

作业三

姓名

学号

日期

第一题 本题考虑使用 **Richardson** 外推技术提高向前差商求给定函数导数的精度。

- (a) (5 分) 使用向前差分计算 $f(x) = \sin(x)$ 在 $x = 1.2$ 处的导数并使用 **loglog** 图展示其精度随离散区间大小 h 的变化。 h 取 $10^0, 10^{-1}, 10^{-2}, 10^{-3}, \dots, 10^{-15}$ 。

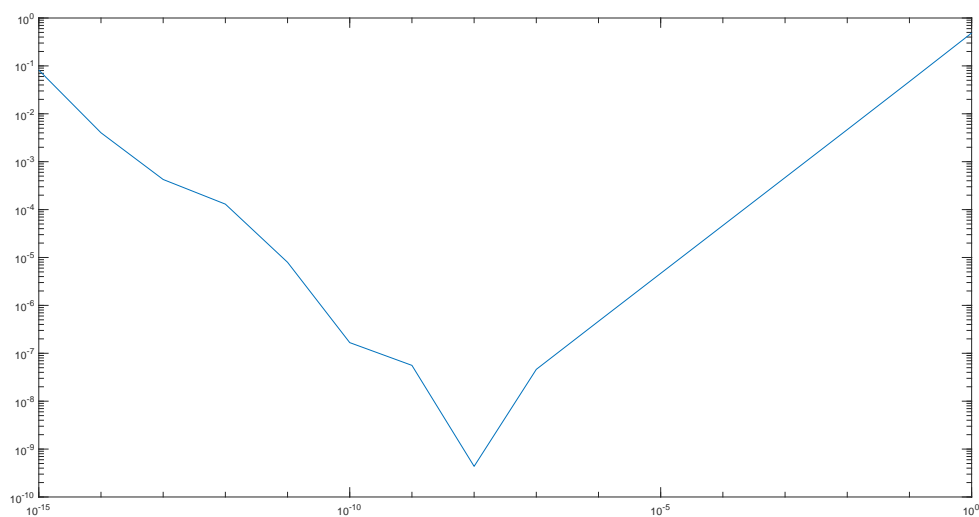


图 1: 精度随离散区间的变化

- (b) (10 分) 推导出使用 **Richardson** 外推技术的向前差商的计算公式并用伪代码给出算法。

答：

由泰勒展开知

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + O(h^3)$$

因此

$$\frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2!}f''(x_0) + O(h^2)$$

记

$$N_1(h) = \frac{f(x_0 + h) - f(x_0)}{h}$$

$$f'(x_0) = N_1(h) - \frac{h}{2!}f''(x_0) + O(h^2)$$

$$f'(x_0) = N_1\left(\frac{h}{2}\right) - \frac{h}{4}f''(x_0) + O(h^2)$$

两式联立得到：

$$f'(x_0) = 2N_1\left(\frac{h}{2}\right) - N_1(h) + O(h^2)$$

记

$$N_2(h) = 2N_1\left(\frac{h}{2}\right) - N_1(h)$$

则

$$f'(x_0) = N_2(h) + \frac{h^2}{12}f'''(x_0) + O(h^4)$$

$$f'(x_0) = N_2\left(\frac{h}{2}\right) + \frac{h^2}{12 \times 4}f'''(x_0) + O(h^4)$$

再次联立得：

$$f'(x_0) = N_2\left(\frac{h}{2}\right) + \frac{N_2\left(\frac{h}{2}\right) - N_2(h)}{3} + O(h^4)$$

$$N_3 = N_2\left(\frac{h}{2}\right) + \frac{N_2\left(\frac{h}{2}\right) - N_2(h)}{3}$$

不断外推得到：

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1} - 1}$$

$$f'(x_0) = N_j\left(\frac{h}{2}\right) + O(h^j)$$

伪代码如下：

给定最大外推次数N, 初始区间长度h

```
for n=0 to N
```

```
    A(n,0)=N1(h/2^n)
```

```
end
```

```
for j=1 to N
```

```

        for i=j to N
            A(i,j)=A(i,j-1)+(A(i,j-1)-A(i-1,j-1))/(2^(j
                -1)-1)
        end
    end
    输出 A(N,N)

```

- (c) (5 分) 使用你的算法计算 $f(x) = \sin(x)$ 在 $x = 1.2$ 处的导数并使用 semilogy 图展示其精度随外推次数变化的情况。请自己选取 h 初始值, 使得用外推方法能够算出的最精确的导数尽量精确。你的回答需要说明你使用外推方法算出的导数值是多少、误差又是多少、 h 的初始值是多少、外推了多少次。

