作业一

叶一腾 **PB18071465 2020 年 12 月 3 日**

第一题 关于精确值的说明，采用相邻迭代相对误差小于 $10^(-15)$ 作为判据。

(a) Matlab 程序显示如下：

```
LW = 'linewidth'; lw = 2;

F = @(x) sin(cos(sin(cos(x))));
I = zeros(22000,1);
N = linspace(2,22000,22000)';
for n = 2:30000
h = 2/(n-1);
x = linspace(-1, 1, n)';
f = F(x);
I(n) = h*(f(1)+f(n))/2;
for i=2:n-1
    I(n) = I(n)+h*f(i);
end
if (n > 1 && abs((I(n)-I(n-1))/I(n-1))<10^(-15))
    disp(n);
    disp(I(n));
    EXACT = I(n);
    break
end
end
semilogy( N, abs(I-EXACT), 'b', LW, lw),hold on
```

输出结果如图 1。

(b) Matlab 程序显示如下：

```
Lw = 'linewidth ' ; lw = 2;
F = @(x) sin(cos(sin(cos(x))));n = 9  ;
h = zeros(n,1); h(1) = 2;
I = zeros(n,1);
```

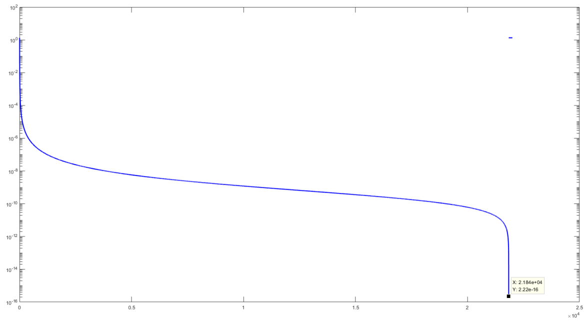

图 1: 复化梯形公式

```
%step 1
R= zeros(n);
R(1,1) = F(1)+F(-1);
I(1) = R(1,1);
%step 2
sum = zeros(n,1);
for k=2 :n
    h(k) = 2/2^(k-1);

    for i=1:(2^(k-2))
        sum(k) = sum(k)+F(-1+(2*i-1)*h(k));
    end
    R(k,1) = (R(k-1,1)+h(k-1)*sum(k))/2;
    for j=2:k
        R(k,j) = R(k,j-1)+(R(k,j-1)-R(k-1,j-1))/(4^(j-1)-1);
        I(k) = R(k,k)
    end
    if( abs(R(k,k )-R(k-1,k-1))<10^(-15) )
        exact = R(k,k);
        disp(exact);
        disp(k);
        break

    end
end
xdots = 2.^((1:n)-2);
```

```
semilogy( xdots, abs(I-exact), 'b', LW, lw),hold on
```

输出结果如图 2。

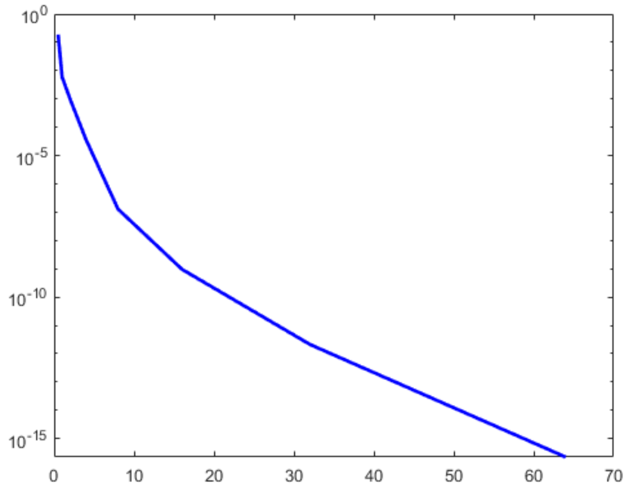

图 2: Richardson 外推方法

(c) MATLAB 程序显示如下:

```
LW = 'LineWidth'; lw = 2;

F = @(x) sin(cos(sin(cos(x))));
N = 16;
I = zeros(N,1);
nvec = 1:N; % number of the quadrature points
for n = nvec
    [x, w] = gauss(n);
    I(n) = w*F(x);
    if (n > 1 && abs((I(n)-I(n-1))/I(n-1))<10^(-15))
    disp(n);
    disp(I(n));
    EXACT = I(n);
    break
    end
end
xdots = 1:N;
semilogy( xdots, abs(I-EXACT), 'b', LW, lw),hold on
```

