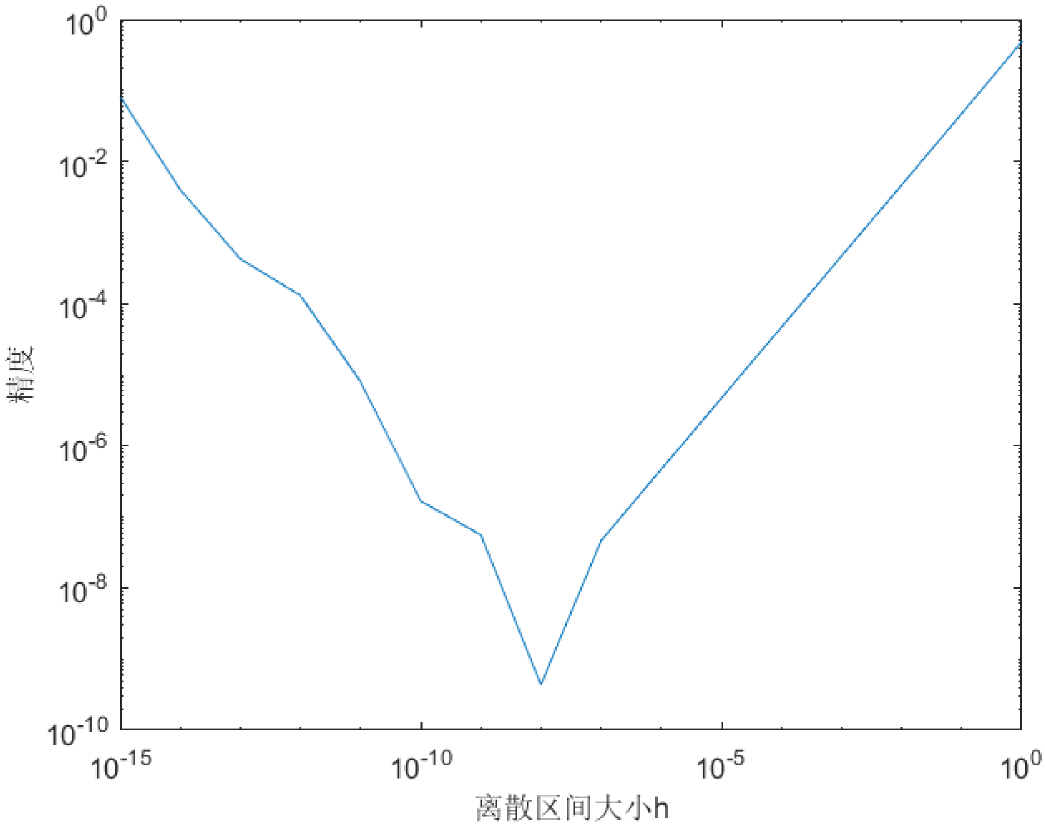
作业一

姓名 学号 日期

第一题 (a) 向前差分近似导数

$$f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h} \tag{1}$$

loglog 图展示精度随离散区间大小 h 的变化:



MATLAB 程序显示如下：

```
clear,clc;
i=1;h=[1,0.1];jdu=[0,0];
while i<17
    jdu(i)=abs((sin(1.2+h(i))-sin(1.2))/h(i)-cos(1.2));
    if i<16
        h(i+1)=h(i)/10;
    end
    i=i+1;
end
loglog(h,jdu)
```

