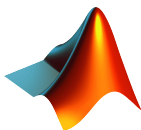


# MatLab Programming Fundamentals

guarantor: Maroš Tunák

tel.: 3465

e-mail: [maros.tunak@tul.cz](mailto: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>

## Visualisation. Two-dimensional graphics.

## Two-dimensional graphics

The possibilities of creating 2-D graphs in MatLab are huge. The following will be listed basic possibilities of using the `plot` command. When `plot` is called, a new Figure window (if not already open) is open, axes are created and points are drawn. The default is to draw in blue color, with no marker type and solid line. line icon.

- for example plotting of function  $x^2$  at an interval of -10 to 10 with division step 2. First, we need to create a vector  $x$  at the specified interval and calculate the functional values for each  $x$ . Simple plotting is done by the command `plot(x,y)`, where the vectors  $x$  and  $y$  must have same size:

```
>> x=-10:2:10;  
>> y=x.^2;  
>> plot(x,y)
```

# Two-dimensional graphics

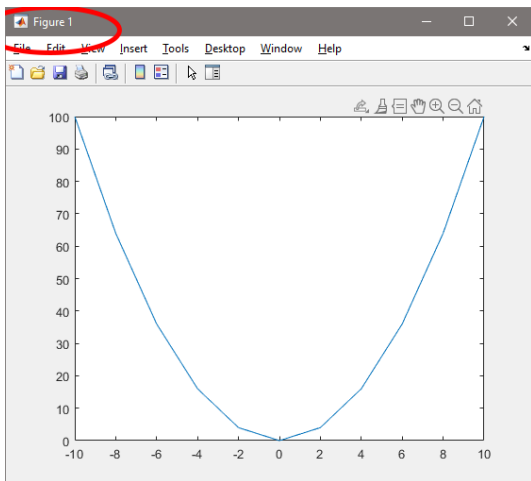


Figure: Figure Window

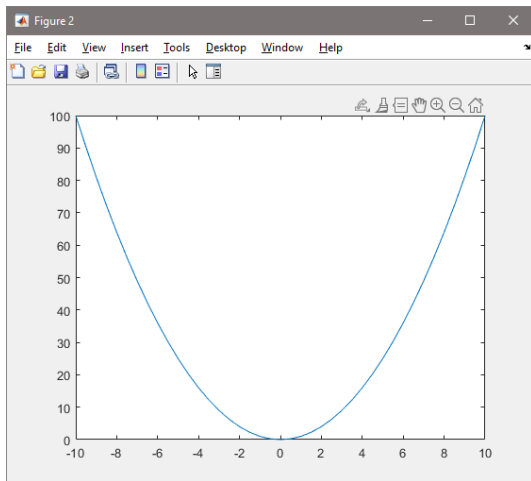
## Two-dimensional graphics

- smaller division step, new figure window

```
>> x=-10:0.5:10;  
>> y=x.^2;  
>> figure,plot(x,y)
```



# Two-dimensional graphics

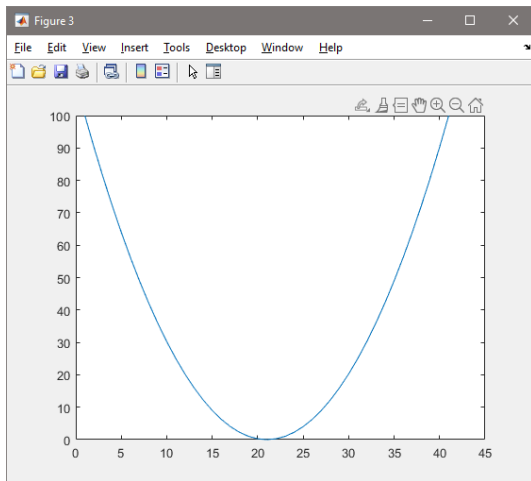


## Two-dimensional graphics

- plot of vector  $y$  against index order

```
>> figure,plot(y)
```

# Two-dimensional graphics

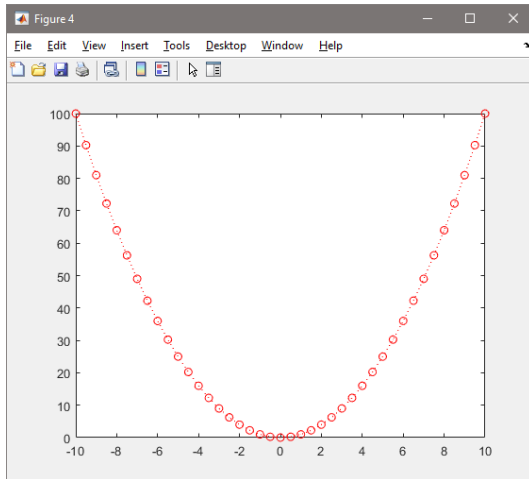


## Two-dimensional graphics

- quick adjustment of line specifications is performed by means of a text string (color, line style, marker type), eg. red color, circle, dotted line

```
>> figure,plot(x,y,':or')
```

