作业三

姓名 和泳毅　　学号 PB19010450　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 $10^0, 10^{-1}, 10^{-2}, 10^{-3}, ..., 10^{-15}$。

输出结果为:

```
h为10^(0)时, 导数值为: -0.1235426821476362
h为10^(-1)时, 导数值为: 0.3151909944996667
h为10^(-2)时, 导数值为: 0.3576915586159579
h为10^(-3)时, 导数值为: 0.3618916745795620
h为10^(-4)时, 导数值为: 0.3623111519190925
h为10^(-5)时, 导数值为: 0.3623530942853393
h为10^(-6)时, 导数值为: 0.3623572885080861
h为10^(-7)时, 导数值为: 0.3623577082834117
h为10^(-8)时, 导数值为: 0.3623577549127788
h为10^(-9)时, 导数值为: 0.3623578104239300
h为10^(-10)时, 导数值为: 0.3623579214462325
h为10^(-11)时, 导数值为: 0.3623656930074048
h为10^(-12)时, 导数值为: 0.3624878175401136
h为10^(-13)时, 导数值为: 0.3619327060278010
h为10^(-14)时, 导数值为: 0.3663735981263017
h为10^(-15)时, 导数值为: 0.4440892098500626
```

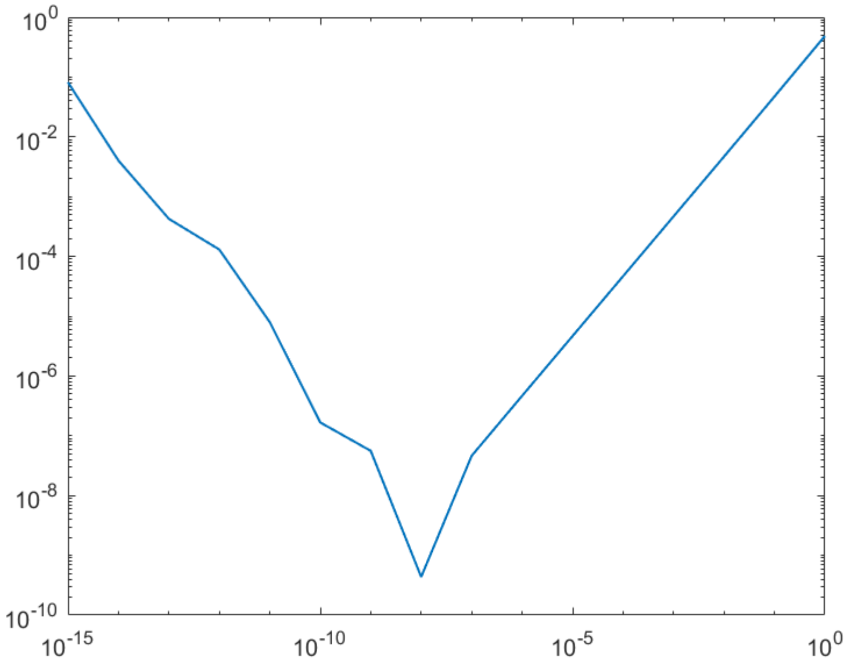

图 1: 1(a) loglog 图

MATLAB 程序显示如下：

```
real = cos(1.2);
h = [1,10^(-1),10^(-2),10^(-3),10^(-4),10^(-5),10^(-6) ...
    ,10^(-7),10^(-8),10^(-9),10^(-10),10^(-11),10^(-12) ...
    ,10^(-13),10^(-14),10^(-15)];
for i=1:16
    diff = FD(1.2,h(i));
    fprintf("h为10^(%d)时， 导数值为：%.16f\n",-(i-1),diff);
    err(i) = abs(real-diff);
end

figure
loglog(h,err);

function diff = FD(x0,h)
    diff = (sin(x0+h)-sin(x0))/h;
end
```

