Principles of Technical Computing

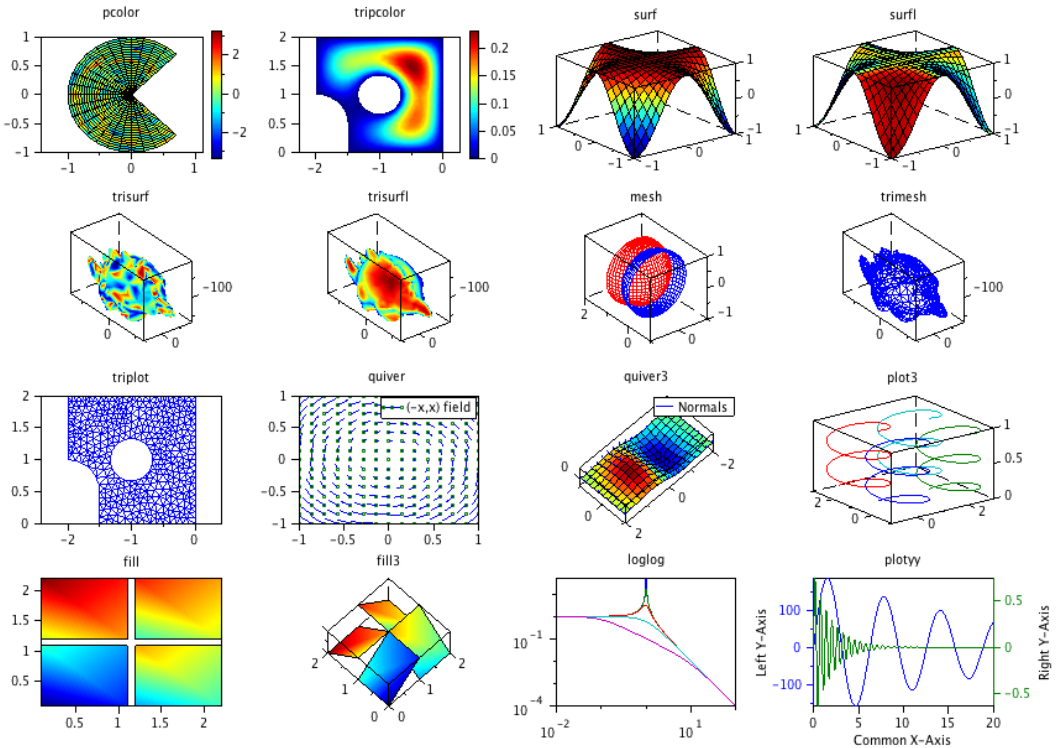
Lecture 3 – Data and Visualization in Matlab (2-D, 3-D)

Matylda Jabłońska-Sabuka

Lappeenranta University of Technology

Overview of Week 3

- Importing and exporting data
- Basic plotting (plot command)
- Axis labels and title, plot legend
- Multiple lines
- Multiple subplots



Importing and exporting data in Matlab

Manual import

Menu Home → Import Data

Import commands

- `importdata`
- `load`
- `xlsread`
- `csvread`

Exporting data

- `save`
- `xlswrite`
- `csvwrite`

Plotting (plot command) and plot properties in Matlab

The `plot` command

The `plot` command is used to create two-dimensional plots. The simplest form of the command is

```
x = -5:10; % values of the argument
y = x.^2 - 20; % values of the function

figure
plot(y)

figure
plot(x,y)

figure
plot(x,y,'k.-')
```

Plots only the function versus the index: 1,2,3...

Plots the function versus the argument: -5,-4,-3...

Plots the function versus the argument with a black line and a black dot marker

The `plot` command has additional optional arguments that can be used to specify the color and style of the line and the color and type of the markers, if any are desired. With these options the command has form

```
plot(x,y,'line specifiers',...
     'PropertyName','PropertyValue')
```

Line Style	Specifier
solid (default)	-
dashed	--
dotted	:
dash-dot	-.

