Matlab与C混合编程API之mxCreateNumericMatrix

zcl.space

目录

1 引言

在Matlab与C混合编程API之mxCreateDoubleMatrix 中，我们使用 mxCreateDoubleMatrix 创建二维双精度矩阵。本文要介绍的 mxCreateNumericMatrix 比 mxCreateDoubleMatrix 用途更广， mxCreateNumericMatrix 也用于创建二维矩阵，但是其可以指定矩阵元素的类型，包括整型和浮点类型。

2 使用语法

```
#include "mex.h"
mxArray *mxCreateNumericMatrix(mwSize m, mwSize n,
  mxClassID classid, mxComplexity ComplexFlag);
```

输入参数 m n 和 ComplexFlag 就不做过多的介绍，请见Matlab与C混合编程API之mxCreateDoubleM一文。这里着重介绍一下 mxClassID 。这个参数表示了矩阵中元素的类型，matlab根据这个类型解释内存中二进制比特的值。比如在C中设置这个值为 mxINT16_CLASS 表示矩阵元素都是16位整型。桥梁函数中的类型与matlab中类型对照表如下：

表 1: matlab和C类型对照表

matlab类型	桥梁函数中对应类型
int8	mxINT8_CLASS
uint8	mxUINT8_CLASS
int16	mxINT16_CLASS
uint16	mxUINT16_CLASS
int32	mxINT32_CLASS
uint32	mxUINT32_CLASS
int64	mxINT64_CLASS
uint64	mxUINT64_CLASS
single	mxSINGLE_CLASS
double	mxDOUBLE_CLASS

3 一个例子

在学习新东西的过程中，我比较喜欢例子。在和别人交流新概念的时候，我也比较喜欢使用例子。接下来，用一个小例子来阐述 mxCreateNumericMatrix 的使用。这个例子使用C创建一个矩阵被matlab使用。代码如下：

```
1 #include "mex.h"
2
3 /* The mxArray in this example is 2x2 */
4 #define ROWS 2
5 #define COLUMNS 2
6 #define ELEMENTS 4
```

```
7
8  void mexFunction(int nlhs, mxArray *plhs[], int nrhs,
9                   const mxArray *prhs[])
10     /* pointer to real data in new array */
11     double  *pointer;
12     mwSize index;
13     /* existing data */
14     const double data[] = {2.1, 3.4, 2.3, 2.45};
15
16     /* Check for proper number of arguments. */
17     if ( nrhs != 0 ) {
18         mexErrMsgIdAndTxt("MATLAB:arrayFillGetPr:rhs",
19             "This function takes no input arguments.");
20     }
21
22     /* Create an m-by-n mxArray; you will copy
23        existing data into it */
24     plhs[0] = mxCreateNumericMatrix(ROWS, COLUMNS,
25                               mxDOUBLE_CLASS, mxREAL);
26     pointer = mxGetPr(plhs[0]);
27
28     /* Copy data into the mxArray */
29     for ( index = 0; index < ELEMENTS; index++ ) {
30         pointer[index] = data[index];
31     }
32     return;
33 }
```

由于功能简单，在桥梁函数中就没有重新调用函数。在这个例子中生成一个 2×2的矩阵，其中的元素为：[2.1 2.3;3.4 2.45]，注意这里的元素顺序，在matlab中元素的位置是按列存放的。所以matlab中的矩阵[2.1 2.3;3.4 2.45]，表示成一维数组就是[2.1 3.4 2.3 2.45]。

代码的第24行调用了 `mxCreateNumericMatrix` ,并指定了 `mxClassID` 为 `mxDOUBLE_CLASS` 。注意：代码的第34行的 `pointer` 不能用 `plhs[0]` 代替，即不能写成：

```
plhs[0][index] = data[index]
```

这是因为，在桥梁函数中 `plhs[0]` 是被matlab代码使用的地址，其指向 `mxArray` 类型变量。而 `pointer` 是被C使用的地址，其指向 `double` 类型变量。

4 创建矩阵的其它几个API

本想把matlab里创建数组的API一个一片博文写出来，后来发现这些API大同小异。如果我还坚持初衷，未免显得累赘，有凑数之嫌（我是那种靠数量取胜的人么？）。

4.1 mxCreateUninitNumericMatrix

与 `mxCreateNumericMatrix` 相比，这个API的唯一区别是创建的矩阵没有初始化，matlab你

能告诉我为什么还要定义这样一个API么？我怎么发现matlab有凑数之嫌呢？用一个能说服我的理由拍醒我吧！

4.2 mxCreateNumericArray

与 `mxCreateNumericMatrix` 相比，这个API的区别在于创建的矩阵不限于二维，可以是N维，所以其调用语法略有不同，如下：

```
#include "mex.h"
mxArray *mxCreateNumericArray(mwSize ndim, const mwSize *dims,
         mxClassID classid, mxComplexity ComplexFlag);
```

其中 `ndim` 指定了要创建的矩阵维度， `dims` 是指向表示维度的数组的指针, `dim[0]` 表示第一维的大小，依次类推。该API创建的N维矩阵所有元素都被初始化为0.

4.3 mxCreateUninitNumericArray

从名字上就可以看出来和 `mxCreateNumericArray` 的区别。不说了，matlab你定义这个API就是在耍流氓。

5 尾声

本文介绍了 `mxCreateNumericMatrix` API的语法和使用过程，并指出 `plhs[0]` 的一个使用限制。