MatLab Programming Fundamentals

guarantor:	Maroš Tunák
tel.:	3465
e-mail:	*maros.tunak@tul.cz*

Course objectives

The aim of the course is to acquire basics knowledge and skills of students the MatLab program. At the end of the course students will be able to use MatLab for their own work and will be ready to deepen their programming skills in MatLab.

MatLab Programming Fundamentals

time requirements:	0p+2c
credits:	4
exercises:	Monday 10:40-12:15; 12:30-14:05 (B-PC2, Tunák M.)
	Tuesday 08:50-10:25; 10:40-12:15 (B-PC2, Tunák M.)
consultation:	Wednesday 10:40-12:15 (E-KHT)

Requirements on student/graded credit

1. participation in exercises (max. 3 absences)
2. elaboration of semester work (after approval of the semester work, you can attend a practical demonstration)
3. practical demonstration of acquired skills (there will be 1-2 examples to solve; elaboration time 1 hour; you can use any materials ...)

Content

IS/STAG Syllabus

1. Getting started with Matlab. Working environment, windows, paths, basic commands, variables. Loading, saving and information about variables. Help.

2. Mathematics with vectors and matrices. Creating vectors and matrices. Indexing. Special matrices. Matrix operations. Element by element operations. Relational operations, logical operations, examples and tricks.

3. Control flow. Loops, conditional statements, examples.

4. Script m-files, Function m-files.

5. Visualisation. Two-dimensional graphics. Three-dimensional graphics.

6. Graphical user interface.

7.-10. Statistics and Machine Learning Toolbox. Basics of statistical data processing, exploratory data analysis, descriptive statistics, data visualisation, hypothesis testing, confidence intervals, regression analysis, control charts.

11.-13. Solution of practical problems in textile and industrial engineering.

Literature

Recommended

MathWorks. *Getting Started with MATLAB.* [Online]. Dostupné z: https://www.mathworks.com/help/matlab/getting-started-with-matlab.html

Study materials

http://elearning.tul.cz

Installation

http://liane.tul.cz/cz/software/MATLAB

Mathematics with vectors and matrices. Creating vectors and matrices. Indexing. Special matrices. Matrix operations. Element by element operations. Relational operations, logical operations, examples and tricks.

Matrix operations

MatLab has two different types of arithmetic operations:

- matrix arithmetic operations are defined by the rules of linear algebra

Command	Operations
'	transposition
+ -	adding and subtracting of matrices
	matrices of same size
+ -	adding and subtracting scalar from matrix elements
*	matrix multiplication
	number of columns of matrix A = number of rows of matrix B

$$A_{mxn} * B_{nxp} = C_{mxp}$$

$$C(i,j) = \sum_{k=1}^{n} A(i,k) * B(k,j)$$

Matrix operations

- have a line vector a, a column vector b and a matrix A

```
>> a=[1 2 3]
a =
     1     2     3

>> b=[4;5;6]
b =
     4
     5
     6

>> A=[1 2 3; 4 5 6]
A =
     1     2     3
     4     5     6
```

Matrix operations

- transposition '

```
>> aT=a'

aT =
     1
     2
     3

>> AT=A'

AT =
     1     4
     2     5
     3     6
```

Matrix operations

- adding and subtracting of matrices (matrices of same size)

```
>> a+a

ans =
     2     4     6

>> A+A

ans =
     2     4     6
     8    10    12

>> A+b

Matrix dimensions must agree.
```

Matrix operations

- adding row and column vectors (the result in this case is a 3-by-3 matrix, where each (i, j) element in the matrix is equal to $a(j) + b(i)$)

```
>> a+b

ans =
     5     6     7
     6     7     8
     7     8     9

>> b-a

ans =
     3     2     1
     4     3     2
     5     4     3
```

Matrix operations

- adding and subtracting scalar from matrix elements

```
>> b+10

ans =
    14
    15
    16

>> A-6

ans =
    -5     -4     -3
    -2     -1      0
```

Matrix operations

- multiplication of scalar and vectors or matrix

```
>> b*10

ans =
    40
    50
    60

>> pi*A

ans =
    3.1416    6.2832    9.4248
   12.5664   15.7080   18.8496
```