The line color specifiers

Line Color	Specifier
red	r
green	g
blue	b
cyan	c
magenta	m
yellow	y
black	k
white	w

The marker type specifiers

Marker type	Specifier
plus sign	+
circle	o
asterisk	*
point	.
cross	x
triangle	^
diamond	d
square	s

PropertyName and PropertyValue

- `LineWidth` – a number in units of points (default 0.5)
- `MarkerSize` – a number in units of points
- `MarkerEdgeColor` – color specifiers from table above
- `MarkerFaceColor` – color specifiers from table above.

Plot of given data

year	1988	1989	1990	1991	1992	1993	1994
sales	8	12	20	22	18	24	27

```
>> yr=[1988:1994];
>> sle=[8 12 20 22 18 24 27];
>> plot(yr,sle,'--r*','LineWidth',2,'MarkerSize',12)
```

Plot of a given function

```
>> x=[-2:0.01:4];
>> y=3.5.^(-0.5*x).*cos(6*x);
>> plot(x,y)
```

What is wrong in these codes?

```
>> yr=1988:1994;
>> sle=[8 12 20 22 18 24 27];
>> plot(yr,sle(1:end-1),'--r*')

???  Error using ==> plot
Vectors must be the same lengths.
```

```
>> x=[-2*pi:0.01:2*pi];
>> y=4^(-2*x)*sin(3*x);
>> plot(x,y)

???  Error using ==> mpower
Matrix must be square.
```

What is wrong in this code?

```
>> yr=[1988:1994];
>> sle=[8 12 20 22 18 24 27];
>> plot(yr,sle,'--t')

???  Error using ==> plot
Error in color/linetype argument.

>> yr=[1988:1994];
>> sle=[8 12 20 22 18 24 27];
>> plot(yr,sle,'--r*','linewidth','markersize',12)

???  Error using ==> mpower
Matrix must be square.
```

Multiple graphs (lines) in the same plot in Matlab

Multiple graphs (lines) in the same plot

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
figure
plot(x,y,'-b',x,yd,'--r',x,ydd,':k')
```

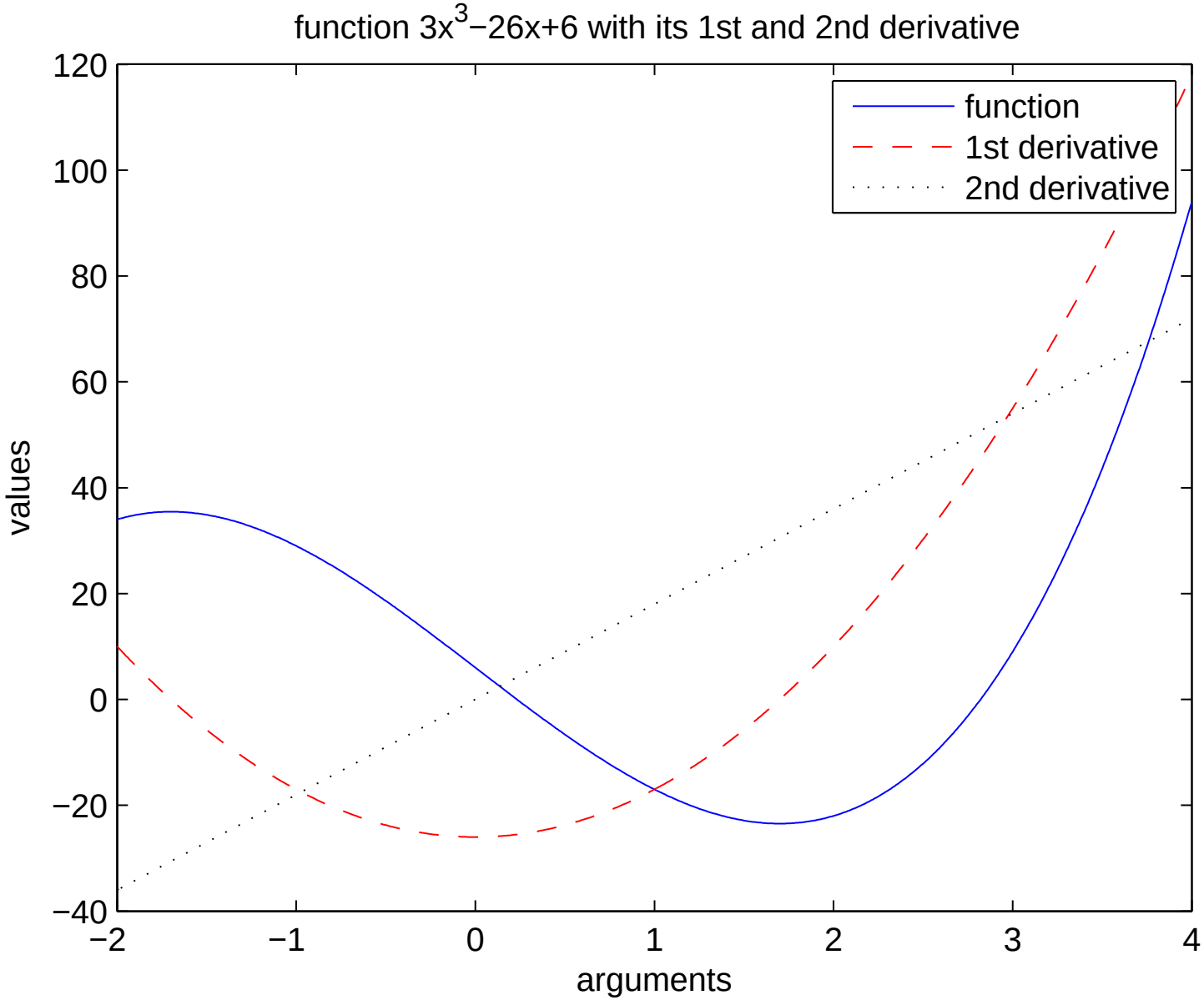
or alternatively

