Chap4

ShangXiaojin

2022 年 12 月 14 日

目录

0.1 Numerical Differentiation and Integration

0.1.1 Ex

Use the formulas given in this section to determine , as accurately as possible , approximations for each missing entry in the following tables

x	$f(x)$	$f'(x)$
2.1	-1.709847	
2.2	-1.373823	
2.3	-1.119214	
2.4	-0.9160143	
2.5	-0.7470223	
2.6	-0.6015966	

0.1.2 code

```
tic
format long
X = 2.1:0.1:2.6;
F = -[1.709847 1.373823 1.119214 0.9160143 0.7470223 0.6015966];
dig(X,F)
toc

function k = dig(X,F)
n = size(X,2);
h = X(2) - X(1);
k = zeros(1,n);
for i =  1:n
    if i >= 3 && n-i>= 2
        k(i) = (F(i-2) - 8*F(i-1) + 8*F(i+1) - F(i+2))/12*h;
    elseif i<=2
        k(i) = (-25*F(i) + 48*F(i+1) - 36*F(i+2) +16*F(i+3) - 3*F(i+4))/12*h;
    elseif n-i< 2
        k(i) = (25*F(i) - 48*F(i-1) + 36*F(i-2) - 16*F(i-3) + 3*F(i-4))/12*h;

    end
end
end
```

0.1.3 Ans

x	$f(x)$	$f'(x)$
2.1	-1.709847	0.038993
2.2	-1.373823	0.028769
2.3	-1.119214	0.022497
2.4	-0.9160143	0.018378
2.5	-0.7470223	0.015442
2.6	-0.6015966	0.013555

0.2 Richardson's Extrapolation

0.2.1 Ex

- The forward-difference formula can be expressed as

$$f'(x_0) = \tfrac{1}{h}[f(x_0+h) - f(x_0)] - \tfrac{h}{2}f''(x_0) - \tfrac{h^2}{6}f'''(x_0) + O(h^3)$$

Use extrapolation to derice an $O(h^3)$ formula for $f'(x_0)$

- Suppose that $N(h)$ is an approximation to M for every M > 0 and that

$$M = N(h) + K_1h + K_2h^2 + K_3h^3 + \dots$$

for some constants $K_1, K_2, K_3, \dot{U} se the value N(h), N(\tfrac{h}{3}), and N(\tfrac{h}{9}) to produce an O(h^3)$

0.2.2 Ans

Ex_1

$$f'(x_0) = \frac{1}{h}[f(x_0+h) - f(x_0)] - \frac{h}{2}f''(x_0) - \frac{h^2}{6}f'''(x_0) + O(h^3)$$

let $\frac{h}{2} = h$ such that

$$f'(x_0) = \frac{2}{h}[f(x_0 + \frac{h}{2}) - f(x_0)] - \frac{h}{4}f''(x_0) - \frac{h^2}{24}f'''(x_0) + O(h^3)$$

$$f'(x_0) = (2) - (1) = (4f(x_0 + \frac{h}{2}) - f(x_0+h) - 3f(x_0))/h + \frac{h^2}{12}f'''(x_0) + O(h^3)$$

Similaly, it can be obtained

$$f'(x_0) = (f(x_0+h) + 32f(x_0 + \frac{h}{4}) - 12f(x_0 + \frac{h}{2}) - 21f(x_0))/h + O(h^3)$$

Ex_2

$$M = N(h) + K_1h + K_2h^2 + K_3h^3 + \dots \tag{1}$$

$$N_2(h) = \frac{3N(\frac{h}{3}) - N(h)}{2} \tag{2}$$

$$N_3(h) = \frac{27N(\frac{h}{9}) - 12N(\frac{h}{3}) + N(h)}{16} \tag{3}$$

0.3 CompositeNumerical Integration

0.3.1 Ex

Determine the values of n and h required to approximate

$$\int_0^2 e^{2x}\sin(3x)dx$$

to within 10^{-4}.Use

- Composite Trapezoidal rule
- Composite Simpson's rule
- Composite Midpoint rule

