

# 实验三

## MATLAB程序设计

### P58 实验三

以下作业任选一组，编程,2022.12月8日18: 10-22: 00上机调试完成。

保存好代码及运行结果。待最后统一上交

#### A组

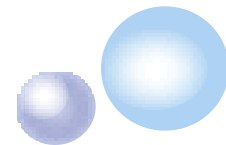
##### 练习1

1. 任选2小题  
3,5,7,9

#### B组

##### 练习1

1. 任选2小题  
2,4,6,8



## **实验目的：**

**熟悉MATLAB程序的基本结构；**  
**了解MATLAB流程控制命令；**  
**能够编写简单的MATLAB命令文件和函数文件。**

## **实验内容：**

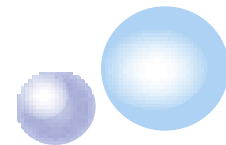
- (1) MATLAB命令文件与函数文件；**
- (2) MATLAB程序的基本结构，包括顺序结构、循环结构和选择结构；**
- (3) MATLAB流程控制命令。**



# MATLAB程序设计

- MATLAB中各种命令可以完成许多单一的任务,对于某些较为复杂的问题,仅靠现有的命令或函数来解决,往往是难以达到目的。为此,要运用MATLAB编程语言编制程序,形成M-文件。
- 程序是使计算机完成各项运算的命令集,运行一个编制好的程序,计算机会从第一条命令行开始,一行接一行地执行相应的命令,直到终止。
- 程序编写调试完成后,需要存盘,形成永久性文件,可以随时对它进行调用或修改。
- 文件名以字母开头,但不能用专用变量名,如pi,ans,eps等。

## 5.1 命令文件与函数文件



用MATLAB语言编写的程序文件,称为M文件. M文件包含命令文件和函数文件两种形式.

**命令文件** 仅由MATLAB命令组成,不接受参数的输入和输出,它的执行方式也比较简单.

前面编写的程序文件都是命令文件。

**函数文件**则是利用MATLAB语言构造了一个新的MATLAB函数,这个函数的使用方法与库函数一样,可以传递参数.

# 函数文件的建立和调用

函数文件由function语句引导，基本结构为：

**function** [输出参数1,...,输出参数n]=函数名(输入参数1,..., 输入参数m)

例如：**function** [y1,y2]=fun1(x,y,a,n) %以fun1作为函数名保存.

**例1** 编写函数文件，求半径为r的球的体积和表面积。

在编辑器窗口输入程序如下：

函数文件

```
function [V,S]=fsphere(r)
V=(4*pi*r^3)/3;
S=4*pi*r^2;
```

保存文件,系统默认文件名为函数名**fsphere**,不要修改。

若求半径为4的球体体积和表面积,只需在命令窗口键入：  
[V,S]=fsphere(4)，回车即可得到结果。

**例2** 建立计算 $p=(a+b)^n$ ,  $q=(a-b)^n$ 的函数,并计算  
 $a=8, b=3, n=5$ 时 $p, q$ 的值。

**首先, 建立函数文件fun1:**

```
function [p,q]=fun1(a,b,n)
    p=(a+b).^n;
    q=(a-b).^n;
```

**其次, 调用fun1,计算 $a=8, b=3, n=5$ 时 $p, q$ 的值:**

```
[p,q]=fun1(8,3,5);
```

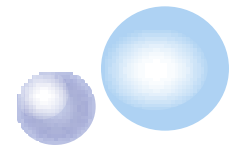
**或**

```
a=8;b=3;n=5;
[p,q]=fun1(a,b,n)
```

**运行结果:**

```
p =
    161051
q =
     3125
```

# M-文件中数据的输入、输出方式



## 1、直接赋值输入:

```
a=[1,2;3,4];
```

```
s= 'any string!';
```

## 2、提示对话输入(input命令)

```
a=input('请输入矩阵 a=');
```

```
s=input('Please input s=');
```

通过键盘  
输入

## 1、直接输出:

```
x
```

## 2、格式控制输出(sprintf命令)

```
fprintf('x=%.0f, y=%.5f\n',pi,pi);
```

例如:

```
fprintf('x=%.5f y=%.0f\n', pi, sqrt(2));
```

运行结果为:  $x=3.14159$   $y=1$

```
fprintf('x=%.5f y=%.4f\n', sqrt(2), pi);
```

运行结果为:  $x=1.41412$   $y=3.1416$

### 例3：建立符号函数sgn(x)

```
function sn=sgn(x)
    if x>0
        sn=1;
    elseif x==0
        sn=0;
    else
        sn=-1;
    end
```