(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:

$$f(x_0+h)=f(x_0)+hf^{'}(x_0)+\frac{h^2}{2!}f^{''}(x_0)+O(h^3) \tag{1}$$

(1)-$f(x_0)$ 得:

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

记

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

$$f^{'}(x_0)=N_1(h)-\frac{h}{2!}f^{''}(x_0)+O(h^2)=N_1(h)+c_1h+O(h^2) \tag{2}$$

$$f^{'}(x_0)=N_1(\frac{h}{2})-\frac{1}{2!}(\frac{h}{2})f^{''}(x_0)+O(h^2)=N_1(\frac{h}{2})+c_1(\frac{h}{2})+O(h^2) \tag{3}$$

2×(3)-(2) 得:

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

$$\begin{aligned}N_2(h)&=N_1(\frac{h}{2})+N_1(\frac{h}{2})-N_1(h)\\&=f^{'}(x_0)-\frac{h^2}{12}f^{'''}(x_0)+O(h^3)\end{aligned}$$

$$f^{'}(x_0)=N_2(h)+\frac{h^2}{12}f^{'''}(x_0)+O(h^3)=N_2(h)+c_2h^2+O(h^3) \tag{4}$$

$$f^{'}(x_0)=N_2(\frac{h}{2})+\frac{1}{12}(\frac{h}{2})^2f^{'''}(x_0)+O(h^3)=N_2(\frac{h}{2})+c_2(\frac{h}{2})^2+O(h^3) \tag{5}$$

(4×(5)-(4))/3 得:

$$f^{'}(x_0)=N_2(\frac{h}{2})+\frac{N_2(\frac{h}{2})-N_2(h)}{3}+O(h^3)\approx N_3(h)$$

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

继续外推下去，得到:

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

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

伪代码显示如下:

```
% 给定 h,n,x0
% 计算 N(n,n)
for i = 1 : n
    N(i,1) = (f(x0+h/2^(i-1))-f(x0)) / (h/2^(i-1))
end
for i = 2 : n
    for j = i : n
        N(j,i) = N(j,i-1) + (N(j,i-1)-N(j-1,i-1)) / (2^(i-1)-1)
    end
end
```

(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。

输出结果为：

```
导数值为： 0.36235775447667373239。
误差为： 0.00000000000000011102。
h初始值为： 67。
外推次数为： 14。
```

