OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

Math review and MATLAB excersises

HiOA ELTS2300 Dynamic systems

Introduction to MATLAB

1) Define variable x with value $x = \frac{\pi}{2}$. Compute the result of $y = x + \sin(x) + \frac{1}{2}x^2$ and store it in a variable y.
2) Compute $z = y^3 + 2.5\cos(y)$
3) Make MATLAB functions that compute y and z.
4) Manually define the column vector
$$x = \begin{bmatrix} 3 \\ 2 \\ -1.5 \end{bmatrix} \in \mathbb{R}^{3\times 1}$$
5) Manually define the row vector
$$b = [0 \quad 1 \quad 5] \in \mathbb{R}^{1\times 3}$$
6) Manually define the matrices
$$A = \begin{bmatrix} 1 & 0 & 0.2 \\ 8 & -3 & 3 \\ -2 & 0.4 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}, B = \begin{bmatrix} 1 & 2 \\ 4 & -1 \\ 2 & 3 \end{bmatrix} \in \mathbb{R}^{3\times 2}, C = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

Compute the following multiplications by hand and verify afterwards the result with MATLAB:

a. $x^T x$
b. $x^T b$
c. xb
d. bx
e. Ax
f. bAx
g. $x^T A b^T$
h. AB
i. $B\hat{}Tx$
j. BC
k. ABC

7) Manually define the matrices
$$A = \begin{bmatrix} 1 & 0 & 1 \\ 5 & -3 & 2 \\ -2 & 4 & 3 \end{bmatrix} \in \mathbb{R}^{3\times 3}, B = \begin{bmatrix} 2 & 5 & 1 \\ -3 & 1 & 1 \\ -1 & 2 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}, x = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \in \mathbb{R}^{3\times 1}$$

Compute the following operations by hand and verify afterwards the result with MATLAB:

a. $A^T A$
b. AA^T
c. AB
d. BA
e. Ax

f. $x^T A$
g. $x^T A x$

8) Random vectors and plots
   a. Create a random column vector with dimension 100, according to a Normal distribution, with zero mean and standard deviation 1. Hint: Type "`help randn`" in MATLAB. Plot the result. Experiment with different colors, line styles, and markers. **Hint**: Type "`help plot`" in MATLAB. Use `xlabel` "sample"and `ylabel` "value" in the figure. (Hint: "`help xlabel`", "`help ylabel`")

   b. Create another random column vector with dimension 100, according to a uniform distribution, mean 5 and standard deviation 2. Hint: Type "`help rand`" in MATLAB. Plot the result.

   c. Plot both vectors in the same figure and with different colors/markers. **Hint**: use "`hold on/off`" commands. For instance "`plot(x1,'.-b');hold on;plot(x2,'or');hold off;grid on`" Use legend command to label each of the plots. Hint: use "`help legend`".

9) Plot the signal

$$y = 2\sin(\omega t + 1)$$

Between time $t = 0s$ and $t = 10s$, with $\omega = 2$.
Hint: Decide on how many samples you need. Create a time vector using "`t=0:h:10`", where `h` is the sampling period. Label x-axis with "Time [s]" and y-axis with "y(t)". Experiment what happens when changing the values of w and h. One option is to use a for loop to create the signal but it is much simpler and efficient to make an operation on the time vector t.

10) Plot the signal

$$y = 2x^3 - x^2 + x - 1$$

In the interval $x \in [-20,20]$. Hint: Writing a dot after a vector means the operation applies to the vector elements. For instance "`x.^2`" takes the square of all elements in vector x; and "`x.^3`" its cubic exponent. Also if x and y are two vectors of the same dimensions, "`x.*y`" indicates the element-wise product of the elements in vectors x and y.

Linear algebra

11) Use MATLAB to solve the following linear system of equations and determine the values of a, b, c:

$$2a + b - 2c = 1$$

$$a + 4.5b + 5c = 4$$

$$-2a + 3b - 8c = -1$$

Hint: Write the system in matrix form and use matrix inverse MATLAB function "inv()".

