Parallel algorithms and programs
TP 4: 2D LU factorization

Jean-Marie Madiot, Mohamed Sid-Lakhdar
jeanmarie.madiot@ens-lyon.fr,
mohamed.sid_lakhdar@ens-lyon.fr

In Linear Algebra, the resolution of linear systems is an important issue. One well known method to solve this problem is the LU factorization. The goal of this TP is the implementation of the LU method in a distributed-memory environment using a 2-Dimension bloc scheem. The figure below represents one step of the this algorithm. This step is first applied on the entire matrix, and then repeated recursively on the consecutive trailing submatries (in red).

Exercise 1

For a matrix size $n = 2^a$ and for a number of processors $p = 2^b$, find the most appropriate splitting of the initial matrix into square submatrices of equal size, such that there is a one-to-one mapping between these submatrices and the set of processors. Then, implement a routine that, for each MPI processor, given n and p, initialize its arrays of MPI communicators *comm_row* and *comm_col* where each element of the arrays is a communicator that gathers the processors on the same row and same column respectively at each step of the LU factorization.

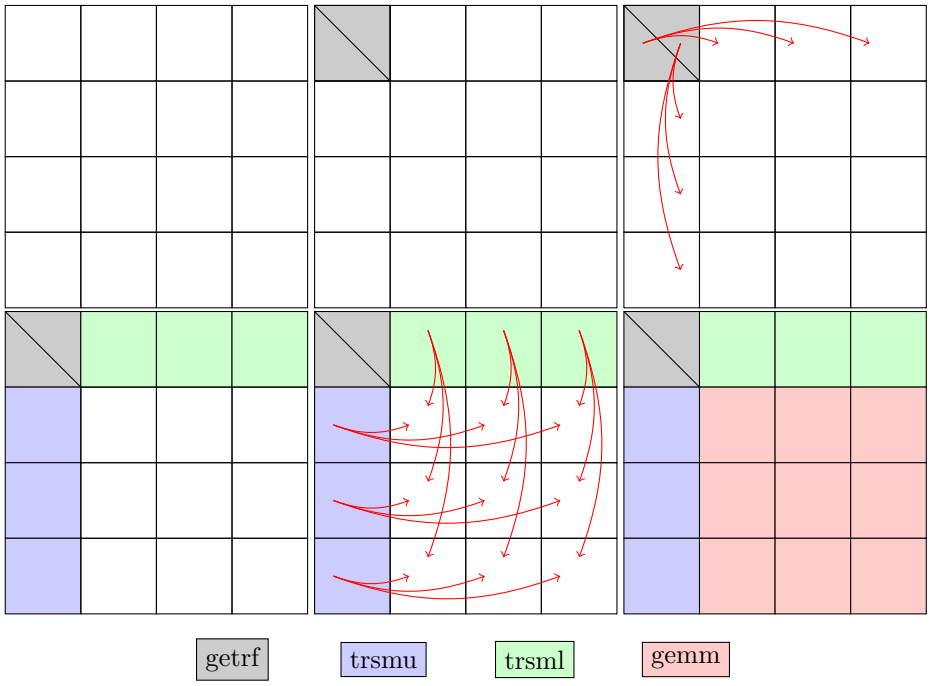
Exercise 2

Implement the LU algorithm in MPI, using the previous matrix and processor configuration. You can use (you are very encouraged to do so) the matrix structure and the $getrf$, $trsml$, $trsmu$ and $gemm$ functions given in the file *LU.c*. In a first time, you can use the collective function *bcast* to transmit the submatrices from one processor to all the remaining in the column.

Exercise 3

The previous one-to-one mapping leads to a progressive loss of processors, so that the load balancing is desequilibred. One solution is to split the initial matrix into much more blocs, and mapping many blocs to each processor. Find

this mapping and implement the associated 2D bloc cyclic LU factorization.



One step of the LU factorization in a 2D bloc scheem. The gray, blue green and red arreas represents the application of the function $getrf$, $trsml$, $trsmu$ and $gemm$ respectively.