MATLAB与C语言之间的三维数组传递

@March 13, 2024

@Jun

发现问题
代码demo
    matrix_3demo.c
    matrix_3d_mo.m
使用方法

发现问题

MATLAB与C语言在三维数组方面储存的方式不同，在matlab中，三维数组的各维度为(行， 列，深)，而在C语言中则是(深， 行， 列)，导致矩阵在二者之间的传递会出现问题，矩阵元素被打乱，搜遍全网都没有人发出相关解决代码，所以自己写了一个demo。

代码demo

matrix_3demo.c

```
#include "mex.h"

void matrix_3d_demo(double *d, mwSize rows, mwSize cols, mwSize
    int i, j, k;

    printf("matlab的三维数组d以一维的形式传入c中\n\n");

    //错误的形式
    printf("当把d的深度看作第 三 维度, 每一页按照c以列储存的下标输出为(错
    printf("d in C：\n");
    // 计算三维数组
```

```
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            for (k = 0; k < pages; k++) {
                printf("depth = %d\n", k);
                printf("%f ", d[i + j * rows + k * rows * cols]
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n");

    //正确的形式
    printf("当把d的深度看作第 一 维度, 每一页按照c以列储存的下标输出为(正
    printf("d in C：\n");
    for (k = 0; k < pages; k++){
        printf("depth = %d\n", k);
        for (i = 0; i < rows; i++){
            for (j = 0; j < cols; j++){
                printf("%f ", d[i + j * rows + k * rows * cols]
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n\n");

    printf("-------------在C中创建的矩阵, 在C中输出-------------\n"

    printf("在c中创建指针形式的以一维线性储存的三维数组d_1d:\n");
    double *d_1d = (double *)calloc(pages*rows*cols, sizeof(doub
    for (i = 0; i < pages*rows*cols; i++) {
            d_1d[i] = i;
    }
    printf("d_3d:\n");
    for (k = 0; k < pages; k++) {
```

```
        printf("depth = %d\n", k);
        for (i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++){
                printf("%f ", d_1d[k * rows * cols + i * cols +
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n");

    //正确的形式
    printf("在c中d_1d赋值给三维数组d_3d(正确的形式): \n");
    double ***d_3d = (double ***)calloc(pages, sizeof(double **)
    for (k = 0; k < pages; k++) {
        d_3d[k] = (double **)calloc(rows, sizeof(double *));
        for (i = 0; i < rows; i++) {
            d_3d[k][i] = (double *)calloc(cols, sizeof(double));
        }
    }
    printf("d_3d:\n");
    for (k = 0; k < pages; k++) {
        printf("depth = %d\n", k);
        for (i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++){
                d_3d[k][i][j] = d_1d[k * rows * cols + i * cols
                printf("%f ", d_3d[k][i][j]);
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n");


    //正确的形式
```

```
    printf("把从matlab中传来的一维形式的三维数组d, 赋值给c中创建的三维数
    printf("d_3d:\n");
    for (k = 0; k < pages; k++) {
        printf("depth = %d\n", k);
        for (i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++){
                d_3d[k][i][j] = d[i + j * rows + k * rows * cols
                printf("%f ", d_3d[k][i][j]);
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n");

    //正确的形式
    printf("c中创建的三维数组d_3d赋值给要输出到matlab中的output_dd(一维
    printf("output_dd在C中的输出为: \n");
    for (k = 0; k < pages; k++) {
        printf("depth = %d\n", k);
        for (i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++){
                output_dd[i + j * rows + k * rows * cols] = d_3d
                printf("%f ", output_dd[i + j * rows + k * rows
            }
            printf("\n");
        }
        printf("\n");
    }
    printf("\n");



}

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxAr
```

```
    // 获取输入参数
    double *d = mxGetPr(prhs[0]);
    mwSize rows = mxGetDimensions(prhs[0])[0];
    mwSize cols = mxGetDimensions(prhs[0])[1];
    mwSize pages = mxGetDimensions(prhs[0])[2];
    printf("rows = %d, cols = %d, pages = %d\n", rows, cols, pag

    // 分配输出参数内存
    plhs[0] = mxCreateNumericArray(3, (const mwSize[]){rows, col

    // 获取输出参数指针
    double *output_dd = mxGetPr(plhs[0]);

    // 调用函数计算结果
    matrix_3d_demo(d, rows, cols, pages, output_dd);
}
```

matrix_3d_mo.m

```
for i=1:3
    for j=1:4
        for k=1:2
            d(i,j,k)= (k-1) * 3 * 4 + (i-1) * 4 + (j-1);
        end
    end
end

disp('d在matlab中的输出为:');
disp(d);



output_dd = matrix_3d_demo(d);
```

```
disp('output_dd在matlab中的输出为(正确的形式):');
disp(output_dd);

clearvars
```

