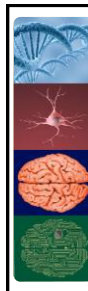
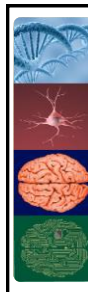


Introduction to Programming 2017/2018

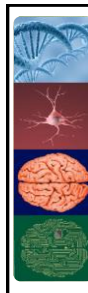
Tirgul 2: Variables - Arrays

Michal Israelashvili
Yocheved Loewenstern



Lesson Outline

- Variables
  - Arrays: scalars, vectors, matrices
  - Indexing: retrieval and assignment
- Arrays concatenation
- Size & length
- Vector transformations
- Arithmetic operations
- Special Matrices – ones, zeros
- Deleting variables



Arrays

- Arrays are tools for arranging and handling data (mainly numerical data).
- We will work only with 2-dimensional arrays (rows and columns).

- Scalar: single number (size 1x1).
- Vector: one line or one column (size 1xN or Nx1).
- Matrix: a table with several lines and/or columns, has to be a rectangle (size MxN).

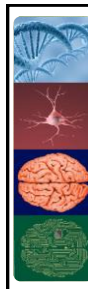
- Examples: *Tirgul2Arrays.m (2-4)*

Arrays concatenation

- *Concatenation* is the process of joining small arrays to make bigger ones.
- New arrays can be created by concatenating their elements within square brackets.
- Examples: *Tirgul2Arrays.m (2)*

Size & length

- The *size* of an array is the number of elements in each dimension.
- The *length* is the number of elements in a **vector**, or the number of elements in the longest dimension of a **matrix**.
- The *numel* command returns the total number of elements in an array.



Size & length - syntax

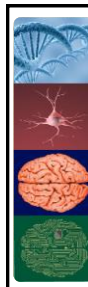
- The syntax of these commands:
  - *outVar* = length(*arrayName*)
  - *outVar* = numel(*arrayName*)
  - *outVar* = size(*arrayName*)
    → *outVar = [numRows, numColums]*
  - *outVar* = size(*arrayName,1*)
    → *outVar = [numRows]*
  - *outVar* = size(*arrayName,2*)
    → *outVar = [numColums]*
- Examples: *Tirgul2Arrays.m (5)*

Vector Transformation

- Turn a row vector into a column vector or a column vector into a row vector (transpose).
- Syntax: vector_name'
- Example:

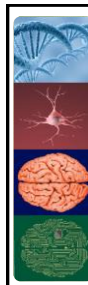
  vector = [5 2 8 7] → vector' = $\begin{pmatrix} 5 \\ 2 \\ 8 \\ 7 \end{pmatrix}$

- Note: Matrix transformation can be performed in a similar way.
- Examples: *Tirgul2Arrays.m (6)*



Arithmetic operations

- We are only dealing with **element-wise** operations in this course (no advanced linear algebra).
- All types of operations can be performed between vectors/matrices and scalars.
- Operations between two matrices/vectors can only be performed on two **equally-sized arrays**.
- Standard "order of operations" is followed. You may enclose mathematical statements in brackets () to set priorities.
- Examples: *Tirgul2Arrays.m (7)*

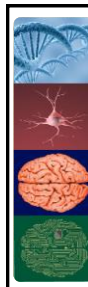


Arithmetic operations

- For element-wise multiplication and division of two matrices or two vectors, and for power of matrices and vectors always use a dot before the mathematical operation(.* , ./ , .^)
- For operations between a vector/matrix and a scalar you can use the regular sign without the dot (* , / , ^)

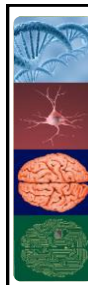
Variables - Special Matrices

- A few commands enable us to create special matrices.
  - 'zeros' – matrix filled with zeros.
  - 'ones' – matrix filled with 1's.
  - 'rand' – matrix filled with random numbers between 0-1.
- syntax: zeros(m,n) → creates a matrix m x n filled with zeros.
- Examples: *Tirgul2Arrays.m (8)*



"rand" command

- The command "rand" generates arrays of any size filled with random numbers between 0-1.
- Syntax: *rand(M,N)*
  example: *myRand = rand(2,3)* → 2x3 matrix of random numbers between 0-1
- For generating random numbers between *a* to *b* use: *myVar = a + (b-a) * rand(M,N)*
  example: *myVar = 5 + (6) * rand(1,4)* →
  row vector containing random numbers between 5 –> 11.
- **Note**: different numbers will be generated each time you use the "rand" command.



Conversion to integers (round numbers)

- Matlab has several commands that convert numbers into **integers** (=round numbers).
- Commands: round, ceil, floor
- Syntax:
  - round(x)
  - ceil(x)
  - floor(x)
- Examples: *Tirgul2Arrays.m (8)*

Deleting Variables

- clear – clear all variables.
- clear var1 var2 – clears only selected variables.
- Note: deleted variables cannot be restored (only calculated again), so be careful!
- Examples: *Tirgul2Arrays.m (9)*

Writing code – A note

- In your code files, try to keep line length under 75 characters.
  - Use … to complete your command at the next line.

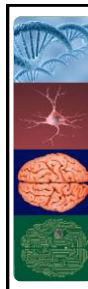
    Examples:

    myVec = [1 3 4 4 5 6 …

    2 3 4 4 3] ;

    sizeMat = size…

    (myMatrix,2) ;



Functions/Commands List

- Arithmetic operations: +,-,.*,./,.^, sqrt
- ;
- Size, length, numel
- ones, zeros
- rand
- round, ceil, floor
- clear
- ‘
- …

Reading Material

- MATLAB Programming Style Guidelines, variables naming conventions, Pages 2-4 (until constants, including).
  - At the web-site: Additional links.