TECHNISCHE UNIVERSITÄT DRESDEN

INTRODUCTION TO MATLAB

Vectors and matrices

Dario Cuevas and Vahid Rahmati

Dresden, 12. November 2014

01 Scripts

- Scripts are successions of commands. Executed in the order found (from top to bottom).
- % at the beginning of a line means that it's a comment and won't be ran.
- Use ; at the end of each command to suppress the output of that command.
- To run the script, use F5.
- Use %% to divide the script in independent cells.
- To run a cell, press ctrl+Enter.
- Script names can have letters, underscores and numbers. Just like variables.
- Remember, all will be saved to the workspace (command window). Variables will be overwritten.

02 Elementwise Operations

Matrix times matrix:

$$\mathrm{A} = \begin{pmatrix} \mathrm{a} & \mathrm{b} \\ \mathrm{c} & \mathrm{d} \end{pmatrix} \mathrm{B} = \begin{pmatrix} \mathrm{w} & \mathrm{x} \\ \mathrm{y} & \mathrm{z} \end{pmatrix}$$

$$\mathrm{A}. * \mathrm{B} = \begin{pmatrix} \mathrm{aw} & \mathrm{bx} \\ \mathrm{cy} & \mathrm{dz} \end{pmatrix} \neq \mathrm{A} * \mathrm{B}$$

$$\mathrm{A}./\mathrm{B} = \begin{pmatrix} \mathrm{a/w} & \mathrm{b/x} \\ \mathrm{c/y} & \mathrm{d/z} \end{pmatrix} \neq \mathrm{A/B}$$

$$\mathrm{A}. \pm \mathrm{B} = \mathrm{A} \pm \mathrm{B} = \begin{pmatrix} \mathrm{a} \pm \mathrm{w} & \mathrm{b} \pm \mathrm{x} \\ \mathrm{c} \pm \mathrm{y} & \mathrm{d} \pm \mathrm{z} \end{pmatrix}$$

$$\mathrm{A}. \hat{} \; 2 = \begin{pmatrix} \mathrm{a}^2 & \mathrm{b}^2 \\ \mathrm{c}^2 & \mathrm{d}^2 \end{pmatrix} \neq \mathrm{A} \; \hat{} \; 2$$

Note: the sizes of the two matrices in elementwise operations must be exactly the same.

02 Exceptions

- `2+ones(2,3)`
- `2*ones(2,3)`
- `2./ones(2,3)`
- `2.^ones(2,3)`

- Do not use `a(1:10,1) = []`. Use `a(:,1) = []` instead

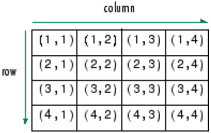
02 Exercises

1. Compute $S(N) = \sum_{n=1}^{N} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N}$, for $N = 100$
2. Compute $G(N) = \sum_{n=1}^{N} x^n = x + x^2 + x^3 + \cdots + x^N, x = 0.5$, for $N = 100$

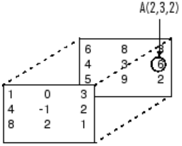
03 Variable types

- Multidimensional arrays
- Cell
- Structures
- Strings

03 Multidimensional arrays



(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

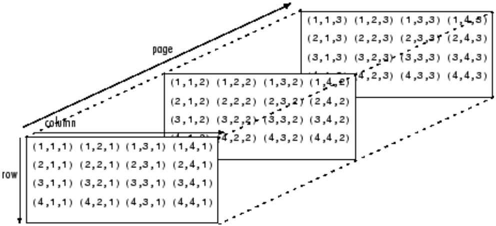


```
A(:,:,1) =

     1     0     3
     4    -1     2
     8     2     1


A(:,:,2) =

     6     8     3
     4     3     6
     5     9     2
```



03 Multidimensional examples

Example 1:

```
A(:,:,1) = magic(5);
A(:,:,2) = zeros(5);
A(:,:,3) = ones(5);
```

Example 2:

```
A = zeros(2,2,4);
```

Example 3:

```
A = ones(3,6,5);
```

Exercise:

- Create a matrix 4x4x3, such that the first layer has 1s in the diagonal, the second has 2s, the third has 3s.
- Create a 6x6x10, such that the first five layers have just 1s, layers from 6 to 9 have just 0s, the 10th layer is:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

03 Cells and structures

Cells:
They are similar to arrays, but each element can have a different size
Example:
To initialize a cell array:

```
A = cell(3,2)
```

To index, use curly brackets:

```
A{1,1} = magic(5);
A{3,2} = zeros(2,1);
```

To index a cell's element's elements: `A{1,1}(1,1)`

Structures:
Like Cells, but indexed with names:
Example:
For a structure named “subject”,

```
subject.age = 30;
subject.country = ’Mexico’;
subject.height = 1.83;
subject.results = [1, 0, 1, 1, 0];
```

To index the element's element, `subject.results(5)`

03 Cells and structures exercises

1. Create a vector-cell CellA whose first element is `[1]`, the second `[1, 2]`, then `[1,2,3]`, etc., until 5. The 6th element is `magic(7)`. The 7th one is empty.
2. Create a structure called MyStruct with elements: NoOfClassmates, CurrentYear, MyCell and Magia. The value of MyCell should be CellA from the previous exercise. The value of Magia should be the 6th element of CellA.
3. From MyStruct, change the 7th element of MyCell (that is, `MyStruct.MyCell{7}`) to `rand(2,10)`

03 Strings

Strings are arrays of letters.

```
A = 'I am a Vahid';
```

They are indexed like an array:

`A(1)` gives I , `A(2)` gives (empty space) ;

To create two-dimensional arrays of chars:

```
B = char(A, 'Yes I am');
```

Note: `C = '52';` is NOT a number. `C+5` throws an error. Examples for indexing:

`A(8:end)` gives Vahid
`B(2,1:3)` gives Yes

Exercise: Substitute Vahid's name for your own in A. You might have to add or delete characters at the end.