

内 容 简 介

本书介绍了 MATLAB7.0 版本的主要功能和将其用于工程计算上的使用方法,包括 MATLAB7.0 概论、数组运算和矩阵运算、基本图形函数、数值运算、高级图形处理、图形用户界面 GUI 设计和 MATLAB 程序设计基础等内容。

本书使用较大篇幅介绍了 MATLAB7.0 版开发环境的操作方法,包括 M 文件编辑器、数组编辑器、GUIDE 等。重点阐述了作为 MATLAB 核心内容的矩阵运算,并针对工程计算上常遇到的问题着重介绍了最小二乘法、多项式拟合、插值、傅立叶分析等几种数值运算方法。

MATLAB 语言以其图形功能而著称,本书在介绍了常用基本图形函数的用法后给出了图形对象、句柄图形系统的概念,从底层深入了 MATLAB 图形化的内核,由浅入深地阐述了用 MATLAB 底层指令实现高级图形处理功能的方法。

本书提供了大量例题,涵盖了正文中涉及的所有内容,与之配套设计的每章习题更能突出本章重点内容。本书可作为高等学校理工科本科学生学习 MATLAB 高级语言的教材和参考书,也可作为广大科技工作者、高校教师及理工科研究生使用 MATLAB 语言的参考书籍。

前 言

MATLAB 一词是“Matrix Laboratory”的缩写,意为“矩阵实验室”,MATLAB 是功能全面的用于科学及工程计算的计算机高级语言。

MATLAB 具有以矩阵运算为基础的数学计算和分析功能。与其它计算机高级语言相比,其强大的数据可视化表现能力即图形功能更具有独特的优势,使 MATLAB 语言广泛地应用于自动控制、数字信号处理、人工智能、通信工程、动态系统仿真等领域。本书由浅入深地讲解了 MATLAB7.0 的开发环境和通用功能,使读者能够迅速入门,以便今后结合各自的专业知识,将这一工程计算语言应用于专业领域,拥有一个从事专业研究的有力工具。

本书共分为 7 章,全面介绍了 MATLAB7.0 的开发环境的使用方法、基本运算方法及多种图形功能。第 1 章介绍了 MATLAB7.0 概论,包括 MATLAB 的产生及发展、MATLAB7.0 的开发环境,重点介绍了 M 文件编辑器的使用方法;第 2 章介绍了数组运算和矩阵运算及一些常用的数学函数;第 3 章介绍了基本图形函数,包括二维绘图函数、三维绘图函数及图形的颜色控制方法;第 4 章介绍了数值运算,包括数据分析、最小二乘拟合、多项式拟合、插值和傅立叶分析等;第 5 章介绍了高级图形处理功能,包括图形对象和句柄、对象的属性和高级图形处理应用实例;第 6 章介绍了图形用户界面 GUI 设计;第 7 章介绍了 MATLAB 程序设计基础,包括数据类型、

程序流程控制和函数等。

除基本例题外,为配合教学的需要,每章都给出了针对性较强的习题。

本书的出版得到了黑龙江教育出版社的大力支持。

本书的例题和习题均在 MATLAB7.0 环境下调试并运行通过。虽然作者对书中的内容反复推敲、修改订正,但由于作者水平的限制,书中难免还存在某些缺点或错误。作者恳切期望得到各方面专家同行和广大读者的指正,E-mail: djw@hrbust.edu.cn。

