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

Visualisation. Three-dimensional graphics.

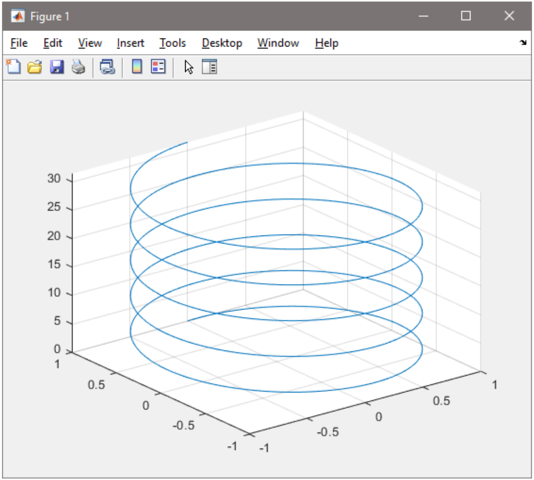
Three-dimensional graphics

MatLab provides a number of functions for displaying 3-D data in the form of lines, meshes or surfaces

- the `plot3` function displays a three-dimensional graph of a set of data points, where x, y, z are three vectors of the same length, for example:

```
>> t = 0:pi/50:10*pi;
>> st = sin(t);
>> ct = cos(t);
>> figure
>> plot3(st,ct,t)
>> grid on
```

Three-dimensional graphics



Three-dimensional graphics

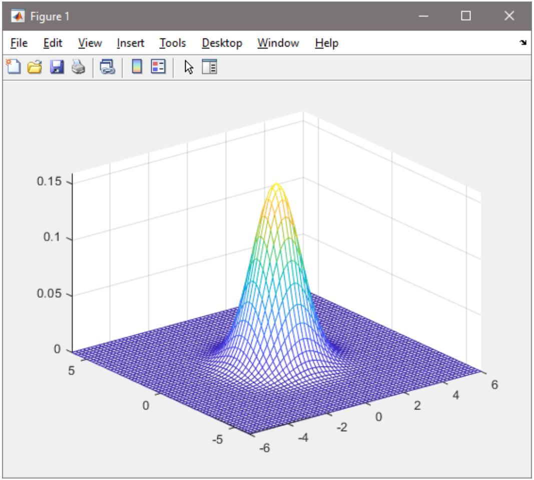
- the surface is defined by the Z - coordinates above the grid in the $x - y$ plane, using straight lines between adjacent points. Functions `mesh,surf` displays the surface in 3-D
- to display the function of two variables $z = f(x, y)$ it is necessary to generate X and Y matrices consisting of repeating rows and columns, over the domain of function (`meshgrid`) and calculate the function value
- Example: function $f(x, y) = \frac{1}{2\pi} exp\left[-\frac{1}{2}\left(x^2 + y^2\right)\right]$ at interval $< -6, 6 >$

```
>> [X Y]=meshgrid(-6:0.2:6);
>> Z=1/(2*pi)*exp(-(X.^2+Y.^2)/(2));
```

- command `mesh` - wireframe surface that color only the lines connecting the defining points

```
>> figure
>> mesh(X,Y,Z)
```

Three-dimensional graphics

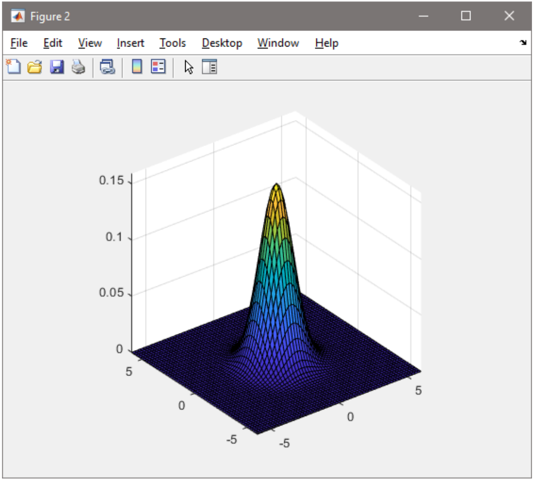


Three-dimensional graphics

- command `surf` - displays both the connecting lines and the faces of the surface in color (`colormap parula`)

```
>> figure
>> surf(X,Y,Z)
>> axis square
```

