

MATLAB 5.1

全攻略宝典

万水
电脑软件全攻略宝典系列

潘晓辉 陈 强 编著
康创团体 策划



中国水利水电出版社
www.waterpub.com.cn

426

TP312.11A
P18

电脑软件全攻略宝典系列

MATLAB 5.1 全攻略宝典

潘晓辉 陈强 编著

康创团体 策划



A0933093



中国水利水电出版社

www.waterpub.com.cn

内 容 提 要

本书较为系统地介绍了 MATLAB 软件的基本知识、操作技巧、以及在数值分析、信号处理和系统控制等领域的应用。全书共包含六章, 内容包括 MATLAB 的基本数据类型、常用函数、控制语句、编程方法、各种二维和三维图形显示和输出方法、图形对象体系、用户界面程序设计、简单的图像处理、各种数值分析算法的 MATLAB 实现、产生特殊数字信号序列、一般数字信号时域和频域分析、使用 MATLAB 对模拟和数字滤波器模型表示、分析设计模拟和数字滤波器、倒谱分析、使用 MATLAB 进行通讯信号处理、MATLAB 在控制系统设计和分析中的应用等等。

图书在版编目(CIP)数据

MATLAB 5.1 全攻略宝典 / 潘晓辉, 陈强编著. -北京: 中国水利水电出版社, 2000.1

(电脑软件全攻略宝典系列)

ISBN 7-5084-0261-8

I . M … II . ①潘 … ②陈 … III . 计算机辅助计算 - 软件包, MATLAB 5.1 IV . TP391.75

中国版本图书馆 CIP 数据核字 (2000) 第 13398 号

书 名	MATLAB 5.1 全攻略宝典
作 者	潘晓辉 陈强 编著
策 划	康创团体
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www. waterpub. com. cn E-mail: sale@waterpub. com. cn 电话: (010) 63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂印刷
规 格	787×1092 毫米 16 开本 24.75 印张 560 千字
版 次	2000 年 1 月北京第一版 2000 年 1 月北京第一次印刷
印 数	0001—6000 册
定 价	35.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社发行部负责调换

版权所有·侵权必究

前 言

本书的写作宗旨是：“面向广大工程技术人员和大专院校在校学生，全面而深入浅出地介绍 MATLAB 软件的原理和使用，通过详实而且尽量简单但能说明问题的例子对 MATLAB 软件进行剖析，促进软件应用和推广。”

如果你是一名刚刚踏入高等院校的学生，正在为老师布置的一大堆数学作业头疼不已，请读一读这本书，她会通过丰富的例子告诉你如何使用一个优秀的软件——MATLAB 多快好省地帮你取得高分。

或者你是一名高年级学生乃至研究生，整天在塞满硬盘的实验数据面前愁眉苦脸，那么也请花点时间翻一翻这本书吧，相信她独特的基础+实战的结构会为你排解不少麻烦。

也许你已经毕业多年，从事了一定时间的实际工作，但是不管你对 MATLAB 知道多少，只要你对于计算机有些认识，这本书一定会在某些方面为你提供方便，和 MATLAB 软件一样成为你的得力助手。

本书的三大特点是：语言平实、说理透彻、示例充分。本书尽量使用接近于工程实际中应用的语言进行表述，对某些领域的名词则小心查证，力求作到准确无误；本书在介绍完成任务使用的各种攻略时，力争多方比较，取得最佳效果，并不时借鉴其它软件或开发工具的类似概念来说明问题；本书中的例子是在作者长期使用 MATLAB 软件的过程中积累的，全部在计算机上调试通过，并且多数具有很强的实用性和启发性。

