MATLAB 讲义

2022 年 7 月 9 日

MATLAB 是美国 MathWorks 公司出品的商业数学软件，用于数据分析、无线通信、深度学习、图像处理与计算机视觉、信号处理、量化金融与风险管理、机器人，控制系统等领域。

MATLAB 语言是由美国的 Clever Moler 博士于 1980 年开发的。设计者的初衷是为解决“线性代数”课程的矩阵运算问题。经几年的校际流传，在 Little 的推动下，由 Little、Moler、Steve Bangert 合作，于 1984 年成立了 MathWorks 公司，并把 MATLAB 正式推向市场。从这时起，MATLAB 的内核采用 C 语言编写，而且除原有的数值计算能力外，还新增了数据图视功能。

MATLAB 是 matrix&laboratory 两个词的组合，意为矩阵工厂（矩阵实验室），软件主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案，并在很大程度上摆脱了传统非交互式程序设计语言的编辑模式。

MATLAB 实用性广，操作相对容易，在本科毕设和数学建模中有重要作用，学习必备的 MATLAB 操作是十分必要的。

MATLAB 基本操作介绍

命令行窗口：是 MATLAB 的主要交互窗口，用于输入命令并显示除图形以外的所有执行结果。MATLAB 命令窗口中的“>>”为命令提示符，表示 MATLAB 正在处于准备状态。在命令提示符后键入命令并按下回车键后，Matlab 就会解释执行所输入的命令，并在命令后面给出计算结果。 如果希望结果不被显示，则只要在语句之后加上一个分号（；）即可。此时尽管结果没有显示，但它依然被赋值并在 Matlab 工作空间中分配了内存。

例：输入代码 a=1;b=2;a+b

运行结果为：ans = 3

工作空间：是 Matlab 用于暂时存储各种变量和结果的内存空间。在该窗口中显示工作空间中所有变量的名称、大小、字节数和变量类型说明，可对变量进行观察、编辑、保存和删除。

变量：变量的命名：变量的名字必须以字母开头（不能超过 19 个字符），之后可以是任意字母、数字或下划线；变量名称区分字母的大小写；变量中不能包含有标点符号。

例：a=1;b=2;a+b;若输入 A+B 则会提示未命名的变量。

clc：清屏（清空命令行窗口）。

clcear all： 清除工作区的所有变量。

预设：预设里可以把字体调整大小等。

布局：可以重新设置命令行窗口、编辑器和工作区的分布。

设置路径：可更改路径、添加路径（为了直接调用函数）。

编辑器：可以新建文件，或者打开一些文件。比如新建脚本，也可以新建函数等。点击“New”，建立扩展名为“.m”的 Matlab 文件，把程序写入这个文件中，保存（或快捷键 control+s 键），然后在命令窗口键入文件名即可运行程序。

%的作用：在程序文件中，对相应的命令行做注释，MATLAB 不执行“%”所在行后面的任何文字。（写给自己看，写给别人看）

逗号和分号的作用：在运算式中，MATLAB 允许多条语句在同一行出现，通常不需要考虑空格；多条命令可以放在一行中，它们之间需要用分号隔开；逗号告诉 MATLAB 显示结果，而分号则禁止结果显示。

MATLAB 最基本的计算器功能:

1）直接在命令窗口写入程序语句，如，键入“a=1+2”回车后，在命令窗口显示“a=3”. 这种方法适合于程序语句非常少的计算和求值；常用的运算符号：+，—，*， .* ，/ , \ ，^（幂）

注意：MATLAB 有左除和右除的区别，斜杠上边朝哪边靠，哪边就是求逆，然后两数（或者矩阵）做乘法运算。

例：代码：a=1;b=2;a/b

运行结果：ans = 0.5000

代码：a\b

运行结果：ans = 2

Matlab 给出的内部函数全都是点运算，例如 sin，cos，log，exp 等等.

例：a=[1 2 3 4]; 则函数对 a 向量的运算为 sin(a)=[sin1　sin2　sin3

sin4]，exp(a)=[exp(1), exp(2), exp(3), exp(4)]。

常用函数命令：

sin-正弦；cos-余弦；tan-正切；cot-余切；

asin-反正弦；acos-反余弦；atan-反正切；acot-反余切；

exp(x)-指数函数 ex；log(x)-对数函数 lnx；

sum-向量元素求和；若 A 是矩阵，则 sum(A)为包含行总和的列向量；

log10(x)-对数函数 log10(x);

abs-取绝对值；

sqrt-开方；

mod(x,y)-返回 x/y 的余数；

imag-求虚部；real-求实部；

floor(x)；对 x 向下取整。

例：imag(2+6i)；

asin(sqrt(2)/2)；

sin(pi/2);exp(1)

sym(asin(sqrt(2)/2))%精确值，输出结果为 ans =pi/4

常用常数的 MATLAB 内部变量：

pi-圆周率 3.14159…; i、j- 单位虚数；inf-无穷大；

eps-浮点误差，a^b；

NaN-空值。

数据的显示格式由 format 命令控制：

format 只是影响结果的显示，不影响其计算与存储；MATLAB 总是

以双字长浮点数（双精度）来执行所有的运算。（双精度浮点数(double)是计算机使用的一种数据类型，使用 64 位（8 字节） 来存储一个浮点数。 它可以表示十进制的 15 或 16 位有效数字，其可以表示的数字的绝对值范围大约是：-1.79E+308 ~ +1.79E+308 ）。

如果结果为整数，则显示没有小数；若结果不是整数，则输出形式有：

浮点算术运算

format long ——(定义输出格式)

例：1/2+1/3

ans =0.83333333333333

若用符号运算：sym(1/2+1/3)

ans = 5/6 ——精确解

%符号计算的一个显著特点是，由于计算中不会出现舍入误差，从而可以得到任意精度的数值解（要计算精确，就要牺牲计算时间和储存空间）。

MATLAB 数据类型：数字、字符和字符串.（字符串简单理解，就是我说的这些话，不需要计算机理解，只要原本输出即可）；

例：代码：str='I love Matlab'

length(str) %字符串的长度，注意空格也占了一个字符

运行结果：str = 'I love Matlab'

ans = 13

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MATLAB 中关于矩阵的运算

矩阵的输入规则：

（1）矩阵元素必须用[　　]括住；

（2）矩阵元素必须用逗号或空格分隔；

（3）在[　　]内矩阵的行与行之间必须用分号分隔。

例：A=[1 2 3;4 5 6];

冒号的作用:

(1)用于生成等间隔的向量，默认间隔为 1；

例：代码：b=1:2:9

运行结果：

b =　1　　3　　5　　7　　9

代码：a=1:10

运行结果：

a =1　　2　　3　　4　　5　　6　　7　　8　　9　　10

(2)用于选出矩阵指定行、列及元素；%可讲矩阵的时候再介绍

例：代码：A=[1 2 3 ;4 5 6;7 8 9]

运行结果：A =

　　1　　2　　3

　　4　　5　　6

　　7　　8　　9

代码：a12=A(1,2)　%把 A 的第一行第二列元素赋给 a12；

运行结果：a12 = 2

代码：A1=A(1,:)　　%把 A 的第一行赋给 A1

运行结果：

A1 =

　　1　　2　　3

代码：B1=A(:,1)　　%把 A 的第一列赋给 B1

运行结果：

B1 =

　　1

　　4

　　7

(3)循环语句（后续再介绍）。

矩阵的运算：

A+B; A-B;

A*B：一般意义的矩阵相乘；

A.*B：对应的数字相乘形成的矩阵

（注意“.*”和“*”的区别：“*”为矩阵乘法,两个矩阵必须满足左边矩阵的列数等于右边矩阵的行数，才能做矩阵乘法;“.*”为点乘运算,是指两个矩阵中对应元素进行乘法运算。）