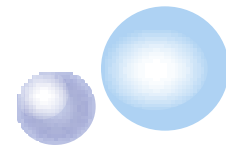
以sgn作为文件名存盘，即建立了函数。

调用：

在命令区执行： **sn=sgn(10)或sn=sgn(-2)**



## 5.2 MATLAB程序的基本结构



MATLAB有三种基本的程序结构：顺序结构、循环结构和选择结构

程序一般包含：

数据输入

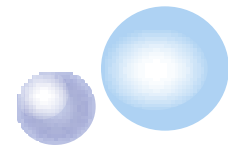
```
a=input('请输入数组 a[n]= ');  
[m,n]=size(a);  
M=a(1);
```

数据处理

```
for i=2:n  
    if M<a(i)  
        M=a(i);  
    end  
end
```

结果输出

```
fprintf(' M=%.5f\n ',M);
```



# 1、MATLAB程序的顺序结构

**顺序结构**是最简单的一种程序结构，在本节以前，我们写的程序基本都是顺序结构。它由多条**MATLAB**语句顺序构成，各语句之间用逗号或分号隔开（若不加逗号或分号，则必须分行编写），程序执行时，也是按照由上至下的顺序进行。



## 2 MATLAB程序的循环结构

有限次循环for-end结构和条件循环while-end结构.

### (1) 有限次循环for-end结构

**格式:** **for** n=n1 : step : n2

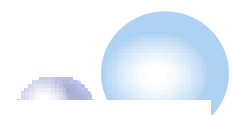
命令集1

**end**

命令集2

**注意:** 结构中for与end要成对出现.

**作用:** 循环变量n从n1开始, 执行命令集1, 遇到end时n自动增加step步长, 同时与n2比较, 当n不超过n2时重复执行命令集1; 当n超过n2时执行命令集2. 当步长为1时, 格式中step可以省略, 即为for n=n1 : n2格式.



## 例4：求n个奇数和： $s=1+3+5+\dots+(2n-1)$

---

程序：

```
n=input('please input n=');  
s=0;  
for i=1:n  
    s=s+(2*i-1);  
    fprintf('i=%.0f, s=%.0f\n',i,s);  
end
```

---

运行程序时, 命令窗口显示：

请输入n=

在等号后面输入100，回车，结果如下：

.....

i=99, s=9801

i=100, s=10000

## 例5：求阶乘： $p=1 \times 2 \times 3 \times \dots \times n$

---

程序：

```
n=input('请输入 n= ');  
p=1;  
for i=1:n  
    p=p*i;  
    fprintf(' i=%.0f, p=%.0f\n ',i,p);  
end
```

---

运行结果如下：

.....

*i=17, p=355687428096000*

*i=18, p=6402373705728000*

*i=19, p=121645100408832000*

*i=20, p=2432902008176640000*

## 例6

设A是一个 $3 \times 5$ 矩阵,

$$A(i,j) = \frac{1}{i+j}, \text{试编程}$$

写出矩阵A.

```
m=3; n=5;
```

```
for i = 1:m
```

```
    for j = 1:n
```

```
        A(i,j) = 1/(i+j);
```

```
    end
```

```
end
```

```
A
```

## 例7 利用 $e \approx 1 + 1 + 1/2! + 1/3! + \dots + 1/n!$ ，求 $e$ 的近似值

程序：

```
n=input('请输入 n= ');
p=1;e=1;
    for i=1:n
        p=p*i;
        p1=1/p;
        e=e+p1;
        fprintf(' i=%.0f, p=%.0f, e=%.8f \n ',i,p,e);
    end
```

**想一想：** 如何有效地控制 $e$ 的近似值的精度，或者说如何修改程序使其根据近似值的精度要求自动控制循环次数？

运行程序时,命令窗口显示: **请输入 $n=$**   
在等号后面输入**10**，回车，运行结果如下

.....

**$i=9, s=2.71828153$**

**$i=10, s=2.71828180$**



## (2) 条件循环while-end结构

格式:    **while** 条件  
          命令集1;  
          **end**  
          命令集2;

其中“条件”是控制循环结束的表达式，一般是由逻辑运算或关系运算及算术运算组成的逻辑表达式。

结构中**while**与**end**要成对出现。

若表达式为真（值非零），执行命令集1,遇到**end**时，返回继续判断表达式真假,若表达式为假（值为零），则停止循环，执行命令集2.

在设计**while-end**循环结构时，应在循环模体命令集1内改表达式的内容，确保在执行了一定次数之后可以结束循环；否则就成了“死循环”，即无限次重复执行该循环.