(b) 由泰勒展开

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

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

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

$$f'(x_0)=N_1(h)-\frac{h}{2}f''(x_0)+O(h^3) \tag{3}$$

同时有

$$f'(x_0)=N_1(\frac{h}{2})-\frac{h}{4}f''(x_0)+O(h^3) \tag{4}$$

2(4)-(3), 有 $f'(x_0)=2N_1(\frac{h}{2})-N_1(h)+O(h^3)\approx N_2(h)$

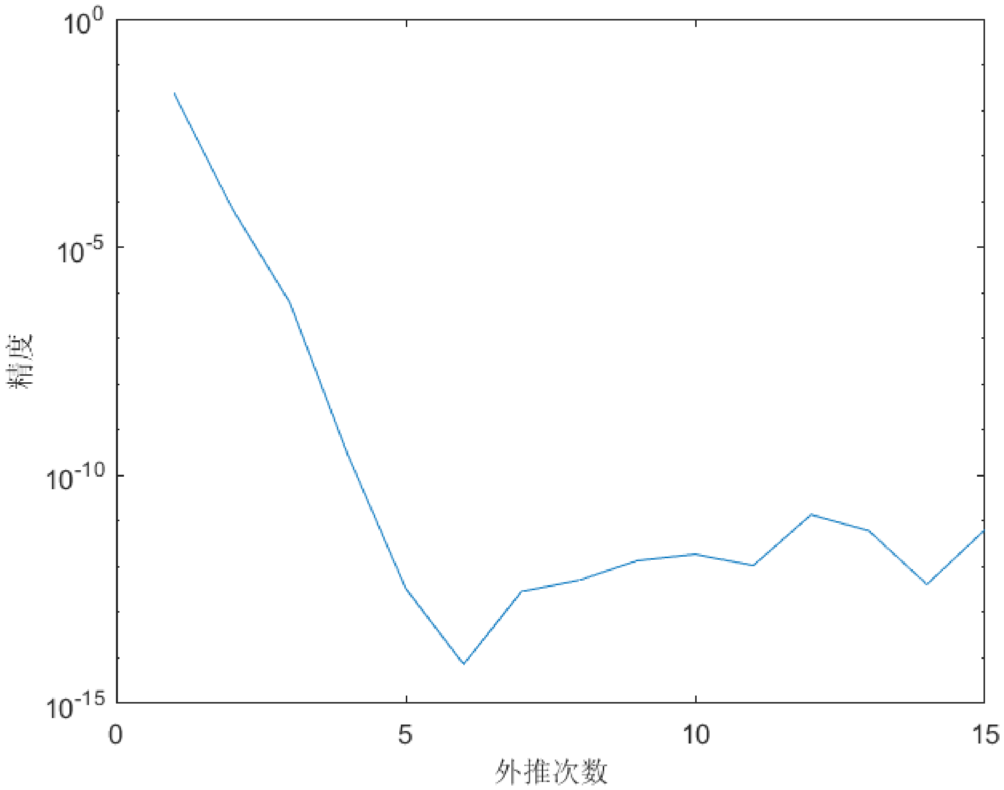
$N_2(h)=2N_1(\frac{h}{2})-N_1(h)$

继续外推下去，得到使用外推技术的向前差分公式

$$\begin{aligned} N_j(h)=N_{j-1}(\frac{h}{2})+\frac{N_{j-1}(\frac{h}{2})-N_{j-1}(h)}{2^{j-1}-1},\quad j=2,3,... \\ f'(x_0)=N_j(\frac{h}{2})+O(h^j) \end{aligned} \tag{5}$$

(c) 选择初始值 h=0.1, 外推方法算出的结果如下:

| 算出导数值 | 误差 | 外推几次 |
|---|---|---|
| 0.338910667767198 | 0.023447086709475 | 1 |
| 0.362429588764464 | 0.000071834287790 | 2 |
| 0.362358365290423 | 0.000000610813750 | 3 |
| 0.362357754193723 | 0.000000000282951 | 4 |
| 0.362357754476339 | 0.000000000000335 | 5 |
| 0.362357754476666 | 0.000000000000007 | 6 |
| 0.362357754476959 | 0.000000000000285 | 7 |
| 0.362357754477180 | 0.000000000000507 | 8 |
| 0.362357754478043 | 0.000000000001369 | 9 |
| 0.362357754474797 | 0.000000000001876 | 10 |
| 0.362357754477738 | 0.000000000001065 | 11 |
| 0.362357754490455 | 0.000000000013782 | 12 |
| 0.362357754482755 | 0.000000000006082 | 13 |
| 0.362357754476263 | 0.000000000000410 | 14 |
| 0.362357754470256 | 0.000000000006418 | 15 |



使用 semilogy 图展示其精度随外推次数变化.

