

# C 语言实现 MATLAB 6.5 中 M 文件的方法

宋 威

(北京工业大学 电子信息与控制工程学院, 北京 100022)

**摘要:** 本文着重描述了运用 MATLAB 命令将 M 文件翻译为 C 语言程序, 并修改为可直接调用的 C 语言函数的方法, 使用该方法所需要注意的方法局限性和 MATLAB 版本差异对方法的影响。运用该方法将能够使 C 语言能直接使用 MATLAB 当中已经完成的数学计算功能, 大大扩充了 C 语言的数学计算功能和开发效率。

**关键词:** MATLAB; M 文件; C 语言接口

众所周知, MATLAB 是一个功能强大的数学软件, 擅长于用矩阵运算完成各种数学功能。但是其程序需要在 MATLAB 环境下解释执行, 效率不高。如果能将它强大的函数库用于 C 语言, 利用 C 来编译执行, MATLAB 将能发挥更大的作用。所以, MATLAB 从 5.0 开始已经提供了与外部 C/C++ 程序的应用程序接口, 为利用 C 语言调用 MATLAB 的函数提供了可能。但是 MATLAB 的接口发展很快, 到 MATLAB 6.5 已经提供了对 VC 7.0 的支持, 同时对 C 的接口相对于 5.X 版本有了一定的改变。

在 MATLAB 当中, 我们利用 M 文件来实现函数, 每一个 M 文件实现一个单独的功能, 这一点和 C 语言当中的函数是相互对应的。所以, 如果我们能将 MATLAB 中的 M 文件转化为 C 语言下的一个单个函数, 就能实现 MATLAB 中相应的功能。

## 一、实现方法

整个过程可分为三个主要部分, 用 MATLAB 将 M 文件翻译为 C 语言文件, 从生成的 C 语言文件提取出有用语句, 编写数据转换程序实现参数格式转换。整个过程最终将把 M 文件翻译成 C 语言中的一个具有相同功能的函数, 供其它的程序调用。

本文用一个最简单的 M 文件来示例:

```
文件名: messay.m  
function c=messay()  
a=3.4;  
b=5.6;  
c=sqrt(a)+sqrt(b);
```

该 m 文件实现了计算  $c = \sqrt{a} + \sqrt{b}$

(一) 将 M 文件编译为 C 语言文件

为了将 M 文件翻译为 C 语言文件, 需要进行一定的设置, 这里假设编写 C 语言的环境为 VC6.0, 在 MATLAB 命令提示符下输入 `mex -setup` 和 `mbuild -setup` 命令, 在相应选项中选择 Microsoft Visual C/C++ 即可。

在 MATLAB 命令行中使用 `mcc` 命令将 `messay.m` 翻译为 C 代码。

```
mcc -m messay.m
```

其中的参数 `-m` 代表 `mcc` 命令将把 m 文件翻译成 C 语言的代码。

翻译命令将在 `messay.m` 所在的文件夹下生成三个 C 语言文件: `messay.h`, `messay.c` 和 `messay_main.c`。其中 `messay_main.c` 提供了 `main()` 函数; `messay.h` 提供了整个程序的函数声明; `messay.c` 包含了 MATLAB 生成的功能函数。这三个文件当中, `messay.c` 中包含了我们所需要的数学函数。

## (二) 提取有用语句

通过分析,发现由 mcc 生成的代码内部参数传送方式由 MATLAB 链接库规定,难以改动,因此需要提取有用的代码,并更改生成代码的参数传递方式。同时从生成代码的注释中可以看出,真正蕴含 M 文件功能实现的代码段都在 Mmessay()函数当中(该函数名的默认构造方式为前缀 M 加上 M 文件的文件名),而其它的生成函数仅实现参数传递和标准化接口服务的功能。

所以提取代码的具体方法是利用 messay.c 当中生成的 static mxArray \* Mmessay(int nargout\_)函数,对该函数进行修改,而其他的生成函数都可以忽略不用。原生成的 Mmessay()代码如下:

```
static mxArray * Mmessay(int nargout_) {
    mexLocalFunctionTable save_local_function_table_
        = mclSetCurrentLocalFunctionTable(&_local_function_table_messay);
    mxArray * c = NULL;
    mxArray * b = NULL;
    mxArray * a = NULL;
    mlfAssign(&a, _mxarray0_);
    mlfAssign(&b, _mxarray1_);
    mlfAssign(&c, mclPlus(mlfSqrt(mclVv(a, "a")), mlfSqrt(mclVv(b, "b"))));
    mclValidateOutput(c, 1, nargout_, "c", "messay");
    mxDestroyArray(a);
    mxDestroyArray(b);
    mclSetCurrentLocalFunctionTable(save_local_function_table_);
    return c;
}
```

