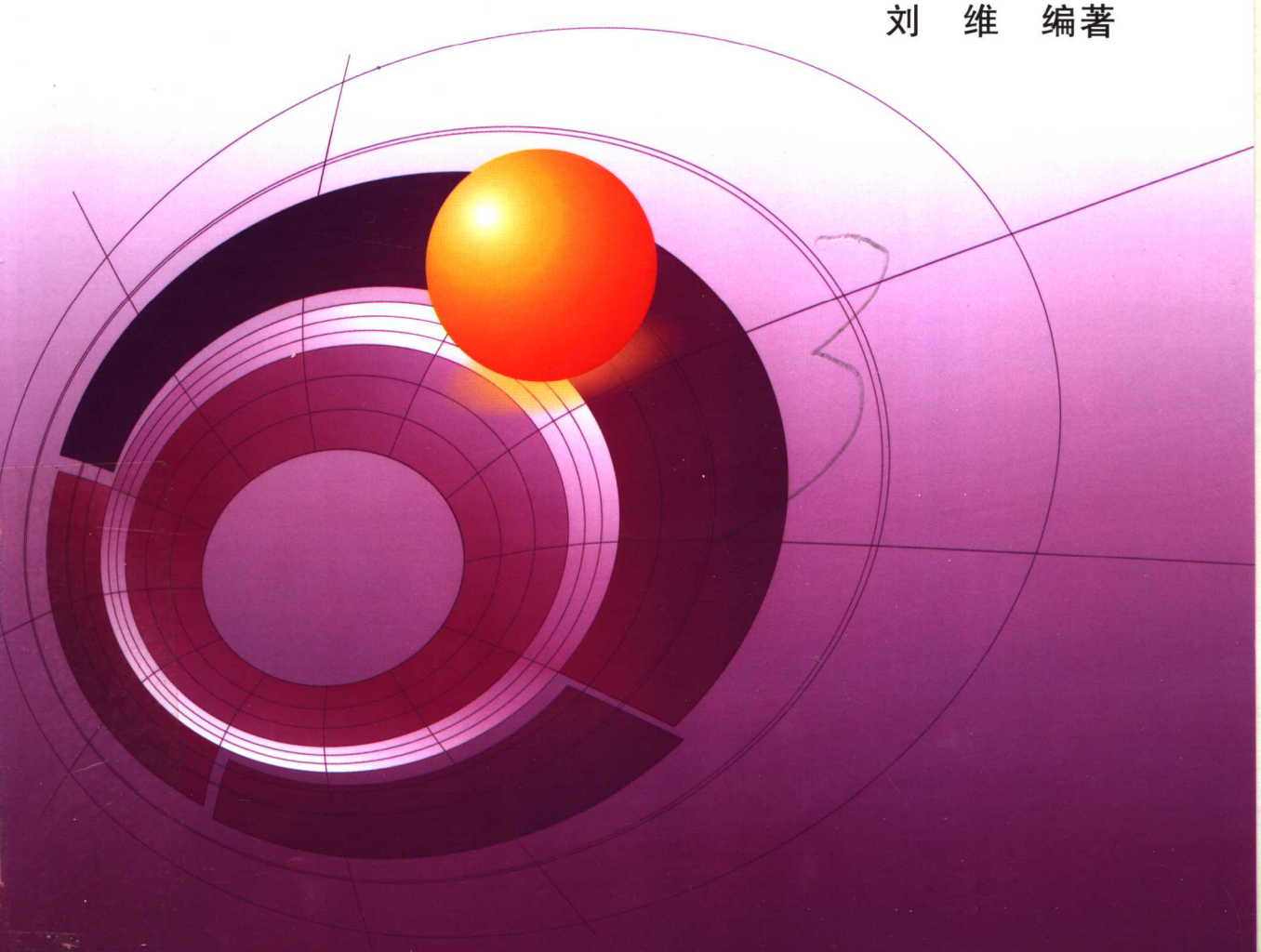
精通Matlab与C/C++混合程序设计

（第2版）

刘　维　编著



北京航空航天大学出版社

CD-ROM INCLUDED

精通 Matlab 与 C/C++ 混合程序设计

(第 2 版)

