# EPNM, Summer semester 2023/24, Project 2.7 for 20 points

1. Create your matrix in Matlab using the following instructions:
   n=1000;
   c7 = [ 0.1*ones(5000,n); 0.2*ones(5000,n); 0.3*ones(5000,n); 0.4*ones(5000,n); 0.3*ones(5070,n)];

2. Write Matlab m-file, which will compute the product

$$\sin(\ln(c7 + 2.7))^T \cdot \ln(\cos(c7) + 1.7)$$

   where functions sine, cosine and natural logarithm (in Matlab `log` function) as well as the addition operator concern element-wise operations, in several versions:

   (a) using the default Matlab matrix operations (with built-in multithreading),

   (b) using the default Matlab matrix operations without built-in multithreading, that is for the single core; this is obtained when you precede the instruction from p. 2a by the instruction:
   ```
   MNCT=maxNumCompThreads(1)
   ```
   The variable `MNCT` will remember the default Matlab number of threads used in matrix operations.

   (c) fully sequential using the `for` loop for one thread,

   (d) fully sequential using the `for` loop for the default number of threads; this will be obtained when you have just started Matlab or after the operation in the point 2b by the instruction:
   ```
   maxNumCompThreads(MNCT)
   ```
   (in the latter case Matlab should answer with "ans = 1")

   (e) start the Parallel Computing Toolbox pool of workers with `parpool`, change the external `for` in the program from p. 2c to `parfor` and repeat the calculations,

   (f) partitioning properly indices (optionally you may also use Matlab cell arrays) split two matrices into $p$ and $p/2$ possibly equal blocks (you should obtain block vectors), where $p$ is the number of cores in your PC (that is `MNCT`); then create, respectively, $p$ and $p/2$ threads using the `parfor` loop in two variants:
   f1) with the matrix operators within each process
   f2) with `for` loops within each process,
   You may change the number of workers by shutting down the parallel pool (click on the four vertical bars icon in the bottom left corner) and writing: e.g., `parpool('local',2)`.

   (g) repeat point 2f using for splitting distributed arrays from Parallel Computing Toolbox of Matlab (function `distributed`).

3. Compare solutions (using Frobenius norm /function `norm`/ to the difference of matrices from points 2a, 2c-2g with the matrix from point 2b) and execution times of all versions. Try to explain the observed differences in the execution times.

4. Write the instruction: `c7=c7(:,1:50);`
   and repeat points 2-3 for the product

$$\sin(\ln(c7 + 2.7)) \cdot \ln(\cos(c7) + 1.7)^T$$

5. Write the report on your work and results (times should be from 2nd or 3rd run of the program).

6. Save the report and your Matlab files compressed to a ZIP file on the Studia server, in module Reports, under the name **"Project-2_7.zip"**.

Project approval:

6 May, 4:15 pm, Room P113