

MATLAB 外部接口应用指南

C语言

与 MATLAB 接口 —— 编程与实例

C YUYAN YU MATLAB JIEKOU BIANCHENG YU SHILI

李传军 编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书以简洁的语言、丰富的实例系统地介绍了C语言与MATLAB接口函数(称之为:C-MEX函数)的编程方法。用实例详细地介绍了MATLAB中所提供的数据类型在C-MEX函数中的编程方法。这些数据类型主要包括:双精度(double)以及非双精度(single、uint8、uint16、uint32、uint64、int8、int16、int32、int64)的数值阵列(如:标量、矢量、矩阵及三维阵列)、字符串、structures阵列、cell阵列、sparse阵列等。

同时,介绍了如何在C-MEX函数中实现调用MATLAB函数;如何用MATLAB本身所提供的BLAS库函数、LAPACK库函数实现C-MEX函数编程;以及如何用外接Intel MKL库函数实现C-MEX函数编程。并给出用C-MEX函数实现MATLAB中的svd、eig、lu、chol、inv、det等数学函数功能的实例,这些库函数的使用可以明显提高运算速度。同时,本书附录还介绍了在C-MEX函数中的错误语法示例、MEX库函数及MX库函数。

本书对C-MEX函数的介绍全面,举例详尽。不仅能够成为MATLAB与MEX混合编程的工程人员的参考书,而且可以作为C语言编程者的参考书,同时也可以作为MATLAB培训的课程教材。

图书在版编目(CIP)数据

C语言与MATLAB接口——编程与实例/李传军编著. —北京:北京邮电大学出版社,2004
ISBN 7-5635-0940-2

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

中国版本图书馆CIP数据核字(2004)第092830号

书 名: C语言与MATLAB接口——编程与实例

编 著: 李传军

责任编辑: 张莉莉

出 版 者: 北京邮电大学出版社(北京市海淀区西土城路10号) 邮编:100876

发行部电话:(010)62282185 62283578(传真)

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 18.75

字 数: 430千字

印 数: 1—3 000册

版 次: 2004年10月第1版 2004年10月第1次印刷

ISBN 7-5635-0940-2/TP·120

定价: 26.00元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

对于大多数算法仿真工作者来说,快速而方便地实现算法仿真,从而验证算法的正确性是极其重要的。Microsoft VC++ 运算速度较快,相对于 MATLAB 而言,编程较为烦琐,而 MATLAB 语言编程灵活,并且有很多工具箱,使用极为方便,但是 MATLAB 的 M 语言编写的仿真程序的仿真速度相对较慢。

C-MEX 函数是基于 C 语言编写的 C 与 MATLAB 的接口函数,是 MATLAB 应用程序接口的一个重要组成部分。通过它不但可以将现有的 C 语言编写的程序轻松地引入到 MATLAB 环境中使用,避免了重复的程序设计,而且可以使用 C 语言为 MATLAB 定制用于特定目的函数(如为了保护算法设计等),以完成在 MATLAB 中不易实现的任务。此外,更重要的一点,使用 C 语言编写的 C-MEX 函数可以有效地提高 MATLAB 环境中数据处理速度和效率。

本书的结构如下:

第 1 章 C-MEX 函数介绍。

第 2 章 编写 C-MEX 函数的步骤,编译链接生成 C-MEX 文件的步骤。

第 3 章 主要介绍了 MATLAB 中所提供的数据类型,包括:双精度(double)以及非双精度(single、uint8、uint16、uint32、uint64、int8、int16、int32、int64)的数值阵列(如:标量、矢量、矩阵及三维阵列)、字符串、structures 阵列、cell 阵列、sparse 阵列等在 C-MEX 函数中的编程方法。

