数字图像处理与Matlab

- Matlab的工作环境
- 创建M文件
- Matlab编程
- 常用流程控制语句
- 图像存储与显示
- 数字图像处理中常用的Matlab函数

一 Matlab的工作环境

- Matlab的工作环境简单，明了，易于操作。
- 目前所使用的Matlab软件一般是6.x版本，其工作环境包括五个部分：

  命令窗口（Command Window）、

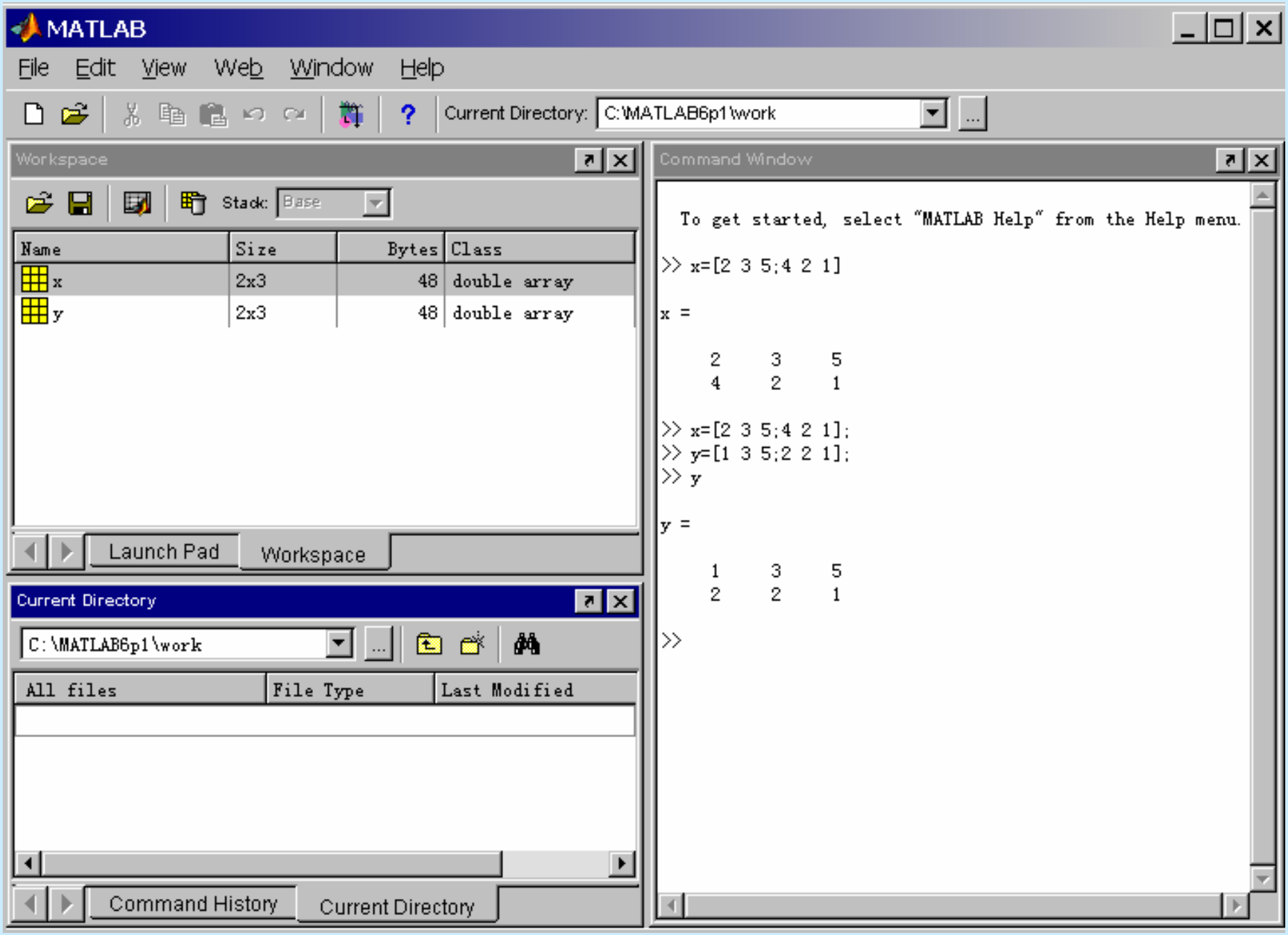
  启动平台（Launch Pad）、

  工作空间（Workspace）、

  命令历史记录（Command History）、

  当前路径窗口（Current Directory）。

如图1所示：


MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work
Workspace
Stack: Base
Name Size Bytes Class
x 2x3 48 double array
y 2x3 48 double array
Launch Pad Workspace
Current Directory
C:\MATLAB6p1\work
All files File Type Last Modified
Command History Current Directory
Command Window
To get started, select "MATLAB Help" from the Help menu.
>> x=[2 3 5;4 2 1]
x =
2 3 5
4 2 1
>> x=[2 3 5;4 2 1];
>> y=[1 3 5;2 2 1];
>> y
y =
1 3 5
2 2 1
>>

1、命令窗口

命令窗口是Matlab的主窗口，用户可以直接在此窗口输入命令，系统将自动显示信息。如在命令窗口中输入指令：

x=[2 3 5; 4 2 1]