MATLAB 程序显示如下:

```
clear,clc;
i=1;h=0.1;daoshu=[0,0];jdu=[0,0];cishu=[1,2];
while i<16
    daoshu(i)=waitui(i,h/2);
    jdu(i)=abs(daoshu(i)-cos(1.2));
    cishu(i)=i;
    i=i+1;
end
semilogy(cishu,jdu);
i=1;
while i<16
    fprintf('%.15f  %.15f   %d\n', daoshu(i),jdu(i),i);
    i=i+1;
end
function N=waitui(j,h)
    if(j==1)
        N=(sin(1.2+h)-sin(1.2))/h;
    else
        N=waitui(j-1,h/2)+(waitui(j-1,h/2)-
        waitui(j-1,h))/(2^(j-1)-1);
    end
end
```

第二题 (a) 复化梯形公式:

$$T(h)=h[\frac{f(a)}{2}+\sum_{k=1}^{n-1}f(a+kh)+\frac{f(b)}{2}] \tag{6}$$

带入具体的积分:$\int_{-\pi}^{\pi}cos(rx)dx$, 容易得到这个定积分的解是 0.

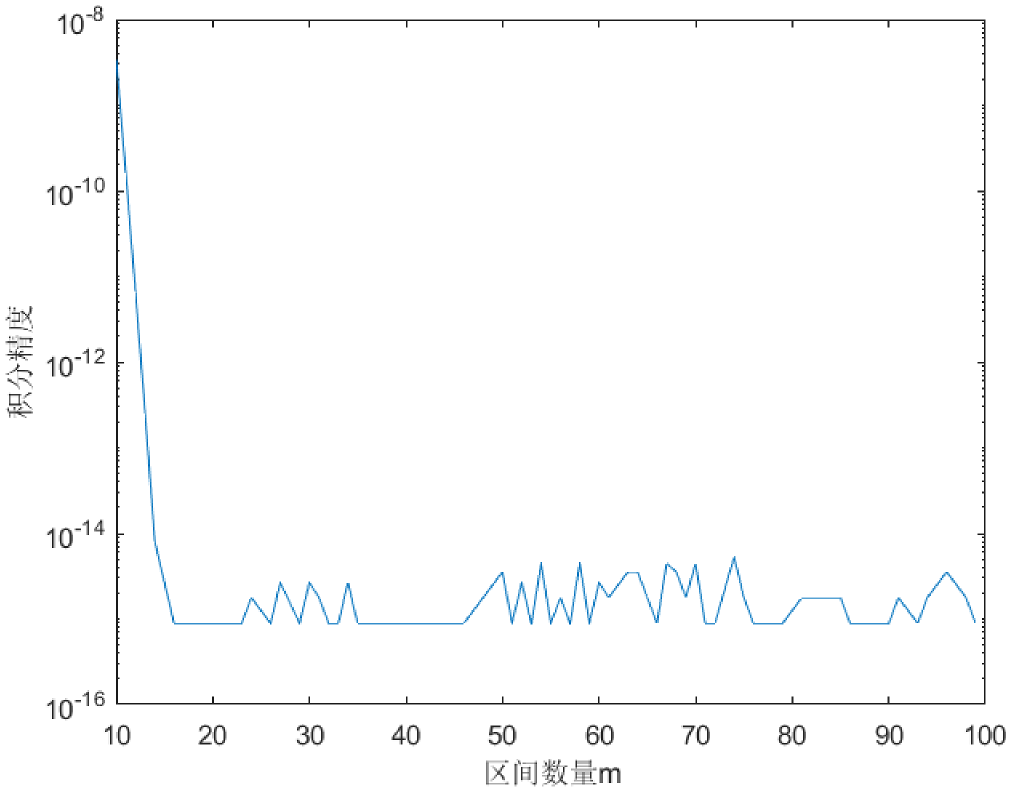
$$\begin{aligned}
T(h)&=h[\frac{cos(-r\pi)}{2}+\sum_{k=1}^{m-1}cos(kh-\pi)+\frac{cos(r\pi)}{2}]\\
&=hcos(r\pi)+\frac{h}{2}\sum_{k=1}^{m-1}[e^{ir(kh-\pi)}+e^{-ir(kh-\pi)}]\\
&=hcos(r\pi)+\frac{h}{2}[e^{-ir(\pi-h)}\frac{1-e^{i(m-1)rh}}{1-e^{irh}}+e^{ir(\pi-h)}\frac{1-e^{-i(m-1)rh}}{1-e^{-irh}}]\\
&=h[cos(r\pi)+e^{-ir(\pi-h)}\frac{1-e^{i(m-1)rh}}{1-e^{irh}}]\\
\lim_{h\to 0}T(h)&=\int_{-\pi}^{\pi}cos(rx)dx
\end{aligned} \tag{7}$$