输出结果如图 3。

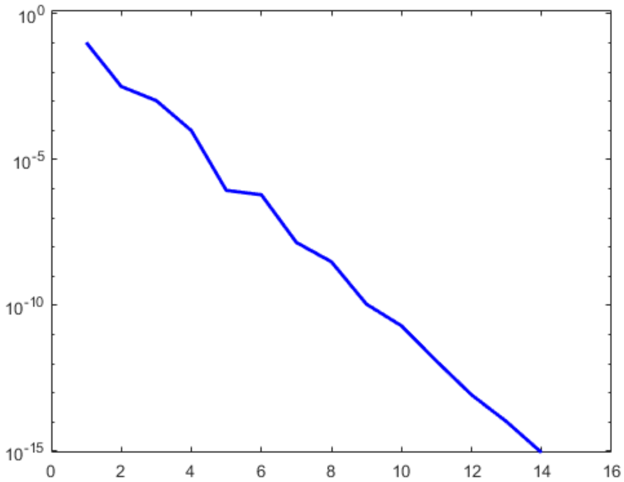

图 3: Richardson 外推方法

(d) 该内容为付费内容

第二题 函数求导与微分矩阵

(a) 证明：

由

$$\ell_j = \prod_{k=0,k\neq j}^{n} (x - x_k) \Bigg/ \prod_{k=0,k\neq j}^{n} (x_j - x_k) \tag{1}$$

直接求导可得

$$\begin{aligned} \ell_j' &= \sum_{m=0,m\neq j}^{n} \frac{\prod_{k=0,k\neq j}^{n}(x - x_k)}{x - x_m} \Bigg/ \prod_{k=0,k\neq j}^{n} (x_j - x_k) \\ &= \sum_{m=0,m\neq j}^{n} \frac{\ell_j(x)}{x - x_m} \end{aligned} \tag{2}$$

从而

$$p'(x) = \sum_{j=0}^{n} \left(f_j \ell_j(x) \sum_{k=0,k\neq j}^{n} (x - x_k)^{-1} \right) \tag{3}$$

(b) Matlab 程序显示如下：

```
LW = 'linewidth'; lw = 2;

n = 16;
```

```matlab
x = linspace(-1, 1, n)';
m = 1001;
xx = linspace(-1, 1, m)';
F = @sin;
f = F(x);
G = @cos;
g = G(xx);
dp = zeros(m, 1);
%计算公式(1)
for k = 1:n
    l = ones(m, 1);
    sum = zeros(m, 1);%求和部分
    for j = 1:k-1
        l = l.*(xx - x(j))/(x(k) - x(j));
        sum = sum+1./(xx-x(j));
    end
    for j = k+1:n
        l = l.*(xx - x(j))/(x(k) - x(j));
        sum = sum+1./(xx-x(j));
    end
    dp = dp + f(k)*l.*sum;
end
R = abs(dp - g);

figure(1)
plot(xx, G(xx), 'k', LW, lw), hold on
plot(xx, dp, 'b',LW, lw)
legend('exact', 'interpolant', 'location', 'nw')

figure(2)
plot(2)
semilogy(xx, R, 'k', LW, lw)
legend('error', 'location', 'nw')
```

输出结果如图 4和 5。

(c) 证明:

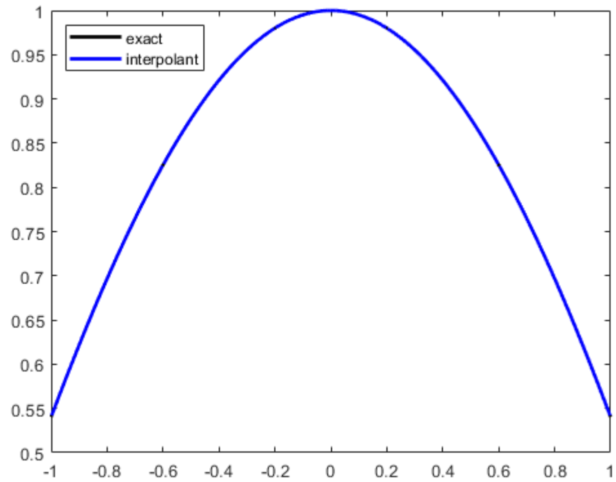
exact
interpolant
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

图 4: 结果

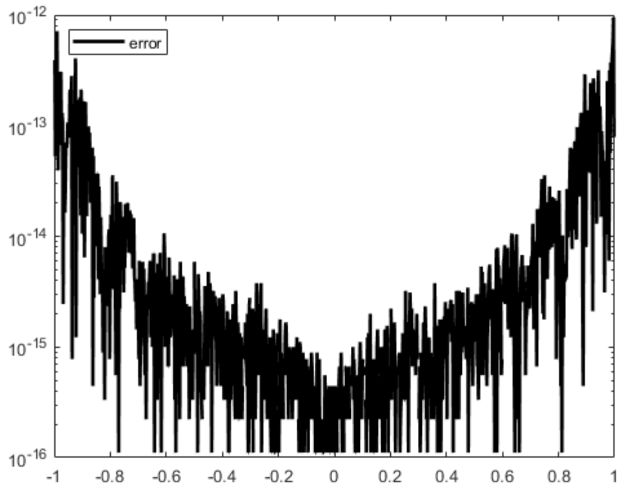
error
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-16}
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

图 5: 误差

由

$$p'(x_i) = \sum_{j=0}^{n} \left(f_j \ell_j(x) \sum_{k=0,k\neq j}^{n} (x_i - x_k)^{-1} \right) \tag{4}$$

可知

$$D_{ij}\ell_j(x_i) \sum_{k=0,k\neq j}^{n} (x_i - x_k)^{-1} \tag{5}$$

当 $i \neq j$ 时，

$$\begin{aligned} D_{ij} &= \prod_{k=0,k\neq j}^{n} (x - x_k) \sum_{k=0,k\neq j}^{n} \frac{1}{x_i - x_k} / \pi_j \Bigg|_{x=x_i} \\ &= \frac{\pi_i}{\pi_j (x_i - x_j)} \end{aligned} \tag{6}$$

当 $i = j$ 时，$\ell_j(x_i) = 1$，

$$D_{ij} = \sum_{k=0,k\neq j} (x_j - x_k)^{-1} \tag{7}$$

(d) MATLAB 程序显示如下：

```
LW = 'linewidth'; lw = 2;

F = @(x) sin(3*x.^2);
dF = @(x) 6*x.*cos(3*x.^2);

for n = 2:2:60;%节点数
x = linspace(-1, 1, n)';%生成均匀插值点
f1 = F(x);%vector f1 均匀采点
df1 = dF(x);%实际导数值
%计算导数
dp1 = calculate_matrix(x, n)*f1;
error1 = abs(dp1 - df1);
figure(1)
dot1 = semilogy(n, max(error1) ,'*r');hold on

%生成切比雪夫点
y = zeros(n,1);
for i=1:n
 y(i) = cos((i-1)*pi/(n-1));
end
```

```
f2 = F(y);%vector f2 切比雪夫采点
df2 = dF(y);%切比雪夫点实际导数值
%计算导数
dp2 = calculate_matrix(y, n)*f2;
error2 = abs(dp2 - df2);
figure(1)
dot2 = semilogy(n, max(error2) ,'*b');hold on
end
legend([dot1, dot2], '等距点', '切比雪夫点','location','sw')
```

