

Matlab





MATLAB启动图标:



1. MATLAB的主窗口：包括10个按钮、六个下拉菜单。（版本不同会导致界面略有不同）其它的几个窗口都包含在这个大的主窗口中。

2.命令窗口

“>>”为运算提示符，表示MATLAB正处在准备状态。当在提示符后输入一段运算式并按回车键后，MATLAB将给出计算结果，然后再进入准备状态。

3.历史窗口

在默认设置下，历史窗口中会保留自安装起所有命令的历史记录，并标明使用时间，这方便了使用者的查询。双击某一行命令，即在命令窗口中执行该行命令。



4. 当前目录窗口

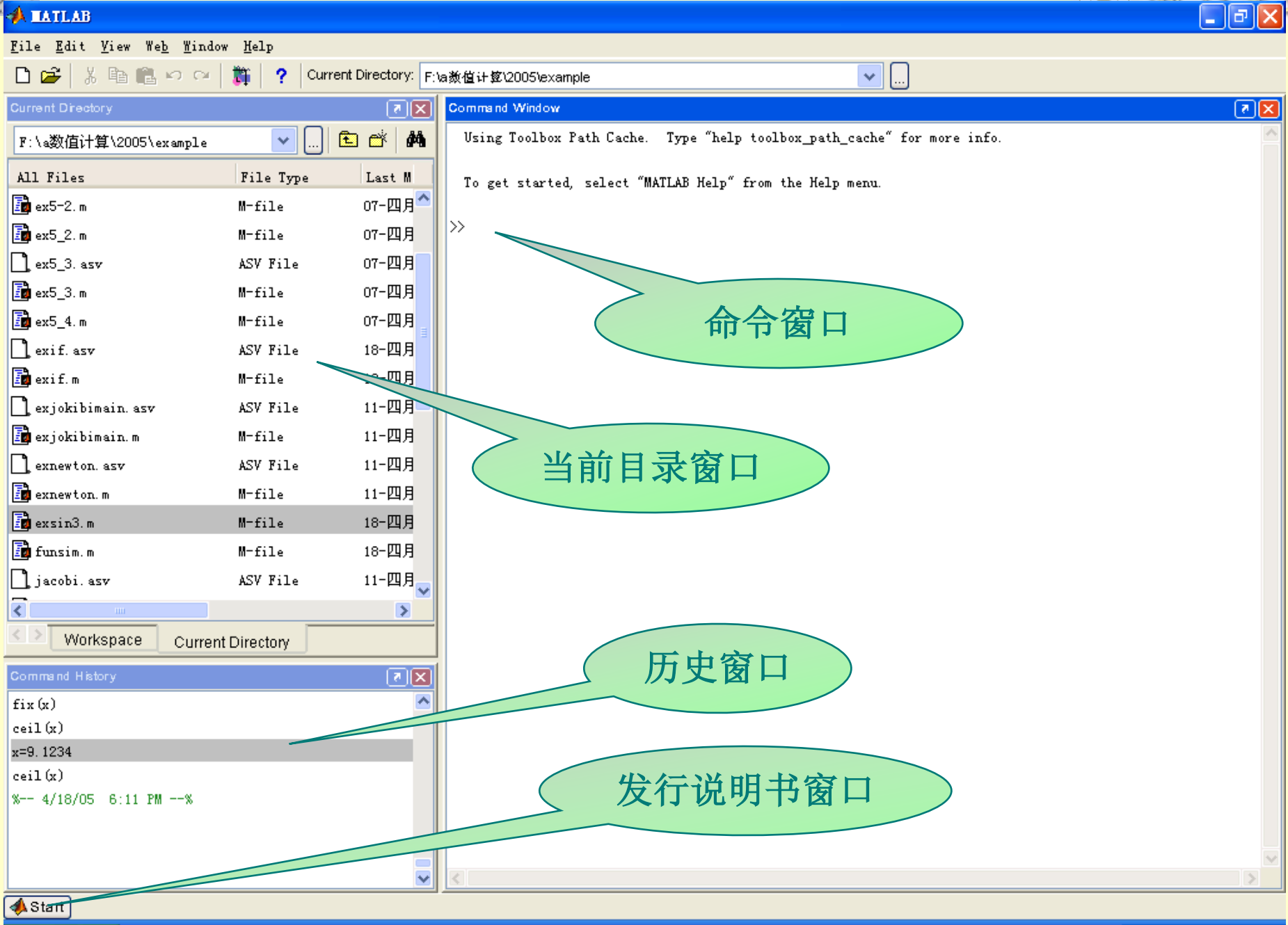
在当前目录窗口中可显示或改变当前目录，还可以显示当前目录下的文件。

5. 发行说明书窗口

用来说明用户所拥有的产品的工具包、演示及帮助信息。

6. 工作间管理窗口

在工作间管理窗口中将显示目前内存中所有的MATLAB变量的变量名、字节数以及类型等信息。





MATLAB

File Edit View Web Window Help

Current Directory: F:\a数值计算\2005\example

Workspace

Name	Size	Bytes	Class
r	73x73	42632	double array
u	73x73	42632	double array
v	73x73	42632	double array
x	1x73	584	double array
y	1x73	584	double array
z	73x73	42632	double array

Stack: Base

Command Window

Using Toolbox Path Cache. Type "help toolbox_path_cache" for more info.

To get started, select "MATLAB Help" from the Help menu.

>>

Command History

```
fix(x)
ceil(x)
x=9.1234
ceil(x)
%-- 4/18/05 6:11 PM --%
```

Workspace Current Directory

Start

开始 2005 MATLAB MATLAB简介.ppt 18:57

工作间管理窗口



MATLAB程序设计语言

1 基本赋值和计算

MATLAB的基本控制都是通过命令窗口实现的。给出一条命令，立即就可以得出该命令的结果。MATLAB能被当作计算器使用。

“>>”是MATLAB的提示符，在提示符后键入一个命令，当按下回车后，就执行该命令。

```
>> 5011+13
```

```
ans =
```

```
5024
```

```
>>
```



同一行上可以有多个命令：

```
>> 2^5,2*(3+2)
```

```
ans =
```

```
32
```

```
ans =
```

```
10
```

```
>>
```

```
>> x=14
```

```
x =
```

```
14
```

```
>> y=3*x
```

```
y =
```

```
42
```

```
>>
```

变量通常用于保存所赋的值和结果。如果没有赋值，**MATLAB**将结果存放在名为`ans`的变量中。



```
>> a=exp(1.2); b=sin(2);  
  
>> c=a*b  
  
c =  
  
    3.0190
```

```
>> a=[1 2;3 4]; b=[5 6;7 8];  
  
>> c=a*b  
  
c =  
  
    19    22  
    43    50
```

代数运算符

- ❖ + 加法
- ❖ - 减法
- ❖ - 负
- ❖ * 乘法
- ❖ ^ 幂
- ❖ / 右除（正常除） $3/4=3\div 4$
- ❖ \ 左除 $3\backslash 4=4\div 3$ 少用，矩阵运算常用



所有的基本数学函数在MATLAB中有定义
注意**MATLAB**的内部函数一定是小写字母。

■ 常用的基本数学函数：

abs(x), 求x的绝对值

sign(x), 求x的符号，如果是正的得1；负的得-1；零得0

sqrt(x), 求x的平方根

exp(x), 求x的指数函数，即 e^x

log(x), 求x的自然对数，即 $\ln x$

log10(x), 求x以10为底的对数

log2(x), 求x以2为底的对数



■ 常用的三角函数:

$\sin(x)$, $\cos(x)$, $\tan(x)$,

$\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$,

$\sinh(x)$, $\cosh(x)$, $\tanh(x)$,

$\text{asinh}(x)$, $\text{acosh}(x)$, $\text{atanh}(x)$ 。



■取整命令和有关命令

round(x), 求最接近 x 的整数。如果 x 是一个向量, 则适用于所有元素。

fix(x), 求0方向最接近 x 的函数。即负 x 向上取整, 正 x 向下取整。

floor(x), 求小于或等于 x 的最接近的整数

ceil(x), 求大于或等于 x 的最接近的整数

rem(x, y), 求整除 x/y 的余数

gcd(x, y), 求整数 x 和 y 的最大公因子

lcm(x, y), 求整数 x 和 y 的最小公倍数



■ 有关复数的函数

angle(z) , 求 z 的相角
abs(z), 求 z 的模
real(z) , 求 z 的实部
imag(z) , 求 z 的虚部
conj(z) , 求 z 的共轭复数



几个格式指令和常量

特殊变量	
ans	用于结果的缺省变量名
pi	圆周率
Inf	无穷大，如1/0
NaN	不定量，如0/0
i和j	$i=j=(-1)^{1/2}$

变量名规则：

1. 必须是不含有空格的单个词
2. 变量名区分字母大小写
3. 变量名最多不超过19个字符
4. 变量名必须以字母打头，之后可以是任意字母、数字或下划线
5. 变量名中不允许使用标点符号