所以可以精确积分。同理可得对于 $\int_{-\pi}^{\pi}sin(rx)dx$ 复化梯形积分也可以精确积分.

如果 m 是 r 的整数倍, 那么划分的区间也会呈和 f(x) 一致的周期性变化, 产生的误差都被周期性抵消了, 导致复化梯形积分得到的结果和精确解一模一样, 误差为 0.

对于上述两个定积分, 结果都是 0.

(b) 使用 semilogy 图画出随着子区间数量 m 变化所得到的积分精度的变化:




MATLAB 程序显示如下:

```
clear,clc;
f=@(x) exp(cos(x));
jque=integral(f,-pi,pi);
m=10;jdu=[0,0];mm=[0,0];j=1;
while j<70
    h=2*pi/m;
    i=1;sum=f(-pi)/2;
    while i<m
        sum=sum+f(-pi+i*h);
        i=i+1;
    end
    sum=sum+f(pi)/2;
    sum=sum*h;
    if(abs(sum-jque)==0)
        m=m+1;
```

```
            continue;
        end
        jdu(j)=abs(sum-jque);
        mm(j)=m;
        m=m+1;
        j=j+1;
end
semilogy(mm,jdu);
```

第三题 (a) 积分区间为 $[x_{n-1}, x_{n+1}]$, 积分节点为 $x_{n+1}, x_n, x_{n-1}$, 计算系数:

$$\begin{aligned}
\alpha h &= \int_{x_{n-1}}^{xn+1} \frac{(x-x_n)(x-x_{n-1})}{(x_{n+1}-x_n)(x_{n+1}-x_{n-1})} dx = \frac{h}{3} \\
\beta h &= \int_{x_{n-1}}^{xn+1} \frac{(x-x_{n+1})(x-x_{n-1})}{(x_n-x_{n+1})(x_n-x_{n-1})} dx = \frac{4h}{3} \\
\gamma h &= \int_{x_{n-1}}^{xn+1} \frac{(x-x_n)(x-x_{n+1})}{(x_{n-1}-x_n)(x_{n-1}-x_{n+1})} dx = \frac{h}{3}
\end{aligned} \tag{8}$$

所以多步法公式为:

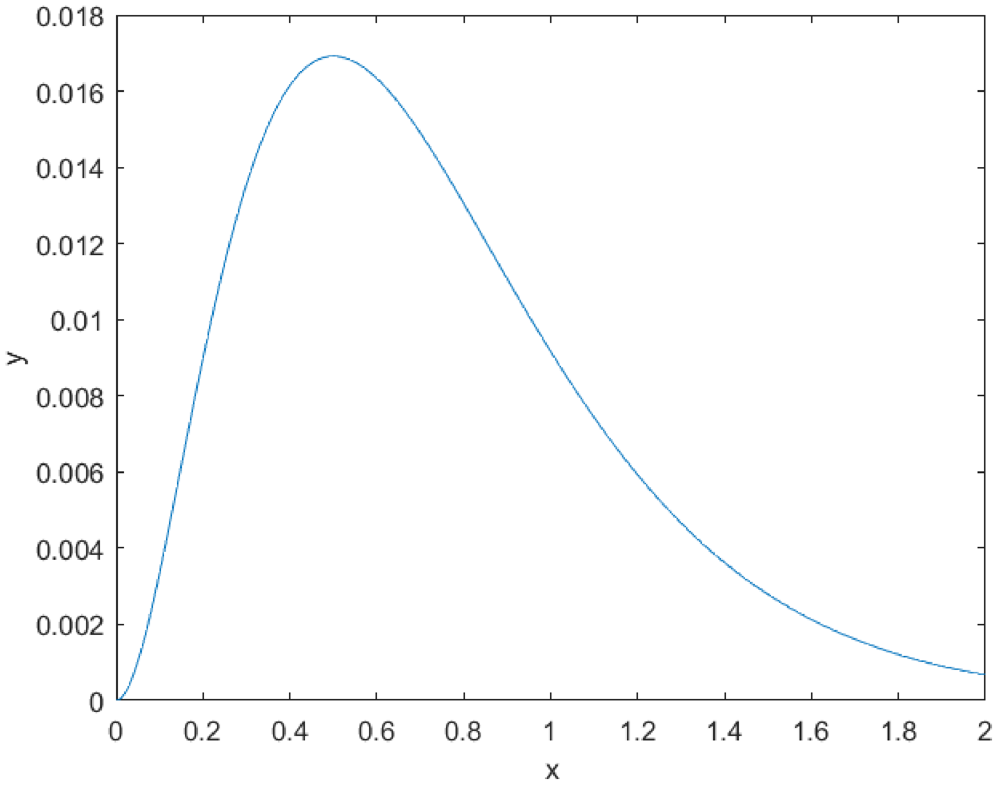
$$y_{n+1} = y_{n-1} + \frac{h}{3}[f_{n+1} + 4f_n + f_{n-1}] \tag{9}$$

(b) 利用泰勒展开估计局部截断误差:

$$\begin{aligned}
T_{n+1} = y_{n+1} - y_{n-1} - \frac{h}{3}[f_{n+1} + 4f_n + f_{n-1}] \\
= [\frac{y^{(5)}(\eta 1) - y^{(5)}(\eta 2)}{120} + \frac{y^{(5)}(\eta 3) - y^{(5)}(\eta 4)}{90}]h^5 \\
= O(h^5)
\end{aligned} \tag{10}$$

因为局部截断误差为 $O(h^{p+1})$, 即此格式的阶数 $p = 4$.

(c) 选取步长值为 0.01, 画出解函数:



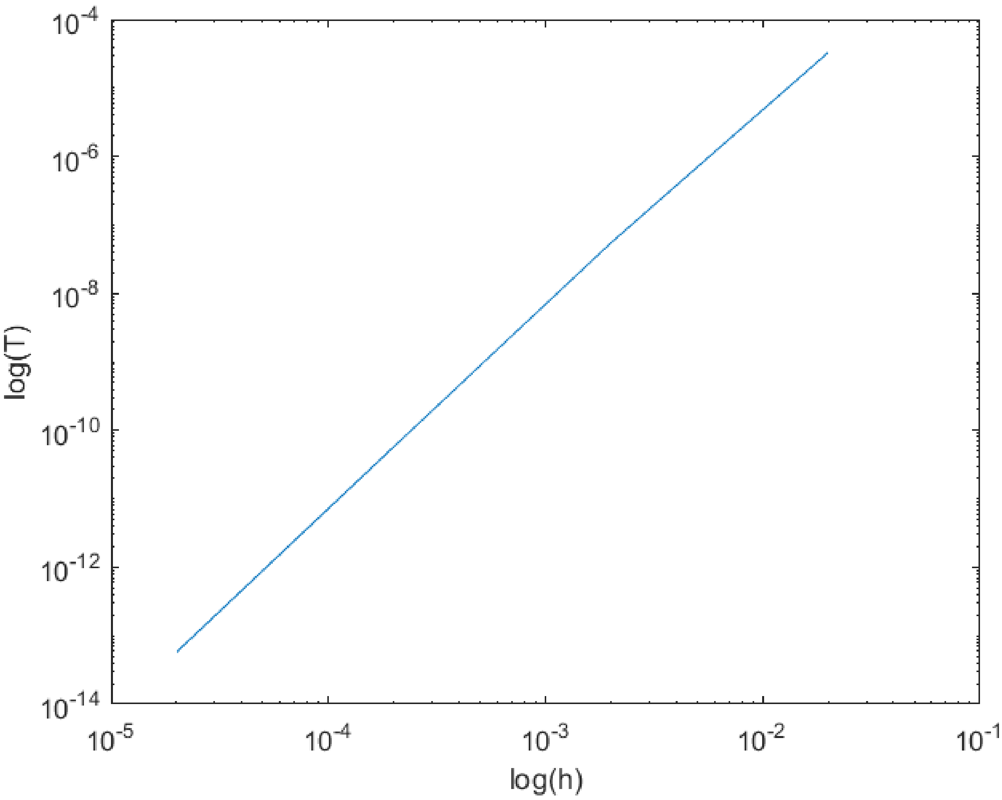

MATLAB 程序显示如下:

```
clear,clc;
f=@(x,y) x*exp(-4*x)-4*y;
y=[0,0];
n=200;
x=linspace(0,2,n+1);
h=2/n;
k1=f(0,0);
k2=f(h/2,h*k1/2);
y(2)=h*k2;
k1=f(x(2),y(2));
k2=f(x(2)+h/2,y(2)+h*k1/2);
y(3)=y(2)+h*k2;
i=3;
while i<n+1
    k3= y(i-1)+h/3*(7*f(x(i),y(i))-
```

```
    2*f(x(i-1),y(i-1))+f(x(i-2),y(i-2)));
    y(i+1)=y(i-1) + h/3*(f(x(i-1),y(i-1))+4*f(x(i),y(i))+
    f(x(i+1),k3));
    i=i+1;
end
plot(x,y);
```

(d) 阶数与局部截断误差的关系:$T_{n+1} = O(h^{p+1})$, 所以 $logT_{n+1} = (p+1)logh + C$. 由 log-log 图可知阶数为 2, 这可能是因为 3.c 中使用了二阶 Runge-Kutta 方法起步, 导致原来是 4 的阶数降到了 2 阶.

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:




MATLAB 程序显示如下:

```
clear,clc;
f=@(x,y) x*exp(-4*x)-4*y;
g=@(x) x^2/2/exp(4*x);
i=1;n=100;yy=[0,0];hh=[0,0];
while i<5
```

```
    hh(i)=2/n;
    yy(i)=abs(duobu(n,f)-g(2));
    n=n*10;
    i=i+1;
end
loglog(hh,yy);
function yy=duobu(n,f)
    y=[0,0];
    x=linspace(0,2,n+1);
    h=2/n;
    k1=f(0,0);
    k2=f(h/2,h*k1/2);
    y(2)=h*k2;
    k1=f(x(2),y(2));
    k2=f(x(2)+h/2,y(2)+h*k1/2);
    y(3)=y(2)+h*k2;
    i=3;
    while i<n+1
        k3= y(i-1)+h/3*(7*f(x(i),y(i))-
        2*f(x(i-1),y(i-1))+f(x(i-2),y(i-2)));
        y(i+1)=y(i-1) + h/3*(f(x(i-1),y(i-1))+4*f(x(i),y(i))+
        f(x(i+1),k3));
        i=i+1;
    end
    yy=y(n+1);
end
```