使用方法

1. matlab命令行执行

   ```
   mex matrix_3d_demo.c
   ```

2. c程序编译成功后，matlab命令行执行

   ```
   matrix_3d_demo
   ```

3. matlab命令行显示：

   ```
   d在matlab中的输出为:

   (:,:,1) =

        0     1     2     3
        4     5     6     7
        8     9    10    11


   (:,:,2) =

       12    13    14    15
       16    17    18    19
       20    21    22    23
   ```

```
rows = 3, cols = 4, pages = 2
matlab的三维数组d以一维的形式传入c中

当把d的深度看作第 三 维度, 每一页按照c以列储存的下标输出为(错误的形式):
d in C：
depth = 0
0.000000 depth = 1
12.000000
depth = 0
1.000000 depth = 1
13.000000
depth = 0
2.000000 depth = 1
14.000000
depth = 0
3.000000 depth = 1
15.000000

depth = 0
4.000000 depth = 1
16.000000
depth = 0
5.000000 depth = 1
17.000000
depth = 0
6.000000 depth = 1
18.000000
depth = 0
7.000000 depth = 1
19.000000

depth = 0
8.000000 depth = 1
20.000000
depth = 0
9.000000 depth = 1
```

```
21.000000
depth = 0
10.000000 depth = 1
22.000000
depth = 0
11.000000 depth = 1
23.000000


当把d的深度看作第 一 维度, 每一页按照c以列储存的下标输出为(正确的形式):
d in C：
depth = 0
0.000000 1.000000 2.000000 3.000000
4.000000 5.000000 6.000000 7.000000
8.000000 9.000000 10.000000 11.000000

depth = 1
12.000000 13.000000 14.000000 15.000000
16.000000 17.000000 18.000000 19.000000
20.000000 21.000000 22.000000 23.000000



-------------在C中创建的矩阵, 在C中输出-------------
在c中创建指针形式的以一维线性储存的三维数组d_1d:
d_3d:
depth = 0
0.000000 1.000000 2.000000 3.000000
4.000000 5.000000 6.000000 7.000000
8.000000 9.000000 10.000000 11.000000

depth = 1
12.000000 13.000000 14.000000 15.000000
16.000000 17.000000 18.000000 19.000000
20.000000 21.000000 22.000000 23.000000
```

```
在c中d_1d赋值给三维数组d_3d:
d_3d:
depth = 0
0.000000 1.000000 2.000000 3.000000
4.000000 5.000000 6.000000 7.000000
8.000000 9.000000 10.000000 11.000000

depth = 1
12.000000 13.000000 14.000000 15.000000
16.000000 17.000000 18.000000 19.000000
20.000000 21.000000 22.000000 23.000000


把从matlab中传来的一维形式的三维数组d, 赋值给c中创建的三维数组d_3d:
d_3d:
depth = 0
0.000000 1.000000 2.000000 3.000000
4.000000 5.000000 6.000000 7.000000
8.000000 9.000000 10.000000 11.000000

depth = 1
12.000000 13.000000 14.000000 15.000000
16.000000 17.000000 18.000000 19.000000
20.000000 21.000000 22.000000 23.000000


c中创建的三维数组d_3d赋值给要输出到matlab中的output_dd(一维):
output_dd在C中的输出为:
depth = 0
0.000000 1.000000 2.000000 3.000000
4.000000 5.000000 6.000000 7.000000
8.000000 9.000000 10.000000 11.000000

depth = 1
12.000000 13.000000 14.000000 15.000000
```

```
16.000000 17.000000 18.000000 19.000000
20.000000 21.000000 22.000000 23.000000


output_dd在matlab中的输出为:

(:,:,1) =

     0     1     2     3
     4     5     6     7
     8     9    10    11


(:,:,2) =

    12    13    14    15
    16    17    18    19
    20    21    22    23
```