第 4 章 C-MEX 函数应用举例,用 C 语言实现 MATLAB 中的多项运算功能。

第 5 章 利用 BLAS 函数编写 C-MEX 函数,实现 MATLAB 中的 sum,加法(+),乘法(*),赋值(=)等函数功能。

第 6 章 利用 LAPACK 函数编写 C-MEX 函数,实现 MATLAB 中的 svd、eig、lu、chol、inv、det 等函数功能。

第 7 章 介绍了灵活编写 C-MEX 函数的方法,重点介绍了复数的处理方法。

第8章 利用 Intel MKL 编写 C-MEX 函数,实现外接函数库的链接。

在本书的附录中还介绍了在 C-MEX 函数中的错误语法示例、MEX 库函数以及 MX 库函数。

本书内容全面,示例详细,不仅能够成为 MATLAB 与 MEX 混合编程的工程人员的参考书,而且可以作为 C 语言编程者的参考书,同时也可以作为 MATLAB 培训的课程教材。

在本书的编写过程中,得到了北京邮电大学出版社张莉莉编辑的大力支持,在这里对她表示衷心的感谢。更要感谢我的导师高攸纲教授,我的父母及夫人对我的关心和支持。

由于编者水平有限,书中错误在所难免,敬请读者不吝赐教。

作者联系方式:li_chuanjun@yahoo.com

编者

2004 年 8 月

目 录

1 C-MEX 函数介绍

1.1	MATLAB 外部接口概述	1
1.1.1	MEX 文件概念	1
1.1.2	C-MEX 文件的应用	2
1.2	mxArray 结构体	3
1.3	MATLAB 数据类型	5
1.4	库函数介绍	6
1.4.1	mx 函数库	6
1.4.2	mex 函数库	7

2 编写编译链接 C-MEX 函数的方法

2.1	MEX 指令	8
2.2	建立 MEX 环境	9
2.2.1	Lcc 编译器 MATLAB 环境配置	10
2.2.2	MS VC++ 编译器 MATLAB 环境配置	14
2.3	用 MATLAB 插件在 MSVC 环境中编写编译链接 C-MEX 函数	16
2.3.1	MATLAB 插件在 MSVC 环境中的配置	16
2.3.2	C-MEX 函数的编写编译链接过程示例	18

3 C-MEX 函数的编写方法

3.1	C-MEX 函数分析	23
3.1.1	C 与 MATLAB 存储方式的区别	23
3.1.2	C-MEX 函数的结构	24
3.1.3	编写 C-MEX 函数的基本流程	26
3.2	标量	30
3.2.1	函数介绍	31
3.2.2	双精度实数型标量	33
3.2.3	双精度复数型标量	34
3.2.4	双精度标量	36

3.3 矢量	39
3.3.1 函数介绍	39
3.3.2 双精度实数型矢量	40
3.3.3 双精度复数型矢量	42
3.3.4 双精度矢量	44
3.4 矩阵	48
3.4.1 函数介绍	49
3.4.2 双精度实数型矩阵	50
3.4.3 双精度复数型矩阵	52
3.4.4 双精度矩阵	54
3.5 三维数组	58
3.5.1 函数介绍	60
3.5.2 双精度实数型三维数组	61
3.5.3 双精度复数型三维数组	64
3.5.4 双精度三维数组	67
3.6 字符串	72
3.6.1 函数介绍	74
3.6.2 实例详解	75
3.7 非双精度数据	77
3.7.1 非双精度数据分类	77
3.7.2 函数介绍	78
3.7.3 实例详解	81
3.8 调用 MATLAB 函数	92
3.8.1 函数介绍	92
3.8.2 实例详解	93
3.9 Structures 阵列	95
3.9.1 结构体介绍	95
3.9.2 函数介绍	96
3.9.3 实例详解	100
3.10 Cell 阵列	105
3.10.1 单元阵列介绍	105
3.10.2 函数介绍	106
3.10.3 实例详解	108
3.11 Sparse 阵列	112
3.11.1 稀疏阵列介绍	112
3.11.2 函数介绍	113
3.11.3 实例详解	115

