

- 合理、完善的知识体系结构
- 内容丰富，重点突出，应用性强
- 免费提供相关程序源代码下载
- 深入、详细剖析 MATLAB 工程应用技术

MATLAB
工程应用书库

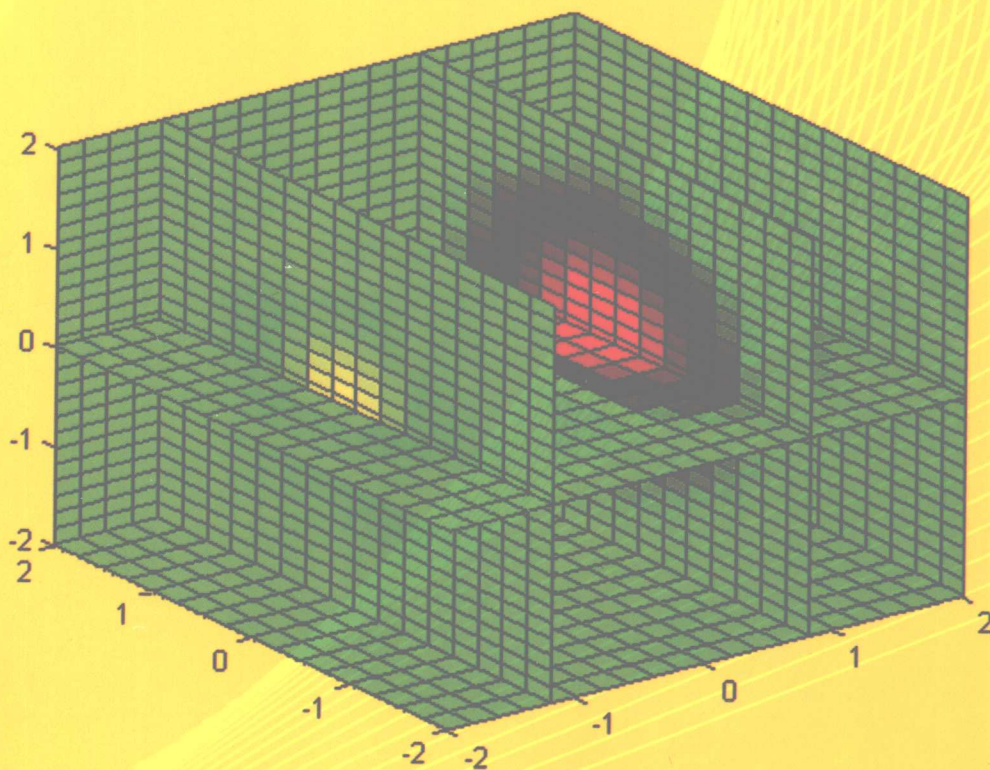
MATLAB

与外部程序接口编程



网上提供源代码下载
www.cmpbook.com

张德丰 等编著



机械工业出版社
CHINA MACHINE PRESS



TP391.75Ma
552
1=

MATLAB 工程应用书库

MATLAB 与外部程序接口编程

张德丰 等编著



机械工业出版社

本书系统地介绍了 MATLAB 与外部程序的接口方法和技巧,重点介绍了 MATLAB 与各种高级语言之间的交互以及 MATLAB 提供的各种计算引擎和服务器引擎。书中给出了大量的 MATLAB 与高级语言结合的编程实例,可帮助读者更好地理解 and 掌握 MATLAB 应用程序接口的使用方法。通过 MATLAB 提供的串口接口,可以实现从外围设备(如 MODEM)直接输入数据到 MATLAB 工作空间,再利用 MATLAB 进行处理。

本书可作为计算机、电子学、信息科学、通信、控制等专业的本科生、研究生以及其他专业技术人员学习 MATLAB 与外部程序接口技术的教材或参考书。

图书在版编目(CIP)数据

MATLAB 与外部程序接口编程 / 张德丰等编著. —北京:机械工业出版社, 2009.1

(MATLAB 工程应用书库)

ISBN 978-7-111-25706-6

I. M… II. 张… III. 计算机辅助计算—软件包, MATLAB—程序设计 IV. TP391.75

中国版本图书馆 CIP 数据核字(2008)第 192921 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:丁 诚 吴鸣飞