例：代码：A=[1 2 3 ; 4 5 6];B=[1 2 3 ; 4 5 6];%A*B 会报错；

A.*B

运行结果：ans =

　　1　　4　　9

16 25 36

例：代码：A=[1 2;3 4];B=[4 0;0 5];a=[1 2 3 4];b=[2 4 6 8];A.*B

运行结果：ans =

4 0

0 20

代码：a./b

ans = 0.5000 0.5000 0.5000 0.5000

a.^2

ans = 1 4 9 16

例：代码：A=[1 2;3 4];C=A.^2;D=A^2

C =

1 4

9 16

D=

7 10

15 22

A/B 表示 A 乘以 B 的逆矩阵；A\B 表示 A 的逆矩阵乘以 B。

两种除法：\和/，分别表示左除和右除。如果 A 矩阵是非奇异方阵，则 A\B 和 B/A 运算可以实现。A\B 等效于 A 的逆左乘 B 矩阵，而 A/B 等效于 B 矩阵的逆右乘 A 矩阵。一般 A\B≠B/A。

例：代码：A=[1 2;3 4];B=[5 6;7 8];

A/B%可以用这个命令变为整数 floor(A/B)

A\B

运行结果：

ans =

3.0000 -2.0000

2.0000 -1.0000

ans =

-3 -4

4 5

可验证(**A/B 表示 A 左乘 B 的逆矩阵；A\B 表示 A 的逆矩阵左乘 B)**

代码：A*inv(B)%A 与 B 的逆矩阵相乘

运行结果：ans =

3.0000 -2.0000

2.0000 -1.0000

代码：inv(A)*B

运行结果：ans =

-3.0000 -4.0000

4.0000 5.0000

矩阵的转置、求逆、求特征值等：

B=A’;

inv(A)（如果不是方阵，会提醒你错误使用）；

rank(A)%求矩阵的秩；

det(A)%求矩阵的行列式；

zeros(m,n)m 行 n 列的 0 矩阵；

ones(m,n)m 行 n 列的 1 矩阵；

eye(n)单位阵；

sum(A):包含行总和的列向量；

rand(m,n)-创建 m 行 n 列的随机矩阵，矩阵元素为 0 到 1 的随机生成的数;

例 代码：eye(2，3)

ans=

1 0 0

0 1 0

代码：zeros(2，3)

ans=

0 0 0

0 0 0

例：rand(3,4) %生成三行四列，矩阵元素为 0 到 1 的随机矩阵；

diag（p）---创建对角矩阵，对角矩阵的对角线元素为向量 p；

例：代码：V=[5 7 2]；A=diag(V)

运行结果：A=

5 0 0

0 7 0

0 0 2

例：diag([1 2 3 4])

注意：diag(1,2,3,4)这个命令的语法是错误的，因为括号里面不是向量。

linspace(a, b, n)—创建行向量，其中 a 和 b 是生成向量的第一个和最后一个元素，n 是元素总数。缺失 n，默认为 100。

例： C=linspace(1,9,5)%从 1 到 9 均匀生成 5 个数字

运行结果：ans = 1　　3　　5　　7　　9

矩阵元素修改可以用 A(*,*)= *来修改

例：a=[1 2 0;3 0 5;7 8 9]

a =

1　　2　　0

3　　0　　5

7　　8　　9

>> a(3,3)=0%把矩阵 a 的第三行第三列数字改为 0

a =

1　　2　　0

3　　0　　5

7　　8　　0

例：B=1:2:9 %从 1 到 9，步长为 2 的行向量（默认步长是 1）

C=**repmat**(B,3,1)%重复三行一列

运行结果：

C =　1　　3　　5　　7　　9

1　　3　　5　　7　　9

1　　3　　5　　7　　9

A=**magic**(5)%幻方 magic(n)

B=**A(2,3)**%取出第二行第三列

C=**A(3,:)**%　　取出第三行

D=**A(:,4)**%　　取出第四列

[m,n]=**find**(A>20)%寻找大于 20 的序号值/矩阵

[m,n]=eig(A)%求矩阵的特征值和特征向量（每一列是特征向量）

例：代码：A=[1 2 3;4 5 6 ;7 8 9];[m,n]=eig(A)

运行结果：

m = -0.2320 -0.7858 0.4082

-0.5253 -0.0868 -0.8165

-0.8187 0.6123 0.4082

n = 16.1168 0 0

0 -1.1168 0

0 0 -0.0000

多项式运算

matlab 语言把多项式表达成一个行向量，该向量中的元素是按多项式降幂排列的。设多项式 $p(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，则表示为 $p=[a_n,a_{n-1},\cdots,a_1,a_0]$

求多项式的根 roots(p) %求出 p(x)=0 的解。

$p(x)=3x^4+5x^3+x+6$,求 p(x)=0 的根.

p=[3,5,0,1,6];

roots(p)

求多项式在点 x0 处的值 polyval(p, x0) %即计算 p(x0)

polyval(p,3)

poly(A)—若 A 为矩阵，求矩阵 A 的特征多项式；

poly(p)—若 p 为向量，求以 p 各个元素为根的多项式，这个命令与 roots

正好是相反的命令。

说明：特征多项式一定是 n+1 维的；特征多项式第一个元素一定是 1；

例：a=[1 2 3;4 5 6;7 8 0];p=poly(a)%求矩阵 A 的特征多项式

运行结果：p = 1.0000 -6.0000 -72.0000 -27.0000

例：b=[1 2 3]; poly(b)%求以 b 各个元素为根的多项式

运行结果：ans =1 -6 11 -6

例：p=[1.0000 -6.0000 -72.0000 -27.0000];

p1=poly2str(p,'x') % 显示数学多项式的形式,变量为 x

运行结果：p1 =x^3 - 6 x^2 - 72 x – 27

%%

MATLAB 中关于极限、微积分、方程求解等运算

符号计算

科学与工程技术中的数值运算固然重要，但自然科学理论分析中各种各样的公式、关系式及其推导就是符号运算要解决的问题。

符号运算的功能

符号表达式、符号矩阵的创建；符号线性代数；因式分解、展开和简化；符号代数方程求解；符号微积分；符号微分方程

什么是符号运算

符号运算与数值运算的区别

数值运算中必须先对变量赋值，然后才能参与运算。符号运算无须事先对独立变量赋值，运算结果以标准的符号形式表达。

' '字符串标识，字符串表达式一定要用' '单引号括起来 Matlab

才能识别；‘ ’里的内容可以是函数表达式，也可以是方程。函数表达式或方程可以赋给字符串或符号变量，以后方便调用。

符号变量是**内容可变的符号对象**。符号变量通常是指一个或几个特定的字符，不是指符号表达式，甚至可以将一个符号表达式赋值给一个符号变量。符号变量有时也称自由变量，它的命名规则和数值变量的命名规则相同。

符号函数和符号方程，判断方法是看带不带等号。

注：在符号对象的比较中，没有”大于”、 ”大于等于”、 ”小于”、 ”小于等于”的概念，而只有是否“等于”的概念。

相关指令为： sym( ) 和 syms( )（symbolic 的缩写）。利用符号命令进行计算时，需要提前用符号命令 **syms** 来定义要使用的变量。

expand-每个因式的乘积进行展开计算

例：看下列式子的展开：

```
>>syms x y a b c t
E1 = expand((x-2)*(x-4)*(y-t))
E2 = expand(cos(x+y))
E3 = expand(exp((a+b)^3))
E4 = expand(log(a*b/sqrt(c)))
E5 = expand([sin(2*t), cos(2*t)])
```