Three-dimensional graphics

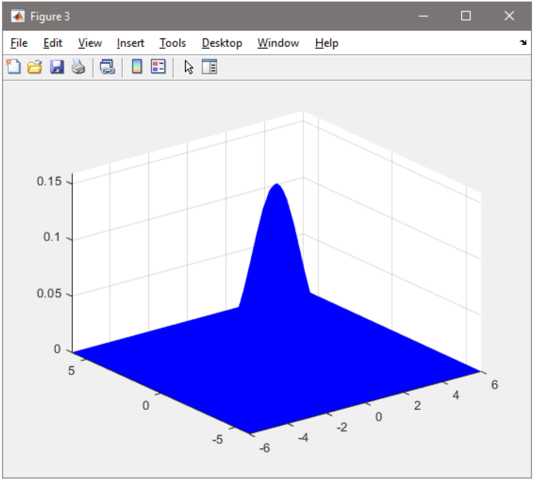


Three-dimensional graphics

- this example shows the same surface as the previous examples, but colors it blue and removes the mesh lines

```
>> figure
>> surf(X,Y,Z)
>> axis square
>> surf(X,Y,Z,'FaceColor','blue','EdgeColor','none')
```

Three-dimensional graphics

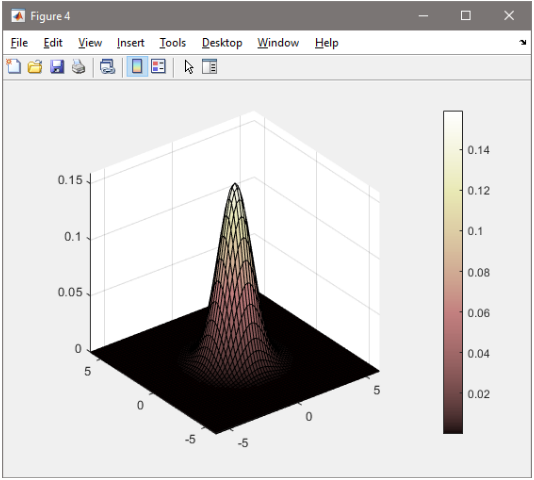


Three-dimensional graphics

- color scale setting, command `colormap`,
- predefined color scales, see help
- color bar display, command `colorbar`

```
>> figure
>> surf(X,Y,Z)
>> axis square
>> colormap pink
>> colorbar
>> doc colormap
```

Three-dimensional graphics

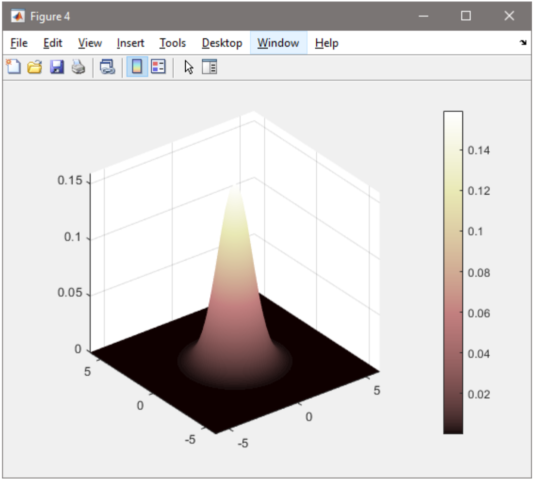


Three-dimensional graphics

- command `shading` controls color shading of the surface

```
>> shading interp
```

