π的计算

李青林,5110309074
郑辉煌,5110209289

1 实验任务1

从单位圆正方形开始，成倍增加边数，求π的近似值

设边数为$4 \cdot 2^n$时，边长为a_n
易得

$$a_{n+1} = \sqrt{\left(\frac{a_n}{2}\right)^2 + \left(1 - \sqrt{1 - \left(\frac{a_n}{2}\right)^2}\right)^2} = \sqrt{2 - \sqrt{4 - a_n^2}}$$

$4 \cdot 2^n$边型面积为

$$S_{n+1} = 4 \cdot 2^n \cdot \frac{1}{4} \cdot a_n = 2^n a_n \approx \pi$$

初始条件为$a_0 = \sqrt{2}$

MATLAB代码:

```
function task1(n)
    a = zeros(1,100);
    a(1) = sqrt(2);
    for i = 2:n
        a(i) = sqrt(2-sqrt(4-a(i-1)^2));
    end
    pi = 2^n * a(n);
    vpa(pi,20)
end
```

n取不同值时，π的值

n	π
10	3.1415914215046352176
14	3.1415926548075892022
20	3.1415965537048196858

可以看出，使用该方法得到的结果精度比较低，并且由于计算过程中的误差积累，反而导致迭代次数增加，误差反而增加的事情发生

2 实验任务2

2.1

用反正切函数的幂级数展开式结合有关公式求π,若要精确到50位数字,试比较简单公式和Machin公式所用的项数.

$$\text{arctan}x = \sum_{n=1}^{\infty}(-1)^{n-1}\frac{x^{2n-1}}{2n-1}$$

简单公式

$$\frac{\pi}{4} = \text{arctan}\frac{1}{2} + \text{arctan}\frac{1}{3}$$

展开,变形得

$$\pi = 4\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{2n-1}\left(\frac{1}{2^{2n-1}} + \frac{1}{3^{2n-1}}\right)$$

MATLAB代码:

```
function task2_1(n)
    ret = 0; k = 1;
    for i=1:n
        ret = ret + k / (2 * i - 1) * ( 1 / 2^(2 * i - 1) ...
        + 1 / 3^(2 * i - 1));
        k = k * -1;
    end
    vpa(ret*4,50)
end
```

n	π
19	3.1415926535899427740616829396458342671394348144531
20	3.1415926535897932384626433832795028841971693993751

Machin公式

$$\frac{\pi}{4} = 4\text{arctan}\frac{1}{5} - \text{arctan}\frac{1}{239}$$

展开,变形得

$$\pi = 4\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{2n-1}\left(\frac{4}{5^{2n-1}} - \frac{1}{239^{2n-1}}\right)$$

MATLAB代码:

```
function task2_2(n)
    ret = 0; k = 1;
    for i=1:n
        ret = ret + k / (2 * i - 1) * ( 4 / 5^(2 * i - 1) ...
        - 1 / 239^(2 * i - 1));
        k = k * -1;
    end
    vpa(ret*4,50)
end
```

n	π
8	3.1415926535886029569155653007328510284423828125000
9	3.1415926535897932384626433832795028841971693993751

可以看出，要精确到50位，简单公式需要20项，而Marchin公式只需要9项，因此Marchin公式更精确

2.2

验证公式

$$\frac{\pi}{4} = \text{arctan}\frac{1}{2} + \text{arctan}\frac{1}{5} + \text{arctan}\frac{1}{8}$$

试试用此公式右端作幂级数展开完成任务2.1所需的项数

$$
\begin{aligned}
&\tan\left(\arctan\frac{1}{2}+\arctan\frac{1}{5}+\arctan\frac{1}{8}\right)\\
=&\frac{\frac{1}{2}+\tan\left(\arctan\frac{1}{5}+\arctan\frac{1}{8}\right)}{1-\frac{1}{2}\cdot\tan\left(\arctan\frac{1}{5}+\arctan\frac{1}{8}\right)}\\
=&\frac{\frac{1}{2}+\frac{1}{3}}{1-\frac{1}{2}\cdot\frac{1}{3}}\\
=&1=\tan\frac{\pi}{4}
\end{aligned}
$$

$\therefore$公式成立
对公式展开，变形得

$$\pi = 4\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{2n-1}\left(\frac{1}{2^{2n-1}}+\frac{1}{5^{2n-1}}+\frac{1}{8^{2n-1}}\right)$$