计算结果为：

```
E1 =x^2*y-x^2*t-6*x*y+6*x*t+8*y-8*t
E2 =cos(x)*cos(y)-sin(x)*sin(y)
```

E3 =exp(a^3)*exp(a^2*b)^3*exp(a*b^2)^3*exp(b^3)

E4 =log(a*b/c^(1/2))

E5 =[2*sin(t)*cos(t), 2*cos(t)^2-1]

factor-因式分解或者质因数分解

例：代码：syms a b x y

F1 = factor(x^4-y^4)

F2 = factor(sym('123456789012345678 90'))

其结果为：

F1 =[x - y, x + y, x^2 + y^2]

F2 =[2, 3, 3, 5, 101, 3541, 3607, 3803, 27961]

求根的方法

例：p=[1 -3 0 1 2];x=roots(p)

使用 solve 函数求根

s=solve(f,v)：求方程关于指定自变量的解；

s=solve(f)：求方程关于默认自变量的解。

其中 f 可以是用字符串表示的方程，或符号表达式；若 f 中不含等号，则表示解方程 f=0。

例：解方程 2x^2-8=0.

代码： syms x

[x]=solve(2*x.^2-8==0)

例：解方程 sin x=cos x。

代码：syms x; solve(sin(x)==cos(x))

运行代码：ans =pi/4

例：线性方程组求解：

x+y =1,

x-11*y=5

解法一：syms x y

[x,y]=solve(x+y ==1,x-11*y==5)

运行结果：x =4/3　　y =-1/3

解法二：A=[1 1; 1 -11]; b=[1;5];

x=sym(linsolve(A,b))%sym 是精确解

运行结果：

x =4/3

-1/3

解法三：A=[1 1; 1 -11];b=[1;5];

x=inv(A)*b%x=sym(inv(A)*b)则是精确解

运行结果：

x =4/3

-1/3

linsolve(A,b)：解线性方程组

例：　A=[1 2 -1; 1 0 1; 1 3 0]; b=[2;3;8]; X=linsolve(A,b)

%注意这里的 b 是列向量。

运行结果：

X = -0.2500

2.7500

3.2500

limit-求函数的极限

limit(F,x,a)为求变量 x 趋向于 a 时候，函数 F 的极限；

limit(F,x,a,'right') or limit(F,x,a,'left')，求相应的 a 的右极限或者 a 的左极限；

命令缺变量 x，则求默认变量的极限；缺 a，则求变量趋向于 0 的极限；

例：syms x a t h n;

L1 = limit((cos(x)-1)/x)　　%求默认变量 x 趋向于 0 的极限

L2 = limit(1/x^2,x,0,'right')

L3 = limit(1/x,x,0,'left')

L4 = limit((log(x+h)-log(x))/h,h,0)

结果显示：

L1 =0

L2 =Inf

L3 =-Inf

L4 =1/x

另外函数也可以为向量的形式或者数列的形式；

例：v = [(1+a/x)^x, exp(-x)];

L5 = limit(v,x,inf,'left')

运行结果：L5 =[exp(a), 0]

例　求 $\lim\limits_{x\to 0}\dfrac{(1+x)^{\frac{1}{x}}-e}{x}$.

syms x;f=((1+x).^(1/x)-exp(sym(1)))./x;　%注意 e 的表示 exp(1)表示 e 的数值，exp(sym(1))表示精确的 e

limit(f,x,0)

运行结果：

ans =-exp(1)/2

例： 求 $\lim\limits_{x\to+\infty}\dfrac{e^x}{x^3}$.

代码：syms x; f=exp(sym(1)).^x./(x^3); limit(f,x,inf)

运行结果：ans =Inf

例： 求 $\lim\limits_{x\to-\infty}\dfrac{e^x}{x^3}$.

代码：syms x;f=exp(sym(1)).^x./(x^3);limit(f,x,-inf)

运行结果：ans = 0

diff-求微分/导数

用法：diff(F,v,n)-求函数关于给定变量 v 的 n 阶导数。变量 v 缺失，则求默认的变量，另外变量 v 和 n 可以互换位置。

例：代码：syms x y t

D1=diff(sin(x^2)*y^2,1)　%计算 $\dfrac{\partial}{\partial x}(y^2\sin x^2)$

D1= diff(sin(x^2)*y^2,2)　%计算 $\dfrac{\partial^2}{\partial x^2}y^2\sin x^2$

D2= diff(sin(x^2)*y^2,y)

subs(D1,x,1)；%在 x=1 处的导数；

subs(D1,[x y],[1 2])%在 x=1 和 y=2 处的导数.

int-求不定积分

用法：R = int(S, v)-对符号表达式 S 中指定的符号变量 v 计算不定积分。注意的是，表达式 R 只是函数 S 的一个原函数，后面没有带任意常数 C。

例>>syms x z t alpha

INT1 = int(x/(1+z^2),z)

INT3 = int([exp(t),exp(alpha*t)])

运行结果：

INT1 =x*atan(z)

INT3 = [exp(t), exp(alpha*t)/alpha]

例：syms x y;

int1=int(sin(x))

int2=int(1./(1+x.^2))

运行结果：

int1 =-cos(x)

int2 =atan(x)

例：代码：syms x;int3 = int(cos(3*x).*sin(x))

运行结果：

int3 =(3*cos(x)^2)/2 - cos(x)^4

int-求定积分%此命令针对能求出不定积分的函数

用法：R = int(S,v,a,b)　　%对表达式 s 中指定的符号变量 v 计算从 a

到 b 的定积分

例：syms x z t alpha;

INT4 = int(x*log(1+x),0,1)

INT5 = int(2*x, sin(t), 1)

运行结果：

INT4 =1/4

INT5 =cos(t)^2

quad-求定积分%此命令针对不能求出不定积分的函数

用法：quad(F,a,b)-F 为被积表达式，b，a 分别为上下限；注意，此积分是数值积分，F 的表达式应该写成点运算的形式，并用引号括起来。

例 计算$\int_0^1 e^{-x^2}dx$

代码：resultInt1=quad('exp(-x.^2)',0,1)%如果用 int 会报错

运行结果：ans =0.7468

常微分方程符号求解

这一部分针对的是常微分方程在实数范围内有解，并且能够解出来的。常微分方程符号求解介绍：

dsolve('eq1','eq2,…','cond1','cond2,…','v')---对给定的常微分方程（组）eq1,eq2,…中指定的符号自变量 v，与给定的边界条件和初始条件 cond1,cond2,….求符号解（即解析解）；若没有指定变量 v，则缺省变量为 t；一阶导 D=d/dx，二阶导 D2=d2/dx2，…。另，无论是方程还是初始条件，均需要用"括起来（单引号）。

例：代码：

DD1 = dsolve('D2y-Dy=exp(x)','x')%

DD2 = dsolve('t*D2f = Df')

运行结果：

DD1 =C2*exp(x) + exp(x)*(x + C1*exp(-x))

DD2 = C4*t^2 + C3

代码：

DD3 = dsolve('(Dy)^2 + y^2 = 1','s')

运行结果：

DD3 = 1

-1

(exp(C9*1i + s*1i)*(exp(- C9*2i - s*2i) + 1))/2

(exp(C6*1i - s*1i)*(exp(- C6*2i + s*2i) + 1))/2

代码：

DD4 = dsolve('Dy = a*y', 'y(0) = b') % 带一个定解条件

DD5 = dsolve('D2y = -a^2*y', 'y(0) = 1', 'Dy(pi/a) = 0') % 带两个定解条件

运行结果：

DD4 =b*exp(a*t)

