

M 语言数据类型

数据的可视化

文件操作

Matlab 和 Simulink 混合编程

编写图形用户界面程序

从 M 语言转成独立应用和 C/C++ 代码

从 VC++ 调用函数

从 VC++ 调用 MATLAB C Math Library

外部接口

# MATLAB

## M 语言高级编程

陈永春 编著



清华大学出版社

# MATLAB M 语言高级编程

陈永春 编著

清华大学出版社

北 京

## 内 容 简 介

MATLAB 提供了一种演算纸方式的编程语言: M 语言。本书力图从各个层面系统地阐述 M 语言, 让读者对 M 语言有一个全面的了解。本书的主要内容有: 简单介绍 MATLAB M 语言的一些特点、M 语言的逻辑和循环控制语句以及异常处理语句, 介绍 MATLAB 提供的最强大的功能之一——数据的可视化, 同时介绍了 Matlab 和 Simulink 混合编程的问题, 包括如何从 Simulink 模型中调用 M 文件及从 M 文件中调用 Simulink 模型等内容。

本书语言通畅、条理清晰、内容详细, 主要适用那些以 MATLAB 作为开发/仿真/实现工具的科技人员以及开发人员, 作为他们进行 MATLAB M 语言编程的必备参考书。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

### 图书在版编目 (CIP) 数据

MATLAB M 语言高级编程/陈永春编著. —北京: 清华大学出版社, 2003  
ISBN 7-302-07514-X

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

中国版本图书馆 CIP 数据核字 (2003) 第 097453 号

出 版 者: 清华大学出版社                      地      址: 北京清华大学学研大厦

<http://www.tup.com.cn>      邮      编: 100084

社 总 机: 010-62770175      客 户 服 务: 010-62776969

责任编辑: 肖 丽

封面设计: 钱 诚

版式设计: 张红英

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开      本: 185 × 260      印 张: 15      字 数: 330 千字

版      次: 2004 年 1 月第 1 版      2004 年 1 月第 1 次印刷

书      号: ISBN 7-302-07514-X/TP · 5537

印      数: 1 ~ 5000

定      价: 24.00 元

---

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704。

# 引 言

MATLAB 提供了一种演算纸方式的编程语言：M 语言。M 语言的编程，相对于 C 语言而言比较简单，但仍然有一些自己的规则。本书作为《从 Matlab/Simulink 模型到代码实现》的姊妹篇，力图从各个层面系统地阐述 M 语言，让读者对 M 语言有一个全面的了解。

本书的内容如下：

- 第 1 章简单地论述了 MATLAB M 语言的一些特点，包括支持演算纸形式的编程、变量无需先定义类型等。
- 第 2 章介绍了 M 语言支持的各种数据类型，包括字符串、数值数据类型、元胞数组、结构等。
- 第 3 章阐述了 M 语言的逻辑和循环控制语句以及异常处理语句。
- 第 4 章介绍了 MATLAB 提供的最强大的功能之一——数据的可视化，阐述了二维/三维图形的绘制、标注以及如何修改图形对象的属性等内容。
- 第 5 章介绍了 M 语言文件操作各个方面的内容。
- 第 6 章介绍了 Matlab 和 Simulink 混合编程的问题，包括如何从 Simulink 模型中调用 M 文件，以及如何从 M 文件中调用 Simulink 模型。
- 第 7 章介绍了如何用 M 语言开发图形界面应用程序的问题。
- 第 8 章阐述了如何利用 MATLAB Compiler 将 M 程序转成独立的应用，如何从 VC++ 中调用 M 函数，以及如何从 VC++/VB 中调用 MATLAB 科学计算及可视化功能等问题。
- 第 9 章简单介绍了 MATLAB 提供的众多外部接口。

# 申 明

本书中有以下三点需要特殊说明：

- 本书中 `>>command_name/command_name(arg1,...,argN)` 表示在 MATLAB 命令窗口输入命令 `command_name` 或者 `command_name(arg1,...,argN)`，其中 `command_name` 可以是一个脚本 M 文件名或者函数名，`>>` 为默认的 MATLAB 命令提示符。
- 本书中 `%MATLABROOT%` 指 MATLAB 安装目录。
- 本书的论述基于 MATLAB R12.1 版本。