**例8** 利用近似公式 $e \approx 1 + 1 + 1/2! + 1/3! + \dots + 1/n!$  求 $e$ 的近似值,  
要求精确到  $.10^{-8}$

程序:

```
p=1;e=1;r=1;i=1;  
while r>=1.0e-8  
    p=p*i;  
    r=1/p;  
    e=e+r;  
    fprintf(' i=%.0f, r=%.9f, e=%.9f \n ',i,r,e);  
    i=i+1;  
end
```

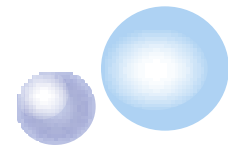
运行结果如下:

.....

**k=10, s=2.7182818011**

**k=11, s=2.7182818262**

**k=12, s=2.7182818283**



### 3 MATLAB程序的选择结构

顺序结构的程序虽然能解决计算、输出等问题，但不能做判断再选择。对于要先做判断再选择的问题就要使用**选择结构**。

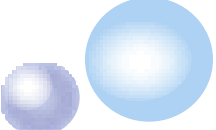
三种常用选择结构：if-else-end结构、 switch-case-end结构、  
try- catch-end结构

#### (1) if-else-end结构

```
if 条件
    命令集1;
end
```

```
if 条件
    命令集1;
else
    命令集2;
end
```

```
if 条件1
    命令集1;
elseif 条件2
    命令集2;
else
    命令集3;
end
```



**例9** 通过键盘输入n个实数，找出其中的最大数并输出。

**程序：**

```
a=input('请输入数组 a[n]= ');
```

```
[m,n]=size(a);
```

```
M=a(1);
```

```
for i=2:n
```

```
    if M<a(i)
```

```
        M=a(i);
```

```
    end
```

```
end
```

```
fprintf(' M=%.5f\n ',M);
```

**例10** 编写一个函数将百分制成绩转换为优(A)，良(B)，中(C)，差(D)四等级。

**分析：**按照通常的等级划分，一般90-100分为优(A)，78-89分为良(B)，60-77分为中(C)，60分以下的为差(D)，因此，可以用多项选择判断结构来实现。

**程序：**

```
fs= input('请输入考试得分fs: ');
```

```
if fs>=90
```

```
    jb=' A ';
```

```
elseif fs>=78
```

```
    jb=' B ';
```

```
elseif fs>=60
```

```
    jb=' C ';
```

```
else
```

```
    jb=' D ';
```

```
end
```

```
jb
```

运行程序，MATLAB命令行窗口将出现以下文字：

**请输入考试得分fs:**

在冒号后面输入 **89**，按回车键，就得到下面结果：

**jb =**

**' B '**

再次运行程序，在冒号后面输入 **91**，就得到 **jb = ' A '**

## (2) switch-case-end结构

**格式如下：**

```
switch 变量或表达式
case 数值1
    命令集1
case 数值2
    命令集2
    .....
case 数值n
    命令集n
otherwise
    命令集n+1
end
```

**例11** 用switch语句实现将百分制成绩转换为优(A),良(B),中(C),合格(D), 不合格(E)五等级.

```
fs= input('请输入分数fs: ');
switch fix(fs/10)
    case {10,9}
        grade='A'
    case 8
        grade='B'
    case 7
        grade='C'
    case 6
        grade='D'
    otherwise
        grade='E'
end
```

### (3) try-catch-end结构

try-catch-end结构用于捕获程序异常。异常捕获机制是MATLAB中用于管理程序运行期间错误的一种结构化方法。

try-catch-end结构的一般形式如下：

```
try
    命令集1
catch
    命令集2
end
```

同学们请在上机时运行教材**例5-15**，体会其用法。

其执行过程是：try语句先执行命令集1语句组，若执行中出错，则将错误信息自动赋值给保留变量lasterr,同时中止执行命令集1，转而去执行命令集2.该结构可为编程者提供错误处置预案，避免死机或其它意外情形发生。



## 5.3 程序的流程控制

编程解决实际问题时，若要在一定条件下跳出当前循环，或终止程序运行，或暂停程序运行等，就需要对程序流程进行控制。MATLAB提供的常用的流程控制命令有continue、break、pause、warning、return、echo和error等。

### (1) continue命令

continue命令通常用在for和while循环结构中，并与if一起使用。作用：在循环过程中，当满足某特定条件时，结束本次循环，即跳过当前循环体中剩余的任何语句，继续下一次循环，直到循环结束。

**例12** 找出1-100之间能被5整除的整数。

**运行结果如下：**

**result=**

**5 10 15 20 25 30 35 40 45 50**

**55 60 65 70 75 80 85 90 95 100**

```
result=[];  
for i=1:100  
    if mod(i,5)~=0  
        continue  
    end  
    result=[result,i]  
end
```