表中的特殊变量在启动MATLAB之后，自动赋予表中取值。如果定义了相同名字的变量，原始特殊取值将会丢失，直至清除所有变量或重新启动MATLAB。一般来讲，应当尽量避免重新定义特殊变量。

```
>>clc all    清屏  
>>clear      删除工作变量
```

clear命令

clear命令可以用来删除一些不再使用的变量。这样可以使得整个工作空间更简洁。例如**clear x1 y1**将删除x1和y1变量。但应当注意，在这一命令下x1和y1之间**不能加逗号（clear x1, y1）**，否则该命令就会被错误地解释成删除x1变量，然后开始下一个语句（其内容为y1），而该语句将被解释成将y1变量的内容显示出来，这样y1变量就不被删除了。

如果想删除整个工作空间中所有的变量，则可以使用clear命令，在该命令后面不用加任何参数。

应当特别注意：使用clear命令，MATLAB工作空间中的全部变量将被无条件删除！系统不会要求你确认这个命令。所有变量都被清除，且不能恢复！

2 用MATLAB处理矩阵

矩阵在MATLAB中是基本的数据单元， MATLAB中矩阵变量名必须是以字母开头的，由字母、数字、下划线组成的字符串。

2.1 形成矩阵 在MATLAB中形成矩阵的办法有多种。生成小型矩阵的常用办法是直接由键盘输入。

```
>> A=[1 3 2;3 1 0;2 1 5]
```

```
A =
```

```
     1     3     2
     3     1     0
     2     1     5
```

```
>>
```

变量A赋值为一个3阶方阵。

1. 矩阵表示原则：

- (1) 矩阵元素列在[]中。
- (2) 每行元素之间用空格或逗号隔开。
- (3) 行与行之间用分号隔开，或在输入时用回车键代替分号。

```
>> B=[4 3 6;5 1 4];  
>>
```

变量**B**赋值为一个**2*3**阶矩阵，但结果并不输出在命令窗口上。

```
>> size(B)  
  
ans =  
  
    2    3  
>> c=size(B)  
  
c =  
  
    2    3  
  
>>
```

```
>>  
length(B)  
  
ans =  
  
    3  
  
>>  
  
max(size(B))
```

创建矩阵的其他方法：

(1) 通过外部数据文件加载

通过“**load**”命令加载
外部数据文件创建矩阵。

(2) 在**M**文件中创建矩阵

函数**size(B)**给出矩阵**B** 的维数。



2.2 特殊矩阵 一些有特殊意义和结构的矩阵，在**MATLAB**中可以很容易得到它们。最常用的特殊矩阵命令有

ones(m,n) 形成一个 $m \times n$ 阶矩阵，它的所有元素均为“1”；
zeros(m,n) 形成一个 $m \times n$ 阶零矩阵；
eye(n) 形成一个 n 阶单位矩阵。

```
>> c=ones(2,3)
```

```
c =
```

```
    1    1    1  
    1    1    1
```

```
>>
```

```
>> c=zeros(size(B))
```

```
c =
```

```
    0    0    0  
    0    0    0
```

```
>>
```

```
>> c=eye(3)
```

```
c =
```

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

```
>>
```

2.3 矩阵的运算 讨论矩阵的算术运算、分块、转置和求逆。假设A和B为前面输入的矩阵

```
>> A=[1 3 2;3 1 0;2 1 5];  
>> B=[4 3 6;5 1 4;3 4 6];  
>> C=A+B
```

C =

5	6	8
8	2	4
5	5	11

```
>>
```

```
>> D=A-B
```

D=

-3	0	-4
-2	0	-4
-1	-3	-1

```
>> E=A*B
```

E =

25	14	30
17	10	22
28	27	46

```
>>
```



```
>> F=A'
```

```
F =
```

```
1 3 2
3 1 1
2 0 5
```

转置

```
>> G=inv(A)
```

```
G =
```

```
-0.1316  0.3421  0.0526
 0.3947 -0.0263 -0.1579
-0.0263 -0.1316  0.2105
```

求逆

```
>>
```

```
A =
```

```
1 3 2
3 1 0
2 1 5
```

在矩阵运算中的“除法”是必须特殊考虑和研究的，设
 $Ax = b$

其中 A 为方阵， x 为所要求解的向量， b 为已知的向量。上述的问题称为线性方程组。使用**MATLAB**的一个语句就可以得到该问题的解。

```
>> b=ones(3,1);  
>> x=A\b  
  
x =  
  
    0.2632  
    0.2105  
    0.0526  
  
>>
```

$A =$

1	3	2
3	1	0
2	1	5



在**MATLAB**这个平台上，可以很方便地进行矩阵元素的运算，
例如

```
>> U(1,2)=A(2,1)+A(3,2)
```

U =

0 4

```
>> V(3,1)=B(2,2)*A(1,2)
```

V =

0
0
3

```
>>
```

A =

1	3	2
2	1	0
2	1	5

B =

4	3	6
5	1	4
3	4	6

```
>> A(1,3)=5;
```

```
>> A(2,2)=4;
```

```
>> A
```

A =

1	3	5
3	4	0
2	1	5

```
>>
```

矩阵的分块也很容易在MATLAB这个平台上进行，例如， $A(2,:)$ 和 $A(:,3)$ 分别表示矩阵A的第2行和第3列，而且上述符号可以作为向量直接参与运算。

```
>> A(2,:)
```

```
ans =
```

```
3    4    0
```

```
>> A(:,3)
```

```
ans =
```

```
5
```

```
0
```

```
5
```

```
>>
```

```
A =
```

```
1    3    5
```

```
3    4    0
```

```
2    1    5
```

设H是一个 10×10 矩阵，则
 $H(:,4:8)$ 表示的是H的第4到第8
列组成的 10×5 矩阵。



```
>> a=[ 1 3 5 7;2 4 6 8;-2 -3 -4 -5;-4 -5 -6 -7]
```

a =

1	3	5	7
2	4	6	8
-2	-3	-4	-5
-4	-5	-6	-7

```
>> v=[1 3 4];
```

```
>> b=a(v,4)
```

b =

7
-5
-7

```
>>
```

```
>> b=a(v,:)
```

b =

1	3	5	7
-2	-3	-4	-5
-4	-5	-6	-7

```
>>
```

在MATLAB中 “:”是十分常用的。例如：

```
>> x=-10:2:10
```

```
x =
```

```
-10 -8 -6 -4 -2 0 2 4 6 8 10
```

```
>> z=-10:2:11?
```

得到x为11维的
向量

```
>> y=0:0.1:2;
```

```
>> size(y)
```

```
ans =
```

```
1 21
```

```
>>
```

得到y为21维的向量，元素为
0,0.1,0.2,...,1.9,2.0



```
>> z=[1:3,2:2:8,6:3:15]
```

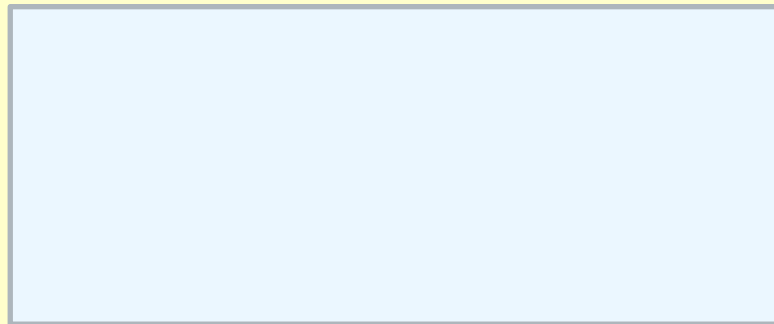
```
z =
```

```
1 2 3 2 4 6 8 6 9 12 15
```

```
>> c=[3.2,4.5;2.4,4.7];
```

```
>> d=[c ones(size(c));zeros(size(c)) eye(size(c))]?
```

```
d =
```



```
>>
```

2.4 矩阵的特殊运算

这里主要介绍矩阵的乘幂运算和矩阵元素的运算。

A^p 得到与A同阶的矩阵为A的p次幂，其中p可以是任何的正数。

```
>> A^2
```

```
ans =
```

```
    20    20    30  
    15    25    15  
    15    15    35
```

```
>>
```

```
A =
```

```
     1     3     5  
     3     4     0  
     2     1     5
```



假设A和B是同阶的矩阵（或向量），

A =

1	3	5
3	4	0
2	1	5

B =

4	3	6
5	1	4
3	4	6

$p=A.*B$ 得到与A和B同阶的矩阵，其中
 $p(i,j)=A(i,j)*B(i,j)$ 类比加减运算

```
>> p=A.*B
```

```
p =
```

