____

Name

Recitation Assignment # 7

Nov. 8, 2006

You may complete this in class. However, if you are unable to do so, it is expected that you complete this for recitation next time.

Where a box appears, call over Travis to check over your progress. When the sheet is complete, you will hand it in. Also, you are expected to email your final programs to Travis.

For today's assignment, we'll do a little i/o (input and output for the uninitiated). There are two commands that you'll need:

The first is the "input" command, which allows the user to enter data.

```
m1=input("What is the mass of the first block? (in kg)")
```

This is very useful if you want to be able to set the initial conditions in your code "by hand."

The second is the "print" statement, which allows the code to print out numbers to the terminal (NOT the display window).

```
print "T1=",T1
```

In this case, we're printing out a text string "T1=", and printing the variable called T1.

<u>Assignment: Thermal Equilibrium</u>

In class, we discussed how to solve for thermal equilibrium between two solids if we know the initial masses, temperatures, and specific heats of each of them. Sometimes, though, it is helpful to have a computer to solve problems for you. This should **not** be a substitute for doing simple problems with Pen and Paper. We will simply use a new numerical method, called *relaxation* to compute the equilibrium temperature of two solids.

This code is deceptively simple. Mine is only 25 lines long. Still, you'll need to keep on your toes.

1. Put the normal header in the code, ask the user to enter the following quantities: 1) Mass of the first block (in kg), 2) Temperature of the first block (in c), 3) Specific Heat of the first block (in J/kg/K)?, 4),5), and 6) should be the same but for the second block.

☐ 2. In order to add a graphical component, set up a scene with the following commands:

```
scene.range=(2,T1+T2,2)
scene.autoscale=0
```

Then create two "boxes" called block1 and block2. The lengths and widths of both should be 1, and the heights should be the same as the temperature. If you can figure out how to do it, make the bottom of the boxes level with one another. The boxes should be different colors, and sitting right next to each other.

3. Now comes the loop part. Create a loop which says:

   ```
   while (abs(T1-T2) > 0.1):
   ```

   This is very different than what you normally do when you write a "while" loop. Normally, we're imagining a movie wherein each step represents a frame of the movie. Here, each step represents an "iteration" – a step closer to reaching the correct solution.

   Set rate(2), and at each timestep, alter the temperature of the two blocks. How?

   Well, the cooler one should be raised in temperature, the hotter one should be lowered. I have found the following relations useful:

   $$\begin{aligned} \delta T_1 &= \frac{(T_2 - T_1)}{20} \\ \delta T_2 &= \frac{(T_1 - T_2)}{20} \end{aligned}$$

   These are NOT the actual changes in temperature, only a first guess. Note, though, that if the temperatures were already the same, that the changes would be zero. This is precisely the relation we want.

   How much heat would be transferred in the cases of these temperature differences?

   $$\begin{aligned} \delta Q_1 &= C_1 m_1 \delta T_1 \\ \delta Q_2 &= C_2 m_2 \delta T_2 \end{aligned}$$

   Now, here's the important part – energy is preserved, and thus the total heat flowing into block 1 is just:

   $$Q_1 = (\delta Q_1 - \delta Q_2)$$

   And the negative of that for block 2.

   The change in temperature for this "iteration" is simply:

   $$\Delta T_1 = \frac{Q_1}{m_1 C_1}$$

   and similarly for block 2. Use this to update your estimate of the temperature of each block.

4. It is a pain to keep typing in all 6 values. So try the following. Create a file (call it "prog7.in" for example), with 1 number on each line, each corresponding to the input. Then when you run the program, you can simply type:

   ```
   python prog7.py < prog7.in
   ```

   This is an example of an input stream, and makes things MUCH easier.

☐ 5. At each step in the integration, update the images of the blocks so that the heights represent the current temperature. When they reach equilibrium, they should be the same height.

6. After the completion of the loop, print out the temperature of each block.

☐ 7. Do problem 5.8 in your textbook using your code.