Three-dimensional graphics

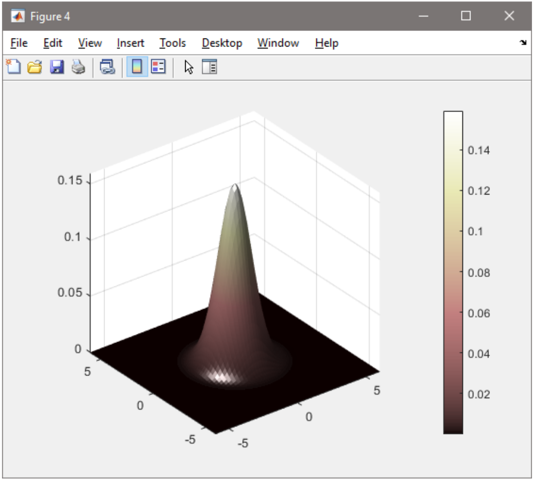


Three-dimensional graphics

- illumination of the surface by lights - the light object placed in the position (from the camera location)

```
>> camlight left
```

Three-dimensional graphics

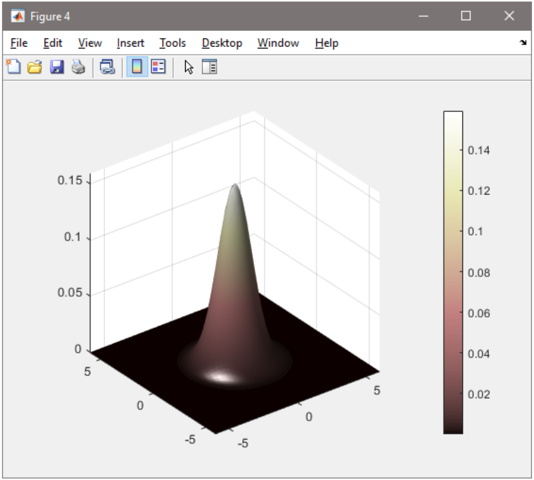


Three-dimensional graphics

- light object effects

```
>> lighting gouraud
```

Three-dimensional graphics

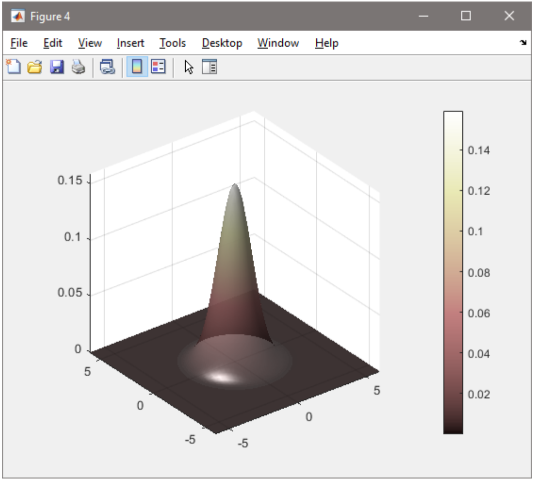


Three-dimensional graphics

- transparency (known as alpha value)

```
>> alpha(.8)
```

Three-dimensional graphics

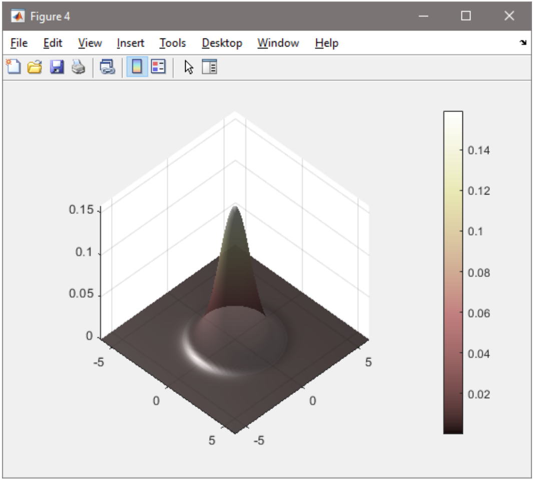


Three-dimensional graphics

- view settings, according to azimuth and elevation `view(az,el)`

```
>> view(45,45)
```