责任编辑:丁 诚 唐洪昌

责任印制:李 妍

保定市中画美凯印刷有限公司印刷

2009 年 1 月第 1 版·第 1 次印刷

184mm×260mm·23.75 印张·588 千字

0001—4000 册

标准书号:ISBN 978-7-111-25706-6

定价:42.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379753 88379739

封面无防伪标均为盗版



前 言

正如 FORTRAN 和 C 等高级语言使人们摆脱了需要直接对计算机硬件资源进行操作一样,被称作第四代计算机语言的 MATLAB,利用其丰富的函数资源,能减轻编程人员的工作量。下面简单介绍一下 MATLAB 的主要特点。

- ◆ 语言简洁紧凑,使用方便灵活,库函数极其丰富。MATLAB 程序书写形式自由,利用丰富的库函数可避开繁杂的子程序编程任务,压缩了一些编程工作量。由于库函数都由本领域的专家编写,均已通过检验。

- ◆ 运算符丰富。由于 MATLAB 是用 C 语言编写的, MATLAB 提供了和 C 语言几乎一样多的运算符,灵活使用 MATLAB 的运算符将使程序变得极为简短。

- ◆ MATLAB 既具有结构化的控制语句(如 for 循环、while 循环、break 语句和 if 语句),又有面向对象编程的特性。

- ◆ 程序限制不严格,程序设计自由度大。例如,在 MATLAB 里,用户无需对矩阵预定义就可使用。

- ◆ 程序的可移植性很好,基本上不做修改就可以在各种型号的计算机和操作系统上运行。

- ◆ MATLAB 的图形功能强大。在 FORTRAN 和 C 语言里,绘图都很不容易,但在 MATLAB 里,数据的可视化非常简单。MATLAB 还具有较强的编辑图形界面的能力。

- ◆ MATLAB 的缺点是,程序的执行速度和其他高级程序相比较慢。由于 MATLAB 的程序不用编译等预处理,也不生成可执行文件,程序为解释执行,所以速度较慢。

- ◆ 功能强大的工具箱是 MATLAB 的另一特色。MATLAB 包含两个部分:核心部分和各种可选的工具箱。核心部分中有数百个核心内部函数,工具箱又分为功能性工具箱和学科性工具箱两类。这些工具箱都是由该领域内学术水平很高的专家编写的,所以用户无需编写自己学科范围内的基础程序。

- ◆ 源程序的开放性。开放性也许是 MATLAB 最受人们欢迎的特点。除内部函数以外,所有 MATLAB 的核心文件和工具箱文件都是可读可改的源文件,用户可通过对源文件的修改并加入自己的文件,构成新的工具箱。

作为目前市面上最为强大的科学计算软件, MATLAB 实现了与众多外部程序或设备的接口。本书系统地介绍了 MATLAB 与外部程序的接口方法和技巧,重点介绍了 MATLAB 与各种高级语言之间的交互以及 MATLAB 提供的各种计算引擎和服务引擎。书中给出了大量的 MATLAB 与高级语言结合的编程实例,使读者能够更好地理解和掌握 MATLAB 应用程序接口的使用方法。

全书共分 11 章。第 1 章包括 MATLAB 系统简介、MATLAB 常用的数据类型以及 MATLAB 接口简介等内容;第 2 章介绍了 C-MEX 混合编程、FORTRAN 语言的 MEX 文件以及操作 MAT 文件等内容;第 3 章介绍了文件的打开和关闭、二进制数据以及使用文件 I/O 函数等内容;第 4 章介绍了 mcc 命令、M 文件与 C/C++ 混合创立可独立运行的程序以及

