



第3章 Matlab软件

第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 1 Matlab开发环境

MATLAB是美国MathWorks公司推出的一套高效能的科学计算软件。

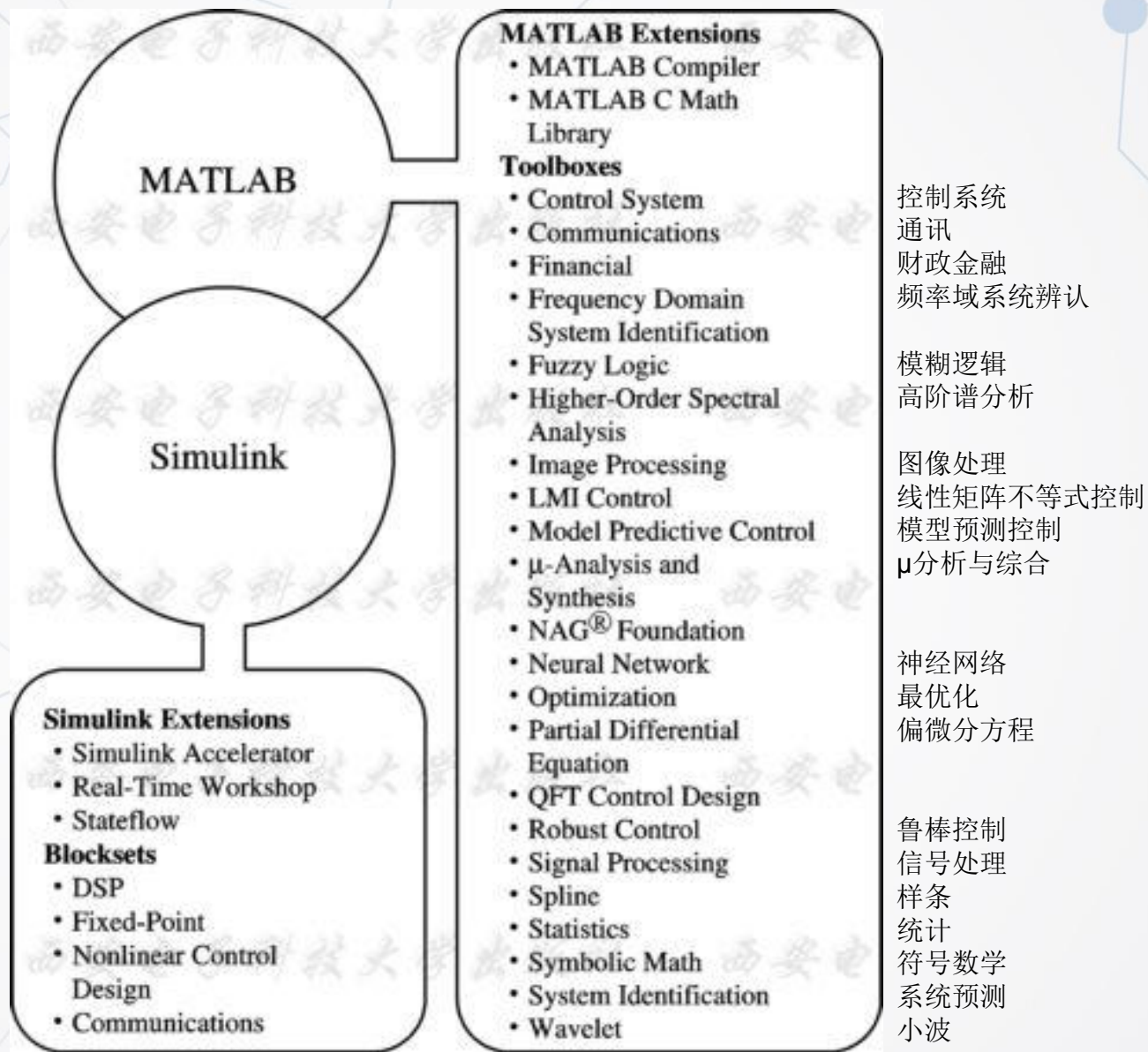
1980年，美国新墨西哥州立大学计算机系主任Clever Moler为帮助学生解决线性代数课程中的矩阵计算问题，编写了Matrix Laboratory(矩阵实验室)，后简称为MATLAB。1984年，MathWorks公司成立，正式把MATLAB 1.0推向市场。至2016年，MATLAB的版本已升级到9.1版。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计等众多科学领域提供了一种全面的解决方案，是当今世界上首

§ 1 Matlab开发环境

屈一指的标准科学计算软件。

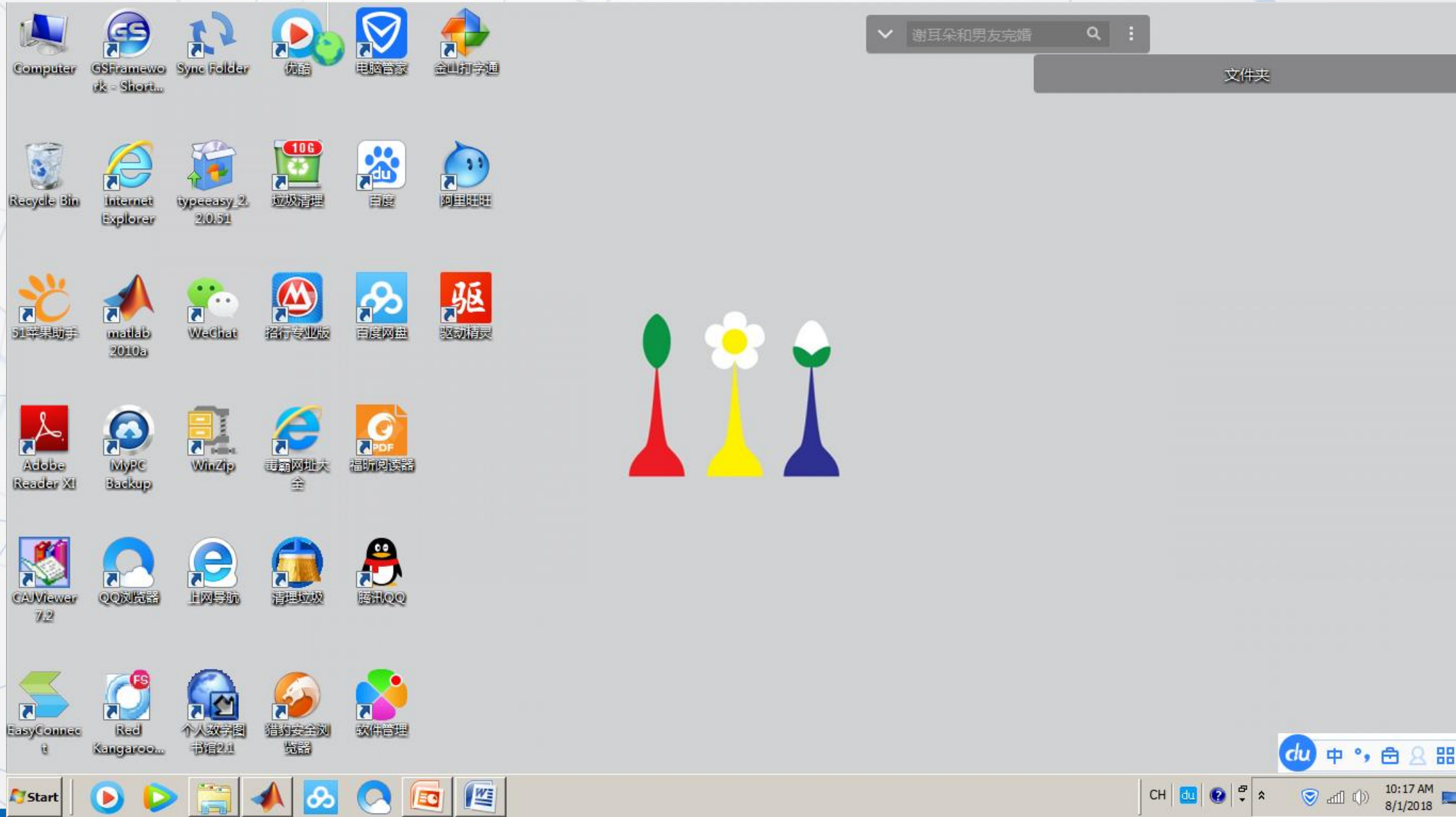
MATLAB的基本数据单位是矩阵，Matlab可直接对矩阵进行操作，而且指令表达式与数学、工程中常用的形式十分相似，故用MATLAB来计算问题要比用C，FORTRAN等语言完成相同的事情简捷得多。

MATLAB功能强大，并为用户提供了许多方便实用的工具箱，可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、信号处理与通讯、图像处理、信号检测、金融建模设计与分析等领域。

