sin(x^2+y)\mathrm{dxdy}$

```
format long
f=@(x,y)exp(-x.^2/2).*sin(x.^2+y); %定义被积函数
I=dblquad(f,-2,2,-1,1) %dblquad函数求解二重积分
```

######################## $\int_{x_m}^{x_M}\int_{y_m(x)}^{y_M(x)} f(x,y)\mathrm{dxdy}$

################################MATLAB################# ###MATLAB R2009a### ######quad2d() ####################

q=quad2d(fun,a,b,c,d,param1,val1,param2,val2,...)

##fun###### a,b,c,d ###### $[a,b]\times[c(x),d(x)]$ #parami#vali############

##8-26######## $I=\int_{-\frac{1}{2}}^{1}\int_{-\sqrt{1-\frac{x^2}{2}}}^{\sqrt{1-\frac{x^2}{2}}} e^{-\frac{x^2}{2}}\sin(x^2+y)\mathrm{dxdy}$

```
format long
fh=@(x)sqrt(1-x.^2/2); fl=@(x)-sqrt(1-x.^2/2); %定义内部积分的上下限
f=@(x,y)exp(-x.^2/2).*sin(x.^2+y); %定义被积函数
I1=double_integral(f,-1/2,1,fl,fh) %梯形法求一般区域的二重积分
I2=quad2d(f,-1/2,1,fl,fh) %MATLAB自带函数求解一般区域的二重积分
```

##8-27######## $I=\int_{-1}^{1}\int_{-\sqrt{1-\frac{x^2}{2}}}^{\sqrt{1-\frac{x^2}{2}}} |x^2+y^2-0.25|\mathrm{dxdy}$

```
fun=@(x,y)abs(x.^2+y.^2-0.25); %定义被积函数
c=@(x)-sqrt(1-x.^2); %内部积分下限函数
d=@(x)sqrt(1-x.^2); %内部积分上限函数
I=quad2d(fun,-1,1,c,d,'AbsTol',1e-8,...
'FailurePlot',true,'Singular',false) %一般区域二重积分求解
```

8.5.5 triplequad()函数

MATLAB###triple quad(################# $\int_{x_m}^{x_M}\int_{y_m}^{y_M}\int_{z_m}^{z_M} f(x,y,z)\mathrm{dxdydz}$, ##############

q=triplequad(fun#xmin#xmax#ymin#ymax#zmin#zmax#tol#@quadl#pl#p2#...)

##############dblquad() ########

##8-28######### $I=\int_0^2\int_0^\pi\int_0^\pi 4xze^{-x^2y-z^2}\mathrm{dzdydx}$

```
format long
f=@(x,y,z)4*x.*z.*exp(-x.^2.*y-z.^2);%定义被积函数
I=triplequad(f,0,2,0,pi,0,pi,1e-7,@quadl)%triple quad函数求解三重积分
```