

APMA 2822B: Home Assignment No 2

Due by October 31, 2025.

The goals of this assignment are:

1. Practicing writing a [simple] code, compiling it and executing on a parallel computer.
2. Estimating the bounds on a theoretically achievable performance using the Roof Line model.
3. Shared Memory model and OpenMP directives and distributed memory model and MPI.

Assignment:

Complete the code we started in the last lecture for solving a 2D equation using 2nd order central differences.

Computational domain is defined on a square with x-coordinates in the range of [0 1] and with y-coordinates also in the range of [0 1].

Consider second order differential equation

$$\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 = f(x,y);$$

Here $\partial^2 u / \partial x^2$ and $\partial^2 u / \partial y^2$ are approximated by the 2nd order finite difference scheme

$$\partial^2 u / \partial x^2 = (u[i-1,j] - 2*u[i,j] + u[i+1,j]) / (\Delta x * \Delta x)$$

$$\partial^2 u / \partial y^2 = (u[i,j-1] - 2*u[i,j] + u[i,j+1]) / (\Delta y * \Delta y)$$

Assume that the exact solution is $\sin(2\pi x) * \cos(2\pi y)$ and, accordingly, $f(x,y) = -2*(2\pi)*(2\pi) \sin(2\pi x) * \cos(2\pi y)$. Solution at the boundary $u(\text{boundary})=0.0$.

Task No 1. Write an iterative solver to find the value of $u(x,y)$ using the shared memory model and OpenMP directives. Measure the time for each iteration and number of iterations required to converge with some level of accuracy. Measure the bandwidth achieved and create a roof-line model for a loop where the solution u is *updated*, and similarly for loops required to compute the convergence error.

Task No 2. Write an iterative solver to find the value of $u(x,y)$ using the distributed memory model and MPI. Make sure that results are correct. Use non-blocking MPI_Isend and MPI_Irecv to exchange data between the partitions. Use MPI_Allreduce when computing the convergence error. Measure the number of iterations needed to converge.

Compare the time to solution required in Task No 1 and Task No 2.

You can use the *gettimeofday* function as a timer, or another timer of your choice.

A simple example of measuring elapsed time in a C++ code using the *gettimeofday* function can be found in: <https://people.cs.rutgers.edu/~pxk/416/notes/c-tutorials/times.html> , scroll to the end of that document.

Plan your solver carefully, think about strategies for testing individual components of the program.