Tutorial - 5

In this tutorial, you will write a serial and an OpenMP programs for the solution of Jacobi iteration method. Consider the solution of system of linear equations, $Ax = b$, where A is the coefficient matrix, x is the unknown vector to be determined and b is a known right-hand-side (RHS) vector. The matrix A considered to be a dense and diagonally dominant matrix given as follows:

$$A = \left[a_{ij}\right]_{n\times n} \quad \text{and} \quad 1 \le i, j \le n,$$

$$a_{ij} = \begin{cases} i + j, & i = j, \\ 1, & i = 1, j = n, \\ 2n - 1, & i = n, j = 1, \\ \frac{1}{n}, & \text{other cases,} \end{cases}$$

The vector b is given by $b = (1, 0, 0, ..., 0)^T$. The point Jacobi method uses the following equation to obtain an improved solution, ${x_i}^{k+1}$ from the guess/previous iteration value of ${x_i}^k$:

$${x_i}^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^k\right). \tag{1}$$

The iterations can begin from an initial guess of ${x_i}^k$ and continue till the solution has convered to some tolerance (ϵ) as per the following criterion:

$$\frac{\left\|x^{k+1} - x^k\right\|}{\left\|x^k\right\|} \le \epsilon. \tag{2}$$

1. Develop a sequential program (in C/C++ or FORTRAN) for solving the system of linear equations described above using Jacobi iteration method.

2. Develop an OpenMP version of the serial program that you developed earlier by suitably modifying the code. Convince yourself that the serial and the parallel programs that you developed are indeed giving correct result. Using the size of the system as $n = 10, 100$ and 1000 and the number of threads as $p = 2, 4$ and 8 time your program. Do you notice any performance improvement using the parallel version of the program?