编者

1999 年 8 月

第一章 新手上路

如果你是第一次使用 MATLAB，本章将教会你如何使用 MATLAB 软件以及一些基础知识，使你能够完成一些简单的任务，并为后续章节的学习做准备。

MATLAB 是 Matrix Laboratory（矩阵实验室）的缩写，其基本的元素是可以任意改变维数的矩阵，MATLAB 具有强大的矩阵处理功能，可以直接对矩阵进行各种复杂的运算，非常适合需要进行大量矩阵处理的地方，因此 MATLAB 被广泛地应用到数学计算、控制系统、信号处理等地方。MATLAB 所有的变量不需要定义而直接使用，这对于科学和工程计算非常的方便。MATLAB 中矩阵的维数可以任意地改变，MATLAB 自动地完成不同维数矩阵的转化。这是 MATLAB 的独特的地方，对内存调度有所了解的人都知道不同维数矩阵的转化涉及到大量的内存操作，MATLAB 已经自动地完成这些复杂的操作，大大方便了程序的设计。MATLAB 经过多年的发展，已经相当的完善，成为线性代数的标准工具。MATLAB 中，向量可以看作 $n \times 1$ 维的矩阵，数组与矩阵之间可以相互转化。一维数组对应向量， n 维数组对应 n 维矩阵。对于矩阵的各种处理方法也同样适合向量和数组。多项式的系数可以用向量来表示，通过对向量的运算，我们可以直接地计算多项式的加减乘除法。因此，在 MATLAB 中，向量、数组、多项式都是矩阵的范畴。

学好灵活地处理各种矩阵是掌握 MATLAB 的基础，本章从最简单的矩阵创建着手，一步一步地向读者介绍处理矩阵的各种方法，由浅入深，覆盖面广，使读者在最短的时间内掌握 MATLAB 的基本使用方法。

MATLAB 有两种命令执行的方式，既可以象计算器一样直接执行输入的命令，又可以象 C 语言一样编制程序。不过，MATLAB 不象其他的编译语言生成可执行文件，而是象 Basic 语言一样解释执行。跟 C 语言比较，MATLAB 的执行速度比较慢，但是在对于实时性要求不高的地方已经可以满足。

本章一共分为三个基本任务：矩阵的创建、矩阵的运算和 MATLAB 的命令执行。矩阵的创建介绍如何在 MATLAB 中建立一个矩阵，矩阵的运算则详细地介绍矩阵的各种基本运算，最后，介绍了 MATLAB 的两种命令执行的方式，着重讲解 MATLAB 的程序执行方式，为以后的学习打好基础。