12) Use MATLAB to solve the following linear system of equations and determine the value of x

$$Ax = b$$

$$A = \begin{bmatrix} 1 & 2 & 6.5 & -2 \\ 0 & 2.3 & 3 & 0 \\ 3.2 & 0 & 3.5 & 7 \\ -2 & 2 & 1.25 & 9 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ -2 \\ 1 \\ 4 \end{bmatrix}$$

13) **Least Squares polynomial fitting**. In this exercise you will use least squares to determine the coefficients of a polynomial of third order that best approximates a number of samples. Suppose that you were given N pairs of points $(y_i, x_i); i \in \{1, \dots, N\}$ and you want to find coefficients of a third order polynomial that best approximate those points. The goal is to find the coefficients (a_3, a_2, a_1, a_0) that minimizes the size of errors w_i in:

$$a_3 x_i^3 + a_2 x_i^2 + a_1 x_i + a_0 = y_i + w_i$$

That is equivalent to minimizing the sum of squares of the errors

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(a_3 x_i^3 + a_2 x_i^2 + a_1 x_i + a_0 - y_i\right)^2 = \frac{1}{N}\sum_{i=1}^{N} w_i^2$$

The individual equations for each sample point can be written as

$$\begin{gathered} a_3 x_1^3 + a_2 x_1^2 + a_1 x_1 + a_0 = y_1 + w_1 \\ a_3 x_2^3 + a_2 x_2^2 + a_1 x_2 + a_0 = y_2 + w_2 \\ \vdots \\ a_3 x_N^3 + a_2 x_N^2 + a_1 x_N + a_0 = y_N + w_N \end{gathered}$$

And in matrix form:

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N^3 & x_N^2 & x_N & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

which has the form of a linear system of equations:

$$A\theta = y + w$$

Where

$$A = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N^3 & x_N^2 & x_N & 1 \end{bmatrix} \in \mathbb{R}^{N\times 4}$$

$$\theta = \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \in \mathbb{R}^{4\times 1}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} \in \mathbb{R}^{N\times 1}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^{N\times 1}$$

The least squares solution to the polynomial fitting can be found by attempting to solve the linear system

$$A\theta = y$$

Since A is in general not square (usually we have many more samples N than the order of the polynomial), we can not take its inverse. Instead the Least Squares solution uses something called the pseudoinverse. If the matrix A^TA is invertible (non zero determinant) then the pseudoinverse is defined as $A^+ = (A^TA)^{-1}A^T$. The Least Squares solution is then:

$$\theta = A^+y = (A^TA)^{-1}A^Ty$$

This expression can also be found by trying to isolate θ from the expression $A\theta = y$ by first multiplying both sides from the left by A^T, and then from the left by $(A^TA)^{-1}$. The expression now follows using the fact that $(A^TA)^{-1}A^TA = I$ is the identity matrix:

$$A\theta = y$$
$$A^TA\theta = A^Ty$$
$$\underbrace{(A^TA)^{-1}(A^TA)}_{I}\theta = (A^TA)^{-1}A^Ty$$

$$\theta = \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = (A^TA)^{-1}A^Ty$$

a. Generate $N = 50$ uniform samples of a signal

$$y = 0.4x^3 + x^2 - 1.5x + 1 + w$$

Where $x \in [-1,1]$ and w is a normally distributed random variable with zero mean and standard deviation $\sigma = 0.5$. Make a figure to plot the signal. **Hints**: use "`x=linspace(-1,1,50)'`" and "`0.5*randn(50,1)`" to generate noise vector. In order to plot you can use marker "o" which uses circles to indicate samples. For instance "`plot(x,y,'ro')`".

b. Use least squares to determine the coefficients (a_3, a_2, a_1, a_0) that best approximate the generated points. Plot the results.

c. Use the same data samples but now try to fit a second order polynomial

$$a_2 x_i^2 + a_1 x_i + a_0 = y_i + w_i$$

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Plot the results and compare them with the third order polynomial fit. Is there a big difference? Hint: Note that the process is similar but now with different matrix $A \in \mathbb{R}^{N\times 3}$ instead of $\mathbb{R}^{N\times 4}$. Use "`hold on/ hold off`" to overlay plots.

d. Experiment with different values of coefficients, noise standard deviation, and number of samples.

e. [Extra, if time allows] Repeat the process with a fourth order polynomial. How does this compare to previous second and third order polynomial fittings?

OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

Maps and functions

14) Write MATLAB functions that represent the following mappings
    a. $f: \mathbb{R}^2 \to \mathbb{R}; f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = x_1^2 + x_2^2 + 1$
    b. $f: \mathbb{R}^2 \to \mathbb{R}^2; f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 - x_2 + 1 \\ \sin x_2 \end{bmatrix}$
    c. $f: \mathbb{R} \to \mathbb{R}^3; f(x) = \begin{bmatrix} x \\ x^2 \\ x^2 + 1 - \frac{1}{x} \end{bmatrix}$
    d. Use matlab “for” loop to implement the function $f: \mathbb{R}^n \to \mathbb{R}; f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \sqrt[2]{x_1^2 + \cdots + x_n^2}$.
    e. Implement the same function but using the fact that if we define column vector $\boldsymbol{x} = [x_1 \quad \cdots \quad x_n]^T$ then $x_1^2 + \cdots + x_n^2 = \boldsymbol{x}^T\boldsymbol{x} = \|\boldsymbol{x}\|^2$. Hint: use «sqrt» function. Note: MATLAB has a built-in function «norm» which does the same but this is not to be used in the excersise.
15) **Map into the unit sphere**. The unit sphere is defined as the set of points in $\mathbb{R}^3$ that have unit length

$$\mathbb{S}^2 = \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\| = 1 \ \} = \left\{\begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 : \sqrt[2]{x^2 + y^2 + z^2} = 1 \ \right\}$$