DD5 =exp(-a*t*1i)/2 + exp(a*t*1i)/2

例：求解线性微分方程组：

代码：

[x,y] = dsolve('Dx = y', 'Dy = -x')

运行结果：

x =C15*cos(t) + C14*sin(t)

y =C14*cos(t) - C15*sin(t)

代码：

[u,v] = dsolve('Du=u+v, Dv=u-v')

运行结果：

u =C16*exp(2^(1/2)*t)*(2^(1/2) + 1) - C17*exp(-2^(1/2)*t)*(2^(1/2) - 1)

v =C16*exp(2^(1/2)*t) + C17*exp(-2^(1/2)*t)

例:Logistic 方程求解

syms x t lamada%注，不用定义符号变量也可以，更符合我们的习惯。

DL = dsolve('Dx = lamada *x*(1-x)', 'x(0) = i0')

运行结果

DL = 1/(1-exp(-lamada*t)*(-1+i0)/i0)

微分方程数值求解

这一部分针对的是微分方程在实数范围内没有解，或者很难求出它的解析解，那么数值求解就是通用的办法。

常微分方程数值求解命令介绍：

[TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0)----- 在区间 TSPAN=[T0,TF]数值求解微分方程 $\frac{dy}{dt} = f(t, y), y(T0) = Y0$ 。

（1）其中 ODEFUN 是右端函数 f(t,y)的句柄或者函数对应的名字；因为是数值求解，所以定义函数的时候，变量之间的运算必须是点运算；函数一定是二维的，即包括自变量和因变量，否则出错。

（2）TSPAN 是数值求解的区间，若区间起始点为 T0，终端为 TF，则 TSPAN=[T0,TF]；

（3）初始条件：微分方程的解在 T0 的初始值为 Y0；

（4）输出端：TOUT,YOUT 为数值解，即各个点上对应的横坐标和纵坐标。ode45 求解时，步长通常不是恒定的，而是变化的，如果要求求解的步长是恒定的，可以令 TSPAN = [T0 T1 T2 …TF]; 其中前后两个点的插距是一样的。

例：求解微分方程

$$\begin{cases} x' = -x^2 \\ x(0) = 1 \end{cases}$$

首先，创建函数 xprim1，将此函数保存为 xprim1.m，如下

```
%xprim1.m
function    xprim=xprim1(t,x)
xprim=-x.^2;
end
```

然后，

```
[t,x]=ode45('xprim1',[0,10],1);
 %也可以这样调用[t,x]=ode45(@xprim1,[0,10],1);
%这样使用是错误的：[t,x]=ode45('-x.^2',[0,10],1);
plot(t,x,'-b',t,x,'or');%绘制常微分方程的解的图像
xlabel('time t0=0,tt=1');
ylabel('x value x(0)=1');
title('一阶常微分方程解法举例');
```

如果要求数值求解的点的横坐标等步长，也可以为：（不妨假定步长为 0.1）

```
tt=0:0.1:10;
[t,x]=ode45(@xprim1,tt,1);
```

例在人口动力学模型中，常用到“捕食者—被捕食者模式”，即

$$\begin{cases} \dfrac{dx_1}{dt} = x_1 - 0.1x_1x_2 + 0.01t \\ \dfrac{dx_2}{dt} = -x_2 + 0.02x_1x_2 + 0.04t \\ x_1(0) = 30 \\ x_2(0) = 20 \end{cases}$$

，这里 x_1 表示被捕食者， x_2 表示捕食者。

求解这个方程组并不难。

首先，创建文件 xprim3.m，如下

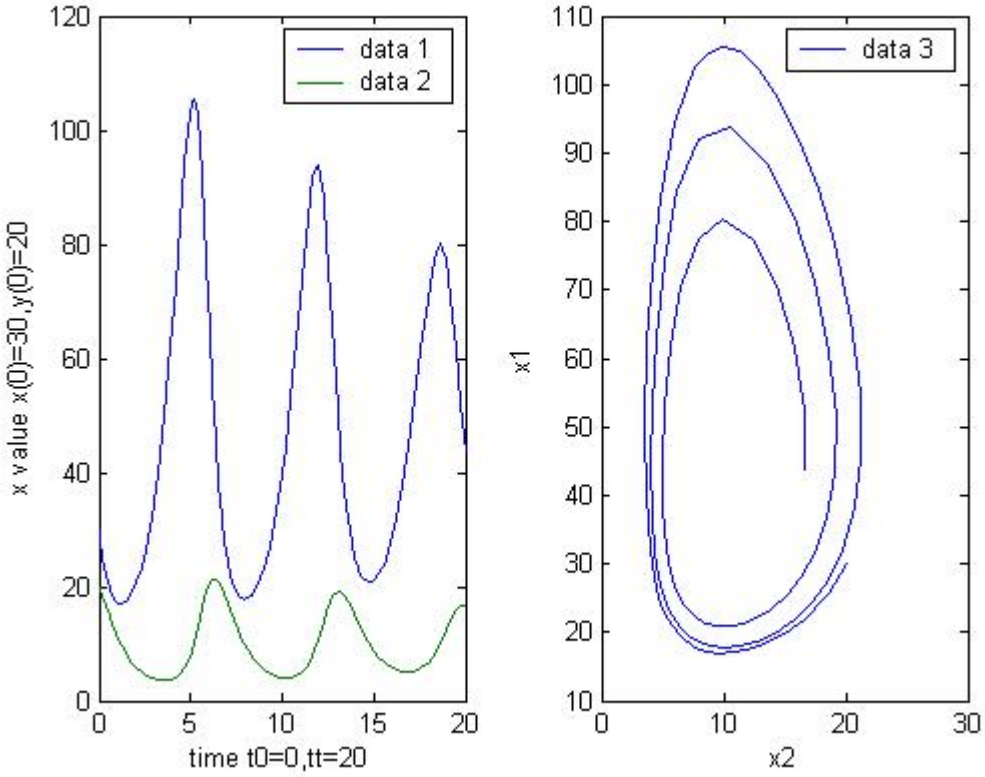
```
function   xprim=xprim3(t,x)
xprim=[x(1)-0.1*x(1).*x(2)+0.01*t;-x(2)+0.02*x(1).*x(2)+0.04*t];
end
```

然后，用 ODE 算法，调用此函数，画出解的图形：

```
[t,x]=ode45('xprim3',[0 20],[30; 20]);
subplot(1,2,1);
plot(t,x);%画出解的图形
xlabel('time t0=0,tt=20');
ylabel('x value x(0)=30,y(0)=20');
subplot(1,2,2);
plot(x(:,2),x(:,1));%根据 x(2)画出 x(1)的平面相位图
xlabel('x2');
```

ylabel('x1');

运行结果



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MATLAB 中关于绘图操作

二维图形绘制　PLOT

%绘制散点图：x=1:10;y=10:19;**scatter**(x,y)

plot(x,y) ——基本格式，以 y(x)的函数关系作出直角坐标图，其中 x,y 均为同维向量；

plot(x1,y1,x2,y2…) ——绘制多条曲线格式；

plot(x,y, ‘s’)—— 此格式用于绘制线形不同点标和颜色的图形，字符串 s 设定**曲线颜色和绘图方式**，s 的介绍：

y	黄色	• 点线
m	粉红	○ 圈线
c	亮蓝	× ×线
r	大红	＋ ＋字线
g	绿色	— 实线
b	蓝色	* 星形线
w	白色	：虚线
k	黑色	—• (--)点划线

例：画正弦函数的图像：代码如下：

```
x=0:0.1:2*pi;
y=sin(x);
figure %建立一个幕布（有时候可以不写）
plot(x,y)
```

