## Authors:

Andrew Kizzier

Rachel Krohn

James Tillma

Jonathan Tomes

## Included:

## Compiling:

Simply use the included make file.

Or for the individual programs:

Shared Memory Version:

gcc –g –wall –fopenmp –o prog4\_shared prog4\_shared.c

Distributed Memory Version:

mpicc –g –wall –std=c99 –lm –o prog4\_dist prog4\_dist.c

## Usages:

## Problem:

Perform LU decomposition of a matrix of a set number of rows. And we must perform this using both a shared memory version and a distributed memory version. We are requiring that the matrix be a square matrix as many of the applications for the L and U matrices require that the matrix be square.

## Algorithms:

We will use Gaussian row elimination in order to create the upper right matrix that is U and the lower left matrix that is L. This requires that we do row swapping as we discover the need to. Upon eliminating a row we insert the scalars used into L. After we have completed all row eliminations we print each matrix (A, P, L, and U) to output. We then print to output that A = (LU)\* P’ to show that LU decomposition was successful.

## Timings: