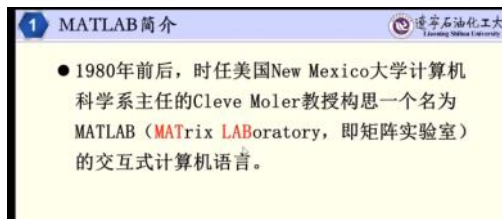


一、简介

2020年2月17日 15:30

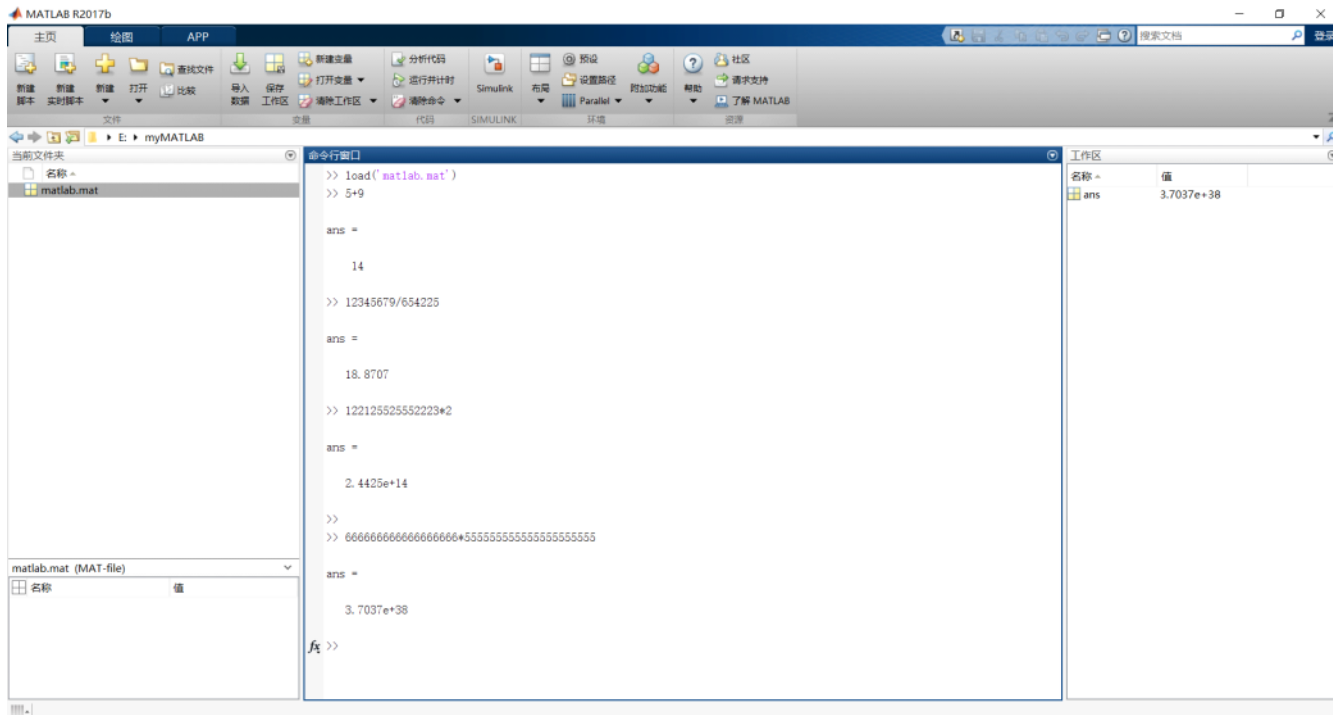
1.优势

- 1) 简洁高效 (数学方面)
 - 2) 科学运算功能 (以矩阵为基本单位)
 - 3) 强大的绘图功能
 - 4) 庞大的工具箱
2. 不足
- 1) 解释型语言, 执行速度慢。(边“编译”边执行)
 - 2) 软件包越来越大



类型:	文件夹
位置:	D:\
大小:	16.9 GB (18,156,231,689 字节)
占用空间:	17.6 GB (18,977,538,048 字节)
包含:	580,863 个文件, 39,377 个文件夹

3. 界面



二、基本运算和基本函数

2020年2月17日 15:39

1.常用常量

eps: 机器浮点运算误差限, 默认值 $2.2204e-16$
i 和 **j**: 纯虚数量, $i = \sqrt{-1}$
Inf: 正无穷大, Inf表示 $+\infty$, -Inf表示 $-\infty$
NaN: 不定式(不是数), 通常由0/0或Inf/Inf等运算得到
pi: π

```
>> pi/2

ans =

    1.5708

>> i

ans =

    0.0000 + 1.0000i

>> 9/i

ans =

    0.0000 - 9.0000i
```

2.变量

字母, 数字, 下划线, 数字的组合。

区分大小写

ans存放最近的运算结果

```
ans =

    0.0000 - 9.0000i
```

3.基本运算

+ - * / 乘方 ^

4.基本函数

sinx=sindy

x是弧度, y是角度。

基本函数:	sin(x)	正弦 (变量为弧度)
	cos(x)	余弦 (变量为弧度)
	tan(x)	正切 (变量为弧度)
	cot(x)	余切 (变量为弧度)
	sind(x)	正弦 (变量为度数)
	cosd(x)	余弦 (变量为度数)
	tand(x)	正切 (变量为度数)

<code>tand(x)</code>	正切（变量为度数）
<code>cotd(x)</code>	余切（变量为度数）

基本函数：

<code>asin(x)</code>	反正弦（返回弧度）
<code>acos(x)</code>	反余弦（返回弧度）
<code>atan(x)</code>	反正切（返回弧度）
<code>acot(x)</code>	反余切（返回弧度）
<code>asind(x)</code>	反正弦（返回度数）
<code>acosd(x)</code>	反余弦（返回度数）
<code>atand(x)</code>	反正切（返回度数）
<code>acotd(x)</code>	反余切（返回度数）

基本函数：

<code>exp(x)</code>	e^x
<code>log(x)</code>	以 e 为底的对数
<code>log2(x)</code>	以 2 为底的对数
<code>log10(x)</code>	以 10 为底对数
<code>sqrt(x)</code>	开方
<code>realsqrt(x)</code>	返回非负平方根
<code>abs(x)</code>	取绝对值
<code>sign(x)</code>	符号函数