# Two-dimensional graphics



# Line specifications

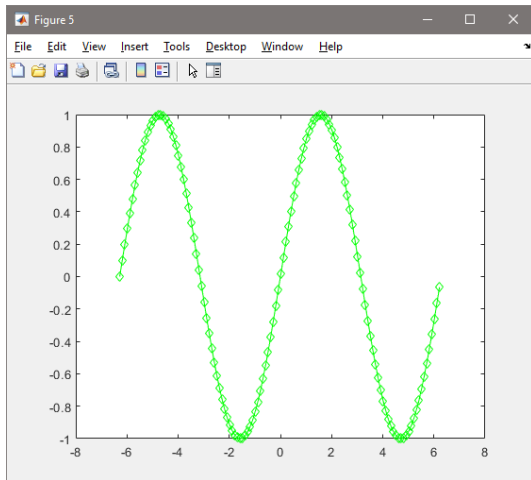
Symbol	Line style	Symbol	Marker type
-	solid line	+	plus
:	dashed line	o	circle
-.	dotted line	*	asterisk
-.	dash-dotted line	.	point
		x	cross sign
Symbol	Color	s	čtverec
r	Red	d	diamond
g	Green	^	upward-pointing triangle
b	Blue	v	downward-pointing triangle
c	Cyan	<	left-pointing triangle
m	Magenta	>	right-pointing triangle
y	Yellow	p	pentagram
k	black	h	hexagram
w	White		

## Two-dimensional graphics

- example of plotting function  $y = \sin(x)$  on interval  $\langle -2\pi, 2\pi \rangle$ , green color, diamond, solid line

```
>> x=-2*pi:0.1:2*pi;  
>> y=sin(x);  
>> plot(x,y,'gd-')
```

# Two-dimensional graphics



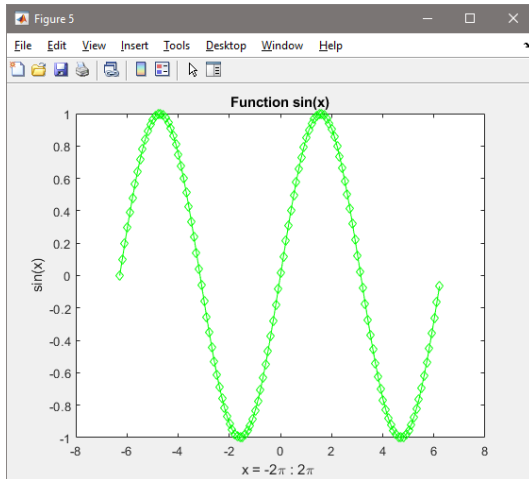


## Two-dimensional graphics

- inserting title, axis labels x and y

```
>> title('Function sin(x)')  
>> xlabel('x = -2\pi : 2\pi ' )  
>> ylabel('sin(x)')
```

# Two-dimensional graphics

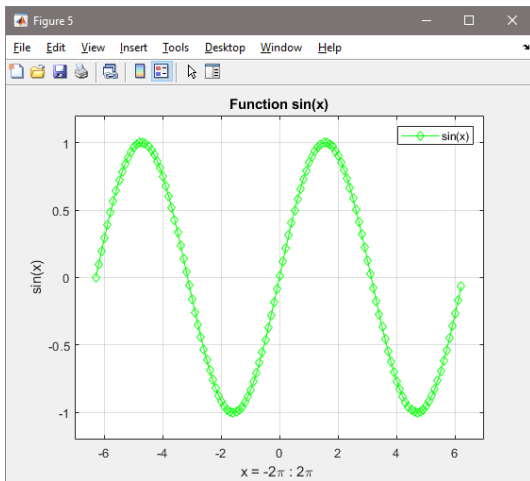


## Two-dimensional graphics

- grid insertion, setting of axis limits, legend

```
>> grid on  
>> axis([-7 7 -1.2 1.2])  
>> legend('sin(x)')
```

