

ENG101 – Inlab 12

One hot car

Based upon a lab by Professor Ken Powell (Aero). It has nothing to do with the heat in the labs. Really.

Due Tuesday/Wednesday April 12th or 13th

Abstract

In this assignment, you will use MATLAB to compute and plot the temperature distribution under the hood of an automobile.

Problem Background

For reasons you can imagine, many ambulances are left running even when they are not being used (I'm sorry, sir, I got here as fast as I could. I couldn't get it to start.) This does not create any particular problems in Michigan's climate, but ambulances parked on black asphalt in Phoenix, in August, are another story. Exploding ambulances don't make for good press for the company that designed the ambulances. Certain parts of engines are quite hot, particularly the catalytic converter and the exhaust manifold. Their temperatures and locations determine the temperature distribution under the hood, and tell you, for example, where NOT to put the fuel line.

One property of heat flow is that in an area such as the one under the hood of the ambulance, the temperature at some location tends to approach the average of the temperature at neighboring locations. You will use this fact to compute the temperature distribution in a simple model of an engine. This is a highly simplified version of what you might see when you open up the hood and look in from the top. The exhaust manifold (hot!) and the catalytic converter (really hot!) are the main features. You will solve the problem by putting a grid of points in the "underhood" area, and calculating the temperature at each of the points. The catalytic converter, the exhaust manifold, and the four walls of the engine will be at constant temperatures; those will be your boundary conditions. You will solve this problem by iteration. In other words, you will begin with a guess for the temperature at each point, then you will loop over successive guesses so that each guess is better than the last. Once the change between the current guess and the last guess is small enough, you will stop and plot your result.

MATLAB Background

Your temperatures will be stored as an $N \times N$ array. The following MATLAB tips will be helpful.

1. It's good practice to make the first statement in any MATLAB script **clear**; This gets rid of any variables from previous runs, like erasing the blackboard before starting a problem.
2. To set the i^{th} row of the $N \times N$ array T to a constant value c , use $T(i,1:N)=c$;
3. To set the j^{th} column of the $N \times N$ array T to a constant c , use $T(1:N,j)=c$;
4. We can define a block of values in an array by the left (L) and right (R) column of the block, and the bottom (B) and top (T) row of the block. To set a block of values defined by jL, jR, iB, iT to a constant c , use $T(iB:iT,jL:jR)=c$;
5. To set an entire $N \times N$ array to a constant c , use $T(1:N,1:N)=c$;
6. To set one array (A) to be equal to another (B) is very simple in MATLAB. Just say $A=B$;

Your Assignment

Write a MATLAB script that carries out the following steps. We have provided a fair bit of the code as a starting point! You should carefully look over that code.

1. Initialize N, the number of points in each direction. While debugging your code, use N=10. For your final run, try N=100, or something even larger. Keep N an even multiple of 10; the problem defined below is based on that assumption.
2. Use the following table to define the locations of the bottom, top, left and right boundaries of the exhaust manifold and the catalytic converter.

Manifold		Catalytic Converter	
MB	$8*N/10$	CB	$2*N/10$
MT	$9*N/10$	CT	$4*N/10$
ML	$4*N/10$	CL	$2*N/10$
MR	$9*N/10$	CR	$3*N/10$

3. Initialize an N x N arrays for storing the temperatures. For your initial guess, just set the entire array to zero.
4. Initialize the scalar variable maxChange to something large (say 1000). This will be used to keep track of how much the temperature distribution changes for each iteration. Initialize the walls to 80 degrees, the Manifold to 400 degrees and the Catalytic Converter to 1200 degrees.
5. Carry out the following steps, until maxChange becomes small (say, less than 0.1 degrees)
 - a. For each point, not on the wall, set the the new temperature at point (i; j) to the average of the temperature at the four nearest points. This can be done by (for each i and j that are not a wall)

$$T_{New} = (T(i-1, j) + T(i+1, j) + T(i, j-1) + T(i, j+1)) / 4;$$
 - b. For the manifold and the catalytic converter, set those points back to their initial temperature. (We could have instead just not changed those, but that is a bit harder).
 - c. Now set **deltaT = abs(TNew-T(i, j))**; and **T = TNew**;
 - d. Keep track of the maximum value of deltaT over all the points, and call that maxChange.
6. You're done! Plot the results with the plotting function "mesh" (try help mesh).