基本函数：

<code>angle(z)</code>	复数 z 的相角(Phase angle)
<code>real(z)</code>	复数 z 的实部
<code>imag(z)</code>	复数 z 的虚部
<code>conj(z)</code>	复数 z 的共轭复数
<code>round(x)</code>	四舍五入至最近整数
<code>fix(x)</code>	无论正负，舍去小数至最近整数
<code>floor(x)</code>	下取整，小于等于 x 的最大整数
<code>ceil(x)</code>	上取整，大于等于 x 的最小整数

取舍函数

floor地板函数
ceil天花板函数

sum(1:i)求和函数

```

命令窗口
>> round(exp(1))
ans =

```

sum(1:i)求和函数

```
>> round(exp(1))  
  
ans =  
  
    3  
  
>> fix(exp(1))  
  
ans =  
  
    2  
  
>> fix(-2.11)  
  
ans =  
  
   -2  
  
>> floor(9.3215)  
  
ans =  
  
    9  
  
>> ceil(2.33)  
  
ans =  
  
    3
```

三、实用技巧

2020年2月17日 16:14

1.help命令

2.doc命令

3.clc清屏

4.clear清变量

5.历史命令

6.显示结果，分号说了算

7.命令补全，Tab键补全

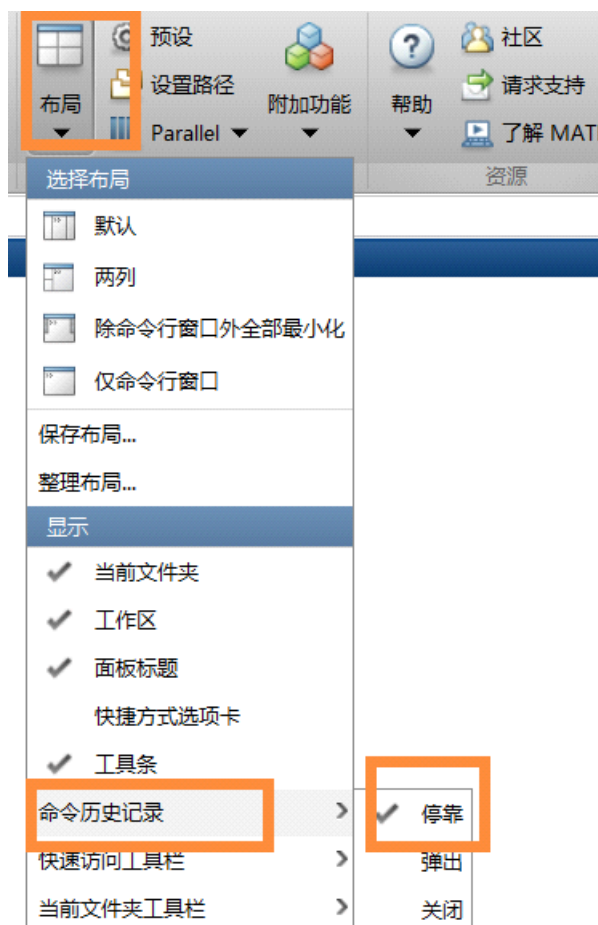
8.注释的方法，用%

注释一大段代码：Ctrl+R

解除注释：Ctrl +T

9.%%分割节

10.关闭绘图窗口close all



四、Matlab与矩阵

2020年2月17日 20:57

1.矩阵的输入(三种方法):

A = [1,2,3;4,5,6;7,8,9]

B = [1 2 3;4 5 6;7 8 9]

C=[1 23 123

2 22 222

4 44 444]

```
out =
```

```
30.500000000000000
```

```
>> B=[1 2 3;4 5 6;7 8 9]
```

```
B =
```

```
1     2     3
4     5     6
7     8     9
```

2.向量的输入:

列向量: E=[1;2;3;4;5;6]

行向量: D=[1,22,333,4444,55555,666666]

```
>> E=[1;2;3;4;5;6]
```

E =

```
1
2
3
4
5
6
```

```
>> D=[1, 22, 333, 4444, 55555, 666666]
```

D =

1 至 4 列

```
1      22      333      4444
```

5 至 6 列

```
55555      666666
```

3.冒号表达式

$V=s_1:s_2:s_3$

起始值 步长 最大值

(默认步长是一, 生成的是行向量)

```
>> G=1:15
```

G =

1 至 9 列

```
1      2      3      4      5      6      7      8      9
```

10 至 15 列

```
10     11     12     13     14     15
```

4.linspace函数生成等间距行向量

Linspace(a,b,n)

a到b等间距的生成n个值

```
>> I=linspace(1,10,11)
```

```
I =
```

1 至 2 列

```
1.0000000000000000    1.9000000000000000
```

3 至 4 列

```
2.8000000000000000    3.7000000000000000
```

5 至 6 列

```
4.6000000000000000    5.5000000000000000
```

7 至 8 列

```
6.4000000000000000    7.3000000000000000
```

9 至 10 列

```
8.1999999999999999    9.1000000000000000
```

11 列

```
10.0000000000000000
```

5.特殊矩阵

特殊矩阵的输入：

<code>zeros(n)</code>	生成n阶零矩阵(所有元素为0)
<code>zeros(m,n)</code>	生成m*n零矩阵
<code>eye(n)</code>	生成n阶单位阵
<code>eye(m,n)</code>	生成m*n主对角线元素为1的矩阵
<code>ones(n)</code>	生成n阶1矩阵(所有元素为1)
<code>ones(m,n)</code>	生成m*n的1矩阵(所有元素为1)
<code>rand(n)</code>	生成n阶随机数矩阵([0,1]均匀分布随机数)
<code>rand(m,n)</code>	生成m*n随机数矩阵

```
>> eye(5)
```

```
ans =
```

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

```
>> eye(3,4)
```

```
ans =
```

```
1 0 0 0
0 1 0 0
0 0 1 0
```

单位矩阵

在矩阵的乘法中，有一种矩阵起着特殊的作用，如同数的乘法中的1，这种矩阵被称为单位矩阵。它是个方阵，从左上角到右下角的对角线（称为主对角线）上的元素均为1。除此以外全都为0。根据单位矩阵的特点，任何矩阵与单位矩阵相乘都等于本身。

1.矩阵的访问

A (行, 列)

A (1, :) 选择第一行

A (2, 2: 4) 选择第二行的2, 3, 4列的数据

A(2,3:end)选择第二行从3到最后的数据

A([1,3],[1,4])第一行和第三行与第一列和第四列的交集

```
J =
```

```
    12    12    52    56    45
    45    42   552    55    55
    12    11    11    11    11
    21    21    12    21    12
```

```
>> J([2, 4], [1, 2])
```

```
ans =
```

```
    45    42
    21    21
```