刘 维 编著

北京航空航天大学出版社

内 容 简 介

本书主要介绍如何运用 Matlab 与 C/C++进行混合程序设计。共分 8 章，主要包括：Matlab 程序设计初步、Matlab 编译器、Matlab 与 C 语言的接口、生成可独立运行的 Matlab 程序、Visual C++调用 Matlab 程序、Matlab DotNet Builder 与 Visual C ++、Matcom 与 C/C++以及 Visual C++调用 Matlab C++数学库。另外，附录中介绍有关动态链接库的基础知识。各章包含大量的实例程序，可供寻求将 Matlab 程序脱离 Matlab 环境的 Matlab 程序设计人员、寻求在 Matlab 中调用 C/C++程序的程序设计人员、寻求在 C/C++中调用 Matlab 程序的程序设计人员学习和参考。

本书附带一张光盘，其中包含各章实例程序的源代码。

图书在版编目(CIP)数据

精通 Matlab 与 C/C++混合程序设计/刘维编著. —2 版.
北京：北京航空航天大学出版社，2008.1
ISBN 978-7-81124-178-5

Ⅰ.精… Ⅱ.刘… Ⅲ.①算法语言—程序设计②C 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2007)第 172811 号

精通 Matlab 与 C/C++混合程序设计
(第 2 版)
刘 维 编著

责任编辑 胡 敏
*
北京航空航天大学出版社出版发行
北京市海淀区学院路 37 号(100083) 发行部电话：(010)82317024 传真：(010)82328026
http://www.buaapress.com.cn E-mail:bhpress@263.net
涿州市新华印刷有限公司印装 各地书店经销
*
开本：787×1 092 1/16 印张：22.75 字数：582 千字
2008 年 1 月第 1 版 2008 年 1 月第 1 次印刷 印数：5 000 册
ISBN 978-7-81124-178-5 定价：39.00 元(含光盘)

前　言

自 MathWorks 推出 Matlab 7.0 以后，Matlab 编译器在很多方面都发生了很大变化。其中最大的变化是 Matlab 编译器不再将 Matlab 程序直接编译为 C/C++代码，而只生成 Matlab 程序接口文件，Matlab 程序则直接交给 MCR(Matlab Component Runtime)来执行。新的 Matlab 编译器架构形成了新的 Matlab 与 C/C++混合程序设计特点：

- Matlab 程序在 MCR 环境下与在 Matlab 环境下执行的效率相同，因此，通过编译 Matlab 程序不会提高 Matlab 程序的效率。
- MCR 的启动时间与 Matlab 程序的启动时间相同，在混合程序设计中应考虑这一点。
- 在 C/C++程序中无论采用哪种调用方式调用 Matlab 程序，最终结果都是由 MCR 执行 Matlab 程序。
- 由于 Matlab 编译器只生成 Matlab 接口文件，而 MCR 接口一般会采用兼容设计，因此，与 Matlab 6.5 及以前版本的软件相比，用户开发 C/C++混合程序设计的复杂度降低，可继承性提高。
- 由于采用 MCR 执行 Matlab 程序而不是将 Matlab 程序编译为 C/C++程序，Matlab 函数和工具箱中可编译的部分大大增加，为用户开发混合编程提供了方便。

正是存在这些诸多不同，原有的 C/C++与 Matlab 混合程序设计的具体实现方法需要进行修正。为此，笔者对《精通 Matlab 与 C/C++混合程序设计》第 1 版中的内容进行了相应增删或修正，形成了本书的内容。

所谓“万变不离其宗”，虽然 Matlab 编译器的架构发生了很大的变化，但 Matlab与 C/C++混合程序设计还是继承了原有思路。读者在应用中可根据自己的需求选择 Matlab 调用 C/C++程序(MEX 文件)、将 Matlab 程序编译为独立可执行文件、C/C++程序调用 Matlab 引擎、C/C++程序调用 Matlab 程序编译后的动态链接库以及 C/C++程序调用 Matlab 程序编译后的 COM 组件等方式进行。

Matlab 调用 C/C++程序通过将其编译为 MEX 文件来实现，Matlab 提供了一组 C 语言 API 函数供用户调用。这组 API 函数是 Matlab 与用户 C 程序之间的桥梁。通过调用 C/C++程序编译的 MEX 文件，用户可以将 Matlab 程序中运算效率不高的代码用 C/C++来实现，从而提高计算效率。