MATLAB 和 C++ 接口中函数注册等内容；第 5 章介绍了在 Visual C++ 中调用 MATLAB 引擎时的环境设置、CMATLABEng 应用实例以及在 Visual FORTRAN 中使用 MATLABEngine 等内容；第 6 章介绍了 MATLAB 与 C 语言混合编程数据类型、MATLAB 调用 C 在 C++ 中调用 MATLAB C++ 数学库函数等内容；第 7 章包括 MATLAB COM Builder 简介、MATLAB COM Builder 与 Visual C++ 之间的数据转换以及 MATLAB COM Builder 的枚举类型等内容；第 8 章介绍了向 MATLAB 中引入 Java 类库、创建和使用 Java 对象以及 Delphi 调用 Mideva 生成的动态链接库等内容；第 9 章介绍了 C++ Builder 与 MATLAB、MATLAB 和 Excel 的混合编程以及 MATLAB 与 Visual Basic 语言接口编程等内容；第 10 章介绍了使用 Matrix<LIB>、在 Visual C++ 中使用 Matcom C++ 矩阵库以及 Matcom C++ 矩阵库的图形和图像显示功能等内容；第 11 章介绍了串口接口、数据的读写以及保存和装载数据等内容。

本书内容丰富、图文并茂、文字流畅，是一本学习和使用 MATLAB 与外部程序接口方法和技巧的有价值的参考书。

参加本书编写的人员有张德丰、许华兴、王旭宝、王孟群、邓恒奋、卢国伟、卢焕斌、伍志聪、庄文华、庄浩杰、许业成、何沛彬、何佩贤、张水兰、张坚、李勇杰、李秋兰、李美妍、陈运英、陈景棠、梁家科、黄达中、陈楚明、林健锋、梁劲强、林振满、周品等。

由于时间仓促，本书错误或疏漏之处在所难免，敬请读者批评指正。

编者

目 录

前言	
第 1 章 MATLAB 外部接口概述	1
1.1 MATLAB 系统简介	1
1.2 MATLAB 常用的数据类型	2
1.2.1 数值阵列	2
1.2.2 字符阵列	4
1.2.3 元组阵列	5
1.2.4 结构体阵列	7
1.2.5 类和对象	9
1.3 MATLAB 接口简介	10
1.3.1 MEX 文件	10
1.3.2 MATLAB 引擎	11
1.3.3 MATLAB C/C++ 数学函数库	11
1.3.4 MATLAB 编译器	12
1.3.5 串口接口	13
第 2 章 MATLAB 的 MEX 及 MAT 文件	14
2.1 C 语言 MEX	14
2.2 C-MEX 混合编程	16
2.3 MEX 文件的创立与调用	18
2.3.1 工程实例	18
2.3.2 编译器选项	21
2.3.3 脚本 M 文件向函数 M 文件的转化	22
2.3.4 MEX 独立应用程序的发布	24
2.3.5 MEX 的编程实例	25
2.4 FORTRAN 语言的 MEX 文件	29
2.4.1 简单的 FORTRAN 语言 MEX 文件示例	29
2.4.2 传递字符串变量	31
2.4.3 在 FORTRAN 语言中调用 MATLAB 命令	33
2.5 操作 MAT 文件	35
2.5.1 MAT 文件格式	35
2.5.2 操作 MAT 文件的 MATLAB API	36
2.6 Visual C++ 调用 MAT 时的环境设置	39
2.7 MAT 文件应用举例	40
第 3 章 MATLAB 数值运算及数据的导入/导出	45





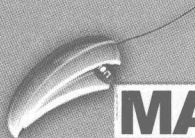
3.1 数值运算	45
3.1.1 矩阵的构造	45
3.1.2 矩阵的基本运算	48
3.1.3 矩阵的函数运算	50
3.1.4 矩阵分解	51
3.2 文本的打开和关闭	55
3.3 二进制数据	56
3.3.1 二进制数据的导入	56
3.3.2 二进制数据的导出	58
3.4 数据分析和统计	59
3.4.1 基本数据分析函数	59
3.4.2 协方差和相关系数函数	65
3.4.3 有限差分	66
3.5 使用文件 I/O 函数	67
3.5.1 二进制数据的读取	68
3.5.2 二进制数据的写入	69
3.5.3 逐行读取文本文件	69
3.5.4 格式化写入文本数据	70
3.6 稀疏矩阵	71
3.6.1 稀疏矩阵的创建和存储	71
3.6.2 查看稀疏矩阵	74
3.6.3 稀疏矩阵的运算	75
第 4 章 MATLAB 编译器	77
4.1 MATLAB 编译器简介	77
4.1.1 MATLAB 编译器使用	77
4.1.2 编译 M 文件的原因	78
4.1.3 MATLAB 编译器的局限性	79
4.2 mcc 编译器典型应用	79
4.2.1 独立可执行文件	79
4.2.2 C 动态链接库	84
4.2.3 C++ 动态链接库	85
4.2.4 C/C++ 动态链接库的不同之处	86
4.3 进一步了解 mcc 命令	87
4.3.1 mcc 常用命令选项	87
4.3.2 捆绑命令文件	88
4.4 M 文件与 C/C++ 混合创立可独立运行的程序	88
4.5 控制代码生成	91
4.5.1 控制代码生成介绍	91