```
figure
plot(x,y,'-b'), hold on
plot(x,yd,'--r')
plot(x,ydd,':k'), hold off
```

Formatting plots (axes, title, legend) in Matlab

Formatting a plot using commands

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
figure
plot(x,y,'-b',x,yd,'--r',x,ydd,':k')
xlabel('arguments')
ylabel('values')
title('function 3x^3-26x+6 with its 1st ...
     and 2nd derivative')
legend('function','1st derivative','2nd derivative')
```


function $3x^3-26x+6$ with its 1st and 2nd derivative
function
1st derivative
2nd derivative
values
arguments
120
100
80
60
40
20
0
−20
−40
−2
−1
0
1
2
3
4

Text specification

- a single character can be displayed as a subscript (type _ before the character) or superscript (type ^ before the character)
- Greek characters are available (type e.g. `\alpha`, `\Omega`)
- the following properties are available:
  `Rotation`, `FontAngle`, `FontName`, `FontSize`, `FontWeight`, `Color`, `BackgroundColor`, `EdgeColor`, `LineWidth`

Handling axes

The `axis` command is used to change the range and the appearance of the axes.

- `axis([xmin,xmax,ymin,ymax])` – sets the limits of both the `x` and `y` axes (`xmin`, `xmax`, `ymin`, `ymax` are numbers)
- `axis equal` – sets the same scale for both axes
- `axis square` – sets the axes region to be square
- `tight` – sets the axes limits to the range of the data

The `grid on`, `grid off` commands add and remove grid lines from the plot.

More specialised plots in Matlab

More specialized plotting

- `semilogy(x,y)` – plots log(`y`) versus `x` with log base 10
- `semilogx(x,y)` – plots `y` versus log(`x`) with log base 10
- `loglog(x,y)` – plots log(`y`) versus log(`x`) with log base 10
- `errorbar` – plots data with error bars around data points
- `bar` – plots data in form of vertical bars
- `barh` – plots data in form of horizontal bars
- `stairs`, `stem`, `pie`, `hist`, `polar`

Exporting figures in Matlab

Saving figures

One of the ways to save a figure is from Figure window Menu File –> Export Setup. The setup allows modification of font or figure size.