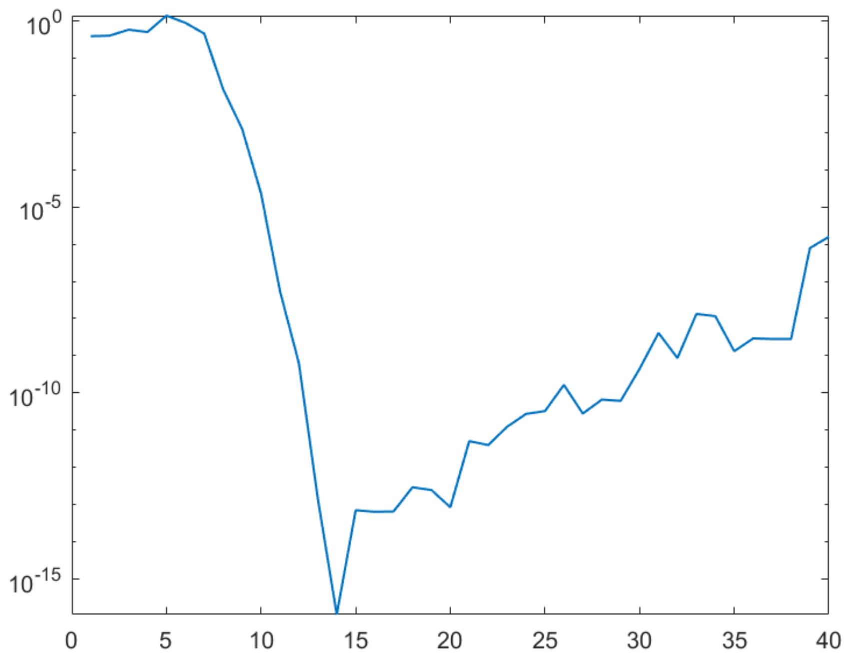

图 2: 1(c) semilogy 图

MATLAB 程序显示如下：

```
real = cos(1.2);
n = [1:40];
N = Table(1.2,67,40);
for i=1:40
    err(i) = abs(N(i,i)-real);
end
figure
semilogy(n,err);
fprintf("导数值为：%.20f。\n",N(14,14));
fprintf("误差为：%.20f。\n",abs(N(14,14)-real));
fprintf("h初始值为：67。\n");
fprintf("外推次数为：14。\n");

function N = Table(x0,h,n)
    N = zeros(n,n);
    for i = 1:n
        N(i,1) = (sin(x0+h/2^(i-1))-sin(x0))/(h/2^(i-1));
    end
    for i = 2:n
        for j = i:n
            N(j,i) = N(j,i-1)+(N(j,i-1)-N(j-1,i-1))/ ...
                (2^(i-1)-1);
        end
    end
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 的整倍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：

记 $f_1(x) = \cos(rx)$，$f_2(x) = \sin(rx)$，$h = \frac{2\pi}{m}$

(1) 若 r 不是 m 的整数倍,

$$\sum_{n=1}^{m-1}\cos(nrh)=\frac{\sin[(m-\frac{1}{2})rh]}{2\sin(\frac{rh}{2})}-\frac{1}{2}=\frac{\sin(2r\pi-\frac{r\pi}{m})}{2\sin(\frac{r\pi}{m})}-\frac{1}{2}=-1$$
$$\sum_{n=1}^{m-1}\sin(nrh)=\frac{\cos(\frac{rh}{2})-\cos[(m-\frac{1}{2})rh]}{2\sin(\frac{rh}{2})}=\frac{\cos(\frac{r\pi}{m})-\cos(2r\pi-\frac{r\pi}{m})}{2\sin(\frac{r\pi}{m})}=0$$

$$\begin{aligned}T_m(f_1)&=h[\frac{1}{2}f_1(-\pi)+\sum_{i=1}^{m-1}f_1(x_i)+\frac{1}{2}f_1(\pi)]\\&=h[\frac{1}{2}\cos(-r\pi)+\sum_{n=1}^{m-1}\cos[r(-\pi+nh)]+\frac{1}{2}\cos(r\pi)]\\&=h[\sum_{n=1}^{m-1}\cos[r(-\pi+nh)]+\cos(r\pi)]\\T_m(f_2)&=h[\frac{1}{2}f_2(-\pi)+\sum_{i=1}^{m-1}f_2(x_i)+\frac{1}{2}f_2(\pi)]\\&=h[\frac{1}{2}\sin(-r\pi)+\sum_{n=1}^{m-1}\sin[r(-\pi+nh)]+\frac{1}{2}\sin(r\pi)]\\&=h\sum_{n=1}^{m-1}\sin[r(-\pi+nh)]\end{aligned}$$

r 为偶数时,

$$T_m(f_1)=h[\sum_{n=1}^{m-1}\cos(rnh)+1]=0$$
$$T_m(f_2)=h\sum_{n=1}^{m-1}\sin(rnh)=0$$

r 为奇数时,

$$T_m(f_1)=h[-\sum_{n=1}^{m-1}\cos(rnh)-1]=0$$
$$T_m(f_2)=-h\sum_{n=1}^{m-1}\sin(rnh)=0$$

由于

$$I(f)=T_m(f)-E_m(f)$$
$$I(f_1)=T_m(f_1)=0,\quad I(f_2)=T_m(f_2)=0$$

于是

$$E_m(f_1)=0,\quad E_m(f_2)=0$$

所以可以精确积分。

(2) 若 r 是 m 的整数倍，即 $r=bm$，$b\in\mathbb{Z}$ 有

$$\sum_{n=1}^{m-1}\cos(nrh)=\sum_{n=1}^{m-1}\cos(2bn\pi)=m-1\quad \text{发散}$$
$$\sum_{n=1}^{m-1}\sin(nrh)=\sum_{n=1}^{m-1}\sin(2bn\pi)=0$$

故 r 为偶数时，

$$T_m(f_1)=h[\sum_{n=1}^{m-1}\cos(rnh)+1]=2\pi$$
$$T_m(f_2)=h\sum_{n=1}^{m-1}\sin(rnh)=0$$

r 为奇数时，

$$T_m(f_1)=h[-\sum_{n=1}^{m-1}\cos(rnh)-1]=-2\pi$$
$$T_m(f_2)=-h\sum_{n=1}^{m-1}\sin(rnh)=0$$

证毕。

结论：

对 $\cos(rx)$ 而言：r 不是 m 的整数倍时，可以精确积分；r 是 m 的整数倍时，当 r 是偶数，计算结果为 2π，当 r 是奇数，计算结果为 -2π。

对 $\sin(rx)$ 而言：r 不是 m 的整数倍时，可以精确积分；r 是 m 的整数倍时，积分结果也是准确的。

验证：

取 $r=12$，因子为 $1,2,3,4,6,12$

对 $\cos(rx)$ 有：

