APMA 2822B: Home Assignment No 4

Due November 18, 2025.

The goals of this assignment are:

1. Practicing writing a [simple] code, compiling it and executing on a parallel computer.
2. Practicing decomposing problem via a distributed memory model and offloading computations to GPUs.

Assignment:

Based on the code you have written for HW3 write a new code with offloading all parallelizable parts of the workflow to GPUs (stencil calculations, evaluating the errors, packing and unpacking data for partition boundaries.

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

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

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

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

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

Assume that the exact solution is **sin(2πx)*sin(2πy)** and, accordingly, f(x,y) = - 2*(2π)*(2π) **sin(2πx)* sin(2πy).** Solution at the boundary u(boundary)=0.0.

Task No 1. Write an iterative solver to find the value of u(x,y) using the shared memory model and offloading to GPU. Measure the time for each operation including memory allocations, memory copies, kernels. 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, each MPI rank will offload calculations to GPUs. 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 you achieve using the solver you built for homework No. 3 and a similar solver but with offloading to GPUs.

Use “events” to measure timing.

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