第一章上机

第一题

问题叙述：

级数 Sn 如下所示，其精确值为 $\frac{1}{2}(\frac{3}{2}-\frac{1}{N}-\frac{1}{N+1})$ 分别以单精度和双精度数据类型，按正向和反向两种顺序分别计算，指出各计算结果的有效位数，并分析有效位数不同的原因。

$$S_N=\frac{1}{2^2-1}+\frac{1}{3^2-1}+\ldots\ldots+\frac{1}{N^2-1}$$

Matlab 代码和运算结果（代码有另附）：

1.1.1 N=10^2，顺序，双精度

```
%double precision; N = 10^2; in order
close all;clear;clc;
N = 10^2;                              %set the limitation
n = 1 ;                                %set 'n' as the counter
sum = 0;                               %initialize 'sum'
true = 0.5 * (1.5 - 1/N - 1/(N+1));    %calculate the true value

while n <= N
    item = 1 / ((n+1)^2 - 1);          %calculate the item
    sum = sum + item;                  %renew the result
    n = n + 1;                         %renew the counter
end

error = abs(true - sum);               %calculate the error
fprintf('true=%e\tsum=%e\terror=%e\n', true, sum, error);
```

```
true=7.400495e-01    sum=7.401475e-01    error=9.803922e-05
```

有效位数 10^-4

1.1.2 N=10^2，顺序，单精度

```
%single precision; N = 10^2; in order
close all;clear;clc;
N = 10^2;                                    %set the limitation
n = 1 ;                                      %set 'n' as the counter
sum = single (0);                            %initialize 'sum'
true = single (0.5 * (1.5 - 1/N - 1/(N+1)));    %calculate the true value

while n <= N
    item = single (1 / ((n+1)^2 - 1));       %calculate the item
    sum = sum + item;                        %renew the result
    n = n + 1;                               %renew the counter
end

error = single (abs(true - sum));            %calculate the error
fprintf('true=%e\tsum=%e\terror=%e\n', true, sum, error);
```

```
true=7.400495e-01    sum=7.401475e-01    error=9.804964e-05
```

有效位数 10^-4

1.2.1 N=10^2，逆序，双精度

```
%double precision; N = 10^2; in reverse order
close all;clear;clc;
N = 10^2;                                    %set the limitation
n = N ;                                      %set 'n' as the counter
sum = 0;                                     %initialize 'sum'
true = 0.5 * (1.5 - 1/N - 1/(N+1));          %calculate the true value

while n >= 1
    item = 1 / ((n+1)^2 - 1);                %calculate the item
    sum = sum + item;                        %renew the result
    n = n - 1;                               %renew the counter
end

error = abs(true - sum);                     %calculate the error
fprintf('true=%e\tsum=%e\terror=%e\n', true, sum, error);
```

```
true=7.400495e-01    sum=7.401475e-01    error=9.803922e-05
```

有效位数 10^-4

1.2.2 N=10^2，逆序，单精度

```
%single precision; N = 10^2; in reverse order
close all;clear;clc;
N = 10^2;                                   %set the limitation
n = N ;                                     %set 'n' as the counter
sum = single (0);                           %initialize 'sum'
true = single (0.5 * (1.5 - 1/N - 1/(N+1)));    %calculate the true value

while n >= 1
    item = single (1 / ((n+1)^2 - 1));      %calculate the item
    sum = sum + item;                       %renew the result
    n = n - 1;                              %renew the counter
end

error = single (abs(true - sum));           %calculate the error
fprintf('true=%e\tsum=%e\terror=%e\n', true, sum, error);
```

```
true=7.400495e-01    sum=7.401476e-01    error=9.810925e-05
```

有效位数 10^-4

其余代码雷同，仅附上计算结果：

2.1.1 N=10^4，顺序，双精度

```
true=7.499000e-01    sum=7.499000e-01    error=9.998006e-09
```

有效位数 10^-8

2.1.2 N=10^4，顺序，单精度

```
true=7.499000e-01    sum=7.498521e-01    error=4.786253e-05
```

有效位数 10^-4

2.2.1 N=10^4，逆序，双精度

```
true=7.499000e-01    sum=7.499000e-01    error=9.998000e-09
```

有效位数 10^-8

2.2.1 N=10^4，逆序，单精度

```
true=7.499000e-01    sum=7.499000e-01    error=0.000000e+00
```

有效位数未知（大于六位）

3.1.1 N=10^6，顺序，双精度

```
true=7.499990e-01    sum=7.499990e-01    error=1.021627e-12
```

有效位数 10^-11

3.1.2 N=10^6，顺序，单精度

```
true=7.499990e-01    sum=7.498521e-01    error=1.468658e-04
```

有效位数 10^-3

3.2.1 N=10^6，逆序，双精度

```
true=7.499990e-01    sum=7.499990e-01    error=9.999779e-13
```

有效位数 10^-12

3.2.2 N=10^6，逆序，单精度

```
true=7.499990e-01    sum=7.499990e-01    error=5.960464e-08
```

有效位数 10^-7

结果分析：

1、双精度运算结果比单精度高，因为双精度保留的有效位数更多。
2、逆序算法比顺序算法精度高，因为顺序计算时，先算大的数后算小的数，导致
3、注意到 N=10^4、单精度逆序时，输出 error=0，是由于单精度有效位数有限，并不是没有误差。

第二题

问题叙述：

用二分法，求方程 x^4-8*x^3-35*x^2+450*x-1001=0 在区间[4.5,6]内的一个实根，要求精确到小数点后第 6 位。

Matlab 代码和运算结果（代码有另附）：

```
%using dichotomy method to approach the zero point
close all;clear;clc;
min = 4.5;                          %initialize the value of 'min'
max = 6;                            %initialize the value of 'max'
half = 0.5 * (min + max);           %initialize the value of 'half'
temp = half;                        %restore the previous 'half' value

while f(half) ~= 0
    if f(half)*f(min) > 0           %renew the value of 'min' or 'max'
        min = half;
    else
        max = half;
    end
    half = 0.5 * (min + max);
    if abs(temp - half) < 5*10^-7   %judge the loop termination condition
        break;
    end
    temp = half;                    %renew the value of 'temp'
end

fprintf('%.6f\n', temp);

function y = f(x)
y = x^4-8*x^3-35*x^2+450*x-1001;
```

程序运算结果为：x = 5.609789

结果分析：

1、用 temp 储存上一次计算的二分值。

2、判断循环的终止条件是：当前二分值与上一次二分值的差的绝对值是否小于 5*10^-7，因为题目要求精确到小数点后六位。