C/C++调用 Matlab 程序用户可以选择调用 Matlab 程序编译后的动态链接

库或 Matlab 程序编译后的 COM 组件,这两种调用方法的执行效率是相同的。动态链接库方式实现起来比较简单,COM 组件方式实现起来较复杂。除此之外,C/C++可以通过 Matlab 引擎直接执行 Matlab 程序,但这种方式不能脱离 Matlab 环境执行。

另外,书中还介绍了另外两种 Matlab 与 C/C++混合程序设计的方法:Matcom C/C++数学库和 Matlab C++数学库。其中,Matcom 是第一个可以将 Matlab *.m 文件编译为 C/C++代码的编译器。Matcom 可以直接将 m 文件编译为 C/C++代码,但只支持 Matlab 5.3 版。现在一般情况下没有必要使用 Matcom编译 Matlab 程序,但 Matcom 的 C++矩阵库仍然有一定的使用价值。Matlab C++数学库是 Matlab 提供的一组封装好的矩阵运算数学库,其使用方法和 Matlab 环境中的编写方法十分类似。如果用户用 Visual C++实现用户界面,而又希望寻找一组高效的矩阵运算数学库的话,那么 Matlab C++数学库是一个不错的选择。

Matlab 与 C/C++混合程序设计方法各有千秋,具体应用还要结合开发者的具体情况进行选择。但无论使用哪种方法,Matlab 的数据结构与 C/C++的数据结构之间的相互访问和转换都是混合编程的关键,这也是本书重点所在,希望读者在阅读和开发过程中引起注意。

本书所有的源代码均可在附带的光盘中找到。第 7 章"Matcom 与 C/C++"的开发和编译环境为 Visual C++ 6.0 与 Matcom 4.5.1;第 8 章"VC++调用 Matlab C++数学库"的开发和编译环境为 Visual C++ 6.0 与 Matlab 6.5.1;其他各章的开发和编译环境为 Visual C++ 6.0与 Matlab 2007。

在本书的编写过程中有幸得到很多同志的支持和帮助,在此感谢所有为本书的完成提供过帮助的同事和朋友。感谢网络上提供 Matlab 与 C/C++混合程序设计资料的网友们,在学习 Matlab 与 C/C++混合程序设计的过程中,这些资料使我受益匪浅。感谢我的妻子齐春溪女士,在她的大力支持和协助之下此书方得以顺利编写完成。

由于作者水平有限,对于书中存在的疏漏之处,请不吝赐教,并欢迎读者与作者探讨交流,电子邮件可发送到 matlab_vc_program@yahoo.com.cn,谢谢!

作 者

2007.10.20

目　　录

第 1 章　Matlab 程序设计初步 …… 1

1.1　Matlab 程序设计特点 …… 1
1.1.1　Matlab Script 文件 …… 1
1.1.2　Matlab 表达式 …… 2
1.1.3　Matlab 函数 …… 4
1.1.4　Matlab 的向量运算 …… 6
1.1.5　Matlab 的程序控制 …… 9
1.2　Matlab 常用的数据类型 …… 12
1.2.1　数值阵列 …… 13
1.2.2　字符阵列 …… 15
1.2.3　元组阵列 …… 16
1.2.4　结构体阵列 …… 18

第 2 章　Matlab 编译器 …… 21

2.1　Matlab 编译器技术概述 …… 21
2.2　Matlab 编译器的功能 …… 22
2.3　使用 Matlab 编译器的准备工作 …… 23
2.4　mcc 编译器典型应用 …… 24
2.4.1　独立可执行文件 …… 24
2.4.2　C 动态链接库 …… 30
2.4.3　C++动态链接库 …… 32
2.4.4　C/C++动态链接库的不同之处 …… 33
2.5　进一步了解 mcc 命令 …… 34
2.5.1　mcc 常用命令选项 …… 34
2.5.2　捆绑命令文件(bundle file) …… 35
2.6　Matlab 编译器高级应用 …… 35
2.6.1　编译 script 文件 …… 35
2.6.2　Matlab 编译器关联分析失效的情况 …… 36
2.6.3　从 C/C++中调用 Matlab 内置函数(built-in function) …… 38
2.6.4　可变参数传递(varargin,varargout) …… 38
2.6.5　Matlab 环境下执行和 MCR 执行的不同之处 …… 39
2.6.6　获取 CTF 文件的目录 …… 40
2.6.7　屏幕打印和错误信息显示函数 …… 41