Matrix operations

- matrix multiplication of a*b vectors results in a scalar (matrix multiplication (number of columns of matrix A = number of rows of matrix B))

```
>> a*b

ans =

    32
```

- and matrix multiplication of vectors b*a is a matrix of size 3 × 3

```
>> b*a

ans =

     4     8    12
     5    10    15
     6    12    18
```

Matrix operations

- A*A' results in a matrix of size 2 × 2

```
>> A*A'

ans =

    14    32
    32    77

>> A*b

ans =
    32
    77
```

Matrix operations

- in the case of a*a, matrix multiplication cannot be performed due to dimensions

```
>> a*a
Error using *Incorrect dimensions for matrix multiplication.  Check
that the number of columns in the first matrix matches the
number of rows in the second matrix.  To perform elementwise
multiplication, use ’.*’.
```

Element-by-Element Operations

- array arithmetic operations are carried out element-by-element, and can be used with multidimensional arrays

Command	Operations
.	element-by-element operations (dot before operand)
.^	dot power
.*	dot product
	matrices of the same dimension
	element-by-element product

$$A_{mxn} .* B_{mxn} = C_{mxn}$$

$$C(i,j) = A(i,j) .* B(i,j)$$

Element-by-Element Operations

- result of a element multiplication of `a.*a` of a matrix of the same size 1 × 3

```
>> a.*a

ans =
     1     4     9

>> A.*A

ans =
     1     4     9
    16    25    36
```

Element-by-Element Operations

- dot power

```
>> a^2
Error using ^
Incorrect dimensions for raising a matrix to a power.  Check that
the matrix is square and the power is a scalar.  To perform
elementwise matrix powers, use .^


>> a.^2

ans =
     1     4     9
```

Relational operators

Relational operators perform relational operations. Returns a logical array whose elements are 1 (true) and 0 (false)

Command	Operation
$<$	less then
$>$	greater then
$<=$	less than or equal
$>=$	greater than or equal
$==$	equal
$\sim=$	not equal

Relational operators

```
>> e=1:5

e =
     1     2     3     4     5

>> e⩾3

ans =
  1 x 5 logical array
   0   0   1   1   1

>> e==2
ans =

  1 x 5 logical array
   0   1   0   0   0
```

Relational operators

- command `find` returns indices and values of nonzero elements

```
>> ind=find(e⩾3)

ind =
     3     4     5

>> [r s]=find(e⩾3)

r =
     1     1     1


s =
     3     4     5

>> e(r,s)=100*e(r,s)

e =
     1     2   300   400   500
```

Logical operators

Logical operators are applied to arrays (applied by elements) or scalars. The result is a logical zero, ie. false, or a logical one, ie. true

Command	Operation
&	AND
|	OR
~	NOT

Logical operators

```
>> (e⩽2) | (e==5)

ans =

  1x5 logical array

   1   1   0   0   1


>> (e⩽4) & (e≠3)

ans =

  1x5 logical array

   1   1   0   1   0
```

Logical operators

```
>> G=[1 0 0 1 1], H=[0 0 1 1 1]
G =
     1     0     0     1     1
H =
     0     0     1     1     1

>> G & H
ans =
  1x5 logical array
   0   0   0   1   1

>> G | H
ans =
  1x5 logical array
   1   0   1   1   1
```

Examples for practice

Examples for practice

1. calculate the volume of cubes with side lengths [2.1 3.3 4] cm
2. calculate the functional values of $y = cos(x)$ for x in points $0, \pi/4, \pi/2, \dots, 2\pi$ and in the vector y change the sign for the element, which are less than zero
3. create vector of numbers $< -1, 2 >$ containing 15 elements and remove the negative elements from the vector
4. generate 3×5 matrix of $N(0, 1)$ random numbers, find the position (row,column) of maximum value and increase the value 10 times
5. for a vector t (t=1:0.2:2), write down the MATLAB expression that will compute:
   - $\ln(2 + t + t^2)$,
   - $\cos(t)^2$ - $\sin(t)^2$,
   - $e^t(1+\cos(3t))$,
   - $\tan^{-1}(t)$.
6. for given matrices solve following equations:

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 1 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & -2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

   - A + X_1 = B,
   - (A - X_2)C = B.

7. create an integer random 5×6 matrix A with elements from interval [-7,7] and compute the sum of its line maxims.

Solution