作 者

2007 年 3 月于哈尔滨

目 录

第 1 章 MATLAB7.0 概论	(1)
1.1 MATLAB 的产生及发展	(1)
1.2 MATLAB7.0 开发环境	(1)
1.2.1 Command Window 使用入门	(1)
1.2.2 Command Window 的编辑与控制	(8)
1.2.3 Command Window 的参数设置	(10)
1.2.4 Current Directory	(12)
1.2.5 Workspace	(14)
1.3 M 文件编辑调试器	(17)
1.3.1 Editor 使用入门	(17)
1.3.2 M 文件的调试	(19)
习题一	(20)
第 2 章 数组运算和矩阵运算	(22)
2.1 数组和矩阵的创建	(22)
2.1.1 MATLAB 中数组和矩阵的概念	(22)
2.1.2 数组和矩阵的创建和访问	(23)
2.1.3 特殊矩阵的创建	(24)
2.2 数组和矩阵的算数运算	(27)
2.2.1 加减运算	(27)
2.2.2 乘除运算	(28)
2.2.3 幂运算	(31)
2.2.4 矩阵的转置	(32)
2.3 关系运算	(32)
2.4 逻辑运算	(33)
2.5 MATLAB 矩阵函数	(34)
2.5.1 特征值分解	(34)
2.5.2 矩阵的逆	(36)
2.5.3 矩阵的翻转和旋转	(37)

2.5.4 矩阵元素的重组	(38)
2.6 MATLAB 数学函数	(39)
2.6.1 三角函数	(39)
2.6.2 指数函数	(40)
2.6.3 复数函数	(41)
2.6.4 取整和求余函数	(42)
习题二	(44)
第3章 基本图形函数	(46)
3.1 二维绘图	(46)
3.1.1 直角坐标绘图	(46)
3.1.2 对数坐标绘图	(51)
3.1.3 极坐标绘图	(52)
3.1.4 多重曲线绘图	(54)
3.1.5 特殊二维图形	(58)
3.2 图形窗口	(62)
3.2.1 图形窗口的菜单和工具条	(62)
3.2.2 图形的控制命令	(64)
3.2.3 图形的标注	(67)
3.3 三维绘图	(69)
3.3.1 三维曲线图	(69)
3.3.2 三维曲面图	(73)
3.4 图形的颜色控制	(79)
3.4.1 RGB 三元组	(79)
3.4.2 Colormap 色图	(79)
3.4.3 图像文件	(81)
习题三	(84)
第4章 数值运算	(86)
4.1 数据分析	(86)
4.1.1 数据统计函数	(86)
4.1.2 差分	(89)
4.2 最小二乘拟合	(90)
4.2.1 解线性方程组	(90)
4.2.2 方程组的最小二乘解	(91)
4.3 多项式拟合	(94)
4.3.1 多项式的创建	(95)
4.3.2 多项式常用函数	(95)
4.3.3 多项式拟合	(98)

4.4 插值.....	(101)
4.4.1 一维插值.....	(102)
4.4.2 二维插值.....	(104)
4.5 傅立叶分析.....	(106)
4.5.1 FFT 函数.....	(106)
4.5.2 数字滤波.....	(107)
习题四	(109)
第5章 高级图形处理	(111)
5.1 图形对象和句柄.....	(111)
5.1.1 图形对象和句柄的概念.....	(111)
5.1.2 图形对象和句柄的创建.....	(112)
5.2 对象的属性.....	(117)
5.3 高级图形处理应用实例.....	(120)
5.3.1 极坐标绘图的细致刻画.....	(120)
5.3.2 矩阵多重曲线绘图的分别修饰.....	(121)
5.3.3 直方图的颜色修改.....	(122)
5.3.4 表面图的细节效果.....	(123)
5.3.5 动画曲线.....	(124)
5.3.6 曲面彩绘.....	(125)
习题五	(126)
第6章 图形用户界面 GUI 设计	(127)
6.1 图形用户界面开发环境 GUIDE	(127)
6.1.1 GUIDE 的启动	(127)
6.1.2 布局区.....	(129)
6.1.3 控件栏.....	(129)
6.1.4 菜单和工具栏.....	(130)
6.2 常用的 uicontrol	(132)
6.2.1 Push Button 按钮	(133)
6.2.2 Static Text 静态文本	(136)
6.2.3 Edit Text 可编辑文本.....	(137)
6.2.4 Slider 滑动条	(137)
6.2.5 Listbox 列表框	(138)
6.2.6 Pop-up Menu 弹出框	(138)
6.3 GUI 设计实例.....	(138)
6.3.1 自制的调色板.....	(138)
6.3.2 音乐播放器.....	(140)
6.3.3 学生成绩查询系统.....	(141)

