

精通Matlab 与C/C++ 混合程序设计

(第3版)

刘 维 编著

登录<http://www.iLoveMatlab.cn/>注册
用户名，验证密码后即可与作者在线交流

卡号: 2011001637836

密码: 



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



精通 Matlab 与 C/C++ 混合程序设计 (第 3 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 Matlab 与 C/C++ 进行混合程序设计的方法和技巧。书中以 Matlab 与 C/C++ 混合程序设计涉及的数据类型转换问题为主线,详细介绍了 Matlab 开发环境和程序设计基础、Matlab 编译器、Matlab 与 C/C++ 语言的接口、生成可独立执行的 Matlab 程序、C/C++ 通过引擎和动态链接库调用 Matlab 程序、C/C++ 通过 COM 组件调用 Matlab 程序、C/C++ 调用 Matlab C++ 数学库、C/C++ 调用 Matcom 数学库等 Matlab 与 C/C++ 混合程序设计的相关内容。本书共分 9 章,各章包含大量实战型的实例程序,可供寻求将 Matlab 程序脱离 Matlab 环境的程序设计人员、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。

本书所有的源代码均附在随书光盘中。其中:第 7 章的开发和编译环境为 Visual C++ 6.0 与 Matcom 4.5.1;第 8 章的开发和编译环境为 Visual C++ 6.0 与 Matlab 6.5.1;其他各章除特别说明之外,开发和编译环境均为 Visual C++ 2010 与 Matlab 2010。

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. -- 3
版. -- 北京:北京航空航天大学出版社,2012.1
ISBN 978-7-5124-0544-8

I. ①精… II. ①刘… III. ①算法语言—程序设计②
C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 149932 号

版权所有,侵权必究。

精通 Matlab 与 C/C++ 混合程序设计 (第 3 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(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:bhpress@263.net 邮购电话:(010)82316936

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1 092 1/16 印张:26 字数:666 千字

2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷 印数:5 000 册

ISBN 978-7-5124-0544-8 定价:55.00 元(含光盘)

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

第 3 版前言

本书的第 1 版和第 2 版出版之后,收到了大量的读者来信。从与读者交流的情况来看,Matlab 与 C/C++ 混合程序设计已经应用到了科学研究和工程开发的各个领域,这与 Matlab 的广泛应用是密切相关的。据相关报道,Matlab 语言已经成为 2010 年用户增长速度最快的计算机程序设计语言之一。作为一种以数值计算为主的脚本语言,能取得这样快速的发展实属难能可贵。

长期以来,Matlab 与 C/C++ 语言的混编架构非常不稳定,经常出现高版本不能兼容低版本的情况,这给开发者带来了很大的麻烦。值得庆幸的是,Matlab 7.0 以来,这一现象已经发生了改变。从 Matlab 7.0 至 Matlab 2010,MathWorks 保持了 Matlab 编译器基本架构的稳定,而且混编程序的向下兼容性越来越好,这对用户来说是一个福音。但用户也应当注意到,Matlab 编译器已经不再将 Matlab 程序全部编译为 C/C++ 代码,而只生成 Matlab 程序接口文件,Matlab 程序则采用 MCR(Matlab Compiler Runtime)来执行。基于 MCR 架构的混编程序设计具有以下特点。

- ◆ Matlab 程序在 MCR 环境下与在 Matlab 环境下执行的效率是相同的,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 来执行 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 及以前版本相比,用户开发 C/C++ 程序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,为用户开发混编程序提供了方便。

虽然 Matlab 混编架构的兼容性有很大的提高,但高版本与低版本之间仍然存在一些不兼容的现象。同时,Matlab 混编架构和 Visual C++ 的开发平台均发

生了较大的变化,本书第 2 版中的部分内容已经不能适应 Matlab 2010 和 Visual C++2010 开发环境的变化,部分实例在新的平台和开发环境中已不能正确运行。为此,在《精通 Matlab 与 C/C++ 混合程序设计(第 2 版)》的基础上进行了相应的增删和修正,形成本书的内容。