The figure can be exported to a number of different formats: jpeg, png, eps, pdf, and more.

It also can be saved as Matlab Figure and, later, opened again without necessity of running the code again.

Saving can also be done automatically using commands `print` or `saveas`.

`print` *`option`* `myfigure.eps` (check Help for options)

3-dimensional plots in Matlab

3-D line plots

A three-dimensional plot is a line that is obtained by connecting points in a three-dimensional space. A basic 3-D plot is created with the `plot3` command of the form

```
plot3(x,y,z,'LineSpecifiers','PropertyName', ...
   PropertyValue)
```

The three vectors with the coordinates of the data points must have the same number of elements.

The line specifiers, properties, and property values are the same as in 2-D plots.

For example, if the coordinates x, y and z are given as a function of the parameter t by

$$x = \sqrt{t}\sin(2t), \quad y = \sqrt{t}\cos(2t), \quad z = 0.5t$$

A plot of the points for $0 \leq t \leq 6\pi$ can be produced

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
figure
plot3(x,y,z,'k','LineWidth',1)
grid on
xlabel('x'), ylabel('y'), zlabel('z')
```

Mesh plots

To make a surface or mesh plot, MATLAB needs not only `x` and `y` as vectors, but grid of points that these vectors create on the 2-D plane. A special function for that is `meshgrid`. An example use looks as follows

```
x=-1:3; y=1:4;
[X,Y]=meshgrid(x,y)
```

For the given mesh one can find the z function values. If, for example

$$z = \frac{xy^2}{x^2 + y^2}$$

then