MATLAB代码:

```
function task2_3(n)
    ret = 0; k = 1;
    for i=1:n
        ret = ret + k / (2 * i - 1) * ( 1 / 2^(2 * i - 1) ...
        + 1 / 5^(2 * i - 1)+ 1 / 8^(2 * i - 1));
        k = k * -1;
    end
    vpa(ret*4,50)
end
```

n	π
19	3.1415926535899427740616829396458342671394348144531
20	3.1415926535897932384626433832795028841971693993751

要精确到50位，该公式需要20项，和简单公式相同。原因是

$$\arctan\frac{1}{5}+\arctan\frac{1}{8}=\arctan\frac{1}{3}$$

3 实验任务3

用数值积分计算π,分别用梯形法和Simpson法精确到10位数字,用Simpson法精确到15位数字.

$$4\int_0^1 \frac{1}{1+x^2}\mathrm{d}x = \pi$$

梯形公式MATLAB代码:

```
function task3_1(n)
    x=0:1/n:1;
    y=1./(1+x.^2);
    s=2*sum(y)-1-0.5;
    2*s/n
end
```

$n = 100000$，结果为3.141592653589588
即要精确到10位，需要划分100000个区间
Simpson公式MATLAB代码:

```
function task3_2(n)
    x = 0:1/n:1; m = length(x); s = 0;
    for i = 1:m-1
        s = s + (x(i+1)-x(i)) / 6  * (1 / ( 1 + x(i)^2) ...
            + 4 / (1 +(x(i) + x(i+1))^2 / 4) +...
            1 / ( 1 + x(i+1)^2) );
    end
    vpa(s*4,15)
end
```

$n = 18$，结果为3.14159265357156
$n = 50$，结果为3.14159265358979
即Simpson公式只需要划分18个区间即可精确到10位,划分50个区间可以精确到15位
因此Simpson公式比梯形公式更精确。

4 实验任务4

4.1

用Monte Carlo法计算π,除了加大随机数,在随机数一定时可重复算若干次后求平均值,看能否求得5位精确数字?

Monte Carlo方法:

MATLAB代码(每次撒k个点，运行n次求平均):

```
function task4_1(k, n) %rand k points, n test average
    sum = 0; a = zeros(n);
    for i = 1:n
        a(i) = monteCalPi( k );
        sum = sum + a(i);
    end
    vpa(sum / n, 10)
end
function y = monteCalPi( k )
    m = 0;
    for n = 1:k
        if rand(1)^2 + rand(1)^2 <= 1
            m = m + 1;
        end;
    end
    y = 4 * m / k;
end
```

取$k = 10000, n = 10000$时,算出$\pi = 3.14157176$

模拟Buffon实验:

模拟方法：将针看作长为l的线段，随机确定中点到平行线的距离x和针与平行线的夹角α,再判断相交
MATLAB代码(平行线间距d，针长$l(l \le d)$, m次投针):

```
function task4_2( l, d, m )
    n = 0;
    for k = 1:m
        x = unifrnd(0, d/2);
        alpha = unifrnd(0, pi);
        if x < 0.5 * l * sin(alpha)
            n = n+1;
        end
    end
    y = 2 * l / d * m / n;
    vpa(y, 10)
end
```

取$l = 10, d = 20, m = 10000$,得$\pi = 3.143665514$
实验证明，随机算法通常难以保证精度，并且计算结果的误差时大时小

5 实验任务5

e是一个重要的超越数

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$$

$$e = 1 + 1 + \frac{1}{2!} + \cdots + \frac{1}{n!} + \frac{e^\theta}{(n+1)!}$$

还有

$$e = \left(1+\frac{1}{n}+\frac{k}{n^2}\right)^n$$

试用上述公式或其他方法近似计算e

使用公式$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$

MATLAB代码:

```
function task5_1( n )
    e = (1 + 1/n) ^ n;
    vpa(e, 50)
end
```

取$n = 10000$时，得$e = 2.718145927$，精确到小数点后3位.

使用公式$e = 1 + 1 + \frac{1}{2!} + \cdots + \frac{1}{n!} + \frac{e^{\theta}}{(n+1)!}$

MATLAB代码:

```
function task5_2( n )
    e = sym(1);
    temp = 1;
    for i = 1:n
        temp = temp * i;
        e = e + sym(1 / temp);
    end
    vpa(e, 10)
end
```

取$n = 10$，得$e = 2.718281801$，可以精确到小数点后7位
对比可知，级数法精度更高.