数据放在方括号内，行与行之间用“;”（分号）间隔，数值之间用空格或逗号间隔。如果命令后不加“;”，则系统自动解释该命令为一个2x3矩阵，并显示结果如下：

x= 2 3 5

4 2 1

若程序有多行语句，且不需要每行都显示结果，可在不需要显示结果的语句后加上“;”。如图2所示：

Matlab的命令窗口非常适用于编写短小的程序，对编写大型、复杂的程序应采用M文件编程方法。

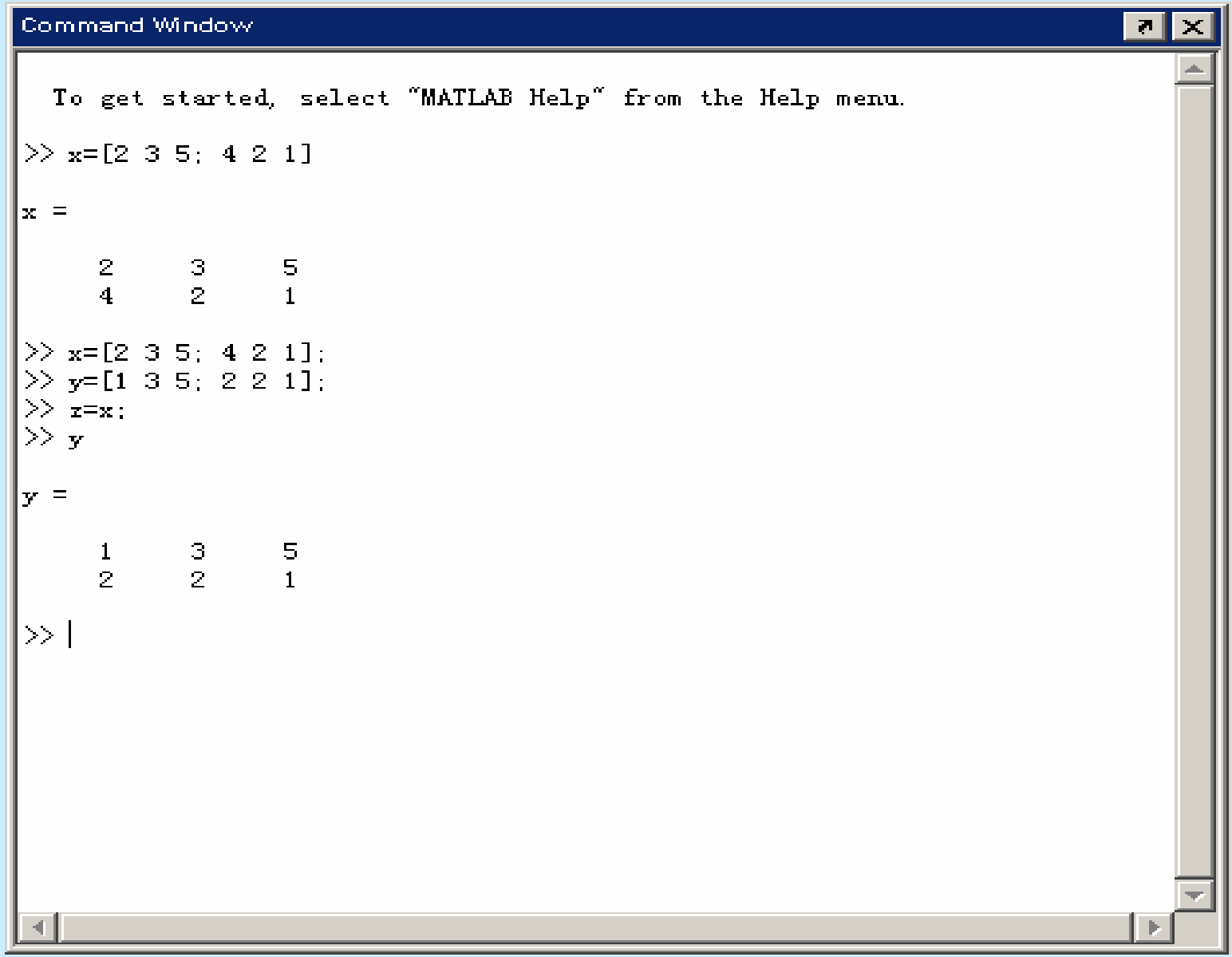

Command Window
To get started, select "MATLAB Help" from the Help menu.
>> x=[2 3 5; 4 2 1]
x =
2 3 5
4 2 1
>> x=[2 3 5; 4 2 1];
>> y=[1 3 5; 2 2 1];
>> x=x;
>> y
y =
1 3 5
2 2 1
>>

图2 命令窗口

2、启动平台

- 当用户需要启动某个工具箱的应用程序时，可以在启动平台（Launch Pad）中实现。

  比如，现在要打开Database Toolbox的帮助（Help）应用程序，找到后双击就会出现Help窗口。如图3所示：

3、 工作空间

Matlab工作空间作为一个独立的窗口，其操作相当方便。它包含着用户已建立的变量，而且变量在工作空间中是以矩阵的形式存储的。例如：在命令窗口中输入的命令有两个变量x、y，在工作空间中就包含这两个变量，双击其中一个变量x，就会出现一个窗口，用来显示变量x的值。如图4所示：