代码：

```
x=0:0.1:2*pi;%步长
y=sin(x);plot(x,y);xlim([0 2*pi])%停靠在右侧
```

代码：

```
x=0:0.1:2*pi;
y=sin(x);
figure %建立一个幕布（有时候可以不写）
plot(x,y,'*r')%先 plot，再加各种附注
title('y=sin(x)')%加标题
```

xlabel('x')%加横坐标标注

ylabel('sin(x)')%加纵坐标标注

xlim([0 2*pi])%让图靠近两边

另外：

text —— 在图形指定位置加标注

gtext —— 将标注加到图形任意位置(用鼠标来控制位置)

例：代码：

x=0:0.02:10;

y=sin(x);

plot(x,y,'b');gtext('正弦函数')%鼠标拖动停下的位置刚好写'正弦函数'。

例：x=0:0.02:10;

plot(x,sin(x),'*r')*红色星星线

hold on　%利用此命令可以在一个窗口上绘制多条曲线

一个窗口绘制多条曲线

例 代码：

x=0:0.02:10;

y1=sin(x);

y2=cos(x);

plot(x,y1,'b')

hold on

plot(x,y2,'m')

例：hold on 的用法举例

```
x=0:0.01:20;
y1=200*exp(-0.05*x).*sin(x);%衰减快
y2=0.8*exp(-0.5*x).*sin(10*x);衰减慢
plot(x,y1);
hold on;
plot(x,y2)
```

%这样画出来在同一个坐标系中，有一个函数的纵坐标的函数值由于太小，近乎一条平的线。

plotyy 函数，用于在一个图中绘制多条 2D 图形的函数，而且这个命令是把曲线自动分成两个颜色，左右分别有两个纵坐标。

例. 代码：

```
x=0:0.01:20;
y1=200*exp(-0.05*x).*sin(x);
y2=0.8*exp(-0.5*x).*sin(10*x);
figure
[AX,H1,H2]=plotyy(x,y1,x,y2,'plot');
set(get(AX(1),'Ylabel'),'String','slow decay')
set(get(AX(2),'Ylabel'),'String','fast decay')
xlabel('Time(\musec)')
title('Multiple decay rates')
set(H1,'LineStyle','--','color','r')%设置线的格式
set(H2,'LineStyle',':','color','k')
```

例.参数方程画圆：

```
t=linspace(0,10);%默认把 0 到 10 分成 100 份；
x=5*sin(t);
y=5*cos(t);
plot(x,y)
```

Taylor 展开的命令

例：syms x y z; taylor(exp(-x))%默认的是五阶展开;在原点

运行结果： - x^5/120 + x^4/24 - x^3/6 + x^2/2 - x + 1

例：taylor(sin(x),x,pi/2,'Order',6)%对 x 在 pi/2 处的 6 阶无穷小（五阶）展开

运行结果： (pi/2 - x)^4/24 - (pi/2 - x)^2/2 + 1

例：taylor(sin(x)*cos(y)*exp(x),[x y z],[0 0 0],'Order',4)

运行结果： x - (x*y^2)/2 + x^2 + x^3/3

例： taylor(exp(-x),x,'OrderMode','Relative','Order',8)

运行结果： - x^7/5040 + x^6/720 - x^5/120 + x^4/24 - x^3/6 + ...
x^2/2 - x + 1

例： taylor(log(x),x,'ExpansionPoint',1,'Order',4)

运行结果： returns x - 1 - 1/2*(x - 1)^2 + 1/3*(x - 1)^3

例： syms x; f=sin(x);t1=taylor(f,x,0,'Order',4);

t2=taylor(f,x,0,'Order',6);

t3=taylor(f,x,0,'Order',8);

例：泰勒展开，图像拟合的例子。（sin x 在原点泰勒展开拟合的例子）

```
x=-3:0.1:3;
y=sin(x);
y1=- x.^3/6 + x;
y2= x.^5/120 - x.^3/6 + x;
y3=-x.^7/5040 + x.^5/120 - x.^3/6 + x;
plot(x,y,'*k')%原函数图像黑色
hold on
plot(x,y1,'+r')%三阶泰勒展开用的红色
hold on
plot(x,y2,'m')%五阶泰勒展开用的粉色
hold on
plot(x,y3,'-.b')%七阶泰勒展开用的蓝色
```

多窗口绘图：figure(n) —— 创建窗口函数，n 为窗口顺序号。

例：

```
t=0:pi/100:2*pi;    y=sin(t);
    y1=sin(t+0.25);     y2=sin(t+0.5);
plot(t,y)% —— 自动出现第一个窗口
figure(2)
plot(t,y1)% —— 在第二窗口绘图
figure(3)
plot(t,y2) %——在第三窗口绘图
```

例：泰勒展开

```
x=-3:0.1:3;
y=sin(x);
y1=- x.^3/6 + x;
plot(x,y,'*k')%原函数图像黑色
figure(2)
plot(x,y1,'+r')%三阶泰勒展开用的红色
```

子图分割命令，subplot(m,n,p)：

将一图形窗口分成 mxn 个小窗口，在第 p 个小窗口中创建一坐标轴，则新的坐标轴成为当前坐标轴。

subplot(1,2,1)%把窗口分为 1 行 2 列

例：

```
t=0:pi/100:2*pi;        y=sin(t);
y1=sin(t+0.25);y2=sin(t+0.5);y3=cos(t);
subplot(1,3,1); plot(t,y);xlabel('x')
subplot(1,3,2); plot(t,y3)
subplot(1,3,3); plot(t,y2)
```

例：既用到了 subplot，又用到了 figure 的代码：

```
x=0:0.1:5;
y=sin(x);
plot(x,y)%第一个窗口画图，分成一行两列
subplot(1,2,1)
```

plot(x,cos(x))

subplot(1,2,2)

plot(x,sin(x),'r')

figure(2)%第二个窗口绘图

plot(x,cos(x)+1,'r')

hold on%第二个窗口绘两个图，hold on 保持当前窗口

plot(x,sin(x))

%%

三维绘图

plot3 —— 基本的三维图形指令（画出来的是三维曲线）

plot3(X,Y,Z) —— X,Y,Z 是长度相同的向量，绘制一条分别以向量 X,Y,Z 为 x,y,z 轴坐标值的空间曲线。

plot3(X,Y,Z) —— X,Y,Z 均是 mxn 的矩阵，绘制 m 条曲线，第 i 条曲线分别以 X,Y,Z 矩阵的第 i 列分量为 x,y,z 轴坐标值的空间曲线。

二维图形的所有基本特性对三维图形全都适用。

grid on(off) 绘制三维网格

text(x,y,z,‘string’) 三维图形标注

子图和多窗口也可以用到三维图形中

例：画三维空间的螺旋线：

t=0:pi/50:10*pi;

plot3(sin(t),cos(t),t)

xlabel('sin(t)')

ylabel('cos(t)')

zlabel('t')

%%%%%%%%%%%%%%%%%%%%%

三维网格图

mesh —— 三维网格绘图函数，不同于 plot3 的是可以绘制出在某一区间内完整的曲面，而不是单根曲线。

调用格式：

mesh(Z) —— Z 为 m×n 的矩阵，将(i,j)作为 Z(i,j)的 X,Y 轴坐标值。

mesh(X,Y,Z) —— X,Y 分别为三维空间的坐标位置，必须均为向量，若 X 和 Y 的长度分别为 m 和 n，则 Z 必须为 mxn 的矩阵。

三维网格作图参考

[X,Y]=meshgrid(x,y) —— x 和 y 为给定的向量，是用来定义网格划分区域的，也可定义网格划分方法；X 和 Y 用来存储网格划分后的数据矩阵。

