Tercera_clase

December 13, 2018

1 Tercera clase (12/12/2018)

```
In [14]: M_1 = zeros(5)
         A_1 = [1,3;-3,1]


M_1 =

     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


A_1 =

     1     3
    -3     1



In [15]: M_1(1:2,1:2) = M_1(1:2,1:2) + A_1


M_1 =

     1     3     0     0     0
    -3     1     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


In [11]: help("zeros")
```

```
ZEROS  Zeros array.
    ZEROS(N) is an N-by-N matrix of zeros.

    ZEROS(M,N) or ZEROS([M,N]) is an M-by-N matrix of zeros.

    ZEROS(M,N,P,...) or ZEROS([M N P ...]) is an M-by-N-by-P-by-... array of
    zeros.

    ZEROS(SIZE(A)) is the same size as A and all zeros.

    ZEROS with no arguments is the scalar 0.

    ZEROS(..., CLASSNAME) is an array of zeros of class specified by the
    string CLASSNAME.

    ZEROS(..., 'like', Y) is an array of zeros with the same data type, sparsity,
    and complexity (real or complex) as the numeric variable Y.

    Note: The size inputs M, N, and P... should be nonnegative integers.
    Negative integers are treated as 0.

    Example:
        x = zeros(2,3,'int8');

    See also EYE, ONES.

    Reference page in Doc Center
        doc zeros

    Other functions named zeros

        codistributed/zeros       codistributor2dbc/zeros      gpuArray/zeros
        codistributor1d/zeros     distributed/zeros
```

In [16]: M_1(4:5,4:5) = M_1(4:5,4:5)+A_1

M_1 =

```
    1    3    0    0    0
   -3    1    0    0    0
    0    0    0    0    0
    0    0    0    1    3
    0    0    0   -3    1
```

```
In [13]: M_2 = zeros(5)
         A_2 = [1,3;-3,1]
         M_2(1:2,1:2) = M_2(1:2,1:2)+A_2
         M_2(2:3,2:3) = M_2(2:3,2:3)+A_2
```

M_2 =

     0   0   0   0   0
     0   0   0   0   0
     0   0   0   0   0
     0   0   0   0   0
     0   0   0   0   0

A_2 =

     1   3
    -3   1

M_2 =

     1   3   0   0   0
    -3   1   0   0   0
     0   0   0   0   0
     0   0   0   0   0
     0   0   0   0   0

M_2 =

     1   3   0   0   0
    -3   2   3   0   0
     0   -3   1   0   0
     0   0   0   0   0
     0   0   0   0   0


```
In [18]: M_3 = zeros(5);
         A=[1,3; -3, 1];

         for k = 1:5-1
             M_3(k:k+1, k:k+1) = M_3(k:k+1, k:k+1) + A;
         end

         M_3
```

```
M_3 =

     1     3     0     0     0
    -3     2     3     0     0
     0    -3     2     3     0
     0     0    -3     2     3
     0     0     0    -3     1
```

In [25]: c = zeros(5,1);
         b_base = [3;4]; % Otra manera es colocar c = [3, 4]'
         for k=1:5-1
             c(k:k+1) = c(k:k+1) + b_base
             pause(5)
         end

```
c =

     3
     4
     0
     0
     0


c =

     3
     7
     4
     0
     0


c =

     3
     7
     7
     4
     0


c =
```

3
7
7
7
4