4.5.2	编译器生成的头文件	93
4.5.3	内部接口函数	95
4.5.4	编译器支持的可执行类型	98
4.5.5	M 代码与 C/C++ 代码的接口	102
4.6	MATLAB 和 C++ 接口中函数注册	103
4.7	综合实例	106
第 5 章	MATLAB 引擎的混合编程	110
5.1	MATLAB 引擎函数简介	110
5.2	MATLAB 引擎函数库	111
5.3	在 Visual C++ 中调用 MATLAB 引擎时的环境设置	116
5.4	MATLAB 引擎类的封装	124
5.4.1	CMATLABEng 类的定义和实现代码	124
5.4.2	CMATLABEng 说明和使用方法	128
5.5	CMATLABEng 应用实例	128
5.6	在 Visual FORTRAN 中使用 MATLABEngine	131
第 6 章	MATLAB 与 C/C++ 语言的接口	136
6.1	MATLAB 与 C 语言混合编程数据类型	136
6.1.1	MATLAB C 语言接口中特殊的数据类型	136
6.1.2	size_t 类型	138
6.2	MATLAB C 语言接口数据类型	138
6.2.1	MATLAB 普通数值阵列的操作	138
6.2.2	稀疏数组阵列	140
6.2.3	MATLAB 结构体阵列	143
6.2.4	MATLAB 元组	146
6.2.5	MATLAB 字符阵列	148
6.3	MATLAB 调用 C	149
6.3.1	MEX 文件接口	150
6.3.2	处理标量数据	150
6.3.3	处理字符串数据	152
6.3.4	处理稀疏矩阵	153
6.3.5	处理结构体和单元阵列	156
6.4	在 C++ 中调用 MATLAB C++ 数学库函数	159
6.4.1	MATLAB C++ 数学库函数介绍	159
6.4.2	对 MATLAB 阵列的操作	163
6.4.3	索引	172
6.4.4	基于 I/O 流的阵列输入	178
6.4.5	库函数调用	182
6.4.6	数学运算符的使用	186

6.5	MATLAB C++工具函数	189
第7章	MATLAB COM Builder 与 Visual C++	192
7.1	COM 基础知识	192
7.2	MATLAB COM Builder 简介	193
7.3	COM Builder 基础知识	193
7.3.1	配置 MATLAB C/C++编译器	193
7.3.2	创建 MATLAB COM Builder 组件	194
7.3.3	MATLAB COM Builder 工具库	197
7.3.4	在 Visual C++中调用 COM 组件的步骤	198
7.4	MATLAB COM Builder 与 Visual C++之间的数据转换	199
7.4.1	VARIANT 数据类型	199
7.4.2	SAFEAPPAY 数据类型	201
7.4.3	SAFEAPPAY 的创建函数	202
7.4.4	MATLAB COM Builder 与 Visual C++之间的数据转换	203
7.5	MATLAB COM Builder 的枚举类型	207
7.6	综合实例	209
第8章	MATLAB 与 Java、Delphi 接口编程	212
8.1	向 MATLAB 中引入 Java 类库	213
8.2	创建和使用 Java 对象	216
8.2.1	创建 Java 对象	216
8.2.2	Java 对象的链接	217
8.2.3	存储和装载对象	218
8.2.4	使用 Java 对象	222
8.3	在 MATLAB 中使用 Java 数组	224
8.3.1	Java 数组的创建	225
8.3.2	为 Java 数组赋值	226
8.3.3	访问 Java 数组元素	227
8.3.4	Java 数组的引用和复制	229
8.3.5	Java 数组的链接	230
8.4	向 Java 对象传递数据	231
8.5	处理 Java 方法调用返回的数据	235
8.6	综合实例	236
8.7	利用 MATLAB 引擎实例混合编程	245
8.7.1	动态链接库介绍	245
8.7.2	在 Delphi 中调用 Visual C++创建的动态链接库的实例	245
8.7.3	MATLAB 引擎动态链接库的设计	250
8.8	Delphi 调用 Mideva 生成的动态链接库	254
8.8.1	Mideva 介绍	254