所谓“万变不离其宗”,Matlab 混编架构几经变化,但 Matlab 与 C/C++ 混合程序设计的思路并没有发生本质的变化。只要掌握了这些基本思路,读者就可以根据具体的需要灵活地选择不同的混编模式。

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 程序中运行效率不高的代码用 C/C++ 来实现,从而提高计算效率。

用户在 C/C++ 调用 Matlab 程序可以选择调用 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,但代码可维护性较好。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。

此外,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++ 矩阵库仍然具有一定的实用价值。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。

Matlab 与 C/C++ 混合程序设计的实践性和技巧性较强,因此书中提供了大量应用实例供读者学习。这些应用实例均是在考虑实际应用的基础上进行设计的,具有很强的实用价值。本书所有实例的源代码均附在随书光盘中,其中第 7

章的开发和编译环境为 Visual C++ 6.0 与 Matcom 4.5.1;第8章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均为 Visual C++ 2010 与 Matlab 2010。

针对本书,北京航空航天大学出版社和 Matlab 中文论坛(<http://www.iLoveMatlab.cn/>)特别提供了读者与作者在线交流的平台(<http://www.ilovematlab.cn/forum-209-1.html>),我希望借助这个平台实现与广大读者面对面的交流,解决大家在阅读此书的过程中遇到的问题,分享彼此的学习经验,从而达到共同进步。

在本书的编写过程中有幸得到很多同志的支持和帮助,在此表示衷心感谢。同时感谢网络上提供 Matlab 与 C/C++ 混合程序设计资料的网友们,在使用 Matlab 与 C/C++ 混合程序设计的过程中,这些资料使我受益匪浅。

由于作者水平有限,对于书中存在的错误之处,恳请广大读者和同行批评指正。本书勘误网址:<http://www.ilovematlab.cn/thread-140363-1-1.html>。

作者

2011 年 10 月

目 录

第 1 章 Matlab 开发环境和程序设计基础	1
1.1 本章导读	1
1.2 Matlab 环境	1
1.2.1 命令行窗口	2
1.2.2 代码编辑器	2
1.2.3 工作空间窗口	4
1.2.4 历史命令窗口	4
1.2.5 利用 Matlab 环境的界面操作	5
1.2.6 Matlab 帮助	8
1.2.7 代码输入提示	8
1.3 Matlab 语言基础	9
1.3.1 Matlab 脚本文件	9
1.3.2 Matlab 运算符与表达式	10
1.3.3 Matlab 函数	13
1.3.4 Matlab 的向量运算	15
1.3.5 Matlab 的程序控制	18
1.3.6 面向对象程序设计	22
1.4 Matlab 常用的数据类型	26
1.4.1 数值阵列	27
1.4.2 字符阵列	29
1.4.3 逻辑阵列	30
1.4.4 元组阵列	31
1.4.5 结构体阵列	33
1.4.6 函数句柄阵列	35
1.5 Matlab 常用数据显示函数	37
1.5.1 figure 窗口	37
1.5.2 绘制曲线	37
1.5.3 显示图像数据	38
1.5.4 显示三维曲面数据	39
第 2 章 Matlab 编译器	41
2.1 本章导读	41
2.2 Matlab 编译器工作原理	41

2.3	Matlab 编译器的发展历程	42
2.4	使用 Matlab 编译器的准备工作	43
2.5	mcc 编译器典型应用	44
2.5.1	独立可执行文件	44
2.5.2	C 动态链接库	53
2.5.3	C++ 动态链接库	55
2.5.4	使用 mxArray 类	57
2.5.5	mxArray 类操作详细说明	58
2.5.6	C/C++ 动态链接库的不同之处	63
2.6	mcc 命令选项、宏及捆绑命令文件	64
2.6.1	mcc 常用命令选项	64
2.6.2	mcc 宏选项	65
2.6.3	捆绑命令文件(bundle file)	66
2.7	Matlab 编译器应用中的常见问题	66
2.7.1	编译 script 文件	66
2.7.2	Matlab 编译器关联分析失效的情况	67
2.7.3	从 C/C++ 中调用 Matlab 内置函数(built-in function)	69
2.7.4	可变参数传递(varargin, varargout)	69
2.7.5	Matlab 环境下执行和 MCR 执行的不同之处	71
2.7.6	获取 ctf 文件的目录	71
2.7.7	屏幕打印和错误信息显示函数	72
2.8	Matlab 配置工具(Deployment Tool)	76
2.9	程序发布	78
2.9.1	MCRInstaller.exe	78
2.9.2	独立可执行程序发布需要附加的文件	78
2.9.3	动态链接库发布需要附加的文件	79

