# Foster’s Algorithm

## Partitioning & Communication

There are two main tasks for this problem of LU decomposition, calculating the U matrix and the L matrix. Calculating the U matrix is divided into each row and determining how to negate the rows in the column below the row’s starting value. Calculating the L matrix, is just a matter of finding the scalar that was used in to negate the rows in calculating the U matrix. If we insert the scalars into a separate matrix to keep track of those, that is less calculating. So the main task we can put into parallel is calculating the scalars to calculate the U. We can also parallelize the creation of the L matrix as it’s just accessing the scalar matrix to build it.

What we have to communicate U, as that is both an input and what we are calculating. We must also communicate the L matrix as well as the Scalar matrix.

## Agglomeration

Each, row in U needs to negate the values in the column below its starting value. Since L is dependent on these scalar values, we need to calculate matrix U first. However, since we have the scalars at the time of calculating U, we can actually combine the two tasks. As we step down through the rows in the U matrix, we can insert the resulting scalars into matrix L. This means we will have to only communicate each row of U at each phase of the program.

## Mapping

We can map each row in U to a process. This would result in solving the U matrix and L matrix in Phases, one for each row. Where the process whose phase it is, broadcasts its row to the other processes and they calculate the scalar necessary to negate the appropriate values in their row, and place this value into their row in the L matrix.