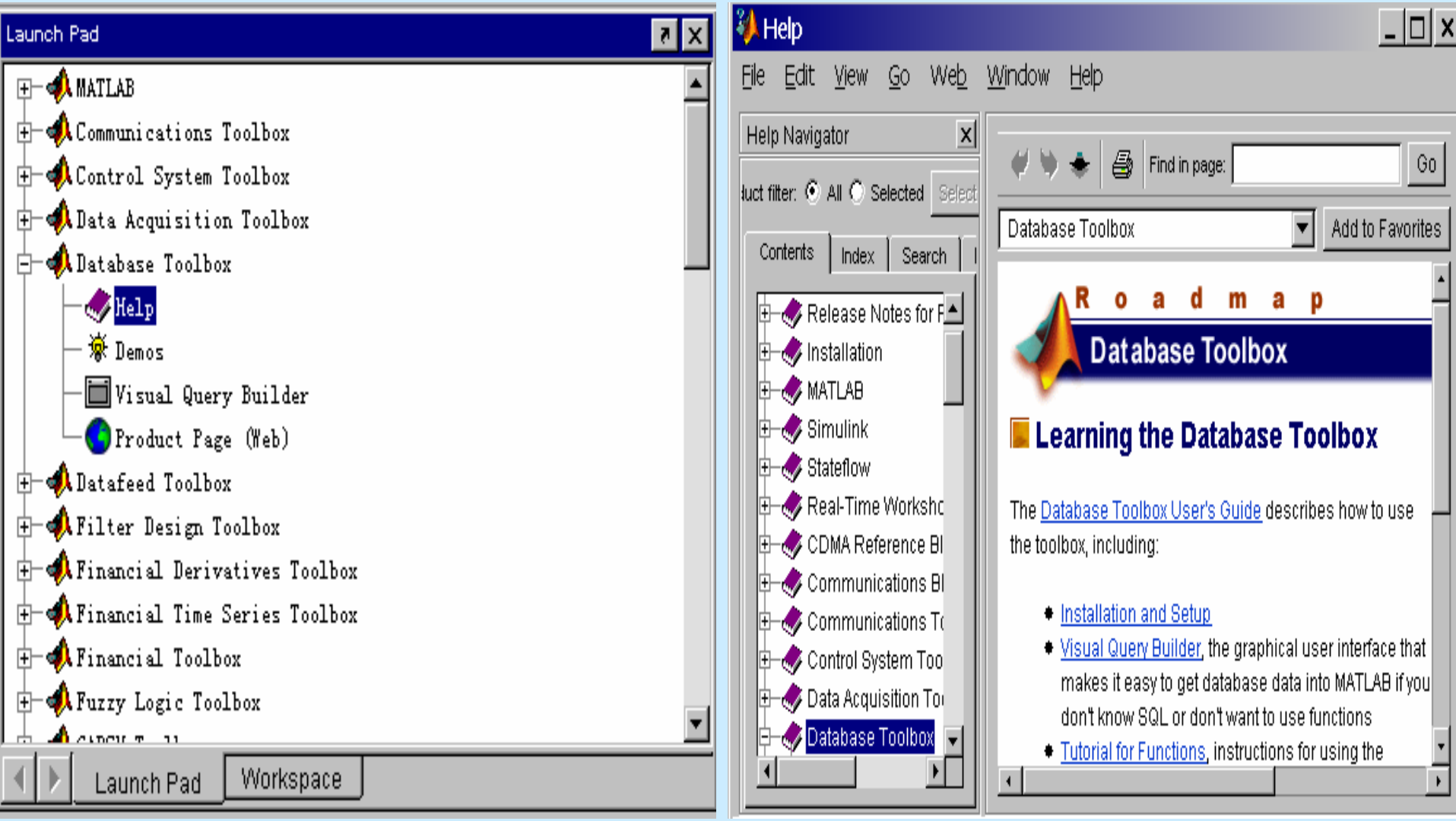

Launch Pad
MATLAB
Communications Toolbox
Control System Toolbox
Data Acquisition Toolbox
Database Toolbox
Help
Demos
Visual Query Builder
Product Page (Web)
Datafeed Toolbox
Filter Design Toolbox
Financial Derivatives Toolbox
Financial Time Series Toolbox
Financial Toolbox
Fuzzy Logic Toolbox
Launch Pad
Workspace
Help
File Edit View Go Web Window Help
Help Navigator
Contents
Index
Search
Release Notes for F
Installation
MATLAB
Simulink
Stateflow
Real-Time Worksho
CDMA Reference Bl
Communications Bl
Communications To
Control System Too
Data Acquisition To
Database Toolbox
Find in page:
Go
Database Toolbox
Add to Favorites
Roadmap
Database Toolbox
Learning the Database Toolbox
The Database Toolbox User's Guide describes how to use the toolbox, including:
Installation and Setup
Visual Query Builder, the graphical user interface that makes it easy to get database data into MATLAB if you don't know SQL or don't want to use functions
Tutorial for Functions, instructions for using the