任务一 矩阵的创建

任务说明

☆任务内容

矩阵是 MATLAB 最基础、最重要的功能，本节内容将向读者介绍矩阵的创建过程。

☆攻略简介

矩阵是 MATLAB 最基础也是最重要的对象之一，MATLAB 有非常强的矩阵操作和处理能力，我们将看到 MATLAB 能够十分方便地实现矩阵的输入，输出。

☆作用

通过详细的讲解，大量的实例，学会对 MATLAB 的基本操作，掌握 MATLAB 中矩阵的创建方法，能够灵活地运用各种矩阵。

☆起始材料

准备好 MATLAB 软件包，最好对线性代数知识有基本的了解。

攻略一 直接输入矩阵

一、启动 MATLAB

启动 MATLAB 后，进入 MATLAB 工作区。

MATLAB 工作区最上方的两行文字是初始提示信息。左下为闪烁光标，等待用户输入命令，如图 1-1 所示。

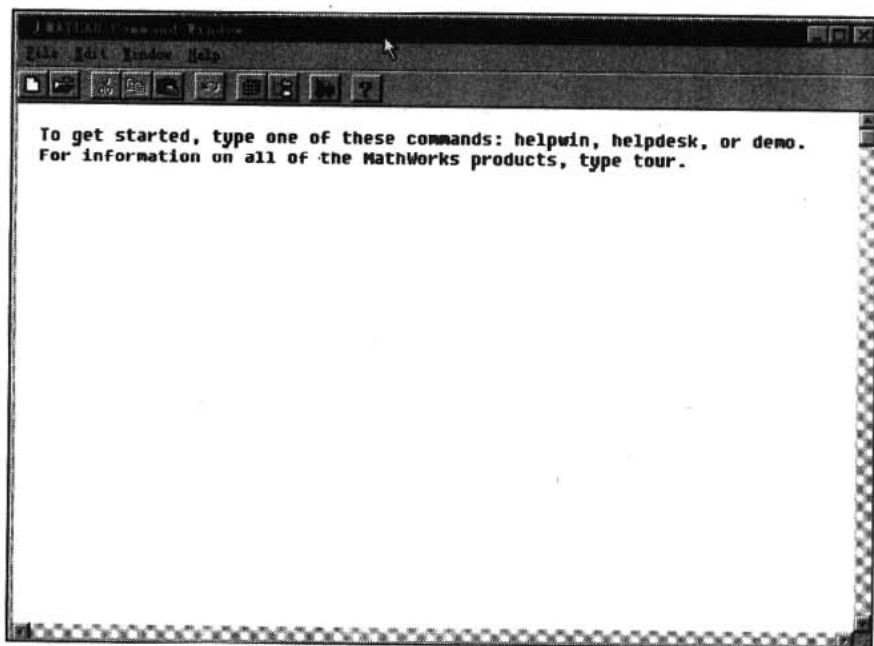


图 1-1 启动 MATLAB

MATLAB 有两种执行方式：命令行方式和程序方式。在工作区中为命令行方式，输入命令，MATLAB 运行并显示结果，例如：

输入：

a=3;

```
b=5;  
a+b  
输出:  
a =  
    3  
b =  
    5  
ans =  
    8
```

MATLAB 显示结果的方式是在命令行中首先显示变量的名称, 然后换行显示变量的值, 可以通过有关选项控制数值显示的格式, 这在后面的章节中会介绍。**MATLAB** 的这种显示方式使得调试工作中的变量监视变得很简单, 只要在命令窗口中键入变量的名字即可。

输入: help

输出:

HELP topics:

matlab\general	- General purpose commands.
matlab \ops	- Operators and special characters.
matlab \lang	- Programming language constructs.
matlab \elmat	- Elementary matrices and matrix manipulation.
matlab \elfun	- Elementary math functions.
matlab \specfun	- Specialized math functions.
matlab \matfun	- Matrix functions - numerical linear algebra.
matlab \datafun	- Data analysis and Fourier transforms.
matlab \polyfun	- Interpolation and polynomials.
matlab \funfun	- Function functions and ODE solvers.
matlab \sparfun	- Sparse matrices.
matlab \graph2d	- Two dimensional graphs.
matlab \graph3d	- Three dimensional graphs.
matlab \specgraph	- Specialized graphs.
matlab \graphics	- Handle Graphics.
matlab \uitools	- Graphical user interface tools.
matlab \strfun	- Character strings.
matlab \iofun	- File input/output.
matlab \timefun	- Time and dates.
matlab \datatypes	- Data types and structures.
matlab \winfun	- Windows Operating System Interface Files (DDE/ActiveX)

matlab\demos	- Examples and demonstrations.
toolbox\compiler	- MATLAB Compiler
comm\comm	- Communications Toolbox
comm\commsfun	- Communications Toolbox SIMULINK S-functions.
comm\commsim	- Communications Toolbox SIMULINK files.
toolbox\fixpoint	- Fixed-Point Blockset
dspblks\dspblks	- DSP Blockset.
dspblks\dspmex	- (No table of contents file)
dspblks\dspdemos	- (No table of contents file)
toolbox\signal	- Signal Processing Toolbox.
stateflow\stateflow	- Stateflow
stateflow\sfdemos	- (No table of contents file)
simulink\simulink	- Simulink
simulink\blocks	- Simulink block library.
simulink\simdemos	- Simulink demonstrations and samples.
simulink\dee	- Differential Equation Editor
toolbox\tour	- MATLAB Tour
toolbox\local	- Preferences.