0.3.2 code

function

- Composite Trapezoidal

```
function t = Com_Trap(f,x,x0,x1,n)
t = subs(f,x,x0) + subs(f,x,x1);
h = (x1- x0 )/n;
t1 = 0;
for i = 1:n-1
    t1 = t1 + subs(f,x,x0 + i*h);
end
t = vpa((t + 2*t1)*(h/2));
end
```

- Composite Midpoint

```
function p = Com_mid(f,x,x0,x1,n)
k = (x1-x0)/n;
p = 0;
for i= 1:n
    p = k*subs(f,x,x0+k/2+(i-1)*k) + p;
end
end
```

- Composite Simpson

```
function g = Inte_Simpson(f,x,x0,x1,n)
g = subs(f,x,x0) + subs(f,x,x1);
k = (x1-x0)/n;
g1 = 0;
g2 = 0;
for i = 1:n/2
    g1 = subs(f,x,x0 + (2*i-1)*k) + g1;
end
for i = 1:n/2-1
    g2 = subs(f,x,x0 + 2*i*k) +g2;
end
g = (k/3) * (g + 2*g2 + 4*g1);
end
```

main body

```
syms x
f = exp(2*x)*sin(3*x);
real = vpa(int(f,x,0,2));
t = 0;
thate1 = abs(real - vpa(Com_mid(f,x,0,2,144)))
thate2 = abs(real - vpa(Com_Trap(f,x,0,2,205)))
thate3 = abs(real - vpa(Inte_Simpson(f,x,0,2,20)))
```

0.3.3 Ans

Method	n	thate
Mid	144	0.000995
Trap	205	0.000982
Simp	20	0.000669

0.4 Romberg Integration

0.4.1 Ex

$$f(x) = \begin{cases} x^3+1 & 0 \le x \le 0.1 \\ 1.001 + 0.03(x-0.1) + 0.3(x-0.1)^2 + 2(x-0.1)^3 & 0.1 \le x \le 0.2 \\ 1.009 + 0.15(x-0.2) + 0.9(x-0.2)^2 + 2(x-0.2)^2 & 0.2 \le x \le 0.3 \end{cases} \tag{4}$$

Apply Romberg integration to the following integrals until $R_{n-1,n-1} and R_{n,n}$ agree to within 10^{-4}.

0.4.2 code

```
clear , clc
syms x
f1 = x^3 + 1;
f2 = 1.001 + 0.03*(x - 0.1) + 0.3*(x-0.1)^2 + 2*(x-0.1)^3;
f3 = 1.009 + 0.15*(x - 0.2) + 0.9*(x - 0.2)^2 + 2*(x - 0.2)^3;
t1 = Com_Trap(f1 ,x,0 ,0.1 ,1);
t2 = Com_Trap(f1 ,x,0 ,0.1 ,2);
t3= Com_Trap(f1 ,x,0 ,0.1 ,4);
R1 = [t1 0 0;
    t2 t2 + 1/3*(t2 - t1) 0 ;
    t3 t3 + 1/3*(t3 - t2) 0 ];

R1(3,3) = R1(3,2) + 1/15*(R1(3,2)-R1(2,2));

t1 = Com_Trap(f2 ,x,0.1 ,0.2 ,1);
t2 = Com_Trap(f2 ,x,0.1 ,0.2 ,2);
t3= Com_Trap(f2 ,x,0.1 ,0.2 ,4);
R2 = [t1 0 0;
    t2 t2 + 1/3*(t2 - t1) 0 ;
    t3 t3 + 1/3*(t3 - t2) 0 ];

R2(3,3) = R2(3,2) + 1/15*(R2(3,2)-R2(2,2));

t1 = Com_Trap(f3 ,x,0.2 ,0.3 ,1);
t2 = Com_Trap(f3 ,x,0.2 ,0.3 ,2);
t3= Com_Trap(f3 ,x,0.2 ,0.3 ,4);
R3 = [t1 0 0;
    t2 t2 + 1/3*(t2 - t1) 0 ;
    t3 t3 + 1/3*(t3 - t2) 0 ];

R3(3,3) = R3(3,2) + 1/15*(R3(3,2)-R3(2,2));
R = R1 + R2 + R3
```