图3 启动平台示例

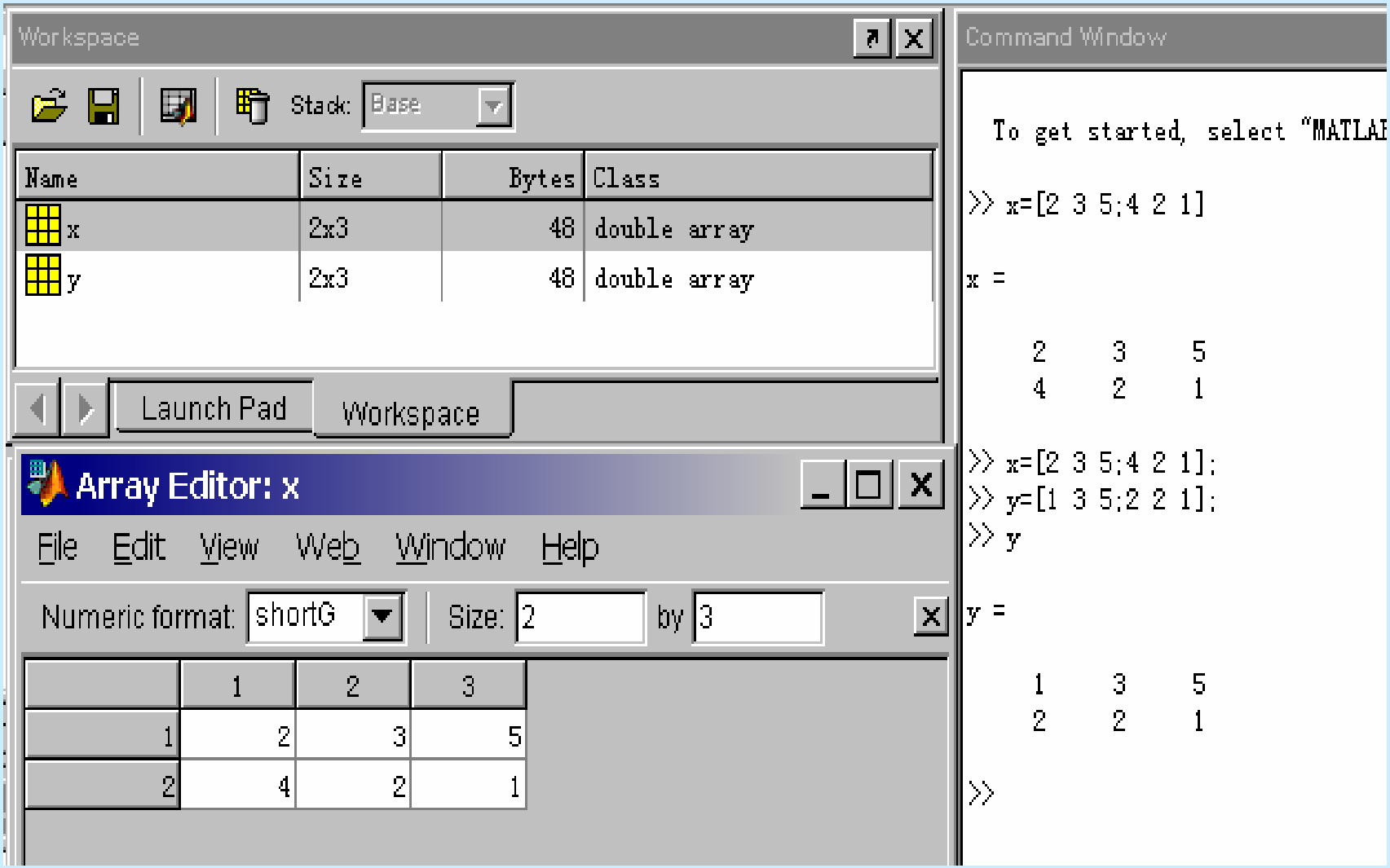

Workspace
Stack: Base
Name Size Bytes Class
x 2x3 48 double array
y 2x3 48 double array
Launch Pad
Workspace
Array Editor: x
File Edit View Web Window Help
Numeric format: shortG
Size: 2 by 3
1 2 3
1 2 3 5
2 4 2 1
Command Window
To get started, select "MATLAB
>> x=[2 3 5;4 2 1]
x =
2 3 5
4 2 1
>> x=[2 3 5;4 2 1];
>> y=[1 3 5;2 2 1];
>> y
y =
1 3 5
2 2 1
>>