4	9	30
15	4	0
6	4	30

```
>>
```

$q = A.^B$ 得到与A与B同阶的矩阵，其中
 $q(i,j) = A(i,j)^{B(i,j)}$

A =

1	3	5
3	4	0
2	1	5

B =

4	3	6
5	1	4
3	4	6

```
>> q=A.^B
```

q =

1	27	15625
243	4	0
8	1	15625

```
>>
```



$r = A ./ B$ 得到与A与B同阶的矩阵，其中
 $r(i, j) = A(i, j) / B(i, j)$

A =

1	3	5
3	4	0
2	1	5

B =

4	3	6
5	1	4
3	4	6

```
>> r=A./B
```

r =

0.2500	1.0000	0.8333
0.6000	4.0000	0
0.6667	0.2500	0.8333

```
>>
```



$s = A \setminus B$ 得到与A与B同阶的矩阵，其中
 $s(i, j) = B(i, j) / A(i, j)$

A =

1	3	5
3	4	1
2	1	5

B =

4	3	6
5	1	4
3	4	6

```
>> s=A.\B
```

```
s =
```

4.0000	1.0000	1.2000
1.6667	0.2500	4.0000
1.5000	4.0000	1.2000

```
>>
```



:	冒号；在MATLAB中，它表示“全部”
;	用于分隔行；禁止显示结果
,	用于分隔列；要求显示结果
.	小数点
'	矩阵或向量转置
()	指出在算术表达式中计算的先后次序,矩阵元素提取
[]	用于构成向量和矩阵
%	用于注释
=	用于赋值



3 用MATLAB绘图——容易

MATLAB提供了许多可以选用的图形功能，这里只作简单介绍

3.1 二维图形函数plot 最常用和最简单的绘图命令 例如

```
plot(x,y,'-');
```

将向量x和y对应元素定义的点依次用实线联接(x和y的维数必须一样)；如果x和y为矩阵，则按列依次处理

```
plot(x1,y1,'*',x2,y2,'+');
```

将向量 x1和y1对应元素定义的点用星号标出，将向量x2和y2对应元素定义的点用 '+' 标出。MATLAB可以划线或者点，它提供的点和线类型如下表



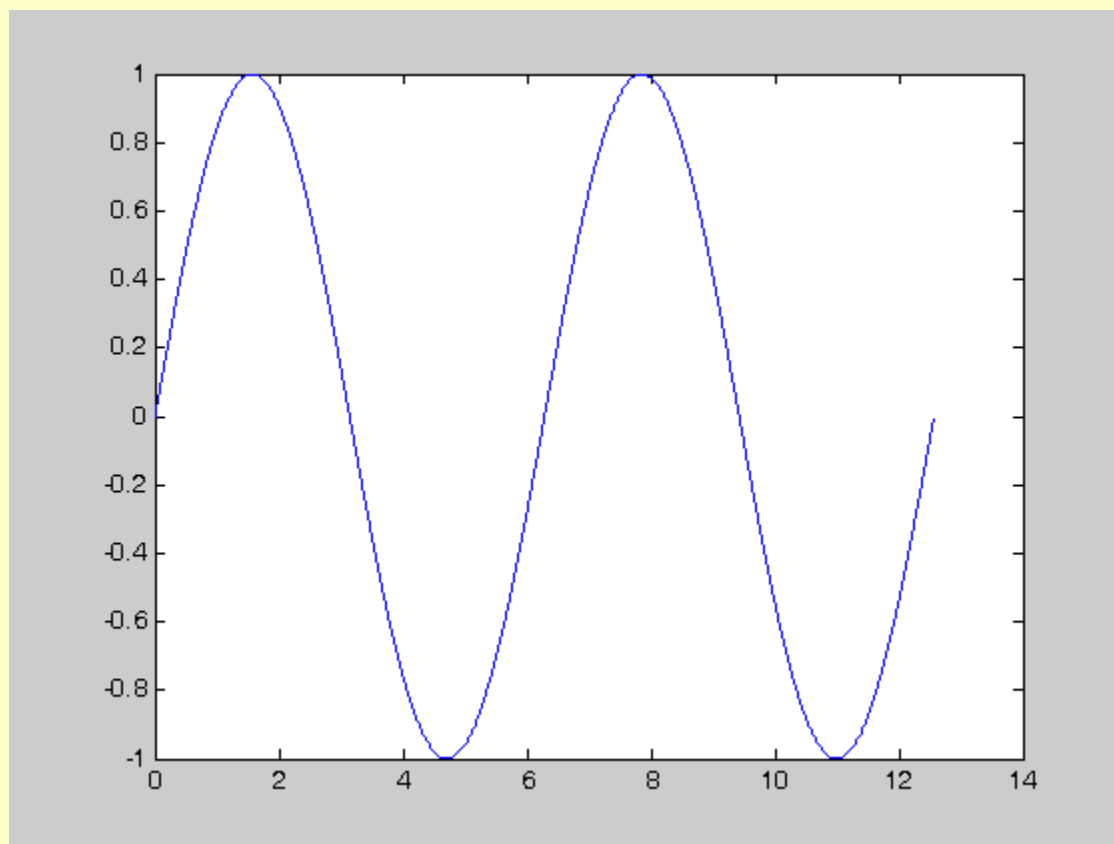
线	符号	点	符号
实线	-	实心圆点	.
虚线	--	加号	+
点	:	星号	*
虚线间点	-.	空心原点	。
		叉号	×

蓝色	b	黄色	y
红色	r	绿色	g



例如 `plot(x,y, '-b')` 将x和y对应的元素定义的点依次用蓝色实线联接。

```
>> x = 0:.002:4*pi;  
>> y=sin(x);  
>> plot(x,y,'-b');
```





3.2 绘图辅助函数

利用这一族函数可以为画出的图像加上标题等内容。

title('...'); 在图形的上方显示‘ ’中指定的内容；

xlabel('...'); 将‘ ’中指定的内容标在**x**轴；

ylabel('...'); 将‘ ’中指定的内容标在**y**轴；

grid; 在图上显示虚线的格；

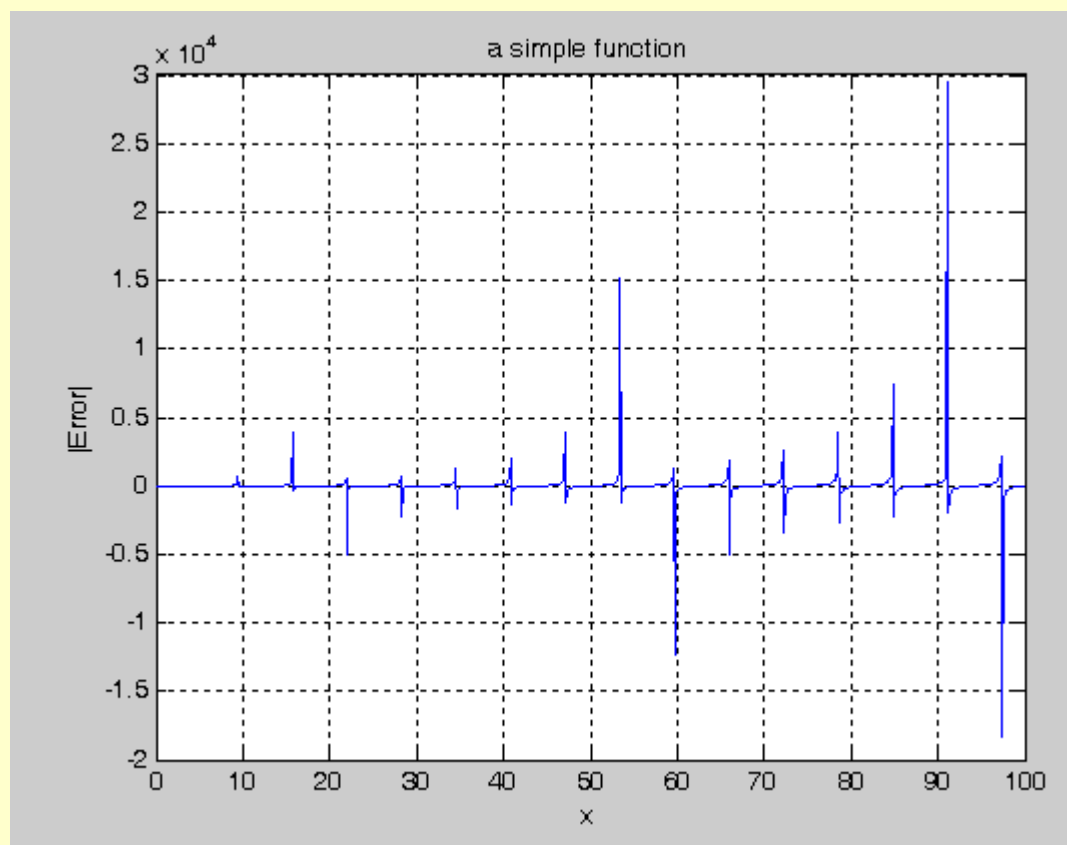
text(x,y, '...'); 将‘ ’中指定的内容显示在**x,y**所定义的位置上；

gtext('...'); 运行到该命令时，屏幕光标位置显示符号“+”等待，它将‘ ’中指定的内容标在鼠标指定的位置；



`axis([xl xr yl yr]);` 其中的4个实数分别定义x和y方向显示的范围;
`hold on;` 后面`plot`的图将迭在一起;
`hold off;` 解除`hold on`命令, **plot**将冲去图形窗口已有图形;
注意上述辅助函数必须放在相应的“`plot`”语句之后。

```
>> x = 0:0.1:100;  
>> y = sin(x).*x./(1+cos(x));  
>> plot(x,y, 'b')  
>> grid;  
>> axis([0 100 -2e4 3e4]);  
>> title('a simple function');  
>> xlabel('x');  
>> ylabel('|Error|');
```





