# 第一章作业

**问题叙述：**

分别以单精度和双精度数据类型用以下近似算法分别计算π的近似值：

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots\right)$$

$$\pi = 6\left(0.5 + \frac{0.5^3}{2\times 3} + \frac{3\times 0.5^5}{2\times 4\times 5} + \frac{3\times 5\times 0.5^7}{2\times 4\times 6\times 7} + \cdots\right)$$

（1） 假定真值未知，要求结果具有至少四位有效数字，给出计算结果；

（2） 如果采用单精度数据类型要求计算结果达到机器精度，此时结果如何？采用双精度数据类型达到单精度机器精度要求以及更高的精度要求，计算结果如何？（测试机器精度：满足 1+ε>1 的最小浮点数）

**问题分析：**

两个级数可以写作：$\pi = 4\sum_{i=0}^{\infty}(-1)^i \times \frac{1}{2i+1}$，$\pi = 6\sum_{i=1}^{\infty}\frac{(2i-1)!!\times 0.5^{2i+1}}{(2i)!!\times(2i+1)} + 3$

根据定义，x’是 x 的一个近似数，表示为：$x^* = \pm 10^k \times 0.\cdots$ 如果 $\left|x - x^*\right| \le \frac{1}{2}\times 10^{k-n}$，则称其近似为 x 的 n 位有效数字。

相对误差：$\varepsilon_a = \frac{\text{当前近似值} - \text{前一近似值}}{\text{当前近似值}} \times 100\%$

由于要求有效数字位数为四位，则可以计算误差容限为：$\varepsilon_s = \left(0.5\times 10^{2-4}\right)\% = 0.5\times 10^{-4}$

计算终止条件为：$\left|\varepsilon_a\right| < \varepsilon_s$

对于第二问，需要先计算出单精度数据类型机器精度，测试机器精度的方法为：不断迭代，寻找满足$1+\varepsilon>1$的最小浮点数。

伪代码如下：

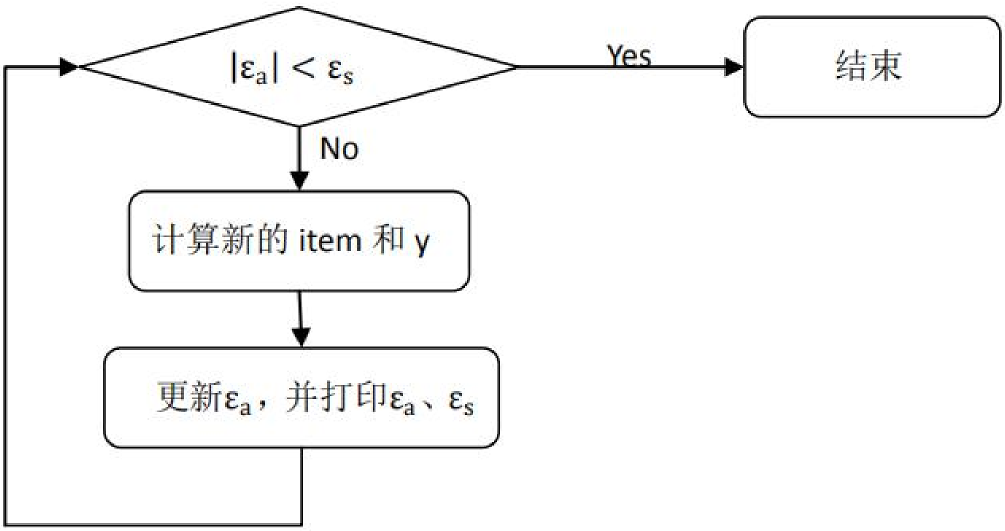
```
Epsilon = 1
DO
    If(Epsilon+1<=1)EXIT
    Epsilon = Epsilon/2
End DO
Epsilon = 2 * epsilon
```