图4 工作空间示例

4、 命令历史记录

- 命令历史记录窗口（Command History）主要显示在命令窗口中已执行过的命令。如图5所示：

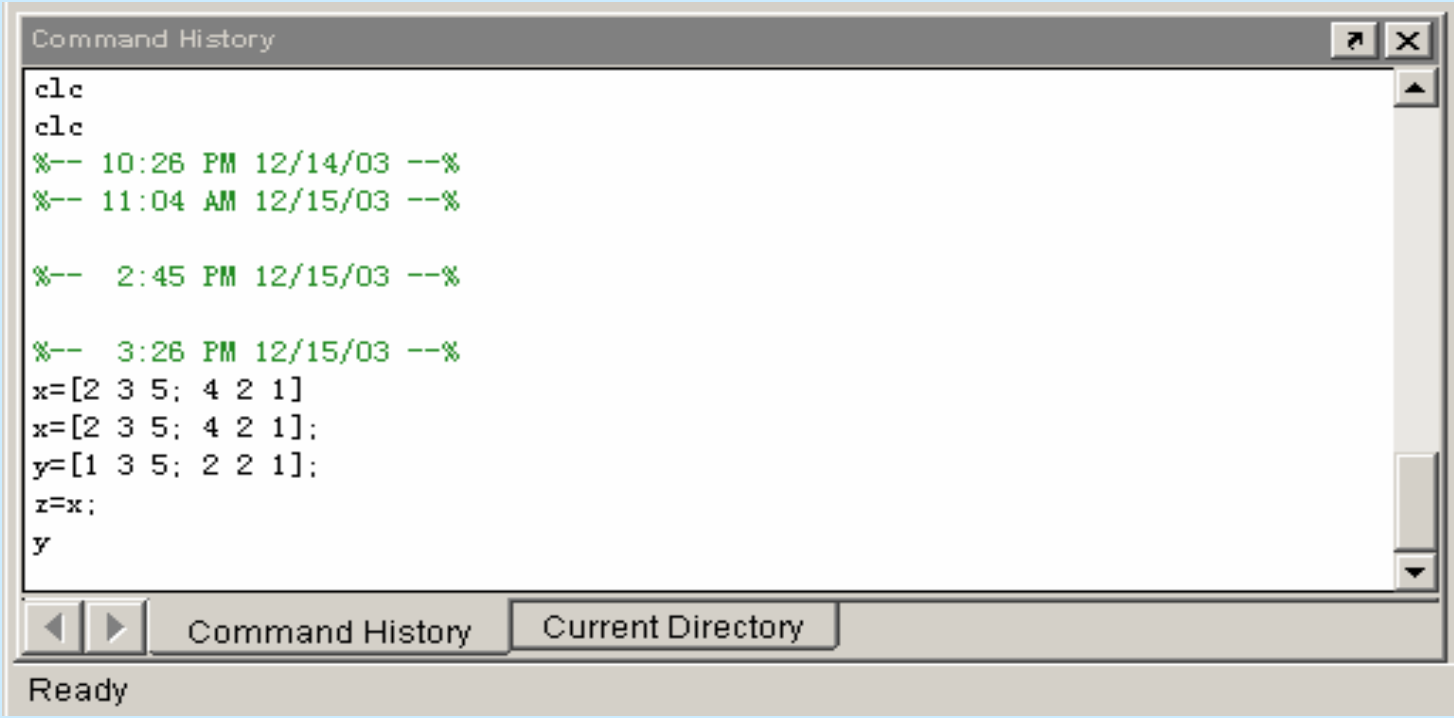


图5 命令历史记录窗口

5、 当前路径窗口

- 当前路径窗口主要显示当前工作在什么路径下进行，包括M文件的打开路径；双击某M文件名，即可打开该文件进行编辑。如图6所示：

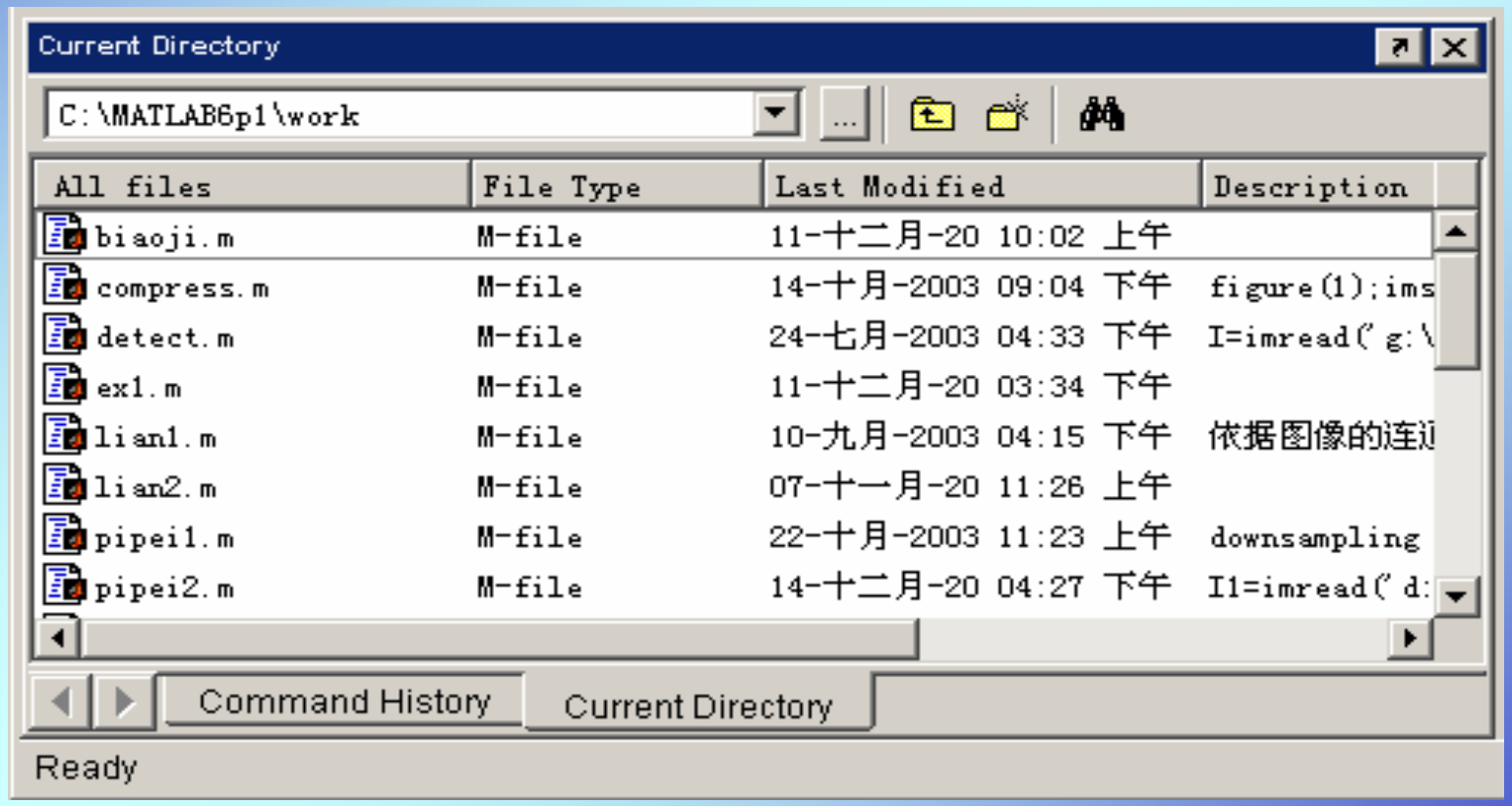


图6 当前路径窗口

二 创建M文件

- 在操作界面的菜单项中，选中File菜单项，出现下拉条，有一项New –M - File，选中点击，即可出现新的M文件窗口。可以通过保存等对M文件操作，并将它放在某一路径下。文件名为*.m，然后就可以在M文件窗口中编程了。如图7示例：