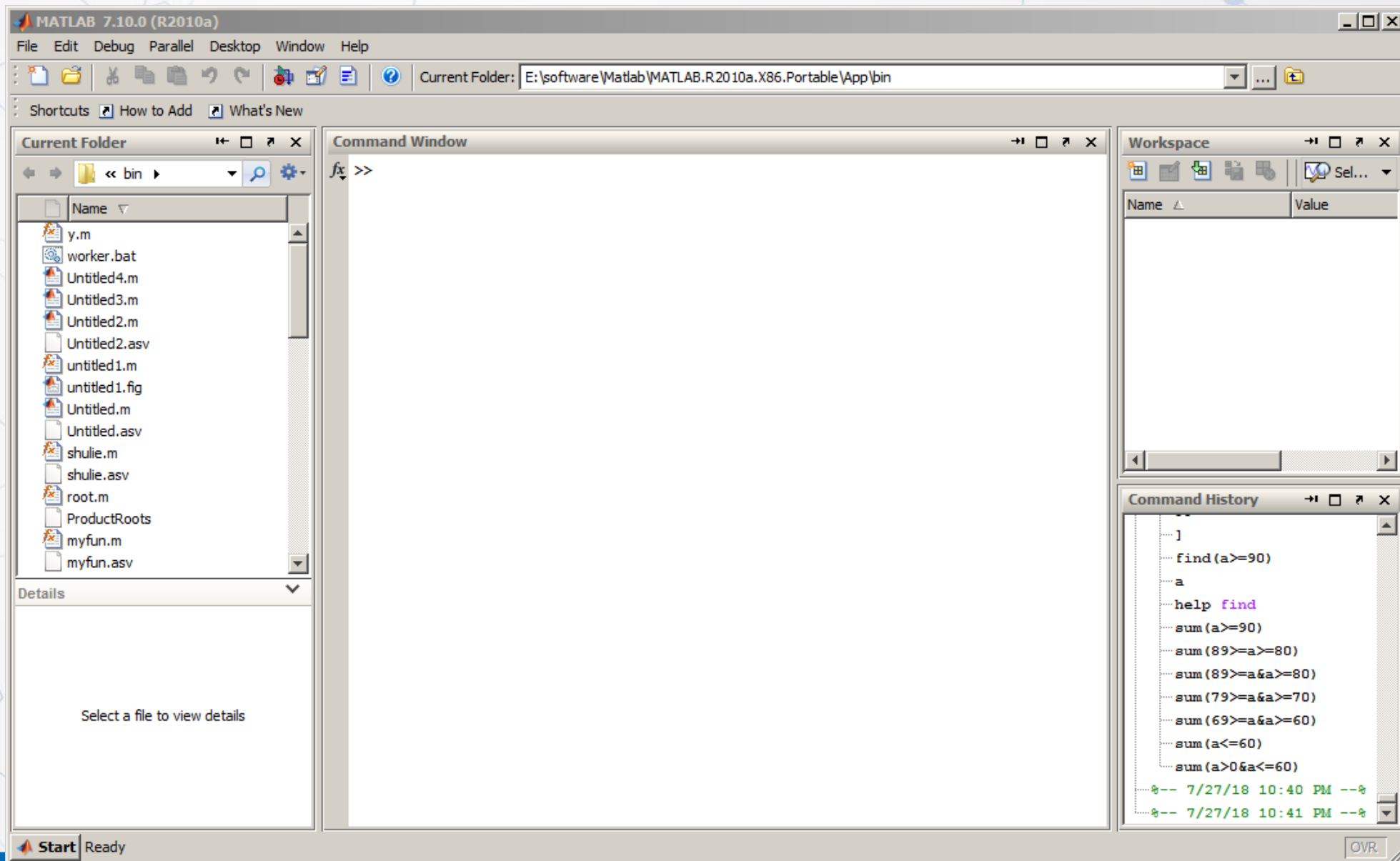


MATLAB的系统组成

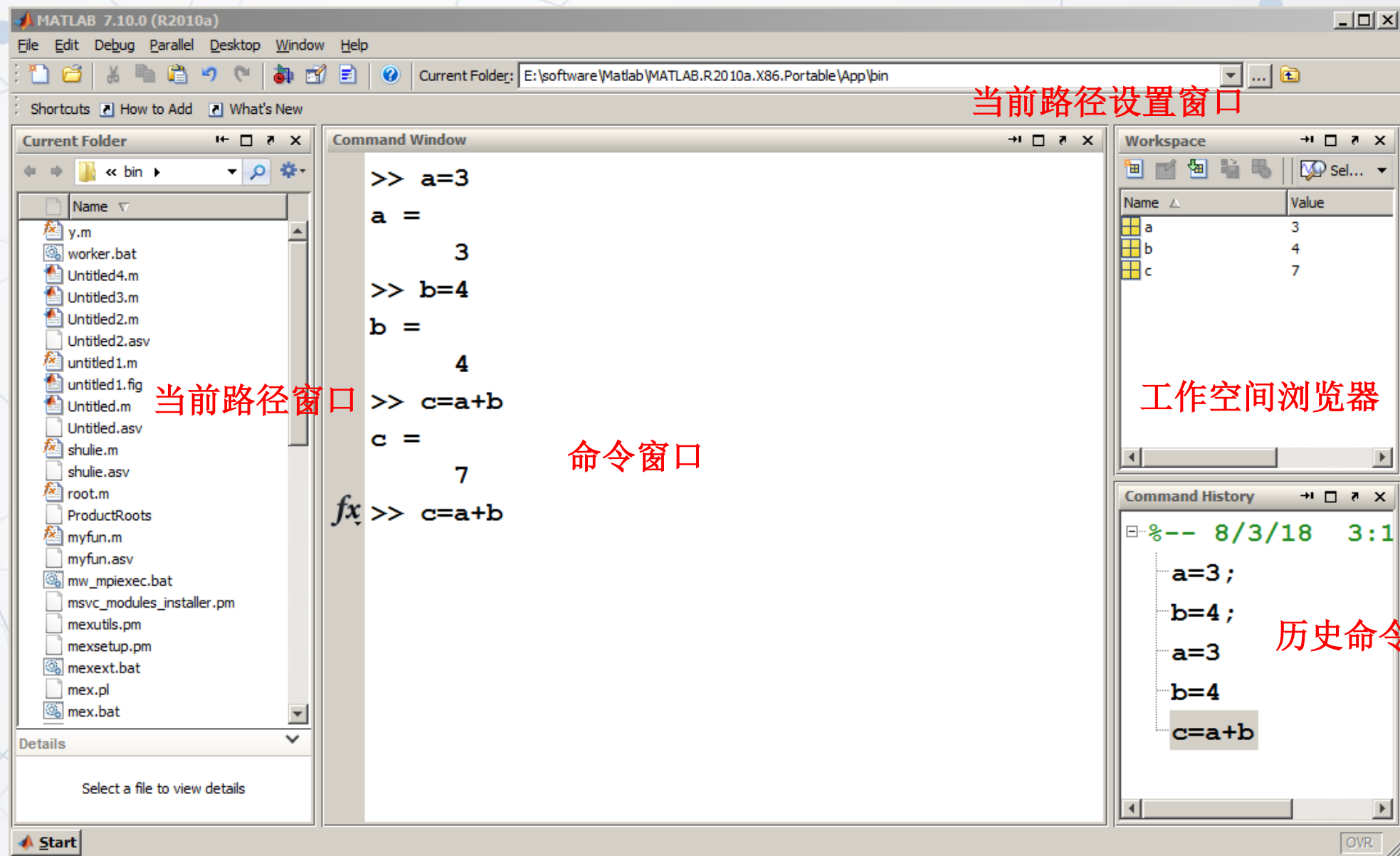
§ 1 Matlab开发环境



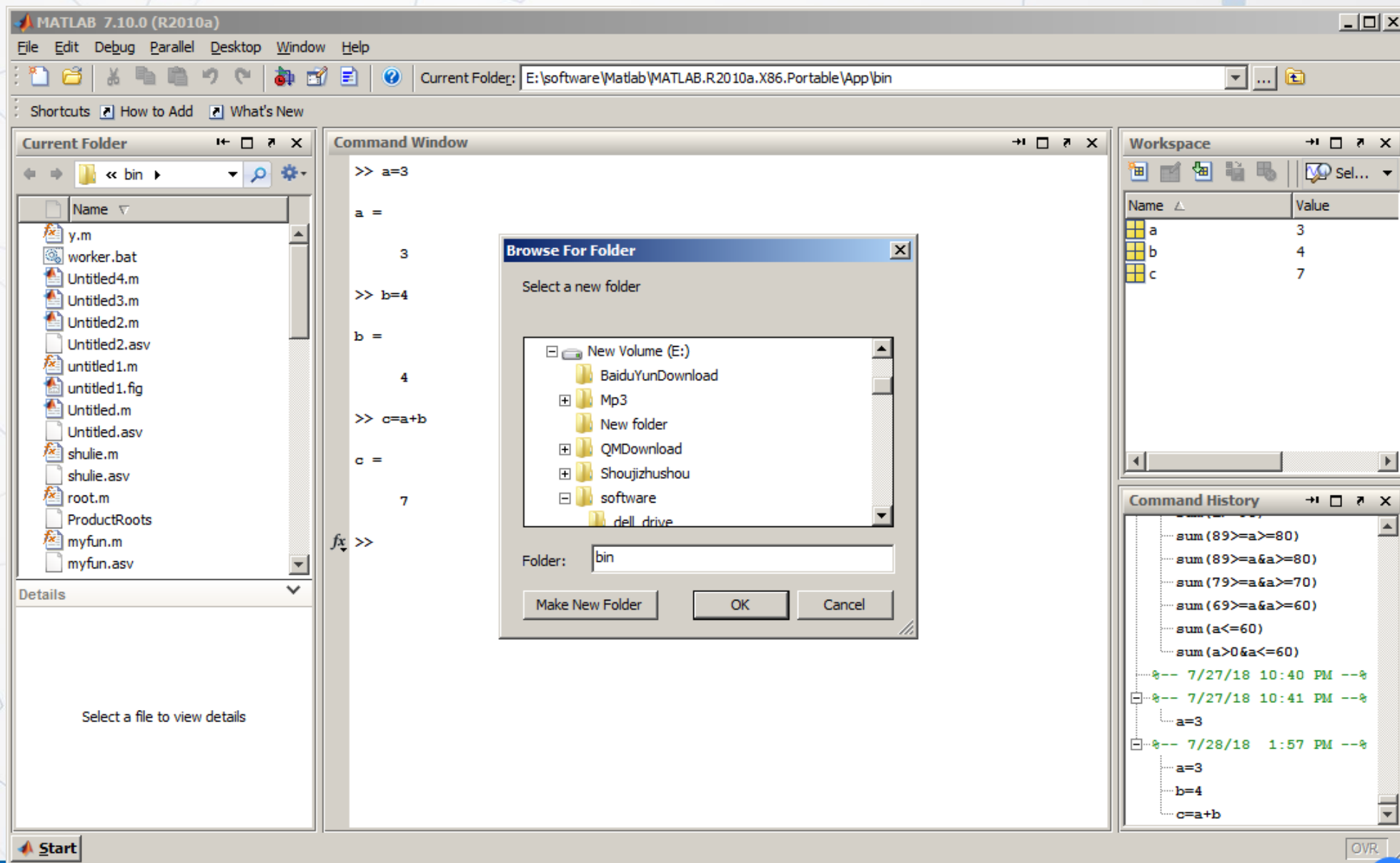
§ 1 Matlab开发环境



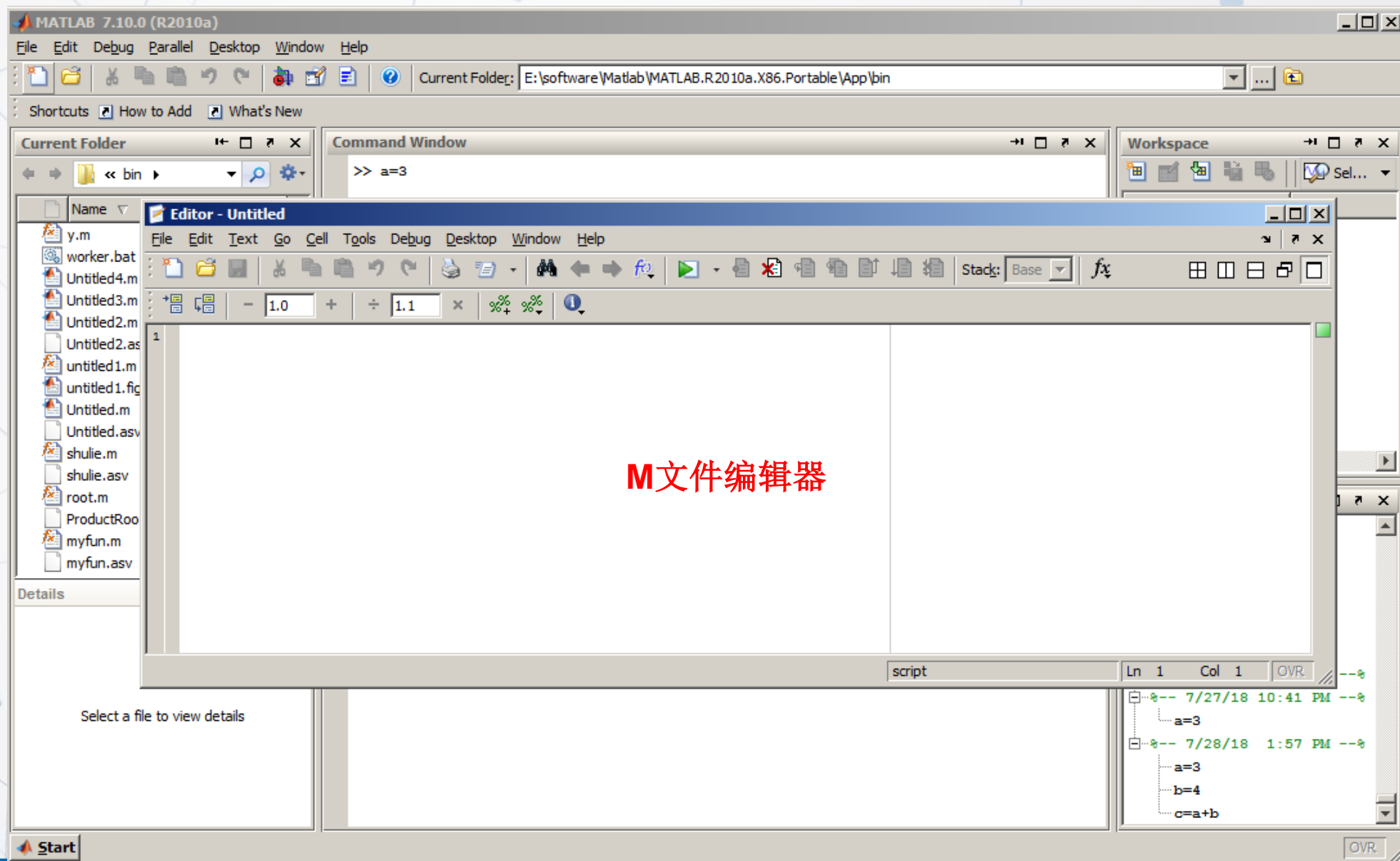
§ 1 Matlab开发环境



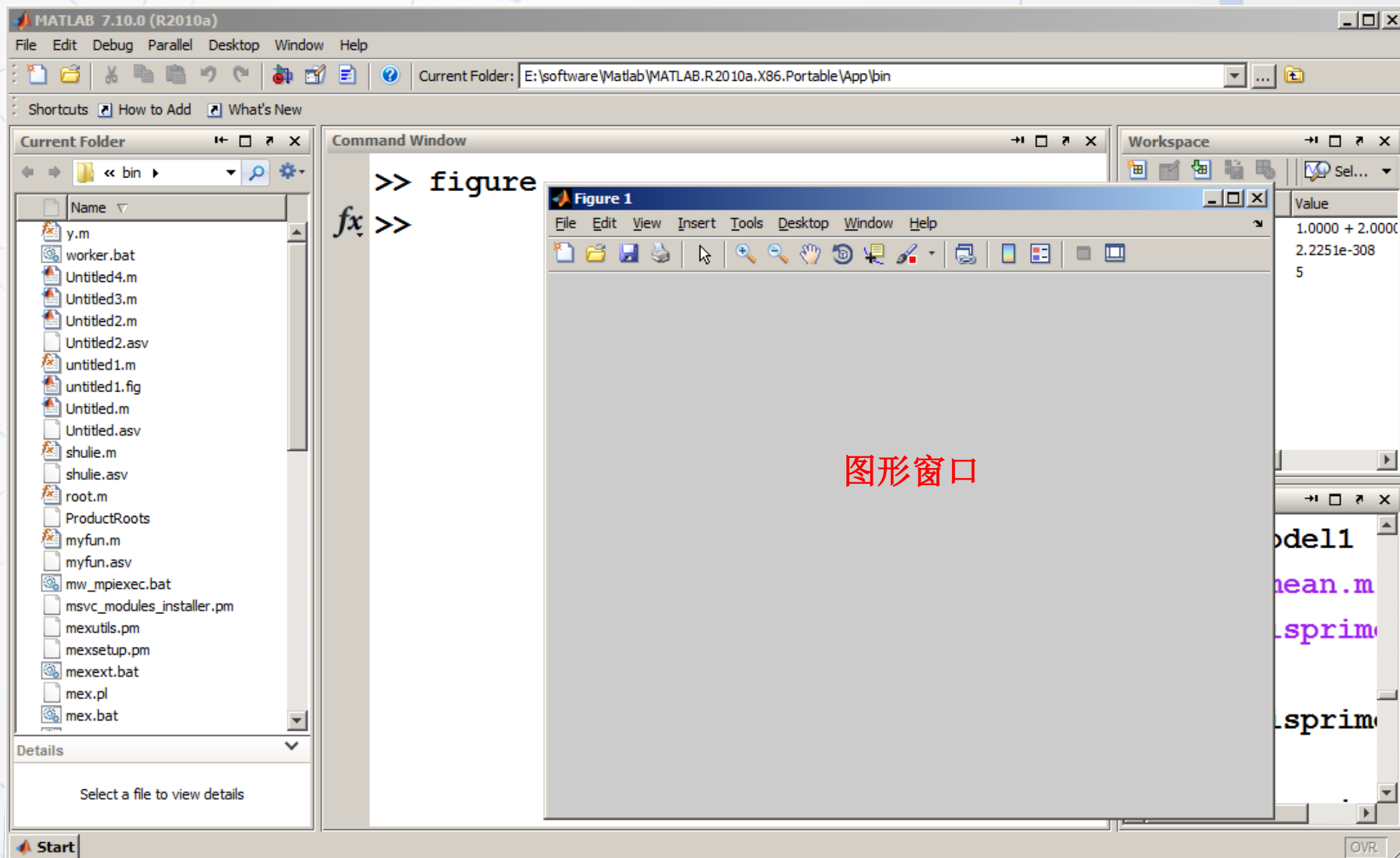
§ 1 Matlab开发环境



§ 1 Matlab开发环境



§ 1 Matlab开发环境



第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 2 Matlab中的变量、操作符、函数

2.1 变量

2.2 操作符

2.3 函数

§ 2 Matlab中的变量、操作符、函数

2.1 变量

- (1) 变量定义及命名规则
- (2) Matlab系统预定义的变量

§ 2 Matlab中的变量、操作符、函数

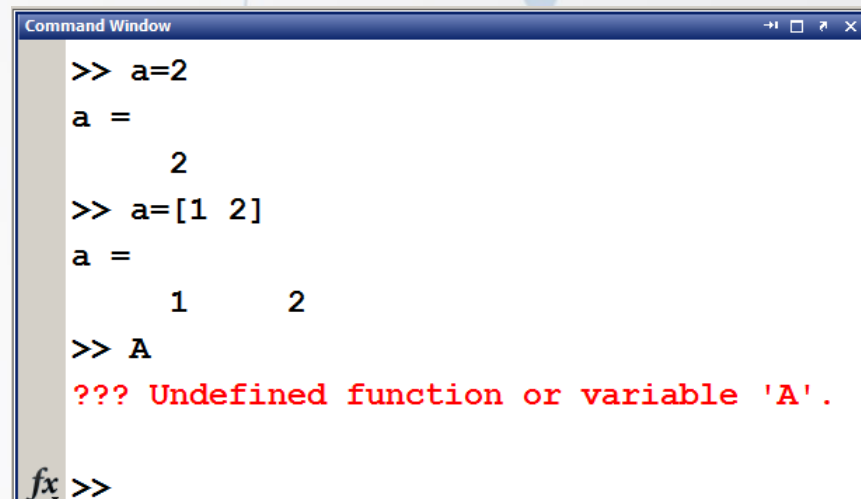
2.1 变量

(1) 变量定义及命名规则

Matlab中的变量不需要事先定义，在遇

到新的变量名时，Matlab会自动建立变量并分配存储空间。当遇到已存在的变量时，Matlab会更新其内容，如有必要会重新分配存储空间。

变量名由字母、数字和下划线构成，并且必须以字母开头，最长为31个字符。Matlab能区分大小写字母，变量A和a是两个完全不同的变量。



```
Command Window
>> a=2
a =
     2
>> a=[1 2]
a =
     1     2
>> A
??? Undefined function or variable 'A'.
fx >>
```

§ 2 Matlab中的变量、操作符、函数

2.1 变量

(2) Matlab系统预定义的变量

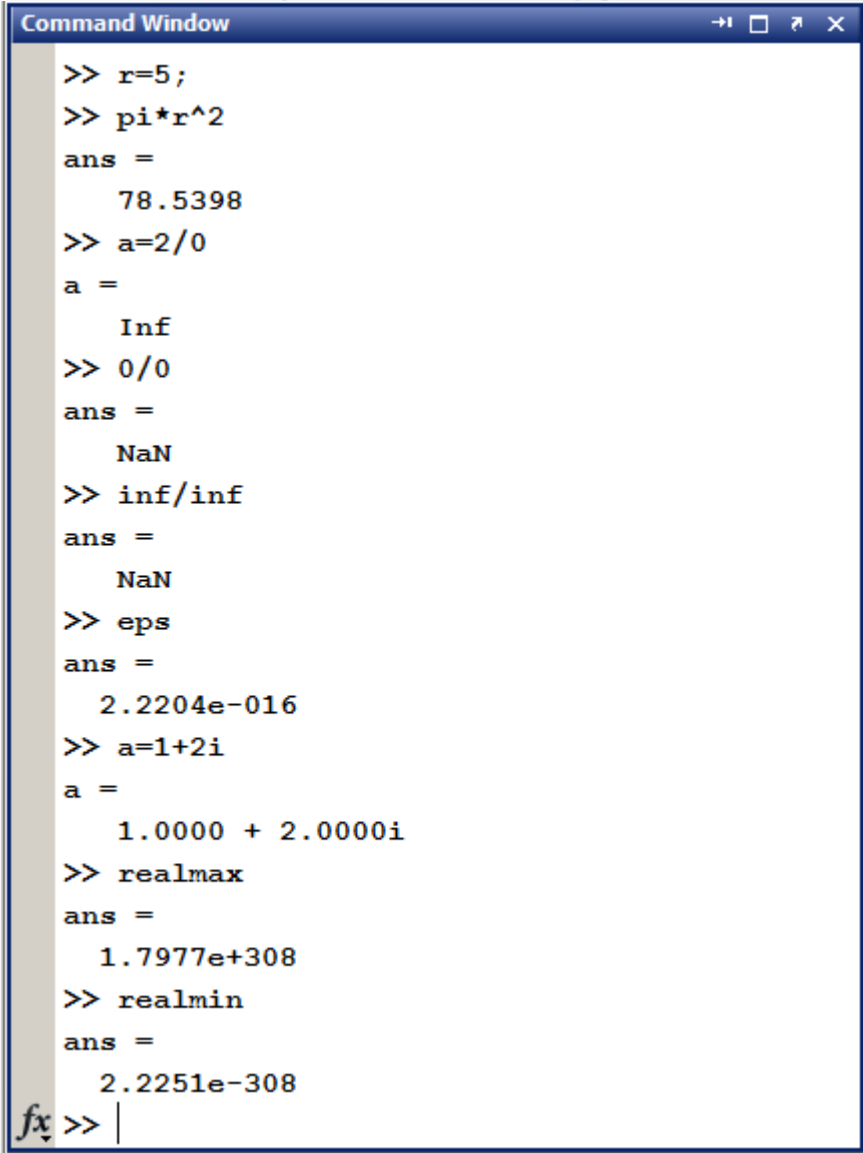
ans 缺省变量 pi 圆周率

inf 无穷大 NaN 不定量

eps Matlab中最小的正数

i,j 虚数单位

realmax (realmin) 最大 (最小) 的正浮点数



```
Command Window
>> r=5;
>> pi*r^2
ans =
    78.5398
>> a=2/0
a =
    Inf
>> 0/0
ans =
    NaN
>> inf/inf
ans =
    NaN
>> eps
ans =
    2.2204e-016
>> a=1+2i
a =
    1.0000 + 2.0000i
>> realmax
ans =
    1.7977e+308
>> realmin
ans =
    2.2251e-308
fx >> |
```


§ 2 Matlab中的变量、操作符、函数

2.2 操作符

- (1) 算术运算符
- (2) 关系运算符
- (3) 逻辑运算符
- (4) 位运算及其它运算符（略）

§ 2 Matlab中的变量、操作符、函数

2.2 操作符

(1) 算术运算符

MATLAB定义了两种不同的算术运算：矩阵运算和阵列运算。矩阵运算由线性代数规则来定义，阵列运算是元素对元素的运算，这两种运算用句点来区分。MATLAB算术运算符包括：

+ 加法

- 减法

* 乘法

/ 除法

\ 左除法

^ 指数

' 复共轭转置

.* 阵列乘法

./ 阵列除法

.\ 阵列左除法

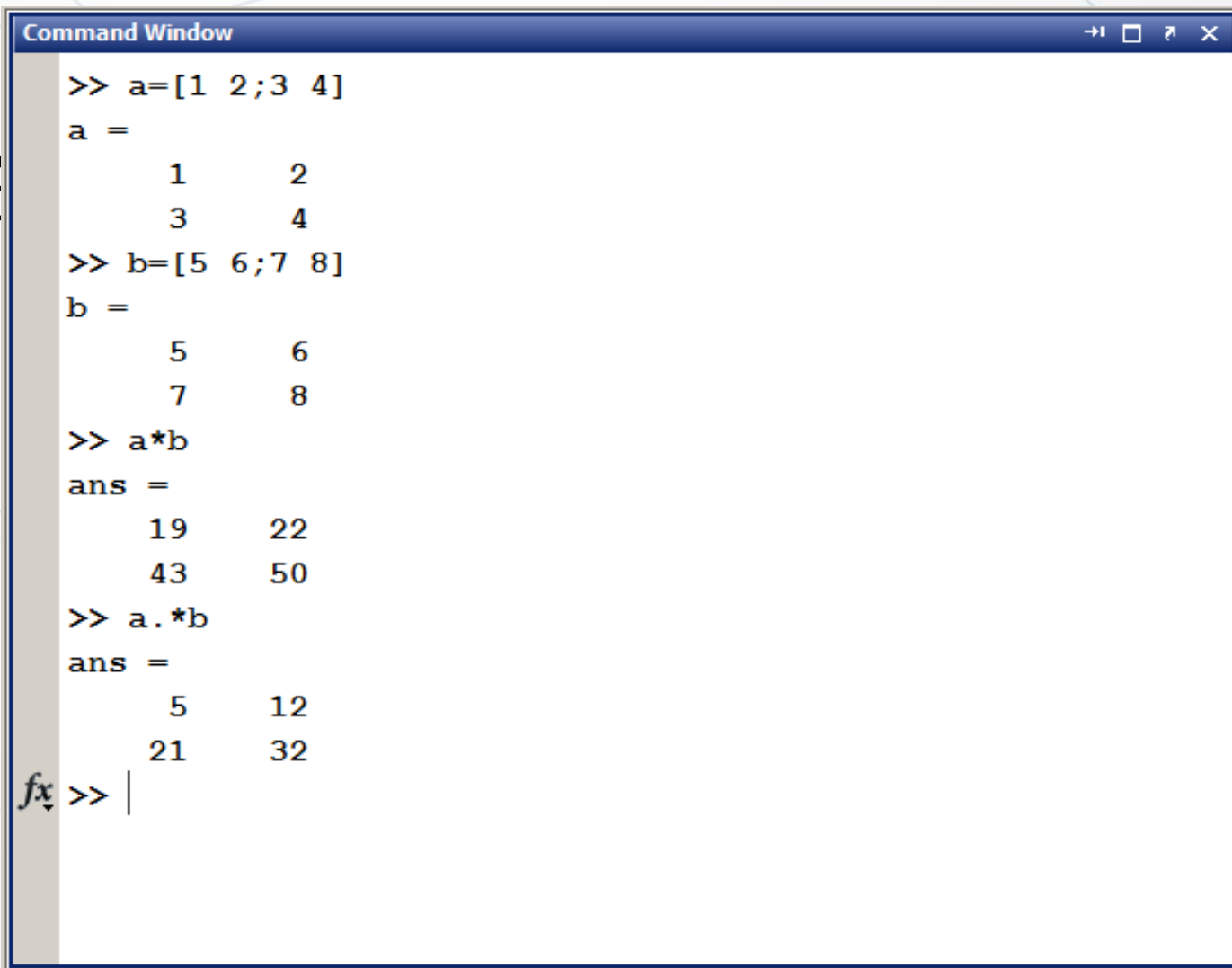
.^ 阵列指数

! 非共轭阵列转置

§ 2 Matlab中的变量、操作符、函数

2.2 操作符

(1) 算术运算符



The screenshot shows the MATLAB Command Window with the following text:

```
Command Window
>> a=[1 2;3 4]
a =
     1     2
     3     4
>> b=[5 6;7 8]
b =
     5     6
     7     8
>> a*b
ans =
    19    22
    43    50
>> a.*b
ans =
     5    12
    21    32
fx >> |
```

The Command Window title bar includes standard window controls (minimize, maximize, close) and a search icon.

§ 2 Matlab中的变量、操作符、函数

2.2 操作符

(2) 关系运算符

< 小于 <= 小于等于

> 大于 >= 大于等于

== 等于 ~= 不等于

```
Command Window
>> a=[1 2;3 1]
a =
     1     2
     3     1
>> b=[-1 0;4 1]
b =
    -1     0
     4     1
>> a>b
ans =
     1     1
     0     0
fx >> |
```

关系运算符可完成两个阵列之间元素对元素的比较，其结果为相同维数的阵列。关系成立时相应的元素为逻辑真(1)，关系不成立为逻辑假(0)。

§ 2 Matlab中的变量、操作符、函数

2.2 操作符

(3) 逻辑运算符

MATLAB提供了三个逻辑操作符&(与)、|(或)、~(非)，逻辑操作的结果为逻辑量0和1构成的矩阵。在逻辑操作中，都是按元素操作的。0表示逻辑假(F)，1表示逻辑真(T)，所有的非零值元素都当作“1”处理。

$a \& b$ ：两标量或矩阵两元素均非0返回1，否则返回0.