3.3 多窗口绘图函数subplot

该函数的形式 `subplot(p,q,r)`该命令将图形窗口分成 p 行 q 列共计 $p \times q$ 个格子，在第 r 个格子上画图，格子是从上到下按行依次记数的。

例如考虑Chebeshev多项式，它可以用其递推公式定义如下：

$$t_0(x) = 1, t_1(x) = x,$$

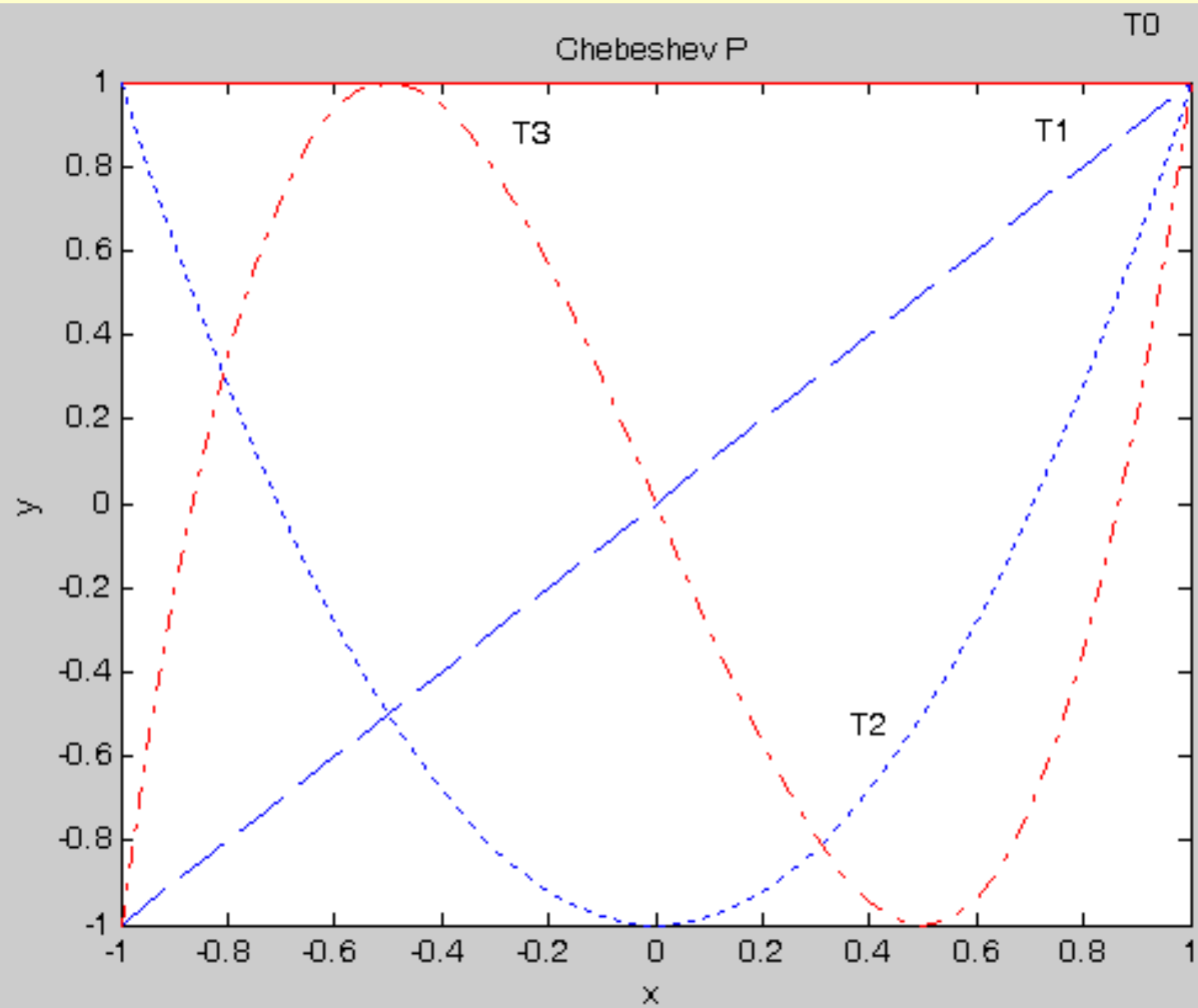
$$t_k(x) = 2xt_{k-1}(x) - t_{k-2}(x), k = 2, 3, \dots$$



将 $[-1, 1]$ 上的前四个Chebeshev多项式画在一张图上

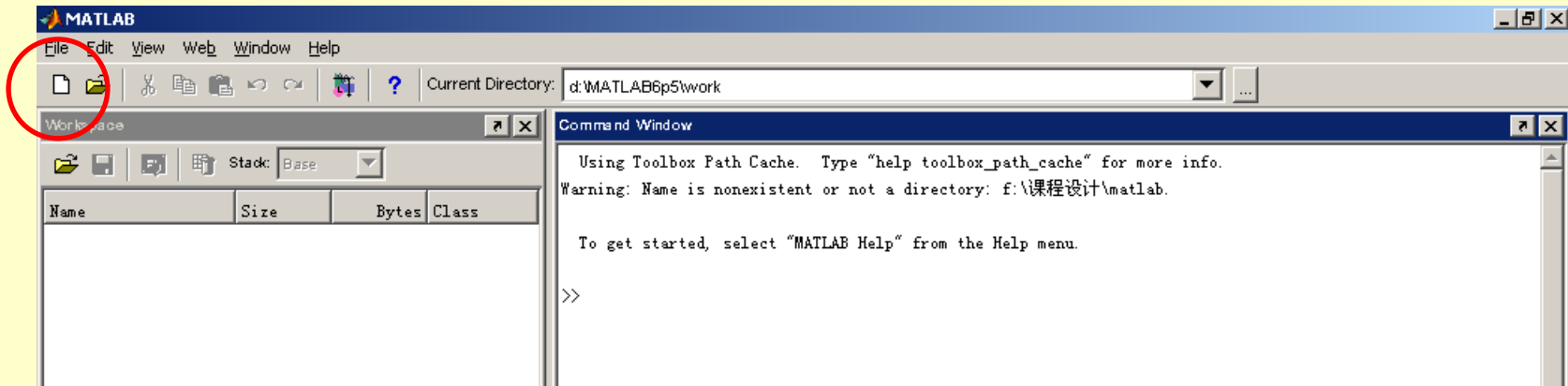
```
x=-1:0.05:1;  
t0=1.0+0*x;  
t1=x;  
t2=2*x.*t1-t0;  
t3=2*x.*t2-t1;  
plot(x,t0,'-r');gtext('T0');  
title('Chebeshev P');  
xlabel('x');  
ylabel('y');  
hold on  
plot(x,t1,'--b');gtext('T1');  
plot(x,t2,':b');gtext('T2');  
plot(x,t3,'-.r');gtext('T3');  
hold off
```