第 3 章 Matlab 与 C/C++ 语言的接口 80

3.1	本章导读	80
3.2	Matlab C/C++ 编译器的设置(MEX)	80
3.3	Matlab 中调用 C/C++ 程序-MEX 文件	81
3.3.1	MEX 文件用途	81
3.3.2	MEX 文件与 M 文件的关系	82
3.3.3	MEX 文件实例	82
3.3.4	MEX 文件结构说明	83
3.3.5	编译 MEX 文件	84
3.3.6	采用 C++ 创建 MEX 文件	84
3.4	Matlab 与 C 语言混编常用的数据类型	85
3.4.1	size_t 类型	85

3.4.2 Matlab C 语言接口数据类型	86
3.5 操作 Matlab 阵列 mxArray 的 mx 函数	88
3.5.1 数值阵列操作函数	88
3.5.2 字符串阵列操作函数	91
3.5.3 逻辑型阵列操作	96
3.5.4 稀疏矩阵操作函数	97
3.5.5 结构体操作函数	99
3.5.6 元组阵列操作函数	102
3.5.7 类对象阵列属性操作函数	103
3.5.8 内存操作函数	106
3.5.9 索引、维数和元素个数操作函数	110
3.5.10 Is 函数	112
3.5.11 其他操作函数	114
3.6 MEX 文件专用的 mex 函数	116
3.6.1 工作区变量操作	116
3.6.2 输出警告和错误信息	122
3.6.3 调用 Matlab 函数和执行 Matlab 语句	123
3.6.4 操作 MEX 文件	126
3.7 Matlab 阵列的操作方法与实例	129
3.7.1 Matlab 普通数值阵列	129
3.7.2 稀疏数组阵列(Sparse Array)	131
3.7.3 Matlab 字符阵列	134
3.7.4 Matlab 元组	135
3.7.5 Matlab 结构体阵列	138
3.7.6 更改 Matlab 数值阵列的维数	141
3.7.7 分析并显示 Matlab 阵列的内容	144
3.8 操作 Matlab MAT 文件的 API 函数	154
3.9 Matlab MAT 文件操作方法	161
3.9.1 向 MAT 文件中写入 mxArray 变量	161
3.9.2 从 MAT 文件中读取 mxArray 变量	164
3.9.3 同时操作 MAT 文件与 Matlab 阵列	167
3.10 在 Visual C++ 中调试 MEX 文件	174
3.10.1 利用 VC++6.0 调试 MEX 文件	174
3.10.2 利用 VC++ 2010 调试 MEX 文件	178
第 4 章 生成可独立运行的 Matlab 程序	181
4.1 本章导读	181
4.2 Matlab 编译可独立执行程序	181
4.2.1 直接编译 m 文件	181