输出结果如图 6。

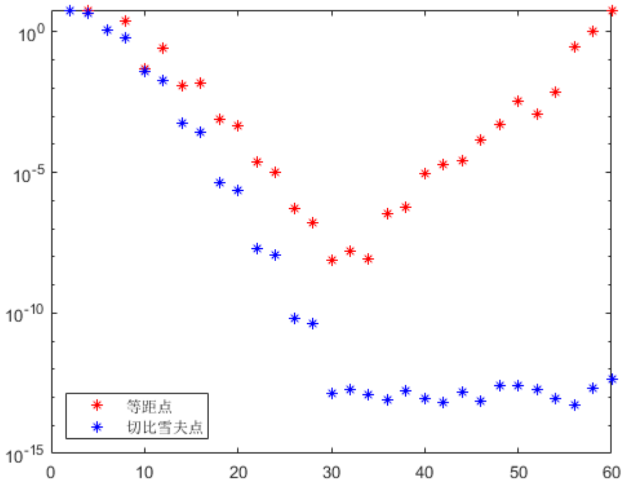

图 6: 比较结果

MATLAB 程序显示如下:

该内容需要登录才可查看

第三题 (a) 由多步法公式，有

$$\mu = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n-1})(x - x_n)(x - x_{n+1})}{(x_{n-2} - x_{n-1})(x_{n-2} - x_n)(x_{n-2} - x_{n+1})} \mathrm{d}x = 0 \tag{8}$$

和

$$\gamma = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n-2})(x - x_n)(x - x_{n+1})}{(x_{n-1} - x_{n-2})(x_{n-1} - x_n)(x_{n-1} - x_{n+1})} \mathrm{d}x = \frac{h}{3} \tag{9}$$

和

$$\beta = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n-2})(x - x_{n-1})(x - x_{n+1})}{(x_n - x_{n-2})(x_n - x_{n-1})(x_n - x_{n+1})} \mathrm{d}x = \frac{4h}{3} \tag{10}$$

和

$$\alpha = \int_{x_{n-1}}^{x_{n+1}} \frac{(x-x_{n-2})(x-x_{n-1})(x-x_n)}{(x_{n+1}-x_{n-2})(x_{n+1}-x_{n-1})(x_{n+1}-x_n)}\mathrm{d}x = \frac{h}{3} \tag{11}$$

(b) 由于局部截断误差是 Lagrange 插值多项式余项的积分，即

$$\begin{aligned} T &= \int_{x_{n-1}}^{x_n} R(t)\mathrm{d}t \\ &= \int_{x_{n-1}}^{x_n} \frac{y^{(5)}(\xi)}{5!}(x-x_{n-2})(x-x_{n-1})(x-x_n)(x-x_{n+1})\mathrm{d}x \\ &= O(h^5) \end{aligned} \tag{12}$$

(c) 由题给函数，和多步法公式

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n-1} + 4f_n + f_{n+1}) \tag{13}$$

可以直接写出递推关系

$$y_{n+1} = [y_{n-1} + \frac{h}{3}(x_{n-1}e^{-5x_{n-1}} + 4x_n e^{-5x_n} + x_{n+1}e^{-5x_{n+1}} - 5y_{n-1} - 20y_n)] \Big/ (1 + \frac{5h}{3}) \tag{14}$$

MATLAB 程序显示如下：

```
LW = 'linewidth'; lw = 2;
n = 100;%节点数
x = linspace(0,2,n)';
y = zeros(n,1);
N = n-1;%分割区间
h = 2/N;
F = @(x) (x.^2).*exp(-5.*x)/2;
f = F(x);
%三阶龙哥方法起步
K1 = zeros(2,1);
K2 = zeros(2,1);
K3 = zeros(2,1);
for i=1:2

    K1(i) = func(x(i),y(i));
    K2(i) = func(x(i)+h/2,y(i)+h*K1(i)/2);
    K3(i) = func(x(i)+h,y(i)-h*K1(i)+2*h*K2(i));
    y(i+1) = y(i)+h*(K1(i)+4*K2(i)+K3(i))/6;
```

```
    end
    fx = @(x) x*exp(-5*x);
    for i=3:n-1
        y(i+1) = (y(i-1)+h*(fx(x(i-1))+4*fx(x(i))+
        fx(x(i+1))-5*y(i-1)-20*y(i))/3)/(1+5*h/3);
    end
    figure(1)
    plot(x, y, 'b', LW, lw), hold on
    plot(x, f,'r', LW, 0.2)
    legend('result', 'exact', 'location', 'ne')
```