2.7 Deployment Tool …… 45
2.8 程序发布 …… 47

第 3 章 Matlab 与 C 语言的接口 …… 48

3.1 Matlab C/C++编译器的设置(mex) …… 48
3.2 Matlab 中调用 C 程序-MEX 文件 …… 49
3.2.1 MEX 文件介绍 …… 49
3.2.2 MEX 文件结构说明 …… 50
3.3 编译 MEX 文件 …… 51
3.4 Matlab 中 mxArray 类型的操作 …… 51
3.5 Matlab 与 C 语言混合编程常用的数据类型 …… 51
3.5.1 size_t 类型 …… 51
3.5.2 Matlab C 语言接口数据类型 …… 52
3.6 操作 Matlab 阵列 mxArray 的 mx 函数 …… 54
3.7 Matlab mex 函数 …… 77
3.8 Matlab 普通数值阵列的操作 …… 87
3.9 稀疏数组阵列(Sparse Array) …… 89
3.10 Matlab 元组 …… 92
3.11 Matlab 结构体阵列 …… 94
3.12 Matlab 字符阵列 …… 97
3.13 Matlab mat API 函数 …… 98
3.14 Matlab API 函数操作的实例 …… 105
3.14.1 更改 Matlab 数值阵列的维数 …… 105
3.14.2 分析并显示 Matlab 阵列的内容 …… 108
3.14.3 向 MAT 文件中写入 mxArray 变量 …… 118
3.14.4 从 MAT 文件中读取 mxArray 变量 …… 121
3.14.5 通讯录(结构体和 MAT 文件) …… 125
3.15 在 Visual C++中调试 MEX 文件 …… 131

第 4 章 生成可独立运行的 Matlab 程序 …… 138

4.1 直接编译 M 文件 …… 138
4.2 Matlab M 文件中调用 C 函数 …… 138
4.3 在 C 语言中调用由 Matlab *.m 文件生成的函数 …… 141
4.4 利用 Visual C++编译 M 文件并去掉控制台窗口 …… 145

第 5 章 Visual C++调用 Matlab 程序 …… 177

5.1 在 Visual C++中调用 Matlab 引擎 …… 177
5.1.1 API 函数介绍 …… 177
5.1.2 Visual C++调用 Matlab 引擎的实例 …… 178

5.2 Visual C++中调用 Matlab *.m 函数编译后的动态链接库 …… 186

第 6 章 Matlab Dotnet Builder 与 Visual C++ …… 198

6.1 COM 基础知识 …… 198
6.1.1 COM 组件概述 …… 198
6.1.2 COM 组件开发的基础知识 …… 199
6.2 DotnetBuilder 基础知识 …… 204
6.2.1 配置 Matlab C/C++编译器 …… 204
6.2.2 使用 Matlab DotnetBuilder …… 204
6.3 Visual C 调用 DotnetBuilder 生成的组件 …… 207
6.4 Matlab Dotnet Builder 与 Visual C++之间的数据转换 …… 218
6.4.1 VARIANT 数据类型 …… 218
6.4.2 SAFEARRAY 数据类型 …… 220
6.4.3 SAFEARRAY 的创建函数 …… 221
6.4.4 Matlab Dotnet Builder 与 Visual C++数据转换 …… 222
6.5 Matlab COM 工具库 …… 227
6.5.1 简　介 …… 227
6.5.2 工具库的类(utility library classes) …… 227
6.5.3 Matlab Dotnet Builder 的枚举类型 …… 233
6.5.4 安装和发布控件 …… 234
6.6 综合实例 …… 235
6.6.1 实例 1　数据转换及数组格式标志的使用 …… 235
6.6.2 实例 2　采用 MWUtil 处理 varargin 输入和 varargout 输出 …… 238
6.6.3 实例 3　MWStruct 和 MWField 操作实例 …… 241
6.6.4 实例 4　MWComplex 操作实例 …… 250
6.6.5 实例 5　MWSParse 操作实例 …… 253

第 7 章 Matcom 与 C/C++ …… 257

7.1 安装 Matcom …… 257
7.2 在 VC++中使用 Matcom C++矩阵库 …… 259
7.3 使用 Matcom C++矩阵库的矩阵类 Mm …… 264
7.3.1 创建数值矩阵 …… 264
7.3.2 创建字符矩阵 …… 265
7.3.3 利用下标访问矩阵的元素 …… 265
7.3.4 获取矩阵数据的指针 …… 266
7.3.5 Mm 矩阵对象的初始化 …… 267
7.3.6 Mm 矩阵类的几个常用函数 …… 267
7.3.7 Matcom C++矩阵库常量 …… 269
7.3.8 调用系统函数 …… 270