2.重新赋值

```
>> B=[B,[3;6;9]]
```

```
B =
```

```
    1    2    3
    4    5    6
    7    8    9
```

换行用; 行内换列用,

1.矩阵的基本运算

A+B 矩阵A与矩阵B的和，要求维数相同

A-B 矩阵A与矩阵B的差，要求维数相同

A*B 矩阵A与矩阵B的正常积，要求维数满足相乘的条件

A.*B Hadamard积, 对应元素乘积，要求矩阵维数相同

A./B 对应元素相除，要求矩阵维数相同

A^n 方阵A自乘n次

A.^n 矩阵A中每个元素的n次方

A/B 矩阵A右除矩阵B，相当于A乘以B的逆

A\B 矩阵A左除矩阵B，相当于A逆乘以B

hadamard乘.*对应元素的乘积

$A/B = B \backslash A$

+ - *

```
>> B+C
```

```
ans =
```

```

     2     25    126
     6     27    228
    11     52    453
```

```
>> B-C
```

```
ans =
```

```

     0    -21   -120
     2    -17   -216
     3    -36   -435
```

```
>> B*C
```

```
ans =
```

```

    17    199   1899
    38    466   4266
    59    733   6633
```

. *

```
>> B.*C
```

```
ans =
```

1	46	369
8	110	1332
28	352	3996

```
./
```

```
>> C./B
```

```
ans =
```

1.0000000000000000	11.500000000000000	41.000000000000000
0.5000000000000000	4.400000000000000	37.000000000000000
0.571428571428571	5.500000000000000	49.333333333333336

^ 和 .^

^必须是方阵

```
>> C^3
```

```
ans =
```

251785	2776103	27954603
436322	4810750	48442950
872644	9621500	96885900

.^无需是方阵

```
>> D.^2
```

```
ans =
```

1	484	1089	1936	25	36
---	-----	------	------	----	----

1.矩阵的常用函数

A.'矩阵的一般转置

A'矩阵的Hermit转置 (A如果是复数矩阵, 则先变为共轭复数, 再转置)

Inv(A)方阵A的逆, 要求A为方阵

Det(A)方阵的行列式

Trace(A)矩阵的迹：在线性代数中，一个 $n \times n$ 矩阵A的主对角线（从左上方至右下方的对角线）上各个元素的总和被称为矩阵A的迹（或迹数），一般记作 $\text{tr}(A)$ 。

$\text{eig}(A)$ 方阵A的特征值：设A是 n 阶方阵，如果存在数 m 和非零 n 维列向量 x ，使得 $Ax=mx$ 成立，则称 m 是矩阵A的一个特征值（characteristic value）或本征值（eigenvalue）。

$[v,d]=\text{eig}(A)$ 方阵的特征值及特征向量

转置

```
>> B'

ans =

     1     4     7
     2     5     8
     3     6     9

>> B.'

ans =

     1     4     7
     2     5     8
     3     6     9
```

迹和行列式

```
>> trace(B)

ans =

    15

>> det(B)

ans =

-9.516197353929915e-16
```

特征值和特征向量

```
>> [v,d]=eig(B)

v =

-0.231970687246286 -0.785830238742067  0.408248290463863
-0.525322093301234 -0.086751339256628 -0.816496580927726
-0.818673499356181  0.612327560228810  0.408248290463863
```

```
>> [v,d]=eig(B)

v =

-0.231970687246286 -0.785830238742067 0.408248290463863
-0.525322093301234 -0.086751339256628 -0.816496580927726
-0.818673499356181 0.612327560228810 0.408248290463863

d =

16.116843969807043      0      0
      0 -1.116843969807042      0
      0      0 -0.0000000000000001
```

2.其他函数:

rank(A)	矩阵A的秩
diag(v)	生成以向量v为主对角线元素的方阵
diag(v,k)	生成主对角线上第k条对角线元素为v的方阵
poly(A)	结果是一个行向量，从最后一个开始往前， 分别为常数、x一次项的系数、x二次项系数，
[m,n]=size(A)	返回A的行数和列数

矩阵的秩:

矩阵的秩是线性代数中的一个概念。在线性代数中，一个矩阵A的列秩是A的线性独立的纵列的极大数，通常表示为 $r(A)$ ， $rk(A)$ 或 $\text{rank } A$ 。

<code>min(x)</code>	向量x的元素的最小值
<code>max(x)</code>	向量x的元素的最大值
<code>mean(x)</code>	向量x的元素平均值
<code>median(x)</code>	向量x的元素的中位数
<code>std(x)</code>	向量x的元素的标准差
<code>diff(x)</code>	向量x的相邻元素的差
<code>sort(x)</code>	对向量x的元素进行排序
<code>length(x)</code>	向量x的元素个数

1.矩阵在图片处理中的简单应用

实时脚本：

对图片的处理

显示: `imshow (A)`

rgb转灰度图片: `rgb2gray(A)`

旋转: `rot90(Ag,2)`

抠图

融合

局部打马赛克

导入图片

```
A=imread(' timg003. jpg' );
```

```
B=imread(' OIP. jpg' );
```

显示图片

```
imshow(A)
```

```
imshow(B)
```

rgb转化为灰度图片

```
Ag=rgb2gray(A);
```

```
Bg=rgb2gray(B);
```

```
imshow(Ag)
```

```
imshow(Bg)
```

图片的旋转

```
imshow(rot90(Ag))//逆时针旋转
```

```
imshow(rot90(Ag,-1))
```

```
imshow(rot90(Ag,2))
```

图片的翻转

```
imshow(fliplr(Ag))
```

```
imshow(flipud(Ag))
```

```
imshow(Ag')
```

图片的压缩

```
imshow(Ag(1:4:end, 1:4:end))
```

图片的裁剪

```
Agx=Ag(536:end, 1:end);
```

```
imshow(Agx)
```

图像的融入

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

```
Ag2=Ag;
```

```
Bg2=Bg;
```

```
Bg2(Bg2>230)=42;
```

```
Ag2(43:(43+m-1), 95:(95+n-1))=Bg2;
```

```
imshow(Ag2)
```

局部打马赛克

```
C=imread('timg2.jpg');
```

```
Cg=rgb2gray(C);
```

```
imshow(Cg)
```

```
Cg2=Cg;
```

```
for i=298:349
```

```
for j=500:557
```

```
Cg2(i, j)=mean(mean(Cg(i:i+15, j:j+15)));
```

```
end
```

```
end
```

```
imshow(Cg2)
```


五、MATLAB绘图功能

2020年2月18日 19:44

1. 二维图的绘制

1) 显函数的绘制

figure创建新窗口

plot (x,y)