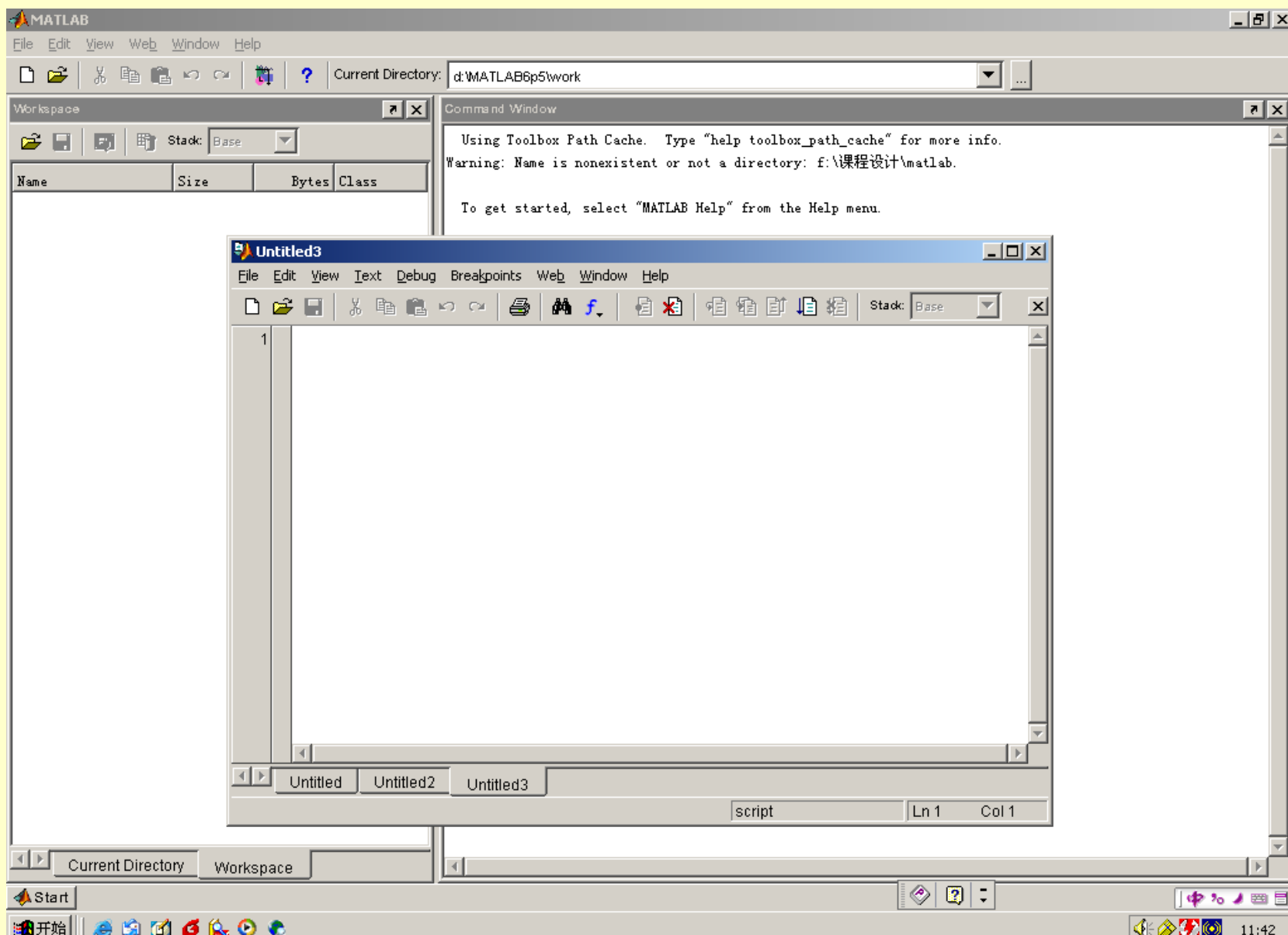


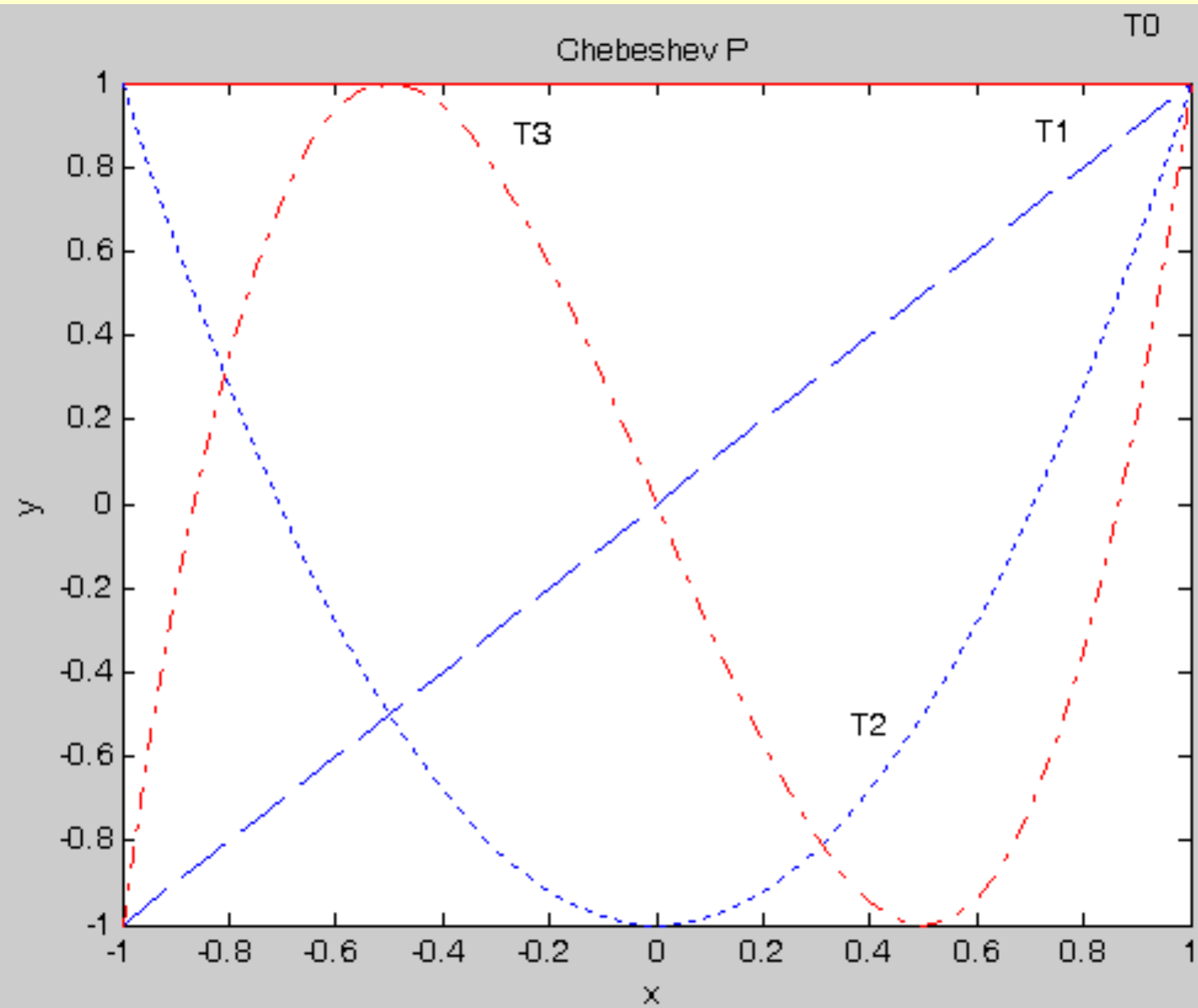




将一个完整的命令集合写入M文件便是一段MATLAB程序，但要注意，编程是在MATLAB的编辑窗口而不是命令窗口。M文件必须有后缀.m（filename.m）。每当用户输入文件名，这些命令就由MATLAB执行。