```
区间        求和项          结果
--------------------------------
 1           0.0         6.28319
 2           1.0         6.28319
 3           2.0         6.28319
 4           3.0         6.28319
 5          -1.0        -5.58059e-16
 6           5.0         6.28319
 7          -1.0         1.19584e-15
```

```
8          -1.0        0
9          -1.0       -1.08511e-15
10         -1.0       -1.25563e-15
11         -1.0        3.6147e-15
12         11.0        6.28319
13         -1.0        4.56106e-15
14         -1.0        1.29549e-15
15         -1.0        9.30098e-16
```

对 $\sin(rx)$ 有：

```
区间        求和项          结果
--------------------------------
 1          0.0        0
 2          0.0        0
 3          0.0       -1.11612e-14
 4          0.0        0
 5          0.0        0
 6          0.0       -2.79029e-14
 7          0.0       -9.96534e-17
 8          0.0        0
 9          0.0       -2.17023e-15
 10         0.0        6.97574e-17
 11         0.0        5.70742e-16
 12         0.0       -4.697e-14
 13         0.0       -2.38785e-15
 14         0.0       -4.98267e-17
 15         0.0        9.30098e-17
```

(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** 图画出随着子区间数量 m 变化所得到的积分精度的变化。

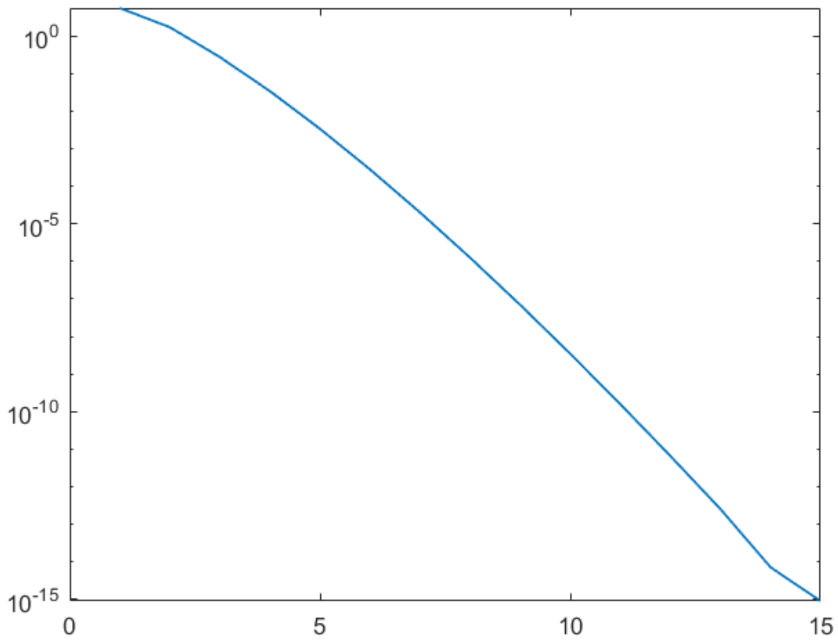

图 3: 2(b) semilogy 图

MATLAB 程序显示如下:

```
real = 7.95492652101284527 45;
for i=1:15
    err(i) = Composite(i,real);
end
figure
semilogy([1:15],err);

function err = Composite(M,real)
    h = 2*pi/M;
    sum = 0;
    x(1) = -pi;
    for i=2:M
        x(i) = x(i-1)+h;
        sum = sum + f(x(i)) ;
    end
    result = h*(0.5*(f(-pi))+sum+0.5*(f(pi)));
    err = abs(result - real);
end
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:

$$y_{n+1} = y_{n-1} + \alpha f_{n+1} + \beta f_n + \gamma f_{n-1} \tag{6}$$

答:

$$y_{n+1} = y_{n-1} + f_{n+1}\int_{t_{n-1}}^{t_{n+1}} \ell_{n+1}(t)dt + f_n\int_{t_{n-1}}^{t_{n+1}} \ell_n(t)dt + f_{n-1}\int_{t_{n-1}}^{t_{n+1}} \ell_{n-1}(t)dt$$

其中,

$$\int_{t_{n-1}}^{t_{n+1}} \ell_{n+1}(t)dt = \int_{t_{n-1}}^{t_{n+1}} \frac{(t-t_n)(t-t_{n-1})}{(t_{n+1}-t_n)(t_{n+1}-t_{n-1})}dt = \frac{h}{3}$$

$$\int_{t_{n-1}}^{t_{n+1}} \ell_n(t)dt = \int_{t_{n-1}}^{t_{n+1}} \frac{(t-t_{n+1})(t-t_{n-1})}{(t_n-t_{n+1})(t_n-t_{n-1})}dt = \frac{4h}{3}$$

$$\int_{t_{n-1}}^{t_{n+1}} \ell_{n-1}(t)dt = \int_{t_{n-1}}^{t_{n+1}} \frac{(t-t_{n+1})(t-t_n)}{(t_{n-1}-t_{n+1})(t_{n-1}-t_n)}dt = \frac{h}{3}$$

有:

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

(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。

答:

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

假设:

$$y_{n+1} = y(t_{n+1}) \quad y_n = y(t_n) \quad y_{n-1} = y(t_{n-1})$$

$$f_{n+1} = f(t_{n+1}, y(t_{n+1})) \quad f_n = f(t_n, y(t_n)) \quad f_{n-1} = f(t_{n-1}, y(t_{n-1}))$$

$$y_{n+1} = y(t_{n-1}) + \frac{h}{3}(y^{'}(t_{n+1}) + 4y^{'}(t_n) + y^{'}(t_{n-1}))$$

$$\begin{aligned} T_{n+1} &= y(t_{n+1}) - y_{n+1} \\ &= y(t_{n+1}) - y(t_{n-1}) - \frac{h}{3}(y^{'}(t_{n+1}) + 4y^{'}(t_n) + y^{'}(t_{n-1})) \end{aligned}$$

