Student name: Abao Zhang (张宝)

Student number: 12332459

Homework 6

Question 1

(a)

核心代码如下，详情见 question-1.py

```
def f_x(f, x, y, h):
    return (f(x + h, y) - f(x - h, y)) / (2 * h)
def f_y(f, x, y, h):
    return (f(x, y + h) - f(x, y - h)) / (2 * h)

def f(x, y):
    return x * y / (x + y)
for h in [0.1, 0.01, 0.001]:
    print(f"while h={h}:")
    print(f"f_x(2, 3)={f_x(f, 2,3, h)}")
    print(f"f_y(2, 3)={f_y(f, 2,3, h)}")
```

当 $h = 0.1$:

$$f_{x(2,3)} = 0.3601440576230519$$
$$f_{y(2,3)} = 0.1600640256102448$$

当 $h = 0.01$:

$$f_{x(2,3)} = 0.36000144000575274$$
$$f_{y(2,3)} = 0.16000064000255554$$

当 $h = 0.001$:

$$f_{x(2,3)} = 0.3600000144001747$$
$$f_{y(2,3)} = 0.16000000640004064$$

精确值

$$f_{x(x,y)} = \frac{y^2}{(x+y)^2} = 0.36$$
$$f_{y(x,y)} = \frac{x^2}{(x+y)^2} = 0.16$$

(b)

当 $h = 0.1$:

$$f_{x(3,4)} = -0.16000938028547684$$
$$f_{y(3,4)} = 0.12002495951283387$$

当 $h = 0.01$:

$$f_{x(3,4)} = -0.1600000938660251$$
$$f_{y(3,4)} = 0.1200002495999497$$

当 $h = 0.001$:

$$f_{x(3,4)} = -0.16000000093863154 f_{y(3,4)} = 0.12000000249601239$$

精确值

$$f_{x(x,y)} = -\frac{y}{(x+y)^2} = 0.16$$
$$f_{y(x,y)} = \frac{x}{(x+y)^2} = 0.12$$

h越小，越接近精确值

Question 2

(a)

总误差$E(f,h)$:

$$|E(f,h)| \leq \frac{\varepsilon}{h} + \frac{Mh^2}{6}$$

代入舍入误差$\varepsilon = 5 \times 10^{-4}$，$M = \max_{a \leq x \leq b}\{|f^{(3)}(x)|\} = 1.5$

使等式右边最小化，即最小化

$$E(h) = \frac{\varepsilon}{h} + \frac{h^2}{4}$$

求导

$$E'(h) = -\frac{\varepsilon}{h^2} + \frac{h}{2}$$
$$E^{(2)}(h) = \frac{2\varepsilon}{h^2} + \frac{1}{2}$$

$E^{(2)}(h) > 0$ 恒成立

进行单调分析后可发现，$E'(h)$ 单增，则其在$h = (2\varepsilon)^{-3} = 0.1$处取得零点，即$E(h)$在该点取得极小值

(b)

直接使用课本上的公式：

$$h = \left(\frac{45\varepsilon}{4M}\right)^{\frac{1}{5}} = 0.3272$$

Question 3

(a)

代入$h = 0.1$

$$f'(3) = \frac{-f(3.2) + 8f(3.1) - 8f(2.9) + f(2.8)}{12h} \approx 0.34 \text{ (保留两位有效数字)}$$

(b)

不会

Question 4

(a)

代入$h = 0.05$

$$f''(1) \approx \frac{f(1.05) - 2f(1) + f(0.95)}{0.05^2} = -0.52$$

(b)

代入$h = 0.1$

$$f''(1) \approx \frac{f(1.1) - 2f(1) + f(0.9)}{0.1^2} = -0.54$$

(c)

代入$h = 0.05$

$$f''(1) \approx \frac{-f(1.1) + 16f(1.05) - 30f(1) + 16f(0.95) - f(0.9)}{120.05^2} \approx -0.51$$

(d)

$$f''(x) = -\cos(x)$$
$$f''(1) \approx -0.5403$$

答案(b) 更准确

Question 5

(a)

中心差分公式

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

相加得到

$$f'(x) + f''(x) = \frac{(2+h)f_1 - 4f_0 + (2-h)f_{-1}}{2h^2} + O(h^2)$$

(b)

前向差分公式

$$f'(x) = \frac{-3f_0 + 4f_1 - f_2}{2h} + O(h^2)$$
$$f''(x) = \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} + O(h^2)$$

相加得到

$$f'(x) + f''(x) = \frac{-3f_0 + 4f_1 - f_2}{2h} + \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} + O(h^2)$$

(c)

$$f'(x) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4)$$
$$f''(x) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2)$$

相加

$$\begin{aligned} f'(x) + f''(x) &= \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4) + \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2) \\ &= \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2) \end{aligned}$$

保留较大的无穷小项$O(h^2)$

Question 6

这里使用了 matlab 的 vpa 高进度浮点数，如果不使用，会导致无法收敛到预期进度（会发生震荡）

```
function y=f1(x)
    y=vpa(60*power(x,45) - 32*power(x,33) + 233*power(x,5) - 47*power(x,2) - 77);
end
digits(50)
[L1, n1] = difflim(@f1, vpa(1 / sqrt(3)), vpa(1e-13));
format long g;
disp(double(L1));
disp(n1);
```

