Assignment 3

➽ **Read the assignment carefully! Remember that the first line of a script must be the call to the script preamble.**

A. Mandatory

Write a MATLAB script Assignment03A_IDxx.m[1] that

- reads the matrices A, B from the file Assignment03.mat with the function call load('Assignment03.mat');
- sets the variable e=1E-16;
- computes and displays $\sin A, \cos A, \sin B, \cos B$, where $\sin X$ $(\cos X)$ is the **matrix sine (cosine)** of X returned by the local function matsincos that you have to write (for the interface see below);
- shows that up to rounding errors

$$\begin{aligned} AB &= BA, & (1)\\ (\sin A)^2 + (\cos A)^2 &= I, & (2)\\ \sin(A+B) &= \sin A \cos B + \cos A \sin B, & (3)\\ \cos(A+B) &= \cos A \cos B - \sin A \sin B, & (4)\\ \exp(iA) &= \cos(A) + i\sin(A), & (5) \end{aligned}$$

  where I is the identity matrix, i is the imaginary unit, and $\exp(iA)$ is the *matrix exponential* (function expm).

The (approximate) validity of the equations (1)–(5) should be shown by using the local function iszero from example_scripts_2.m.

Write a **local** function matsincos that computes the matrix sine and the matrix cosine of a square numerical matrix.[2] The interface of the local function looks like this:

```
function [sinX,cosX]=matsincos(X,e) % local function
% Input   X: a square matrix
%         e: only terms with 2-norm > e contribute to the Taylor series
% Output  sinX: matrix sine of X
%         cosX: matrix cosine of X
```

Continued on page 2!

[1] xx is your two-digit ID number number

[2] See https://en.wikipedia.org/wiki/Trigonometric_functions_of_matrices

The function should check that there is at least one input argument X that is a square numerical matrix and issue a warning if it is not, in which case the output matrices should be empty. If there is only one input argument, e should be set to 10^{-14}. A single loop is permitted in the code of msincos. Please keep in mind that the terms of the Taylor series can be computed recursively. *(4 pt)*

B. Optional

Write a MATLAB script Assignment03B_IDxx.m[3] that computes the ordered list of all divisors of a natural number n (including 1 and n) directly from the prime factorization of n. At most one loop is allowed; if you find a solution without loops you earn an additional point. The code must work for arbitrary n. Test your code with $n = 945743994$ and compare your result with the one from the function divisors. *(2 pt)*

Useful functions: factor, perms, combnk, cumprod, unique, ...

Please make sure that the relevant results and *only* those are shown in the output to the command window, so that I can check the correctness quickly and without digging into the code.

Submit the script(s) until 5pm on the day before the next course, i.e. April 14, 2021. Any violation of the naming convention will lead to the rejection of the submission!

[3] xx is your two-digit ID number number