6.3.4 正弦波合成器	(143)
习题六	(144)
第7章 MATLAB 程序设计基础	(146)
7.1 MATLAB 的数据类型	(146)
7.1.1 数值数组	(146)
7.1.2 字符数组	(148)
7.1.3 元胞数组	(154)
7.1.4 结构数组	(156)
7.2 程序流程控制	(159)
7.2.1 条件语句	(159)
7.2.2 分支语句	(161)
7.2.3 循环语句	(161)
7.2.4 其它流程控制语句	(163)
7.3 函数	(165)
7.3.1 脚本文件和函数文件	(165)
7.3.2 局部变量和全局变量	(169)
习题七	(170)
习题答案	(172)
习题一答案	(172)
习题二答案	(173)
习题三答案	(178)
习题四答案	(183)
习题五答案	(189)
习题六答案	(191)
习题七答案	(194)

第1章 MATLAB7.0 概论

1.1 MATLAB 的产生及发展

MATLAB 一词是“Matrix Laboratory”的缩写,意为“矩阵实验室”,MATLAB 是功能全面的用于科学及工程计算的计算机高级语言。

二十世纪七十年代末,美国新墨西哥大学计算机科学系主任 Cleve Moler 教授用 FORTRAN 语言编写了一组调用 LINPACK 和 EISPACK 库程序的简单易用的接口程序,以减轻学生的编程负担,这就是萌芽状态的 MATLAB。1984 年成立的 MathWorks 公司正式将 MATLAB 推向市场,此时 MATLAB 的内核已改为由 C 语言编写,并在原来数值运算的基础上增加了数据可视化功能,使用更加方便。

MATLAB 具有以矩阵运算为基础的数学计算和分析功能。与其它计算机高级语言相比,其强大的数据可视化表现能力即图形功能更具有独特的优势,使 MATLAB 语言广泛地应用于自动控制、数字信号处理、人工智能、通信工程、动态系统仿真等领域。二十世纪九十年代, MATLAB 就已经成为国际控制界公认的标准计算软件。

MATLAB 在推出以来功能不断提高完善,经历了多个版本阶段,1993 年的 4.0 版、1997 年的 5.0 版、2001 年的 6.1 版和 2002 年的 6.5 版是功能改进较大的版本。2004 年 5 月 Mathworks 公司又推出了 MATLAB 的最新版本——MATLAB7.0。本书由浅入深地讲解了 MATLAB7.0 的开发环境和通用功能,使读者能够迅速入门,以便今后结合各自的专业知识,将这一工程计算语言应用于专业领域,拥有一个从事专业研究的有力工具。

1.2 MATLAB7.0 开发环境

MATLAB7.0 安装后在桌面上显示快捷方式图标。双击图标即可打开如图 1-1 所示的操作桌面(Desktop)。该操作桌面主要包括:指令窗(Command Window)、历史指令窗(Command History)、工作空间 Workspace)、当前目录(Current Directory)、开始按钮(Start)以及主菜单和工具栏等几部分。

1.2.1 Command Window 使用入门

Command Window 的缺省位置在 Desktop 的右侧,该窗口用来输入指令并显示指令执行结果。MATLAB7.0 在就绪状态时 Command Window 显示指令行提示符“>>”,在其后的光标处即可输入指令。