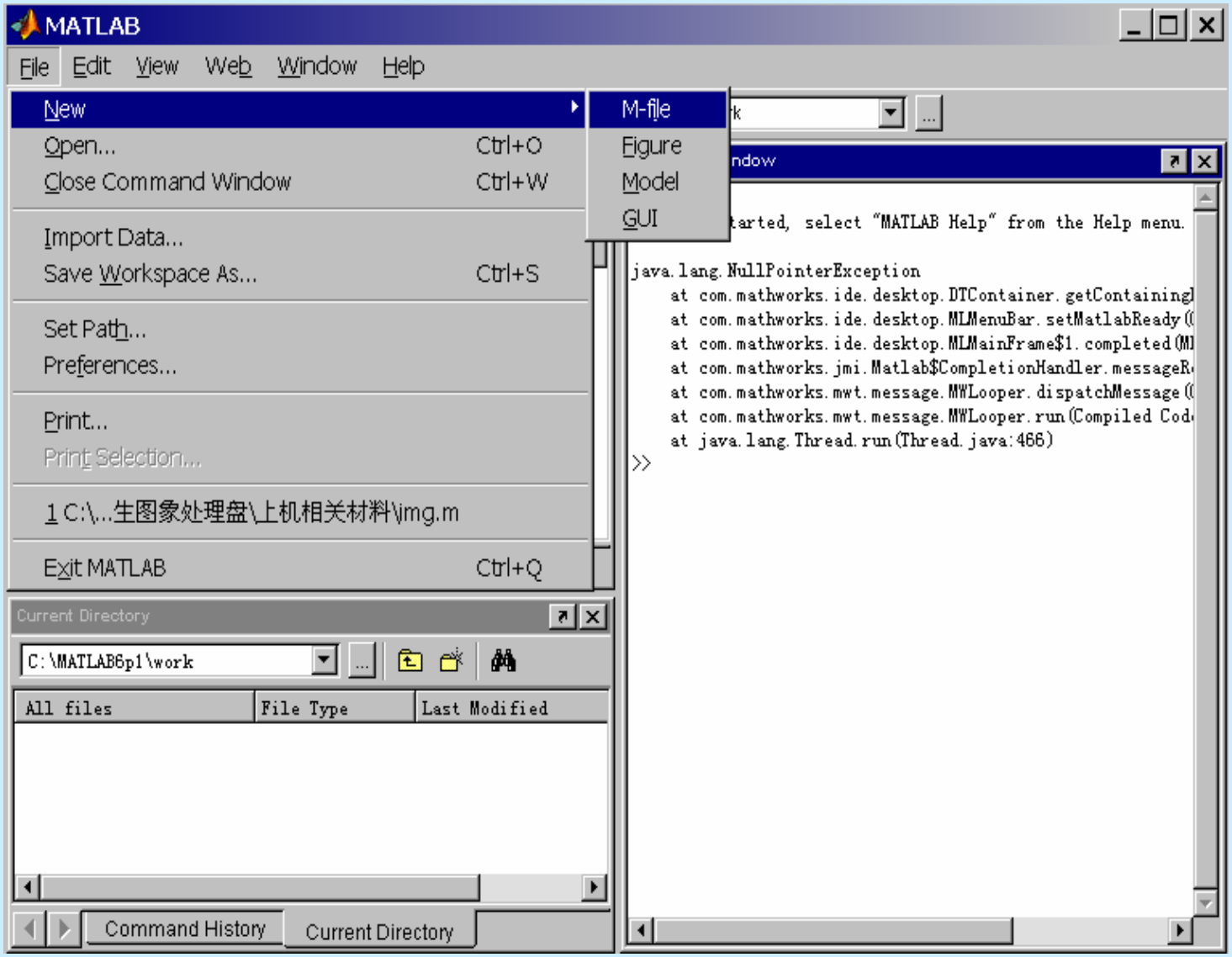
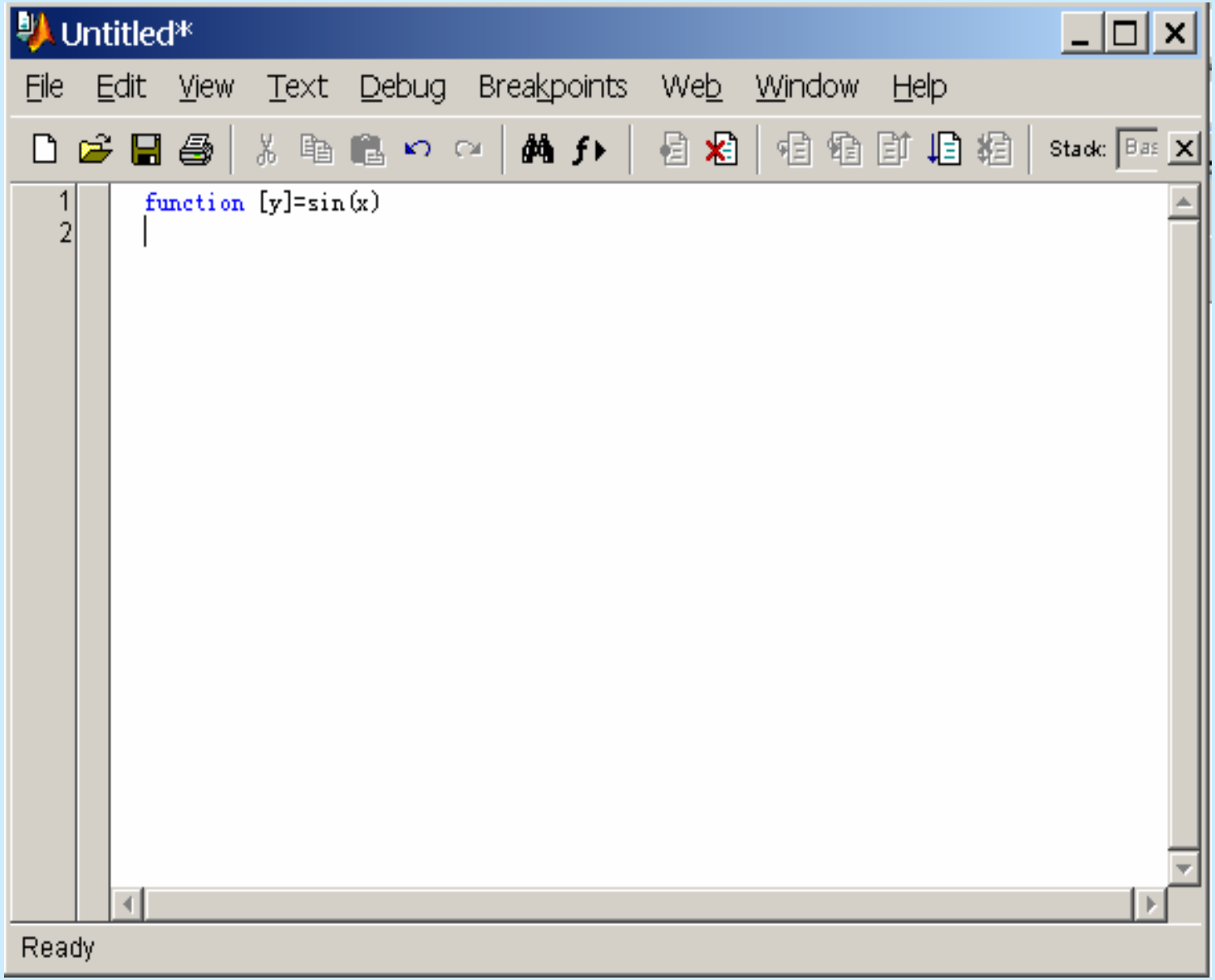

MATLAB
File Edit View Web Window Help
New
Open... Ctrl+O
Close Command Window Ctrl+W
Import Data...
Save Workspace As... Ctrl+S
Set Path...
Preferences...
Print...
Print Selection...
1 C:\...生图象处理盘\上机相关材料\img.m
Exit MATLAB Ctrl+Q
M-file
Figure
Model
GUI
Current Directory
C:\MATLAB6p1\work
All files
File Type
Last Modified
Command History
Current Directory
java.lang.NullPointerException
>>