# Two-dimensional graphics

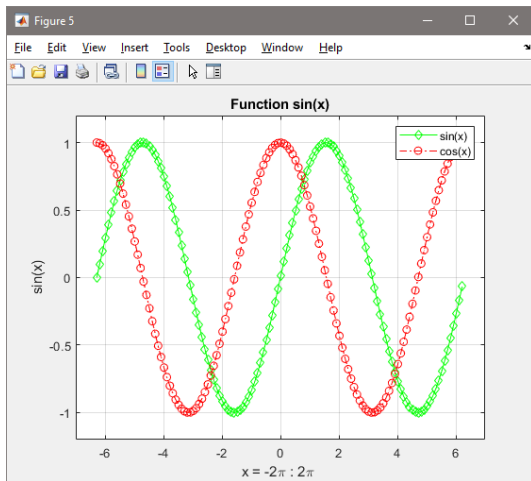


## Two-dimensional graphics

- to the same graph display function  $z = \cos(x)$  on the interval  $\langle -2\pi, 2\pi \rangle$ , color red, circle, dashed line

```
>> z=cos(x);  
>> hold on  
>> plot(x,z, 'or-.')  
>> legend('sin(x)', 'cos(x)')
```

# Two-dimensional graphics

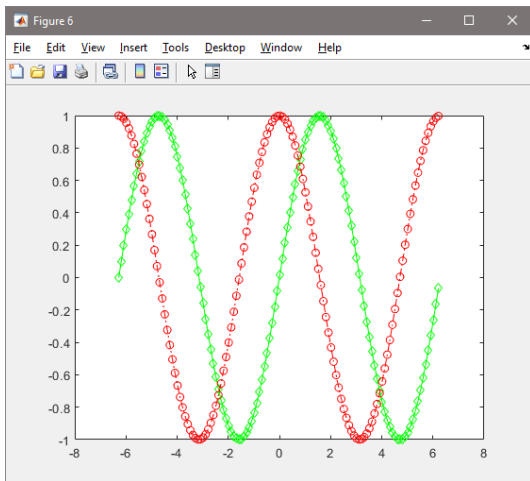


## Two-dimensional graphics

- or individual pairs can be plotted with one command

```
>> figure,plot(x,y,'gd-',x,z,'or-.')
```

# Two-dimensional graphics





## Two-dimensional graphics

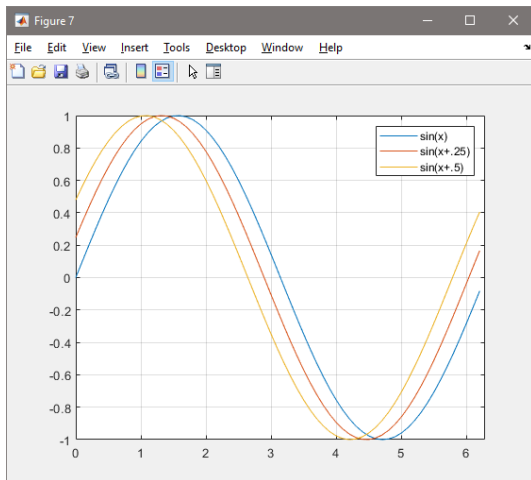
Command	Operation
» plot(y)	plot a vector y vs index
» plot(x,y)	plot a vector y vs vector x
» plot(x,y,'p')	plot a vector y vs vector x with parameters
» title('text')	title above a graph
» xlabel('text')	label text on the x axis
» ylabel('text')	label text on the y axis
» text(x,y,'text')	adds text at the point (x,y)
» gtext('text')	adds text at manually indicated point
» grid on/off	adds grid to the plot at the tick marks
» axis([xmin xmax ymin ymax])	axis limits
» xlim([xmin xmax])	x axis limits
» ylim([ymin ymax])	y axis limits
» legend('fun1','fun2')	plots a legend box
» legend off	delete the legend box

## Two-dimensional graphics

- Example: draw three functions  $y_1 = \sin(x)$ ,  $y_2 = \sin(x + 0.25)$ ,  $y_3 = \sin(x + 0.5)$  on interval  $\langle 0, 2\pi \rangle$  in different color with solid line, legend, grid

```
>> x=0:0.1:2*pi;  
>> y1=sin(x);  
>> y2=sin(x+0.25);  
>> y3=sin(x+0.5);  
>> plot(x,y1,x,y2,x,y3)  
>> xlim([0,2*pi])  
>> grid on  
>> legend('sin(x)', 'sin(x+.25)', 'sin(x+.5)')
```

# Two-dimensional graphics

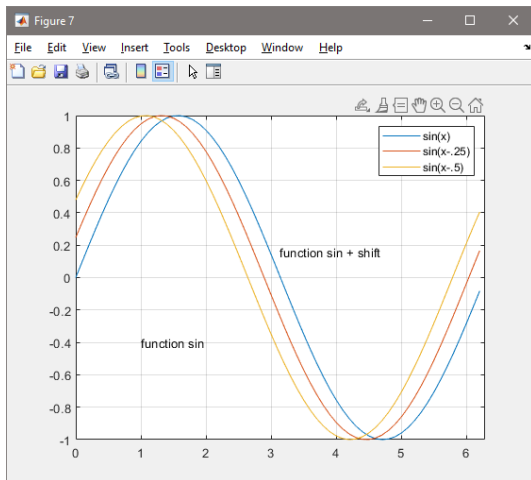


## Two-dimensional graphics

- insert text at a defined position or user selected

```
>> text(1,-0.4,'function sin')  
>> gtext('function sin + shift')
```