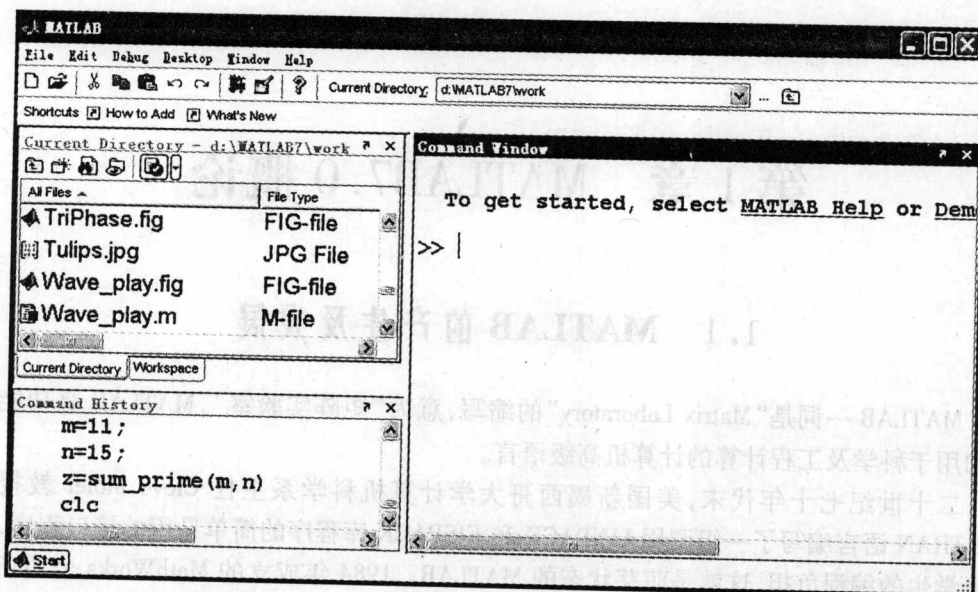


图 1-1 MATLAB7.0 Desktop

1.2.1.1 计算器使用方法

对于初学者来说, Command Window 可以像使用计算器一样简单地使用。

【例 1-2-1】计算 $5 + 6 \times 3.5 \div 8$ 的值。

```
>> 5 + 6 * 3.5 / 8
```

在提示符“>>”后输入以上指令, 再按【Enter】键, Command Window 显示出如图 1-2 所示的结果。

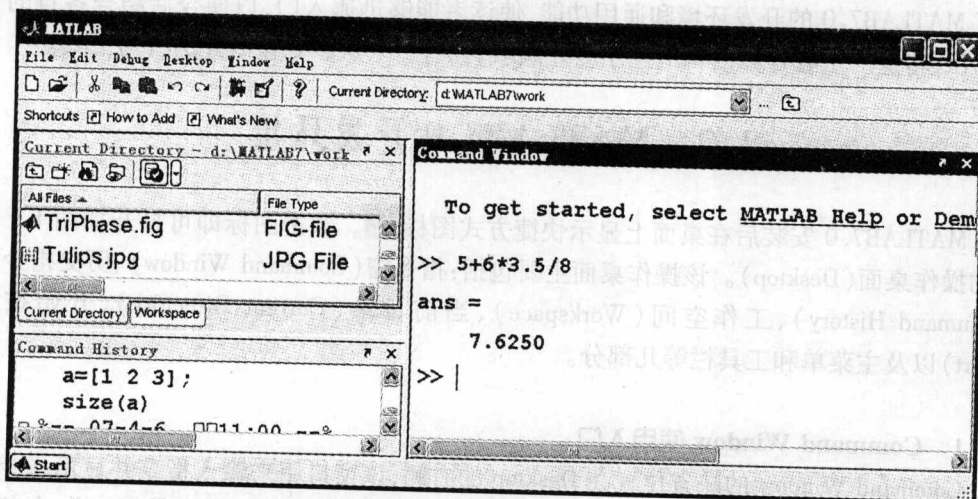


图 1-2 Command Window 的计算器用法

运算结果显示为:

```
ans =
    7.6250
```

其中“ans”为 MATLAB 的一个默认变量,是“answer”的缩写,意为“运算答案”。

【例 1-2-2】计算 $(3+2) \times 4 - 2^3$ 的值。

```
>> (3+2)*4-2^3
```

```
ans =
    12
```

1.2.1.2 变量和表达式

MATLAB 采用表达式语言,输入的指令由 MATLAB 解释运行。MATLAB 指令的形式有两种:

- ①表达式
- ②变量 = 表达式

在 1.2.1.1 节中采用的计算器使用方法即为第①种 MATLAB 指令形式。由于第①种 MATLAB 指令形式在计算出表达式的值后未指定赋值给哪一变量,因此 MATLAB 将运算结果赋值给默认变量“ans”。第②种 MATLAB 指令形式与第①种的区别在于,将运算结果赋值给指定的变量。

【例 1-2-3】计算 $(3+2) \times 4 - 2^3$ 的值。

```
>> x = (3+2)*4-2^3
```

```
x =
    12
```

