Homework 4

1. Further modify the work done in homework 3:

   - Gather (using MPI_Gather()) the complete solution to process 0 every L timesteps where L is defined at runtime and output the solution, the exact solution as well as their difference to disk (in separate files). Also do this at the end of the run.
   - Use MPI_Reduce() to have your program output the maximum (and its indexed location) as well as the average absolute error and the rms errors every K-th timestep, where K is provided at runtime (contrast this with the OpenMP code where it was requested that $K = L$).
   - In the case of Dirischlet BC can you write the code more compactly by employing in the case of sends and receives at the fixed boundaries MPI_PROC_NULL?

2. Eager Beaver[1]: Come up with as tight (in terms of steps) a schedule of communications for the Dirischlet BC problem that uses only MPI_Ssend() and MPI_Recv() avoiding deadlock.

   - Think of alternating directions and matching sends and receives.
   - Can the schedule you come up with be applied to the case of periodic boundary conditions? Are there any further complications?

12.950 Parallel Programming for Multicore Machines Using OpenMP and MPI
IAP 2010