例：在 $0 \leq x \leq 2\pi$ 上绘制 $y = 2e^{-0.5x} \cos(4\pi x)$

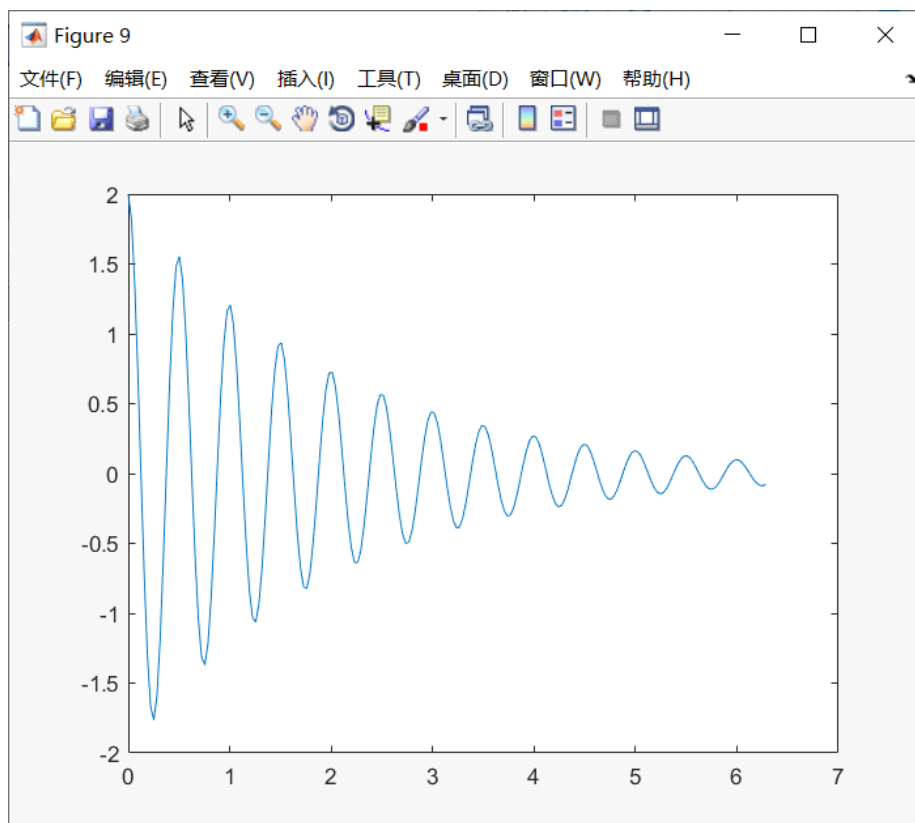
%绘制显函数

figure;

x=0:pi/100:2*pi;

y=2*exp(-0.5*x).*cos(4*pi*x);

plot(x,y);



2) 参数方程

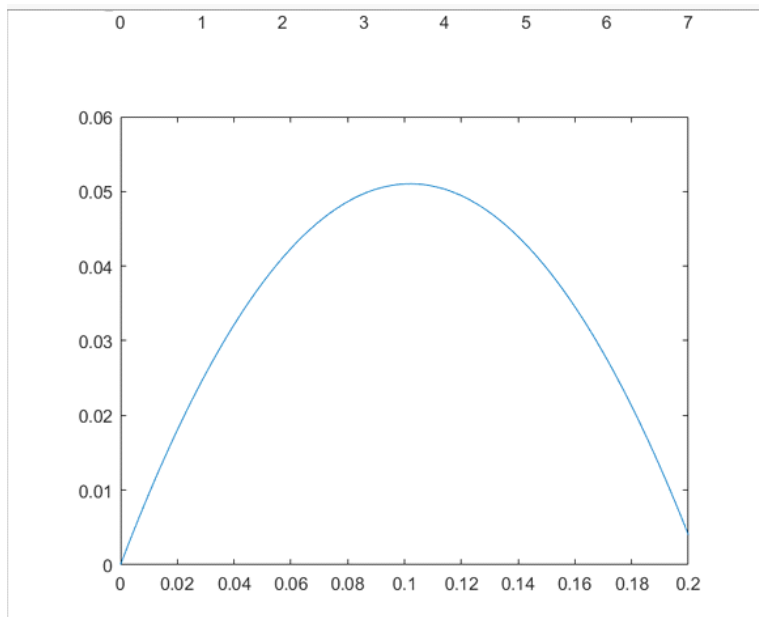
在 $0 \leq t \leq 0.2$ 上绘制抛物线方程:

$$\begin{cases} x = v_1 t \\ y = v_2 t - \frac{1}{2} g t^2, (v_1 = v_2 = 1, g = 9.8) \end{cases}$$

参数方程

figure;

```
t=0:0.001:0.2;
x=t;
y=t-0.5*9.8*t.^2;
plot(x,y);
```



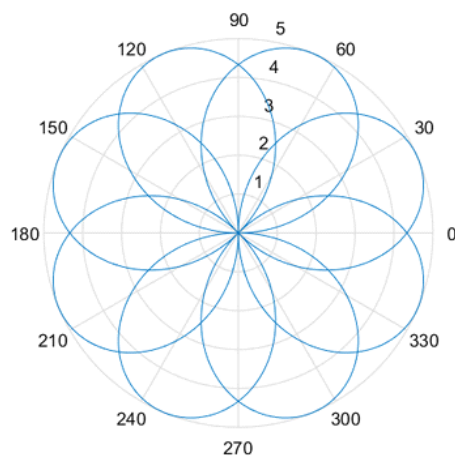
3) 极坐标下绘图

`Polar(theta,rho)` theta为极角, rho为极径。

例：在极坐标系下绘制 $\rho = 5\sin(4\theta/3)$, ($0 \leq \theta \leq 6\pi$)

极坐标下

```
figure;
th=0:0.01:6*pi;
rh=5*sin(4*th/3);
polar(th,rh);
```

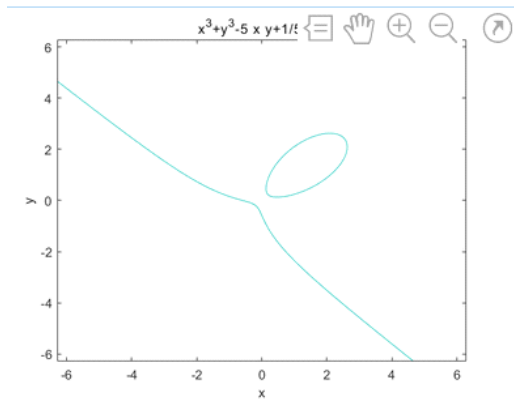


4) 直角坐标系隐函数的绘图

`ezplot(f)` 默认-2pi-2pi
`Ezplot(f,[a,b])`

```
ezplot('x^3+y^3-5*x*y+1/5')
ezplot('x^3+y^3-5*x*y+1/5', [-4 4 -5 5])
ezplot('x^3+y^3-5*x*y+1/5', [-4 4])
```

```
ezplot('x^3+y^3-5*x*y+1/5')
```