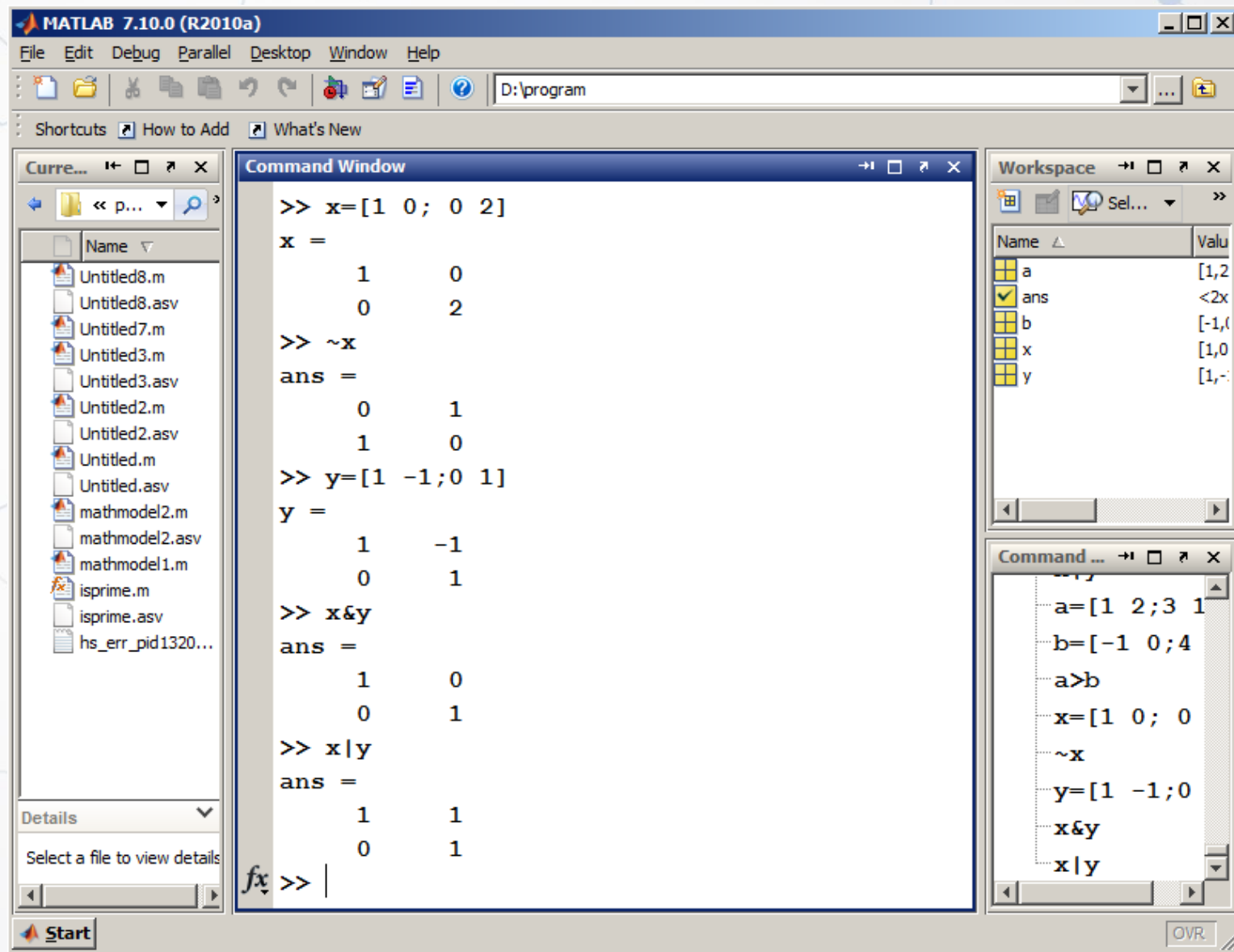
$a | b$ ：两标量或者矩阵两元素均0返回0，否则返回1.

$\sim a$ ：标量或者矩阵元素为0返回1，标量或矩阵元素非0返回0.

§ 2 Matlab中的变量、操作符、函数

2.2 操作符

(3) 逻辑运算符



The screenshot displays the MATLAB 7.10.0 (R2010a) environment. The Command Window shows the following sequence of operations:

```
>> x=[1 0; 0 2]
x =
     1     0
     0     2

>> ~x
ans =
     0     1
     1     0

>> y=[1 -1;0 1]
y =
     1    -1
     0     1

>> x&y
ans =
     1     0
     0     1

>> x|y
ans =
     1     1
     0     1
```

The Workspace window on the right lists the following variables:

Name	Value
a	[1,2]
ans	<2x
b	[-1,0]
x	[1,0]
y	[1,-1]

The Command History window at the bottom shows the executed commands:

```
a=[1 2;3 1]
b=[-1 0;4
a>b
x=[1 0; 0
~x
y=[1 -1;0
x&y
x|y
```

§ 2 Matlab中的变量、操作符、函数

2.3 函数

从本质上，MATLAB函数可分为以下三类：

- (1) MATLAB的内部函数，这种函数是系统自带的函数；
- (2) MATLAB系统附带的各种工具箱中的M文件所提供的大量实用函数，使用这些函数时，需安装相应的工具箱；
- (3) 用户自己增加的函数。

第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 3 基本操作

3.1 矩阵的输入

3.2 矩阵的操作

3.3 基本数学运算

§ 3 基本操作

3.1 矩阵的输入

- (1) 输入元素列表
- (2) 从外部数据文件中读取矩阵
- (3) 利用MATLAB内部函数与工具箱函数产生矩阵
- (4) 用户自己编写M文件产生矩阵

§ 3 基本操作

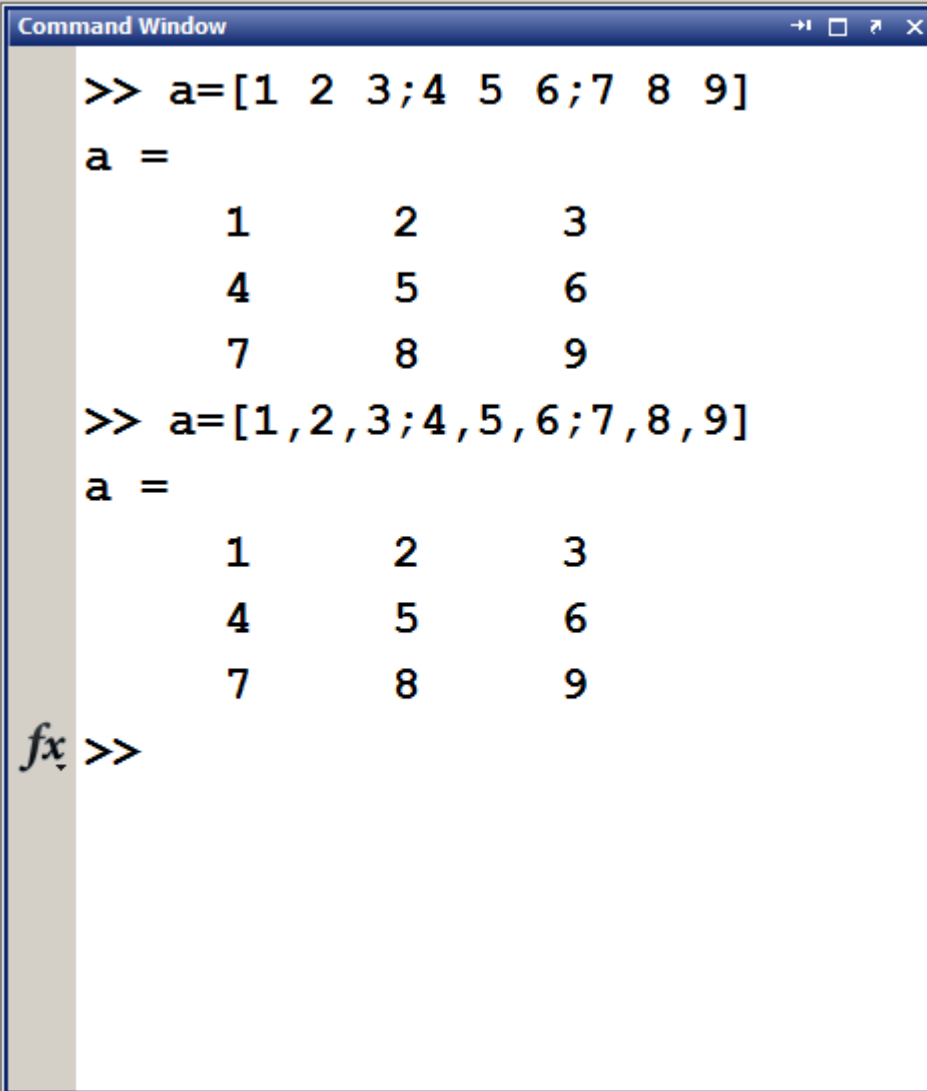
3.1 矩阵的输入

(1) 输入元素列表

矩阵行中的元素以空格或逗号间隔；

矩阵行之间用分号或回车间隔；

整个元素列表用方括号括起来。



```
Command Window
>> a=[1 2 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> a=[1,2,3;4,5,6;7,8,9]
a =
     1     2     3
     4     5     6
     7     8     9
fx >>
```

§ 3 基本操作

3.1 矩阵的输入

(3) 利用MATLAB内部函数与工具箱函数产生矩阵

`eye(m,n)`产生 $m \times n$ 阶单位矩阵

`zeros(m,n)`产生 $m \times n$ 阶全0矩阵

`ones(m,n)`产生 $m \times n$ 阶全1矩阵

`magic(n)`产生 n 阶魔方矩阵

`rand(m,n)`产生 $[0, 1]$ 之间均匀分布的随机矩阵

`randn(m,n)`产生均值为0、方差为1的标准正态分布的随机矩阵

`vander(v)`产生范德蒙德(Vandermonde)矩阵

`diag(v)`产生对角矩阵

§ 3 基本操作

3.1 矩阵

(3) 利用

The image displays the MATLAB 7.10.0 (R2010a) software interface. The Command Window shows the following commands and their outputs:

```
>> eye(2)
ans =
     1     0
     0     1

>> magic(3)
ans =
     8     1     6
     3     5     7
     4     9     2

>> rand(2,3)
ans =
    0.2785    0.9575    0.1576
    0.5469    0.9649    0.9706

>> diag([1 2])
ans =
     1     0
     0     2

>> fliplr(vander([1 2 3]))
ans =
     1     1     1
     1     2     4
     1     3     9
```

Yellow labels on the right side of the Command Window identify the operations:

- 二阶单位矩阵 (2x2 Identity Matrix)
- 三阶魔方矩阵 (3x3 Magic Matrix)
- 2×3随机矩阵 (2x3 Random Matrix)
- 对角矩阵 (Diagonal Matrix)
- 以1、2、3为基础元素的范德蒙矩阵 (Vandermonde Matrix with basis elements 1, 2, 3)

The Workspace window on the right shows the following variables and their values:

Name	Value
a	1.0000 + 2.0000i
ans	[1, 1, 1; 1, 2, 4; 1, 3, 9]
c	31.4159
flag	0
m	56
r	5
s	78.5398
theta	<1x629 double>
x	<1x629 double>
y	<1x10000 double>
y1	<1x629 double>
y2	<1x629 double>

The Command History window at the bottom right shows the sequence of commands entered:

```
eye(2)
magic(3)
rand(2,3)
diag([1 2])
vander(3)
eye(2)
magic(3)
rand(2,3)
diag([1 2])
fliplr(vander([1 2 3]))
```

§ 3 基本操作

3.2 矩阵的操作

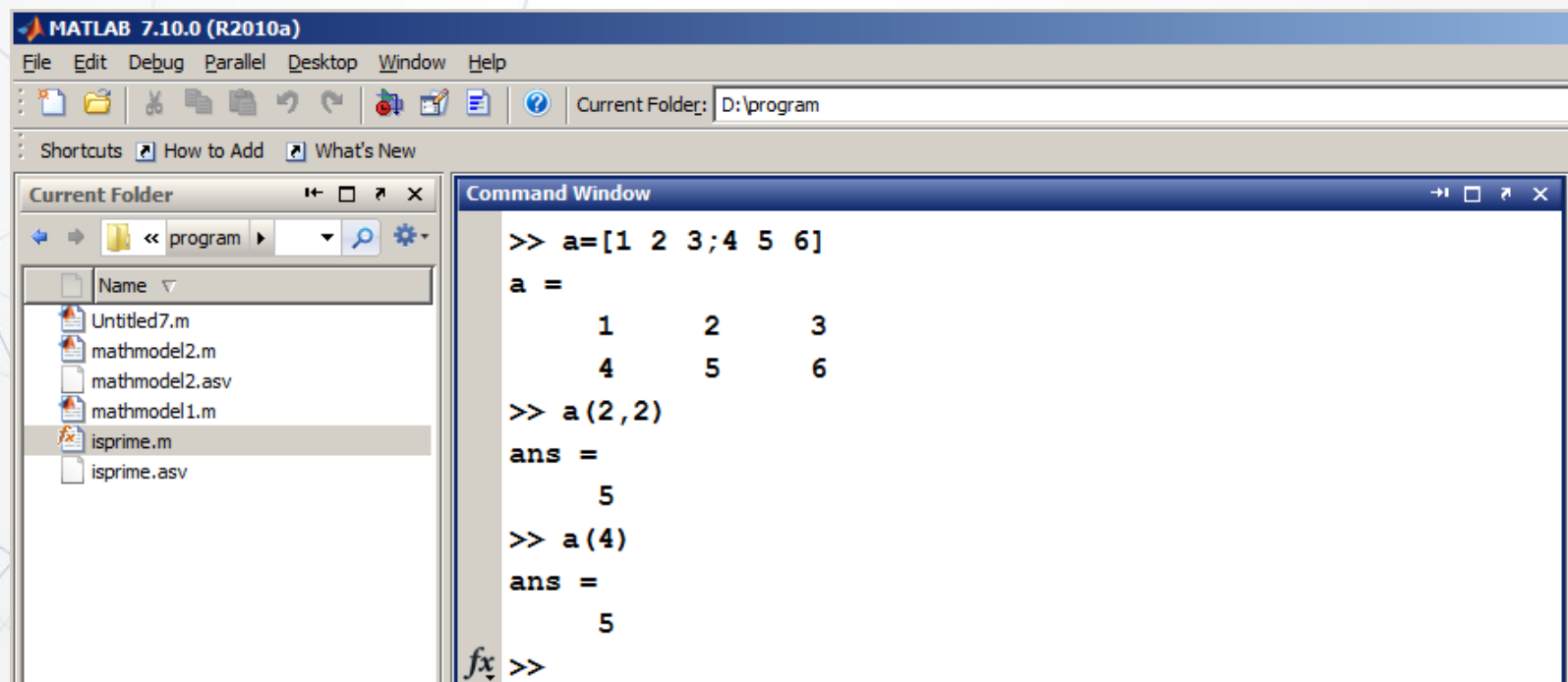
- (1) 矩阵元素的读取
- (2) 矩阵的转置
- (3) 矩阵的扩大
- (4) 矩阵的缩小
- (5) 矩阵的变换
- (6) 矩阵的计算

§ 3 基本操作

3.2 矩阵的操作

(1) 矩阵元素的读取

矩阵 a 中的元素可通过双下标 $a(i,j)$ 或单下标 $a(i)$ 来存取



The image shows the MATLAB 7.10.0 (R2010a) interface. The Command Window displays the following commands and outputs:

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

>> a(2,2)
ans =
     5

>> a(4)
ans =
     5
```

The Current Folder window shows the following files:

- Untitled7.m
- mathmodel2.m
- mathmodel2.asv
- mathmodel1.m
- isprime.m
- isprime.asv

§ 3 基本操作

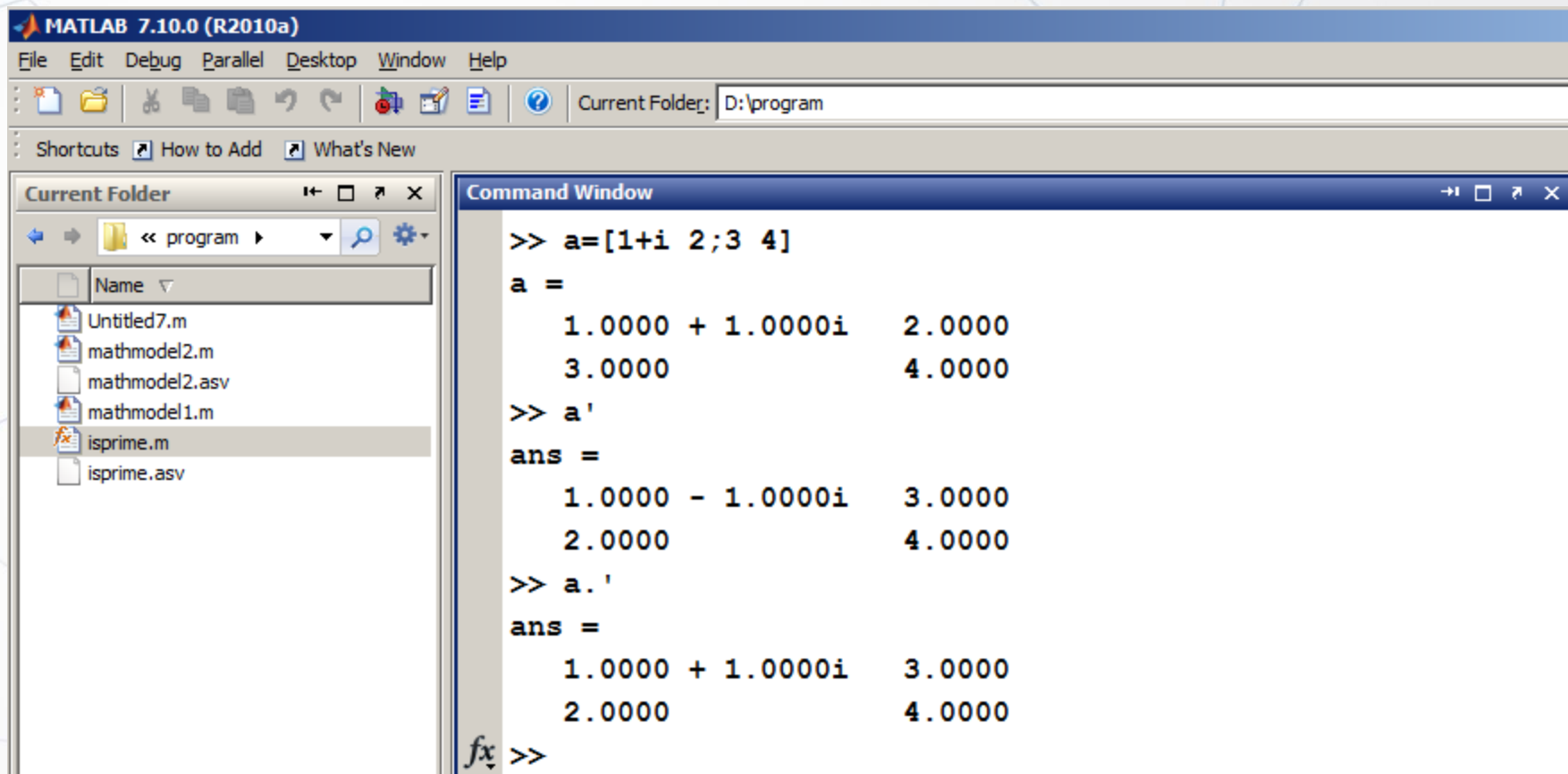
3.2 矩阵的操作

(2) 矩阵的转置

利用转置运算符

a' 共轭转置

$a.'$ 非共轭转置



The image shows the MATLAB 7.10.0 (R2010a) interface. The Command Window displays the following commands and results:

```
>> a=[1+i 2;3 4]
a =
    1.0000 + 1.0000i    2.0000
    3.0000            4.0000
>> a'
ans =
    1.0000 - 1.0000i    3.0000
    2.0000            4.0000
>> a.'
ans =
    1.0000 + 1.0000i    3.0000
    2.0000            4.0000
```

The Current Folder window shows the following files:

- Untitled7.m
- mathmodel2.m
- mathmodel2.asv
- mathmodel1.m
- isprime.m
- isprime.asv

§ 3 基本操作

3.2 矩阵的操作

(3) 矩阵的扩大

MATLAB提供了三种方法来实现这一功能：[连接操作符\[\]](#)、[阵列连接函数cat](#)和[重复函数repmat](#)。

像分块矩阵构造大矩阵一样，通过连接操作符[]将小矩阵连接成大矩阵。

§ 3 基本操作

3.2 矩

(3) 矩

The screenshot displays the MATLAB 7.10.0 (R2010a) environment. The Command Window shows the following commands and their outputs:

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

The Workspace window shows the following variables and their values:

Name	Value
a	[1,2;3,4]
ans	<4x4 double>
b	[5,6;7,8]
c	31.4159
flag	0
m	56
r	5
s	78.5398
theta	<1x629 double>
x	<1x629 double>
y	<1x10000 double>
y1	<1x629 double>

The Command History window shows the following commands:

```
a=[1 2 3;4 5 6]
a(2,2)
a(4)
a=[1+i 2;3 4]
a'
a.'
a=[1 2;3 4]
b=[5 6;7 8]
[a b]
[a;b]
```

The Current Folder window shows the following files:

- Untitled7.m
- mathmodel2.m
- mathmodel2.asv
- mathmodel1.m
- isprime.m
- isprime.asv

The isprime.m (MATLAB Function) window shows the following code:

```
??????
isprime(m)
```

§ 3 基本操作

3.2 矩阵的操作

(4) 矩阵的缩小

将大矩阵变成小矩阵的方法有两种：抽取法和删除法。抽取法是指从大的矩阵中抽取其中的一部分，从而构成新的矩阵；删除法是在原来矩阵中，利用空矩阵[]删除指定的行或列。

§ 3 基本操作

3.2

(4)

The screenshot displays the MATLAB 7.10.0 (R2010a) environment. The Command Window shows the following sequence of commands and their outputs:

```
>> a=[1:5;6:10;11:15]
a =
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15

>> b=a(:,2:4)
b =
     2     3     4
     7     8     9
    12    13    14

>> b(2,:)=[]
b =
     2     3     4
    12    13    14
```

Two yellow labels are placed next to the commands in the Command Window:

- 抽取法** (Extraction Method) is next to the command `b=a(:,2:4)`.
- 删除法** (Deletion Method) is next to the command `b(2,:)=[]`.

The Workspace window on the right shows the following variables and their values:

Name	Value
a	<3x5 double>
ans	<4x4 double>
b	[2,3,4;12,13,14]
c	31.4159
flag	0
m	56

The Command History window at the bottom right shows the following commands:

```
[a b]
[a;b]
[a b;b a]
a=[1:5;6:10;11:15]
b=a(:,2:4)
b(2,:)=[]
```

§ 3 基本操作

3.2 矩阵的操作

(5) 矩阵的变换

rot90(a) 使矩阵a逆时针旋转90度

tril(a) 提取矩阵下三角矩阵

triu(a) 提取矩阵上三角矩阵

fliplr(a) 将矩阵a左右翻转

flipud(a) 将矩阵a上下翻转

```
Command Window
>> a=[1 2 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> rot90(a)
ans =
     3     6     9
     2     5     8
     1     4     7
>> tril(a)
ans =
     1     0     0
     4     5     0
     7     8     9
>> fliplr(a)
ans =
     3     2     1
     6     5     4
     9     8     7
fx >>
```

§ 3 基本操作

3.2 矩阵的操作

(6) 矩阵的计算

size(a)返回a的大小

det(a)计算a的行列式

rank(a)返回a的秩

trace(a)返回a的迹

inv(a)输出a的逆矩阵

pinv(a)输出a的伪逆矩阵

rref(a)返回a的行最简形矩阵

poly(a)输出a的特征多项式

[vec,val]=eig(a)输出a的特征值和特征向量

orth(a)输出a的正交矩阵

```
Command Window
>> a=[1 2 3;4 5 6;2 4 6]
a =
     1     2     3
     4     5     6
     2     4     6
>> size(a)
ans =
     3     3
>> rank(a)
ans =
     2
>> rref(a)
ans =
     1     0    -1
     0     1     2
     0     0     0
>> [vec,val]=eig(a)
vec =
    -0.3067    -0.4082     0.2471
    -0.7277     0.8165    -0.8334
    -0.6135    -0.4082     0.4943
val =
    11.7446         0         0
         0     0.0000         0
         0         0     0.2554
fx >>
```


§ 3 基本操作

3.3 基本数学运算

abs(x) 绝对值

sqrt(x) 平方根

round(x) 四舍五入取整

rem(x,y) 余数

gcd(x,y) 最大公约数

lcm(x,y) 最小公倍数

max(x) 最大值

min(x) 最小值

sum(x) 求和

log10(x) 常用对数

log(x) 自然对数

exp(x) 自然指数

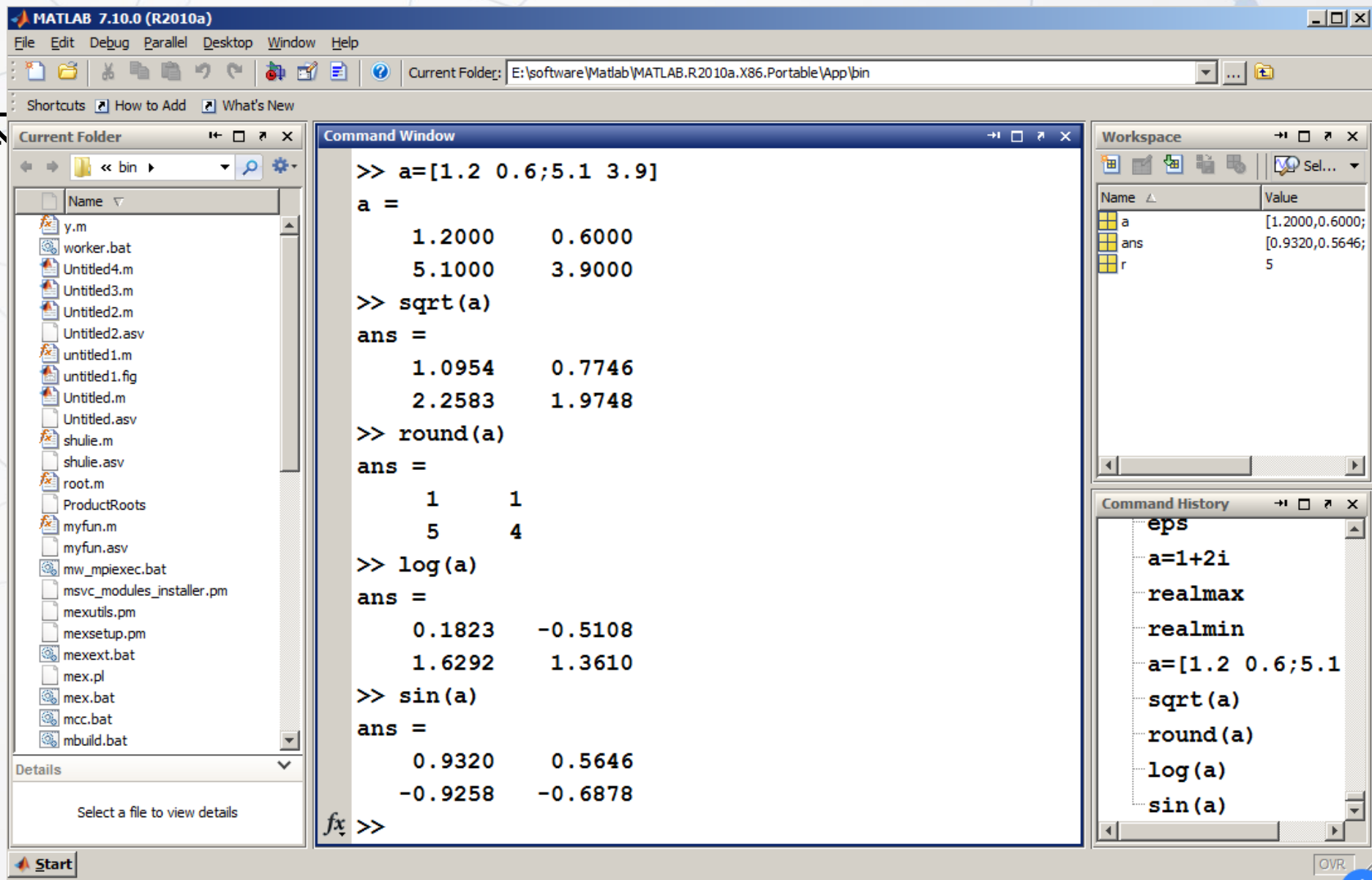
sin(x)、cos(x)、tan(x)、cot(x)、sec(x)、csc(x) 三角函数 (角度单位为弧度)

asin(x)、acos(x)、atan(x) 反三角函数

sinh(x)、cosh(x)、tanh(x)、coth(x) 双曲函数

§ 3 基本操作

3.3 基本



第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 4 绘图功能

4.1 二维图形

4.2 高级绘图函数

§ 4 绘图功能

4.1 二维图形

- (1) 绘图步骤
- (2) 常用函数
- (3) 相关函数

§ 4 绘图功能

4.1 二维图形

(1) 绘图步骤

- 1) 产生x轴、y轴数据
- 2) 打开一个新的图形窗口
- 3) 利用二维曲线绘图函数绘制曲线

§ 4 绘图功能

4.1 二维图形

(2) 常用函数

`plot(x,y,'s')` 绘制以向量`x,y` 的元素为横、纵坐标的曲线

`plot(x, 's')` 绘制以向量`x` 的元素下标为横坐标、`x` 的元素为纵坐标的曲线

`plot(x1,y1, 's1',x2,y2, 's2',...)` 在一个坐标系中绘制多条曲线

以上命令中，`s,s1,s2` 为参数，可缺省，包括线型（如实线-、虚线--、点线：、点划线- .）、颜色（如红色r、绿色g、黄色y、蓝色b）、标记符号（如点.、星号*、圆圈o、加号+）等，详见`plot`帮助文件。

§ 4 绘图功能

4.1 二维图形

(2) 常用函数plot

```
Command Window
>> help plot
PLOT Linear plot.
PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
then the vector is plotted versus the rows or columns of the matrix,
whichever line up. If X is a scalar and Y is a vector, disconnected
line objects are created and plotted as discrete points vertically at
X.

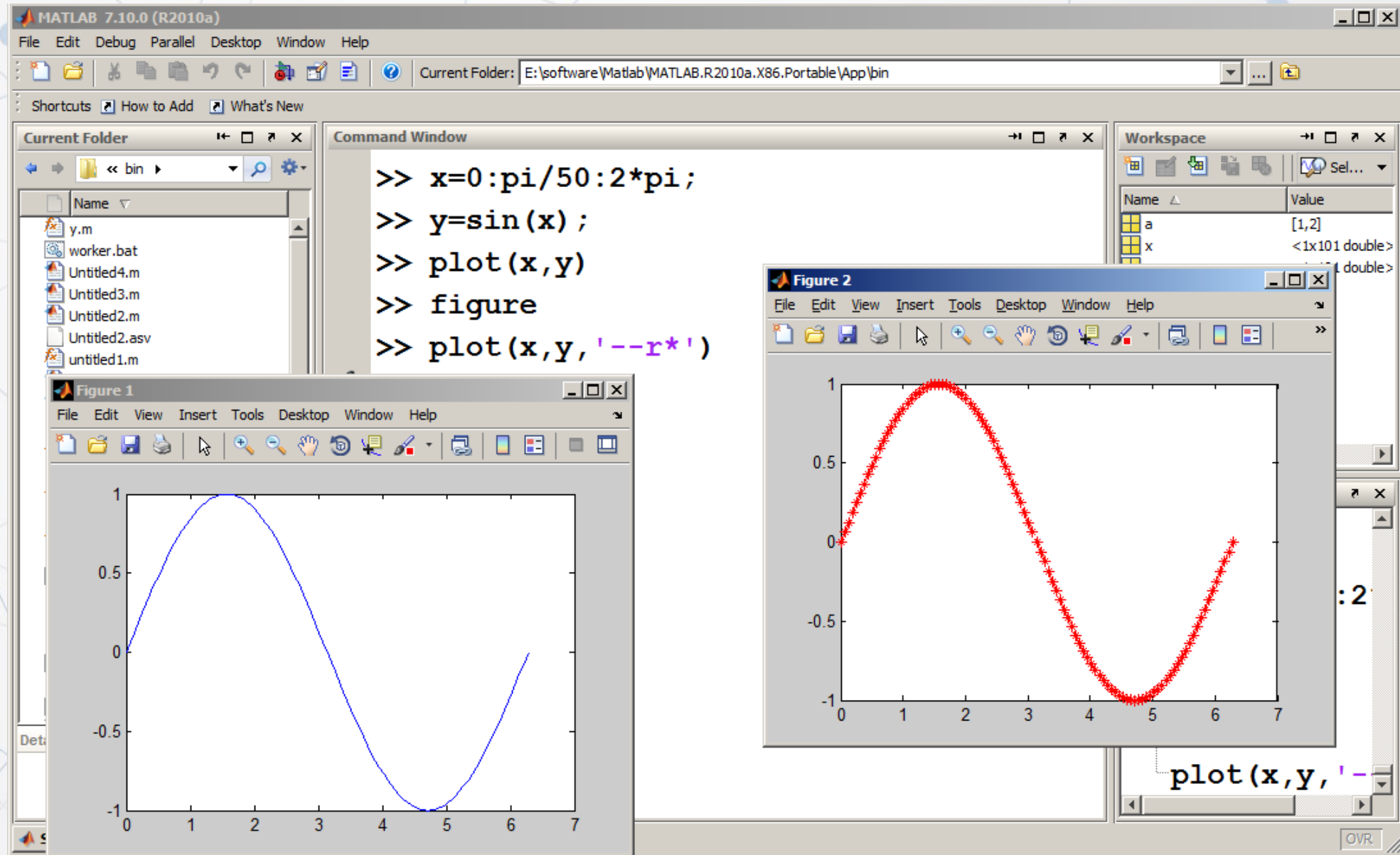
PLOT(Y) plots the columns of Y versus their index.
If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
In all other uses of PLOT, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with
PLOT(X,Y,S) where S is a character string made from one element
from any or all the following 3 columns:

      b      blue      .      point      -      solid
      g      green     o      circle     :      dotted
      r      red       x      x-mark    -.     dashdot
      c      cyan      +      plus       --     dashed
      m      magenta   *      star       (none) no line
      y      yellow    s      square
      k      black     d      diamond
      w      white     v      triangle (down)
                        ^      triangle (up)
                        <      triangle (left)
                        >      triangle (right)
                        p      pentagram
                        h      hexagram

