HPC Homework 3

Yuwei Jiang, N11031694

In my realization of Jacobi and GS, I set the first input n to be the number of interior points, i.e. the mesh size is $h = \frac{1}{n+1}$, number of points being updated in each iteration is n, and the number of all points including the boundary points are n+2. The second input M is the maximum number of iterations. I set the stopping criterion to be: $tol = 10^{-5}$. The iteration stops when the L2 norm of residue is smaller than tol, or the number of iteration reaches M.
The following experiments are tested on Stampede using Intel compilers.

1 OMP Jacobi

As shown in previous homework, the residue after a fixed number of iterations is independent of the number of threads. For Jacobi, we tested mesh size: n=50, 100, 200. The number of iteration required to satisfy the stopping criterion and the residue when the iteration stopped are: for n=50, 7045 iterations, $residue = 9.987768 \times 10^{-6}$; for n=100, 28348 iterations, $residue = 9.996490 \times 10^{-6}$; for n=200, 115101 iterations, $residue = 9.999012 \times 10^{-6}$. The timings are:

nthreads\n	50	100	200
20	0.6031324	1.800407	10.64825
50	4.363857	27.03929	98.12974
100	5.357732	28.93228	98.46263

The strange thing is that for fixed mesh size (fixed amount of computation), the time increases as the number of threads increases.

2 OMP Red-black GS

The number of iteration required to satisfy the stopping criterion and the residue when the iteration stopped are: for n=50, 3614 iterations, $residue = 9.988796 \times 10^{-6}$; for n=100, 14533 iterations, $residue = 9.990715 \times 10^{-6}$; for n=200, 58969 iterations, $residue = 9.999950 \times 10^{-6}$. The timings are:

nthreads\n	50	100	200
20	0.3412227	1.818936	3.923450
50	0.9273614	5.184860	27.36898
100	1.203803	10.78586	34.44705

It can be seen that the timings for red-black GS is always smaller than Jacobi with the same mesh size and number of threads, thus red-black GS is a better smoother than Jacobi. But the same problem as Jacobi smoother exists: for fixed n, time increases as the number of threads increases.