答:

导数值为 0.3623577544766737 真实值为 0.3623577544766736

误差为 10^{-16} h 初始值是 67, 外推了 14 次

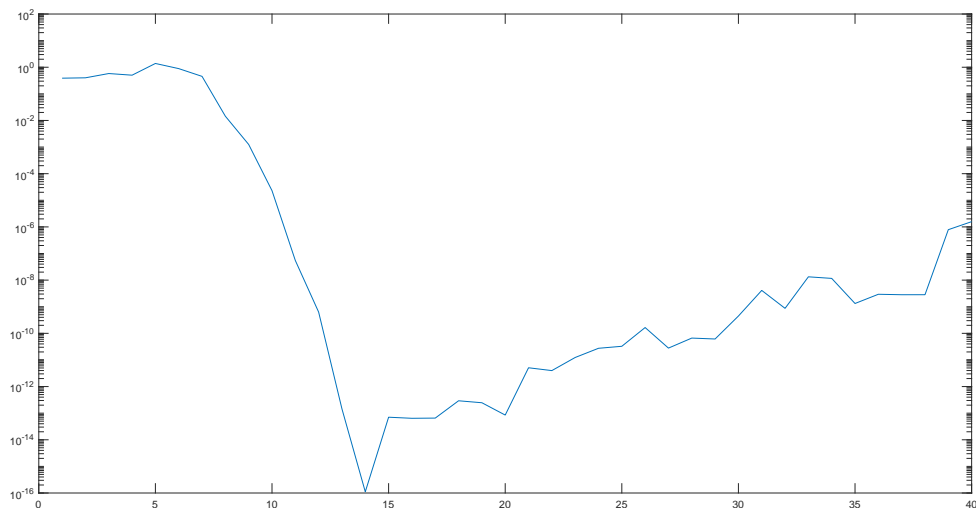


图 2: 精度随外推次数的变化

(a) 的 MATLAB 程序显示如下:

```

H=zeros(1,16);
y=zeros(1,16);
exact=cos(1.2);
err=zeros(1,16);
for i=0:15

```

```

H(i+1)=10^(-i); %设定步长
y(i+1)=forward(1.2,H(i+1));
end
for i=1:16
err(i)=abs(y(i)-exact);
end
loglog(H,err);
function y=f(x)
y=sin(x);
end
function y=forward(x,h)
y=(f(x+h)-f(x))/h;
end

```

(c) 的 MATLAB 程序显示如下:

```

N=40; %最大迭代次数
exact=cos(1.2);
err=zeros(1,N);
A=zeros(N,N);
X=zeros(1,N);
h=67;
for i=1:N
A(i,1)=forward(h/2^(i-1));
end
for k=2:N
for i=k:N
A(i,k)=A(i,k-1)+(A(i,k-1)-A(i-1,k-1))/(2^(
k-1)-1);
end
end

for i=1:N
err(i)=abs(A(i,i)-exact);
X(i)=i;
end
semilogy(X,err);

```

```
fprintf('%.16f\n',A(14,14)-exact);
function y=f(x)
y=sin(x);
end
function y=forward(h)
y=(f(1.2+h)-f(1.2))/h;
end
```

第二题 本题讨论使用复化梯形公式求周期函数的积分。

- (a) (10 分) 证明当 r 不是 m 的整倍数的情况下, 使用 m 个子区间的复化梯形公式可以精确积分

$$\int_{-\pi}^{\pi} \cos(rx)dx \quad \text{和} \quad \int_{-\pi}^{\pi} \sin(rx)dx$$

注意: 这一结论类似于课堂上我们定义的代数精度。同时说明如果 r 为 m 的整数时复化梯形公式对于上面两个积分会给出怎样的结果。

证明:

对于 $\cos(rx)$, 由复化梯形的计算公式得:

$$I_m(f) = \frac{2\pi}{m} \left(\sum_{n=1}^{m-1} \cos\left(\frac{2nr\pi}{m}\right) + \frac{1}{2}(\cos(0) + \cos(2\pi r)) \right)$$

$$\sum_{n=1}^{m-1} \cos\left(\frac{2nr\pi}{m}\right) = \frac{\sin\left(2r\pi - \frac{r\pi}{m}\right)}{2\sin\left(\frac{r\pi}{m}\right)} - \frac{1}{2}$$

