ASSIGNMENT 4

HPC1 Fall 2014

Due Date: ***Tuesday, November 11***

(please submit your report electronically (by email), in one PDF file, as *hw4-yourUBitname.pdf*)

Problem 1: Consider the Laplace equation in two dimensions (we will consider Dirichlet boundary conditions in the unit interval),

$$\nabla^2 \phi = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \phi = 0. \tag{1}$$

Now take the case of a uniform square grid of M points in each direction, and using a simple two-point numerical derivative, it is not difficult to show that the optimum value of ϕ at the point $(i, j) = (x_i, y_j)$ is given by:

$$\phi_{ij} \simeq \frac{1}{4} (\phi_{i+1,j} + \phi_{i,j+1} + \phi_{i-1,j} + \phi_{i,j-1}) \qquad (i, j) \in (1..M, 1..M). \tag{2}$$

The *relaxation method* consists of iterating this equation to obtain a new, improved solution from the previous iteration:

$$\phi_{ij}^{n+1} \simeq \frac{1}{4} \left(\phi_{i+1,j}^{n} + \phi_{i,j+1}^{n} + \phi_{i-1,j}^{n} + \phi_{i,j-1}^{n} \right) \qquad (i, j) \in (1..M, 1..M) \tag{3}$$

A simple summary of the technique (otherwise known as Jacobi iteration):

1. Apply a square lattice with uniform spacing - label the points (i, j).
2. Apply the fixed boundary condition values.
3. Make an initial guess for the interior points $\phi_{i,j}^{0}$.
4. Iterate until convergence, using the "cross" scheme in Eq. 3 (note that better schemes are available - feel free to derive one, or look them up in standard references).

a. Write a serial code to perform the above solution for the problem on the unit square ($0 \leq x, y \leq 1$) with boundary conditions

$$\begin{aligned} \phi(x, 0) &= \sin(\pi x), \\ \phi(x, 1) &= \sin(\pi x) e^{-\pi}, \\ \phi(0, y) &= \phi(1, y) = 0. \end{aligned}$$

You can verify that the analytic solution to this problem is $\phi(x, y) = \sin(\pi x) e^{-\pi y}$.

b. How does the number of Jacobi iterations required for convergence in your code depend on the grid size? Show a plot of this behavior. Take a reasonable threshold for convergence, say to where the L2-norm of the solution difference between iterations reaches 10^{-5}.

c. Parallelize your solver using OpenMP.

d. Take a grid size that is significant enough to require some time (say 1000 or so, but feel free to innovate), and do a study in parallel scalability - how well does your parallel program scale with additional processors? Plot the execution time, parallel speedup, and parallel efficiency.