## (2) break 语句

**break**一般用来终止while或for循环, 通常和if语句结合使用. 如果条件满足, **break**命令将立即跳出当前循环。在多层循环嵌套中, **break**只终止最内层的循环。

**例13** Fibonacci 数组的元素满足Fibonacci规则:

$$a_1 = a_2 = 1, a_{k+2} = a_k + a_{k+1} (k = 1, 2, 3, \dots)$$

求出该数组中第一个大于10000的元素。

运行结果如下:

*a=*

*1 1 2 3 5 8 13 21 34 55 89*  
*144 233 377 610 987 1597 2584*  
*4181 6765 10946*

*b=*

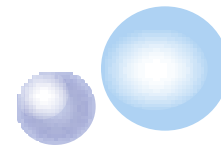
*10946*

第1个大于10000的数是10946。

```
n=100;  
a=[1,1];  
for i=3:n  
    p=a(i-1)+a(i-2);  
    a=[a,p];  
    if p>10000  
        break;  
    end  
end  
a  
b=a(length(a))
```



### (3) pause 语句



调用格式	功能
<code>pause</code>	暂停执行M文件,用户从键盘上按任意键后继续执行.
<code>pause(n)</code>	暂停n秒后继续执行M文件。可实现动画效果.
<code>pause on</code>	允许其后的暂停命令起作用
<code>pause off</code>	不允许其后的暂停命令起作用

**例14** 动态显示数列极限  $\lim_{n \rightarrow \infty} (1 + \frac{1}{n})^n$  的变化趋势

`hold on`

`axis([0,100,2,3]);`

**for** `n=1:90`

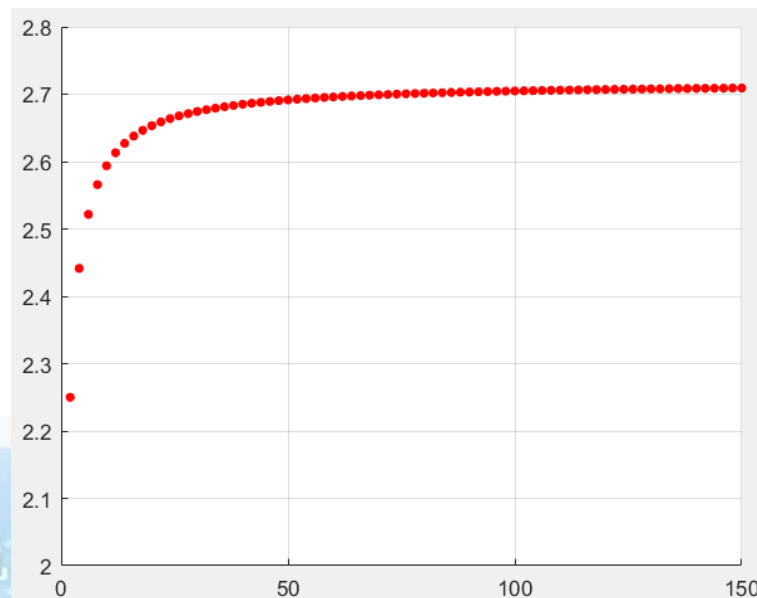
`an=(1+1/n)^n;`

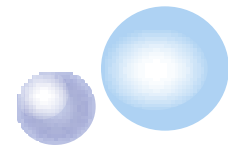
`plot(n,an,'r*');`

**pause(0.05);**

`fprintf('n=%d an=%.4f\n',n,an);`

**end**





## (4) return语句

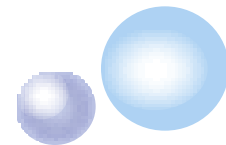
return语句终止当前命令的执行，并且返回到调用函数或者等待键盘输入指令。

调用格式如下：

```
.....  
if 表达式  
    return;  
end
```

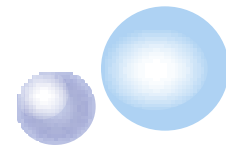
.....