图7 M文件创建示例


Untitled*
File Edit View Text Debug Breakpoints Web Window Help
Stack:
1 function [y]=sin(x)
2
Ready

M文件窗口

三　Matlab编程

- Matlab编程主要是编辑M文件，Matlab的M文件有两类：

  脚本文件和函数文件。

1、脚本文件

- 我们将原本在Matlab环境下直接输入的语句，放在一个以.m为后缀的文件中，这一文件就称为脚本文件。有了脚本文件，就可在Matlab中输入脚本文件名（不含后缀），这时Matlab会打开这一脚本文件，并依次执行脚本文件中的每一条语句。这和在Matlab中直接输入语句的结果完全一致。

2、函数文件

另一类M文件是函数文件，它的第一行必须是函数定义行。函数文件有五部分构成：

- 函数定义行
- H1行
- 函数帮助文件
- 函数体
- 注释

- 例：函数文件 mean.m
- function y＝mean（x） 函数定义行；
- ％ MEAN Average or mean value。 H1行

  ％ for vectors，MEAN(X) is the mean value of x.

  ％ …….

  ％……. 函数帮助文本（带％的语句）
-
- [m,n]＝size(x);

  if m= = 1

  m= =n 函数体

  end

  y=sum(x)/m；

我们以这个mean函数为例来说明函数的各个部分。

- (1) 函数定义行
- function y = mean (x)

  关键字 输出变量 函数名 输入变量
- 当不含输出变量时，则直接略去输出部分或用空括号表示，例如：

  function Printresults(x);

  function [] =rintfresults(x);
- 当函数具有多个输出变量时，则以方括号括起；当函数具有多个输入变量时，则直接用圆括号括起。如：

  function [x,y,z]=sphere(theta,phi,pho);
- 所有在函数中使用和生成的变量都为局部变量（除非用globle语句定义），这些变量值只能通过输入和输出变量进行传递。因此调用函数时应通过输入变量将参数传给函数。

- （2） H1行

  在脚本和函数文件中，以%开头的行称为注释行，%后语句不被matlab执行。

  在函数文件中，第二行一般为注释行，该行称为H1行，实际上它是帮助文件的第一行。

- （3）函数帮助文本

  以%开头，比较详细的说明函数，从H1行开始，到非%开始行结束。

- （4） 函数体

  完成一定功能的函数实体，它采用任何matlab可使用的命令，包括mablab提供的函数和用户自己设计的函数。

- （5） 注释

  以%开头的行，它可出现函数的任何位置，也可以加在语句行之后，以便对本行进行注释。

- 在函数文件中，除了函数定义和函数体外，其它部分都是可以省略的，不是必须有的。但作为一个函数，为了提高其可用性，应加上H1行和函数帮助文本；为了提高函数的可读性，应加上适当的注释。

四　常用的循环控制语句

- 条件语句
- 指定次重复循环语句
- 不定次重复循环语句
- 情况切换语句

- 条件语句

由 if，else if，end 语句构成：

```
if 条件
    执行语句
end
```

```
if 条件1
    执行语句1
elseif 条件2
    执行语句2
else
    执行语句3
end
```

- **指定次重复循环语句**

for语句可完成指定次重复的循环，这是广泛应用的语句。for语句还可以嵌套使用，从而构成多重循环。格式如下：

```
for 变量＝初始值：步长：终值
    执行语句
end
```

```
for 变量1＝初始值：步长：终值
    for 变量2＝初始值：步长：终值
        执行语句
    end
end
```

若步长为1，可以省略。

- **不定次重复循环语句**

  while语句可完成不定次重复的循环，它与for语句不同，每次循环前要判别其条件，如果条件为真或非零值，则循环，否则结束循环。而条件是一表达式，其值必定会受到循环语句的影响。

- 在重复循环语句中可使用break语句退出循环。

  格式：

```
While 条件
    执行语句
end
```

- 例：

```
r=1;
while r<10
 r=r+1;
    if r>7
      break ;
    end
end
```

- 情况切换语句

switch语句可根据表达式的不同取值执行不同的语句，这相当于多条if语句的嵌套使用。

格式：switch 表达式

case 值1

执行语句1

case 值2

执行语句2

case 值3

执行语句3

otherwise

执行语句

end

- 例：变量var={-1,0,1}

```
switch var
    case –1
        disp(“var is -1”)
    case 0
        disp(“var is 0”)
    case 1
        disp(“var is 1”)
end
```

五　图像存储与显示

- 图像存储
- 图像显示

- **图像存储**

在Matlab中，图像是以矩阵形式存储的，对图像的操作也就是对矩阵的操作。

例：图像名用Image表示，则Image表示一个图像矩阵，i代表矩阵的行，j代表矩阵的列。

Image（i,j）代表图像在点（i,j）的灰度值。

- **图像的显示**

图像分为

- 有格式图像：如. bmp格式；. tif 格式；…
- 无格式图像：如. img（以二进制格式存储）

不同的格式显示方式不同。

1、有格式图像显示

- 对有格式图像，可用imread（）函数读入，用imshow（）函数显示。

例：I＝imread(‘d:\image\x.bmp‘);

或　I＝imread（’d:\image\x’,’bmp’）；

路径　格式

```
figure(1);
imshow（I）；％显示图像I
```

2、无格式图像显示（以二进制格式存储 ）

- 首先要用fopen（）函数打开某一路径下的文件，然后用fread（）函数读到一个变量里，该变量为矩阵变量。
- 例：fid＝fopen（‘d:\img\lena.img’,’r’）;%fid为文件句柄

```
data＝（fread（fid，[256，256]，‘uint8’））’;
figure(1);
images(data,[255]); %图像为256级灰度
colormap(gray);　％显示灰度图像
axis image　　％对图像加坐标。
```