Write a function that implements the map into the unit sphere $f: \mathbb{R}^3 \to \mathbb{S}^2$ where:

$$f(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} = \frac{1}{\sqrt[2]{\boldsymbol{x}^T\boldsymbol{x}}}\boldsymbol{x}$$

$$f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \frac{1}{\sqrt[2]{x^2 + y^2 + z^2}}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

16) Generate 100 random points in $\mathbb{R}^3$ with zero mean and standard deviation $\sigma = 2$. Plot the original points and their result after being passed through the unit sphere map of the previous excersice. Hint: You can use a for loop to iterate over the 100 samples, or attempt the excersice in a matrix form. A normally distributed random vector in $\mathbb{R}^3$ with entries according to a standard deviation $\sigma = 2$ can be generated with «`2*randn(3,1)`». Yu can generate 100 samples in one call by using «`2*randn(100,3)`». This is a matrix where each row represents a point, first column contains x-coordinates, the second y-coordinates, etc. Use «plot3» matlab function. Use different colors for $\boldsymbol{x}$ and $f(\boldsymbol{x})$.

Calculus

17) **Derivative of a function**. Given a signal

$$y(t) = 2\sin(\omega t + 1)$$

a. Plot the signal in the interval $t \in [0,10]$. **Hint**: Decide on how many samples you need. Create a time vector using "`t=0:h:10`", where `h` is the sampling period. Alternatively you can use "`t=linspace(0,10,N)`", where N is the number of samples. (In this case the sampling time becomes $h = \frac{10}{(N-1)}$)
b. Determine its derivative analytically using a table of derivatives. Plot the analytical derivative of the function in the same time interval.

The derivative of a function is defined as

$$y'(t) = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h}$$

If you defined a vector in the previous point (a)

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} y(0) \\ y(h) \\ y(2h) \\ \vdots \\ y((N-1)h) \end{bmatrix} \in \mathbb{R}^{N \times 1}$$

One can attempt to estimate the derivative by assuming that h is small enough and computing the difference between samples divided by the elapsed time between them:

$$\boldsymbol{y}' = \begin{bmatrix} y'(0) \\ y'(h) \\ y'(2h) \\ \vdots \\ y'((N-1)h) \end{bmatrix} = \begin{bmatrix} \frac{y_2 - y_1}{h} \\ \frac{y_3 - y_2}{h} \\ \frac{y_4 - y_3}{h} \\ \vdots \\ \frac{y_N - y_{N-1}}{h} \end{bmatrix} = \begin{bmatrix} \frac{y(h) - y(0)}{h} \\ \frac{y(2h) - y(h)}{h} \\ \frac{y(3h) - y(2h)}{h} \\ \vdots \\ \frac{y(Nh) - y\left((N-1)h\right)}{h} \end{bmatrix} \in \mathbb{R}^{N-1 \times 1}$$

Note that by doing this, if the original signal vector has dimension $\mathbb{R}^{N \times 1}$, the estimated derivative has one sample less, that is dimension $\mathbb{R}^{N-1 \times 1}$. This is called the «dirty derivative» og signal $y(t)$.

a. Compute the dirty derivative of the signal in (a) and plot it together with the analytical derivative of the function you found in (b).
b. Experiment with different values of sample time h. Does it have any effect? Can you guess why it is called dirty derivative?
c. [Extra, if time allows]. Now assume that we measure the signal with a certain measurement error w which is normally distributed, has zero mean, and standard deviation $\sigma = 0.1$

$$y(t) = 2\sin(\omega t + 1) + w$$

Plot the dirty derivative of the signal. Experiment with different values of sampling time h and noise standard deviation $\sigma = 0.1$. What is the effect of the measurement noise on the derivative estimate?

Complex numbers

18) Given the following complex numbers

$$a = 1i + 1$$

$$b = 2i + 3$$

$$c = -i - 2$$

$$d = 3i$$

Compute the following expressions by hand and verify the result in MATLAB:

a. $a + b$
b. $a - c$
c. ab
d. abc
e. $\frac{a}{c}$
f. $\frac{d}{b-c}$
g. $\frac{a+b}{c+d}$
h. $a\left(\frac{b}{d} + 1\right)d$