8.8.2 应用实例	254
8.9 通过外部调用实现混合编程	259
8.9.1 外部调用方法介绍	259
8.9.2 应用实例	260
第9章 MATLAB 与其他语言的混合编程	262
9.1 C++ Builder 与 MATLAB	262
9.1.1 Mideva 软件平台	262
9.1.2 C++ Builder 直接调用 MATLAB 函数	263
9.1.3 C++ Builder 调用 MATLAB 工具箱函数转换后的 DLL	264
9.1.4 C++ Builder 与 MATLAB 混合编程的另一种实现	266
9.1.5 综合实例	268
9.2 MATLAB 和 Excel 的混合编程	270
9.2.1 MATLAB 和 Excel 的混合编程介绍	270
9.2.2 通过 Excel Link 实现 Excel 和 MATLAB 的数据共享	270
9.2.3 通过 Excel 生成器	276
9.2.4 直接将 MATLAB 工作区间的数据复制到 Excel	278
9.3 MATLAB 与 Visual Basic 语言接口编程	279
9.3.1 DDE 编程	279
9.3.2 MATLAB 调用 Visual Basic 组件	281
9.3.3 Visual Basic 调用 MATLAB 组件	289
9.3.4 综合实例	290
第10章 Matcom	292
10.1 Matcom 的安装	292
10.2 编译独立的可执行程序	293
10.3 使用 Matrix<LIB>	295
10.3.1 创建一个新的 C++工程	295
10.3.2 初始化工作	296
10.3.3 函数调用	297
10.4 在 Visual C++中使用 Matcom C++矩阵库	299
10.5 使用 Matcom C++矩阵库的矩阵类 Mm	302
10.5.1 创建字符矩阵	302
10.5.2 创建数值矩阵	303
10.5.3 利用下标访问矩阵的元素	304
10.5.4 获取矩阵数据的指针	304
10.5.5 Mm 矩阵对象的初始化	305
10.5.6 Matcom C++矩阵库常量	306
10.5.7 Mm 矩阵类的几个常用函数	306
10.5.8 调用系统函数	308





10.6	Matcom 用于图形显示的常用函数	310
10.7	Matcom C++矩阵库的图形和图像显示功能.....	310
10.8	Matcom 进行图像显示的常用函数	312
10.9	Matcom 应用实例	312
第 11 章	MATLAB 与外设的数据交换及混合编程	318
11.1	串口接口	318
11.1.1	什么是 MATLAB 串口接口	318
11.1.2	什么是串口通信	318
11.1.3	串口接口标准	318
11.1.4	使用串口通信电缆连接两台设备	319
11.1.5	串口信号和针分配	320
11.1.6	串口数据格式	320
11.1.7	检测操作平台的串口信息	320
11.2	串口对象的建立	321
11.2.1	建立一个串口对象	321
11.2.2	连接到设备	323
11.2.3	配置通信参数	324
11.3	数据的读写	325
11.3.1	验证双通道示波器信息举例	325
11.3.2	控制对 MATLAB 命令行的访问	326
11.3.3	向串口设备写数据	326
11.3.4	从设备读入数据	328
11.4	事件和函数回调	331
11.4.1	回调函数应用举例	331
11.4.2	事件类型和回调属性	332
11.4.3	存储事件信息	333
11.4.4	建立和执行一个回调函数	333
11.5	记录数据到磁盘	334
11.5.1	记录数据的应用举例	335
11.5.2	创建多个记录文件	335
11.5.3	设定一个文件名	335
11.5.4	记录文件的格式	336
11.6	保存和装载数据	336
11.6.1	串口对象的读与写操作的举例	336
11.6.2	断开连接和清空工作空间	337
11.7	混合编程应用实例	338
11.7.1	数字图像处理	338
11.7.2	MATLAB 图像处理工具箱	338