2. 二维图的修饰

1) 利用工具栏中的工具

2) 利用命令修饰

plot(X,Y,LineSpec) 利用绘图参数LineSpec来设置线型、节点类型和颜色

2 二维图的修饰

遼寧石油化工大學
Liaoning Shihua University

线型

-	--	:	-.
实线	短划线	虚线	点划线

节点类型

o	+	*	.	x	s	d
^	v	>	<	p	h	

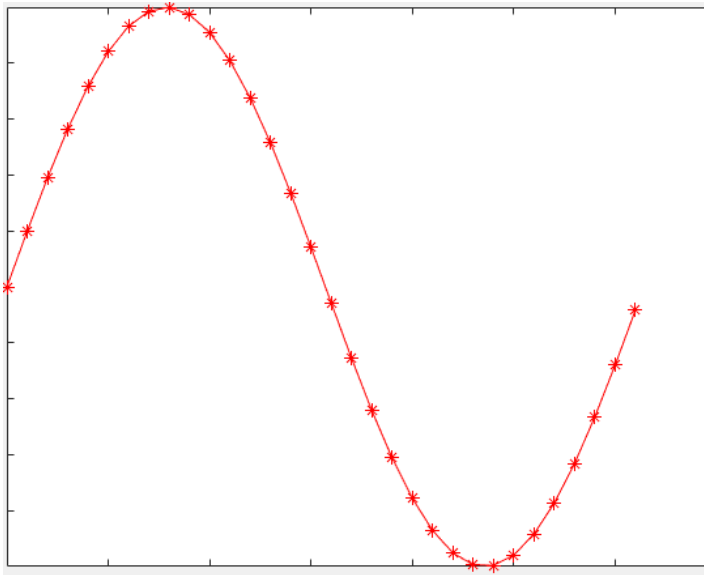
颜色

y	m	c	r	g	b	w	k
黄色	紫红	蓝绿	红色	绿色	蓝色	白色	黑色

于晶贤老师讲数学建模

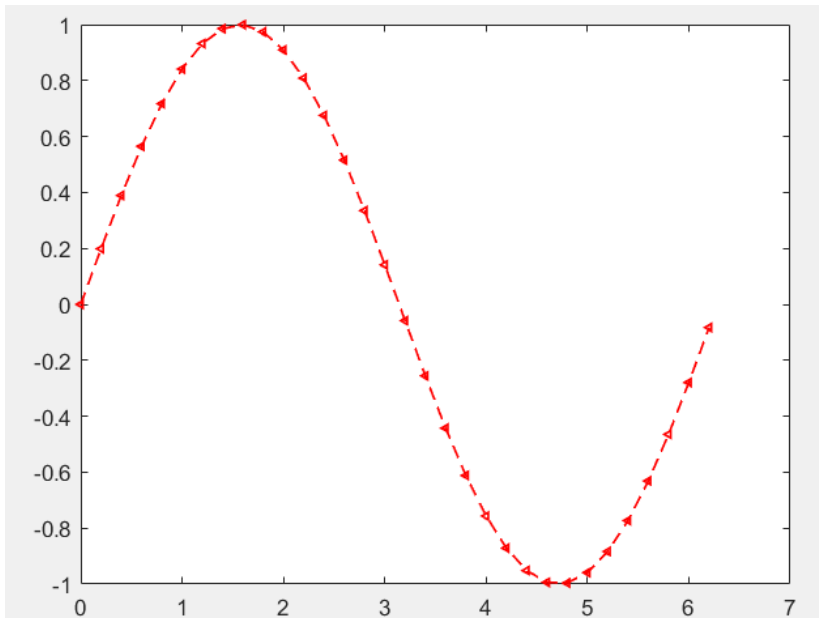
10/22

```
figure;
x=0:0.2:2*pi;
y=sin(x);
plot(x,y,'r*-')
```



plot(X, Y,属性1,属性1的值,属性2,属性2的值,)
修改线型(LineStyle)、节点类型(Marker)、节点大小(MarkerSize)、线宽(LineWidth)、颜色(Color)等.

```
figure;
x=0:0.2:2*pi;
y=sin(x);
%plot(x,y,'r*')
plot(x,y,'color','r','LineStyle','--','Marker','o','LineWidth',1,'MarkerSize',3)
```



图的题目和XY轴的名称, 曲线名称

```
title('生物的s型曲线');
```

```
xlabel('X轴');
```

```
ylabel('Y轴');
```

```
legend('曲线一');
```

3) 一个坐标系下画多条曲线

Hold on命令

```
%数学之心
```

```
figure;
x=-2:0.01:2;
y=sqrt(2*sqrt(x.^2)-x.^2);
z=asin(abs(x)-1)-pi./2;
plot(x,y);
grid on;
hold on;%在一个图中画多个图
plot(x,z);
title('');
legend('心');
```

4) 绘制子图

子图 `subplot(m,n,p)` 将图形窗口分成 $m \times n$ 个绘图区，即每行 n 个，共 m 行，区号按行优先编号，选定第 p 个区为当前活动区。在每一个绘图区单独绘制图形。

先行在列

%绘制2行2列的子图

```
clc;
x=-pi/2:pi/10:pi/2;
y1=sin(x);
y2=cos(x);
y3=tan(x);
y4=cot(x);
subplot(2,2,1),plot(x,y1,'ro-');
title('sinx');
subplot(2,2,2),plot(x,y2,'b--');
title('cosx');
subplot(2,2,3),plot(x,y3,'g*-');
title('tanx');
subplot(2,2,4),plot(x,y4,'ys:');
title('cotx');
```