# Two-dimensional graphics



## Two-dimensional graphics

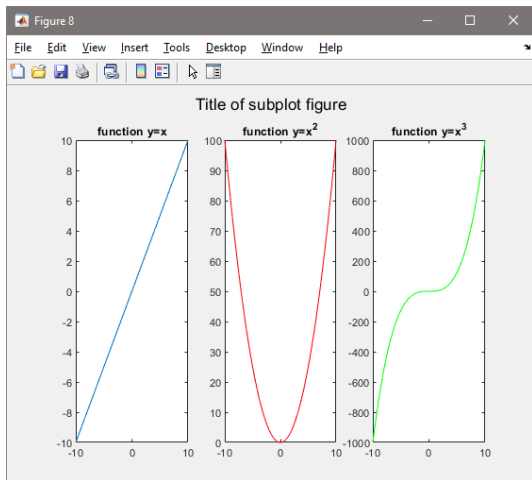
Command	Operation
» figure	new figure window
» clf	clear the current figure
» close	closing of current figure
» hold on/off	retain current graph in figure
» subplot(m,n,p)	create axes in tiled positions

## Two-dimensional graphics

- Sometimes it is useful to bring several coordinate systems into one graphic window. This is possible using the `subplot` command, which creates a  $m \times n$  coordinate system matrix in one graphical window. The first number represents the number of rows, the second number of columns and the third position from the upper left corner. For example, functions  $\sin(x)$ ,  $\cos(x)$ ,  $x^2$  side by side

```
>> x=-10:0.1:10;
>> y1=x;
>> y2=x.^2;
>> y3=x.^3;
>> subplot(1,3,1),plot(x,y1),title('function y=x')
>> subplot(1,3,2),plot(x,y2,'r'),title('function y=x^2')
>> subplot(1,3,3),plot(x,y3,'g'),title('function y=x^3')
>> sgtitle('Title of subplot figure')
```

# Two-dimensional graphics



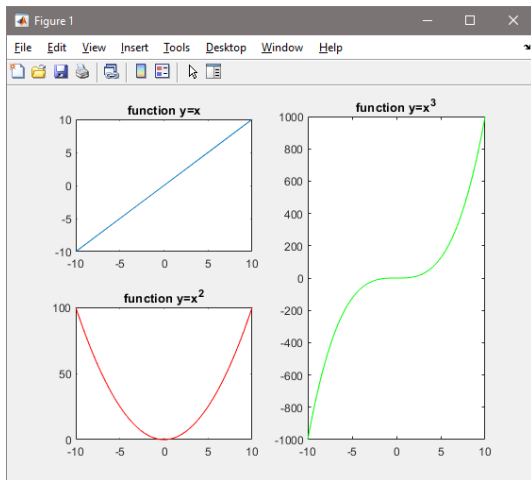


## Two-dimensional graphics

- or joining positions

```
>> figure
>> subplot(2,2,1),plot(x,y1),title('function y=x')
>> subplot(2,2,3),plot(x,y2,'r'),title('function y=x^2')
>> subplot(2,2,[2 4]),plot(x,y3,'g'),title('function y=x^3')
```

# Two-dimensional graphics



# Two-dimensional graphics

Examples for practice

## Examples for practice

- 1 Draw a graph of the theoretical values of area coverage  $Z$  for set of  $n$  jerseys 1, 2, ..., 8, if  $Z = [0.615 \ 0.631 \ 0.657 \ 0.694 \ 0.671 \ 0.716 \ 0.701 \ 0.734]$ , insert axis labels.
- 2 Draw a graph of the dependence of porosity  $P[\%] = [1.91 \ 5.46 \ 8.73 \ 11.42 \ 15.01]$  on the relative elongation of kinesiotape  $\epsilon[\%] = [0 \ 13.3 \ 25 \ 33.3 \ 50]$ . Insert axis labels.
- 3 Generate a  $X$  vector that contains 100 values from  $N(0, 1)$  and in one Figure window draw even  $X$  data with a red circle against the index and to the graph next to it odd  $X$  data with a green circle and a line.
- 4 Draw probability density functions of normal distribution  $N(\mu, \sigma^2)$  for parameters  $\mu = [0 \ 0 \ 0 \ 2]$  and  $\sigma = [1 \ 2 \ 0.5 \ 1]$ , ie.  $N(0.1)$ ,  $N(0.2)$ ,  $N(0.0.5)$ ,  $N(2.1)$  at  $x$  in  $< -10, 10 >$  to one chart with different color, insert legend. The probability density function of a normal distribution is defined by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

## Solution