11.7.3 Visual C++的图像处理位图文件读/写操作	339
11.8 实例框架	341
11.8.1 框架搭建	341
11.8.2 模块划分	343
11.8.3 应用程序功能添加	356
11.9 混合编程的实现方法	360
11.9.1 图像形态学——MATLAB 引擎数据交互实现	360
11.9.2 图像直方图统计——MATLAB 引擎命令实现	363
参考文献	368



第 1 章 MATLAB 外部接口概述



MATLAB 是当今世界上使用最为广泛的数学软件之一。它具有相当强大的数值计算、数据处理、系统分析、图形显示以及符号运算等功能，是一个完整的数学平台。在这个平台上，只需寥寥数语就可以完成十分复杂的功能，大大提高了工程分析计算的效率。另外，由于 MATLAB 的广泛使用，出现了为各个领域专门使用的工具箱(即在某一研究领域常用数学工具的函数包)，这些工具箱的出现进一步促进了 MATLAB 的流行。

1.1 MATLAB 系统简介

MATLAB 是一种高效的科学计算软件，能将强大的计算功能、可视化和程序设计整合在一个极易使用的开发环境中。在该环境下，各种问题和计算都以数学的方式来表达。MATLAB 的应用领域相当广泛，如：

- 数学和计算。
- 算法开发。
- 获取数据。
- 建模、仿真。
- 数据分析和可视化编程。
- 科学和工程作图。

MATLAB 是一个交互式的操作系统，其基本数据元素为阵列，且阵列的维数没有限制，可以解决工程上的许多计算问题，尤其是那些带有矩阵和矢量的公式，而且使用起来相当简洁。这些问题若使用 C 或 FORTRAN 语言编程来实现，需要花费较长的时间。

MATLAB 名称的本来含义是矩阵实验室(Matrix Laboratory)，其创建的最初目的就是为了使繁琐的矩阵处理和运算变得容易。最初的 MATLAB 是用 FORTRAN 语言编写的，并且采用了当时极为流行的线性代数软件包 LINPACK 和基于特征值计算的软件包 EISPACK 中大量可靠的子程序。

随着 MATLAB 的应用范围越来越广，MATLAB 的最初开发者 Morle 博士等数学家和一些软件专家成立了现在的 MathWorks 公司。如今的 MATLAB 已用 C 语言进行了全面的改写，增添了丰富多彩的图形处理功能，并且形成了一个规模庞大的工具箱(Toolbox)。

工具箱中包含了大量 MATLAB 函数(M 文件)，从而大大延伸了 MATLAB 处理特定问题的能力，如信号处理(Signal Processing)、控制系统(Control Systems)、神经网络(Neural Networks)、模糊逻辑(Fuzzy Logic)、小波分析(Wavelet Analysis)和鲁棒控制(Robust Control)等，其中包含了大量的现代工程技术学科的内容，极为实用。

MATLAB 7.1 是 MathWorks 公司推出的最新产品，适用于各种硬件平台和操作系统，功

能进一步增强，用户界面更为人性化。

1.2 MATLAB 常用的数据类型

MATLAB 所有的数据类型都可以用一种数据类型，即 MATLAB 阵列(Array)来表达。常用的 MATLAB 阵列类型有数值类型(整型、单精度型和双精度型) (numeric array)、元组类型(cell array)、结构体类型(structure array)、字符类型(char array)和逻辑型(logical array)。另外还有其他一些类型的 MATLAB 阵列，如图 1-1 所示。