For more help on directory/topic, type "help topic".

help 是很有用的命令，如果需要某个函数的帮助可输入“**help** 函数名”，例如：**help sin**、**help exp** 等等。

二、输入第一个矩阵

1. 输入矩阵

启动 MATLAB，进入 MATLAB 工作区，输入：

```
A=[1 1 1;2 2 2;3 3 3]
```

则 MATLAB 输出：

```
A=
    1    1    1
    2    2    2
    3    3    3
```

可见矩阵的输入非常简单易懂，注意下面的三个约定：

(1) 元素之间用空格或逗号","间隔。

(2) 用[]把所有元素括起来。

(3) 用分号";"分隔每一行。

矩阵 A 为一全局变量，一直保存在内存中。

例如，输入：

A'

输出：

ans=

```
1 2 3
1 2 3
1 2 3
```

其中 A' 表示矩阵 A 的转置，如图 1-2 所示。

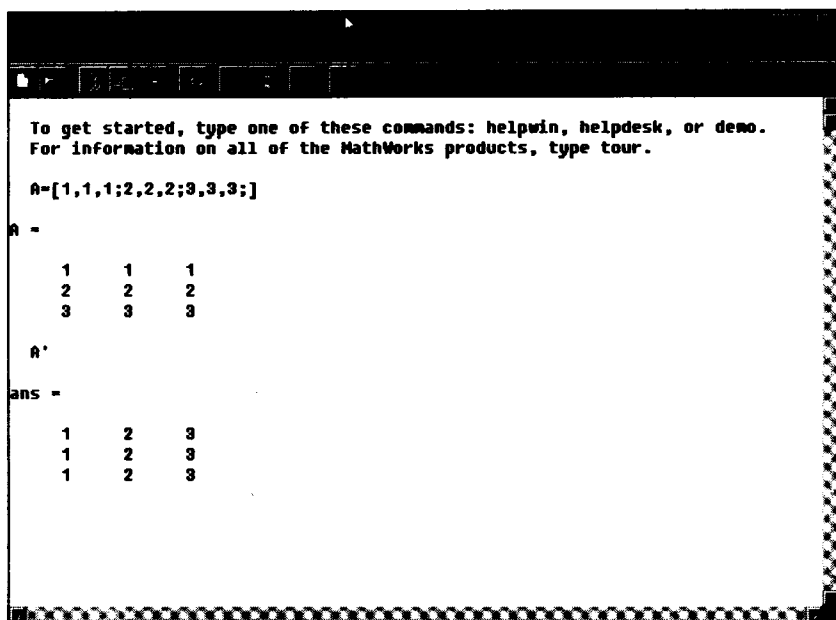


图 1-2 输入第一个矩阵

三、矩阵的元素

1. 表达式

矩阵的元素可以用任意形式的表达式，例如：

```
A=[-8    3+4*5    exp(0);(2+4)/3  1+1    2/2]
```

输出：

A=

```
-8    23    1
 2     2    1
```

2. 矩阵的组合

可以用小矩阵构成大矩阵，例如：

```
A=[10    20    30];
```

```
B=[A; 0    4    9]
```

输出:

```
B=
```

```
10    20    30
```

```
0     4     9
```

注意, 第一句最后有一个";", 表示矩阵 A 的结果不要显示出来, 这对 MATLAB 的编程十分重要, 它使程序不必显示中间结果, 关于编程后面将详细讲解。

又如:

```
A=[3,4,5;6,7,8];
```

```
B=[1+2,4*5,6];
```

```
C=[sin(0.5*pi),sqrt(4),0];
```

```
D=[A;B;C]
```

输出:

```
D=
```

```
3     4     5
```

```
6     7     8
```

```
3    20     6
```

```
1     2     0
```

3. 用 ":" 表示元素序列

对于元素序列, MATLAB 提供了很方便的表达方式, 例如:

```
A=[1:3;1:2:5;0.3:0.1:0.5]
```

输出:

```
A=
```

```
1     2     3
```

```
1     3     5
```

```
0.3   0.4   0.5
```