其作用是当执行到if模块且表达式为真时，执行return语句，程序被终止，提前结束程序的运行。



## (5) warning命令

调用格式	功能
<b>warning('message')</b>	显示警告信息message，其中message为文本信息
<b>warning('message', a1,a2,... )</b>	显示警告信息message，其中message包含转义字符，且每个转义字符的值将被转化为,a1,a2,...的值
<b>warning on</b>	显示其后所有警告命令警告信息
<b>warning off</b>	不显示其后所有警告命令警告信息
<b>warning debug</b>	当遇到一个警告时，启动测试程序



## 例15 编写一个求 $f(x) = x^2 \sin x + \log_2 x$ 的函数文件

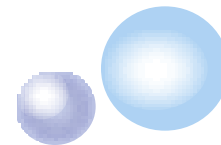
第1步：编写MATLAB函数文件。

```
function y=fun5_19(x)
a1='负数';
a2=0;
if x<0
    y=[];
    warning('x的值不能为%s!',a1);
    return;
elseif x==0
    y=[];
    warning(' x的值不能为%d!',a2);
    return;
else
    y=x.^2.*sin(x)+log(x)./log(3);
end
```

第2步：运行函数文件

```
>>y= fun5_19 (-1)
警告: x的值不能为负数!
> In fun5 (line 7)
y =
    []
>> fun5_19 (0)
警告: x的值不能为0!
> In fun5 (line 11)
ans =
    []
>> y= fun5_19 (9)
y =
    35.3816
```

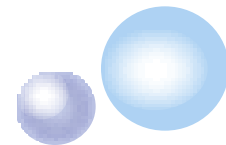
## (6) echo命令和error命令



调用格式	功能
echo on	显示M文件执行过程
echo off	不显示M文件执行过程
echo	在上面两个命令之间切换
echo filename on	显示函数名为filename的函数文件的执行过程
echo filename off	关闭函数名为filename的函数文件的执行过程
echo filename	在上面两个命令之间切换
echo on all	显示所有函数文件的执行过程
echo off all	关闭所有函数文件的执行过程

**error命令用来显示错误信息，同时返回键盘控制。其用法和warning命令非常相似，这里就不详细介绍了。**

## 5.4 应用举例



**例16** 一个三位数，如果它的每个位上的数字的3次幂之和等于它本身，则称此数为水仙花数，例如，153就是一个水仙花数。编写脚本文件，求所有的水仙花数。

**方法1:**

```
clear,clc
a=[];
for k=100:999
    k1=fix(k/100);
    k2=rem(fix(k/10),10);
    k3=rem(k,10);
    if k==k1^3+k2^3+k3^3
        a=[a,k];
    end
end
disp(a)
```

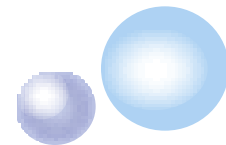
**方法2:**

```
clear,clc
a=[];
for i=1:9
    for j=0:9
        for k=0:9;
            if i*100+j*10+k==i^3+j^3+k^3
                a=[a, i*100+j*10+k];
            end
        end
    end
end
disp(a)
```

**运行结果为:**

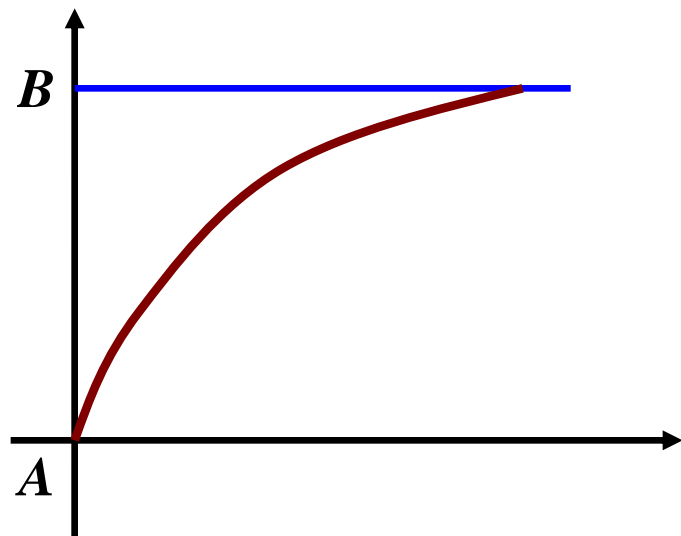
*153 370 371 407*

## 例17 导弹打击过程仿真



设 $A(0,0)$ 为一导弹发射点，发现位于 $B(0,100)$ 处一架敌机沿水平方向逃离，随即发射一枚导弹予以打击，现已知导弹时刻对准敌机，且速率为飞机速率的两倍（设飞机速度为1）。

试编程模拟导弹打击敌机的动态过程，并实时给出飞机和导弹的位置坐标。若要在敌机飞行距离不超过60时（我方空域）将其击落，导弹的速率应提高到多少？



这是一个典型的追击问题,可以通过建立微分方程求得导弹运行曲线（此法将在实验11介绍）,也可以通过计算机仿真来模拟追击过程. 所谓计算机仿真就是利用计算机对实际动态系统的结构和行为进行编程模拟和计算,以此来预测系统的行为效果。

