Parallel Computing

Assignment 2 (5 Marks)
Deadline: 17th October 11:59PM

1. Stencil computations (5 marks): Consider a 2D grid of data stored as a Matrix of doubles. The elements are updated iteratively until some condition is satisfied. The element's new value is the average of the element's current value and the current values of its four neighbors. Every element in the matrix is similarly updated. At the end of each iteration, the maximum value change is determined (the difference between its old value and its current computed value). That maximum of all these differences is computed across the entire matrix. If the maximum difference for a given iteration falls below a given threshold, the calculation ends, else, the next iteration starts. To read: https://en.wikipedia.org/wiki/Iterative_Stencil_Loops

   From wikipedia: "Iterative Stencil Loops (ISLs) are a class of numerical data processing solution which update array elements according to some fixed pattern, called a stencil. They are most commonly found in computer simulations, e.g. for computational fluid dynamics in the context of scientific and engineering applications. Other notable examples include solving partial differential equations, the Jacobi kernel, the Gauss–Seidel method, image processing and cellular automata. The regular structure of the arrays sets stencil techniques apart from other modeling methods such as the Finite element method. Most finite difference codes which operate on regular grids can be formulated as ISLs."

   - Parallelize the computation using both MPI and OpenMP.
   - Any number of processes and threads per process can be launched.
   - Input to the program: Matrix size (n), and threshold (t). Example: export OMP_NUM_THREADS=2; mpirun -np 4 ./pgm 10 0.01 (A 10X10 random matrix of doubles is generated (using the function given), and the computation is run till the threshold value of 0.01 is met). The computation is parallelized across 4 processes and 2 threads per process.
   - The data of the matrix should be distributed across processes, and the computation should be fairly load balanced across threads and processes.
   - Time the computation after the matrix generation using the timer provided (Do not time with dangling threads).
   - Output: Time, number of iterations, and number of total threads launched.
   - For a fixed threshold, vary the value of n, and plot the computation time. i.e., n along x-axis, and time along y-axis.
   - Upload the code as a single file (your roll number in lower case should be used to name the file). Also upload the plot as a single file/image.
   - Any kind of cheating will be heavily penalized.

- The best performing code will win a prize.

2. Parallelize using MPI, the N-queens problem. Input: n, Output: Number of solutions (2 marks).

Please note that the maximum marks you can get in this assignment is 5. You can choose to skip the second question, and only attempt the first.