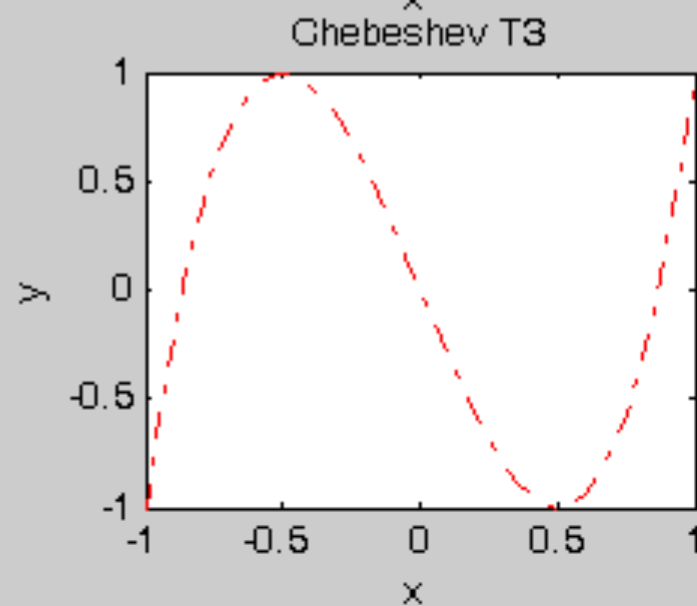
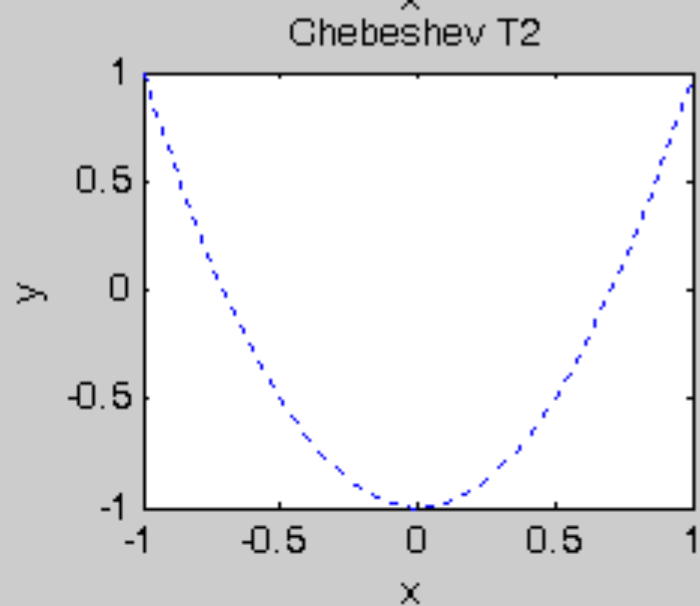
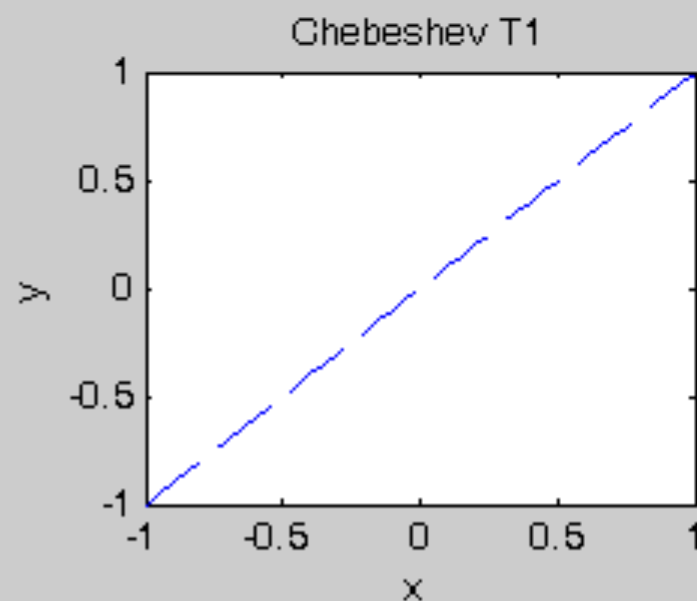
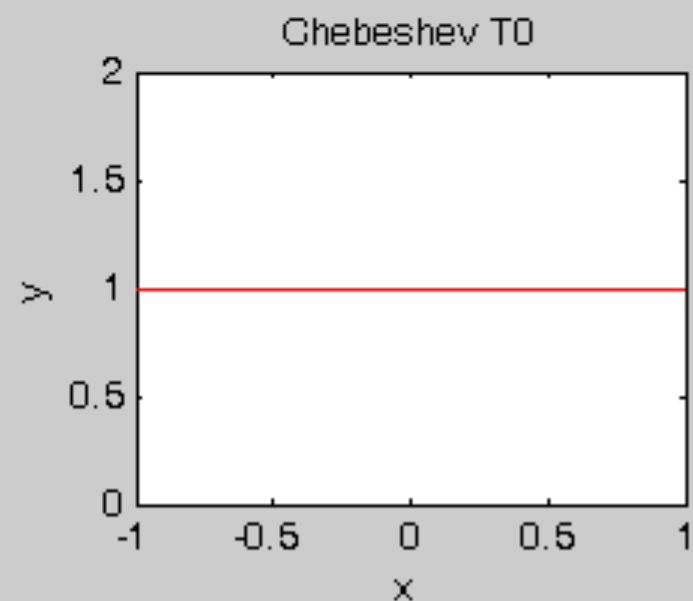






使用subplot则程序

```
x=-1:0.05:1;  
t0=1.0+0*x;t1=x;  
t2=2*x.*t1-t0;t3=2*x.*t2-t1;  
subplot(2,2,1);plot(x,t0,'-r');  
title('Chebeshev T0');xlabel('x');ylabel('y');  
  
subplot(2,2,2);plot(x,t1,'--b');title('Chebeshev  
T1');  
xlabel('x');ylabel('y');  
  
subplot(2,2,3);plot(x,t2,':b');  
title('Chebeshev T2');xlabel('x');ylabel('y');  
  
subplot(2,2,4);plot(x,t3,'-.r');  
title('Chebeshev T3');xlabel('x');ylabel('y');
```





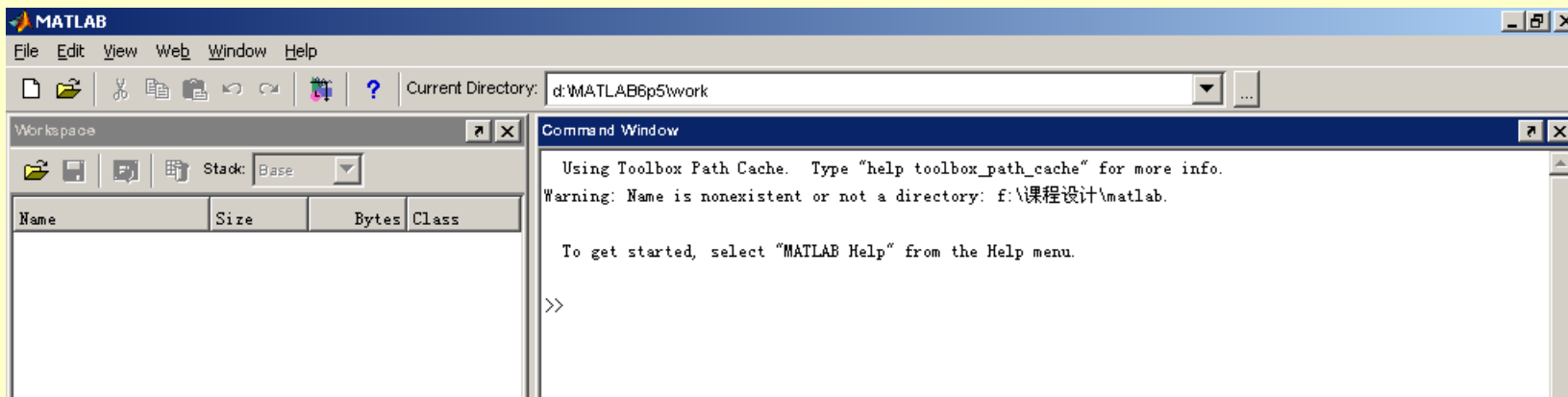
4 用MATLAB编程

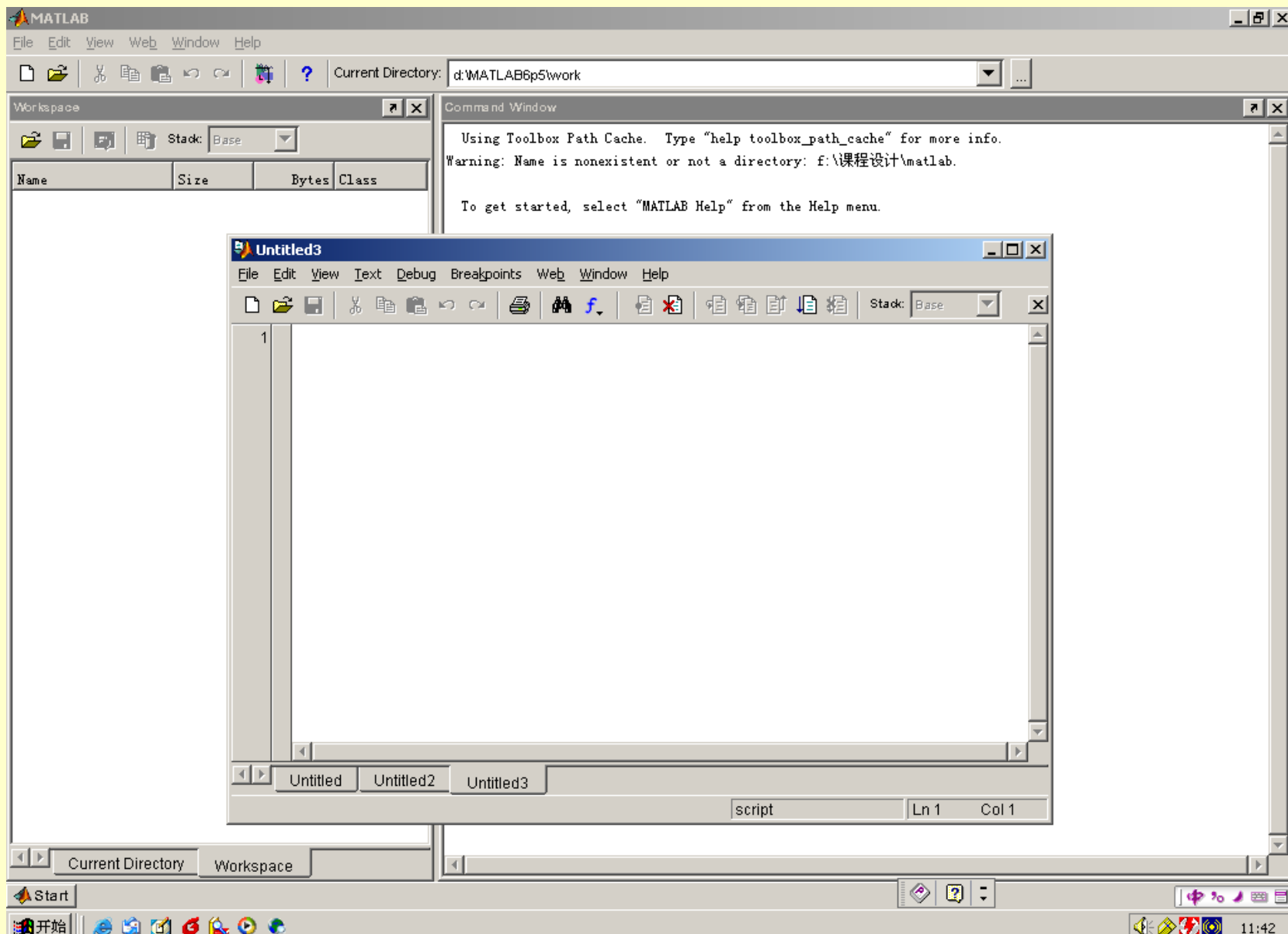
将一个完整的命令集合写入M文件便是一段MATLAB程序，但要注意，编程是在MATLAB的编辑窗口而不是命令窗口。M文件必须有后缀.m（filename.m）。每当用户输入文件名，这些命令就由MATLAB执行。