For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus
at each data point; PLOT(X,Y,'bd') plots blue diamond at each data
```

§ 4 绘图功能



§ 4 绘图功能

4.1 二维图形

(3) 相关函数

`subplot(m,n,p)` 将图形窗口分割成 $m \times n$ 个窗格，设置第 p 个窗格为当前窗格

`title('string')` 给当前坐标系图形加上标题

`xlabel('string')`, `ylabel('string')`, `zlabel('string')` 添加 x 、 y 、 z 轴的标记

`legend('string1','string2',...)` 在坐标区添加图例

`text(x,y, 'string')` 在图形的 (x,y) 处添加说明

§ 4 绘图功能

4.1 二维图形

(3) 相关函数

`loglog(x,y)` 在对数坐标系中绘制

`semilogx(x,y)`, `semilogy(x,y)` 在半对数坐标系中绘制图形

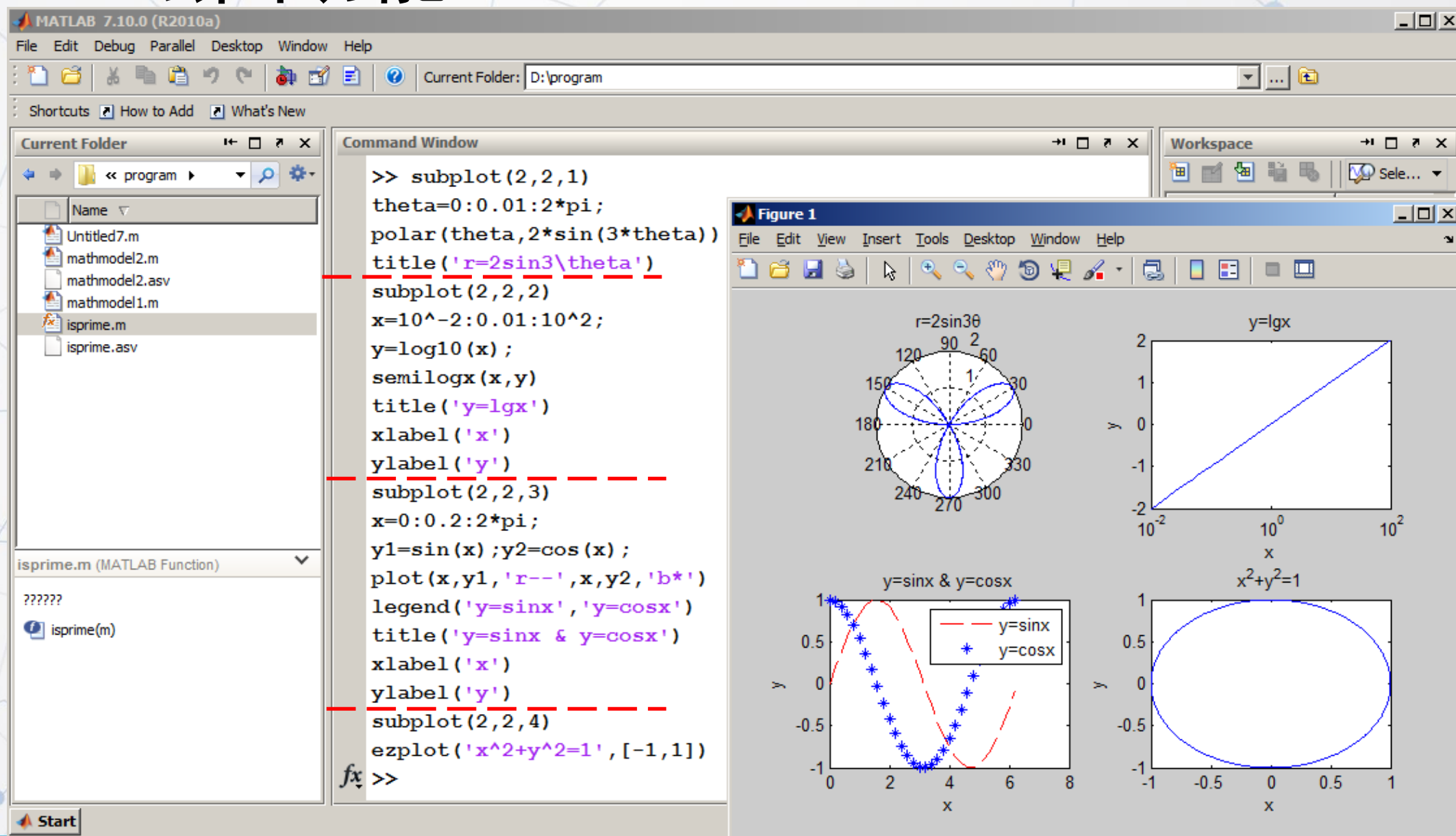
`polar(theta,r)` 绘制极坐标 $r = r(\theta)$ 的图像

`ezplot` 无需数据准备，直接画出函数的图形，可用于绘制参数方程或隐函数所确定的函数的图像

§ 4 绘图功能

4.1

(3)



§ 4 绘图功能

4.1 二维图形

4.2 高级绘图函数

(1) 三维曲线

(2) 三维曲面

(3) 其它函数

§ 4 绘图功能

4.2 高级绘图函数

(1) 三维曲线

`plot3(x,y,z,'s')` 绘制以向量`x,y,z` 的元素为横、纵、竖坐标的三维曲线

`plot3(x1,y1,z1,'s1',x2,y2,z2,'s2',...)` 在一个坐标系中绘制多条曲线

以上命令中，`s,s1,s2` 为参数，与`plot` 命令中用法相同。

§ 4 绘图功能

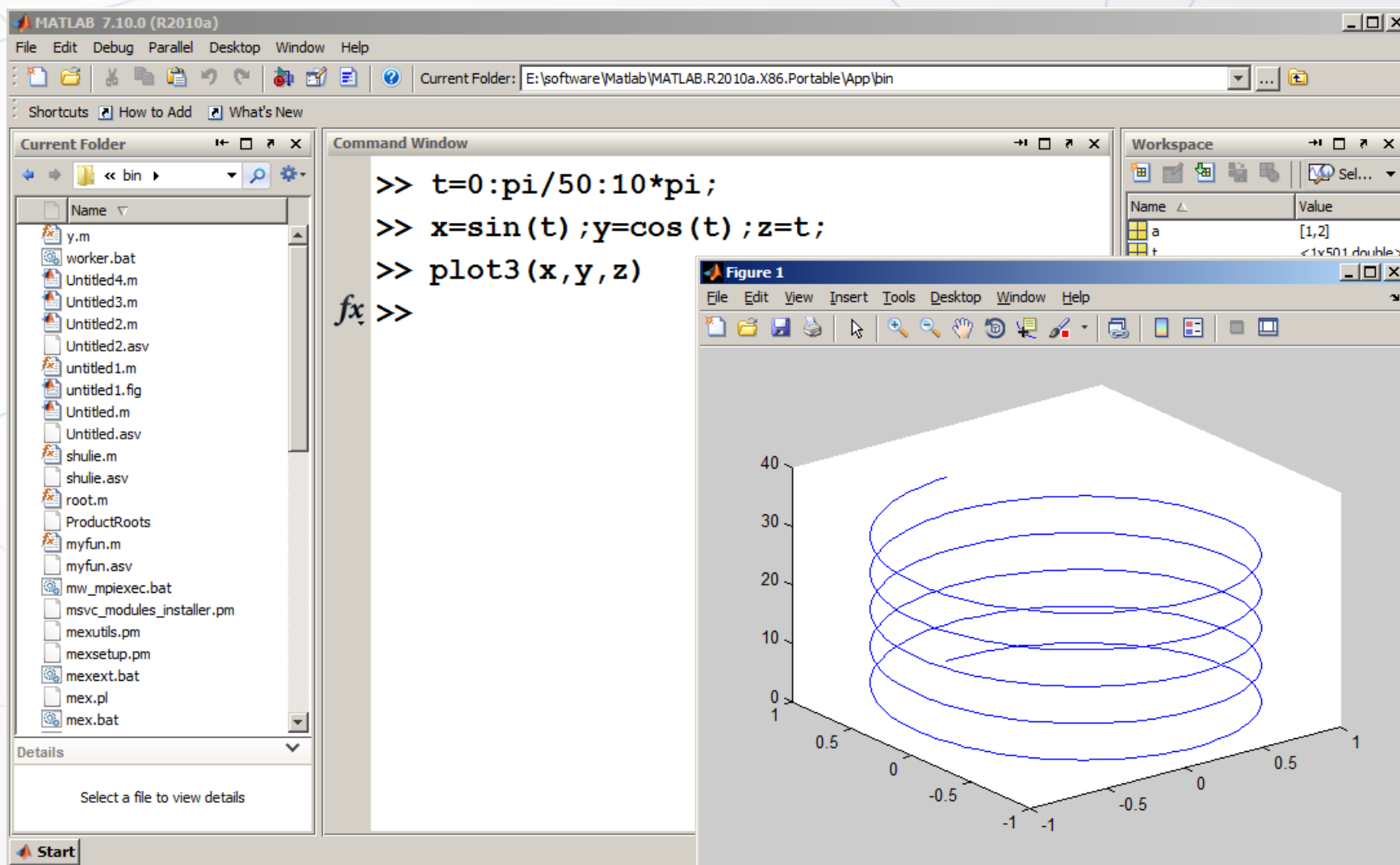
4.2 高级绘图函数

(1) 三维曲线

例 绘制螺旋线

$$\begin{cases} x = \sin t \\ y = \cos t \ (t \in [0, 10\pi]) \\ z = t \end{cases}$$

的图像。



§ 4 绘图功能

4.2 高级绘图函数

(2) 三维曲面

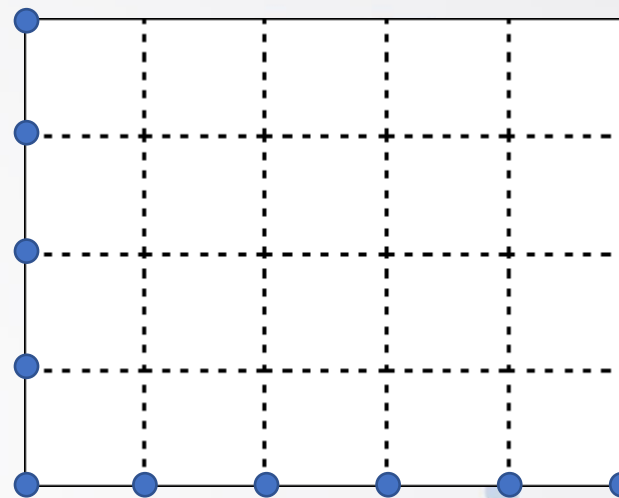
$[X,Y]=\text{meshgrid}(x,y)$ 基于向量 x 和 y 中包含的坐标返回二维网格坐标。

(此为绘制三维曲面所必需)

$\text{surf}(X,Y,Z)$ 绘制以矩阵 X,Y,Z 的元素为横、纵、竖坐标的三维曲面

$\text{mesh}(X,Y,Z)$ 绘制以矩阵 X,Y,Z 的元素为横、纵、竖坐标的三维网格曲面

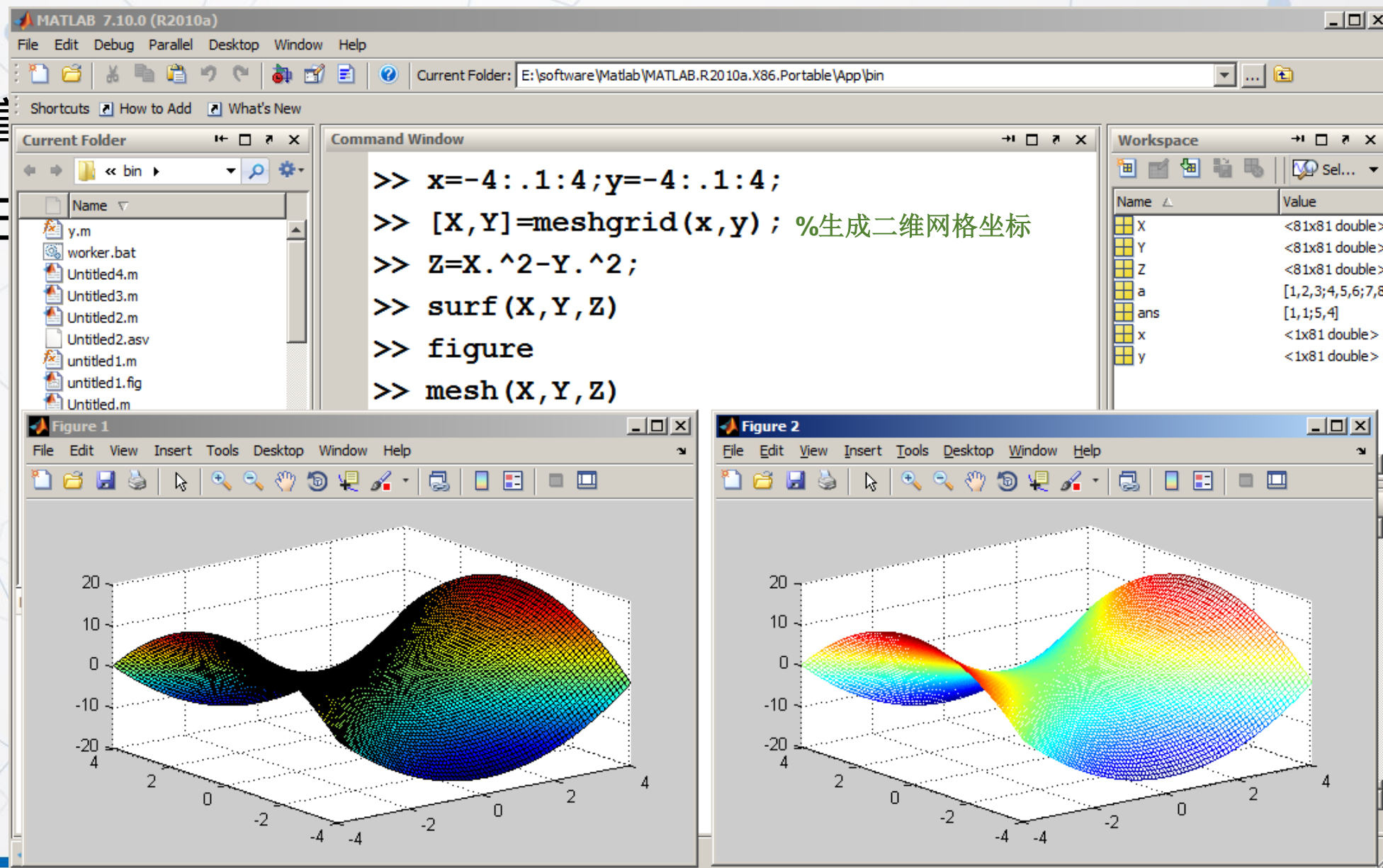
例 绘制马鞍面的图像 $z=x^2-y^2$



§ 4 绘图功能

4.2

(2)



§ 4 绘图功能

4.2 高级绘图函数

(3) 其它函数

bar 绘制条形图

pie 绘制饼图

area 二维图形填充区域

stairs 绘制梯形图

hist 绘制柱状图

rose 绘制角度的柱状图

cylinder 绘制柱面图

sphere 绘制球形图

ellipsoid 绘制椭球体

polyarea 绘制多边形

fill 填充二维多边形

ribbon 绘制带状图

contour 绘制矩阵的等高线

stem 绘制离散序列数据

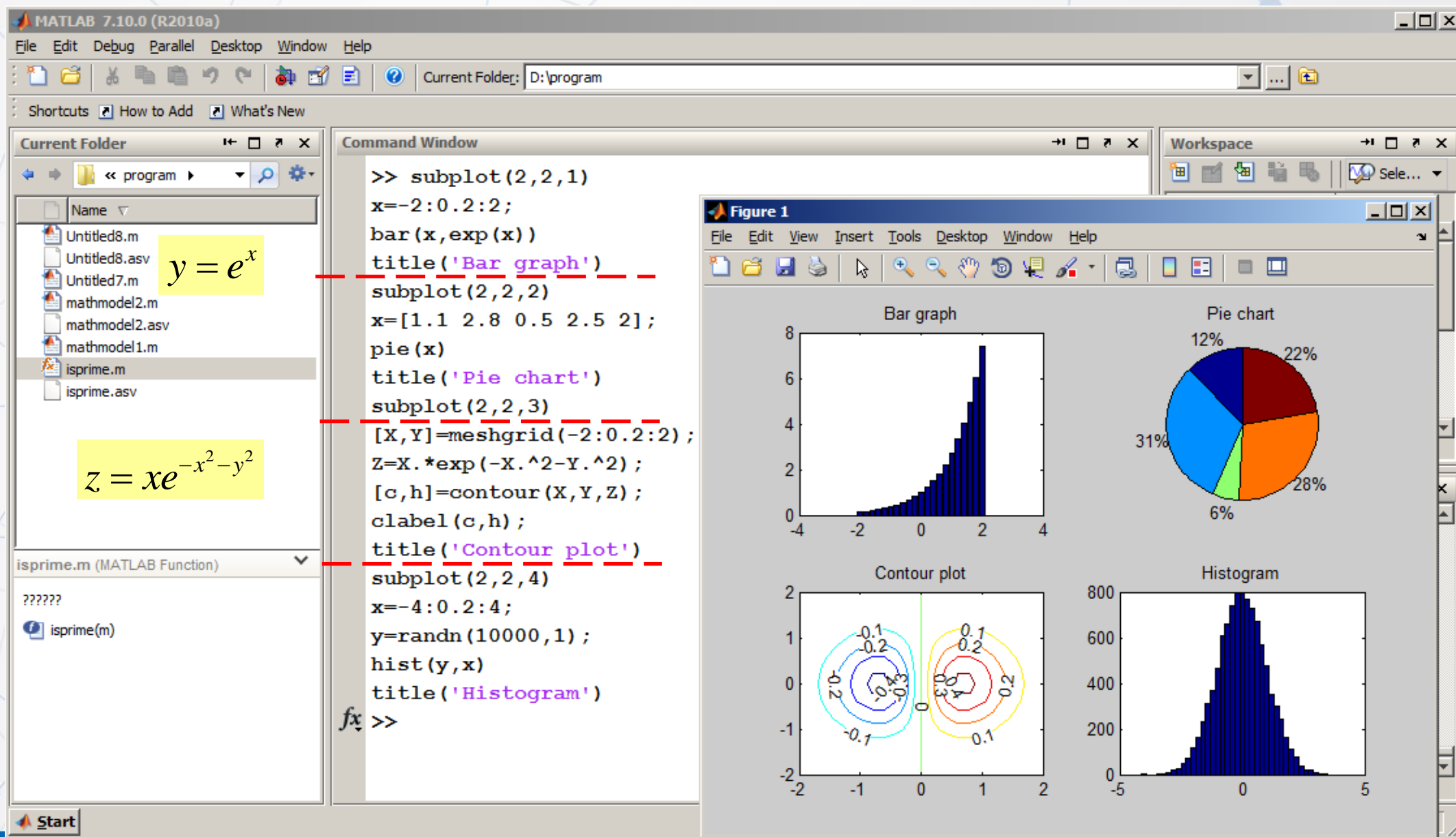
plotmatrix 绘制矩阵的散布图

scatter 绘制散布图

§ 4 绘图功能

4.2

(3)



第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 5 基本应用

5.1 导数、极限、积分

5.2 代数方程(组)

5.3 微分方程(组)

5.4 拟合与回归

5.5 插值

5.6 其它

§ 5 基本应用

5.1 导数、极限、积分

(1) 导数

`diff(f,x,n)`

(2) 极限

`limit(f,x,a)`

```
Command Window
>> syms a x y;
>> f=a*x^2;
>> diff(f,x)%求导数
ans =
2*a*x
>> diff(f,x,2)%求二阶导数
ans =
2*a
>> f=x*y/(x^2+y^2);
>> diff(f,x)%求偏导数
ans =
y/(x^2 + y^2) - (2*x^2*y)/(x^2 + y^2)^2
>> diff(f,x,2)%求二阶偏导数
ans =
(8*x^3*y)/(x^2 + y^2)^3 - (6*x*y)/(x^2 + y^2)^2
>> limit((x^2-1)/(x-1),x,1)%求极限
ans =
2
>> limit(tan(x),x,pi/2,'left')%求左极限
ans =
Inf
>> f=(x*y+1)/(x+y^2);
>> limit(limit(f,x,1),y,0)%求二元函数的极限
ans =
1
fx >>
```

已知 $f = ax^2$, 求 f' 和 f''

已知 $f = \frac{xy}{x^2 + y^2}$, 求 f'_x 和 f''_{xx}

$\lim_{x \rightarrow 1} \frac{x^2 - 1}{x - 1}$

$\lim_{x \rightarrow \frac{\pi^-}{2}} \tan x$

$\lim_{\substack{x \rightarrow 1 \\ y \rightarrow 0}} \frac{xy + 1}{x + y^2}$

§ 5 基本应用

5.1 导数、极限、积分

(3) 积分

`int(f,x)`

不定积分□

`int(f,x,a,b)`

定积分 (a, b 可为无穷大)

`int(int(f,y,c,d),x,a,b)`

二重积分 (二次积分形式)

`trapz(x,y)`

已知函数的离散数据点 (x,y) , 求定积分

`quad(fun,a,b,tol)`

求定积分 , tol 为容许误差 , 可缺省

`dblquad(fun,a,b,c,d)`

矩形区域上的二重积分

§ 5 基本应用

5.1

(3)

MATLAB 7.10.0 (R2010a)

File Edit Debug Parallel Desktop Window Help

Current Folder: D:\program

Shortcuts How to Add What's New

Current Folder

Name

- Untitled8.m
- Untitled8.asv
- Untitled7.m
- mathmodel2.m
- mathmodel2.asv
- mathmodel1.m
- isprime.m
- isprime.asv

isprime.m (MATLAB Function)

??????

isprime(m)

Command Window

```
>> syms x y;  
>> q1=int('sin(2*x)',x) %求不定积分  
q1 =  
sin(x)^2 - 1/2  
>> q2=int('exp(sqrt(x))',x,0,1) %求定积分  
q2 =  
2  
>> f=x*sin(y);  
>> q3=int(int(f,y,x,sqrt(x)),x,0,1) %求二重积分  
q3 =  
7*cos(1) + 11*sin(1) - 13  
>> clear  
>> x=0:pi/100:pi;y=sin(x);  
>> q4=trapz(x,y) %由函数的离散数据点(x,y),求定积分  
q4 =  
1.9998  
>> q5=quad(@(x)exp(-x.^2/2),0,1) %由函数表达式求定积分  
q5 =  
0.8556  
>> f=@(x,y)y*sin(x)+x*cos(y); %定义二元函数  
>> q6=dblquad(f,pi,2*pi,0,pi) %矩形区域上的二重积分  
q6 =  
-9.8696  
fx >>
```

Workspace

Name Value

f	@(x,y)y*sin(x)+x
q4	1.9998
q5	0.8556
q6	-9.8696
x	<1x101 double>
y	<1x101 double>

$\int \sin 2x dx$

$\int_0^1 e^{\sqrt{x}} dx$

$\int_0^1 \int_x^{\sqrt{x}} x \sin y dy dx$

由 $y = \sin x$ 的离散数据点求 $\int_0^\pi \sin x dx$

$\int_0^1 e^{-\frac{x^2}{2}} dx$

$\int_0^\pi \int_\pi^{2\pi} (y \sin x + x \cos y) dx dy$

§ 5 基本应用

5.2 代数方程(组)

- (1) 求解代数方程(组)的主要命令
- (2) 求解线性方程组的相关命令

§ 5 基本应用

5.2 代数方程(组)

(1) 求解代数方程(组)的主要命令

`roots(p)` 求 p 表示的多项式的根（ p 为包含多项式系数的行向量）

`fzero(f,x0)` 求一元函数 f 在点 x_0 附近的根（ x_0 可为区间）

`solve(f,x)` 以 x 为变量求解代数方程 f

`solve(f1,f2,...,x1,x2,...)` 以 $x_1,x_2,...$ 为变量求解代数方程组 $f_1,f_2,...$

§ 5 基本应用

5.2

(1)

求解
代数
方程

The image shows the MATLAB 7.10.0 (R2010a) interface. The Command Window contains the following code and results:

```
>> p=[1 0 3 -1];  
>> roots(p) %求多项式方程x^3+3x-1=0的根  
ans =  
    -0.1611 + 1.7544i  
    -0.1611 - 1.7544i  
     0.3222  
>> fzero('exp(x)-x-3',[1 2])  
ans =  
    1.5052  
>> solve('a*x^2 + b*x + c','x') %解一元二次方程ax^2+bx+c=0  
ans =  
    -(b + (b^2 - 4*a*c)^(1/2))/(2*a)  
    -(b - (b^2 - 4*a*c)^(1/2))/(2*a)  
>> s=solve('x-3*y=sin(x)','2*x+y=cos(y)'); %求解代数方程组  
>> s=[s.x s.y]  
s =  
[ 0.49662797440907460178544085171994, 0.0067214622395756734146654770697884]  
fx >>
```

Yellow boxes highlight the following mathematical problems:

- 求 $x^3 + 3x - 1 = 0$ 的根
- 求 $e^x = x + 3$ 在区间 $(1, 2)$ 内的根
- 求 $ax^2 + bx + c = 0$ 的根
- 求解 $\begin{cases} x - 3y = \sin x \\ 2x + y = \cos y \end{cases}$

The Workspace window on the right shows variables: a, ans, b, c, f, p, q4, n5. The Command Window also shows the result of the solve function for the system of equations.

§ 5 基本应用

5.2 代数方程(组)

(2) 求解线性方程组的相关命令

`null(A,'r')` 求齐次线性方程组 $Ax=0$ 的基础解系, $r=\text{rank}(A)$

`A\b`、`inv(A)*b` 求非齐次线性方程组 $Ax=b$ 的惟一解,

其中 `inv(A)` 为非奇异矩阵 A 的逆矩阵

`pinv(A)*b` 求有无穷多解的非齐次线性方程组 $Ax=b$ 的某一特解,

其中 `pinv(A)` 为奇异矩阵 A 的伪逆矩阵

§ 5 基本应用

5.2 代数

(2) 求解

$ax=b$

Example 1: Unique Solution

```
>> a=[1 1 1;2 1 2;1 1 0];
>> b=[2 2 3]';
>> [rank(a) rank([a b])]
ans =
     3     3
>> x=a\b%或x=inv(a)*b 求方程组的唯一解
x =
     1
     2
    -1
```

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ 2x_1 + x_2 + 2x_3 = 2 \\ x_1 + x_2 = 3 \end{cases}$$

Example 2: Infinite Solutions

```
>> a=[1 1 1;2 2 2;2 2 -1];
>> b=[2 4 5]';
>> [rank(a) rank([a b])]
ans =
     2     2
>> null(a,'r')%求导出组的基础解系
ans =
    -1
     1
     0
>> pinv(a)*b%求方程组的特解
ans =
     1.1667
     1.1667
    -0.3333
```

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + 2x_3 = 4 \\ 2x_1 + 2x_2 - x_3 = 5 \end{cases}$$

Workspace:

Name	Value
a	[1,1,1;2,2,2;2,2,-1]
ans	[1.1667;1.1667;-0.3333]
b	[2;4;5]
c	<1x1 sym>
f	@(x,y)y*sin(x)+x*cos
p	[1,0,3,-1]
q4	1.9998
q5	0.8556
q6	-9.8696
s	<1x2 sym>
x	[1;2;-1]
y	<1x101 double>

Command History:

```
a=[1 1 1;2 2 2;2 2 -1];
b=[2 4 5]';
[rank(a) rank([a b])]
null(a,'r')%求导出组的基础解系
pinv(a)*b%求方程组的特解
a=[1 1 1;2 1 2;1 1 0];
b=[2 2 3]';
[rank(a) rank([a b])]
```

§ 5 基本应用

5.1 导数、极限、积分

5.2 代数方程(组)

5.3 微分方程(组)

5.4 拟合与回归

5.5 插值

5.6 其它

§ 5 基本应用

5.3 微分方程(组)

- (1) 求微分方程(组)的解析解
- (2) 求微分方程(组)的数值解

解析解是一些严格的公式，给出任意的自变量就可以求出其因变量，也就是问题的解，他人可以利用这些公式计算各自的问题；数值解是采用某种计算方法，如有限差分、有限元、数值逼近等方法，得到的解。别人只能利用数值计算的结果，而不能随意给出自变量并求出计算值。

§ 5 基本应用

5.3 微分方程(组)

(1) 求微分方程(组)的解析解

`dsolve ('eqn1','eqn2', ..., 'cond1', 'cond2',..., 'val')`

'eqn1','eqn2', ..., 为微分方程(组)

'cond1', 'cond2', ..., 为初始条件

'val'为独立变量，可缺省，缺省独立变量为t

§ 5 基本应用

5.3 微分方程(组)

(1) 求微分方程(组)的

$$\begin{cases} y'' - 4y' + 3y = 8e^{5x} \\ y(0) = 3 \\ y'(0) = 9 \end{cases}$$

```
Command Window
>> dsolve('Dy = -a*y') %求一阶微分方程的通解 Dy为y的导数
ans =
C2/exp(a*t)
>>
dsolve('D2y-4*Dy+3*y=8*exp(5*x)', 'y(0)=3', 'Dy(0)=9', 'x') %
二阶微分方程的特解 D2y为y的二阶导数
ans =
exp(3*x) + exp(5*x) + exp(x)
>>
[x,y]=dsolve('Dy=2*y+3*x', 'Dx=y-2*x', 'x(0)=2', 'y(0)=1', 't') %求微分方程组的解
x =
(7^(1/2)*exp(7^(1/2)*t)*(2*7^(1/2) - 3))/14 +
(7^(1/2)*(2*7^(1/2) + 3))/(14*exp(7^(1/2)*t))
y =
(exp(7^(1/2)*t)*(2*7^(1/2) - 3))/2 - (2*7^(1/2) +
3)/(2*exp(7^(1/2)*t)) +
(7^(1/2)*exp(7^(1/2)*t)*(2*7^(1/2) - 3))/7 +
(7^(1/2)*(2*7^(1/2) + 3))/(7*exp(7^(1/2)*t))
fx >>
```

$$\frac{dy}{dx} = -ay$$

$$\begin{cases} \frac{dy}{dt} = 2y + 3x \\ \frac{dx}{dt} = y - 2x \\ x(0) = 2 \\ y(0) = 1 \end{cases}$$

§ 5 基本应用

5.3 微分方程(组)

(2) 求微分方程(组)的数值解

`ode45` , `ode23` , `ode113` , `ode15s` , `ode23s` , `ode23t` , `ode23tb`

`[T,Y] = ode45(odefun,tspan,y0)`

`odefun` 是函数句柄，可以是函数文件名、匿名函数句柄或内联函数名；

`tspan` 是区间 `[t0 tf]` 或者一系列散点 `[t0,t1,...,tf]`；`y0` 是初始值向量；

`T` 返回列向量形式的时间点；

`Y` 返回对应`T`中时间点的微分方程(组)的解向量

§ 5 基本应用

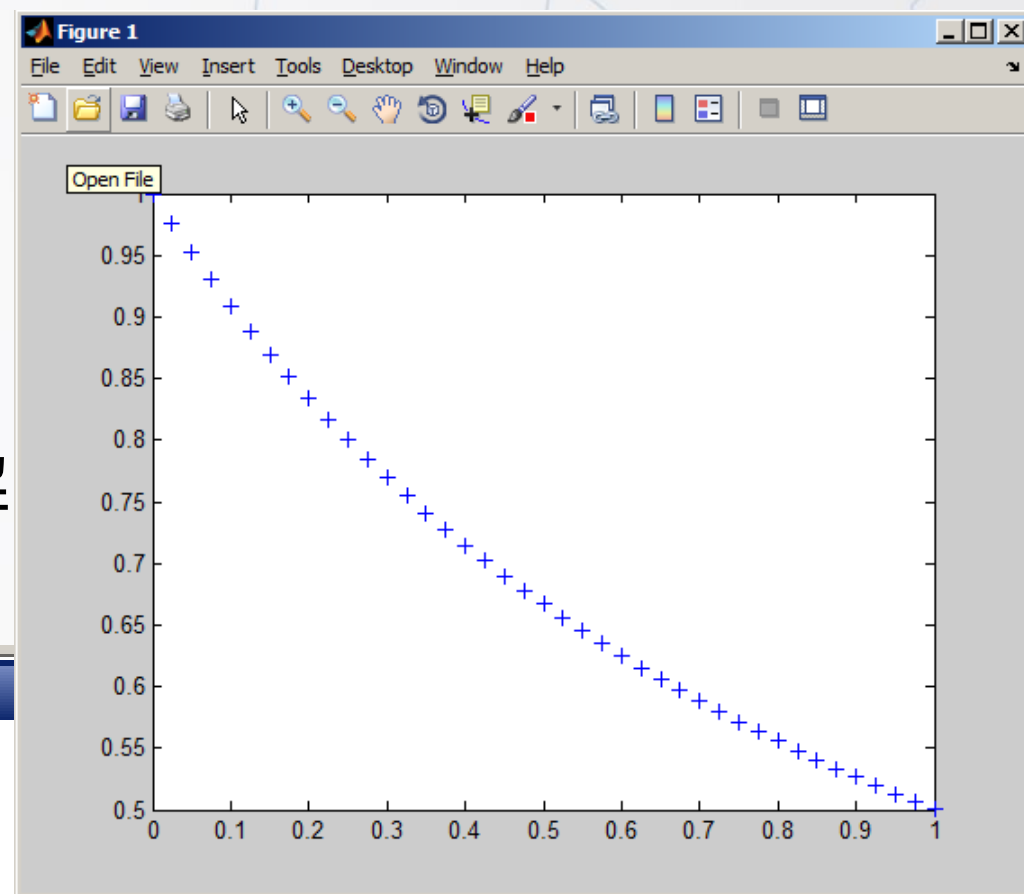
5.3 微分方程(组)

(2) 求微分方程(组)的数值解

例 求微分方程 $\begin{cases} y' = -y^2 \\ y(0) = 1 \end{cases} (0 \leq x \leq 1)$ 的数值解

Command Window

```
>> func=@(x,y) -y^2;  
>> [x,y]=ode45(func,[0 1],1);%求解微分方程  
>> plot(x,y, '+') %绘制数值解图像  
fx >>
```

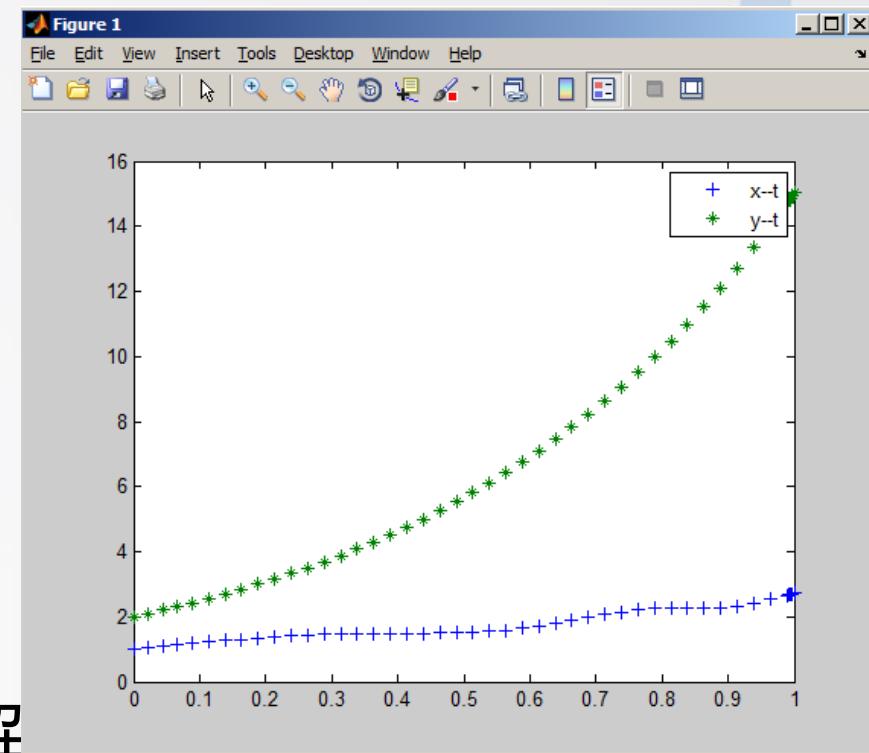


§ 5 基本应用

5.3 微分方程(组)

(2) 求微分方程(组)的数值解

例 求微分方程组
$$\begin{cases} x'(t) = x \sin y + x \\ y'(t) = \cos x + 2y \\ x(0) = 1 \\ y(0) = 2 \end{cases} \quad (0 \leq t \leq 1)$$
 的数值解



Command Window

```
>> func=@(t,z)[z(1)*sin(z(2))+z(1);cos(z(1))+2*z(2)];%z=[x y]
>> [t,z]=ode45(func,[0,1],[1;2]);%求解微分方程组的数值解
>> plot(t,z(:,1),'+',t,z(:,2),'*') %绘制数值解图像
>> legend('x--t','y--t') %添加图例
fx >>
```

§ 5 基本应用

5.1 导数、极限、积分

5.2 代数方程(组)

5.3 微分方程(组)

5.4 拟合与回归

5.5 插值

5.6 其它

§ 5 基本应用

5.4 拟合与回归

拟合是数据处理和数值计算的一种重要方法，它要求用一条相对光滑的曲线来近似地描述给定的一组数据点满足的函数关系。具体提法：设给定的一组数据点 $(x_i, y_i) (i=1, \dots, n)$ 近似地满足函数关系 $y=f(x)$ ，试确定 $y=f(x)$ 的具体形式。这里 $y=f(x)$ 称为拟合函数或经验公式，不要求它经过每一个数据点，只需使之与各数据点之间的距离尽可能小即可，其具体形式可经由经验、散点图或数学建模等确定。

§ 5 基本应用

回归是统计分析的一种重要方法，用来确定两种或两种以上变量间相互依赖的定量关系。回归分析按照涉及的变量的多少，分为一元回归和多元回归分析；按照自变量和因变量之间的关系类型，可分为线性回归分析和非线性回归分析。如果在回归分析中，只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括一个因变量和两个以上的自变量，且自变量之间存在线性相关，则称为多元线性回归分析。

§ 5 基本应用

5.4 拟合与回归

$p = \text{polyfit}(x, y, n)$ n 次多项式拟合($n=1$ 时为线性拟合), p 为多项式系数向量

$[b, \text{bint}, r, \text{rint}, \text{stats}] = \text{regress}(y, X, \alpha)$ 多元线性回归分析

(y 为因变量数据向量, X 为自变量 x 和一系列具有相同行数, 值是1的向量构成的数据矩阵, α 为显著性水平, 可缺省; b 为拟合出来的多元线性模型的系数向量, bint 为参数的置信区间, r 为残差, rint 为残差的置信区间, stats 包括决定系数、F统计量的值、接受概率和剩余方差, bint 、 r 、 rint 、 stats 可缺省)

$\text{beta} = \text{nlinfit}(x, y, f, \text{beta0})$ 非线性回归分析, f 为函数, beta0 为 f 中参数初值

§ 5 基本应用

The image shows the MATLAB 7.10.0 (R2010a) interface. The Command Window displays the following code and output:

```
>> format short
>> x=[1 2 3 4 5 6 7 8 9 10];%导入数据
>> y=[3.21 2.1 4.8 8.7 5.8 8.32 7.65 5.43 3.32 2.76];
>> y1=polyfit(x,y,7) %7 次多项式拟合
y1 =
    Columns 1 through 7
   -0.0006    0.0266   -0.4721    4.3792   -22.6858    64.3874   -88.4515
    Column 8
    46.0660
```

The Workspace window shows the following variables:

Name	Value
X	<14x6 double>
b	[-1.3539e...
bint	<6x2 double>
h	<1x12 double>
hi	<1x45 double>
r	<14x1 double>

The Command History window shows the following commands:

```
y=[3.21 2.1 4.8 8.7 5.8 8.32 7.65 5.43 3.32 2.76];
y1=polyfit(x,y,7)
format short
x=[1 2 3 4 5 6 7 8 9 10];
y=[3.21 2.1 4.8 8.7 5.8 8.32 7.65 5.43 3.32 2.76];
y1=polyfit(x,y,7)
```

拟合函数

$$y = -0.0006x^7 + 0.0266x^6 - 0.4721x^5 + 4.3792x^4 - 22.6858x^3 + 64.3874x^2 - 88.4515x + 46.0660$$

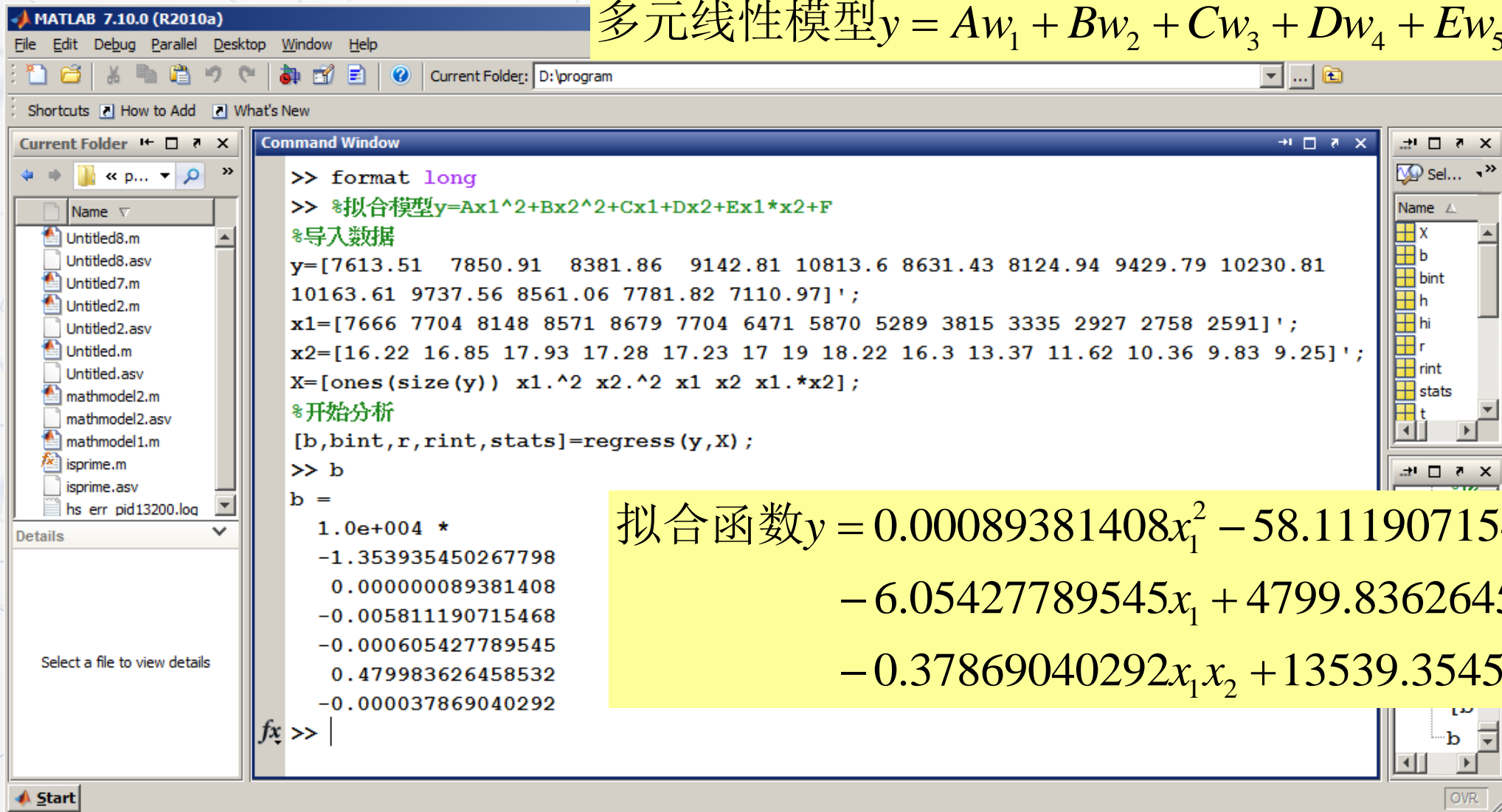
§ 5 基本应用

拟合模型

$$y = Ax_1^2 + Bx_2^2 + Cx_1 + Dx_2 + Ex_1x_2 + F$$

多元线性模型

$$y = Aw_1 + Bw_2 + Cw_3 + Dw_4 + Ew_5 + F$$



The screenshot shows the MATLAB 7.10.0 (R2010a) Command Window. The script performs a regression analysis on a dataset. The data is loaded into variables y, x1, and x2. A design matrix X is constructed using ones, x1 squared, x2 squared, x1, x2, and their product. The regress function is used to fit the model, and the coefficients b are displayed.