**首先**，将整个追击过程离散化，以 $dt$ 为时间间隔，每经过时间 $dt$ ，导弹改变一次方向对准敌机；

**其次**，根据条件和追击规律，计算出每隔 $dt$ 时间，飞机和导弹的位置坐标，以及它们之间的距离，如此持续直到二者之间的距离足够小为止；

**最后**，利用plot指令绘制导弹和敌机位置变化曲线。

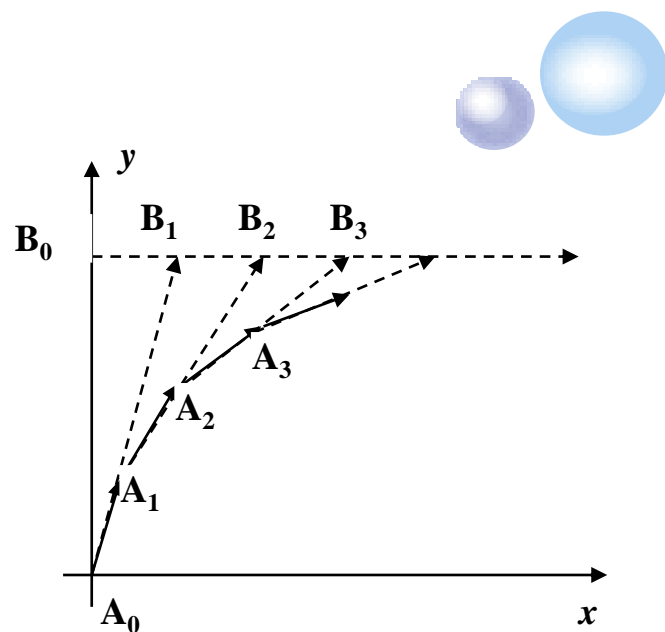


图5.3 导弹追击敌机示意图

建立如图所示坐标系. 设敌机的速率为 $v$ ，则导弹的速率为 $2v$

$$\overrightarrow{OB}_{k+1} = \overrightarrow{OB}_k + \overrightarrow{B_k B_{k+1}} = \overrightarrow{OB}_k + vdt \vec{i} \quad k = 0, 1, 2, 3, \dots$$

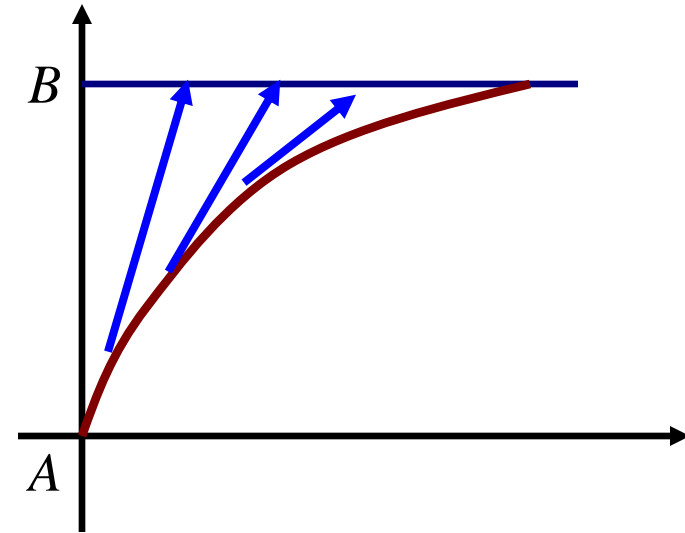
$$\overrightarrow{OA}_{k+1} = \overrightarrow{OA}_k + \overrightarrow{A_k A_{k+1}} = \overrightarrow{OA}_k + 2vdt \frac{\overrightarrow{A_k B_{k+1}}}{\left\| \overrightarrow{A_k B_{k+1}} \right\|} \quad k = 0, 1, 2, 3, \dots$$

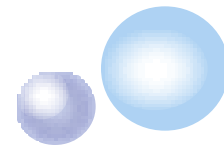
$\vec{i}$ 是与 $x$ 轴方向一致的单位向量



## MATLAB程序:

```
k=0;
A=[0,0];           %导弹初始位置
B=[0,100];         %飞机初始位置
v=1;
dt=1;              %离散时间改变量
d=100;             %相距距离
while d>0.5
    plot(A(1),A(2),'r'); %画导弹位置
    hold on
    plot(B(1),B(2),'b*'); %画飞机位置
    pause(0.2);
    k=k+1;
    B=B+[v*dt,0];       %飞机移动位置
    e=B-A;              %导弹指向飞机向量
    d=norm(e); e0=e/d;  %取向量方向（单位化）
    A=A+2.0*v*dt*e0;    %导弹追击位置
    fprintf('k=%.0f A(%.2f,%.2f) B(%.2f,100) d=%.2f\n',k,A(1),A(2),B(1),d);
end
```