%meshgrid(x)是生成绘制 3D 图形所需的网格数据。在进行绘图操作时， 往往需要一些采样点，然后根据这些采样点来绘制出整个图形。在进行 3D 绘图操作时，涉及到 x、y、z 三组数据，而 x、y 这两组数据可以看做是在 Oxy 平面内对坐标进行采样得到的坐标对(x, y)。

此命令的作用是将给定的区域按一定的方式划分成平面网格。

meshgrid—— 网线坐标值计算函数

z=f(x,y) — 根据 x,y 坐标找出 z 的高度

例：绘制 z=x^2+y^2 的三维网线图形（画椭圆抛物面的上半部分）

代码：x=-5:5; y=x;

[X,Y]=meshgrid(x,y);

Z=X.^2+Y.^2;

plot3(X,Y,Z)%三维曲线

figure(2)

mesh(X,Y,Z)%mesh 是由一系列二维线条表示，是网格状的图形

figure(3)

surf(X,Y,Z);%是一片片曲面，与前面两个作为对比

例：画椭圆抛物面的上半部分（另一种方法）

x=-8:0.5:8;

[xx,yy]=meshgrid(x);%[X,Y]=meshgrid(x) 与 [X,Y]=meshgrid(x,x) 是 等同的

z=xx.^2+yy.^2;　%椭圆抛物面的上半部分

surf(xx,yy,z);　%三维曲面图形绘制，surf 生成的是表面图形，由一系列面片拼接生成的

例：画马鞍面(双曲抛物面)

x=-10:10; y=x;

[X,Y]=meshgrid(x,y);

Z=X.^2-Y.^2;

surf(X,Y,Z)

xlabel('自变量 x')

ylabel('自变量 y')

zlabel('函数 z')

title('马鞍面')

例：x=-10:10; y=x;

    [X,Y]=meshgrid(x,y);

    Z=X.*Y;

surf(X,Y,Z)

%surfc(X,Y,Z) 创建一个三维曲面图，其下方有等高线图。曲面图是一个具有实色边和实色面的三维曲面。该函数将矩阵 Z 中的值绘制为由 X 和 Y 定义的 x-y 平面中的网格上方的高度。曲面的颜色根据 Z 指定的高度而变化。

grid on%加了网格线

axis square%变成正方形更好看

例：画球（用参数表达式画的，半径为 1，两个参数 u 和 v）

clear;

u=-4:0.2:4;%步长可以调整为 0.5 或者 1 看看效果

v=-4:0.2:4;

[U,V]=meshgrid(u,v);

X=cos(U).*sin(V);

Y=cos(U).*cos(V);

Z=sin(U);

surf(X,Y,Z)%画单位球

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

例：圆锥面 $x = u\sin v$，$y = u\cos v$，$z = u$；

```
clear;
u=-4:0.2:4;
v=-4:0.2:4;
[U,V]=meshgrid(u,v);
X=U.*sin(V);
Y=U.*cos(V);
Z=U;
mesh(X,Y,Z);
figure(2)
surf(X,Y,Z)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

统计图形的绘制

例 饼图：例如分割比例 2:4:3:5

pie([2 4 3 5])%二维的画饼图

% 0 表示连续（表示连在一起的），而 1 表示突出，{}中的为标注，

例 pie3([2 4 3 5],[0 1 1 0],{'North','South','East','West'})

例 直方图

代码：x=[30,100,20,30,40,55,65]; bar(x)

cylinde(r,n) —— 三维柱面绘图函数

r 为半径；n 为柱面圆周等分数

例：绘制三维陀螺锥面

```
t1=0:0.1:0.9;
t2=1:0.1:2;
r=[t1 -t2+2];
[x,y,z]=cylinder(r,30);
surf(x,y,z);
grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MATLAB 中程序结构

关系运算：

<：小于

<=：小于或等于

>：大于

>=：大于或等于

==：等于

~=：不等于

1.当两个比较量是**标量**时，直接比较两数的大小。

若关系成立，关系表达式结果为 1，否则为 0；

例

```
a=1;b=2;a>b       %运行结果：ans = 0
a=1;b=2;a~=b      %运行结果：ans = 1
```

2.当参与比较的量是两个维数相同的**矩阵**时，比较是对两矩阵相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终

的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由 0 或 1 组成。（关系操作,真返回 1，假返回 0）

例：a=1:9; b=10-a;

r0=(a<4)

r1=(a==b)

例 a=[1 2 3 4 5 6 7];

b=[7 6 5 4 3 2 1];

c1=a>b

运行结果：c1 =

0 0 0 0 1 1 1

3. 当参与比较的一个是标量，而另一个是矩阵时，则把标量与矩阵的每一个元素按标量关系运算规则逐个比较，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由 0 或 1 组成。

例：代码：d=[1 2 3 4 5 6 7];e=5; c2=d<=e

运行结果：c2= 1 1 1 1 1 0 0

逻辑操作（真返回 1，假返回 0）

逻辑“非”操作优于关系操作，关系操作优于逻辑操作（除逻辑非操作）

&与、和；两者都成立为真，否则为假

| 或；；两者只要有一个成立即可为真，否则为假

~ 否；如果条件为假，则为真，否则为假；

相异元素返回 1，相同元素返回 0；

例：a=-3:3;

A=~(a>0)

B=~a>0　　%与 A 的差别，理解逻辑操作优于关系操作

C=~a

D=a>-2&a<1 %与 D=(a>-2)&(a<1)的命令结果一致。

a=1;b=2;a&b　%a、b 可以是条件或是数字，数字非 0 皆为真

a|b

1.赋值语句同其他编程语言；

2.条件控制语句，基本格式：

if-else-end 分支结构

（1）if　expression

command

end

如果 expression 为真，执行命令 command，否则跳过该组命令

（2）如果有两种选择，可采用下面的结构

if　expression

command1

else

command2

end

如果如果 expression 为真，执行命令 command1，如果为假，执行

command2

（3）如果有多种选择，采用下面结构

if expression1　　%判决条件

command1　　%表达式为真，执行 command1，结束此结构

else if expression2　%判决表达式 2

command2　　%若表达式 1 为假，表达式 2 为真，执行 command2，结束此结构

else if expression3　%判决表达式 3

command3　　%若表达式 2 为假，表达式 3 为真，执行 command3，结束此结构

……

else

command n　　%当前面所有表达式为假时，执行 commandn 命令

end

例：输入一个数字，与 1 比较大小，分三种情况决定 y 的取值：代码

x=input('请输入 x=');

if x>1

y=x+1;

elseif x==1

y=x+2;

else%也可以输入 elseif x<1

y=x+3;

```
end
y
```

循环语句

```
for variable=i:j:k
语句
end
```

例 计算从 1 加到 100 之和

```
% 可以直接调用 MATLAB 的函数 sum(1:100)或者 sum([1:100])
%或者自己写代码
s=0;
for i=1:100
s=s+i;
end
s
```

例：计算 6！：%可以直接调用 MATLAB 的命令 prod(1:6)，或者自己写代码如下：

```
s=1;
for   n=1:6
s=s*n;
end
```

例：计算 20！%prod(1:20)是 20！

或者用 MATLAB 语句：

```
 s=1;
for   n=1:20
s=s*n;
end
s
```

或者自己写函数，保存为 m 文件，调用：myprod(6)

```
function s=myprod(n)%函数调用时直接输入 myprod(10)
myprod=1;
for i=1:1:n
myprod=myprod*i;
end
s=myprod;
end
```