做 $Taylor$ 展开：

$$
\begin{aligned}
y(t_{n+1}) &= y(t_n) + hy'(t_n) + \frac{h^2}{2!}y''(t_n) + \frac{h^3}{3!}y^{(3)}(t_n) + \frac{h^4}{4!}y^{(4)}(t_n) + \frac{h^5}{5!}y^{(5)}(t_n) + O(h^6) \\
y(t_{n-1}) &= y(t_n) - hy'(t_n) + \frac{h^2}{2!}y''(t_n) - \frac{h^3}{3!}y^{(3)}(t_n) + \frac{h^4}{4!}y^{(4)}(t_n) - \frac{h^5}{5!}y^{(5)}(t_n) + O(h^6) \\
y'(t_{n+1}) &= y'(t_n) + hy''(t_n) + \frac{h^2}{2!}y^{(3)}(t_n) + \frac{h^3}{3!}y^{(4)}(t_n) + \frac{h^4}{4!}y^{(5)}(t_n) + O(h^5) \\
y'(t_{n-1}) &= y'(t_n) - hy''(t_n) + \frac{h^2}{2!}y^{(3)}(t_n) - \frac{h^3}{3!}y^{(4)}(t_n) + \frac{h^4}{4!}y^{(5)}(t_n) + O(h^5)
\end{aligned}
$$

代入后有：

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

所以此格式是四阶精度。

(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：

$$y' = xe^{-4x} - 4y, \quad y(0) = 0 \tag{7}$$

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。

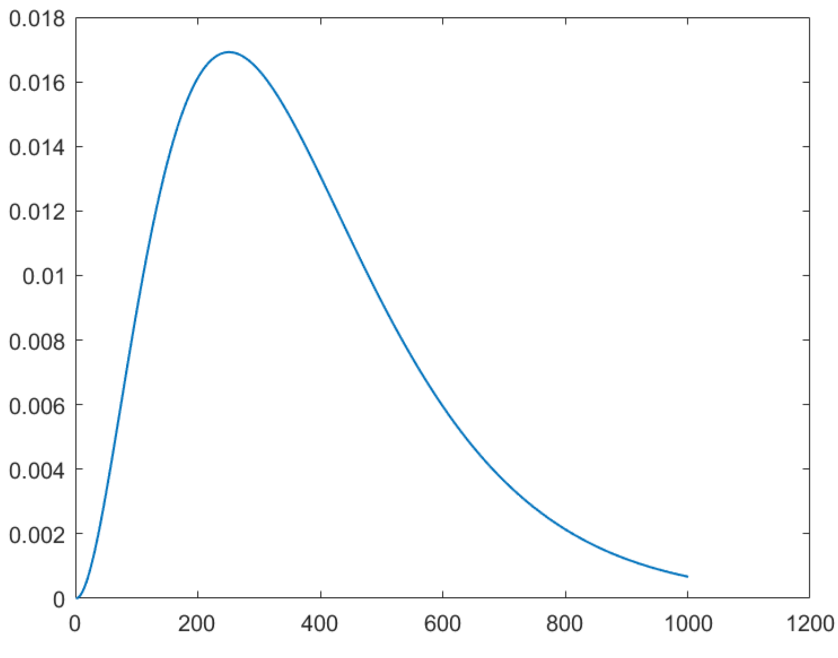

图 4: 3(c) 解函数图

MATLAB 程序与第四问一起给出。

(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。

对

$$y' + 4y = xe^{-4x}$$

有

$$\begin{aligned} P(x) = 4, \quad Q(x) = xe^{-4x} \\ y = e^{-\int 4dx}(\int xe^{-4x} \cdot e^{\int 4dx} dx + C) \\ = e^{-4x}(\frac{1}{2}x^2 + C) \\ = \frac{1}{2}x^2 e^{-4x} + Ce^{-4x} \end{aligned}$$