当计算时前后两项的差值小于机器精度时，即停止迭代输出结果。

**算法设计：**

由于每一次加入 y 的新项 item，即每一项具有递推关系，故按照以下步骤

设计算法：




测试精度为机器精度时，将判断条件改为新加项小于机器精度即可。

**第一问 MATLAB 程序：**

第一个级数：

```
%This program is developed to the value of pi
%with single-precision data and double-precision
%data respectively
%The effective number of digits is four
%Use the first series
clear;
clc;
%Calculate using double-precision data
sum=0;                          %The result of calculation
flag=1;                         %The symbol of each item
i=0;                            %The index of each item
e_a=1;
while e_a>0.5*0.0001
    sum=sum+4*flag/(2*i+1);
    e_a=4/((2*i+1)*sum);
    flag=-flag;
    i=i+1;
end
format long;
sum,                            %print
%Calculate using single-precision data
sum=0;
flag=1;
```

```
i=0;
e_a=1;
while e_a>0.5*0.0001
    sum=sum+single(4*flag/(2*i+1));
    e_a=single(4/(2*i+1))/sum;
    flag=-flag;
    i=i+1;
end
sum,                         %print
```

程序运行结果：

单精度：3.1416788

双精度：3.141671189676991

第二个级数：

```
%This program is developed to the value of pi
%with single-precision data and double-precision
%data respectively
%The effective number of digits is four
%Use the second series
clear;
clc;
%Calculate using double-precision data
sum=3;                            %The result of calculation
i=1;                              %The index of each item
e_a=1;
while e_a>0.5*0.0001
     sum=sum+6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i));
     e_a=6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i)*sum);
     i=i+1;
end
format long;
sum,                              %print
%Calculate using single-precision data
sum=3;
i=1;
e_a=1;
while e_a>0.5*0.0001
    sum=sum+single(6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2
*i)));
    e_a=single(6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i))
)/sum;
    i=i+1;
end
sum,                             %print
```

```
%Calculate the double factorial of a number
function y=fac(x)
     if x==0
        y=1;
        return;
    end
    y=1;
    for i=x:-2:1
        y=y*i;
    end
end
```

程序运行结果：

单精度：3.1415765

双精度：3.141576715774867

**第一问进一步分析：**

单精度数据为小数点后七位，双精度数据为小数点后十五位，从保留有效位数上来看，实际上用单精度和双精度数据都可以得到准确的数据。

我们通过分析代码实现的过程发现：第二个级数比第一个级数收敛得更快。所以可以寻找更快收敛的级数来优化算法使计算速度变得更快。

**第二问 MATLAB 程序：**

第一个级数：

```
%This program is developed to the value of pi
%with single-precision data and double-precision
%data respectively
%The calculation accuracy is the machine accuracy
%of single-precision data
%Use the first series
clear;
clc;
%Calculate machine accuracy
epsilon=1;
while (epsilon+1>1)
    epsilon=single(epsilon/2);
end
epsilon=epsilon*2;
%Calculate using single-precision data
sum=0;                       %The result of calculation
i=0;                         %The index of each item
flag=1;                      %The symbol of each item
while single(4/(2*i+1))>epsilon
    sum=sum+flag*single(4/(2*i+1));
    flag=-flag;
```

```
    i=i+1;
end
format long;
sum,                            %print
%Calculate using double-precision data
sum=0;
i=0;
flag=1;
while 4/(2*i+1)>epsilon
    sum=sum+4*flag/(2*i+1);
    flag=-flag;
    i=i+1;end
sum,                            %print
```

程序运行结果：

单精度：3.1415970

双精度：3.141592593985150

第二个级数：

```
%This program is developed to the value of pi
%with single-precision data and double-precision
%data respectively
%The calculation accuracy is the machine accuracy
%of single-precision data
%Use the first series
clear;
clc;
%Calculate machine accuracy
epsilon=1;
while (epsilon+1>1)
    epsilon=single(epsilon/2);
end
epsilon=epsilon*2;
%Calculate using single-precision data
sum=3;                           %The result of calculation
i=1;                            %The index of each item
while
    single(6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i)))>ep
    silon
     sum=sum+single(6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2
     *i)));
    i=i+1;
end
format long;
sum,                            %print
```

```
sum=3;
i=1;
while
     6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i))>epsilon
    sum=sum+6*0.5^(2*i+1)*fac(2*i-1)/((2*i+1)*fac(2*i));
     i=i+1;
end
sum,                             %print
%Calculate the double factorial of a number
function y=fac(x)
    if x==0
        y=1;
        return;
    end
    y=1;
    for i=x:-2:1
        y=y*i;
    end
end
```