4 C-MEX 函数应用举例

4.1 复数的处理方法	119
4.1.1 复数的 fortran 结构	119
4.1.2 mat2fort 函数	120
4.1.3 fort2mat 函数	121
4.1.4 mxArray2fort 函数	122
4.1.5 fort2mxArray 函数	123
4.2 实现 MATLAB 中 sum 函数功能	124
4.2.1 双精度实数型	124
4.2.2 双精度复数型	129
4.3 实现 MATLAB 中赋值功能	133
4.3.1 双精度实数型	134
4.3.2 双精度复数型	137
4.4 实现 MATLAB 中 $z = \text{sum}(x.*y)$ 功能	141
4.4.1 双精度实数型	141
4.4.2 双精度复数型	144
4.5 实现 MATLAB 中 $z = \alpha * x + y$ 功能	148
4.5.1 双精度实数型	148
4.5.2 双精度复数型	152

5 利用 BLAS 函数编写 C-MEX 函数

5.1 BLAS 函数介绍	159
5.2 用 BLAS 实现 MATLAB 中 sum 函数功能	159
5.2.1 用 dasum 函数实现双精度实数型阵列 sum 函数功能	159
5.2.2 用 dasum 函数实现双精度复数型阵列 sum 函数功能	164
5.3 用 BLAS 实现 MATLAB 中赋值功能	168
5.3.1 用 dcopy 函数实现双精度实数型阵列赋值功能	168
5.3.2 用 zcopy 函数及复数指针实现双精度复数型阵列赋值功能	172
5.4 用 BLAS 实现 MATLAB 中 $z = \text{sum}(x.*y)$ 功能	175
5.4.1 用 ddot 函数实现双精度实数型矢量的 $z = \text{sum}(x.*y)$ 功能	176
5.4.2 用 zdotu 函数及复数指针实现双精度复数型 矢量的 $z = \text{sum}(x.*y)$ 功能	179
5.5 用 BLAS 实现 MATLAB 中 $z = \alpha * x + y$ 功能	182
5.5.1 用 daxpy 函数实现双精度实数型阵列的 $z = \alpha * x + y$ 功能	182
5.5.2 用 zaxpy 函数及复数指针实现双精度复数型 阵列的 $z = \alpha * x + y$ 功能	186

6 利用 LAPACK 函数编写 C-MEX 函数

6.1	LAPACK 函数介绍	192
6.2	用 LAPACK 实现 MATLAB 中矩阵求逆	192
6.2.1	用 dgetrf 和 dgetri 函数实现双精度实数型矩阵求逆	193
6.2.2	用 zgetrf、zgetri 函数及复数指针实现双精度复数型矩阵求逆	195
6.3	用 LAPACK 实现 MATLAB 中 LU 分解	198
6.3.1	用 dgetrf 函数实现双精度实数型矩阵 LU 分解	198
6.3.2	用 zgetrf 函数及复数指针实现双精度复数型矩阵 LU 分解	205
6.4	用 LAPACK 实现 MATLAB 中 SVD 分解	213
6.4.1	用 dgesvd 函数实现双精度实数型矩阵 SVD 分解	213
6.4.2	用 zgesvd 函数及复数指针实现双精度复数型矩阵 SVD 分解	219
6.5	用 LAPACK 实现 MATLAB 中特征分解	225
6.5.1	用 dgeev 函数实现双精度实数型矩阵特征分解	226
6.5.2	用 zgeev 函数及复数指针实现双精度复数型矩阵特征分解	231
6.6	用 LAPACK 实现 MATLAB 中 det 功能	236
6.6.1	用 dgetrf 函数实现双精度实数型矩阵 det 功能	236
6.6.2	用 zgetrf 函数及复数指针实现双精度复数型矩阵 det 功能	239
6.7	用 LAPACK 实现 MATLAB 中 chol 分解	244
6.7.1	用 dpotrf 函数实现双精度实数型矩阵 chol 分解	245
6.7.2	用 zpotrf 函数及复数指针实现双精度复数型矩阵 chol 分解	247

