Homework 5

1. Further modifying the wave equation solver, consider a true 2D wave equation in the unit square $[0,1]^2$,

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{1}$$

with Dirichlet boundary conditions on all boundaries and initial conditions

$$u(x,y,0) = \sin(2\pi x)\sin(2\pi y), \frac{\partial u}{\partial t}(x,y,0) = 0 \tag{2}$$

with an exact solution of $u(x,t) = \sin(2\pi x)\sin(2\pi y)\cos(2\sqrt{2}\pi ct)$.

   (a) Using a centered, 2nd order finite difference (in space and time) scheme

$$\begin{aligned} \frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\delta t^2} &= c^2 \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\delta x^2} \right. \\ &\left. + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\delta y^2} \right) \end{aligned} \tag{3}$$

   and a mesh with N_x points ($[1-Nx]$) in the $x-$ and N_y ($[1-Nx]$) in the $y-$direction, write a parallel program employing a 1D block data decomposition strategy to solve Equation (1). As before, the program should output the *maximum* and its location, average and rms error every K timesteps, dump the complete solution and error every L timesteps and work with an arbitrary number of processors (down to the limit of one gridpoint per processor). For simplicity it may be assumed that the number of processors divides the number of gridpoints exactly (though the more general case is not very much more complicated).

   (b) Consider $N = PQ$ processors, where N is not a prime number and has at least two factors, P and Q. Rewrite your program so that it uses a 2D data decomposition. To keep things simpler, assume that P divides N_x and that Q divides N_y. Distribute the data among the processors using the following scheme:
   Block $[iN_x/P+1,(i+1)N_x/P] \times [jN_y/Q+1,(j+1)N_y/Q]$ is assigned to processor $n = i + jQ$, for $i \in [0,P-1]$ and $j \in [0,Q-1]$. Each

processor needs to exchange boundary point values with 4 neighbours (NSWE: North, South, West and East) or less (for boundary blocks). Write the halo exchange code using

i. Your own packing and unpacking (for array rows in Fortran and columns in C/C++ that are not contiguous in memory)

ii. The appropriate derived datatype

(c) What is the operation cost of the Finite Difference scheme per grid-point? Given (N_x, N_y) and (P, Q), calculate CoC, the ratio of communication "volume" to computation cost for the scheme for the 1D and the 2D block decompositions.

- As $N = PQ$ increases, what happens to each CoC?
- For large N which CoC is more favorable?
- For the 2D decomposition, given (N_x, N_y) (of generally different values) is there a way to choose P, Q in order to mimimize CoC?

(d) Rewrite the code in a more compact form employing a cartesian topology.

(e) Compare the message exchange pattern that you have chosen to the ones used in the Halo.f program you can download from the class website.

2. Eager Beaver material to work on beyond the end of the class. Further modifying the codes you wrote in the previous homeworks, employ hybrid programming to produce MPI-OpenMP versions of your codes:

- for the case of the 1D wave equation on the unit line with both types of boundary conditions
- for the case of the 1D wave equation on the unit square with 1D block decomposition
- for the case of the 2D wave equation on the unit square with a 1D block decomposition
- for the case of the 2D wave equation on the unit square with a 2D block decomposition