```
>> format long
>> %拟合模型y=Ax1^2+Bx2^2+Cx1+Dx2+Ex1*x2+F
%导入数据
y=[7613.51 7850.91 8381.86 9142.81 10813.6 8631.43 8124.94 9429.79 10230.81
10163.61 9737.56 8561.06 7781.82 7110.97]';
x1=[7666 7704 8148 8571 8679 7704 6471 5870 5289 3815 3335 2927 2758 2591]';
x2=[16.22 16.85 17.93 17.28 17.23 17 19 18.22 16.3 13.37 11.62 10.36 9.83 9.25]';
X=[ones(size(y)) x1.^2 x2.^2 x1 x2 x1.*x2];
%开始分析
[b,bint,r,rint,stats]=regress(y,X);
>> b
b =
1.0e+004 *
-1.353935450267798
0.000000089381408
-0.005811190715468
-0.000605427789545
0.479983626458532
-0.000037869040292
```

$$\begin{aligned} \text{拟合函数 } y = & 0.00089381408x_1^2 - 58.11190715468x_2^2 \\ & - 6.05427789545x_1 + 4799.83626458532x_2 \\ & - 0.37869040292x_1x_2 + 13539.35450267798 \end{aligned}$$

§ 5 基本应用

5.5 插值

插值问题要求对于给定的 n 个数据点 (x_i, y_i) ，找到一个函数 $y=f(x)$ ，使 $f(x_i)=y_i$ ($i=0, 1, \dots, n$)，其中 (x_i, y_i) 为插值节点， $f(x)$ 为插值函数， $f(x_i)=y_i$ 为插值条件。根据插值函数的类型，插值问题可分为代数插值和样条函数插值两种。代数插值的插值函数为多项式，亦称为多项式插值；样条函数插值的插值函数为样条函数，其中样条的概念来源于船舶、飞机等设计中描绘光滑曲线用的绘图工具，而数学上的样条函数则是若干段多项式函数光滑连结而成的分段函数。

§ 5 基本应用

`yi=interp1(x,y,xi,'method')` 一维插值，其中 (x,y) 为插值节点(x 不必为单调数列)， xi 为被插值点，`method`为插值方式，包括`nearest`(最邻近点插值)、`linear`(线性插值，缺省)、`spline`(三次样条插值)、`cubic`(三次多项式插值)。

`yi=spline(x,y,xi)` 三次样条插值，同`interp1(x,y,xi,'spline')`。

`zi=interp2(x,y,z,xi,yi,'method')` 二维插值，其中 (x,y,z) 为插值节点(x,y 为二维网格坐标，且为单调数列，可用`meshgrid`命令生成)， (xi,yi) 为被插值点(亦为二维网格坐标)，`method`为插值方式，包括`nearest`、`linear`、`cubic`。

`zi=griddata(x,y,z,xi,yi, 'method')` 散乱点插值函数

§ 5 基本应用

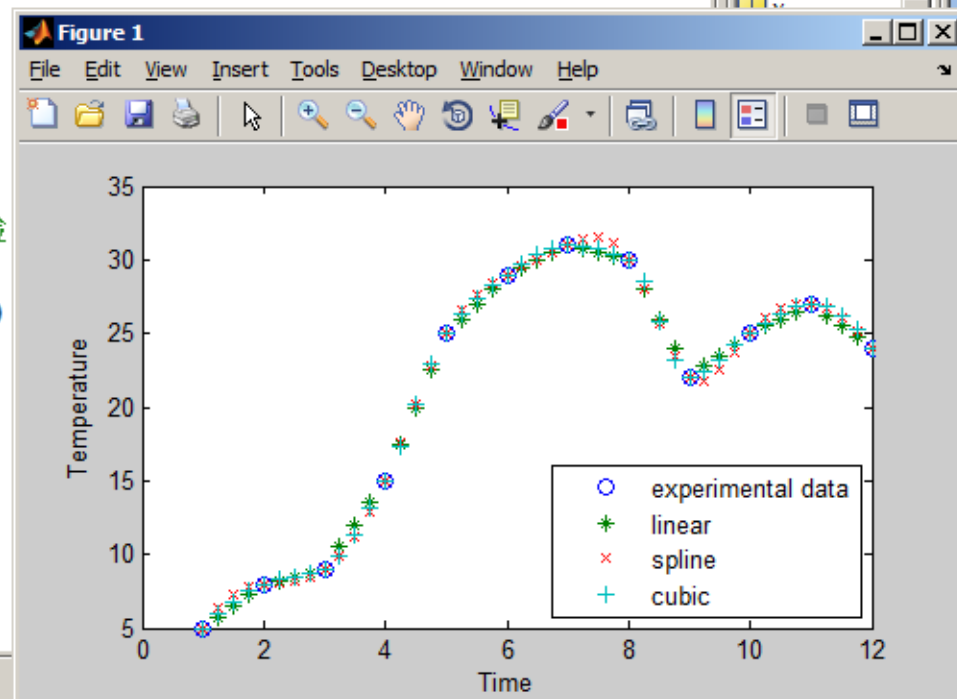
一维插值

```
MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: D:\program

Command Window

>> %插值节点
h=1:12;
t=[5 8 9 15 25 29 31 30 22 25 27 24];
hi=1:.25:12;%被插值节点
ti1=interp1(h,t,hi);%线性插值
ti2=interp1(h,t,hi,'spline');%三次样条插值
ti3=interp1(h,t,hi,'cubic');%三次多项式插值
plot(h,t,'o',hi,ti1,'*',hi,ti2,'x',hi,ti3,'+') %效果检验
%添加图例
legend('experimental data','linear','spline','cubic',4)
xlabel('Time')%x轴标注
ylabel('Temperature')%y轴标注
fx >>
```

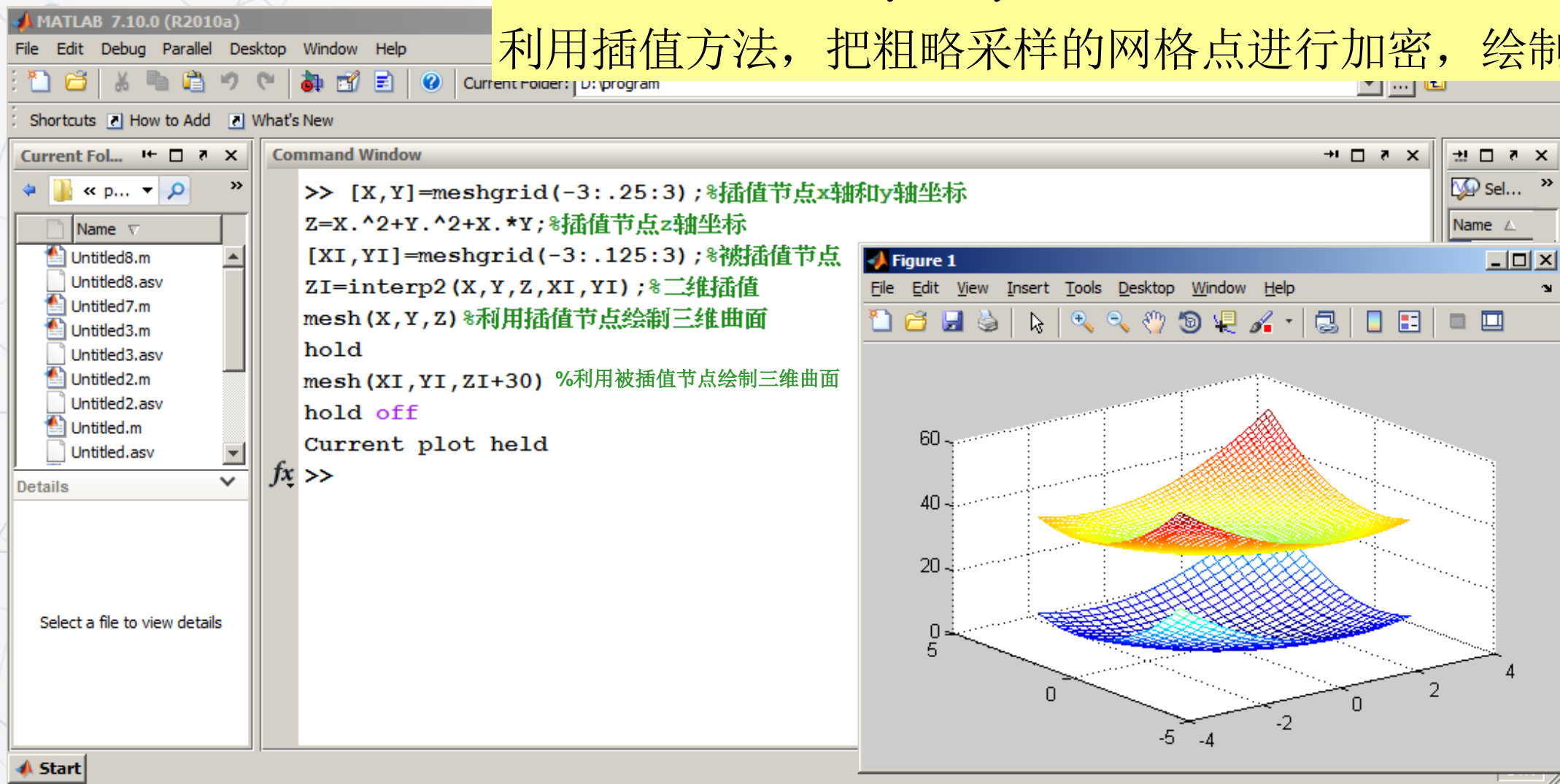
等价于 $ti2=spline(h,t,hi);$



§ 5 基本应用

在三维曲面 $z = x^2 + y^2 + xy$ 上粗略采样，绘制三维曲面
利用插值方法，把粗略采样的网格点进行加密，绘制曲面

二
维
插
值



§ 5 基本应用

5.6 其它

(1) 图与网络优化问题

graphminspantree 求无向图的最小树

graphshortestpath 求有向图中的最短路

graphmaxflow 求有向图中的最大流

§ 5 基本应用

5.6 其它

(2) 统计分析

max(x)最大值

min(x)最小值

mean(x)平均值

median(x)中间值

std(x)标准差

var(x)方差

cov(x,y)协方差

corrcoef(x,y)相关系数

normplot(x)正态性检验

[h,p,ci]=ttest(x,m)假设检验

[muhat,sigmahat,muci,sigmaci]=normfit(x)参数估计

第3章 Matlab软件

§1 Matlab开发环境

§2 Matlab中的变量、操作符、函数

§3 基本操作

§4 绘图功能

§5 基本应用

§6 M文件编程

§ 6 M文件编程

MATLAB 的M 文件有两类：命令文件和函数文件

6.1 命令文件

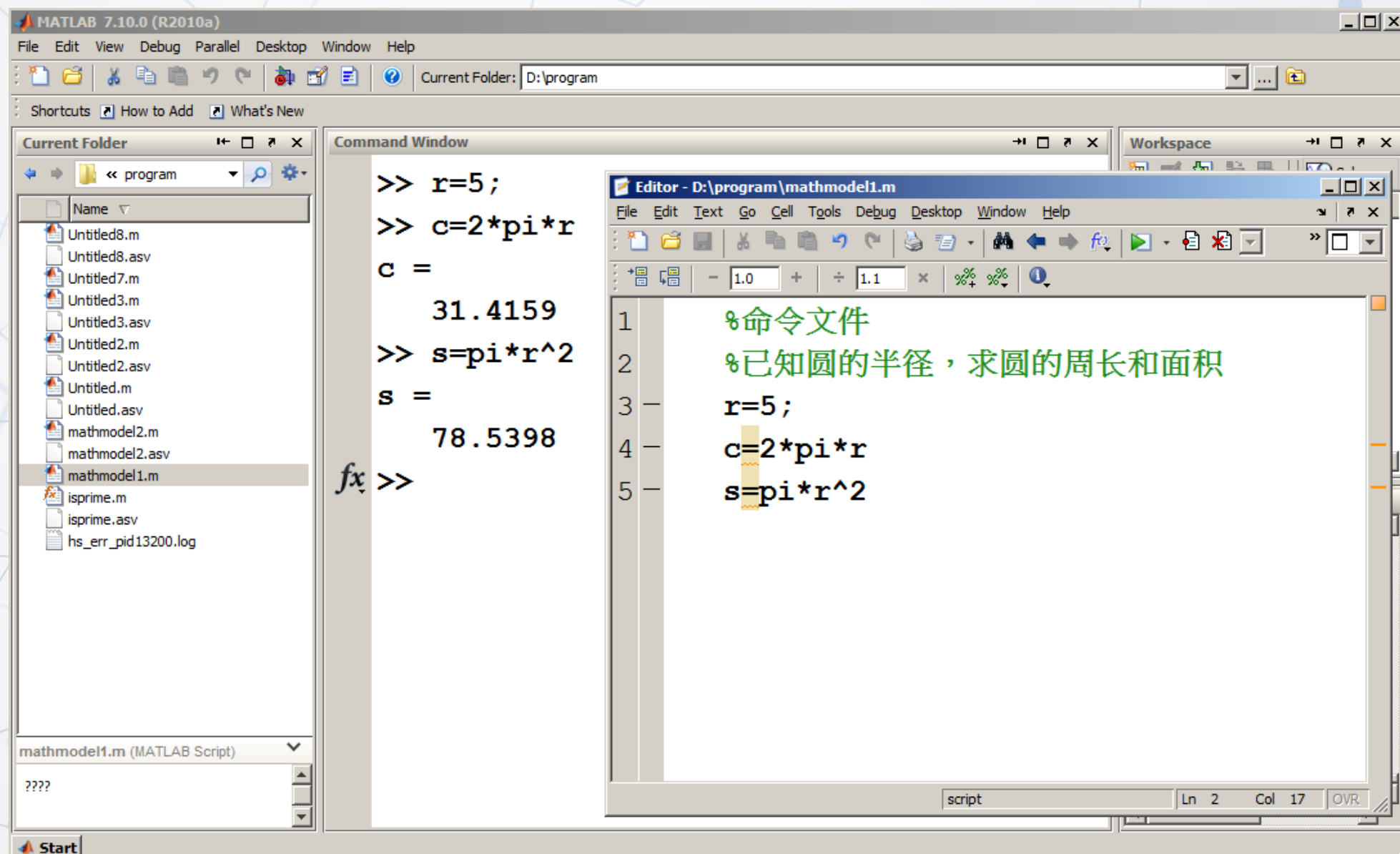
6.2 函数文件

§ 6 M文件编程

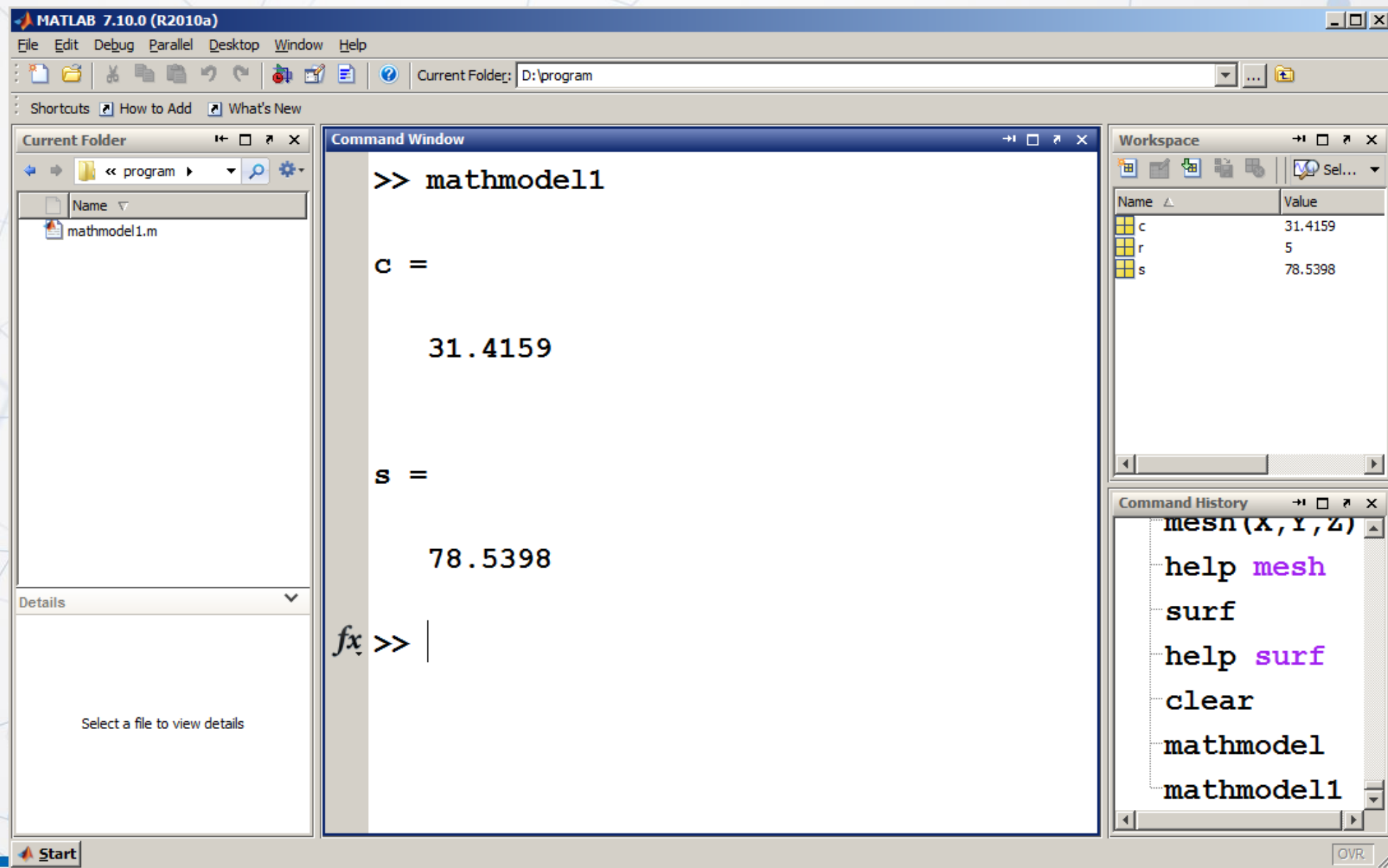
6.1 命令文件

打开M文件编辑器，将原本要在MATLAB环境下直接输入的语句，放在一个以 .m 为后缀的文件中，这一文件就称为命令文件。在命令窗口中的“>>”提示符下键入命令文件的文件名，可运行命令文件；在M文件编辑窗口点击菜单或工具栏的运行按钮也可运行。

§ 6 M文件编程



§ 6 M文件编程

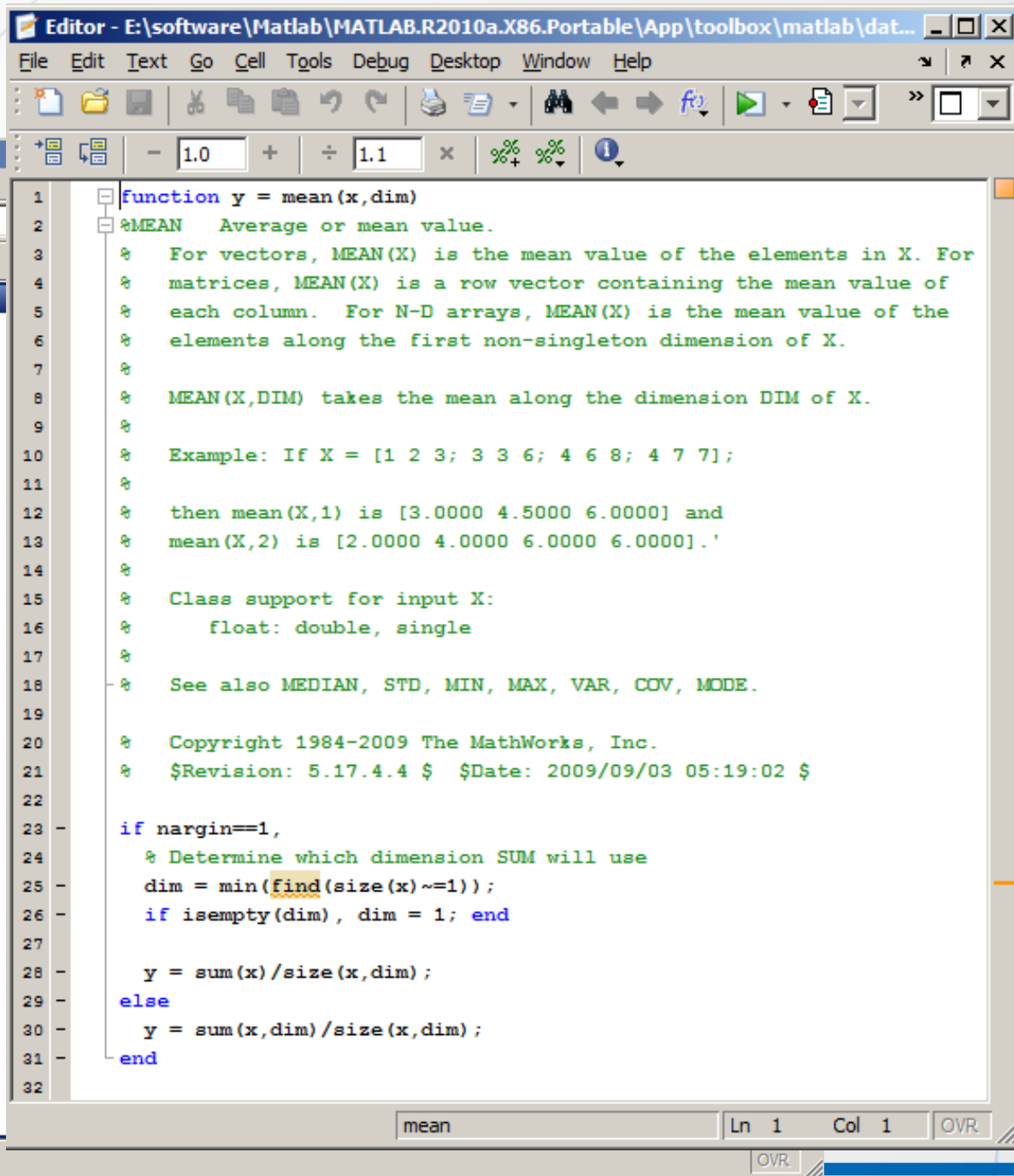
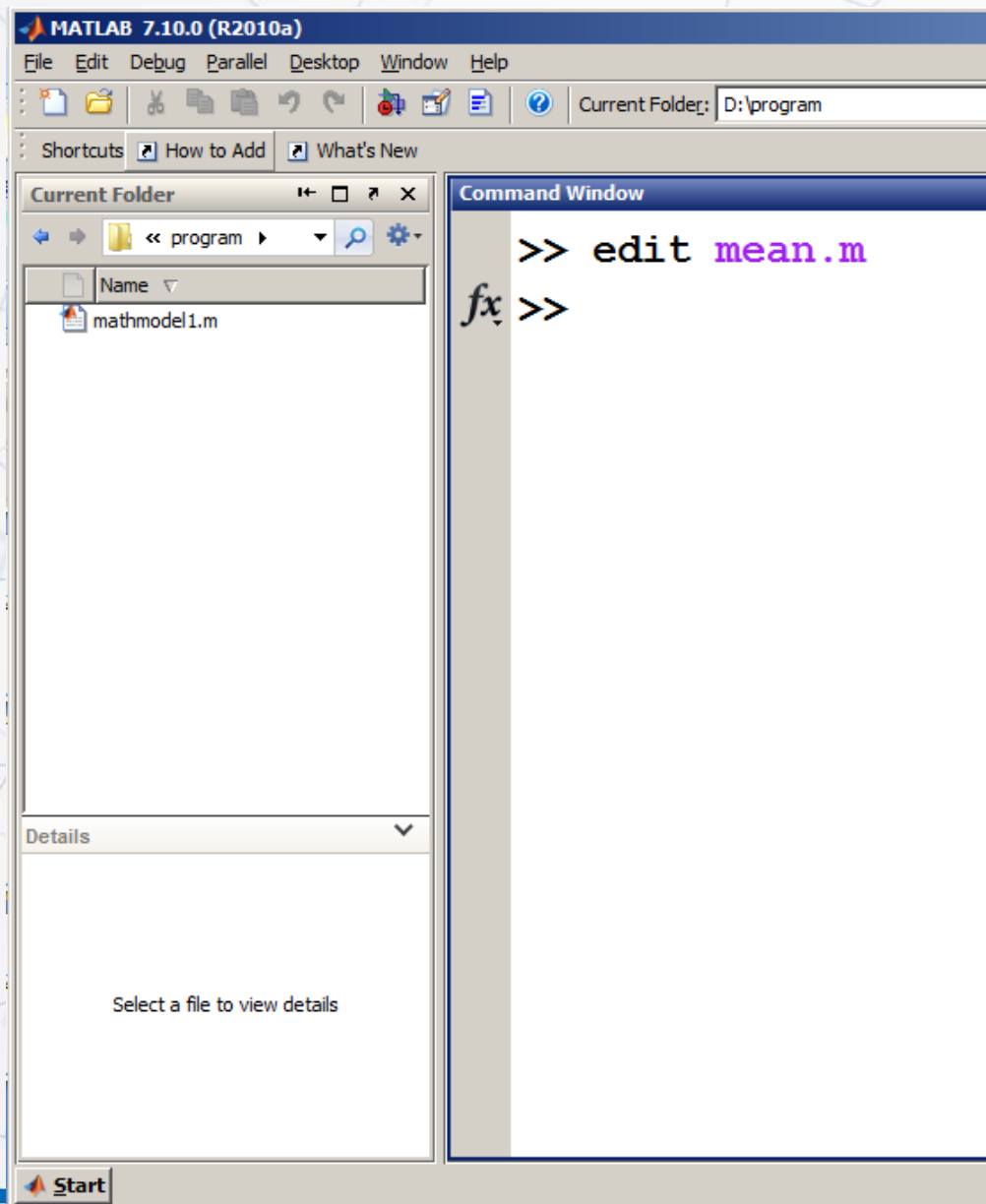


§ 6 M文件编程

6.2 函数文件

函数文件由五部分组成：函数定义行、H1 行、函数帮助文本、函数体、注释。MATLAB的内部函数都是由函数文件定义的，例如mean函数由函数文件mean.m 定义：

§ 6 M文件编程



§ 6 M文件编程

6.2 函数文件

流程控制语句可改变程序执行的流程，MATLAB的流程控制语句有以下四类：

- if , else , elseif , end构成的条件语句。
- switch , case , otherwise , end构成的情况切换语句。
- while , end构成不定次重复的循环语句。
- for , end构成指定次重复的循环语句。

§ 6 M文件编程

6.2 函数文件

- if语句