输出结果如图 7。

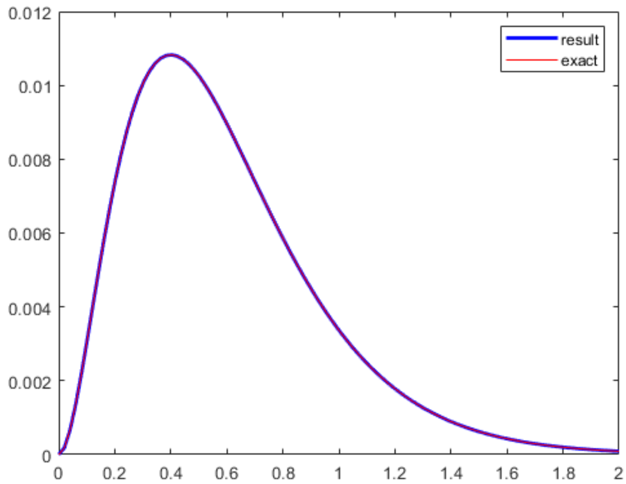

图 7: 结果

(d) 该一阶方程的解析解可直接利用公式

$$y = e^{-1}\int \mathrm{d}x(e\cdot xe^{-5x}) + Const \tag{15}$$

其中 $e = \exp(\int 5\mathrm{d}x)$。
由初值 $y(0)=0$ 可得精确解为

$$y = \frac{1}{2}x^2e^{-5x} \tag{16}$$

MATLAB 程序显示如下:

```
    LW = 'linewidth'; lw = 2;
s = 100;
```

```
fin = 1000;
error = zeros(fin-s+1,1)
for n =s:fin%节点数
x = linspace(0,2,n)';
y = zeros(n,1);
N = n-1;%分割区间
h = 2/N;
F = @(x) (x.^2).*exp(-5.*x)/2;
f = F(x);
%三阶龙哥方法起步
K1 = zeros(2,1);
K2 = zeros(2,1);
K3 = zeros(2,1);
for i=1:2

    K1(i) = func(x(i),y(i));
    K2(i) = func(x(i)+h/2,y(i)+h*K1(i)/2);
    K3(i) = func(x(i)+h,y(i)-h*K1(i)+2*h*K2(i));
    y(i+1) = y(i)+h*(K1(i)+4*K2(i)+K3(i))/6;
end
fx = @(x) x*exp(-5*x);
for i=3:n-1
    y(i+1) = (y(i-1)+h*(fx(x(i-1))+4*fx(x(i))+fx(x(i+1))-5*y(i-1)
end
error(n-s+1) = max(f-y);
end
figure(1)
dots = s:fin
loglog( dots, error, 'b',LW,lw),hold on
```

输出结果如图 8。可以直接看出整体误差是四阶的。

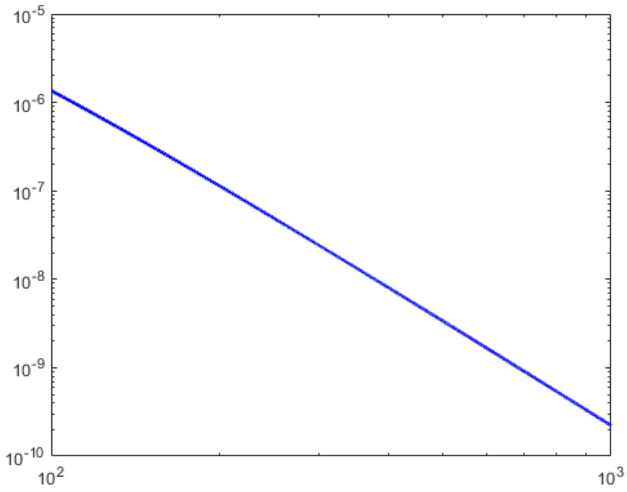

图 8: 结果