Lab 9: Numeric Simulation, e.g. of Heat Flow Equations  
(or, "You Can’t Heat a Frying Pan with a Soldering Iron")

In this lab, you’ll be writing a "numerical simulation" very much like the one discussed in lecture. You may simulate the flow of heat through a flat metal plate, or (if you get official permission by email) some other system such as the “n-body” problem. In any simulation, you should include an analysis of “numerical stability” issues (Step 3 below) as well as programming the simulation. The instructions below focus on heat flow; if you wish to simulate some other system, ask your lab instructor how to proceed.

Pre-Lab Work:

What values or errors do the following Python expressions produce? Get the lab files and review helpful_numpy_examples.py if necessary, and then predict each result below by drawing “box and arrow” diagrams, and then check your answers by typing (or copy-pasting) each group of statements into an interactive python session. Within each group, some of the results depend on prior steps, so remember to enter all the steps, in the order given, to check an answer.

- \( x = 1.0/3 \)
  \( 3 \times x \)
  \( y = 10 \times x - 3 \)
  \( 3 \times y \)
  \( x-y \)

- \( \text{wire} = [300.0, 300.0, 300.0] \)
  \( \text{wire} \)
  \( \text{wire}[1] \)
  \( \text{plate} = [\text{wire}, \text{wire}, \text{wire}, \text{wire}] \)
  \( \text{plate} \)
  \( \text{plate}[1] \)
  \( \text{plate}[1][1] = 305 \)
  \( \text{plate}[1][1] \)
  \( \text{plate} \)
  \( \text{wire} \)

- \( \text{from numpy import *} \)
  \( \text{wire} = \text{zeros}(3, \text{dtype=double}) \)
  \( \text{wire} \)
  \( \text{wire}[:] = 300 \)
  \( \text{wire} \)
  \( \text{wire}[1] \)
  \( \text{plate} = \text{zeros}([4,3], \text{dtype=double}) \)
  \( \text{plate}[:] = 111 \)
  \( \text{plate} \)
  \( \text{plate}[1][1] \)
  \( \text{plate}[1] \)
  \( \text{plate}[:] = 1 \)
  \( \text{plate}[1] = \text{wire} \)
  \( \text{plate} \)
  \( \text{plate}[:] = \text{wire} \)
  \( \text{plate} \)
  \( \text{plate}[1] \)
  \( \text{plate}[1][1] = 305 \)
  \( \text{plate}[1][1] \)
  \( \text{plate} \)
  \( \text{wire} \)
Lab Work:

Get the starter files for the numerical_simulation project, which include the one-dimensional simulation from lecture (in wire.py). Edit the file plate.py to perform the simulation described below. You may find it helpful to refer to interface-text.py (which contains sample functions that start the simulation) or the wire simulation files (from lecture). You should not need to look at interface-graphics.py at all, but you can run this file to see the graphical output.

In the plate simulation, you should use a two-dimensional array (i.e., a matrix) to store the temperatures on each small square (1cm by 1cm in this lab) of the plate. We will be simulating a plate that is 1cm thick, so each pair of adjacent regions of the plate will have a one-square-centimeter connection. The size of the plate is given by the size parameter to the create_plate function, and the initial temperature is given by the temperature parameter. You must build the simulate_plate and simulate_plate_N_steps functions, as follows:

1. The simulate_plate function returns an updated array of temperatures after one 1-second time step. In each step (i.e., in one call to simulate_plate):
   - Heat is transferred to or from each part of the plate and each of its (up to four) neighbors, at a rate determined by the difference in temperature times the heat transfer constant (given by the htc parameter).
   - Heat is transferred to or from each part of the plate and the air, at a rate that is determined by the difference in temperature times an air heat transfer constant (given by the ahtc parameter). This constant is different from the heat transfer constant used for neighboring parts of the metal plate, but is the same everywhere on the plate — to simplify the simulation, we assume that the edges are insulated, so that they do not lose heat to the air more quickly than the rest of the plate. The air temperature, which is given by the airtemp parameter, should remain the same through the simulation.
   - The change in temperature for each part of the plate is determined by the change in heat times a heat capacity constant (given by the hcc parameter).
   - One part of the plate (identified by the hotx and hoty parameters) is kept at a constant high temperature (given by the hottemp parameter).

2. The simulate_plate_N_steps function should simulate N seconds of time by repeatedly using the simulate_plate function. It should also keep a list of the average temperature after each time step, and print this list after all N steps are done.

3. Edit the interface-text.py file so that it will only perform a simulation for sensible values. You should rule out situations like having htc, ahtc, and hcc all set to 10, in which case the simulation “goes berserk”, while still allowing valid simulations. Include a comment about why you are ruling out the cases you rule out.

   It may be easiest to think separately about issues involving heat flow within the plate and heat flow to the air, rather than thinking about how these might combine to create a problem. For this lab, it is fine to just have two separate tests: one to rule out instability due to heat flow within the plate, and another to rule out instability due to the air.

   (Technically, checking for problematic values should be done in the graphical interface as well, but doing so is beyond the scope of this class.)

Remember to commit your files when you are done.