7.4 Matcom C++矩阵库的图形和图像显示功能 …… 271
7.5 Matcom 用于图形显示的常用函数 …… 273
7.6 Matcom 进行图像显示的常用函数 …… 273
7.7 Matcom 的应用实例 …… 274
7.7.1 实例 1 Mm 矩阵的创建及使用 …… 274
7.7.2 实例 2 图形绘制的基本功能演示 …… 278
7.7.3 实例 3 利用 Matcom 绘制动态曲线 …… 282
7.7.4 实例 4 利用 Matcom C++矩阵库进行图像显示 …… 293
7.7.5 实例 5 Matcom 二维和三维曲线绘制综合应用 …… 303

第 8 章 Visual C++调用 Matlab C++数学库 …… 316

8.1 Matlab C++ 数学库介绍 …… 316
8.2 在 Visual C++工程中调用 Matlab C++数学库 …… 316
8.3 Matlab C++数学库的使用 …… 318
8.3.1 输入和输出矩阵 …… 318
8.3.2 操作 Matlab mwArray 阵列 …… 322
8.3.3 调用系统函数 …… 341

附录 动态链接库基础知识 …… 344

A.1 为什么使用动态链接库? …… 344
A.2 C/C++语言实现动态链接库 …… 345
A.3 C/C++语言动态链接库的不同 …… 348
A.4 动态链接库的调用方式 …… 348
A.4.1 隐式链接 …… 348
A.4.2 显式链接 …… 350

参考文献 …… 353

第 1 章　Matlab 程序设计初步

1.1　Matlab 程序设计特点

Matlab 语言是一种解释型的高级语言，它包含自己的数据结构、程序流控制及文件输入输出等功能。Matlab 语句可以在 Matlab 控制窗口中直接执行，也可以采用脚本(script) *.m 文件和函数(function) *.m 文件的形式来实现。

1.1.1　Matlab Script 文件

采用 Matlab Script 文件，可以一次执行多条 Matlab 语句，这是一种比较简单的实现方式。由于 Matlab 的变量不需要定义，所以，可以很方便地按照程序的计算流程编写 Matlab Script 文件。如下面的 testscript.m 文件就是一个采用 Script 文件一次执行多条 Matlab 语句的实例。

```
%保存为 testscript.m 文件
x=-pi:0.01:pi;                          %pi 在 Matlab 中是常量，代表圆周率
y=sin(x);                               %生成绘制数据
ynoise=y+(rand(1,length(y))-0.5)*0.2;   %加均匀噪声
plot(x,y,'r.');
hold on;                                %重复绘制不擦除背景
plot(x,ynoise);
hold off;
```

上述 Script 文件的执行结果如图 1-1 所示。

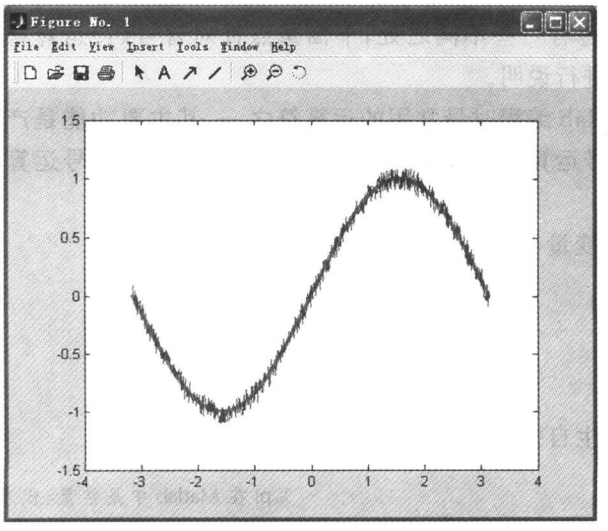


图 1-1　testscript.m 文件执行结果

1.1.2 Matlab 表达式

Matlab 的主要运算符如表 1-1 所列。

表 1-1 Matlab 的主要运算符

算术运算符		关系运算符		逻辑运算符	
+	数值阵列元素加	<	小于	&	按位与
-	数值阵列元素减	<=	小于等于	|	按位或
.*	数值阵列元素乘	>	大于	~	按位非
./	数值阵列元素右除	>=	大于等于	xor	异或
.\	数值阵列元素左除	==	相等	&&	逻辑与
:	冒号运算符	~=	不相等	||	逻辑或
.^	数值阵列元素幂				
.'	数值阵列元素转置				
'	数值阵列转置(对于实数与.'相同;对于复数,转置的同时对复数求共轭)				
*	数值阵列乘				
/	数值阵列右除				
\	数值阵列左除				
^	数值阵列幂				

