**1. MEX的编写格式**

写MEX程序其实就是写一个DLL程序，所以你可以使用C，C++，Fortran等多种编程语言来写。

编写MEX程序的编辑器可以使用MATLAB的代码编辑器，也可使用自己的C++编辑器，如VS2008等。

用MATLAB的编辑器的好处是，MEX函数会加粗高亮显示，这给程序编写带来便利，可惜无法动态调试。如用VC即可编译也可调试，比较方便。mex的编译结果实际上就是一个带输出函数mexFunction 的dll文件，所以会用VC编写和调试dll，就会用VC编写和调试MEX程序。

**a. MEX文件格式**

#include "mex.h"

void mexFunction( int nlhs, mxArray \*plhs[],  
                  int nrhs, const mxArray \*prhs[] )  
{

}

四个参数分别用来输出和输入数据: **nlhs 输出参数个数，plhs 输出参数指针** (nrhs和prhs是输入参数相关的)。

注意: 我们对**输出和输入参数的操作都是通过指针的方式进行的**。(这点很容易理解，因为我们的计算结果是需要传递给MATLAB的，实际上我们传递的不是数据，而是指针。MATLAB可以通过这些指针，访问内存中的数据。)

**b. 操作输入数据**

**对输入数据进行操作，需要通过MEX函数mxGetPr 得到数据的指针地址。 mxGetM 和 mxGetN 得到矩阵数据的行和列 (返回整数)。对于实矩阵，我们可以定义 double \*M; 来对实矩阵数据操作。如:**

double \*M;

int m,n;

// 指针指向第一个参数的数据地址

M = mxGetPr(prhs[0]);

m = mxGetM(prhs[0]);

n = mxGetN(prhs[0]);

需要注意的是，**MATLAB矩阵数据的存储顺序是"从上到下，从左到右"**的，这点和Fortran是一样的。也就是说对于MATLAB的m x n的矩阵A。 A(1,1) 就是 \*M，A(2,1) 就是 \*(M+1) ，以此类推，A(i,j) 就是 \*(M + m\*(j-1) + (i-1)).

注意: MATLAB的指标从1开始，C的指标从0开始。

**c. 操作输出数据**

对于输出数据，我们需要首先分配内存空间，有专门的mex函数可以使用，如:

plhs[0] = mxCreateDoubleMatrix(m,n, mxREAL); //生成m x n 的实矩阵。

同输入数据一样，要对输出数据操作，我们也需要一个**指向数据的指针变量**，如

double \*A;

A = mxGetPr(plhs[0]);

下面介绍一下如何使用VS2008编写MEX并编译调试。

**2. VC中编写MEX**

打开VS2008, 新建项目, 选择MFC DLL.

**a. 配置项目属性**

打开项目属性配置页，C++ -> 附加包含目录 加入MATLAB安装目录下的 \extern\include 路径。

连接器 -> 附加库目录 加入MATLAB的 \extern\lib\win32\microsoft 路径。

连接器 -> 输入 -> 附加依赖项 输入libmx.lib libeng.lib libmat.lib libmex.lib 这四个lib文件。

**b. 编辑输出函数**

在项目源文件的. def 中EXPORTS段加入 mexFunction， 如:  
EXPORTS  
    ; Explicit exports can go here  
部分改为  
EXPORTS  
    ; Explicit exports can go here  
    mexFunction  
**c. 编写MEX文件**

项目文件中新建一个C++文件 如 mexproc.cpp，里面按前面介绍的格式编写代码即可。

**d. VC编译MEX**

像编译其他程序那样直接编译即可，成功会生成dll文件。如果编译链接时出错，根据错误提示，检查一下lib和h的路径是否正确，有无缺少lib文件，代码是否有语法错误等。

**3. VC中调试MEX**

要调试MEX程序就要先编译，再调用她。所以我们需要在MATLAB中调用这个函数，并在VC的MEX程序相应位置处下断点即可。调用的函数名就是dll的主文件名，你可以根据自己的需要改名。我们用mymexfun.dll为例，先在VC的 mexFunction 函数代码段开始处F9下断。然后Ctrl+Alt+P附加MATLAB.exe进程。这样就可以运行命令调试程序了。我们可以在MATLAB的命令行里输入命令:

          [输出变量] = mymexfun(输入变量)

(如果命令找不到，检查一下matlab当前路径，和path路径。)

程序一旦被调用，就会被断在我们的断点处。接着你就可以像调试C++程序那样调试MEX程序了。

**Compiling MEX-Files with the Microsoft Visual C++ IDE**

|  |
| --- |
| **Note**   This section provides information on how to compile source MEX-files in the Microsoft Visual C++ IDE. It is not totally inclusive. This section assumes that you know how to use the IDE. If you need more information on using the Microsoft Visual C++ IDE, refer to the corresponding Microsoft documentation. |

To build MEX-files with the Microsoft Visual C++ integrated development environment:

1. Create a project and insert your MEX source files.
2. Create a .def file to export the MEX entry point. On the **Project** menu, click **Add New Item** and select **Module-Definition File (.def)**. For example:

LIBRARY MYFILE EXPORTS mexFunction <-- for a C MEX-file or EXPORTS \_MEXFUNCTION -- for a Fortran MEX-filepre>

1. On the **Project** menu, click **Properties** for the project to open the property pages.
2. Under C/C++ General properties, add the MATLAB include folder, matlab\extern\include, as an additional include folder.
3. Under C/C++ Preprocessor properties, add MATLAB\_MEX\_FILE as a preprocessor definition.
4. Under Linker General properties, change the output file extension to .mexw32 if you are building for a 32–bit platform or .mexw64 if you are building for a 64–bit platform.
5. Locate the .lib files for the compiler you are using under *matlabroot*\extern\lib\win32\microsoft or *matlabroot*\extern\lib\win64\microsoft. Under Linker Input properties, add libmx.lib, libmex.lib, and libmat.lib as additional dependencies.
6. Under **Linker Input properties**, add the module definition (.def) file you created.
7. Under **Linker Debugging properties**, if you intend to debug the MEX-file using the IDE, specify that the build should generate debugging information. For more information about debugging, see [Debugging on the Microsoft Windows Platforms](http://www.mathworks.com/help/techdoc/matlab_external/f32489.html#f18756).

If you are using a compiler other than the Microsoft Visual C++ compiler, the process for building MEX files is like that described above. In step 4, locate the .lib files for the compiler you are using in a folder of *matlabroot*\extern\lib\win32 or *matlabroot*\extern\lib\win64. For example, if you are using an Open Watcom C/C++ compiler, look in *matlabroot*\extern\lib\win32\watcom.

<http://www.mathworks.com/help/techdoc/matlab_external/f24338.html>

4.Project->Settings  
4.1 General下将Microsoft Foundation Classes改为Use MFC in a Static Library  
4.2 Debug 下Executable for debug session中加入C:\matlab\bin\win32\MATLAB.exe  
4.3 C/C++ 下Category:Preprocessor下Preprocessor definitions 中加入MATLAB\_MEX\_FILE  
                      :Precomplied Headers下选Not Using precomplied headers  
4.4 Link下Category：Debug下我选了Generate mapfile  
                    ： Input下Object/library modules：libmx.lib libmex.lib libmat.lib libeng.lib  
                    ：General下将Debug/mymex.dll改为Debug/mymex.mexw32

--  
***Anony***