本例题与【例 1-2-2】的要求相同,但由于采用了“变量 = 表达式”的指令形式, MATLAB 将运算结果赋值给变量“x”。当然,就题目的要求来说使用哪种形式均可。

由于 MATLAB 采用 C 为内核,因此 MATLAB 的变量命名规则与 C 语言类似:

- ①变量名以英文字母开头;
- ②变量名中可以包含英文字母、数字和下划线;
- ③变量名区分大、小写英文字母,即:“A”与“a”是不同的两个变量。

与 C 语言的变量命名规则不同的是:MATLAB 变量名不能以下划线开头,如:

```
>> _x=5
```

```
??? _x=5
```

```
|
```

Error: Missing variable or function.

在使用非法的变量名时, MATLAB 会给出上述的错误信息提示。Command Window 错误信息显示的缺省颜色为红色, 初学者最好不要改动此项颜色设置。

MATLAB 中有一些默认变量(预定义变量), 在启动 MATLAB 后默认变量自动产生, 用户在编程时最好不给默认变量赋新值, 以免引起混淆。MATLAB 常用的默认变量如表 1-1 所示。

表 1-1 MATLAB 常用默认变量

默认变量名	意义
ans	运算结果
pi	圆周率 π
Inf, inf	无穷大 ∞
NaN, nan	非数(Not a Number)
i, j	虚单元
realmax	最大正实数
realmin	最小正实数
eps	机器零阈值

【例 1-2-4】MATLAB 常用默认变量 pi、Inf、NaN 和 i 的用法举例。

```
>> a=pi/2
```

```
a =
```

```
1.5708
```

```
>> sin(a)
```

```
ans =
```

```
1
```

```
>> cos(pi/4)
```

```
ans =
```

```
0.7071
```

```
>> b=1
```

```
b =
```

```
1
```

```
>> c=b/0
```

Warning: Divide by zero.

c =

Inf

>> d = c/c

d =

NaN

>> e = 0/0

Warning: Divide by zero.

e =

NaN

>> z = 2 + 3 * i

z =

2.0000 + 3.0000i

>> abs(z)

ans =

3.6056

>> real(z)

ans =

2

>> imag(z)

ans =

3

其中 $\text{abs}(z)$ 、 $\text{real}(z)$ 和 $\text{imag}(z)$ 分别为求复数 z 的模、实部和虚部的函数。

【例 1-2-5】MATLAB 默认变量 eps 的用法举例。

>> e = 0 / 0

Warning: Divide by zero.

e =

NaN

>> e = 0/eps

e =

0

>> a = eps

a =

```

2.2204e-016
>> b = eps * 2
b =
4.4409e-016
>> c = 3 + eps
c =
3
>> d = eps * eps
d =
4.9304e-032

```

由以上运行结果可见,“eps”为计算机所能识别的最小的非零数,在采用 IEEE 浮点运算的计算机上,数值的相对精度就是 eps,不同计算机的 eps 的值会有所不同。

1.2.1.3 指令行中的标点符号

MATLAB 指令不能识别中文标点符号,因此标点符号要在西文状态下输入。指令行中常用的标点符号及其作用如表 1-2 所示。

表 1-2 指令行中常用标点符号

标点符号	作用
方括号 []	输入数组或作为函数的多个输出参数列表
空格	分隔数组中同一行的各元素
逗号 ,	分隔数组中同一行的各元素或用于指令末尾分隔下一条指令
分号 ;	分隔数组中不同的行或用于指令末尾表示此指令的运行结果不显示
冒号 :	产生等间隔的元素
百分号 %	其后的字符为非执行的注释文字
圆括号 ()	数组元素引用或作为函数的输入参数列表

【例 1-2-6】MATLAB 指令行中常用标点符号的用法举例。

① 方括号、逗号、空格、分号的用法

```

>> a = [1 2 3]
a =
1    2    3
>> b = [1,2,3]
b =
1    2    3
>> c = [1 2;3 4]