当 r 不是 m 的整数倍的时候, $\sin\left(\frac{r\pi}{m}\right) \neq 0$

$$\begin{aligned} \text{原式} &= -\frac{\sin\left(\frac{r\pi}{m}\right)}{2\sin\left(\frac{r\pi}{m}\right)} - \frac{1}{2} + \frac{1}{2}(\cos(0) + \cos(2\pi r)) = 0 \\ &\because \int_{-\pi}^{\pi} \cos(rx)dx = I_m(f) + E_m \\ &\therefore E_m = 0 \end{aligned}$$

此时复化梯形公式可以精确积分。

当 r 是 m 的整数倍的时候, 假设 $r = km, k \in N_+$

$$\sum_{n=1}^{m-1} \cos\left(\frac{2nr\pi}{m}\right) = \sum_{n=1}^{m-1} \cos(2k\pi n) = m - 1$$

$$I_m(f) = \frac{2\pi}{m} \times (m - 1 + 1) = 2\pi, \quad E_m = 2\pi$$

此时不能精确积分。

对于 $\int_{-\pi}^{\pi} \sin(rx)dx$, 当 r 不是 m 的整数倍时结论相同 (即可以精确积分)

现考虑 $r = km$ 的情况:

$$I_m(f) = \frac{1}{2}(\sin(0) + \sin(2\pi r)) + \frac{2\pi}{m} \sum_{n=1}^{m-1} \sin(2k\pi n) = 0$$

$$E_m(f) = T(f) - I_m(f) = 0$$

‘此时仍可以精确积分。

- (b) (10 分) 由于任意的以 2π 为周期的周期函数都可以表示为由正弦与余弦函数的线性组合, 所以上一问中所证明的定理实际上告诉我们求解一个周期函数的积分的有效方法正是复化梯形公式。现使用复化梯形公式和不同数量的子区间个数来求 $f(x) = e^{\cos(x)}$ 在 $[-\pi, \pi]$ 的积分, 并用这个函数的真实积分值作为参照使用 **semilogy** 图画出自随着子区间数量 m 变化所得到的积分精度的变化。

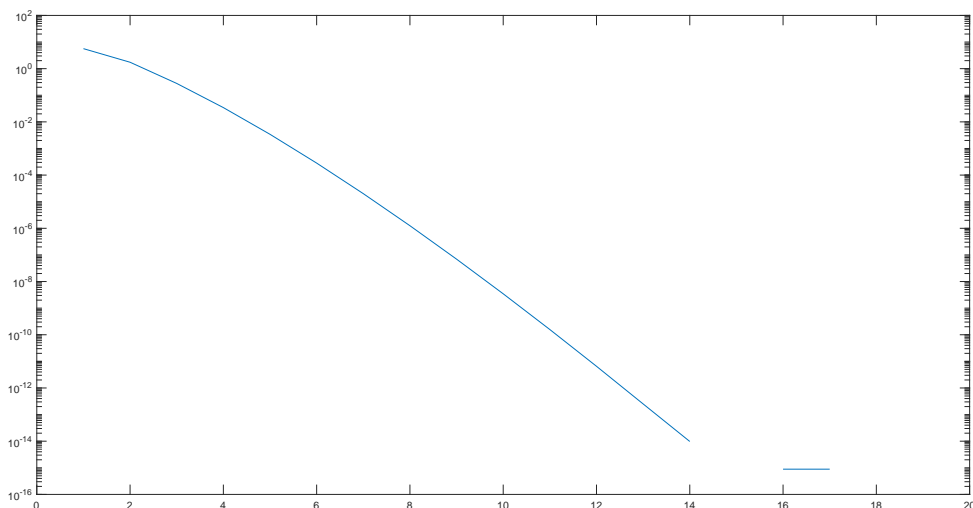


图 3: 精度随自区间数量 m 的变化

(b) 的 MATLAB 程序显示如下:

```
exact=7.9549265210128452;%精确解(来自  
Mathematica)  
N=20;%分割区间  
err=zeros(1,N);  
m=linspace(1,N,N);
```

```

for i=1:N %复化梯形公式
h=2*pi/m(i);
x=linspace(-pi,pi,m(i)+1);
ans=0.5*f(x(1))+0.5*f(x(m(i)+1));
for j=2:m(i)
ans=ans+f(x(j));
end
ans=ans*h;
err(i)=abs(ans-exact);
end

semilogy(m,err);

function y=f(x)
y=exp(cos(x));
end

```

第三题

(a) (10 分) 推导出如下格式的多步法公式:

$$y_{n+1} = y_{n-1} + \alpha f_{n+1} + \beta f_n + \gamma f_{n-1} \quad (1)$$

解: 由数值积分公式得

$$\begin{aligned} \alpha &= \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_n)(x - x_{n-1})}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})} dx = \frac{h}{3} \\ \beta &= \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n+1})(x - x_{n-1})}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})} dx = \frac{4h}{3} \\ \gamma &= \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n+1})(x - x_n)}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})} dx = \frac{h}{3} \end{aligned}$$

因此

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

(b) (10 分) 推导此格式的局部截断误差, 并由此指明此格式的阶数。假设前 n 步的计算结果都是准确的, 则有

$$y_{n-1} = y(x_{n-1}) = y_n - y'_n h + \frac{y''_n}{2!} h^2 - \frac{y'''_n}{3!} h^3 + \frac{y^{(4)}_n}{4!} h^4 - \frac{y^{(5)}_n}{5!} h^5 + O(h^6)$$

$$f_{n+1} \approx y'(x_{n+1}) = y'_n + y''_n h + \frac{y''_n}{2!} h^2 + \frac{y_n^{(4)}}{3!} h^3 + \frac{y_n^{(5)}}{4!} h^4 + O(h^5)$$

$$f_n = f(x_n, y_n) = y'_n \quad (*)$$

$$f_{n-1} = y'(x_{n-1}) = y'_n - y''_n h + \frac{y''_n}{2!} h^2 - \frac{y_n^{(4)}}{3!} h^3 + \frac{y_n^{(5)}}{4!} h^4 + O(h^5)$$

全部带入 (a) 中公式得

$$y_{n+1} = y_n + y'_n h + \frac{y''_n}{2!} h^2 + \cdots + \frac{7y_n^{(5)}}{360} h^5 + O(h^6)$$

而

$$y(x_{n+1}) = y_n + y'_n h + \frac{y''_n}{2!} h^2 + \cdots + \frac{y_n^{(5)}}{5!} h^5 + O(h^6)$$

$$\therefore T_{n+1} = y(x_{n+1}) - y_{n+1} = -\frac{1}{90} h^5 y_n^{(5)} + O(h^6)$$

因此此公式是四阶的。

(c) (10 分) 选取合适的步长值, 用此格式在 $[0, 2]$ 上解如下的初值问题:

$$y' = xe^{-4x} - 4y, \quad y(0) = 0 \quad (2)$$

使用你刚刚推导出的格式 (1), 并利用二阶 **Runge-Kutta** 方法 (即课本公式 (7.15)) 起步。画出解函数。

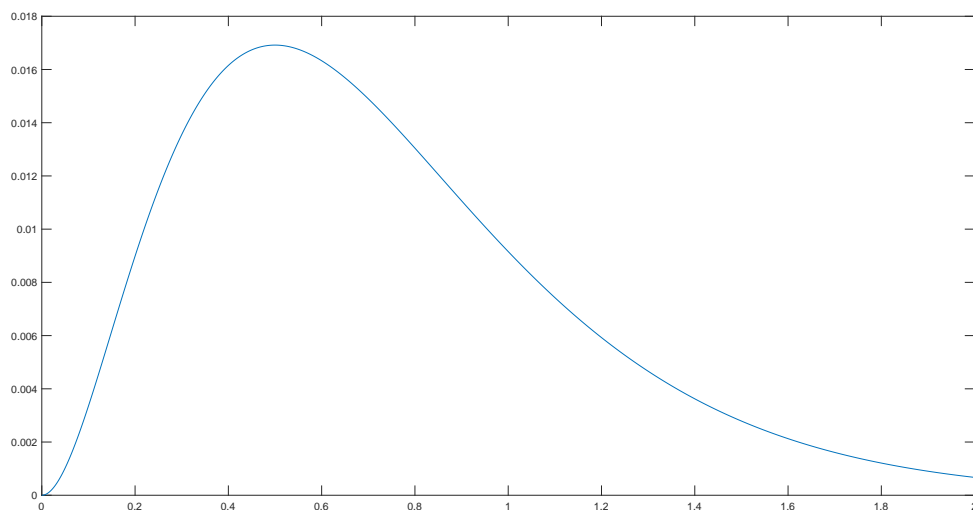


图 4: 解函数的图像

(d) (10 分) 推导出 (2) 的精确解。对比该精确解在 $x = 2$ 这一点的值, 用 **log-log** 图展示所用方法的阶数, 并给出解释。

解：使用常数变易法。

首先计算通解

$$y' = -4y$$

解得：

$$y = C(x)e^{-4x}$$

代回微分方程得：

$$C'(x) = x$$

$$C(x) = \frac{1}{2}x^2 + c \quad c \in R$$

由 $y(0) = 0$ 得： $C(x) = \frac{1}{2}x^2$

$$\therefore y = \frac{1}{2}x^2 e^{-4x}$$

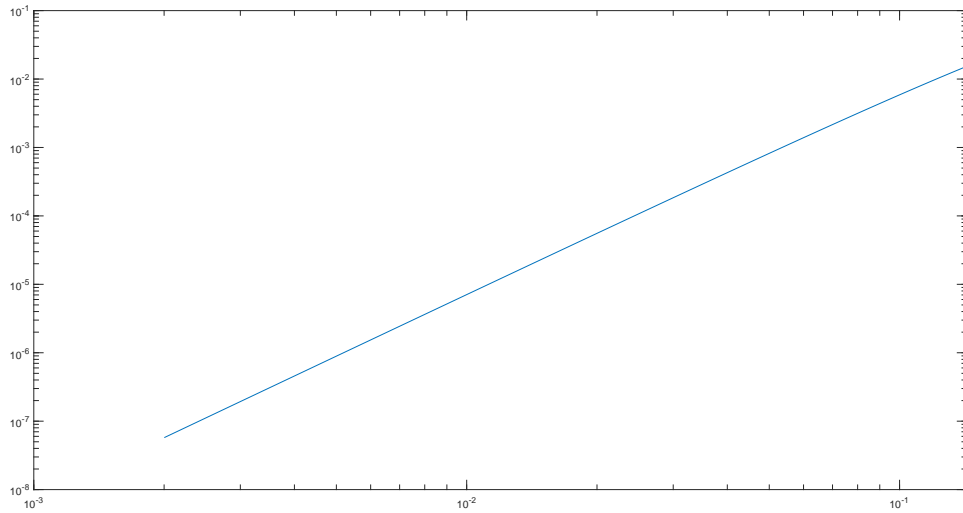


图 5: 用步长和误差估计阶数

图像的斜率为 3，因此该方法是三阶的

个人分析如下：

由于 y_1 是由二阶 Runge-Kutta 方法得到的，其误差为 $O(h^3)$

因此 $y_1 = y(x_1 + ch^3) \quad c \neq 0$

对于 (*), 在计算 y_2 的时候 y_1 扮演 y_n 的角色，此时

$$f_n = y'(x_n + ch^3) = y'_n + ch^3 y''_n + O(h^6)$$

带入 y_{n+1} 的表达式得

$$y_{n+1} = y_n + y'_n h + \frac{y''_n}{2!} h^2 + \frac{y'''_n}{3!} h^3 + \left(\frac{4c}{3} y''_n + \frac{y_n^{(4)}}{4!} \right) h^4 + O(h^5)$$

$$\therefore T_{n+1} = Ch^4 + O(h^5) \quad C \neq 0$$

因此该方法的阶数为三阶

(c) 的 MATLAB 程序显示如下:

```
N=1000;
h=2/(N-1);
x=linspace(0,2,N);
y=zeros(1,N);
err=zeros(1,N);
x(1)=0;y(1)=0;
k1=f(x(1),y(1));
k2=f(x(1)+h/2,y(1)+h/2*k1);
y(2)=y(1)+h*k2;
err(2)=abs(Y(x(2))-y(2));
f1=f(x(1),y(1));
f2=f(x(2),y(2));
for i=3:N
y(i)=3/(3+4*h)*(y(i-2)+h/3*(x(i)*exp(-4*x(i))
+4*f2+f1));
f1=f2;
f2=f(x(i),y(i));
end
plot(x,y)

function s=f(x,y)
s=x*exp(-4*x)-4*y;
end

function y=Y(x)
y=0.5*x^2*exp(-4*x);
end
```

(d) 的 MATLAB 程序显示如下:

```

S=15; %初始区间长度（为了绘图美观从15开始取）
err=zeros(1,1001-S);
H=zeros(1,1001-S);
for N=S:1000
h=2/(N-1);
H(N-S+1)=h;
x=linspace(0,2,N);
y=zeros(1,N);
x(1)=0;y(1)=0;
k1=f(x(1),y(1));
k2=f(x(1)+h/2,y(1)+h/2*k1);
y(2)=y(1)+h*k2;
f1=f(x(1),y(1));
f2=f(x(2),y(2));
for i=3:N
y(i)=3/(3+4*h)*(y(i-2)+h/3*(x(i)*exp(-4*x(i))
+4*f2+f1));
f1=f2;
f2=f(x(i),y(i));
end
err(N-S+1)=abs(y(N)-Y(2));
end

loglog(H,err);

function s=f(x,y)
s=x*exp(-4*x)-4*y;
end

function y=Y(x)
y=0.5*x^2*exp(-4*x);
end

```