格式 $n:s:m$ 表示产生从 n 到 m 步长为 s 的一系列值。当 s 缺省时, 步长为 1。步长可以为正、负或小数。

又如:

```
X=[5:-1:1];
```

```
Y=X*2;
```

```
Z=sqrt(X);
```

```
A=[X Y Z]
```

输出:

```
A=
```

```
5   10   2.2361
4    8   2.0000
3    6   1.7321
2    4   1.4142
1    2   1.0000
```

可见用这种方法表示元素序列非常方便,在后面矩阵的运算和程序中将大量采用这种方法。

4. 复数的表示

MATLAB 支持复数的运算,复数的虚部用 i 或 j 表示,例如:

```
a=1+2i
```

或 $a=1+2j$

两者表示的结果一样。

复数可以直接运算,例如:

```
a=3+4i;
```

```
b=5+5i;
```

```
a+b
```

输出:

```
ans =
```

```
8.0000 + 9.0000i
```

乘法:

```
a * b
```

输出:

```
ans =
```

```
-5.0000 +35.0000i
```

模:

```
abs(a/b)
```

输出:

```
ans =
```

```
0.7071
```

下面是复数运算的一些常用函数:

- (1) **abs** 返回复数的模。
- (2) **angle** 返回复数的相角。
- (3) **conj** 返回共轭复数。
- (4) **imag** 返回复数的实部。
- (5) **real** 返回复数的虚部。

矩阵的元素可以是复数,输入复数矩阵有两种表示方法:

(1) $A=[23, 12, 45]+i[2,45,8.9]$ 。

(2) $B=[23+2i,12+45i,45+8.9i]$ 。

以上两个矩阵完全相等。

攻略二 利用函数创建矩阵

一、常用特殊矩阵

1. 空矩阵

MATLAB 中用 “[]” 表示空矩阵，空矩阵的特点：

(1) 空矩阵不包含任何元素，它的维数为 0×0 。

(2) 空矩阵可以在运算中传递。

例如：

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

```
A(2,:)=[]
```

输出：

```
A =
```

```
1     2     3
7     8     9
```

可以看出，空矩阵有矩阵缩维的作用。

2. 单位矩阵、全 1 矩阵、全 0 矩阵

eye 单位矩阵

ones 全 1 矩阵

zeros 全 0 矩阵

例如：

```
A=ones(3,3)
```

```
B=zeros(3,3)
```

```
C=eye(3,3)
```

输出：

```
A =
```

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

```
B =
```

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

```
C =  
    1    0    0  
    0    1    0  
    0    0    1
```

3. 其它特殊矩阵

diag	对角矩阵
compan	伴随矩阵
magic	魔方矩阵
rand	随机矩阵
randn	正态随机矩阵
tril	下三角矩阵
triu	上三角矩阵

例如:

```
A=magic(3)
```

输出:

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

A 矩阵的元素由 1~10 组成, 行、列和对角线之和相等。

```
A=rand(3,3)
```

```
B=tril(A)
```

输出:

```
A =  
    0.9501    0.4860    0.4565  
    0.2311    0.8913    0.0185  
    0.6068    0.7621    0.8214  
  
B =  
    0.9501         0         0  
    0.2311    0.8913         0  
    0.6068    0.7621    0.8214
```

二、特殊矩阵的生成

1. 矩阵的变换

rot90	矩阵逆时针旋转 $n \times 90$ 度
-------	-------------------------

`fliplr` 矩阵左右翻转

`flipud` 矩阵上下翻转

例如:

```
A=rand(3,3)
```

```
B=rot90(A,3)
```

```
C=fliplr(A)
```

```
D=flipud(A)
```

输出:

A =

0.4447	0.9218	0.4057
0.6154	0.7382	0.9355
0.7919	0.1763	0.9169

B =

0.7919	0.6154	0.4447
0.1763	0.7382	0.9218
0.9169	0.9355	0.4057

C =

0.4057	0.9218	0.4447
0.9355	0.7382	0.6154
0.9169	0.1763	0.7919

D =

0.7919	0.1763	0.9169
0.6154	0.7382	0.9355
0.4447	0.9218	0.4057

2. 矩阵的部分元素的抽取

利用以上的特殊矩阵, 可以非常方便的抽取矩阵中地元素。

例如:

```
A=[1,2,3,4;5,6,7,8;10,12,14,16]
```

```
B=diag(A)
```

```
C=triu(A,1)
```

```
D=triu(A,-1)
```

输出:

A =

1	2	3	4
5	6	7	8
10	12	14	16

B =

1
6
14

C =

0 2 3 4
0 0 7 8
0 0 0 16

D =

1 2 3 4
5 6 7 8
0 12 14 16

三、稀疏矩阵的存储

1. sparse 函数

在实际应用中，常常会遇到大量的矩阵处理，而这些矩阵中的大部分元素为 0，称其为稀疏矩阵。由于稀疏矩阵的大多数元素为 0，用完全矩阵的方法计算比较浪费，MATLAB 提供了很好的方法来专门处理稀疏矩阵。

sparse 函数用来生成稀疏矩阵。sparse 函数有两种常用格式：

(1) sparse(A) 用于把完全矩阵压缩为稀疏矩阵。

例如：

A=[0,1,0,0;0,3,0,4;5,0,0,0;0,0,0,7]

A =

0 1 0 0
0 3 0 4
5 0 0 0
0 0 0 7

sparse(A)

ans =

(3,1) 5
(1,2) 1
(2,2) 3
(2,4) 4
(4,4) 7

(2) sparse(i,j,u)函数。

sparse(i,j,u)函数用于直接生成稀疏矩阵。i,j 为向量，分别对应行号和列号，u 也为向

量，存储非 0 元素的值。

例如：

```
%用直接输入法输入前面的稀疏矩阵
```

```
i=[1,2,2,3,4];
```

```
j=[2,2,4,1,4];
```

```
u=[1,3,4,5,7];
```

```
A=sparse(i,j,u)
```

```
A =
```

```
(3,1)      5
```

```
(1,2)      1
```

```
(2,2)      3
```

```
(2,4)      4
```

```
(4,4)      7
```

2. full 函数

full 函数把稀疏矩阵还原为完全矩阵。

例如：

```
%把前面的稀疏矩阵还原为完全矩阵
```

```
full(A)
```

```
ans =
```

```
0     1     0     0
```

```
0     3     0     4
```

```
5     0     0     0
```

```
0     0     0     7
```

攻略三 利用数组创建矩阵

一、数组的表示

1. 数组与矩阵

MATLAB 中数组与矩阵的概念基本相同，向量相当于一维数组，矩阵相当于二维数组，矩阵可以用数组直接引用，例如：

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

```
x=A(1,1)
```

```
y=A(2,3)
```

```
A(3,1)=0
```

返回：

```
x =
```



```

1
y =
6
A =
1    2    3
4    5    6
0    8    9

```

2. 下标

数组的下标可以用标量、向量、冒号来表示：

(1) 标量表示 $A(n,m)$ 。

n,m 都为整数，其中 n 为行号， m 为列号。

(2) 向量表示 $A(u,v)$ 。

u,v 为向量， u 的元素对应所在行， v 的元素对应所在列。

例如：

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

```
B=A(2:3,1)
```

```
C=A(1:2,[1,2])
```

```
A([1,3],[1,3])=0
```

返回：

```
B =
```

```
4
```

```
7
```

```
C =
```

```
1    2
```

```
4    5
```

```
A =
```

```
0    2    0
```

```
4    5    6
```

```
0    8    0
```

(3) 冒号表示 $A(:,u)$

冒号 “:” 表示全部的行(列)。

例如：

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

```
B=A(1,:)
```

```
C=A(:,[2,3])
```

```
D=A(:,)
```