4.2.2	Matlab M 文件中调用 C/C++ 函数	182
4.2.3	在 C/C++ 语言调用由 Matlab *.m 文件生成的函数	184
4.2.4	利用 VC++ 编译 m 文件接口函数,并去掉控制台窗口	190
第 5 章 C/C++ 通过引擎和动态链接库调用 Matlab 程序 ...		220
5.1	本章导读	220
5.2	在 VC++ 中调用 Matlab 引擎	220
5.2.1	API 函数介绍	220
5.2.2	VC 调用 Matlab 引擎的实例	221
5.3	VC++ 中调用 Matlab *.m 函数编译后的动态链接库	229
第 6 章 C/C++ 通过 COM 组件调用 Matlab 程序		241
6.1	本章导读	241
6.2	COM 基础知识	241
6.2.1	COM 组件概述	241
6.2.2	COM 组件开发的基础知识	242
6.3	Matlab 编译 COM 组件基础知识	247
6.3.1	配置 Matlab C/C++ 编译器	247
6.3.2	使用 Matlab DeploymentTool	248
6.3.3	实现 COM 组件的方法	249
6.3.4	实现 COM 组件的属性	249
6.3.5	编译 COM 组件	250
6.4	C/C++ 调用 DeploementTool 生成的组件	250
6.5	Matlab COM 组件与 C/C++ 之间的数据转换	258
6.5.1	VARIANT 数据类型	258
6.5.2	COleVariant 和 _variant_t	260
6.5.3	SAFEARRAY 数据类型	260
6.5.4	SAFEARRAY 创建函数	261
6.5.5	Matlab 数据类型的 VARIANT 类型表示	262
6.5.6	VARIANT 变量和 Matlab 阵列的转换规则	263
6.5.7	阵列格式标志(Array Formatting Flags)	265
6.5.8	数据转换标志(Data Conversion Flags)	266
6.6	Matlab COM 工具库	267
6.6.1	简介	267
6.6.2	MWUtil	267
6.6.3	MWFlags	268
6.6.4	MWStruct 类	269
6.6.5	MWField 类	271
6.6.6	MWComplex 类	272

6.6.7	MWSparse 类	272
6.6.8	MWArg 类	273
6.6.9	Matlab MWComUtil 的枚举类型	273
6.7	安装和发布 Matlab COM 组件	275
6.8	综合实例	276
6.8.1	实例 1—数据转换及数组格式标志的使用	276
6.8.2	实例 2—采用 MWUtil 处理 varargin 输入、varargout 输出	279
6.8.3	实例 3—MWStruct 和 MWField 操作实例	282
6.8.4	实例 4—MWComplex 操作实例	291
6.8.5	实例 5——MWSParse 操作实例	294
第 7 章	C/C++调用 Matcom 数学库	297
7.1	本章导读	297
7.2	Matcom 与 Matlab	297
7.3	安装 Matcom	297
7.4	在 Visual C++中使用 Matcom C++ 矩阵库	299
7.5	使用 Matcom Matrix C++矩阵库的矩阵类 Mm	304
7.5.1	创建数值矩阵	304
7.5.2	创建字符矩阵	305
7.5.3	利用下标访问矩阵的元素	305
7.5.4	获取矩阵数据的指针	306
7.5.5	Mm 矩阵对象的初始化	307
7.5.6	Mm 矩阵类的几个常用函数	307
7.5.7	Matcom C++矩阵库常量	309
7.5.8	调用系统函数	310
7.6	Matcom 矩阵库的图形和图像显示功能	311
7.7	Matcom 用于图形显示的常用函数	313
7.8	Matcom 进行图像显示的常用函数	314
7.9	Matcom 的应用实例	314
7.9.1	实例 1—Matcom Mm 矩阵的创建及使用	314
7.9.2	实例 2—图形绘制的基本功能演示	318
7.9.3	实例 3—利用 Matcom 绘制动态曲线	323
7.9.4	实例 4—Matcom 进行图像显示	335
7.9.5	实例 5—Matcom 二维和三维曲线绘制综合应用	346
第 8 章	C/C++调用 Matlab C++数学库	360
8.1	本章导读	360
8.2	Matlab C++ 数学库	360
8.3	在 Visual C++工程中调用 Matlab C++数学库	360

8.4 Matlab C++ 数学库的使用	362
8.4.1 输入/输出矩阵	362
8.4.2 操作 Matlab mxArray 阵列	367
附录 动态链接库基础知识	394
A.1 为什么使用动态链接库?	394
A.2 C/C++ 语言实现动态链接库	395
A.3 C/C++ 语言动态链接库的不同	397
A.4 动态链接库的调用方式	398
A.4.1 隐式链接	398
A.4.2 显式链接	399

第 1 章 Matlab 开发环境和程序设计基础

1.1 本章导读

作为后续章节的基础,本章介绍了 Matlab 开发环境和程序设计的基础知识,重点针对 Matlab 环境、Matlab 语言基础、Matlab 数据类型和 Matlab 数据显示函数进行了说明。

