CSC 191B: Lab #6: Cool It!

Learning Outcomes

- Manipulation of 2D arrays and functions of two variables
- Visualization of surfaces in 3D using MATLAB's tools
- Using simulation to answer scientific questions

Background. In this lab we'll build on the previous lab and simulate the rectangular plate being cooled off when the heat sources are turned off and it's surrounded by ice water. As before, the plate is 6 units wide and 4 units long. We assume at time $t = 0$ the plate has been heated by the two heat sources and is modeled by the function

$$T(x, y) = 100e^{-0.4((x-1)^2+0.7(y-3)^2)} + 80e^{-0.2(2(x-5)^2+1.5(y-1)^2)}.$$

(This is the same function that is implemented in `temperature.m` from Lab 5.)

As you might expect, the heat from the two hot spots will dissipate into the rest of the plate and slowly dissipate into the ice water, cooling the entire plate down. We will assume that there's a lot of ice water around the plate, so the plate itself won't change the temperature of the water. Modeling this behavior accurately requires some physics and an ability to solve partial differential equations. We'll jump ahead to a simple model that simulates the behavior on a discrete grid reasonably accurately. Our simulation proceeds according to the following rule.

> The temperature at an interior mesh point at time $t + 1$ is the average of the temperature at its four neighbor mesh points at time t.

An interior mesh point at (x, y) has a "north" neighbor at $(x, y + \delta y)$, an "east" neighbor at $(x + \delta x, y)$, a "south" neighbor at $(x, y - \delta y)$, and a "west" neighbor at $(x - \delta x, y)$, where δx and δy are the mesh-points spacings in the x and y directions, respectively.

Problem Statement.

a) The first thing we'll do is set up the plate at time $t = 0$. As in the previous lab, use `linspace` and `meshgrid` and the `temperature` function to store the temperature value on the grid points in a matrix `T`. Pick the number of grid points such that the distance between neighboring grid points is 0.1. Then, set the temperature on the boundary to zero by setting the first and last row and column of `T` to be zero.

b) Next, write a function called `average` that computes the change in a single time step. That is, the function should take a single matrix as input that stores the temperature values at some time t and return as output a single matrix storing the temperatures at time $t + 1$. Remember to fix the temperature of the boundary points at zero (they should not be updated).

c) Finally, write a loop to simulate the change in temperature through time. For example, the following loop will display the simulation using `surf`, assuming the variables and `average` function are defined correctly. Note that the axis limits are fixed so that the scale of the visualization is constant across time.

```
for i = 1:nSteps,
    surf(x,y,T)
    axis([0 6 0 4 0 100])
    title(['time step = ', num2str(i)],'FontSize',14)
    pause(.1)
    shg
    T = average(T);
end
```

After your simulation is working, use it to complete the following tasks:

1. Save the `surf` 3D visualization as a pdf for time 0, time 100, and time 200.
2. Use `pcolor(T), colormap('hot'), caxis([0,100]), colorbar` to visualize the behavior in 2D (with a coloring that fits our simulation!), and save the images as a pdf at times 0, 100, and 200.
3. Determine how many time steps it takes to get the average temperature over the entire plate below 20. Note that the average temperature at time 0 is 40.1645.

Discussion.

1. What do each of the commands `colormap('hot'), caxis([0,100]), colorbar` do? Why did I suggest to include them with the `pcolor` command?
2. Suppose we kept the heat sources on during the simulation. How would you change your code to adapt to this scenario, and what do you predict the temperature distribution would do over time?

What to turn in.

- MATLAB file or files that contains the script used to generate the plots and the answers to the discussion questions.
- Six PDF files that contain the `surf` and `pcolor` plots.

Grading rubric

- Code: 25 points each for the script and the `average` function, which should be well-organized and well-documented
- Plots: 5 points each for the plots, which should be clearly labeled
- Discussion: 10 points each for the discussion questions