如果本书的出版能给使用 MATLAB 的读者带来一些帮助和启发, 编者将深感荣幸! 欢迎读者朋友通过 E-mail: [yongchunchen@hotmail.com](mailto:yongchunchen@hotmail.com) 和我交流使用的经验或者讨论相关的问题。读者也可登录北京九州恒润科技有限公司的网址 <http://www.hirain.com> 或者 MathWorks 公司的网址 <http://www.mathworks.com> 查询有关 MATLAB 的最新信息。有关 MATLAB 的购买事宜, 请与北京九州恒润科技有限公司联系。

陈永春  
2003年5月6日

# 目 录

<b>第 1 章 M 语言基础</b> .....	<b>1</b>
1.1 演算纸方式的计算.....	1
1.2 变量无需定义类型.....	3
1.3 变量都是矩阵吗.....	4
1.4 矩阵操作.....	5
1.5 M 文件类型.....	20
1.6 如何获得帮助.....	28
<b>第 2 章 数据类型</b> .....	<b>30</b>
2.1 变量都是双精度浮点数吗.....	30
2.2 char (字符).....	33
2.3 数值数据类型.....	36
2.4 cell (元胞) 数组.....	40
2.5 Structure (结构).....	42
2.6 Java class (Java 类).....	44
2.7 function handle (函数句柄).....	49
2.8 一个例子.....	53
<b>第 3 章 控制语句</b> .....	<b>56</b>
3.1 逻辑控制语句.....	56
3.2 循环控制语句.....	58
3.3 异常处理语句.....	61
3.4 内存预分配.....	62
<b>第 4 章 数据的可视化</b> .....	<b>64</b>
4.1 二维图形绘制.....	64
4.2 图形标注.....	67
4.3 子图.....	73
4.4 三维图形绘制.....	74
4.5 专用绘图语句.....	77
4.6 图形 I/O.....	86
4.7 图形句柄.....	97

第 5 章 文件操作.....	116
5.1 高级操作.....	116
5.2 低级操作.....	122
5.3 输入向导 uiimport.....	125
第 6 章 Matlab 和 Simulink 混合编程.....	130
6.1 Matlab 和 Simulink 的接口.....	130
6.2 从 Simulink 模型中调用 M 文件.....	130
6.3 从 M 文件中调用 Simulink 模型.....	137
第 7 章 图形用户界面设计.....	141
7.1 GUIDE 简介.....	141
7.2 回调函数.....	146
7.3 图形用户界面设计的一个实例.....	147
第 8 章 将 M 程序转换成独立的应用.....	162
8.1 选择 C/C++ compiler.....	163
8.2 从 M 程序自动生成独立的应用和 C/C++源代码.....	166
8.3 不是所有的 M 程序都能转换成 C/C++代码.....	181
8.4 从 VC++中调用 m 函数.....	188
8.5 从 VC++中调用 MATLAB C Math Library.....	193
8.6 从 VB 中调用 MATLAB 功能.....	208
第 9 章 外部接口.....	214
9.1 MEX 文件.....	214
9.2 MATLAB 引擎.....	218
9.3 ActiveX.....	223
9.4 Java 接口.....	224
9.5 串口操作.....	227
参考文献.....	229

# 第 1 章 M 语言基础

MATLAB 是美国 MathWorks 公司开发的用于教育、工程与科学计算的软件产品，它向用户提供从概念设计、算法开发、建模仿真到实时实现的理想集成环境。无论是进行科学研究、产品开发，还是从事教育事业，MATLAB 产品都是非常有效的工具。相对于其他类似于 MATLAB 的仿真软件，MATLAB 的一个显著特点就是它提供了一种用于编程的高级语言——M 语言。通过这种语言，用户可以用类似于数学公式的方式来编写算法，大大降低了编程所需的难度并节省了时间，从而让用户把主要的精力集中在算法的构思而不是编程上。

## 1.1 演算纸方式的计算

MATLAB 提供了演算纸方式的计算，只要使用 M 语言语法，在 MATLAB 的命令窗口就可以实现这种计算。以下则是一些简单的 M 语言及其计算结果。

```
>>clear all
>>var1=1000

var1 =

    1000

>>A=[1 2;3 4]

A =

     1     2
     3     4

>>rank(A)

ans =

     2

>>inv(A)
```



```
ans =  
-2.0000    1.0000  
 1.5000   -0.5000
```

```
>> eig(A)
```

```
ans =  
-0.3723  
 5.3723
```

```
>> [U,S,V]=svd(A)
```

```
U =
```

```
-0.4046   -0.9145  
-0.9145    0.4046
```

```
S =
```

```
 5.4650    0  
    0    0.3660
```

```
V =
```

```
-0.5760    0.8174  
-0.8174   -0.5760
```

```
>> B=[5 6;7 8]
```

```
B =
```

```
 5    6  
 7    8
```

```
>> C=A+B
```

```
C =
```

```
 6    8  
10   12
```

```
>> D=A*B
```

```
D =  
  
    19    22  
    43    50  
  
>>E=A/B  
  
E =  
  
    3.0000   -2.0000  
    2.0000   -1.0000
```

