差分方程

差分方程就是中学数列递推公式，已知前项推出后项，许多差分方程是无法求出通项的，所以运用计算机求解差分方程就是通过 for 循环从 1 开始不断向后递推

例：斐波那契数列$a_{n+2}=a_{n+1}+a_n, a_1=a_2=1$

这是一个二阶的递推数列，在写循环程序的时候，如果你要推导第n项的值，则循环只需要算$1:n-2$，循环的最后一项到底是$n-2$还是$n-1$还是n，得看初值是几个。

如果你要算到n项，那最后一次循环算得是$f(n)=f(n-1)+f(n-2)$和$f(i+2)=f(i+1)+f(i)$比较得出循环的最后一项$i=n-2$

```
function f=fib(n)
    f=zeros(n,1);%向量初始化
    f(1)=1;f(2)=1;%赋初值
    for i=1:n-2
        f(i+2)=f(i+1)+f(i);%循环计算斐波那契数列
    end
end
```

命令行窗口

```
>> f=fib(10)

f =

     1
     1
     2
     3
     5
     8
    13
    21
    34
    55
```

例：遗传病问题：有一种遗传病基因为a，正常的基因为A，已知父母中至少有一人基因完全正常记为AA，另一人基因为Aa，则后代的基因产生的概率如下

	AA-AA	AA-Aa
AA	1	1/2
Aa	0	1/2

已知a_n和b_n分别是第n代人中AA基因型人数占比和Aa基因型人数占比，建立差分方程，分析a_1和b_1取不同值时，两种基因型人数占比的变化。

根据上表，可以得出，第$n+1$代人是AA基因型人数占比，由前一代人中$AA-AA$组合的人数占比与前一代人中$AA-Aa$组合的人数占比的一半构成，即

$$a_{n+1}=a_n+\frac{1}{2}b_n$$

第$n+1$代人是Aa基因型人数占比，由前一代人中$AA-Aa$组合的人数占比的一半构成，即

$$b_{n+1}=\frac{1}{2}b_n$$

最终得到差分方程组

$$
\begin{cases}
a_{n+1} = a_n + \dfrac{1}{2} b_n \\
b_{n+1} = \dfrac{1}{2} b_n
\end{cases}
$$

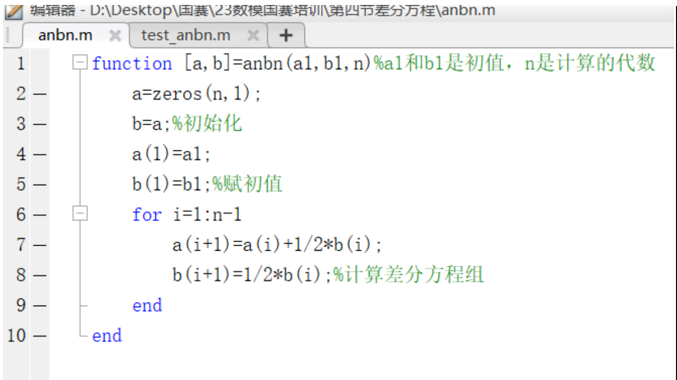
注意：将两式相加可得到$a_{n+1} + b_{n+1} = a_n + b_n = 1$，因为两种人占比和应该是 100%

这里第一代人a_1和b_1取不同值的时候，a_n和b_n的变化趋势可能会不同，列举三种情况。

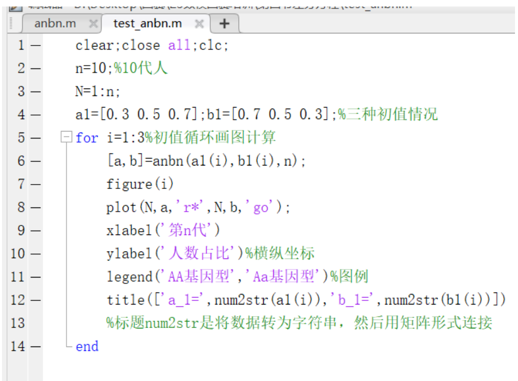
（1）$a_1 = 0.3, b_1 = 0.7$；（2）$a_1 = b_1 = 0.5$；（3）$a_1 = 0.7, b_1 = 0.3$

本题中，当差分方程变为差分方程组时，经需要在循环和初值处改变，其中循环改为

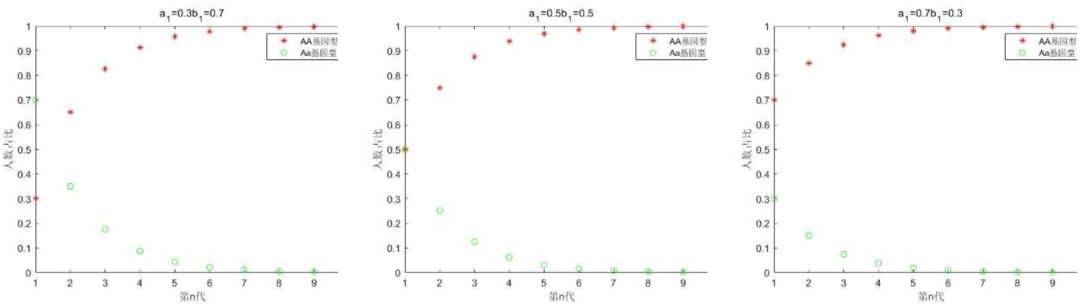
```
for i=1:n-1
a(i+1)=a(i)+1/2*b(i);
b(i+1)=1/2*b(i);
end
```



```
function [a,b]=anbn(a1,b1,n)%a1和b1是初值，n是计算的代数
    a=zeros(n,1);
    b=a;%初始化
    a(1)=a1;
    b(1)=b1;%赋初值
    for i=1:n-1
        a(i+1)=a(i)+1/2*b(i);
        b(i+1)=1/2*b(i);%计算差分方程组
    end
end
```



```
clear;close all;clc;
n=10;%10代人
N=1:n;
a1=[0.3 0.5 0.7];b1=[0.7 0.5 0.3];%三种初值情况
for i=1:3%初值循环画图计算
    [a,b]=anbn(a1(i),b1(i),n);
    figure(i)
    plot(N,a,'r*',N,b,'go');
    xlabel('第n代')
    ylabel('人数占比')%横纵坐标
    legend('AA基因型','Aa基因型')%图例
    title(['a_1=',num2str(a1(i)),'b_1=',num2str(b1(i))])
    %标题num2str是将数据转为字符串，然后用矩阵形式连接
end
```



三种初值的变化趋势

结论是无论初值如何取，数列的变化趋势永远是 $\lim\limits_{n\to\infty} a_n = 1$, $\lim\limits_{n\to\infty} b_n = 0$，最终这种隐性的遗传基因会自行消亡。

针对本题中的差分方程组，可写成矩阵形式，即

$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} a_n \\ b_n \end{pmatrix}$$

记 $X_n = \begin{pmatrix} a_n \\ b_n \end{pmatrix}, A = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{pmatrix}$，则方程可变为 $X_{n+1} = AX_n$

```
function X=anbn_else(a1,b1,n)
X=zeros(2,n);%初始化
X(1,1)=a1;
X(2,1)=b1;%赋初值
A=[1 1/2;
    0 1/2];%矩阵
for i=1:n-1
    X(:,i+1)=A*X(:,i);%循环计算
end
end
```

效果是一样的