建立函数文件—function 的用法

function 用来定义函数，一般创建一个函数要放在一个.m 文件里来定义。举个简单的例子，比如建立一个 myfunction.m，实现函数功能为 sin(x)+cos(y)，然后在文件中写为：

```
function   z=myfunction(x,y)
z=sin(x)+cos(y);
end
```

保存文件的时候务必要把文件的名字跟 myfunction 保持一致。

然后就可以在 matlab 窗口直接调用 myfunction(x,y)来实现计算

sin(x)+cos(y)的功能。

%设置路径→搜索路径，添加文件夹，然后才能调用，例如调用代码如下：

```
x=pi/4;y=pi/4;myfunction(x,y)
ans = 1.4142
```

写函数保存并且调用的例子

```
function [s,p]=fcircle(r)%调用的时候用[s,p]=fcircle(3)
%CIRCLE    calculate the area and perimeter of a circle of radii r
%r                 圆半径
%s                 圆面积
%p                 圆周长
s=pi*r*r;
p=2*pi*r;
end
```

while 循环结构

```
while expression
commands;
end
```

例：计算从 1 加到 100 之和

```
sum=0;
i=1;
while i<101;
```

sum=sum+i;

i=i+1;

end

sum

switch-case 结构

switch expression

case　test1

command1;　　%当　ex 等于 test1 时，执行 command1，然后跳出该结构

case　test2

command2;　　%当　ex 等于 test2 时，执行 command2，然后跳出该结构

case　test3

command3;　　%当　ex 等于 test3 时，执行 command3，然后跳出该结构

….

case　testk

commandk;　　%当　ex 等于 testk 时，执行 commandk，然后跳出该结构

otherwise

commandn　　%当　ex 不等于前面所有检测值时，执行该组命令 commandn

```
end
```

例：比如输入分数的例子：

```
t=input('填入你考试的分数  n=');
flag=floor(t/25);%向下取整的意思
switch flag
case 4%case 后面不能写区间
disp('绝顶聪明！');
case 3
disp('优秀！');
case 2
disp('一般！');
case 1
disp('同志仍需努力！');
case 0
disp('加油！');
otherwise
disp('输入分数有问题，请重试！');
end
```

也可以用 if 和 elseif 来写

例：比如输入分数的例子：

```
t=input('填入你考试的分数  n=');
if t>100|t<0
```

```
disp('输入分数有问题，请重试！');
elseif   t==100
disp('聪明绝顶！');
elseif t>=80%可以是区间
disp('优秀！');
elseif t>=60
disp('一般！');
else
disp('加油！');
end
```

break 语句和 continue 语句

与循环结构相关的语句还有 break 语句和 continue 语句。它们一般与循环语句配合使用。

break：终止循环的执行。

continue：结束本次循环，继续下一次循环。

例　求[100，200]之间第一个能被 21 整除的整数。

程序如下：

```
for n=100:200
    if rem(n,21)~=0
            continue
    end
    break
```

```
end
n
```

或者：

```
for n=100:200
    if rem(n,21)==0
            break
    end
end
n
```

例　猜数游戏。首先由计算机产生[1,100]之间的随机整数，然后由用户猜测所产生的随机数。根据用户猜测的情况给出不同提示，如猜测的数大于产生的数，则显示“High”，小于则显示“Low”，等于则显示“You won”，同时退出游戏。用户最多可以猜 7 次。

代码：%用到了 for 循环，if 判断语句，break 跳出暂停语句等

```
x=fix(100*rand);      %a random number calculated by the computer 电脑自己随机生成的数字
n=7;
test=1;
for k=1:7
      numb=int2str(n);% int2str(N)  将 n 视为整数矩阵，并将其转换为表示整数的字符数组
      disp(['You have a right to ',numb,' guesses'])
```

```
    disp(['A guess is a number between 0 and 100'])
    guess=input('Enter your guess:');
    if guess<x
        disp('Low')
    elseif guess>x
        disp('High')
    else
        disp('You won')
        test=0;
        break;
    end
    n=n-1;
end
if test==1
    disp('You lost')
end
```

例如：用循环法求积分举例子： $s=\int_a^b e^{-\frac{x}{2}}*\sin(x+pi/6)$

求函数 f(x)在[a,b]上的定积分，其几何意义就是求曲线 y=f(x)与直线 x=a，x=b，y=0 所围成的曲边梯形的面积。为了求得曲边梯形面积，先将积分区间[a,b]分成 n 等分，每个区间的宽度为 h=(b-a)/n，对应地将曲边梯形分成 n 等分，每个小部分即是一个小曲边梯形。近似求出每个小曲边梯形面积，然后将 n 个小曲边梯形的面积加起来，就得到

总面积，即定积分的近似值。近似地求每个小曲边梯形的面积，常用的方法有：矩形法、梯形法以及辛普生法等。

如果直接调用 matlab 的积分代码：

```
quad('exp(-0.5.*x).*sin(x+pi/6)',0,3*pi)
```

运行结果：

```
ans =

    0.9008
```

或者自己写函数代码：

```
a=0;b=3*pi;
n=1000; h=(b-a)/n;
x=a; s=0;
f0=exp(-0.5*x)*sin(x+pi/6);
for i=1:n
    x=x+h;
    f1=exp(-0.5*x)*sin(x+pi/6);
    s=s+(f0+f1)*h/2;
    f0=f1;
end
s
```

```
s=

  0.9008
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

其他一些题目

1.求 n 个人中至少有两个人生日相同的概率：

代码：

```
n=20;1-prod(365-n+1:365)./(365.^n)
%prod(4:6)=120
%从 4 乘以 5 乘以 6
```

2.求一个三位数之和为 k 的概率。

```
%nchoosek(4,2)  是 4*3/2 的意思.从 4 个里面选 2 个
```

代码：

```
n=0;k=21;
if(k>27||k<1)
    p=0
else if(k>18&k<=27)
        for i=k-18:9
            a=19+i-k+a
        end
            p=a/900
    else if(k>9&k<=18)
            for i=1:9
                if k-i>9
                    a=19+i-k+a
                else a=a+k-i+1
```

```
                end
            end
            p=a/900
        else if(k>1&k<=9)
                p=(nchoosek(k,2)+k)/900
            else p=1/900
            end
        end
    end
end
```

插值：插值可以理解为，要根据一个用表格表示的函数，计算表中没有的函数

插值命令用法介绍：

interp1(x,y,xq,’method’)-x,y 是给定点的横坐标和纵坐标形成的向量；xq 是插值点对应的横坐标（如果是向量，则对应多个插值点的横坐标）；‘method’是所使用的插值方法，缺失默认为线性插值，可选择为：'linear'-线性插值；'cubic'-三次埃尔米特插值；'spline'-样条插值。

例：

```
>>x=[0.5 0.6 0.7]';
y=[-0.693147 -0.510826 -0.356675]';
yi_1=interp1(x,y,0.54)%线性插值
yi_2=interp1(x,y,0.54,'spline')%三次样条插值
```

另外也有二维函数插值命令：interp2(x,y,z,xq,yq,’method’);用法类似，需要时自己查阅资料使用。

数据拟合

多项式拟合命令介绍：

polyfit(x,y,n)—x 为拟合数据点对应的横坐标，y 为拟合数据点对应的纵坐标，n 为要拟合的多项式的次数。

例 1. 某数据如下，试找出 x 与 y 之间的函数关系

```
x=[0.0        0.25       0.50        0.75          1];
y=[0.10     0.35       0.81       1.09     1.96];
clear all
x=[0.0        0.25       0.50        0.75          1];
y=[0.10     0.35       0.81       1.09     1.96];
p1=polyfit(x,y,1)%一次拟合
xx=0:0.1:1;
y1=polyval(p1,xx);
plot(x,y,'r*')
hold on;
plot(xx,y1,'g')
```

例 2.：