M语言生成的变量都存储在 MATLAB 的基本工作空间。

```
>>whos  
  
Name      Size      Bytes Class  
  
A         2x2       32 double array  
B         2x2       32 double array  
C         2x2       32 double array  
D         2x2       32 double array  
E         2x2       32 double array  
S         2x2       32 double array  
U         2x2       32 double array  
V         2x2       32 double array  
ans       2x1       16 double array  
var1      1x1        8 double array  
  
Grand total is 35 elements using 280 bytes  
Grand total is 34 elements using 272 bytes
```

## 1.2 变量无需定义类型

就像前面所看到的那样，在 MATLAB 中，在给变量赋值之前，不需要定义它的类型。例如 1.1 节中对变量 `var1` 赋值 1000，并没有事先定义 `var1` 的数据类型。MATLAB 会自动决定变量的类型，并为它分配内存空间。对上述变量 `var1`，MATLAB 将它默认定义为双精度浮点型，分配 8 个字节的存储空间。M 语言中的变量名（包括函数名）是以英文字母开头的英文字母、下划线和阿拉伯数字的组合，有效长度不超过 31。

### 1.3 变量都是矩阵吗

M 语言支持类似于数学公式的编程。例如， $C=A+B$  就可完成矩阵 A 和矩阵 B 的相加运算，并把结果存储在 C 中。MATLAB 中所有的变量都没有维数的限制（维数自动扩展），并且是以数组（array）的方式存储。但在数学意义上，基本上可以把所有的变量都当作矩阵来理解，尤其是对数值变量（对于结构数组，元胞数组最好不要当作矩阵来理解）。例如在  $C=A+B$  中，变量 A 和 B 都是以  $2 \times 2$  维数组的方式存储的（存储方式为按列存储，而 C/C++ 中的数组变量是按行存储的，这个区别需要注意），在数学意义上可将 A 和 B 当作两个  $2 \times 2$  维的矩阵， $C=A+B$  完成的便是两个矩阵的相加运算。同样的， $[U,S,V]=\text{svd}(A)$  实现对矩阵 A 的奇异值分解。在 MATLAB 中所有变量的维数都可自动扩展，但始终保持它的矩形结构。例如：

```
>>var1=1000

var1 =

    1000

>>whos var1

Name          Size          Bytes Class
var1          1x1            8 double array

Grand total is 1 elements using 8 bytes

>>var1(10)=100
var1 =

Columns 1 through 9
1000    0    0    0    0    0    0    0    0

Column 10
    100

>>whos var1
```

```
Name          Size          Bytes Class
var1          1x10          80 double array
Grand total is 10 elements using 80 bytes

>>var1(2,5)=15

var1 =

Columns 1 through 9

1000    0    0    0    0    0    0    0    0
0    0    0    15    0    0    0    0

Column 10

    100
    0

>>whos var1

Name          Size          Bytes Class
var1          2x10          160 double array
Grand total is 20 elements using 160 bytes
```

## 1.4 矩阵操作

前面讲到，在数学意义上，基本上可以把所有的数值变量都当作矩阵来理解。在这个意义上，来看一下 MATLAB 中的矩阵操作。

### 1. 矩阵的输入

在 MATLAB 的命令窗口或者 M 文件中输入矩阵的方式如下：

- 分号 (;) 为行分隔符，逗号 (,) 或者空格为列分隔符，中括号 ([ ]) 为矩阵输入符

```
>>a=[1,2;3 4]
```

```
a =
     1     2
     3     4
```

- 使用冒号 (:) 生成数列，一般形式为 A:B:C，其中 A 为初值，B 为步长，C 为终值，默认步长值为 1（输入形式为 A:C）

```
>>b=2:-0.5:0
b =
     2.0000     1.5000     1.0000     0.5000     0
>> b1=1:5
b1 =
     1     2     3     4     5
```