在生成代码当中, mclSetCurrentLocalFunctionTable 和 mclSetCurrentLocalFunctionTable 函数为两个外部函数,将参数传给外部,与其相关的部分都对 C 程序使用数学函数没有影响。最终实际有用并执行运算的只有一句:

```
mlfAssign(&c, mclPlus(mlfSqrt(mclVv(a, "a")), mlfSqrt(mclVv(b, "b"))));
```

实际上,由 MATLAB 翻译的 C 语句中,大部分的和实际计算有关的语句和自生成的函数都以 mlf 开头,所以寻找有用语句的简单方法就是直接寻找 mlf 为前缀的代码。

## (三) 参数格式转换

应当指出, MATLAB 所有的计算都是基于一种名为 mxArray 的数据结构之上的,所有的浮点数、向量或者是矩阵在 MATLAB 当中都是通过 mxArray 结构来进行存储和传递的。当然, MATLAB 所提供的的所有数学函数也都是基于这样一种数据结构进行运算的。所以,要使用 MATLAB 的生成代码,就必须将 C 语言当中常用的浮点数和整数转换为 mxArray 结构。

本例中利用 MATLAB 函数 mxArray \* mlfScalar(double v) 和函数 double \* mxGetPr(mxArray \*) 来实现参数格式转换。函数 mlfScalar() 将 double 型变量存入一个新建的 mxArray 结构中,并返回指针,而函数 mxGetPr() 将 mxArray 结构保存的实数的实部取出。至于其它参数转换方法可参看参考文献 3 中的相关部分。

最终可以编写这样一个利用了 MATLAB 数学函数并实现  $c = \sqrt{a} + \sqrt{b}$  计算的函数:

```
double Mmessay(double ina, double inb) {
    mxArray *a,*b,*c; //三个用于 MATLAB 数学函数计算的参数
    double *outc; //计算结果变量
    a=mlfScalar((double)ina); //利用 mlfScalar()进行类型转换
    b=mlfScalar((double)inb);
    mlfAssign(&c, mclPlus(mlfSqrt(mclVv(a, "a")), mlfSqrt(mclVv(b, "b"))));
}
```

```

    out=mxGetPr(c);           //c 获得结果的实部，即结果
    mxDestroyArray(a);       //释放空间
    mxDestroyArray(b);
    mxDestroyArray(c);
    return *out;
}

```

到此，整个翻译过程完成，但是还不能直接调用。在这个函数当中运用到了 MATLAB 的数学库函数 `mlfSqrt()`、`mlcPlus()`和数据转换函数 `mlfScalar()`、`mxGetPr()`。由于这些函数是固化在链接库当中的，为了连接执行，必须加入几个库文件和几个静态链接库 `lib` 文件。所需要的库文件为 `mcc` 命令生成的 `messay.c` 文件当中所加入的库文件，一般为 `libmatlb.h`，而需要加入的静态链接库文件如下：

`libmat.lib`, `libmatlb.lib`, `libmex.lib`, `libmx.lib`

如果没有以上文件，可以用 VC 的 `lib` 命令将 MATLAB 相应的 `def` 文件转化为 `lib` 文件，转化格式为 `lib /def:filename.def /machine:ix86 /out:filename.lib`。

## 二、方法的局限

使用本文所用的方法可以将 M 文件翻译为 C 语言的函数，但是要受到两个因素的制约。

### （一）功能的实现受到 MATLAB C 函数库的限制

这种翻译的机制是由 MATLAB 提供的，`mcc` 命令能直接翻译的函数也仅局限于 MATLAB 原有的函数。因为这些函数已经被 MATLAB6.5 编译好，一般以 `mlf` 为前缀，存于动态链接库当中并可被 C 语言直接调用。这些函数在参考文献 4 中可以查到。而超出了这个范围的函数，并在 M 文件当中被嵌套使用，在用 `mcc` 进行翻译的时候，`mcc` 将在函数名前加上前缀 `mlf`，并进一步翻译该函数。