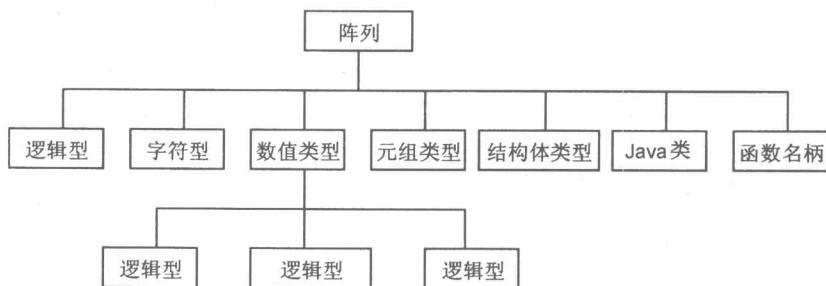


图 1-1 MATLAB 阵列数据类型

广义上说，MATLAB 只有一种数据类型，即阵列，只不过 MATLAB 阵列有不同的类型。MATLAB 阵列类似于数组的概念，一个 MATLAB 阵列可以看做是某一 MATLAB 数据类型(狭义的)的数组。举例来说，一个 $m \times n$ 的结构体阵列类似于一个 $m \times n$ 的结构体数组；一个 $m \times n$ 的双精度型阵列类似于一个 $m \times n$ 的双精度数据的数组。如果 $m=1$ ， $n=1$ 则此时的 1×1 阵列相当于 C 语言的变量。对于多维数组，与 C 语言的数组元素在内存中按行排列的排列顺序不同，所有 MATLAB 阵列在内存中都是按列排列的。

各种类型的阵列在 MATLAB 中使用之前是不用声明的，而且各种类型阵列的使用方式十分相似。但是如果进行 MATLAB 与 C/C++混合编码，MATLAB 数据类型和 C/C++数据类型的交互是需要解决的关键问题之一。因而，熟悉和掌握 MATLAB 数据类型的特点对于 MATLAB 与 C/C++混合编程的实现是非常重要的。下面对 MATLAB 几种主要阵列类型的使用进行说明。

1.2.1 数值阵列

MATLAB 的主要核心部分就是高效率的数值计算，因而数值类型的 MATLAB 阵列也是 MATLAB 开发环境中最常用的阵列类型。MATLAB 数据类型大致有整型、单精度浮点型和双精度浮点型 3 类，其中整型又有 8 位、16 位、32 位、64 位及无符号和有符号之分。由于 MATLAB 所有的运算都采用双精度浮点型，因而整型阵列如果要参加运算，必须转换为双精度型。同样，由于 MATLAB 所有的运算都采用双精度浮点型，因而，MATLAB 数值阵列中最常用和最方便的阵列类型是双精度浮点型阵列。MATLAB 数值阵列的初始值可以直接设定，也可以通过其他函数生成。例如：

```
A=[1 2 3;4 5 6;7 8 9]; %直接设定双精度数值阵列 A 的值
```

```
B=rand(3,3); %通过函数生成双精度阵列 B 的值
```

其中，“[”、“]”和“;”是 MATLAB 比较常用的符号。“[”和“]”表示 MATLAB 阵列构造的开始和结束；“;”在“[”和“]”之间表示 MATLAB 阵列一行的结束，如果“;”放在某条语句的末尾，则表示本条语句的输出不在 MATLAB Command 窗口中显示。比较下面两条语句在 MATLAB Command 窗口中执行结果的不同之处。

```
C=magic(3)
```

```
C =
```

```
8     1     6
```

```
3     5     7
```

```
4     9     2
```

```
>> C=magic(3);
```

```
>>
```

实际上，MATLAB 数值阵列与 C/C++中数组的概念类似，只不过在存储方式上，MATLAB 与 FORTRAN 相同，多维数组采用按列存储的方式，而 C/C++则采用按行存储的方式，在进行 MATLAB 与 C/C++混合编程时，这是一个重要的细节。