```
Z = X.*Y.^2./(X.^2 + Y.^2)
mesh(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z')
```

Other 3-D built-in plots

- `surf(X,Y,Z)`
- `meshz(X,Y,Z)`
- `meshc(X,Y,Z)`
- `surfc(X,Y,Z)`
- `surfl(X,Y,Z)`
- `waterfall(X,Y,Z)`
- `contour3(X,Y,Z,n)`

3-D plots with special graphics

See example use from MATLAB Help

- `sphere`
- `cylinder`
- `bar3`
- `stem3`
- `scatter3`
- `pie3`

Multiple subplots within one Figure window in Matlab

Multiple subplots within one Figure window

You can display multiple plots in the same figure window and print them on the same piece of paper with the `subplot` function.

`subplot(m,n,i)` breaks the figure window into an *m*-by-*n* matrix of small subplots and selects the *i*the subplot for the current plot. The plots are numbered along the top row of the figure window, then the second row, and so forth.

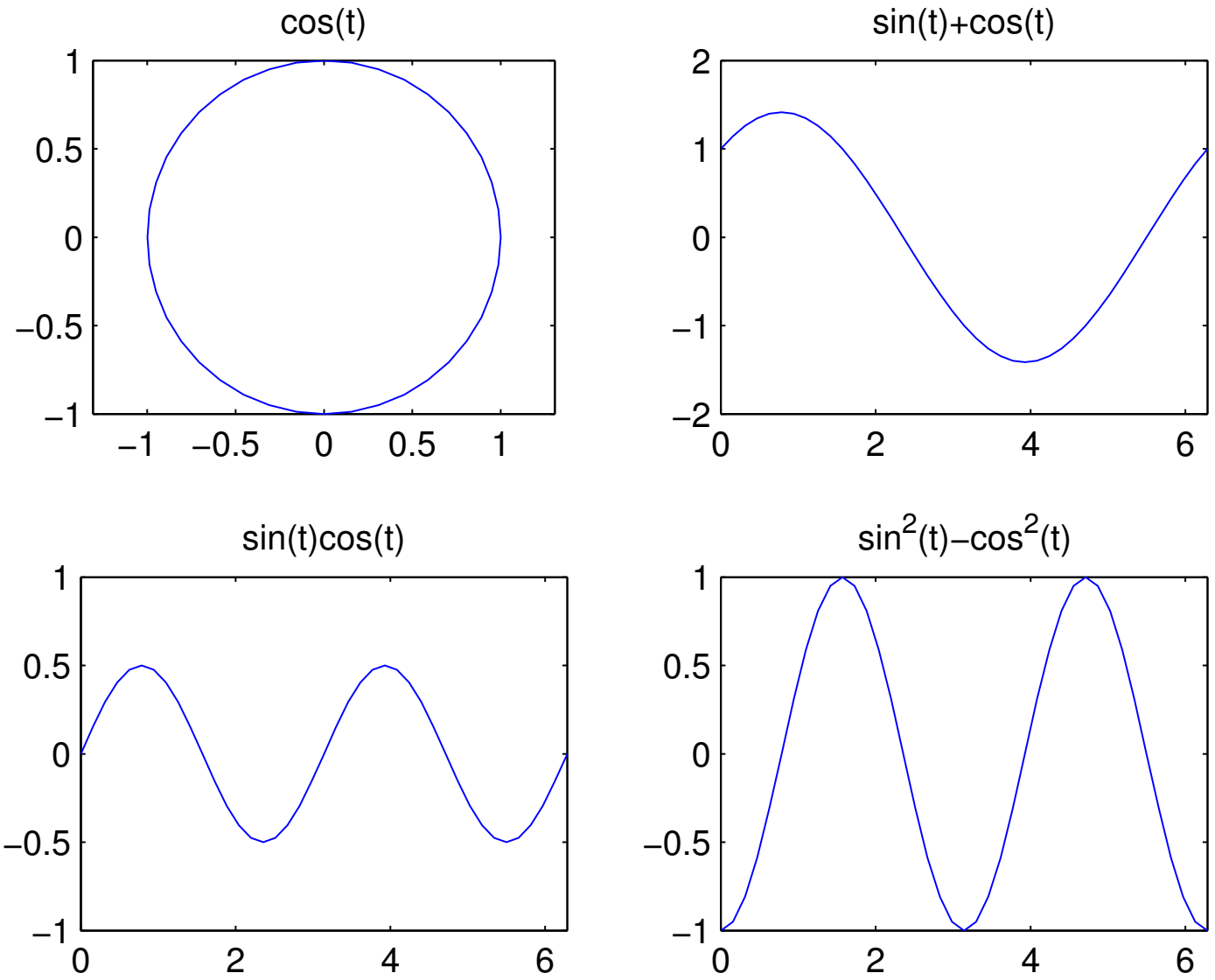
For each separate plot, the `subplot` command has to typed. After it, place the regular plot specification, including the `plot` command.

Example

Plot the following functions on interval $[0, 2\pi]$ in one figure with four different subplots:

- plot of $\cos(t)$ against $\sin(t)$
- plot of $\sin(t) + \cos(t)$
- plot of $\sin(t)\cos(t)$
- plot of $\sin^2(t) - \cos^2(t)$

```
t = 0:pi/20:2*pi;
subplot(2,2,1)
plot(sin(t),cos(t))
title('cos(t)')
axis equal
subplot(2,2,2)
z = sin(t)+cos(t);
plot(t,z)
title('sin(t)+cos(t)')
axis([0 2*pi -2 2])
subplot(2,2,3)
z = sin(t).*cos(t);
plot(t,z)
title('sin(t)cos(t)')
axis([0 2*pi -1 1])
subplot(2,2,4)
z = (sin(t).^2)-(cos(t).^2);
plot(t,z)
title('sin^2(t)-cos^2(t)')
axis([0 2*pi -1 1])
```


cos(t)
1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1
sin(t)+cos(t)
2
1
0
−1
−2
0
2
4
6
sin(t)cos(t)
1
0.5
0
−0.5
−1
0
2
4
6
sin2(t)−cos2(t)
1
0.5
0
−0.5
−1
0
2
4
6

The subplot index can also be ruled by loops.

```
n = 10; m = 1000;
x = rand(n,m);
figure
for k = 1:n
subplot(2,5,k)
plot(1:m,x(k,:),'.'), axis([1 m 0 1]);
end
```

or with "automatized" indices

```
subplot(ceil(n/2),2,k)
```

(here 2 can also be any other natural number)