```

```

c =
    1     2
    3     4
>> d=[1,2;3,4]
d =
    1     2
    3     4

```

② 逗号、分号用在指令末尾

```

>> x=5,y=3*x
x =
    5
y =
   15
>> x=5;
>> y=3*x
y =
   15
>> x=5;y=3*x
y =
   15
>> x=5;y=3*x;
>>

```

③ 冒号的用法

```

>> k=1:2:9
k =
    1     3     5     7     9
>> f=1:2:10
f =
    1     3     5     7     9
>> g=3:1:6
g =
    3     4     5     6
>> g=3:6

```

```

g =
    3    4    5    6
>> k=1:0.5:3
k =
    1.0000    1.5000    2.0000    2.5000    3.0000
>> k=1:5:30
k =
    1     6    11    16    21    26

```

④ 百分号的用法

```

>> m=3 % m is number of students in this class
m =
    3

```

⑤ 圆括号的用法

```

>> a=[1 2 3 4;5 6 7 8]
a =
    1     2     3     4
    5     6     7     8
>> a(1,1)
ans =
    1
>> a(2,3)
ans =
    7
>> a(2,3)=100
a =
    1     2     3     4
    5     6    100    8

```

1.2.2 Command Window 的编辑与控制

1.2.2.1 指令行的编辑

Command Window 不是普通的文本编辑器,每次输入一行指令后,按【Enter】键该指令就立即执行,不能再修改。虽然可以用鼠标单击 Command Window 中已经输入过的指令

行,但光标位置却不能随之移动,光标始终位于最下行的提示符“>>”后,此处即为输入焦点,等待输入新的指令。由此可见,Command Window 非常类似于操作系统的“C:\命令提示符”使用方式,即 DOS 操作系统的输入方式。

若要重新执行输入过的指令或相似指令,可在提示符“>>”后按【上箭头】和【下箭头】键,将输入过的指令重新显示出来,再按【Enter】键执行该指令,或在指令重新显示出来后稍作修改,再按【Enter】键,用于相似指令的执行。

利用 MATLAB Desktop 中的 Command History 窗口也可以重新执行输入过的指令。如图 1-3 所示,在 Command History 窗口中双击某一条指令,该指令则重新在 Command Window 中执行一次。

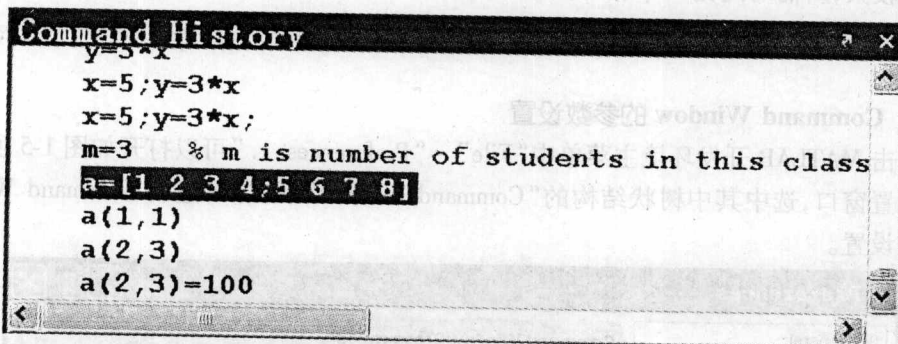


图 1-3 Command History 用法

1.2.2.2 Command Window 的控制

在 MATLAB 开发环境下 Command Window 的缺省状态是“Dock”状态,即 Command Window 镶嵌在 Desktop 中。有两种方法可以将 Command Window 独立出来:

- ①点击 MATLAB 开发环境主菜单中“Desktop”-“Undock Command Window”;
- ②点击 Command Window 右上角的  “Undock”图标。

独立出来的 Command Window 如图 1-4 所示。与图 1-1 中“Dock”状态的 Command Window 相比,独立出来的 Command Window 有了自己的操作菜单,可以进行文件操作、编辑和调试。当然也可以使用 Command Window 菜单中“Desktop”-“Dock Command Window”将独立出来的 Command Window 镶嵌回 MATLAB 开发环境的 Desktop 中。