```
clear all
x=[0.0 0.9 1.9 3.0 3.9 5.0]';
y=[0     10     30    50    80    110]';
```

p1=polyfit(x,y,1)%1 次多项式拟合（是个向量，表达的是多项式的系数）

p2=polyfit(x,y,2)%2 次多项式拟合

```
xx=0:0.1:5;
y1=polyval(p1,xx);
y2=polyval(p2,xx);
plot(x,y,'r*');
hold on;
plot(xx,y1,'g');
plot(xx,y2,'b');
```

简单的统计命令

1：频数表制作

[N,M]=hist(x,k)—N 为 k 个小区间的频数形成的向量，M 为 k 个小区间的中点形成的向量，此命令为数组 x 的频数表，它将区间[min(x),max(x)]等分为 k 份，缺省时 k=10。

```
clear all
x=[1 2 3 4 5 6 7 1 2 5 8];
k=6;
[N,M]=hist(x,k)
```

hist(x,k)%不加[N,M]，则结果为制作数组 x 的直方图，k 的意义同上

2：mean(x)—求数组 x 的均值;

3：median(x)—求数组 x 的中位数（若向量 x 是偶数，结果取值为

（x(n/2)+x(n/2+1)）/2

4：std_x=[std(x),var(x),range(x)]--返回数组的标准差，方差，极差（若除以 n 则用 std(x,1),var(x,1)）；

5：cov(x,y)—求向量 x 和 y 的协方差；

6：corr1=corrcoef(x,y)—求向量 x，y 的相关系数；

线性规划求解

x = linprog(c , A , b , Aeq , beq , lb , ub)是求解如下线性规划问题的命令：

$$FVAL = \min c^T x$$
$$s.t.\begin{cases} Ax \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases}$$

其中 c 是目标函数的系数向量，A 是不等式约束 $Ax \le b$ 的系数矩阵，b 是不等式约束 $Ax \le b$ 的常数项；

Aeq 是等式约束 $Aeq \cdot x = beq$ 的系数矩阵，beq 是等式约束 $Aeq \cdot x = beq$ 的常数；

lb 是 X 的下限，ub 是 X 的上限，x 是向量[x1,x2,...xn]即决策变量。

如果模型中不包含不等式约束条件，可用［］代替 A 和 b 表示缺省；如果没有等式约束条件，可用［］代替 Aeq 和 beq 表示缺省；如果某个 xi 无下界或上界，可以设定 lb（i）＝－inf 或 ub（i）＝inf；

注意：此命令求的是目标函数最小值，而不是最大值。如果求最大值，可以在目标函数前乘一个-1；约束不等式要求小于等于，通过两边乘以-1，可以把大于等于的不等式转化成小于等于。

[x, FVAL] = linprog(c , A , b , Aeq , beq , lb , ub)

用［x , FVAL］代替上述各命令行中左边的 x，则可得到在最优解 x 处的函数值 FVAL；

例：求解线性规划问题

$$\begin{cases} \min z = -2x_1 - x_2 + x_3 \\ s.t.\ \ 3x_1 + x_2 + x_3 \le 60 \\ \quad x_1 - x_2 + 2x_3 \le 10 \\ \quad x_1 + x_2 - x_3 \le 20 \\ \quad x_j \ge 0, j = 1,2,3 \end{cases}$$

代码：

```
c=[-2 -1 1]';%不加矩阵的转置也可以
A=[3 1 1;1 -1 2;1 1 -1];
b=[60 10 20]';
Aeq=[];beq=[];
lb=[0 0 0]';
ub=[inf inf inf]';
[x, FVAL] = linprog( c , A , b , Aeq , beq , lb , ub )
```

运行结果：

```
x =
    15.0000
     5.0000
     0.0000
FVAL =
   -35.0000
```

例：求解线性规划问题

$$\begin{cases} \min z = x_1 - x_2 + x_3 + x_5 - x_6 \\ s.t. \quad 3x_3 + x_5 + x_6 = 6 \\ x_2 + 2x_3 - x_4 = 10 \\ -x_1 + x_6 = 0 \\ x_3 + x_6 + x_7 = 6 \\ x_j \geq 0, j = 1, \cdots, 7 \end{cases}$$

代码：

```
clear;
c=[1 -1 1 0 1 -1 0]';
A=[];b=[];
Aeq=[0 0 0 3 1 1 0;0 1 2 -1 0 0 0;-1 0 0 0 0 1 0;0 0 1 0 0 1 1];
beq=[6 10 0 6];
lb=[0 0 0 0 0 0 0]';
ub=[inf inf inf inf inf inf inf]';
[x, FVAL] = linprog( c , A , b , Aeq , beq , lb , ub )
```

运行结果：

```
x =
    0.0000
   12.0000
    0.0000
    2.0000
    0.0000
    0.0000
```

6.0000

FVAL =

-12.0000

%%

MATLAB 制作动画：

一般来讲，matlab 制作动画有四种方式：

1.以质点运动轨迹来显示动画

2.以电影播放的方式来显示动态效果

3.以对象方式显示

4.以旋转颜色的方式显示

Demo:demos/…/vibrating logo

质点运动轨迹：

使用 comet、comet3 函数，前者是二维，后者是三维 。

comet(y)　　显示质点绕向量 y

comet(x,y)　　显示质点绕向量 y 与 x

comet(x,y,p)　其中 p 为轨迹尾巴的长度，

默认 p=0.1

例：模拟平抛运动：

```
vx = 40;
t = 0:0.001:10;
x = vx*t;
y = -9.8*t.^2/2;
```

comet(x,y,0.22)

例：模拟导弹发射：

vx = 100*cos(1/4*pi);

vy = 100*sin(1/4*pi);

t = 0:0.001:15;

x = vx*t;

y = vy*t-9.8*t.^2/2;

comet(x,y)

匀速圆周运动：

sita = 0:0.0001:2*pi;

r = 10;

x=r*cos(sita);

y=r*sin(sita);

comet(x,y)

注： 如何调节速度？使用步长调节速度！

sita = 0:0.0001:2*pi;

r = 10;

x=r*cos(sita);

y=r*sin(sita);

comet(x,y)

%comet3 与 comet 的用法相类似，可以在帮助文件里找到例子：

t = -10*pi:pi/250:10*pi;

comet3((cos(2*t).^2).*sin(t),(sin(2*t).^2).*cos(t),t)

电影播放次序模式：

保存想要产生动画的图片，存储为一系列各种类型的二维、三维图，再像放电影的方式按次序播放出来。

动画生成的步骤：

1.调用 moviein 函数对内存进行初始化，创建一个足够大的矩阵，存放当前坐标轴大小的一系列指定图形。

2.对动画中的每一帧生成图形，并把它们放到帧矩阵中(即:将当前的图片抓取为电影的画面) —— getframe

3.从帧矩阵中回放动画—— movie，可以指定次数和播放速度。

例：

```
axis equal    %坐标相同
M=moviein(8)    %产生矩阵
set(gca,'Nextplot','replacechildren')
for    j=1:8
plot(fft(eye(j+8)))
%eye 为单位矩阵，fft 为快速傅立叶变换
M(:,j)=getframe;
end
movie(M,2,1)    %M 为播放对象，2 为播放次数，1 为每秒播放的帧数，默认为 12
```

例子：一球绕一曲线前进：

```
clc;clear;
n=100;
x=0:pi/n:2*pi
y=sin(x);
k=0;
for t=0:pi/n:2*pi
k=k+1;
x(k)=t
y(k)=sin(t);
m=plot(x,y,x(k),y(k),'or')
grid
getframe;
end
```