但是，这种翻译受到 MATLAB 参数传递的限制，而不能直接调用，在编译时会出现找不到相应的外部函数的错误。解决办法是手动将所有被翻译的函数进行参数传递方式的调整。如果 M 文件当中包含的函数被嵌套翻译的层数很深，这样的工作量是巨大而且不可接受的。

同时很多工具箱当中定义的函数也是不能使用这种方法进行翻译的。MATLAB 的工具箱更新速度很快，而相应的 MATLAB C 的函数库有一定滞后，导致很多最新的工具箱当中的函数是不能被翻译的。

### （二）翻译本身存在的限制

因为这种翻译是遵守 C 语言要求的，因而对于内存分配要求和 C 语言不同的函数和一些关于图形显示类型的函数（包括大量的 GUI 相关函数）也不能被正确的翻译。例如 `mash.m` 和 `step.m` 这两个较常用的 MATLAB 函数，由于上述的限制，就不能用本方法进行翻译。

对于上述的问题，可以利用在 C 程序当中运用 MATLAB 引擎的方法动态调用 MATLAB 的库函数，基本上可以解决上述所有的函数不能被正确翻译和图形显示的问题。但是，运用 MATLAB 引擎的方法需要利用 ActiveX 的自动化服务器，在运行的时候程序会在后台执行一个 MATLAB 的线程而不能完全脱离 MATLAB 的环境，也就意味着在纯 C 的环境下是不能运用的，必须要先安装 MATLAB 并能够在运行时支持多线程工作。具体的方法可以参阅参考文献 3。

## 三、MATLAB 5.X 和 MATLAB 6.5 的区别

对于本方法有以下几点区别需要注意：

1. 程序当中所需的库文件由 5.X 版本的 `matrix.h`、`mcc.h`、`matlab.h` 改为 `mex.h`、`libmatlb.h`、`libmatlbm.h` 等库文件。

2. 程序所需要加入的静态链接库文件由 5.X 需要的 `libmmfile.lib`、`libmatlb.lib`、`libmcc.lib`、`libmx.lib`

改为 libmat.lib、libmatlb.lib、libmex.lib、libmx.lib 四个文件。

3. API 函数改动很多, 虽然数学函数库即 mlf 前缀的函数少有改动, 但是关于变量建立, 内存管理和数据类型转换的函数发生改变, 即很多原 mcc 前缀的函数改为用 mx 为前缀的函数代替, 使得很多 5.X 翻译的 C 程序代码不能在 6.5 相应的库下运行通过。

4. 6.5 版本中直接增加了在 VC 环境下对 M 文件的支持。在执行 mbuild -setup 的配置命令后, MATLAB 在 VC 中提供了 MATLAB Project Wizard, 可在 VC 环境下直接建立 MATLAB 的工程来翻译 M 文件。但是这种翻译方法在遇到未定义函数嵌套时将错误的把函数名翻译为变量名, 而 mcc 命令将进一步翻译内部嵌套的函数。

#### 参考文献

- [1] 刘志俭等. Matlab 应用程序接口用户指南[M]. 北京: 科学出版社, 2000
- [2] 石波, 陈淑珍, 沈海鸥. VC 与 MATLAB 接口剖析[J]. 计算机工程, 2000, 26(3): 98-100
- [3] *MATLAB External Interfaces Reference*. Revised for MATLAB 6.5.1 Online only [M]. MathWorks, Inc., September 2003
- [4] *MATLAB C Math Library Reference*. Revised for Version 2.2 (Release 12.1) Online only [M]. MathWorks, Inc., June 2001

## The Method of Using M-files of MATLAB6.5 in C Language

SONG Wei

(School of Electronic Information & Control Engineering,  
Beijing University of Technology, Beijing 100022, China )

**Abstract:** This paper provided a specific method to translate the M-files of MATLAB to C codes, pointed out the limitations of such method, and compared the differences between Matlab6.5 and the previous versions in realizing it. By using this Method, the C language is able to directly call functions that are well defined in MATLAB, and the efficiency of code is improved.

**Key words:** Matlab; M-file; C language