Matlab 的算术运算符对于矩阵运算是非常方便的,可以大致将其分为针对数值阵列元素和针对数值阵列整体的两类数学运算符。其中针对数值阵列元素的数学运算符的运算方式可以理解为是数值阵列的单个数学元素逐个按顺序进行运算的运算符,而针对数值阵列整体的数学运算符的运算对象则是数值阵列整体(一般情况下为矩阵和向量)。Matlab 的运算符与普通高级语言的运算符有一些不同之处,下面重点针对冒号(:)运算符、点(.)运算符及左除(\)和右除(/)运算符进行说明。

":"运算符是 Matlab 编程时最常用的运算符之一,其主要功能是产生一个等间距的数据向量。恰当地使用冒号运算符可以使 Matlab 程序非常简洁。冒号运算符最常见的两个用途如下所述。

① 产生循环控制变量

```
for i=1:100
    %循环程序
end
```

② 生成数据时产生自变量

```
x=-pi:0.01:pi;                    %pi 在 Matlab 中是常量,代表圆周率
y=sin(x);
```

“.”运算符和其他运算符联合使用，则表示对阵列的元素进行操作。如 A. * B 表示 A 和 B 的相同位置元素进行相乘，A * B 则表示阵列 A 与 B 相乘(矩阵相乘或者向量相乘)。

“/”和“\”分别表示 Matlab 数值阵列右除和左除，其中 A/B 表示线性方程 AX＝B 的解，A\B 表示线性方程 XA＝B 的解；如果“/”或“\”与“.”配合使用，则表示数值阵列相同位置的元素分别进行右除或左除，其中 A.\B 表示 B(i)/A(i)，A./B 表示 A(i)/B(i)。

下面的一组 Matlab 语句则说明了 Matlab 算术运算符的使用方法。

```
%保存为 testoperat.m 文件
A=[1 2 3;4 5 6;7 8 9;];
B=[1 1 1;1 1 1;1 1 1;];

%矩阵元素相加、相减、相乘
C=A+B;
D=A-B;
E=A. * B;

%矩阵元素左除和右除
%其中 F 和 G 的结果相同
%均为：
%     1.0000    0.5000    0.3333
%     0.2500    0.2000    0.1667
%     0.1429    0.1250    0.1111
F=A.\B;
G=B./A;

%冒号运算符
N=1:100;

%幂运算符
A1=A.^2;

%转置
%Z1 为：
%     1.0000+1.0000i   4.0000+1.0000i   7.0000+1.0000i
%     2.0000+1.0000i   5.0000+1.0000i   8.0000+1.0000i
%     3.0000+1.0000i   6.0000+1.0000i   9.0000+1.0000i
%Z2 为：
%     1.0000-1.0000i   4.0000-1.0000i   7.0000-1.0000i
%     2.0000-1.0000i   5.0000-1.0000i   8.0000-1.0000i
%     3.0000-1.0000i   6.0000-1.0000i   9.0000-1.0000i
Z=A+B*i;                    %构造复数
Z1=Z.';
Z2=Z';
```

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

%计算方程 AX=B 的解
%x+y+z=1
%2x+y+2z=1
%x+2y+3z=1
X=A\B;                    %X=[0.5000 1.0000 -0.5000]'

%计算方程 XA=B' 的解
%x+2y+z=1
%x+y+2z=1
%x+2y+3z=1
X=B'/A;                   %X=[1 0 0]
```

Matlab 的关系运算符和逻辑运算符与 C 语言的使用方法一样,只是 C 语言中表示非的"!"运算符在 Matlab 中用"~"来代替。另外,Matlab 的逻辑运算符可以直接对数值阵列进行操作,例如:

```
A=[1 1 0 0 1];
B=[0 1 0 0 1];
C=A&B;                    %C=[0 1 0 0 1]
D=A|B;                    %D=[1 1 0 0 1]
E=~A;                     %E=[0 0 1 1 0]
F=xor(A,B);               %F=[1 0 0 0 0]
```

1.1.3 Matlab 函数

除了采用 Script 文件编写 Matlab 程序以外,Matlab 语言也可以通过函数的形式编写 Matlab 程序。通过 Matlab 函数编写的 Matlab 程序便于维护,而且对于使用者而言,只需要了解函数的输入和输出即可,因此,要比 Script 文件容易使用得多。

Matlab 函数的定义方法如图 1-2 所示。

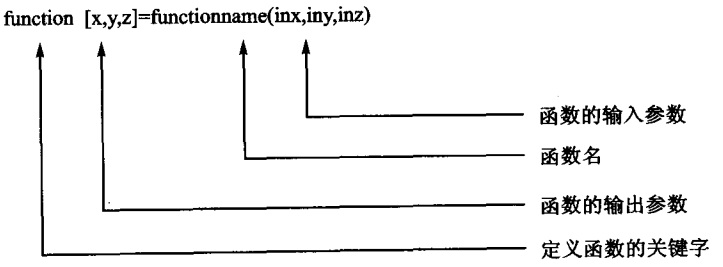


图 1-2 Matlab 函数的结构

函数体的写法与 Script 文件的写法类似，使用函数时需要注意的是：第一，保存 Matlab 函数程序时必须采用函数名作为文件名，其扩展名为. m；第二，Matlab 函数输入的外部变量如果在函数体内发生变化，则函数结束时，外部变量的值不会发生变化，类似于 C 语言输入参数的值传递。

如编写下面的函数，试图调换 x 和 y 的值，实际上这样做是不能实现的。

```
%保存为 change.m 文件
function [] = change(x,y)
t = x;
x = y;
y = t;
```

测试程序如下：

```
x = 10;
y = 1;
change(x,y);              %此时 x = 10,y = 1
```

另外，Matlab 有两个函数 nargin 和 nargout，分别返回 Matlab 函数的输入和输出参数的个数。如果采用元组变量 varargin 和 varargout 来替代常规的输入和输出参数，则函数就可以接受数目可变的输入和输出参数。下面的例子就是用来说明这两个函数的用法。

```
%保存为 testnarin_out.m 文件
function [varargout] = testnarin_out(varargin)
% function [x,y,z] = testnarin_out(m,n,k)
disp('输入参数的个数');
disp(nargin);
disp('输出参数的个数');
disp(nargout);
if(nargout>nargin)
    error('输出参数的个数不能大于输入参数的个数!\n');
end
varargout{1:nargout} = varargin{1:nargout};
```

测试程序如下：

```
m = rand(1,10);n = randn(5);z = magic(4);
testnarin_out(m,n);
x = testnarin_out(m,n);
[x,y,z] = testnarin_out(m,n);
```

则程序运行结果如下所示。

```
输入参数的个数
2
输出参数的个数
0
```

```
输入参数的个数
2
输出参数的个数
1
输入参数的个数
2
输出参数的个数
3
??? Error using==> testnarin_out
输出参数的个数不能大于输入参数的个数!\n
```

1.1.4 Matlab 的向量运算

Matlab 程序设计语言是解释型语言,其循环语句的效率非常低。但是,由于在算法上作了特殊的优化,Matlab 程序设计语言对于向量和矩阵运算的效率却很高。因而在进行 Matlab 程序设计时,一个很重要的原则就是尽量少地使用循环语句。可以做一个测试,同样是乘法,对于采用 for 循环和阵列元素相乘运算符“. * ”,两者的执行时间可以相差数十倍。

```
%保存为 testfor.m 文件
%清空 workspace 的变量
clear all;
clc;
%Matlab 循环语句与向量运算的测试语句
a=1:1000000;
b=1000000:-1:1;
sum_axb=0 ;
tic;                                %计时开始
for i=1:1000000
    sum_axb=sum_axb+a(i)*b(i);
end;
time1=toc;                          %计时结束并输出采用循环语句进行运算的时间

sum_axb=0;
tic;                                %计时开始
sum_axb=sum(a.*b);
time2=toc;                          %计时结束并输出采用向量运算消耗的时间

result1=strcat('采用循环语句运算消耗的时间为:',num2str(time1),'秒');
result2=strcat('采用向量运算消耗的时间为:    ',num2str(time2),'秒');

disp(result1);
disp(result2);
```

测试结果为：

```
采用循环语句运算消耗的时间为:0.718 秒
采用向量运算消耗的时间为:    0.031 秒
```

将 Matlab 的循环语句运算转换为向量运算需要在 Matlab 程序编写过程中加入一些技巧，其中最常见的形式如下所述。

① 用向量化运算代替 for 和 while 循环运算。如：