- ◆ Matlab 环境包括各种工具和 M 语言开发环境,重点介绍了命令行、代码编辑器、工作空间、历史命令、数据导入、自动生成代码、代码提示和帮助等方面的内容。
- ◆ Matlab 语言是一种基于矩阵运算的脚本语言,重点介绍了构成 Matlab 语言的基本元素、主要控制结构、运行 Matlab 语言的脚本(文件)和函数(文件),优化 Matlab 语言的向量运算和面向对象程序设计基础。
- ◆ Matlab 变量只有一种类型即阵列,但 Matlab 阵列类型非常丰富。既有较为简单的、可以存储单一类型数据的数值阵列、字符阵列和逻辑阵列,又有较为复杂、可以存储不同类型数据的结构体阵列和元组阵列,同时,还有可以存储 Matlab 函数的函数句柄阵列。掌握和熟悉 Matlab 数据类型相关知识和操作是 Matlab 与 C/C++ 语言混合程序设计的关键。
- ◆ 除数据处理之外,数据可视化是 Matlab 具备的一项重要功能。在利用 Matlab 语言进行的应用开发的过程中,数据可视化函数的应用非常普遍。因此,对常用的数据可视化函数和技术进行说明。

1.2 Matlab 环境

Matlab 是一个笼统的概念,通常所说的用 Matlab 解决某一问题,一般可以有如下两种含义。

- ◆ 通过图形化操作,利用 Matlab 环境提供的工具或应用解决某一问题。
- ◆ 通过 Matlab 语言开发或调用其他开发者编写的程序解决某一问题。

为了避免混淆,对 Matlab 环境和 Matlab 语言分别进行如下说明(后续章节均遵循这一说明)。

(1) Matlab 环境

Matlab 环境指 Matlab 软件可提供的一切非可编程的工具和应用总称。

工具和应用包括命令行窗口(Command Window)、代码编辑器(Editor)、历史命令窗口(Command History)、菜单和工具条等。随着 Matlab 的不断升级换代,Matlab 环境可提供的功能越来越丰富,越来越强大。用户不用编写代码,只通过界面操作就可以实现相当多的功能。

(2) Matlab 语言

Matlab 语言指可以在 Matlab 环境中执行,符合 Matlab 环境语法要求的计算机语言。

Matlab 语言有时简称为 M 语言,Matlab 语言可以直接输入到命令行窗口中执行,也可以保存在扩展名 .m 的文本文件中通过 Matlab 命令行窗口执行,或者由 Matlab 提供的 pcode 函

数编译为扩展名为.p的加密文件通过 Matlab 命令行窗口执行。

本节重点介绍 Matlab 环境的一些基础知识。Matlab 环境是开发、运行和调试 Matlab 代码的主要场所,Matlab 环境提供的与程序设计密切相关的工具主要包括:编辑器(Editor)、命令行窗口(Command)、工作空间(Workspace)窗口和代码分析器(Profiler)。下文将一一进行介绍。

1.2.1 命令行窗口

在介绍命令行窗口之前,首先明晰一个概念。在 Matlab 程序设计中,命令和语句的界限比较模糊。在本书中,凡是通过命令行窗口直接输入的代码统称为命令;凡是通过代码编辑器或者其他文本编辑器保存在*.m文件中的代码统称为程序语句或代码。

几乎所有的 Matlab 代码或命令均通过命令行窗口执行。通过 Matlab 命令行窗口可以执行单条命令、多条命令以及符合 Matlab 语法要求的任何程序代码。

比如:

```
>> a = 1;  
>> b = 2;  
>> c = a + b  
c =  
3  
>>
```

上述命令中,“>>”为 Matlab 命令行窗口命令输入提示符。

Matlab 命令的种类繁多,具有强大功能。在 Windows 操作系统中的 Matlab 命令行窗口中甚至可以通过“dos”命令执行 dos 指令。比如,通过 dos 命令查看当前工作机的主机名,其命令如下所示。

```
>> dos('hostname');  
THINK  
>>
```

或者通过 dos 命令执行 Windows 关机命令(下述第一条命令通知 Windows 一小时后关机,第二条命令取消执行 Windows 关机命令)。

```
>> dos('shutdown /s /f -t 3600');  
>> dos('shutdown /a');
```