7 灵活编写 C-MEX 函数的方法

7.1	复数 complex 结构	251
7.1.1	双精度复数 complex 结构	251
7.1.2	mat2complex 函数	252
7.1.3	complex2mat 函数	253
7.1.4	mxArray2complex 函数	254
7.1.5	complex2mxArray 函数	255
7.2	用 zdotu 函数及复数结构指针 实现 MATLAB 中 $z = \text{sum}(x .* y)$ 功能	255
7.2.1	程序代码	256
7.2.2	分析	258
7.2.3	编译链接与测试	260
7.3	用 zaxpy 函数及复数结构指针 实现 MATLAB 中 $z = \alpha * x + y$ 功能	260
7.3.1	程序代码	261
7.3.2	分析	264

7.3.3 编译链接与测试	265
7.4 建立自己的函数库	267
8 用 INTEL MKL 编写 C-MEX 函数	
8.1 Intel MKL 介绍	269
8.2 Intel MKL 库函数的链接方法	270
8.2.1 静态链接	271
8.2.2 动态链接	271
8.3 用 Intel MKL 及复数结构指针实现 MATLAB 中 $z = \text{sum}(x. * y)$ 功能	272
8.4 用 Intel MKL 及复数结构指针实现 MATLAB 中 $z = \alpha * x + y$ 功能	275
附录	
附录 1: C-MEX 函数的语法错误示例	281
附录 2: MX 库函数	283
附录 3: MEX 库函数	286
参考文献	288

1 C-MEX 函数介绍

1.1 MATLAB 外部接口概述

MATLAB 系统是一个功能完善的程序设计和数据处理的集成环境,使用它所提供的功能、内建函数以及大量的工具箱,几乎可以完成所有的任务,并且无需借助外界的帮助,是一个完全独立的系统。MATLAB 系统更是一个友好的系统,MATLAB 提供了功能完整的应用程序接口(Application Program Interface,简称 API)来和外部程序或其他程序语言沟通,只要遵循 MATLAB API 的标准,可以实现以下几项接口工作:

1. 从 MATLAB 调用以 C 语言或 Fortran 语言编写的函数;
2. 将 MATLAB 当成一个计算引擎(Computing Engine),并从 C 或 Fortran 程序调用此 MATLAB 引擎,进行 MATLAB 的各种运算或图形显示;
3. 以 C 或 Fortran 程序读写 MATLAB 专用的 MAT 文件;
4. MATLAB 与微软程序(Excel 或 Access)之间的交互;
5. 在微软的窗口环境下,利用 DDE(Dynamic Data Exchange)来和其他应用程序沟通并传送数据;
6. MATLAB 与 Java 之间的接口。

本书将针对以上第一点(从 MATLAB 调用 C 语言编写的函数,称之为 C-MEX 函数)进行详细地介绍。

1.1.1 MEX 文件概念

MEX 文件是一种按一定格式,使用 C 语言或 FORTRAN 语言编写的,由 MATLAB 解释器自动调用并执行的动态链接函数,在不同的操作系统上,MEX 文件名的后缀是不同的,在 Microsoft Windows 系统中,MEX 文件类型的后缀名为 dll(dynamic link library)。表 1.1 对不同操作系统中文件的扩展名类型进行了总结。

表 1.1 不同系统中文件的扩展名

平 台	MEX 文件的扩展名	平 台	MEX 文件的扩展名
Alpha	.mexexp	HPUX Version 11.x	.mexhpux
HP700 Version 10.20	.mexhp7	IBM RS/6000	.mexrs6
Linux	.mexglx	SGI,SGI64	.mexsg
Solaris	.mexsol	Windows	.dll

MEX 文件的使用极为方便,与 MATLAB 的内建函数的调用方式完全相同。MATLAB 系统规定,MEX 文件的执行优先级高于 M 文件。这样,当用户执行一个 MEX 文件时,如果在当前目录下存在两个文件名相同的 MEX 文件和 M 文件,那么 MEX 文件将优先执行。

由于 MEX 文件无法显示相应的函数信息,因此,一般情况下,在构造一个 MEX 文件的同时,就应该编写一个 MATLAB 的 M 文件,作为相应的帮助文件,并且存放在同一个目录下。在该 M 文件中,不包含任何可执行语句,只包含一些帮助信息,用来对同名的 MEX 文件的功能及输入输出参数进行说明,这样,用户就可以在 MATLAB 命令窗口下使用 help 命令获取帮助信息。