3.三位图的绘制

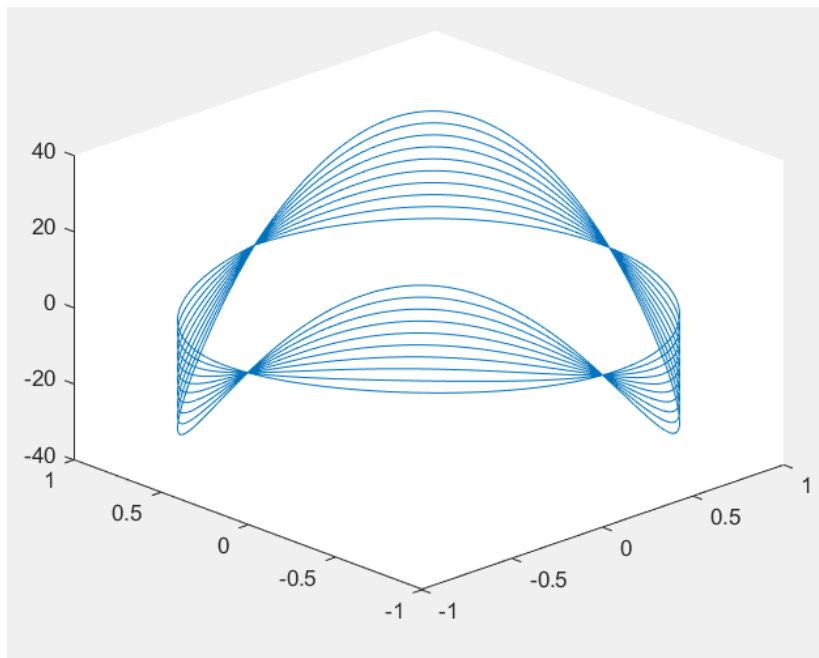
1) 绘制三维曲线

`plot3 (x,y,z)`

例：画出如下曲线
$$\begin{cases} x = \sin t \\ y = \cos t \\ z = t \sin t \cos t \end{cases} \quad (0 \leq t \leq 20\pi)$$

```
clc;
clear;
figure;
t=0:pi/100:20*pi;

x=sin(t);
y=cos(t);
z=t.*sin(t).*cos(t);
plot3(x,y,z);
```



2) 三维曲面的绘制

用到mesh和meshgrid函数

%三维曲面

clc;

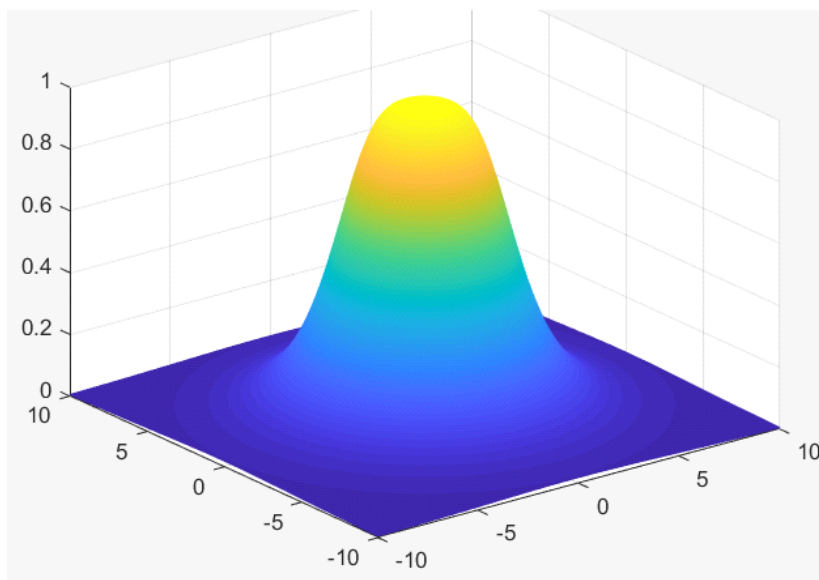
figure;

[x,y]=meshgrid(-10:0.01:10);

z=1./(1+(x.^2+y.^2).^2/200);

mesh(x,y,z);

%surf/surfl





















































3) 统计图形

直方图，散点图，柱状图和饼图.....

Types of MATLAB Plots

There are various functions that you can use to plot data in MATLAB®. This table classifies and illustrates the common graphics functions.

Line Plots	Pie Charts, Bar Plots, and Histograms	Discrete Data Plots	Polar Plots	Contour Plots	Vector Fields	Surface and Mesh Plots		Volume Visualization	Animation	Images
										
										
										
										
										
										
										
										
										
										

心性函数：

%数学之心

figure;

clc;

x=-2:0.01:2;

y=sqrt(2*sqrt(x.^2)-x.^2);

z=asin(abs(x)-1)-pi./2;

plot(x,y);

grid on;

hold on;%在一个图中画多个图

plot(x,z);

title("");

legend('心');

六、自定义函数

2020年2月19日 22:03

1. 创建自定义函数

1. 在Matlab中新建脚本文件

2. 按如下格式要求书写函数

```
function [输出变量列表] = 函数名(输入变量1, 输入变量2, ...)  
    根据输入变量计算各个输出变量  
end
```

输出变量不止一个可以用“,” 隔开

3. 保存脚本，脚本的名字必须函数名

区分大小写

4. 在其他程序中调用此函数，调用格式为

```
[输出变量列表] = 函数名(输入变量1, 输入变量2, ...)
```


七、循环和选择结构

2020年2月20日 20:18

1.for循环

for循环的一般结构：

```
for i=V
    循环结构体
end
```

i是循环变量，V是一个向量，i在V中不断的取值，每取一个值就执行一次循环体的内容

1) 例子

例1：利用for循环编写程序计算 $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{100}$

```
s=s+1/i;
clc;
clear;
s=0;
for i=1:100%运行100次
    s=s+1/i;
end
```

例2：利用for循环编写程序计算 $S = 1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \cdots + \frac{1}{99}$

```
clc;
clear;
s=0;
for i=1:2:99
    s=s+1/i;
end
s
```

例3: 利用for循环编写程序计算

$$S = \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \frac{1}{19}$$

(20以内的质数的倒数之和) {可以很灵活}

```
clc;
clear;
s=0;
for i=[2,3,5,7,11,13,17,19]
    s=s+1/i;
end
s
```

2.while循环

1) 结构体

while循环的一般结构:

```
while (条件)
    循环结构体
end
```

条件是一个逻辑表达式, 当其值为真(true)时, 执行结构体, 直到其值为假(false)时结束

2) 例子

例: 利用while循环编写程序计算 $S = 1 + 2 + 3 + \cdots + n$

求和值S大于100条件下n的最小值.

```
clc;
clear;
s=0;
i=0;
while(s<=100)
    i=i+1;
    s=s+i;
end
```

3.if语句

1) 结构体

1. If语句用法1

```
if 条件
    命令组1
else
    命令组2
end
```

当条件满足时执行命令组1，当条件不满足时执行命令组2

```
clc;
clear;
d=input('请输入分数: ');
if (d>=60)
    out=1;
else
    out=0;
end
out
```