```
if expression  
    statements  
end
```

```
if expression1  
    statements1  
else  
    statements2  
end
```

```
if expression1  
    statements1  
elseif expression2  
    statements2  
else  
    statements3  
end
```

§ 6 M文件编程

6.2 函数文件

- switch语句

```
switch switch_expr
case case_expr,
    statements1
case {case_expr1,...}
    statements2
...
otherwise,
    statements3
end
```

- for语句

```
for variable=expr
    statements
end
```

- while语句

```
while expression
    statements
end
```

§ 6 M文件编程

例 编写M函数

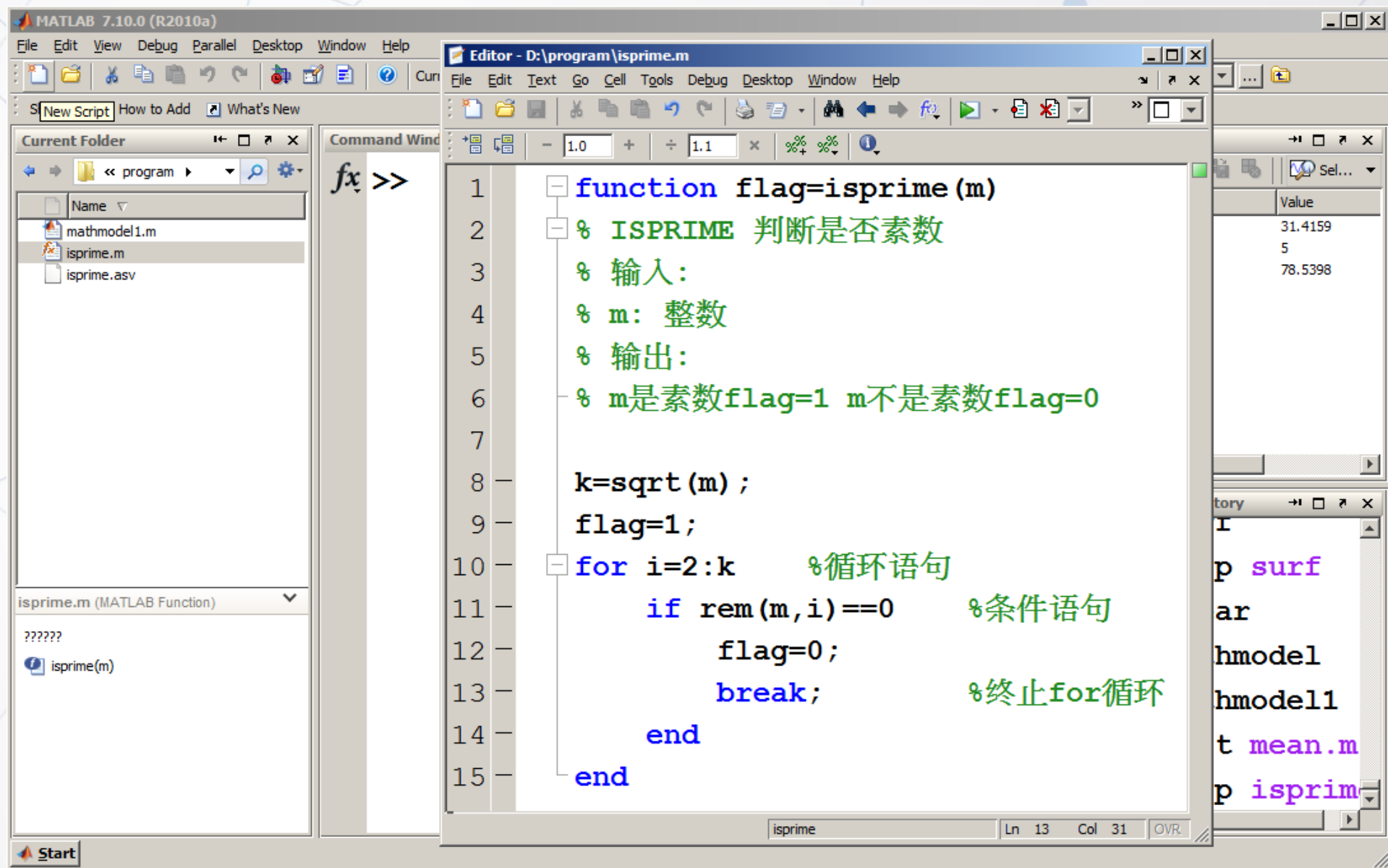
实现：

判断一个数 m

是否为素数。

编写判断素数

的函数文件



§ 6 M文件编程

例 编写M函数

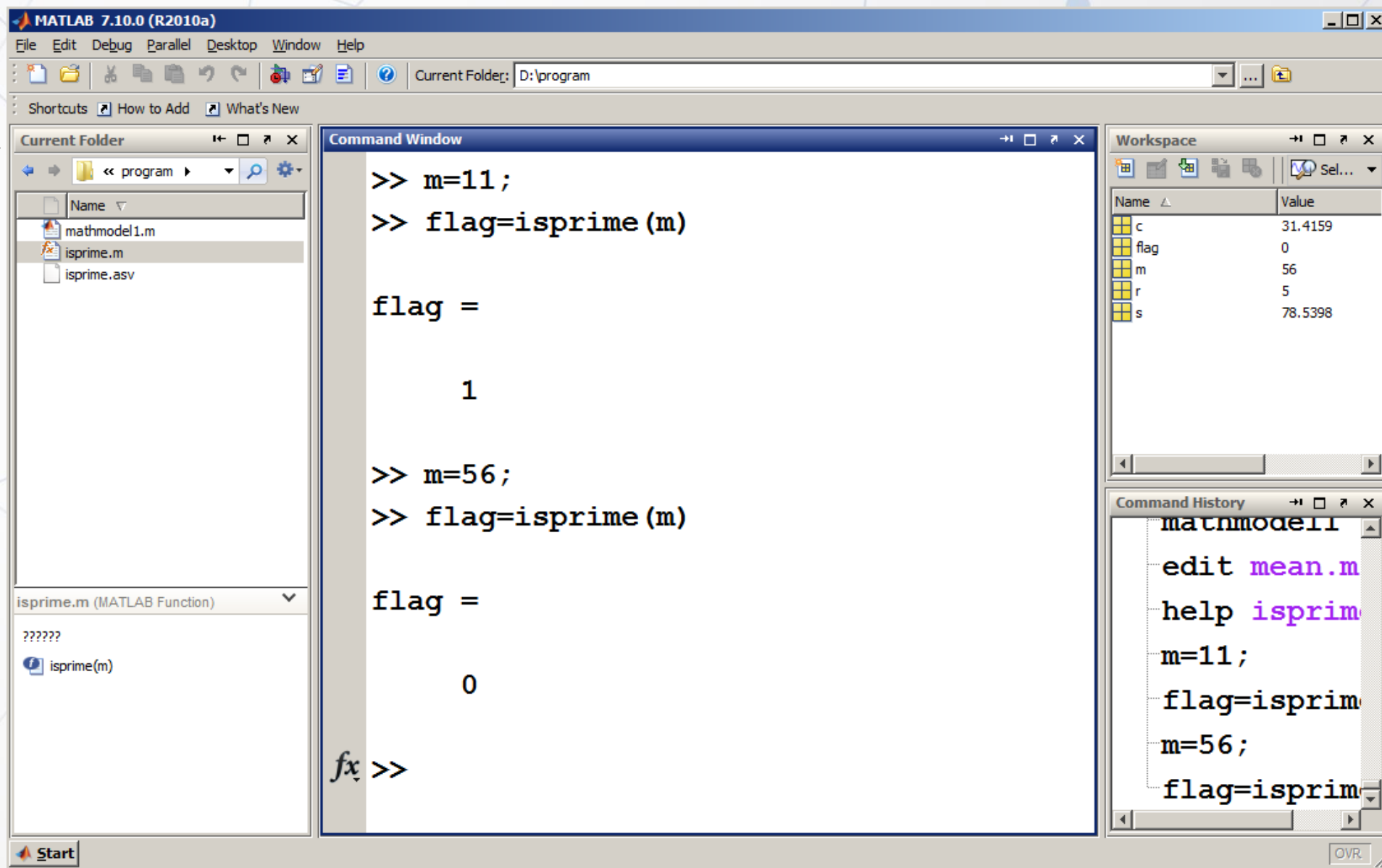
实现：

判断一个数m

是否为素数。

在命令窗口调

用函数文件



§ 6 M文件编程

例 编写M函数

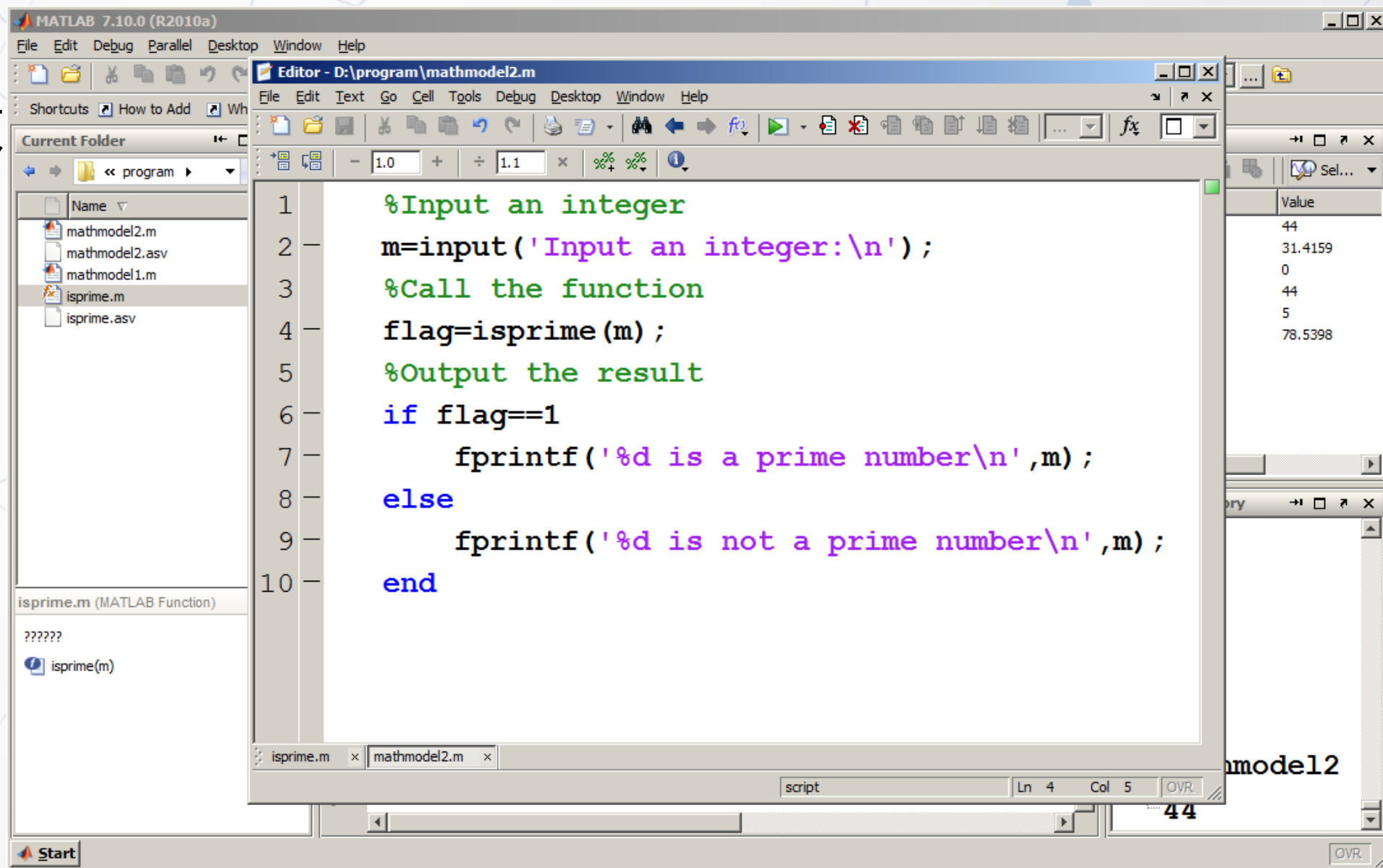
实现：

判断一个数m

是否为素数。

编写命令文件

调用函数文件



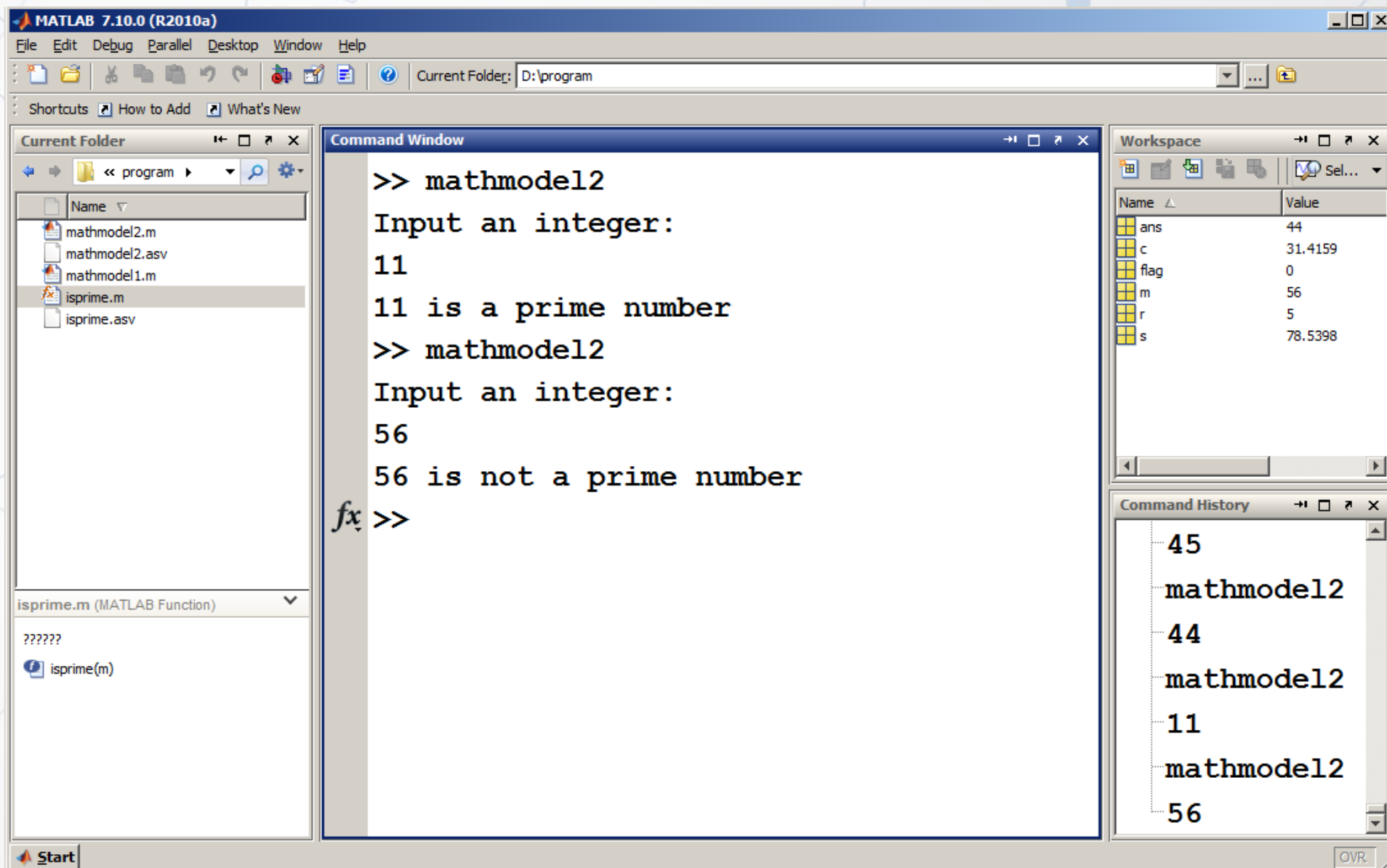
§ 6 M文件编程

例 编写M函数

实现：

判断一个数 m

是否为素数。



The background features a light blue and white abstract pattern. It consists of thin, irregular blue lines that form a network of polygons and paths. Scattered throughout this network are several small, solid blue circular dots. On the left side, a thick blue vertical arrow points upwards. At the bottom, a thick blue horizontal arrow points to the right. The overall aesthetic is clean and modern, typical of a technical or academic presentation.

第3章 Matlab软件