由初值

$$y(0) = 0$$

有

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

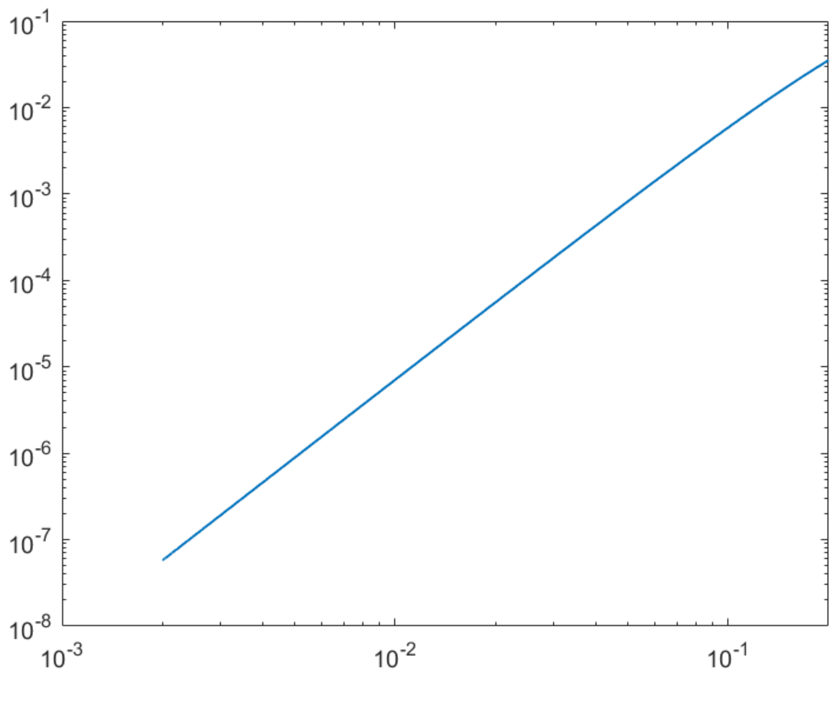

图 5: 3(d) log-log 图

解释：

在 loglog 图中取两点，计算斜率发现是三阶精度。

由于我们使用二阶 **Runge-Kutta** 方法起步，该方法为二阶方法，即

$$y(t_1) = y(t_0) + hf(t_0 + \frac{h}{2}, y(t_0) + \frac{h}{2}f(t_0, y(t_0))) + O(h^3)$$

之后采用多步法

$$y(t_2) = y(t_0) + \frac{h}{3}y^{'}(t_2) + \frac{4h}{3}y^{'}(t_1) + \frac{h}{3}y^{'}(t_0) - \frac{1}{90}h^5 y^{(5)}(t_1) + O(h^6)$$

$y(t_1)$ 项的误差乘以系数 $\frac{4h}{3}$ 得到 $O(h^4)$ 被传递下去，所以整体看来为三阶精度。

MATLAB 程序显示如下：

```
%%
y = result(1000);
plot(y);
%%
err = zeros(1,991);
real = 1/2*exp(-4*2)*2^2;
for i=10:1000
    y = result(i);
    err(i-9) = abs(y(i+1)-real);
end
h = zeros(1,991);
for i=10:1000
    h(i-9) = 2/i;
end
figure
loglog(h,err);

%%
function y = result(n)
    h=2/n;
    x(1)=0;
    for i=2:n+1
        x(i) = x(i-1)+h;
    end

    y = zeros(1,n+1);
```

```
    y(1)=0;
    y(2)=RK2(y(1),h,x(1));

    for i=3:n+1
        y(i) = MS(x,y,h,i);
    end
end

function y1 = MS(x,y,h,i)
    temp0 = f(x(i-2),y(i-2));
    temp1 = f(x(i-1),y(i-1));
    y1 = (3/(3+4*h))*(y(i-2)+(h/3)* ...
        (x(i)*exp(-4*x(i))+4*temp1+temp0));
end

function y1 = RK2(y0,h,x)
    k1 = f(x,y0);
    k2 = f(x+h/2,y0+h*k1/2);
    y1 = y0 + h*k2;
end

function y1 = f(x0,y0)
    y1 = x0*exp(-4*x0)-4*y0;
end
```