因此,在同一个目录上应存在三个相同文件名,不同扩展名的三个文件。它们分别为:

C-MEX 函数: turbo_encoder.c 用以生成 C-MEX 文件

C-MEX 文件: turbo_encoder.dll 用以完成计算功能

M 文件: turbo_encoder.m 用以提供帮助信息

如果为了保护算法设计,可将 turbo_encoder.c 移走。

1.1.2 C-MEX 文件的应用

MATLAB MEX 文件是 MATLAB 系统与 C 语言的外部接口。通过它用户可以完成以下几项功能:

1. 充分利用资源

C 语言已经有着广泛的应用,有大量的 C 程序代码,如卷积编码,卷积解码,Turbo 编码,Turbo 解码等等,可将其作为计算子程序,通过 C-MEX 函数的接口程序 mexFunction 函数生成 C-MEX 文件,可以在 MATLAB 系统中如调用 MATLAB 的内建函数一样调用生成的 C-MEX 文件,从而使资源得到充分利用,避免重复的算法设计工作。

2. 解决 M 语言运算速度的瓶颈

当使用 MATLAB 进行大规模的数据处理时,MATLAB 无法避免使用 for 循环或 while 循环,由于 MATLAB 本身对于循环执行效率较低,会出现运算速度的瓶颈。通过 C 语言编写 MEX 文件,然后在 MATLAB 中调用,从而大大地提高数据处理效率。

作者以移动通信中误码率的统计作一个运算速率的分析与比较,MATLAB 中内建有误码率的统计函数 biterr.m,作者编写 MEX 文件 berc.dll。运算速度分析代码如下:

```
clc;
clear all;
a=randint(1,30000);
b=randint(1,30000);
tic;
[f2,f3]=biterr(a,b);
toc;
```

```
tic;
[e1,e2,e3]=berc(a,b);
toc;
sum1=e3-f3
sum2=e2-f2
测试结果如下:
>> berc test
elapsed_time =
    30.4940
elapsed_time =
    0.1400
sum1 =
    0
sum2 =
    0
```

从上面的运算速度来分析,由 MEX 文件 berc.dll 的运算速率是 biterr.m 的运算速度的 300 倍。对于 MATLAB 运算性能的分析,请见 help profile 命令查询 profile 的使用说明。

3. 隐藏算法设计细节,保护知识产权

C 的源代码经过编译后,已变成二进制的代码,虽然 MATLAB 能够调用它,但使用者无法看到源程序代码了。因此,采用 MEX 文件的形式,就可以隐藏程序代码,达到保护知识产权的目的。

4. 扩展 MATLAB 对硬件的编程能力

通过 MEX 文件,用户可以直接对硬件进行编程,如串行端口的数据传送与接收,设备驱动等,进一步扩展 MATLAB 的编程能力。

1.2 mxArray 结构体

C 语言与 MATLAB 之间的接口是通过一个由 C 语言编写的 mxArray 结构体,该结构定义在 <matlab 根目录>\extern\include 目录下 matrix.h 文件中。所有的有关 MEX 文件的示例及头文件均存放在 <matlab 根目录>\extern 目录下。

针对 MATLAB 中所有数据类型,如数值阵列(双精度、单精度、int8、uint8、int16、uint16、int32、uint32 等)、字符串、离散矩阵、单元阵列、结构体、对象、多维阵列、逻辑阵列、空阵列,这些 MATLAB 的数据类型,都可用一个 mxArray 结构体来定义。

其具体定义为:

```
struct mxArray_tag {
    void      * reserved;
    int       reserved1[2];
```

```

void      * reserved2;
int        number_of_dims;
unsigned int reserved3;
struct {
    unsigned int    scalar_flag:1;
    unsigned int    flag1:1;
    unsigned int    flag2:1;
    unsigned int    flag3:1;
    unsigned int    flag4:1;
    unsigned int    flag5:1;
    unsigned int    flag6:1;
    unsigned int    flag7:1;
    unsigned int    private_data_flag:1;
    unsigned int    flag8:1;
    unsigned int    flag9:1;
    unsigned int    flag10:1;
    unsigned int    flag11:4;
    unsigned int    flag12:8;
    unsigned int    flag13:8;
} flags;
unsigned int reserved4[2];
union {
    struct {
        void * pdata;
        void * pimag_data;
        void * reserved5;
        int reserved6[3];
    } number_array;
} data;
};

```

它主要包含以下几个方面的内容: 阵列的维数 `int number_of_dims`; 阵列的实部数据指针 `void * pdata`; 阵列的虚部数据指针 `void * pimag_data`; 阵列的数据类型标志 `struct {...} flags`; 阵列的维数指针。

MATLAB 接口函数库已经对 `mxArray` 结构体进行了很好的封装, 用户无需关心太多细节, 所有这些 `mxArray` 结构体的内容都可以通过 `mx` 为前缀的函数从结构体 `mxArray` 中获得。如:

```

int mxGetM(const mxArray * array_ptr);
int mxGetN(const mxArray * array_ptr);

```

```
double * mxGetPr(const mxArray * array_ptr);
```

```
double * mxGetPi(const mxArray * array_ptr);
```

等等。用户只需知道对于不同类型的阵列,如常量、矢量、矩阵、三维数组、cell 阵列、sparse 阵列等通过相应的 mx 为前缀的函数获取相应的行数、列数、实部数指针、虚部指针等 mxArray 结构体的参数就可以了,在第 3 章,作者将针对不同的数据类型进行详细地举例说明。

1.3 MATLAB 数据类型

在进行外部接口编程时,获取数据类型信息是经常需要的一种操作,在 mxArray 结构体中保存着 MATLAB 中所有数据类型信息,如双精度阵列、数值阵列(非双精度阵列)、字符串、离散阵列、单元阵列、结构体、对象、多维阵列、逻辑阵列、空阵列。这些 MATLAB 的数据类型信息。这些信息在 C 语言中是通过 mxClassID 的枚举类型变量来保存的,该枚举类型变量定义在 < matlab 根目录 > \ extern \ include 目录下 matrix.h 文件中。

该枚举类型变量定义如下:

```
typedef enum {
    mxUNKNOWN_CLASS = 0,
    mxCELL_CLASS,
    mxSTRUCT_CLASS,
    mxLOGICAL_CLASS,
    mxCHAR_CLASS,
    mxSPARSE_CLASS,      /* OBSOLETE! DO NOT USE */
    mxDOUBLE_CLASS,
    mxSINGLE_CLASS,
    mxINT8_CLASS,
    mxUINT8_CLASS,
    mxINT16_CLASS,
    mxUINT16_CLASS,
    mxINT32_CLASS,
    mxUINT32_CLASS,
    mxINT64_CLASS,      /* place holder-future enhancements */
    mxUINT64_CLASS,     /* place holder-future enhancements */
    mxFUNCTION_CLASS,
    mxOPAQUE_CLASS,
    mxOBJECT_CLASS
}
```

```
{ mxClassID;
```

它分别代表不同类型的 mxArray 阵列对象,这些定义如下:

1. mxCELL_CLASS 代表单元阵列
2. mxSTRUCT_CLASS 代表结构体
3. mxLOGICAL_CLASS 代表逻辑阵列
4. mxCHAR_CLASS 代表字符串
5. mxSPARSE_CLASS 代表离散阵列
6. mxDOUBLE_CLASS 代表双精度
7. mxSINGLE_CLASS 代表单精度
8. mxINT8_CLASS 代表 int8 类型
9. mxUINT8_CLASS 代表 uint8 类型
10. mxINT16_CLASS 代表 int16 类型
11. mxUINT16_CLASS 代表 uint16 类型
12. mxINT32_CLASS 代表 int32 类型
13. mxUINT32_CLASS 代表 uint32 类型
14. mxINT64_CLASS 代表 int64 类型 /* 备用 */
15. mxUINT64_CLASS 代表 uint64 类型 /* 备用 */
16. mxOBJECT_CLASS 代表对象类型