```
for i=-pi:0.1:pi
    y(i)=sin(i);
end
```

可以用下面的向量运算来代替。

```
x=-pi:0.1:pi;
y=sin(x);
```

另外，如果上例中还有两个向量之间的运算，可以用类似“.*”的数值阵列元素的运算函数来代替。即

```
for i=1:1000000
    sum_axb=sum_axb+a(i)*b(i);
end;
```

可以被

```
sum_axb=sum(a.*b);
```

来代替。

② 采用一些经过优化的 Matlab 向量运算函数。

经常用于向量运算的 Matlab 函数如表 1-2 所列。

表 1-2　Matlab 常用向量函数

函数名	函数功能
all	判断数值阵列是否全部非零
any	判断数值阵列是否有非零元素
reshape	变换数值阵列的各维元素数目
find	返回非零元素在阵列中的位置及其数值
sort	将数组按升序排序
sum	求数组的和
repmat	扩展阵列

除了 repmat 函数，表 1-2 中的其他函数都比较容易理解。下面就以 repmat 函数为例，说明如何采用 Matlab 的向量函数替换循环语句的技巧。repmat 函数用于扩展 Matlab 阵列，对于矩阵

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

它的扩展矩阵 B 为

$$B = \text{repmat}(A,5,3) = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \end{bmatrix}$$

下段程序运用 repmat 函数绘制如图 1-3 所示的三维曲面图。

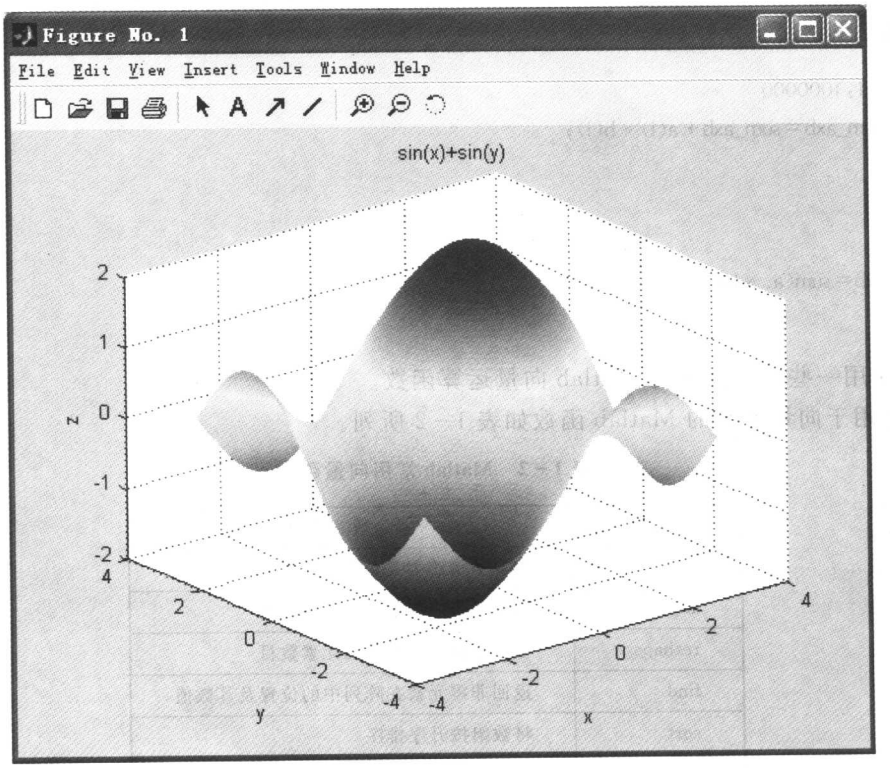


图 1-3 sin(x)+sin(y)的三维曲面图

```
%保存为 testrepmat.m 文件
%运用 repmat 函数
%绘制 sin(x)+sin(y)的三维曲面图
%首先清空 workspace 的变量
```