程序运行结果：

单精度：3.1415923

双精度：3.141592511157862

**第二问进一步分析：**

观察代码运行结果发现，由于使用了单精度数据，机器精度即单精度数据的最小位，计算结果约为 1.1921e-07。

即使是都使用了机器精度，单精度数据和双精度数据的计算结果仍有差异。单精度由于保留位数少，在计算每一项时都会有舍入误差，累加后误差会增加。因此在结果上使用单精度数据会比使用双精度数据误差更大。

横向分析两个级数，第一个级数运行了 10e07 次而第二个级数运行了 6 次。在迭代次数上，次数越少，舍入误差越少，计算速度越快，计算结果越精确。因此收敛越快的级数，对我们的计算越有利。

**更高精度的分析：**

由于级数一迭代次数过多，我们选择收敛更快的级数二来分析。

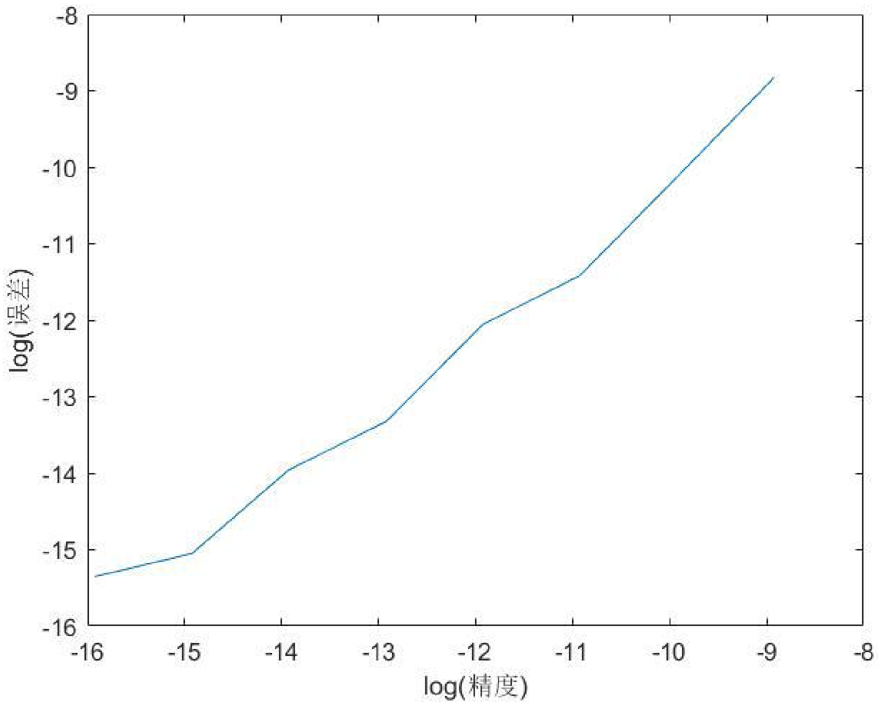
选择更高的精度，分别用第二个级数来计算 π 的值，并将其与真值进行比较，分析精度与误差之间的关系：

MATLAB 程序：

```
clear;
clc;
%Calculate machine accuracy
epsilon=1;
while (epsilon+1>1)
    epsilon=single(epsilon/2);
```

```
end
epsilon=epsilon*2;
epsilon=double(epsilon);
for i=1:8
    e_p(i)=epsilon/10^(i+1);
    sum=3;
    j=1;
    while
        6*0.5^(2*j+1)*fac(2*j-1)/((2*j+1)*fac(2*j))>e_p(i)

        sum=sum+6*0.5^(2*j+1)*fac(2*j-1)/((2*j+1)*fac(2*j))
        ;
        flag=-flag;
        j=j+1;
    end
    delta(i)=abs(sum-pi);
end
plot(log10(e_p),log10(delta));

xlabel('log(精度)');

ylabel('log(误差)');

%Calculate the double factorial of a numbe
function y=fac(x)
    if x==0
        y=1;
        return;
    end
    y=1;
    for i=x:-2:1
        y=y*i;
    end
end
```

得到精度与误差的关系如下：




图表 精度与误差的关系

**小结：**

分析结果我们可以得知，使用单精度数据比使用双精度数据的误差更大；级数收敛的快慢也会影响结果，收敛越快，迭代次数越少，误差越小，运算速度越快；另外，计算需要精度越高，与真值的误差越小。

我们在进行级数的数值计算时，尽量选取收敛快的级数，使用双精度数据并尽可能提高精度，这样对我们的计算最有利。