同时,matrix.h 文件还创建了数值类型的枚举型变量,用这个枚举型变量区分实数与复数,如下所示:

```
typedef enum {  
    mxREAL,  
    mxCOMPLEX  
} mxComplexity;
```

其中:mxREAL 代表实数,而 mxCOMPLEX 代表复数。

这些类型定义与 MATLAB 中 M 文件中数据类型是一一对应的。

1.4 库函数介绍

在 MATLAB 的外部接口函数库中,存在 4 种类型与 C 有关的库函数。它们分别为 C Engine Routines、C MAT-File Routines、C MEX-Functions、以及 C MX Functions 这 4 种类型的库函数。其中,C MEX-Functions 和 C MX Functions 分别是以 mex 和 mx 为前缀的,它们分别完成不同的功能。

1.4.1 mx 函数库

C 语言 mx 函数库是 MATLAB 外部接口函数库中提供的一系列函数,它们均以 mx 为前缀,如 mxGetM,mxGetN 等等,主要功能是为用户提供一种在 C 语言中创建、访问、

操作和删除 mxArray 结构体对象的方法,例如函数 mxGetPr,其功能就是获取 mxArray 结构体对象的实部指针。

有关 C 语言中所有的 mx-函数均在目录 <matlab 根目录> \ extern \ include 下的头文件 matrix.h 中得到声明,在以后的章节中,将结合实例对这些函数进行详细的解释和说明。

具体的函数名称请见附录 2,或参阅 apiref.pdf 文件。

1.4.2 mex 函数库

C 语言 mex 函数库是 MATLAB 外部接口函数库中提供的一系列函数,它们均以 mex 为前缀,如 mexPrintf, meCallMATLAB 等等,主要功能是与 MATLAB 环境进行交互,从 MATLAB 环境中获取必要的阵列数据,并且返回一定的信息,包括文本提示、数据阵列等。

有关 C 语言中所有的 mex-函数均在目录 <matlab 根目录> \ extern \ include 下的头文件 matrix.h 中得到声明,在以后的章节中,将结合实例对这些函数进行详细的解释和说明。

具体的函数名称请见附录 3,或参阅 apiref.pdf 文件。

2 编写编译链接 C-MEX 函数的方法

2.1 MEX 指令

使用 MEX 指令不仅能够创建 C 语言 MEX 函数,而且可以创建 Fortran 语言的 MEX 函数。具体的创建是由 mex-setup 命令来决定的。

在 MATLAB 命令窗口下键入 help mex 会得到 mex 相应指令信息。完整的 MEX 指令格式如下:

```
MEX [option1...optionN] sourcefile1 [...sourcefileN]
      [objectfile1...objectfileN] [libraryfile1...libraryfileN]
```

其中, option1...optionN 是 MEX 指令的命令行参数选项;
sourcefile1[...sourcefileN]为参与编译生成 MEX 文件的所有 C 原文件;
[objectfile1...objectfileN]为参与编译的对象文件;
[libraryfile1...libraryfileN]为参与编译的函数库文件;

在表 2.1 中对 MEX 命令的控制参数进行了简要说明。

表 2.1 MEX 命令控制参数

-argcheck	库函数的参数检查(仅用于 C 语言 MEX 文件)
-c	仅完成编译,生成对象文件
-D<name>	宏定义,相当于 C 语言“# define <name>”
-D<name> # <value>	宏定义,相当于 C 语言“# define <name> <value>”
-f <optionsfile>	提供选项文件的名字和位置
-g	建立一个包含 debug 信息的 MEX 文件
-h[elp]	列出帮助信息
-I<pathname>	将路径<pathname>包含入编译器的搜索路径
-inline	内嵌 MX 函数
-n	非执行标志
-O	优化对象代码
-outdir <dirname>	指定文件的输出路径