- 使用函数生成矩阵

```
>>c=rand(2,4)
c =
     0.9501     0.6068     0.8913     0.4565
     0.2311     0.4860     0.7621     0.0185
>>d=randsrc(10,10,[0 1 2 3;0.25 0.25 0.3 0.2])
d =
     2     2     1     1     1     0     2     1     3     0
     2     0     1     3     2     2     2     2     3     3
     0     1     0     2     1     2     0     0     0     1
     1     1     2     0     0     0     1     1     0     1
     2     3     1     3     0     0     1     2     0     1
     2     0     3     2     1     0     1     3     0     1
     1     2     2     0     0     2     0     1     2     1
     2     3     1     0     1     0     2     1     0     2
     2     3     2     2     2     1     2     3     3     0
     0     2     1     2     0     2     2     0     0     0
```

```
>>e=magic(5)

e =

    17     24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>>f=pascal(6)

f =

     1     1     1     1     1     1
     1     2     3     4     5     6
     1     3     6    10    15    21
     1     4    10    20    35    56
     1     5    15    35    70   126
     1     6    21    56   126   252
```

### ■ 使用 load 打开 MATLAB 专用的数据格式文件 (MAT 文件)

```
>>load my_data.mat
```

MATLAB 中的矩阵必须是矩形的，没有定义的元素默认为零，例如：

```
>>h(5,6)=100

h =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0    100
```

虽然只定义了一个元素 (5,6)，MATLAB 自动将 h 扩展为一个 5×6 维的矩阵，没有定义的元素默认为零。

## 2. 矩阵索引

如图 1.1 所示，在 MATLAB 中有两种方法可以对矩阵元素进行索引。

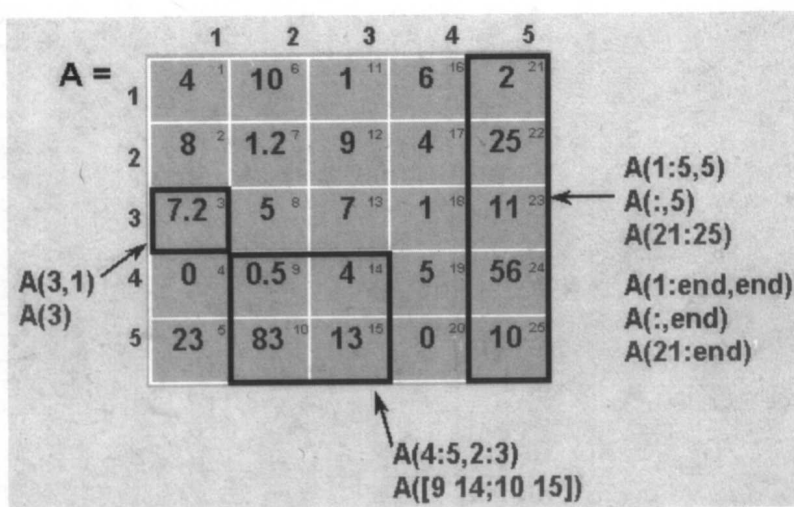


图 1.1 矩阵索引

### ■ 使用行、列的形式

MATLAB 从矩阵的左上角开始索引，索引的初始值是 1。例如要访问图 1.1 中矩阵 A 中值为 9 的元素，可以在圆括号中以相应的行、列值指明。

```
>>A(2,3)
```

```
ans =
```

```
9
```

### ■ 使用绝对下标索引

MATLAB 最初是用 FORTRAN 语言编写的，FORTRAN 语言中对数组数据的存储是按列依次进行的。尽管 MATLAB 的数学代码现在已经用 C 语言重写了（C 语言中对数组数据的存储是按行依次进行的），这种性质却保存了下来，也就是说，在 MATLAB 中对矩阵的绝对下标索引是按从上到下、从左到右的顺序进行的，初始值是 1。例如，使用绝对下标索引从 A 中提取 9 的做法如下：

```
>> A(12)
```

```
ans =
```

```
10
```

索引值本身必须是自然数，但被索引的对象可以是：

■ 一个元素单元

```
>>A(3,1)

ans =

    7.2000

>>A(3)

ans =

    7.2000
```

■ 一串连续的元素

```
>>A(1:5,5)

ans =

    2
   25
   11
   56
   10

>>A(:,5)

ans =

    2
   25
   11
   56
   10

>>A(21:25)

ans =

    2    25    11    56    10
```



```
>>A(1:end,end)
ans =

     2
    25
    11
    56
    10

>>A(:,end)
ans =

     2
    25
    11
    56
    10

>>A(21:end)
ans =

     2     25     11     56     10
```

#### ■ 任意元素列表

```
>>A(4:5,2:3)
ans =

     0.5000     4.0000
    83.0000    13.0000

>>A([9 14;10 15])
ans =

     0.5000     4.0000
    83.0000    13.0000
```