KU LEUVEN

MATLAB

Vector creation

Topics

- Creating Vectors
  - manually entering the elements
  - *how to build an array in a fast way*
  - `:`
  - `linspace, logspace`

lingo

- MATLAB has two different types of arithmetic operations: array operations and matrix operations.
  - Matrix operations follow the rules of linear algebra.
  - Array operations execute element by element operations and support multidimensional arrays.

lingo

- **array**:
  - a collection of numbers, (elements or entries) referenced by one or more indices running over different index sets.
  - most basic data structure in MATLAB
- **dimension** of the array: the number of indices needed to specify an element.
  - MATLAB also supports data structures that have more than two dimensions.

lingo

- **matrix** is a two-dimensional array with special rules for addition, multiplication, and other operations. (*linear algebra* world).
- **vector** is a matrix for which one dimension has only the index 1
- In MATLAB, the index sets are always sequential integers starting with 1.

Creating Vectors

- Different ways:
  - Enumeration: specify each element explicitly,
  - Use the colon : operator,
  - Use the commands `linspace` or `logspace`
  - Elementary (built-in) arrays

Creating vectors: enumeration

- Create an array: use the square brackets **[ ]**
- Specify each element explicitly
- Numbers (or variables) inside the brackets can be separated by blanks, commas or semicolons
  - blank or comma separator: separate elements in a row
  - semicolon: separate rows

```
vec1 = [1, 2, 3, 4.6, 8, 9]

vec1 = 1×6

1.0000 2.0000 3.0000 4.6000 8.0000 9.0000


vec2 = [1 2 33.6i]

vec2 = 1×3 complex

1.0000 + 0.0000i 2.0000 + 0.0000i 0.0000
+33.6000i


vec3 = [1.1; 1.2; 66.78; -9.31e9]

vec3 = 4×1

10^9 ×

0.0000
0.0000
0.0000
-9.3100
```


Creating Vectors: Colon Operator

- Create vectors, array subscripting, and for iterations
- The colon (:) is one of the most useful operators in MATLAB:
  from_ : in increments of_ : to _
- The colon operator uses the following rules to create regularly spaced vectors:
  - j:k is the same as [j,j+1,...,k]
  - j:k is empty if j > k
  - j:i:k is the same as [j,j+i,j+2i, ...,k]
  - j:i:k is empty if i > 0 and j > k or if i < 0 and j < k

Example

```
vec1 = [1, 2, 3, 4.6, 8, 9]
vec1 = 1×6
1.0000 2.0000 3.0000 4.6000 8.0000 9.0000

vec2 = [1 2 33.6i]
vec2 = 1×3 complex
1.0000 + 0.0000i 2.0000 + 0.0000i 0.0000 +33.6000i

vec3 = [1.1; 1.2; 66.78; -9.31e9]
vec3 = 4×1
10^9 ×
0.0000
0.0000
0.0000
-9.3100
```


Creating Vectors

- `linspace`
  - syntax: `x = linspace(first, last, n)`
  - creates linearly spaced row vector starting with *first*, ending at *last* , having n elements
- `logspace`
  - syntax: `x = logspace(first, last, n)`
  - creates logarithmically spaced row vector starting with 10^{first}, ending at 10^{last} , having n elements

Demo

- Creating vectors
- *File: create_vectors.mlx used in screencast matlab_array_create_vectors*