Exercise set 0

Norwegian University of Science and Technology
Department of Mathematics

This set of exercises is meant to give a short introduction into the usage of MATLAB.

1 **Linear algebra and plotting:**

Find and plot the polynomial of degree 3 that interpolates the points given in the following table:

i	1	2	3	4
x_i	-2	0	1	3
y_i	-16	-3	-1	24

In other words: Find a polynomial

$$p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

that satisfies $p(x_i) = y_i$ for $i = 1,\ 2,\ 3,\ 4$.

a) Verify that the coefficients satisfy the linear system

$$\begin{pmatrix} 1 & -2 & 4 & -8 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 3 & 9 & 27 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} -16 \\ -3 \\ -1 \\ 24 \end{pmatrix}.$$

b) Use MATLAB to solve the linear system.

c) Use MATLAB for plotting the interpolation polynomial.

MATLAB resources

MATLAB is already installed at certain computer labs at NTNU. There is also the (highly recommended) possibility of installing it on your own computer. Instructions for that can be found at `https://innsida.ntnu.no/it-hjelp`.

A short introduction in MATLAB (in Norwegian) can be found at `https://wiki.math.ntnu.no/drift/stud/matlab` (the pdf-file *Introduksjonskurs for IMF-studenter i matlab*). The documentation in MATLAB itself is very extensive. It can be accessed either from the start menu of by typing `doc` in the command line. In addition, there is an introduction at `http://www.mathworks.se/moler/chapters.html`.

Plotting

Simple plots can be created with the command `plot`. It is, however, recommended to improve the appearance of the plot by playing around with variables and commands like:

`'Color'`	color of the graph plotted
`'LineWidth'`	line width of the graph plotted
`grid on/off`	plot a grid over the function plot
`xlabel('x')`	annotation of the x-axis
`ylabel('y')`	annotation of the y-axis
`title('z')`	title of the plot

$\boxed{2}$ **Some simple programming:**

Euler's number e can, for instance, be computed using either of the formulas

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n$$

or

$$e = \sum_{k=0}^{\infty} \frac{1}{k!}.$$

a) Write two MATLAB-programs that compute the numbers

$$a_n = \left(1 + \frac{1}{n}\right)^n$$

and

$$b_m = \sum_{k=0}^{m} \frac{1}{k!}$$

for different values of n and m and compare the results with the true value of e.

b) One of the two methods does not seem to converge to e. Which one? Why?

Programming in MATLAB

MATLAB-files have the ending `.m` and may either be scripts (which simply contain several lines of MATLAB-code that are executed when the file is called) or functions (depending additionally on input variables and usually returning some result).

A simple MATLAB-function for computing the square of a number (or the pointwise square of an array) may consist of the following lines:

```
function y = f(x)

y = x.^2;
```

This code has to be stored in a file `f.m` (the file should have the same name as the function). Then, calling for instance `f(2)` in the command line of MATLAB gives the result `4`.

$\boxed{3}$ Consider the following two segments of pseudocode:

Program A:

Data: a vector $a = [a_0, a_1, \dots, a_n]$ of real numbers, a real number x;
Output: a real number y;
Initialization: $y \leftarrow a_0$;
for $k = 1$ **to** n **do**
  $y \leftarrow y + a_k x^k$;
end

Program B:

Data: a vector $a = [a_0, a_1, \dots, a_n]$ of real numbers, a real number x;
Output: a real number y;
Initialization: $y \leftarrow a_n$;
for $k = n - 1$ **to** 0 **by** -1 **do**
  $y \leftarrow a_k + xy$;
end

a) What do these programs actually do?

b) In theory, both programs should yield the same result. Can they be expected to do so also numerically?

c) Which of the programs is usually preferable?