2. If语句用法2

```
if 条件1
    命令组1
elseif 条件2
    命令组2
elseif 条件3
    命令组3
.....
elseif 条件n
    命令组n
else
    命令组n+1
end
```

```
clc;
clear;
d=input('请输入分数: ');
if(d>=90)
    disp('优');
```

```
elseif(d>=80)
    disp('良');
elseif(d>=70)
    disp('中')
elseif(d>=60)
    disp('及格');
else
    disp('不及格');
end
```

3. If语句用法3

```
if 条件
    命令组1
end
```

2) 输入输出

```
d=input('请输入分数: ');
disp('不及格');
```

```
请输入分数: 55
不及格
```

```
>>
```

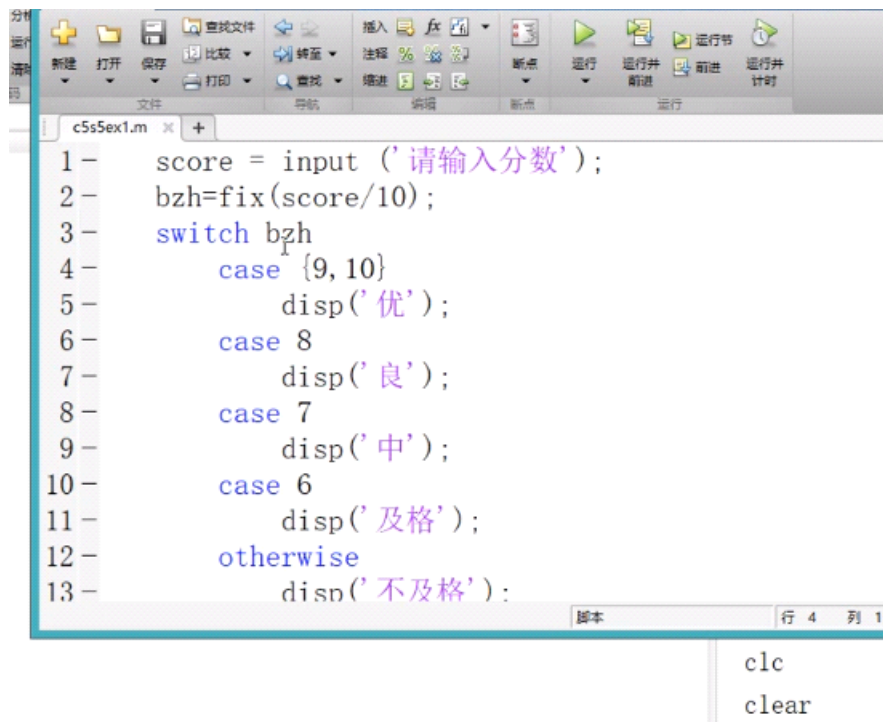
4.switch语句

1) 结构体

switch语句用法

```
switch 表达式
case 表达式1
    语句组1
case 表达式2
    语句组2
    .....
case 表达式n
    语句组n
otherwise
    语句组n+1
end
```

2) 例子



```

clc;
clear;
score=input('请输入分数: ');
bzh=fix(score/10);
switch(bzh)
    case{9,10}
        disp('优');

    case 8
        disp('良');

    case 7
        disp('中');

    case 6
        disp('及格');
    otherwise
        disp('不及格');
end

```

例2：某商场对顾客购买的商品实行打折销售，标准如下
(商品价格用**price**来表示)：

price<200	没有折扣
200<=price<500	%3折扣
500<=price<1000	%5折扣
1000<=price<5000	%8折扣
5000<=price	%10折扣

num2cell(5:9) 向量转变为一个一个的数字{5, 6, 7, 8, 9}

```
clc;
clear;
prize=input('请输入价格: ');
switch fix(prize/100)
    case {0,1}

        rate=0;
    case {2,3,4}
        rate=0.03;

    case num2cell(5:9)
        rate=0.05;

    case num2cell(10,49)
        rate=0.08;
    otherwise
        rate=0.1;
end
prize = prize*(1-rate)
```

5.循环的中断

MATLAB中关于中断执行的命令有: **break/continue/return**

break:跳出循环体, 但只能跳出一层循环, 当有多层循环时只能跳出包含break的最内层循环

continue:终止当次循环中余下的语句, 进行下一次循环

return:所有余下命令均不执行, 终止程序

1) break

```
A=[1,-1,8,-3,9];
n=length(A);
for(i=1:n)
    if(A(i)<0)
        i,A(i)
        break;
    end
end
s=sum(A)
```

```
i = 2
ans = -1
```

```
s = 20
```

2) continue

```

A=[1,-1,8,-3,9];
n=length(A);
for(i=1:n)
    if(A(i)<0)
        i,A(i)
        return;
    end
end
s=sum(A)

```

```

i = 2
ans = -1

```

3) return

```

A=[1,-1,8,-3,9];
n=length(A);
for i=1:n
    if(A(i)<0)
        i,A(i)
    end
    continue;
end
s=sum(A)

```

```

i = 2
ans = -1
i = 4
ans = -3
s = 14

```

例2：判断一个矩阵中是否存在小于0的元素，如果存在，输出该元素及其行标、列标，余下元素无需判断。最后输出该矩阵的转置矩阵。

```

A=[1,-4,8;-3,9,1;-1,-2,3];
[m,n]=size(A);
flag=0;
for i=1:m
    for j=1:n
        if A(i,j)<0
            i,j,A(i,j)
            flag=1;
            break;
        end
    end
    if flag==1
        break;
    end
end
s=A'

```


八、元胞数组

2020年2月21日 21:40

1.定义

1. 元胞数组简介

- (1) 元胞数组是MATLAB的一种特殊数据类型
- (2) 元胞数组可以看作是一种无所不包的通用矩阵，元胞数组中元素称为元胞或者单元，可以是任何一种数据类型的数据
- (3) 元胞数组中的每一个元素可以具有不同的尺寸、可以占用不同的内存空间。
- (4) 元胞数组的内存空间可以动态分配。

2.创建

(1) 直接赋值法

(a)内容索引法:赋值语句的左边用大括号{}将标识单元的下标括起来，右边为单元的内容。

(b)单元索引法:赋值语句的左边用小括号()将标识单元的下标括起来，右边用{}将存储于单元中的数据括起来。

```
>> A{1,1}='Hello,World';  
>> A{1,2}=[1 2 3 4];  
>> A{2,1}=50;  
>> A{2,2}={'你好',1:3};  
>> B(1,1)={'Hello,World'};  
>> B(1,2)=[1 2 3 4];  
>> B(2,1)=50;  
无法从 double 转换为 cell。
```

```
>> B(2,1)={50};  
>> B(2,2)={'你好',1:3};
```