Three-dimensional graphics

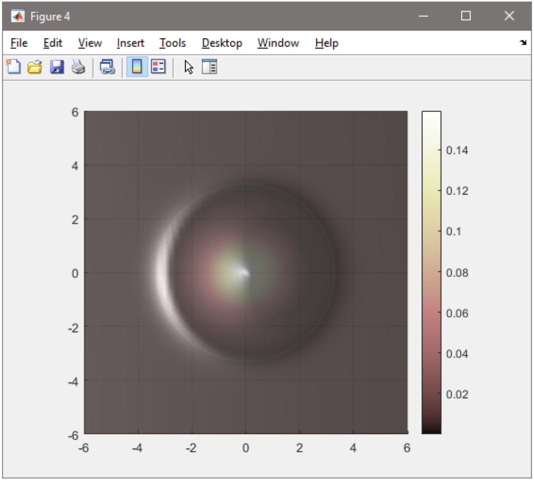


Three-dimensional graphics

- 2-D view (az=0,el=90)

```
>> view(2)
```

Three-dimensional graphics

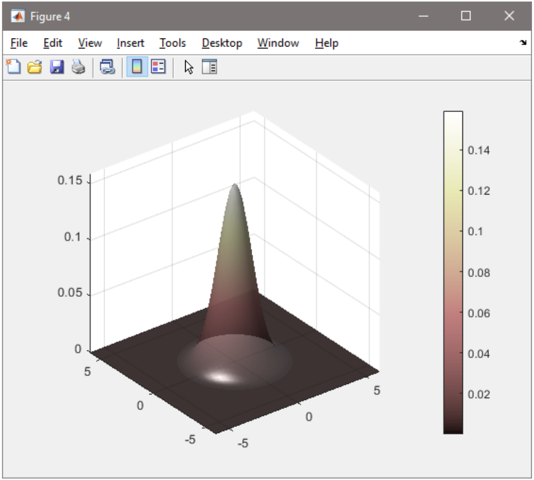


Three-dimensional graphics

- setting the 3-D default view (az=-37.5,el=30)

```
>> view(3)
```

Three-dimensional graphics

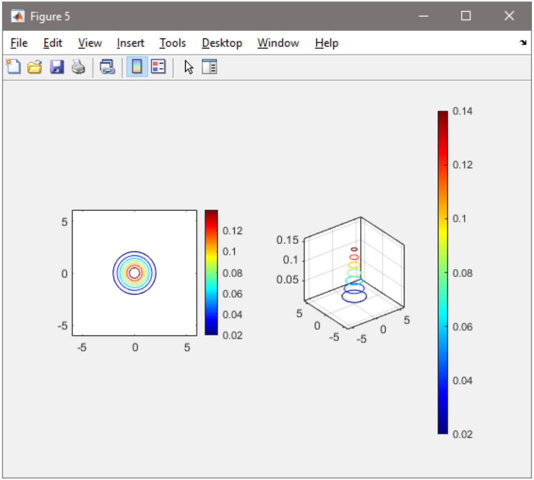


Three-dimensional graphics

- `contour` - contour 2-D graph
- `contour3` - contour 3-D graph

```
>> figure
>> subplot(1,2,1),contour(X,Y,Z)
>> axis square
>> colorbar
>> subplot(1,2,2),contour3(X,Y,Z)
>> axis square
>> colorbar
>> colormap jet
```

