MCS 572, Spring 2008
Midterm exam

1. Download the single-processor program heat.c from the course webpage which computes the heat distribution problem using Jacobi iteration. Compile the program with

```
gcc -march=i686 -O3 heat.c -lm
```

run it and visualize the resulting two files heat_start.dat and heat_end.dat using gnuplot. For this, issue at gnuplot prompt

```
splot 'heat_start.dat' with lines
```

2. Write a parallel program which solves the heat distribution problem on a square domain of \( M \times M \) points using iterative methods introduced in 6.3.2 with strip partitioning. The overall structure of the program is the following: the master node initiates the array of \( M \times M \) points with prescribed boundary conditions and starting points (see below), then sends the appropriate chunks of data to the slave nodes for processing. The slave nodes occasionally compute the termination norm (see below) for their strips, send their norms to the master node, and receive the instruction from the master node of whether to continue. If the master node indicates that the solution has been reached, the slaves send their respective parts of the solution to the master node and exit. In this case, the master node receives the computed strips from the slave nodes, concatenates them into the whole domain, saves the solution and exits.

3. Use the following numerical parameters for subsequent computations:

   (a) Linear size of the domain: \( M = 128, 192, 256 \).
   
   (b) Number of processors: \( p = 2, 4, 8 \).
   
   (c) Boundary conditions: use either of the two sets of boundary conditions defined in heat.c.

4. Initial condition: \( h_{ij} = 0 \) for \( 0 \leq i, j < (M - 1) \).

5. Termination condition: \( \max |h_{ij}^\tau - h_{ij}^{\tau-1000}| \leq 10^{-8} \) over all \( 0 \leq i, j < (M - 1) \), where \( \tau \) is the current iteration. Note that the termination condition is tested every 1000 iterations (i.e. the current data is compared with the data 1000 iterations ago). To check the termination condition, the slave nodes locally compute \( \max |h_{ij}^\tau - h_{ij}^{\tau-1000}| \) for their strips and send the resulting number to the master node. The master node receives local strip norms, computes the maximum among them, checks against the termination condition and instructs the slave nodes whether to continue.

6. Show the 3D mesh gnuplot plots with the solution.

7. Measure the speedup of the parallel computation versus the serial program (you should have a 3 \times 3 table of speedups for \( p = 2, 4, 8 \) and \( M = 128, 192, 256 \)).