1.2.2 代码编辑器

通过 edit 命令启动代码编辑器 Matlab Editor(如图 1-1 所示),在 Matlab Editor 完成代码的编辑、调试和其他操作。

下面介绍采用代码编辑器可以完成的最常用的两项功能。

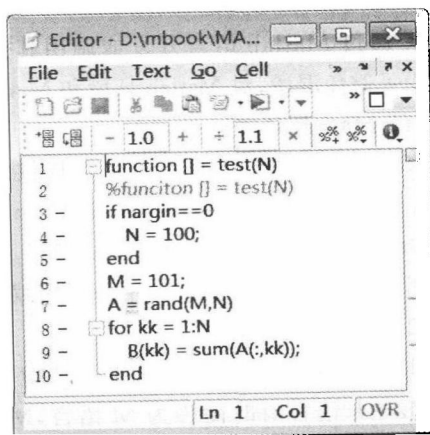


图 1-1 Matlab Editor 窗口

(1) 代码调试(主要功能查看 Debug 菜单)

通过快捷键 F12 可以在当前光标所在处插入一个断点,这样通过快捷键 F5 执行当前函数文件或脚本文件时,程序会在断点处停止,用户可以通过命令行窗口或工作空间查看 Matlab 变量的内容,完成程序调试功能。在断点处,通过 F9 快捷键继续执行后续 Matlab 代码。

(2) 代码性能分析

通过选择 Tools→Open Profiler 菜单项打开代码分析器(Profiler),在代码编辑器中输入待分析的函数或命令(这里采用 test 函数作为分析对象,见图 1-1),然后按 Enter 键执行此命令。待命令执行完毕后,Matlab 代码分析器会生成分析结果,如图 1-2 和图 1-3 所示。其中图 1-2 展示了 test 函数执行的信息概要,图 1-3 展示了 test 函数各行代码性能分析的结果。