赋值具有的非单一 rhs 维度多于非单一下标数

```
>> B(2,2)={{'你好',1:3}};
```


(2) 利用函数cell创建

```
>> C=cell(2,2);
```

(3) 利用 {} 直接创建元胞数组的所有单元

```
>> D={'Hello,World',[1 2 3 4];50,{'你好',1:3}};
```

3.元胞数组的访问

元胞数组的寻访有以下三种形式：

(1) 用 () 将下标括起来访问其单元的大小和类型

(2) 用 {} 将下标括起来寻访其单元的内容

(3) 用 celldisp () 函数对元胞数组的所有内容进行寻访

```
>> D(1,2)
```

```
ans =
```

```
1×1 cell 数组
```

```
{1×4 double}
```

```
>> D{1,2}
```

```
ans =
```

```
1    2    3    4
```

```
>> celldisp(D)
```

```
D{1,1} =
```

```
Hello,World
```

```
D{2,1} =
```

```
50
```

```
D{1,2} =
```

```
1    2    3    4
```

D{2,2}{1} =

你好

D{2,2}{2} =

1 2 3

4.删除

1) 删除列/行

```
>> A(:,1)=[]
```

2) 删除全部

```
>> A(:)=[]
```

5.相关的函数

5. 元胞数组相关函数

celldisp: 显示元胞数组中所有的内容

cell: 创建空的元胞数组

cellplot: 利用图形方式显示内容

cell2mat: 将数组转变成为普通的矩阵

mat2cell: 将数值矩阵转变成为cell数组

num2cell: 将数值数组转变成为cell数组

iscell: 判断输入是否为cell数组

九、Matlab读写excel

2020年2月21日 22:06

1.读取

1. 从Excel中读取数据到MATLAB

标准命令格式 `[num,txt,row] = xlsread(filename,sheet,xlRange)`

filename:文件的名称,用单引号引起来,例如: 'Data.xlsx'

sheet:工作表的名称或者编号,例如:'Sheet1',或者是数字1

xlRange:工作表中的指定区域,例如: 'A2:D5'

num:存储工作表中的数值数据,为一个矩阵

txt:存储工作表中的文本数据,为一个元胞数组

row:存储工作表中的数值数据和文本数据,为一个元胞数组

```
[num,txt,row]=xlsread('Data.xlsx','Sheet1','A1:F21');
```

txt和row可以缺省, sheet和xlRange可以缺省

```
>> [num,txt,row]=xlsread('内部冲突指数.xlsx',1,'B8:F14');
```

2.写入

2. 将MATLAB中的数据写入到Excel

标准命令格式

```
xlswrite(filename,A,sheet,xlRange)
```

filename:文件的名称,用单引号引起来,需要带上扩展名(xls或者xlsx),例如: 'Data.xlsx'

A:可以是MATLAB中的矩阵、字符数组或者元胞数组

sheet:工作表的名称或者编号,例如: 'Sheet1', 或者是数字1

xlRange:工作表中的指定区域,例如: 'A2:D5', 注意区域的大小要与A的维数一致

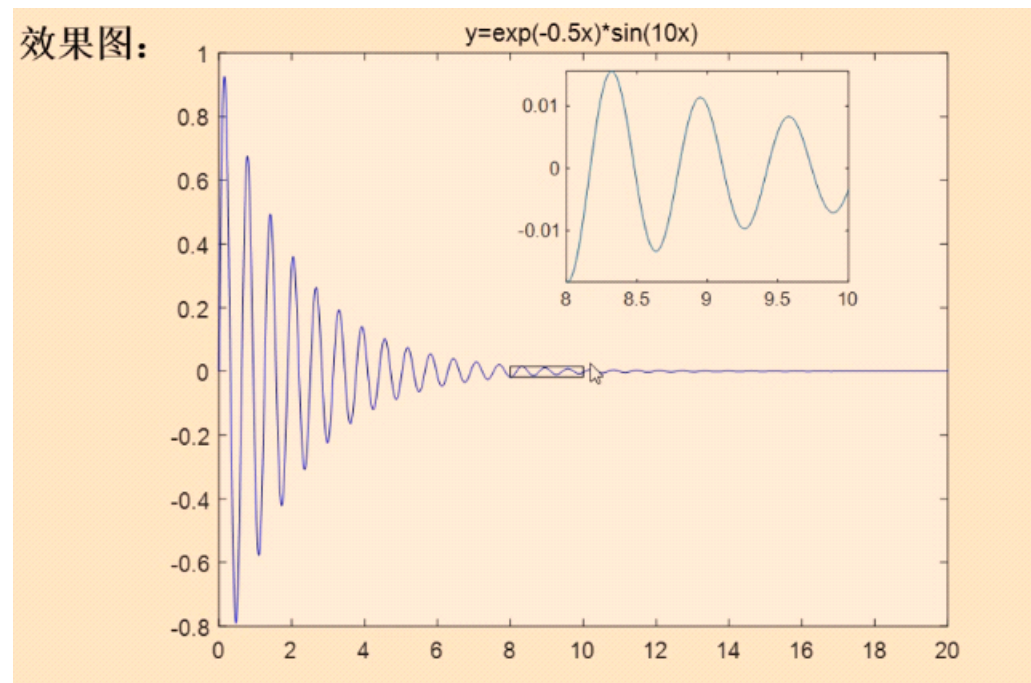
注意事项:

写入Excel时,被写入的Excel文件不能打开,否则会出现错误提示“不可写。它可能被其他进程锁定。”

十、画中画

2020年2月22日 21:19

1.效果图:



2.实现

1) `axes('Position',[0.5 0.6 0.3 0.3]);`画小图

四个参数: 0~1之间

第一个参数到y轴的距离。

第二个参数到x轴的距离。

三四参数是新图的宽和高。

2) `rectangle('Position',[a b c d])`画矩形

(a,b) 是左下角的坐标

c, d是宽度和长度

3)

```
clc;
clear;
% figure;
x=0:0.05:20;
y=exp(-0.5*x).*sin(10*x);
% plot(x,y);
x1=8:0.05:10;
y1=exp(-0.5*x1).*sin(10*x1);
plot(x,y)
axes('Position',[0.5 0.6 0.3 0.3]);
plot(x1,y1)
```