下面，给出一个利用 MATLAB 读取灰度位图的数据，并进行二值化的例子。通过这个实例将会熟悉 MATLAB 数值阵列的使用。

程序中要求打开一个位图文件的对话框，如图 1-2 所示，程序执行的结果，如图 1-3 所示。

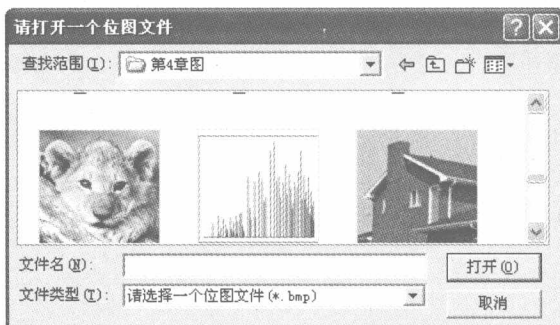


图 1-2 选择要处理的文件

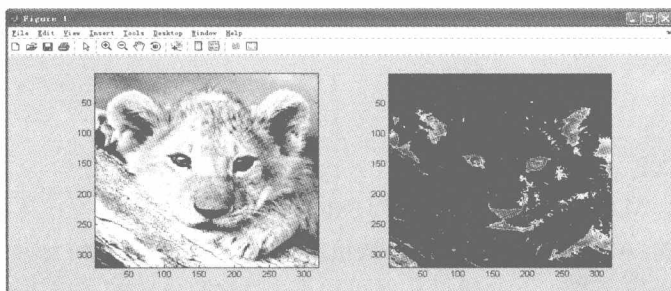


图 1-3 程序执行的结果

程序代码如下:

```

%保存为 processgrayimage.m
function []=processgrayimage()
%function []=processgrayimage()
%说明:
% MATLAB 读取的位图图像数据是 8 位无符号整型
% MATLAB 显示和存储图像时,也需要是 8 位无符号整型
% 或者将所有的数据归到[0 1]之间
% 因而采用 double 和 uint8 进行整型和双精度型之间的转换就比较方便
[name,path]=uigetfile({'*.bmp','请选择一个位图文件(*.bmp)'},'请打开一个位图文件');
file=strcat(path,name);
[l,map]=imread(file);
if size(l,3)==3
    l=rgb2gray(l);
end
%将图像数据转换为 double 型数据以方便处理
l=double(l);
l1=l-100;
signl1=sign(l1);
coefl1=(signl1+abs(signl1))/2;
%大于 125 的图像部分
l1=l.*coefl1;
%小于 125 的图像部分
l2=l.*(1-coefl1);
l1=(l1/max(max(l1)))*255;
l2=(l2/max(max(l2)))*255;
%将数据转换为 unsigned int8 型数据,以方便显示
l1=uint8(l1);
l2=uint8(l2);
figure;
h1=subplot(1,2,1);
subimage(l1);
h2=subplot(1,2,2);
subimage(l2);
truesize;

```

1.2.2 字符阵列

和 C/C++不同, MATLAB 的字符采用 UNICODE 标准,用双字节表示。MATLAB 字符阵列可转换为相应的 ASCII 码阵列,普通数值阵列也可以转换为字符型阵列。MATLAB 提供了常用的字符串操作函数,下面通过实例可以熟悉 MATLAB 的字符串操作。


```

%保存为 teststring.m 文件
function[]=teststring()
% MATLAB 字符操作函数
% mat2str 函数生成 eval 可以执行的字符串
data=rand(3,3);
sdata=mat2str(data);
eval(sdata);
%掷硬币的游戏,如果 is 是"1"则为正面,否则为反面
%字符串比较
%字符和数字之间的转换
str='1';
is=num2str(round(rand));
if strcmp(str,is)
    disp('硬币的正面');
else
    disp('硬币的反面');
end
%输出 26 个字母的 ASCII 码值
strletter='ABCDEFGHIJKLMNOPQRSTUVWXYZ';
valletter=double(strletter);
disp(valletter);
disp(valletter);
ans =
    0.0153    0.9318    0.8462
    0.7468    0.4660    0.5252
    0.4451    0.4186    0.2026
硬币的正面
Columns 1 through 23
    65    66    67    68    69    70    71    72    73    74    75    108    77    78
    79    80    81    82    83    84    85    86    87
Columns 24 through 26
    88    89    90

```

1.2.3 元组阵列

MATLAB 元组是 MATLAB 特有的数据结构。它是一种特殊的阵列。MATLAB 元组的元素可以是任意一种 MATLAB 类型的阵列。如图 1-4 所示是一个 2×3 的元组阵列，包括结构体阵列、数值型阵列、字符型阵列及元组阵列。它们都可以作为元组阵列的一个元素。

元组阵列的索引方式与数值型阵列、结构体阵列及字符型阵列不同，可以采用“{”和“}”及“(”和“)”两种索引方式。这两种方式的不同之处是通过“{”和“}”得到的是相应的元组阵列元素，而通过“(”和“)”得到的是一个包含相应元组阵列元素的 1×1 元组阵