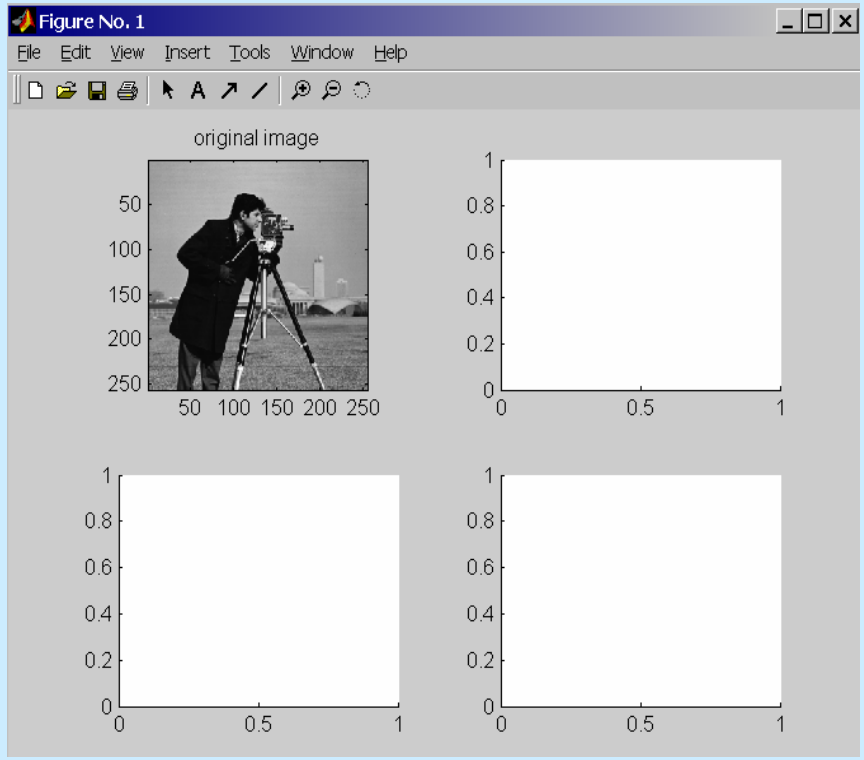
- 若不能直接对data进行处理，可以进行转换后在处理；

转换语句：　　data=uint8(data);

3、同屏显示多个图像

- 可用subplot(m,n)将图形窗分为m*n个子窗口，然后取第一、第二...子窗口显示不同的图像，实现同屏显示多个图像。例如：

```
figure(1);
%取2×2个子屏中的第一个子屏
subplot（2,2,1);
……
%显示第一个图像
imshow(I1);
……
%取2×2个子屏中的第四个子屏
subplot(2,2,4);
……
%显示第四个图像
imshow(I4);
```



六 数字图像处理中常用的matlab函数

- Size（）函数
- Zeros（）函数
- Fft2(), ifft2()函数
- Imhist() 函数
- Histeq() 函数
- Imrotate() 函数
- Imnoise() 函数
- Edge() 函数
- Title()函数
- Xlable(),Ylable()函数

- Size（）函数

获取图像矩阵大小。一般是应用于有格式图像，因无格式（二进制）图像的大小在读入时已知。

例：I＝imread(‘d:\img\radar’,’bmp’);%读入图像

[x,y]＝size(I)；%获取图像大小

得到x,y的值，该图像大小就是X×Y。

- Zeros（）函数

零矩阵函数。

例：I=zeros(100,100)；%I为100×100的零矩阵，矩阵%中元素全为零。

Imshow(I)；%显示一个100×100的黑方块。

例如作业题1，生成一个外边黑中间一块是白的图像，可以先生成一个全黑的图像，然后在中间作一个双重循环，赋象素值为255或1。

```
I1＝zeros(128,128)；%生成一个128×128的全黑图像
for I=38:1:90
    for j=58:1:70
        I1（i,j）＝255；%或I1(i,j)=1；
    end
end
imshow（I1）；%I1即为所求图形。
```

- fft2(), ifft2()函数

  fft2()函数为二维快速傅立叶变换函数；

  ifft2()函数为二维逆快速傅立叶变换函数。

  对一幅图像进行傅立叶变换后，得到它的频谱。

例：

```
I2=fft2(I1)；%对图像I1进行二维快速傅立叶变换
Imshow(I2);%显示频谱图。
I3=ifft2(I2)；%对图像I2进行二维逆快速傅立
              % 叶变换, 得到原图像
```

- Imhist() 函数

  图像直方图函数。

  例：imhist(I)；%显示图像I的直方图；

- Histeq() 函数

  直方图均衡化函数。

  例：

```
I1=histeq(I)；%对图像I进行直方图均衡化。
Imshow(I1);%显示均衡化后的图像I1
```

- Imrotate() 函数： 旋转图像函数；

格式： J = imrotate(I,angle,method)

I：被旋转图像　　J：旋转后的图像

angle：旋转角度

method：可为'nearest‘、'bilinear‘、'bicubic'

例： I=imread(‘ic.tif’);%读入图像ic.tif

J＝imrotate(I,45,’bilinear’);%对图像I旋转45度

imshow(I)；%显示原图I

figure,imshow(J)；%显示旋转后的图像J

- Imnoise() 函数： 给图像增加噪声。

格式： J=imnoise(I, ’噪声类型’, 参数)；

I:待加噪声图像；

噪声类型：高斯噪声、盐噪声等；

参数：噪声密度(0--1)；

J：加入噪声的图像。

例： J=imnoise(I, ‘salt&pepper’， 0.02)；%给图象I增加盐噪声

- Edge() 函数

边缘检测函数。

J＝edge(I,‘检测算子‘)； %J是对图像I用某算子

%进行边缘检测后的边缘图；

%J是二值图像，即黑、白二色。

例：

J＝edge(I,’Roberts’)； %用Roberts算子对图像I进行边缘检测

imshow(J); %显示边缘图

- Title()函数

给图像加标题 ；

例： title(‘图像变换结果图‘)；

%图像标题为“图像变换结果图”

- Xlable(),Ylable()函数

对图像的x轴、y轴加标注。

例： Xlable(‘时间t’);%x轴代表’时间t’

Y1able(‘函数f(t)’);% y轴代表函数f(t)

- 关于Matlab函数的具体应用，可以查Matlab的Help，Help中有详细的说明。