# COMS 6100
# Steady State Temperature Distribution Problem
# Due Thursday, December 8, 2016 @ 10:00 AM

## 1 Introduction

Write a Fortran program to calculate the steady-state temperature for a two-dimensional plate. Model the plate as a 500 by 500 grid. Initially, each cell of the grid is at room temperature (20 degrees). At $t_0$ the heating elements are turned on and instantly jump to 200 degrees. The heating elements are all of cells on the left, top and right boundaries (including all four corners). Additionally there are four heating cells at the follow positions: (171,171), (343, 171), (171, 343) and (343, 343). The problem is to calculate the average temperature of the non-heating cells (not the boundaries, nor the four single-cells) and how many iterations it takes to get there. Use single precision floating point values for your temperatures.

To calculate the new temperature value of a cell at time $t$, use the following formula:

$$x_{i,j}^{t} = \frac{x_{i+1,j}^{t-1} + x_{i-1,j}^{t-1} + x_{i,j+1}^{t-1} + x_{i,j-1}^{t-1} + 4 * x_{i,j}^{t-1}}{8.0} \tag{1}$$

except for when $i =$ the bottom row, where $x_{i-1,j}^{t} = 20$ (room temperature).

Use the following condition for steady state:

$$\left| x_{i,j}^{t} - x_{i,j}^{t-1} \right| < 0.01 \tag{2}$$

# 2 Assignment

Work with one other person in the class.

Produce an animation (movie) of the temperatures array over time (see section below).

Collectively report the following (using LaTeX):

1. The execution time for a serial and parallel version of the program. Use appropriate evaluation metrics (at least those discussed is class). Timing calls should be made as the first and nearly last executable lines of code in your program. Temporarily disable writing output files when timing the execution.
2. Report the number of iterations necessary to come to a steady state and what the average temperature is for the non-heating cells.

Individually report:

1. The division of tasks (*e.g.*, who worked on what)
2. The percentage of the total work you did
3. The number of hours you spent on this project

Submit your reports (as PDFs), program and animation(s) to the appropriate dropbox in D2L (elearn.mtsu.edu)

## Animation

For this assignment, make an animation of the temperatures varying per iteration. For instructions on how to use MATLAB to make movies, see the following URL: `http://www.math.canterbury.ac.nz/~c.scarrott/MATLAB_Movies/movies.html`. If you prefer to use `gnuplot` and `convert`, see the instructions below.

### Animations Using gnuplot and convert

First, you will need to save the values of each element of the hotplate to a file. For efficiency concerns, save every hundredth iteration. If you're working on one of the babbage worker nodes or the supernode, please save them in `/sandbox/$USER/` (so that you don't exceed your quota). For `gnuplot` to make 3D plots, print out the row, column and temperature. Additionally, a blank line is needed in between each row. Also, an additional row of values is needed at the end of the output file to force `gnuplot` to print out the last row. Name each file something like `hotPlate0000.csv`. Then save the following `gnuplot` commands to a file (such as `hotPlate0000.gp`):

```
set xrange [1:501]
set yrange [1:501]
set ylabel 'Row'
set xlabel 'Column'
set xtics out
set ytics out
unset border  # turns off the border (so that we can see the edges)
#unset colorbox  # uncomment to remove the color palette on the right
set datafile separator ','

set palette color
set pm3d map
# corners2color c1 tells gnuplot not to average the neighboring values
set pm3d corners2color c1
set terminal gif

set title 'Iteration 0000'
set output 'hotPlate0000.gif'
splot 'hotPlate0000.csv' notitle
```

Next, make a gif file by executing "gnuplot hotPlate0000.gp". Repeat these steps for each iteration.

Once all of the gif files have been generated, you can combine them to make an animated gif with ImageMagick's `convert` command:

```
convert  -dispose  previous  -delay 1  /sandbox/$USER/hotPlate*0.gif \
 -loop 1  /sandbox/$USER/movie.gif
```

The animated gif can be viewed in a web browser.

# 3 Grading

1. Program (serial & parallel) [30%]
2. Animation [25%]
3. Collective Report [35%]
    (a) Serial vs. parallel performance analysis [25%]
    (b) Correct number of iterations and average temperature of non-heating cells [10%]
4. Individual Report [10%]