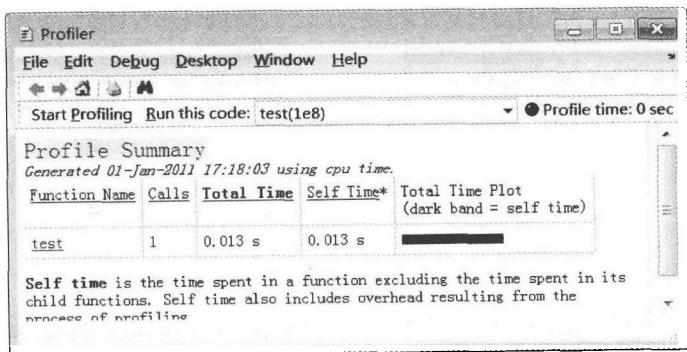


图 1-2 Matlab Profiler 窗口 1

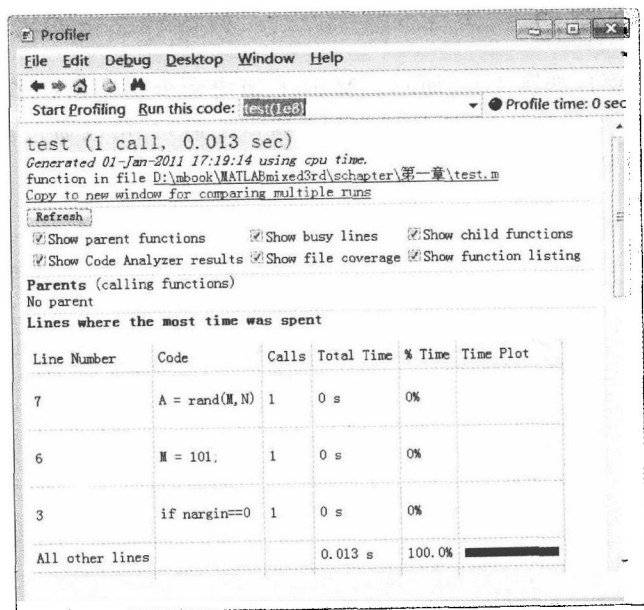


图 1-3 Matlab Profiler 窗口 2

1.2.3 工作空间窗口

Matlab 环境中创建的所有变量均由 Matlab 工作空间管理。常用的 Matlab 工作空间包括基础工作空间(Base Workspace)和函数工作空间。其中基础工作空间管理在 Matlab 命令行窗口或者脚本文件中创建的变量,而函数工作空间管理在函数中创建的变量。函数工作空间和基础工作空间相互之间不可见(global 变量例外)。在命令行窗口中通过 workspace 命令打开 Matlab 工作空间窗口。打开后的 Matlab 工作空间窗口如图 1-4 和图 1-5 所示(为了说明基础工作空间和函数工作空间,图 1-4 和图 1-5 中执行了 test 函数,所以可以同时查看 test 函数空间和基础空间)。通过 Matlab 工作空间窗口可以查看和编辑 Matlab 变量,双击变量名称可以打开变量编辑器编辑变量,右击该变量,可以通过右键快捷菜单完成多种对 Matlab 变量的操作。

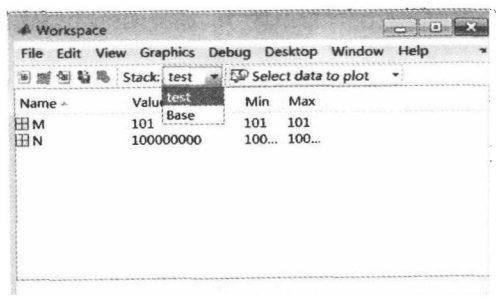


图 1-4 Matlab 工作空间窗口(test 函数空间)

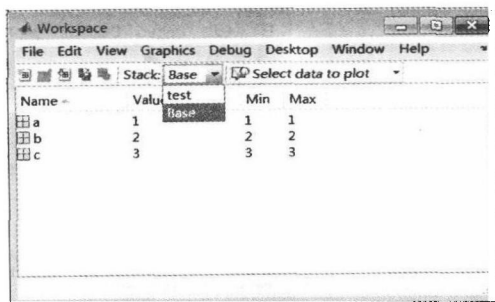


图 1-5 Matlab 工作空间窗口(Base 工作空间)

1.2.4 历史命令窗口

Matlab 历史命令窗口如图 1-6 所示。Matlab 历史命令窗口保存了在 Matlab 命令行窗口中输入的所有命令(上次清空 Matlab 历史命令窗口之后)。可以通过鼠标左键和右键快捷菜单操作 Matlab 历史命令。