0.4.3 Ans

$$\begin{pmatrix} 0.30275 & 0 & 0 \\ 0.30250625 & 0.302425 & 0 \\ 0.3024453125 & 0.302425 & 0.302425 \end{pmatrix} \tag{5}$$

0.5 Gaussian Quadrature

0.5.1 Ex

Approxiamate the following integrals using Gaussian Quadrature with n = 2,3,4,5 to the exact values of the integrals.

- $\int_0^1 x^2 \sin(x) dx$

0.5.2 code

```
clc, clear
format long
f =@(x) (1/2)*((x+1)/2)^2*exp(-(x+1)/2);
t2 = f(-1/sqrt(3)) + f(1/sqrt(3));
A1=[0.7745966692 0.5555555556
0.0000000000 0.8888888889
-0.7745966692 0.5555555556];
t3 = 0;t4 = 0; t5 = 0;
for i = 1:3
    t3 = t3 + A1(i,2) * f(A1(i,1));
end
A2=[0.8611363116 0.3478548451
0.3399810436 0.6521451549
-0.3399810436 0.6521451549
-0.8611363116 0.3478548451];
for i = 1:4
    t4 = t4 + A2(i,2) * f(A2(i,1));
end
A3=[0.9061798459 0.2369268850
0.5384693101 0.4786286705
0.0000000000 0.5688888889
-0.5384693101 0.4786286705
-0.9061798459 0.2369268850];
for i = 1:5
```

```
    t5 = t5 + A3(i,2) * f(A3(i,1));
end
[t2,t3,t4,t5]
```

0.5.3 Ans

n	P(x)
2	0.159410430966379
3	0.160595386815970
4	0.160602777514260
5	0.160602794113723

0.6 Mutiple Integrals

0.6.1 Ex

Use Algorithm with n=m=4,and Gaussian Quadrature with n=3 to approximate the following double integrals.

- $\int_0^{0.5}\int_0^{0.5} e^{y-x}dydx$

0.6.2 code

- Gauss

```
clc,clear
syms x
syms y
f = 1/16 * exp((y-x)/4);
A = [0.7745966692 0.5555555556
0.0000000000 0.8888888889
-0.7745966692 0.5555555556];
f1 = A(1,2) * subs(f,y,A(1,1));
f2 = A(2,2) * subs(f,y,A(2,1));
f3 = A(3,2) * subs(f,y,A(3,1));
g1 = 0;g2 = 0;g3 = 0;
for i = 1:3
    g1 = g1 + A(i,2) * subs(f1,x,A(i,1));
end
for i = 1:3
    g2 = g2 + A(i,2) * subs(f2,x,A(i,1));
end
```

```
for i = 1:3
    g3 = g3 + A(i,2) * subs(f3,x,A(i,1));
end
g = vpa(g1 + g2 + g3)
real = 0.2552519304;
theta1 = abs(real - g)
```

- Simpson

```
clc,clear
syms x
syms y
f = exp(y-x);
A = [];
for i = 1:5
    A = [A subs(f,y,(i-1)*0.125)];
end
M = Inte_Simpson(A(1),x,0,0.5,4) + Inte_Simpson(A(5),x,0,0.5,4);
for i = 2:4
    if mod(i,2) == 0
        M = M+4*Inte_Simpson(A(i),x,0,0.5,4);
    elseif mode(i,2) > 0
        M = M+2*Inte_Simpson(A(i),x,0,0.5,4);
    end
end
Ans1 = vpa((0.125/3)*M)
theta2 = abs(0.2552519304 - Ans1)
```

0.6.3 Ans

Method	outcome	thate
Guass(n=3)	0.25525192651465210169986117556611	0.0000000038853478916375842275193296766305
Simpson	0.25525262154254027459421843780846	0.00000069114254028125677303472302062701

Guass(n=3) is more accurate than Simpson.

0.7 addition