结果如下：

```
             1        24149615818.5484                       0
           0.1        82.9630753269441        24149615735.5853
          0.01        75.2511624674657        7.71191285947832
         0.001        75.1742713499854      0.0768911174803053
        0.0001        75.1735024617194    0.000768888266048535
         1e-05         75.173494772839    7.68888036952712e-06
         1e-06        75.1734946959502    7.68888159830394e-08
         1e-07        75.1734946951813    7.68892014290516e-10
         1e-08        75.1734946951736    7.68639306409055e-12
         1e-09        75.1734946951735    8.31070092028317e-14
         1e-10        75.1734946951735    2.71207732806691e-15

    10
```

可以发现$f'\left(\frac{1}{\sqrt{3}}\right) = 75.1734946951735$为精确到小数点后 13 位的值

(b)

同样的

```
function y=f2(x)
    y = vpa(sin(x*x*x - 7*x*x + 6*x + 8));
end

digits(50)
[L2, n2] = difflim(@f2, vpa((1-sqrt(5))/2), vpa(1e-13));
format long g;
disp(double(L2));
disp(n2);
```

```
                         1        -0.330457939147994                         0
                       0.1          2.73814600116262          3.06860394031061
                      0.01          2.96689401375905         0.228748012596436
                     0.001          2.96552904676237       0.00136496699668427
                    0.0001          2.96551497140378      1.40753585921063e-05
                     1e-05          2.96551483060756       1.4079621870426e-07
                     1e-06          2.96551482919959      1.40796694416192e-09
                     1e-07          2.96551482918551      1.40798219232309e-11
                     1e-08          2.96551482918537      1.40698528689546e-13
                     1e-09          2.96551482918537      1.65326813074297e-15

     9
```

可以发现$f'\left(\frac{1-\sqrt{5}}{2}\right) = 2.96551482918537$为精确到小数点后 13 位的值

Question 7

函数只需要进行简单修改即可，具体修改了三行代码，代码如下

```
function [L,n]=difflim(f,x,toler)
    %Input - f is the function input as a string 'f'
    %- x is the differentiation point
    %- toler is the tolerance for the error
    %Output-L=[H' D' E']:
    %H is the vector of step sizes
    %D is the vector of approximate derivatives
    %E is the vector of error bounds
    %- n is the coordinate of the ''best approximation''
    max1=15;
    h=1;
    H(1)=h;
    % !!!!修改了下面一行
    % D(1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
    D(1)=(-feval(f,x+2*h)+8*feval(f, x+h)-8*feval(f, x-h)+feval(f,x-2*h))/(12*h);

    E(1)=0;
    R(1)=0;
    for n=1:2
        h=h/10;
        H(n+1)=h;
        % !!!!修改了下面一行
        % D(n+1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
        D(n+1)=(-feval(f,x+2*h)+8*feval(f, x+h)-8*feval(f, x-h)+feval(f,x-2*h))/
(12*h);
        E(n+1)=abs(D(n+1)-D(n));
        R(n+1)=2*E(n+1)/(abs(D(n+1))+abs(D(n))+eps);
    end
    n=2;
    while((E(n)>E(n+1))&(R(n)>toler))&n<max1
        h=h/10;
        H(n+2)=h;
        % !!!!修改了下面一行
        % D(n+2)=(feval(f,x+h)-feval(f,x-h))/(2*h);
        D(n+2)=(-feval(f,x+2*h)+8*feval(f, x+h)-8*feval(f, x-h)+feval(f,x-2*h))/
(12*h);
        E(n+2)=abs(D(n+2)-D(n+1));
        R(n+2)=2*E(n+2)/(abs(D(n+2))+abs(D(n+1))+eps);
        n=n+1;
    end
    n=length(D)-1;
    L=[H' D' E'];
end
```

具体程序可以查看 question7.m，结果如下

```
                 1     -1.59137417504566e+19                         0
               0.1          75.0857250132672       1.59137417504566e+19
              0.01          75.1734854395341         0.0877604262669056
             0.001          75.1734946942478       9.25471375486796e-06
            0.0001          75.1734946951735       9.25638357900176e-10
             1e-05          75.1734946951736       9.25640009452352e-14
             1e-06          75.1734946951735       1.66798957551393e-14
```

6

1	-0.303872865181299	0
0.1	3.67331992907901	3.9771927942603
0.01	2.96568581351549	0.707634115563516
0.001	2.96551484641143	0.000170967104059866
0.0001	2.96551482918709	1.72243358186292e-08
1e-05	2.96551482918537	1.7225615045183e-12
1e-06	2.96551482918537	8.30625673112041e-16
1e-07	2.96551482918537	4.70996369223222e-16

7

由结果可知

- $f'\left(\frac{1}{\sqrt{3}}\right) = 75.1734946951736$为精确到小数点后 13 位的值
- $f'\left(\frac{1-\sqrt{5}}{2}\right) = 2.96551482918537$为精确到小数点后 13 位的值

两者收敛速度（n 越小表示迭代越少）明显更快很多