M文件的两种形式：

命令/脚本(script)文件：命令的简单叠加

函数 (function)文件：进行参数传递和函数调用，第一句以function语句为引导。







MATLAB 7.4.0 (R2007a)

File Edit View Debug Desktop Window Help

Current Directory: D:\a-数值计算\参考程序\416

Current Directory - D:\a-数值计算\参考程序\416

All Files	Type	Size	Date Modified
gaussj.m	M-file	3 KB	06-4-3 下午3:36
Leastsquare.m	M-file	2 KB	06-4-3 下午3:36
output.dat	DAT File	3 KB	07-3-23 上午11:
Untitled.asv	Editor Autosave	1 KB	06-4-3 下午3:46
Untitled.m	M-file	1 KB	06-4-3 下午3:47

Command Window

To get started, select MATLAB Help or Demos from the Help menu.

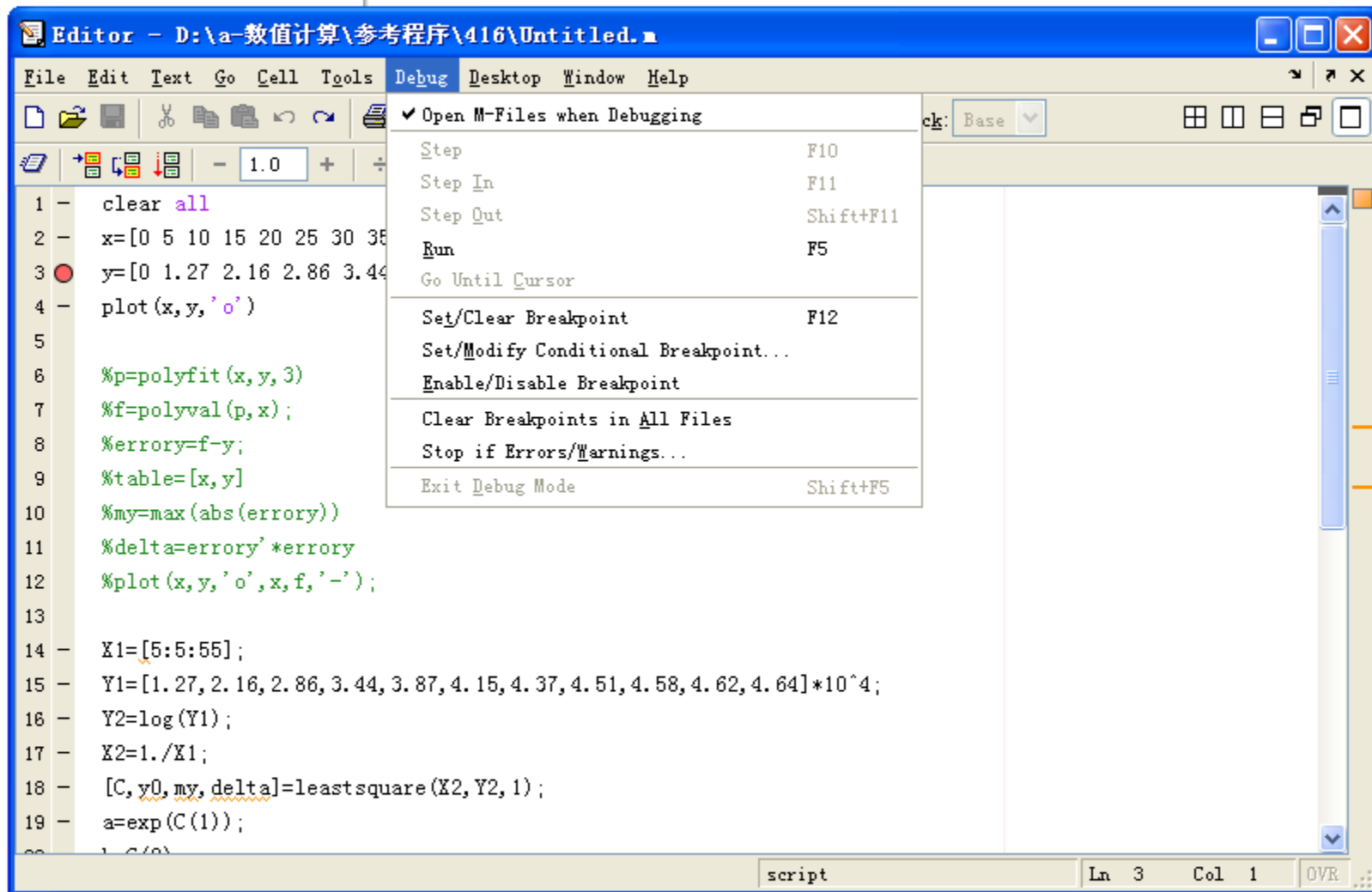
Editor - D:\a-数值计算\参考程序\416\Untitled.m

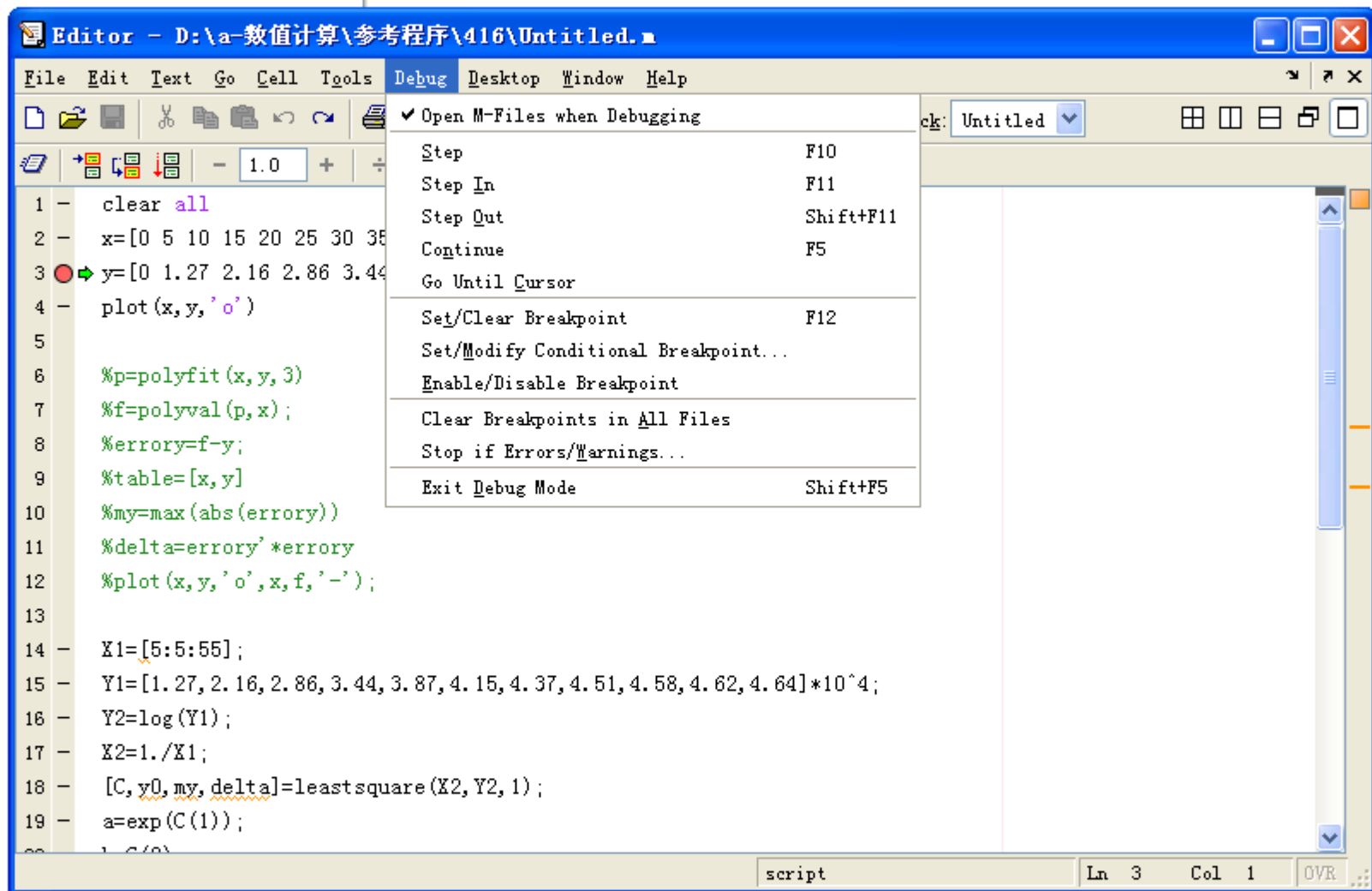
File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - clear all
2 - x=[0 5 10 15 20 25 30 35 40 45 50 55];
3 - y=[0 1.27 2.16 2.86 3.44 3.87 4.15 4.37 4.51 4.58 4.62 4.64].*10^4;
4 - plot(x,y,'o')
5
6 - %p=polyfit(x,y,3)
7 - %f=polyval(p,x);
8 - %error=f-y;
9 - %table=[x,y]
10 - %my=max(abs(error))
11 - %delta=error'*error
12 - %plot(x,y,'o',x,f,'-');
13
14 - X1=[5:5:55];
15 - Y1=[1.27,2.16,2.86,3.44,3.87,4.15,4.37,4.51,4.58,4.62,4.64]*10^4;
16 - Y2=log(Y1);
17 - X2=1./X1;
18 - [C,y0,my,delta]=leastsquare(X2,Y2,1);
19 - a=exp(C(1));
20 - %C(2)
```

script Ln 3 Col 1 OVR







MATLAB还提供了一般程序语言的基本功能。

4.1 for循环语句

```
n=2;  
m=3;  
for i=1:n  
    for j=1:m  
        A(i,j)=sin((i+j)/(m+n));  
    end  
end
```

A =

0.389418342308651	0.564642473395035	0.717356090899523
0.564642473395035	0.717356090899523	0.841470984807897



循环中的步长是可以选择的。如

```
for i=n:-2:n/2  
    A(i)=sin(i+n);  
end
```

即循环控制变量从 n 开始，步长是-2到 $n/2$ 结束。

创建矩阵

$$A = \begin{pmatrix} 5 & 1 & 0 & 0 & 0 \\ 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 5 & 1 & 0 \\ 0 & 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 1 & 5 \end{pmatrix}$$



$$A = \begin{pmatrix} 5 & 1 & 0 & 0 & 0 \\ 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 5 & 1 & 0 \\ 0 & 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 1 & 5 \end{pmatrix}$$

```
A=zeros(5);
```

```
for k=1:4
```

```
    A(k,k)=5;
```

```
    A(k,k+1)=1;
```

```
    A(k+1,k)=1;
```

```
end
```

```
    A(5,5)=5;
```





4.2 while循环语句

与计算机的其他高级语言一样，**while**循环语句要由关系运算和逻辑运算给出的逻辑值控制，该语句的一般形式为

```
while 逻辑表达式  
    一族可执行语句  
end
```

MATLAB中的关系运算有

==	相等	~=	不等
<	小于	<=	小于等于
>	大于	>=	大于等于

MATLAB中的逻辑运算有

&	与		或	~	非
---	---	--	---	---	---



有时在循环正常结束前终止循环是有用的，这可以用命令**break**来实现。如果**break**命令用于嵌套循环的内循环，那么只能终止内部循环，外部循环仍然继续。

```
sum=0;
for m=1:100
    if(sum>100)
        m        %当sum>100时，显示m
        break;   %中止运算
    end
    sum=sum+m;
end
```



4.3 if条件语句

```
if logical expression  
    statements  
end
```

```
if logical expression  
    statements 1  