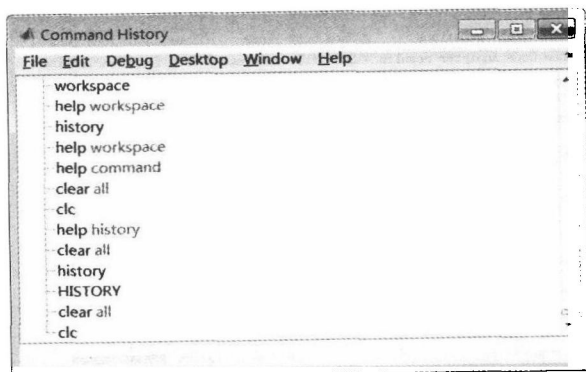


图 1-6 Matlab 历史命令窗口

① 通过鼠标左键在历史命令行窗口中单击选中需要重复执行的命令,然后将其拖放至 Matlab 命令行窗口(如图 1-7 所示)。

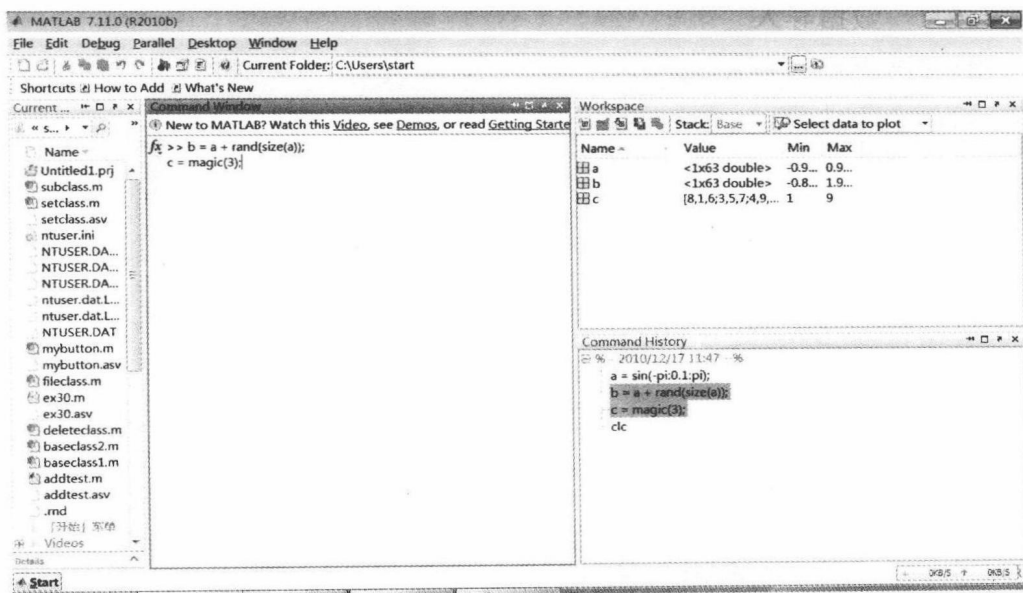


图 1-7 Matlab 历史命令窗口(直接拖放)

② 通过右键快捷菜单操作 Matlab 历史命令(如图 1-8 所示)。

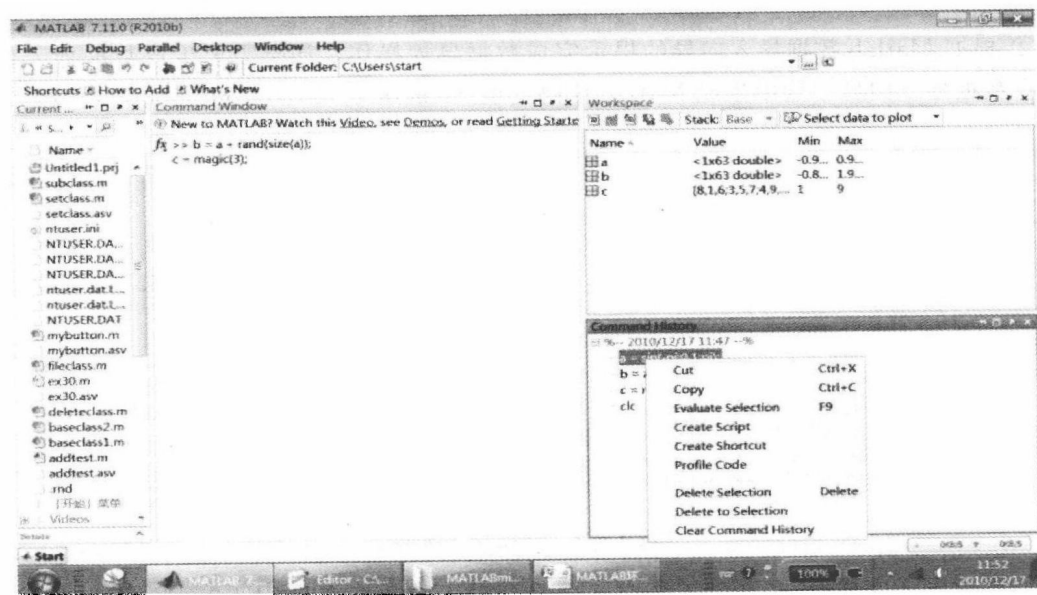


图 1-8 Matlab 历史命令窗口(通过右键快捷菜单)

1.2.5 利用 Matlab 环境的界面操作

利用 Matlab 环境的界面操作可以完成一些简单的数据处理操作,同时可以利用 Matlab 提供的工具自动生成 Matlab 代码,从而简化 Matlab 程序的编写过程。下文两个实例分别利用 Matlab 环境提供的工具生成数据导入和数据可视化的代码。