Parallel Programming WS15 **HOMEWORK #5**

Submit by email to: `fabregat@aices.rwth-aachen.de`
Due by: Friday, November 27th, 5pm.

MPI – Part 3

1. In this task, you will play with different modes of point-to-point communication. The computational problem to be solved is the following:

   Matrix $A \in \mathbb{R}^{n \times n}$ is partitioned and distributed over the four processors p_0, p_1, p_2, p_3 as follows:

   $$A = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c} p_0 & p_1 \\ \hline p_2 & p_3 \end{array}\right), \text{ with } A_{TL} \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}.$$

   Matrices $B \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{n \times n}$ are defined as

   $$B = \left(\begin{array}{c|c} B_{TL} & B_{TR} \\ \hline B_{BL} & B_{BR} \end{array}\right) := \left(\begin{array}{c|c} A_{TL} + A_{TR} & A_{TL} - A_{TR} \\ \hline A_{BL} + A_{BR} & A_{BL} - A_{BR} \end{array}\right)$$

   and

   $$C = \left(\begin{array}{c|c} C_{TL} & C_{TR} \\ \hline C_{BL} & C_{BR} \end{array}\right) := \left(\begin{array}{c|c} B_{TL} + B_{BL} & B_{TR} + B_{BR} \\ \hline B_{TL} - B_{BL} & B_{TR} - B_{BR} \end{array}\right).$$

   Each processor has enough local memory to store exactly two matrices of size $\frac{n}{2} \times \frac{n}{2}$. The goal is to compute the (distributed) matrix C.

   Your task is to write a program that computes the operation above. The program takes one command-line argument `n`, the size of the matrix (you can assume it is a multiple of 2). Each of the 4 processes initializes its own $\frac{n}{2} \times \frac{n}{2}$ chunk of the matrix A, and performs the necessary communication and computation. You may need to explicitly synchronize the processes with `MPI_Barrier`. If you do, minimize the number of barriers used. You can initialize the matrix in any way you want; we suggest you do it in a way that you can easily check the result.

   Implement multiple versions of the program, each time using one of the following communication modes:

   a) 2-sided communication based on `MPI_Ssend` and `MPI_Recv`;

   b) 2-sided communication based on `MPI_Isend` and `MPI_Recv`;

   c) 2-sided communication based on `MPI_Send` and `MPI_Irecv`.

2. The goal of this task is to optimize for memory usage. We ask you to solve the same problem as in Task 1, this time with a restriction in the amount of memory available per process. Now, each process can only store one submatrix of size $\frac{n}{2} \times \frac{n}{2}$ plus one vector of size $\frac{n}{2}$. Implement one single version using the communication mode of your choice.