else  
    statements2  
end
```

```
if logical expression1  
    statements 1  
elseif logical expression2  
    statements2  
end
```

```
if logical expression1  
    statements 1  
elseif logical expression2  
    statements2  
else  
    statements3  
end
```



创建矩阵

$$A = \begin{pmatrix} 5 & 1 & 0 & 0 & 0 \\ 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 5 & 1 & 0 \\ 0 & 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 1 & 5 \end{pmatrix}$$

```
for k=1:5
    for j=1:5
        if k==j
            A(k,k)=5;
        elseif abs(k-j)==1
            A(k,j)=1;
        else
            A(k,j)=0;
        end
    end
end
```




4.4 switch-case条件语句

```
switch logical expression  
case value 1  
    statements1  
case value 2  
    statements2  
...  
otherwise  
    statements  
end
```



检测掷一次骰子所得的点数是单数还是双数？

```
function dicetest(result)
```

```
switch result
```

```
case {1,3,5}
```

```
disp('odd number!');
```

```
case {2,4,6}
```

```
disp('even number!')
```

```
otherwise
```

```
disp('What kind of dice do you  
have?')
```

```
end
```

```
>> dicetest(1)
```

```
odd number!
```

```
>> dicetest(4)
```

```
even number!
```

```
>> dicetest(7)
```

```
What kind of dice do you have?
```



4.5 内部函数

MATLAB有许多内置的库函数，

除数学函数外，MATLAB还提供许多机器函数

pause 程序将暂停在该函数所在的位置，击任意键程序继续执行

echo on 在命令窗口显示正在执行的程序指令

tic,toc 记时函数：**tic**开始记时，**toc**给出前一个**tic**开始的机器时间



4.6 用户自定义函数

MATLAB允许用户使用M文件定义函数。函数名与文件名相同。将下面一段程序存为funsim.m

```
function p=funsim(x)
% define a simple function
p=sqrt(x)-2*x^3+cos(x);
```

所给出的函数是

$$\sqrt{x} - 2x^3 + \cos(x)$$

调用

```
y=funsim(1)
```

例 求解函数值 $\frac{a(e^{\frac{50}{a}} + e^{-\frac{50}{a}})}{2} - a - 1$

解： 定义函数M文件f.m如下：

```
function y=f(a)
y=a*(exp(50/a)+exp(-50/a))/2-a-1;
```

保存成文件名为f.m的M文件后，取初值a=5000，在命令窗口输入

```
>> y=f(5000)
```

```
y =  
-0.7500
```





注意：

(1) **函数名和文件名必须相同**

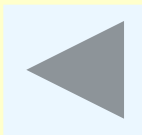
(2) 开头应以function语句开始，第2条以后可加入注释行和其他运算语句。

(3) M函数中使用的变量，除输入和输出变量以外，所有变量都是局部变量，即在该函数返回之后，这些变量会自动在MATLAB的工作空间中清除掉。如果想使这些中间变量在工作空间中起作用，则应该把它们设置成全局变量。

全局变量是由MATLAB提供的global命令来设置的，一般在M函数的开头定义。命令形式为：**global a b c**

不同的全局变量名用空格隔开。**global命令应当在工作空间和M函数中都出现。**如果只在一方出现，则不被承认为全局变量。

(4) 函数文件中可调用其它一般M文件





函数的输出可以多于一个

```
function [x1,x2]=quadroot(a,b,c)
% solve quadratic equation
%  $ax^2+bx+c=0$ 
ds=sqrt(b*b-4*a*c);
x1=(-b+ds)/2/a;
x2=(-b-ds)/2/a;
```

将它存为**quadroot.m**,
它给出二次方程的根,
如此定义的函数可以与
MATLAB的内部函数同样
使用。

```
>> [y1,y2]=quadroot(1,8,15)
```

```
y1 =
```

```
-3
```

```
y2 =
```

```
-5
```



上机实验

1 报告要求:

要求对每次作业写出实习报告，描述（用程序框图或简练语言）算法步骤，附源程序（加详细注释），记录并分析计算结果。

2 程序设计语言：**MATLAB**。要求源程序必须有良好的注释风格、可扩展性和通用性。



3 实验内容：学习完相应章节之后单独布置。

4.报告上交时间：考试以前报告全部上交，
一个上机程序一个报告。



~~END~~