Three-dimensional graphics

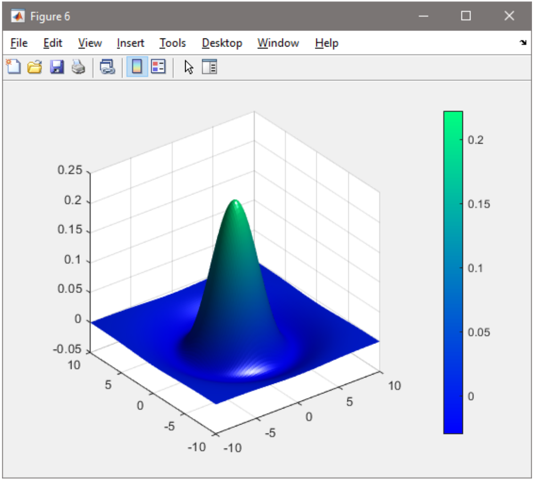


Three-dimensional graphics

- Example: Draw the LoG (Laplacian of Gaussian) function $LoG = \nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$ at interval $< -10, 10 >$ for $\sigma = 3$

```
>> [X Y]=meshgrid(-10:0.2:10);
>> sig=3;
>> Z=-((X.^2+Y.^2-2*sig^2)./(sig^4)).*exp(-(X.^2+Y.^2)/(2*sig^2));
>> figure
>> surf(X,Y,Z),
>> colormap winter
>> colorbar
>> axis square
>> shading interp
>> camlight
```

Three-dimensional graphics



Three-dimensional graphics

- Example: display surface of textile fabric (10×10 mm with division step 0.1 mm) captured by Talysurf contactless laser profilometer, where the profile is stored in the form of X, Y, Z coordinates in the text file *surface.txt*.
- import data from a text file

```
>> data=importdata('surface.txt');
```

- we use the values (X, Y) to define coordinates in the $x - y$ plane for which the value of Z is measured. Command `reshape` can restructure data so that triplets (X, Y, Z) form a rectangular grid (see MatLab Help: Representing Data as a Surface):

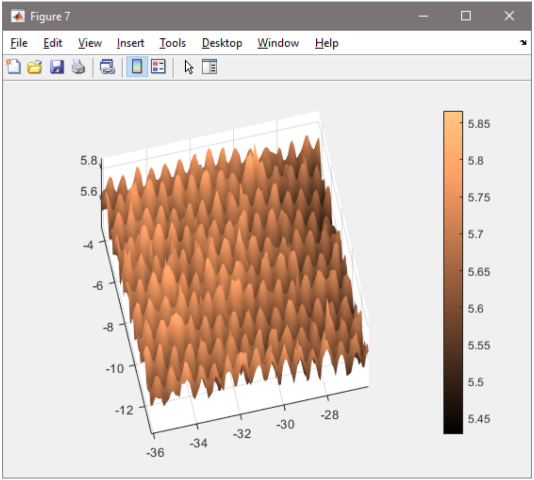
```
>> X = reshape(data(:,1),100,100);
>> Y = reshape(data(:,2),100,100);
>> Z = reshape(data(:,3),100,100);
```

Three-dimensional graphics

- display the surface of the fabric as in the previous cases

```
>> surf(X,Y,Z)
>> axis tight
>> axis square
>> shading interp
>> colormap copper
>> colorbar
>> view(-13,72)
```

Three-dimensional graphics



Three-dimensional graphics

Command	Description
» meshgrid	generates point coordinates
» plot3	lines and points in 3-D
» contour3	creates 3-D contour lines
» mesh	wireframe surface
» meshc	combination of mesh and contour functions
» meshz	creates a mesh in space including the zero plane
» surf	surface
» surfc	combination of surf and contour
» surfl	creating a surface with lighting

Three-dimensional graphics

Command	Description
» colorbar	colorbar scale
» colormap	colormap
» shading	smoothing edges
» view	defining a point of view on a chart in 3-D

Two-dimensional graphics

Examples for practice

Examples for practice

1. Draw the surface of the sphere with the center $[x_0 = 0, y_0 = 0, z_0 = 0]$ and the radius $r = 1$. Parametric expression of spherical surface is given by:

$$x = x_0 + r\ cos\phi\ sin\theta$$
$$y = y_0 + r\ sin\phi\ sin\theta$$
$$z = z_0 + r\ cos\theta$$

   for $0 < \phi \leqslant 2\pi,\ 0 \leqslant \theta \leqslant \pi$.

2. in the same figure draw another sphere with the radius $r = 2$ centred at

$$[x_0 = 1, y_0 = 1, z_0 = 1]$$

Solution