## 例18 用二分法求函数 $x^2-2=0$ 的正实根.

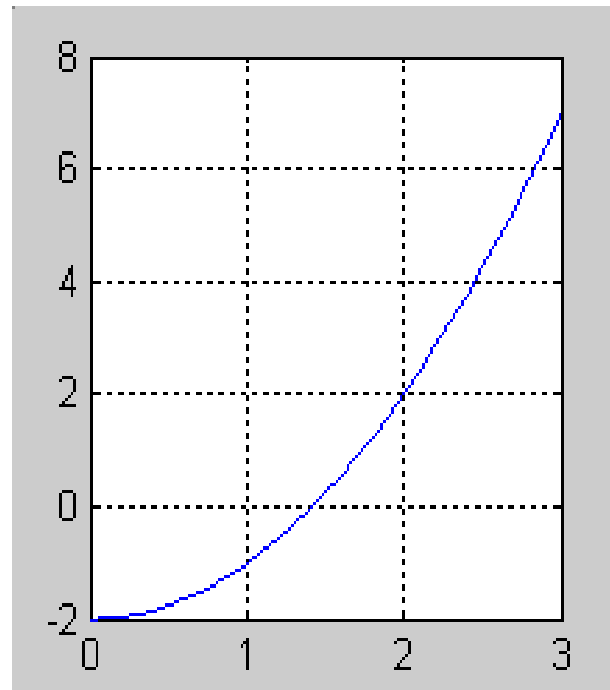
$$f(x) = x^2 - 2, [a, b] = [1, 2], f(a)f(b) < 0$$

$$1) c = (a + b) / 2 :$$

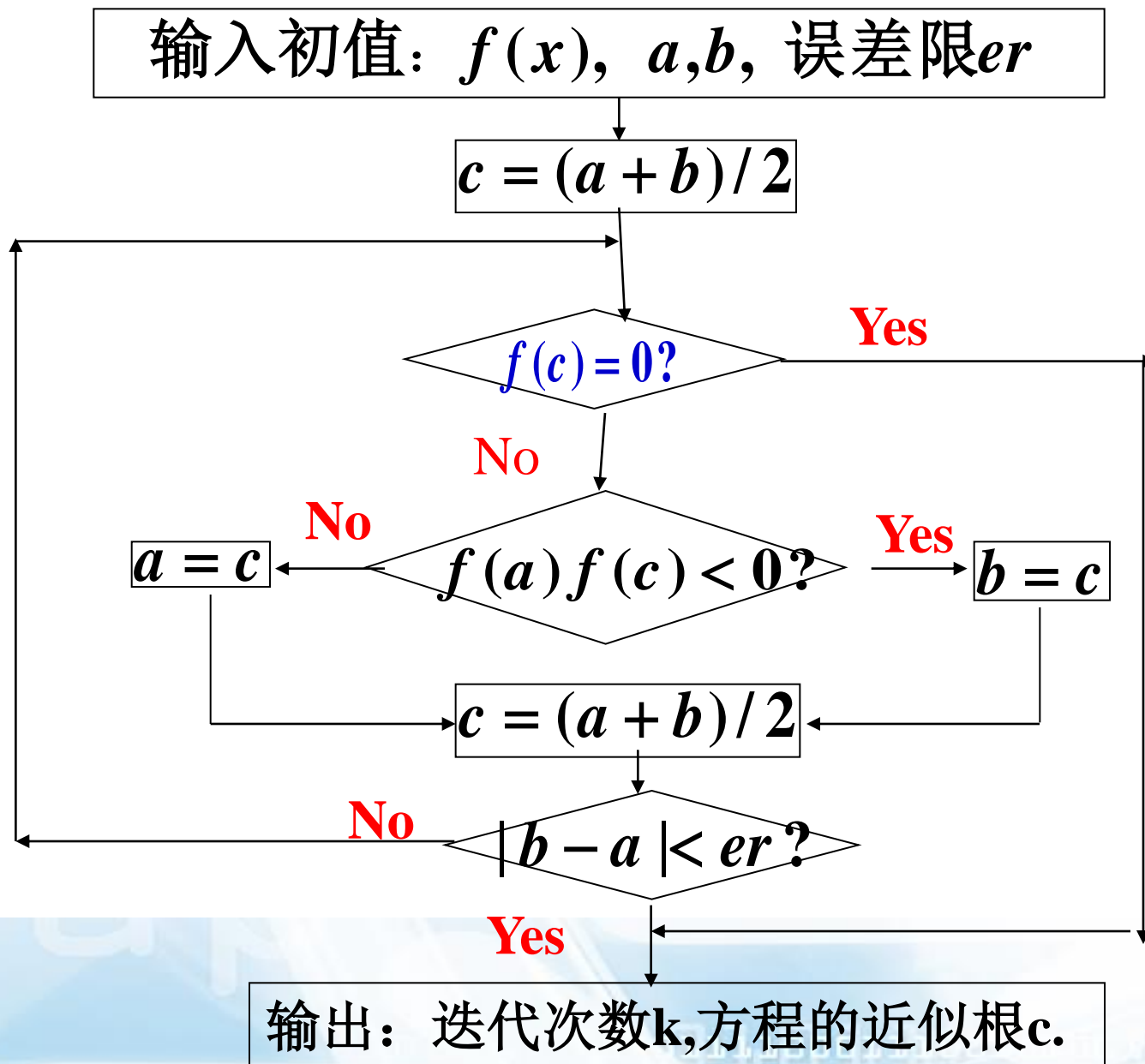
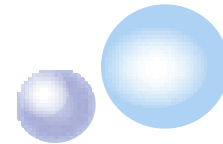
$$\text{if } f(c) = 0 (\text{或 } |f(c)| < r), \quad g = c;$$