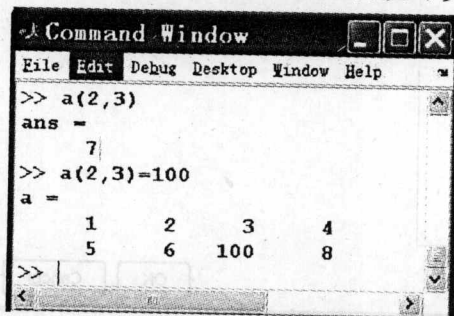


图 1-4 独立的 Command Window

1.2.2.3 Command Window 常用指令

`clc`——清除 Command Window 中显示的内容；

`help functionname`——在 Command Window 中显示关于函数 *functionname* 的帮助信息；

`lookfor keyword`——在 Command Window 中显示与关键字 *keyword* 有关的函数名称和命令。

使用 `help` 和 `lookfor` 指令时,根据已安装的 MATLAB 组件的多少,指令所需的时间不同,安装 MATLAB 组件越多,所需时间越长。由于未指出具体的函数名,lookfor 指令查找的范围较大,所需时间比 `help` 指令要长一些。

`format formatname`——将 Command Window 中显示数值的格式设为 *formatname*。

1.2.3 Command Window 的参数设置

点击 MATLAB 开发环境主菜单中“File”-“Preferences...”可以打开如图 1-5 所示的参数设置窗口,选中其中树状结构的“Command Window”项,即可进行 Command Window 的参数设置。

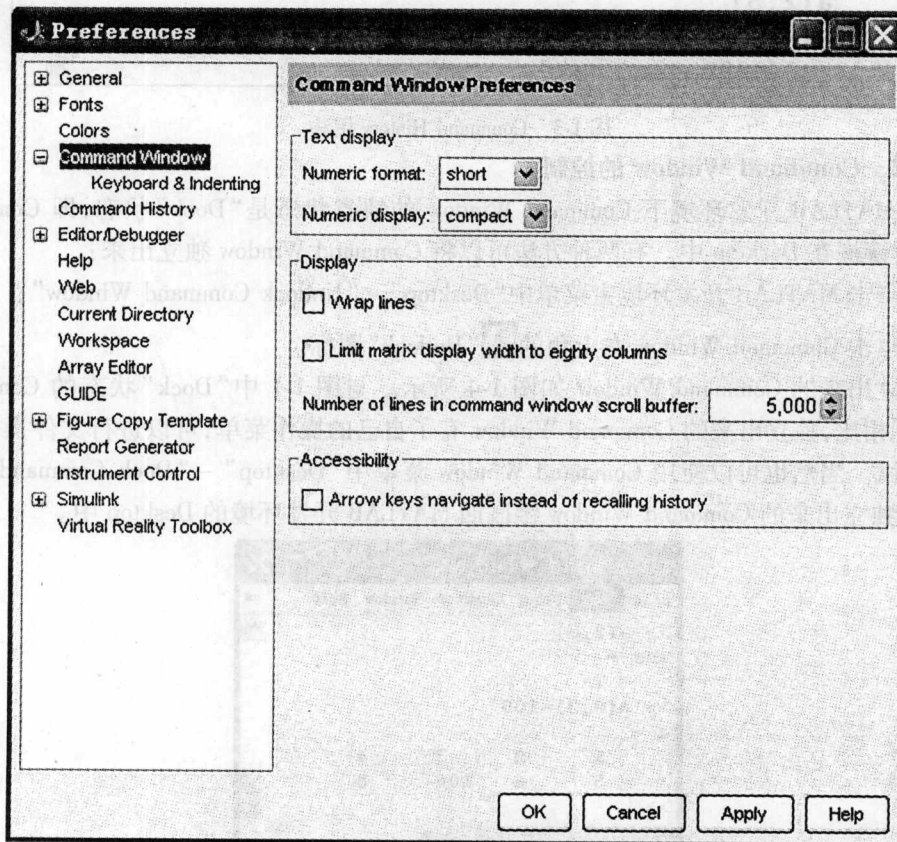


图 1-5 Command Window 的参数设置