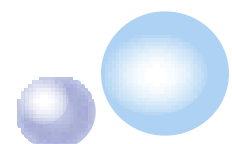
$$\text{elseif } f(c)f(a) < 0 \\ \quad b = c;$$

$$\text{else } a = c;$$



2) if  $|b - a| < \epsilon$ , stop, 输出根。 否则 goto 1).





## 二分法求根程序:

```
a=1;b=2;  
er=1.0e-8;  
f= @(x) x.^2-2;  
k=1;  
c=(a+b)/2;  
while abs(b-a)>er  
  if abs(f(c))==0  
    fprintf('k=%.0f,r=%.8f\n',k,r);  
    break;  
  end
```

```
    if f(a)*f(c)<0  
        b=c;  
    else  
        a=c;  
    end  
    c=(a+b)/2;  
    fprintf('k=%.0f,c=%.8f\n',  
        k,c);  
    k=k+1;  
end
```

### 运行结果

.....

k=24,c=1.41421357

k=25,c=1.41421355

k=26,c=1.41421356

**例** 计算  $I = \lim_{n \rightarrow \infty} \left( \frac{1}{n\sqrt{n+1}} + \frac{\sqrt{2}}{n\sqrt{n+\frac{1}{2}}} + \cdots + \frac{\sqrt{n}}{n\sqrt{n+\frac{1}{n}}} \right)$

**解:** 记  $S_n = \frac{1}{n\sqrt{n+1}} + \frac{\sqrt{2}}{n\sqrt{n+\frac{1}{2}}} + \cdots + \frac{\sqrt{n}}{n\sqrt{n+\frac{1}{n}}} = \sum_{k=1}^n \frac{\sqrt{k}}{n\sqrt{n+\frac{1}{k}}}$

$$\sum_{k=1}^n \frac{1}{n} \cdot \frac{\sqrt{k}}{\sqrt{n+1}} \leq S_n \leq \sum_{k=1}^n \frac{1}{n} \cdot \frac{\sqrt{k}}{\sqrt{n+\frac{1}{n}}}$$

$$\lim_{n \rightarrow \infty} \text{左} = \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{\sqrt{n}}{\sqrt{n+1}} \cdot \sqrt{\frac{k}{n}} \cdot \frac{1}{n} = \lim_{n \rightarrow \infty} \sqrt{\frac{n}{n+1}} \cdot \lim_{n \rightarrow \infty} \sum_{k=1}^n \sqrt{\frac{k}{n}} \cdot \frac{1}{n} = 1 \cdot \int_0^1 \sqrt{x} dx = \frac{2}{3}$$

$$\lim_{n \rightarrow \infty} \text{右} = \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{n}{\sqrt{n^2+1}} \cdot \sqrt{\frac{k}{n}} \cdot \frac{1}{n} = \lim_{n \rightarrow \infty} \frac{n}{\sqrt{n^2+1}} \cdot \lim_{n \rightarrow \infty} \sum_{k=1}^n \sqrt{\frac{k}{n}} \cdot \frac{1}{n} = 1 \cdot \int_0^1 \sqrt{x} dx = \frac{2}{3}$$

$$\therefore I = \frac{2}{3}$$