Problem set 2

Ge 108

Due 12 October 1999

1 One dimensional diffusion

A rod is made out of half steel (the left half) and half wood (the right half) (thermal conductivities of $k = 42$ J/sec/K/m and $k = 0.41$ J/sec/K/m, respectively.) The steel end of the rod is held at 200 C while the wood end is held at 20 C. What is the eventual steady-state temperature distribution within the rod?

2 Thermal waves

(a) Average soil has a density (note the units!) of 2.5 g/cm^3, a specific heat of 0.2 cal/C/g, and a thermal conductivity of 0.0023 cal/sec/cm/C. What is the thermal skin depth in soil as a function of frequency. What is the skin depth for a 24 hour osciallation. A 365 day oscillation? A 20,000 year osciallation?

(b) If you lived in a place where the average temperature was 18 C that had yearly sinusoidal oscillations of $\pm$30C, how deep would you have to bury your pipes so that they wouldn't ever freeze?

3 More thermal waves

Program `diffuse.pro` (IDL) or `diffuse.m` (MATLAB) on the web page solves the thermal diffusion equation in one dimension with a sinusoidally varying temperature at the surface.

Comments for the **IDL** version only: This is a true IDL *program*, rather than collection of commands, so it means that you have to *compile* it before running it. To compile, just type (in IDL) `.run diffuse`. Once compiled, it is essentially a new IDL command that can be used like any other IDL command. To run, type `diffuse,`*period*, where *period* is a number giving the time period of the oscillation that you want to watch. The program will run forever until you hit a key.

Comment for **MATLAB** version only: This program is a true MATLAB funtion, rather than just a collection of commands. That means that it can take input and give output and that it essentially becomes a MATLAB command like any other (read the section under *Scripts and Functions* on the MATLAB web page). To run this program just type `diffuse(`*period*`)`, where *period* is a number giving the time period of the oscillation that you want to watch. The program will run forever until you type cntrl-C.

(a) Run the program for osciallation periods of 100,1000, and 10000 and estimate the e-folding decay length scale for each thermal osciallation (remember, e is about 2.7, so look for the length it takes the wave to decay to 1/2.7 of its initial value). Does your estimate from the numeric equation fit the formula for the thermal skin depth?

(b) Modify the program for the case where the temperature at the far end is held constant at 250 degrees. (**IDL only:**To run the modified program you must now recompile, so type `.run diffuse` again). Run this program with the same three values for the period. How much is the solution affected in these three cases? As always, print out your modified program and turn it in with the problem set.

4 More numerics

Program `diffuse2D.pro` or `diffuse2D.m` is the logical extension of the numeric diffusion solution into two dimensions. The way it is currently set up, it starts with a plane that is 10 degrees in most places but with a box equal to 80 degrees and it watches the evolution of the temperature. The heart of the solution of the differential equation is a single line in the middle of a double loop.

(a) Explain where each of the terms in this line comes from in terms of the diffusion equation in two dimensions.

Run this program. Note that it is SLOW. Moving to two dimensions means that we suddenly have the square of the number of elements to calculate. This case is one where we would start to think about how to do the job with a better, faster integrator.

(b) What does the solution look like after a time of about 100? Explain this shape in terms of a Fourier series (don't actually *do* any Fourier analysis, just wave your hands and talk about how you *might*.

(c) Modify the program the program so that the initial conditions are such that there is a vertical stripe of temperature 80 degrees 10 pixels wide in the center of the plate and the temperature is 10 degrees everywhere else. Also modify the program so that it tells you the temperature in the center of the plate as a function of time. How long does it take for the temperature in the center to decay to 1/e of its initial value. How does this relate to the time scale we estimated in class for the longest-wavelength Fourier component?

(d) Notice that in the program we lumped all of the physcial constants into something called `const`. If the value of `const` were increased by a factor of 10, what would we have to do to the program to maintain the same numeric accuracy?

5 Sources of heat

The diffusion equation is a time-rate-of-change equation just like many that we discussed last week, but here the source term happens to be derivatives in the spatial function. We could also explicitly add a source term to the equation (or a sink).

(a) Using just our general ideas from last week of rates of change and sources and sinks, write the formal diffusion equation that includes an energy source (that could, in principle, vary in space and time). What are the units of the term that you added to the diffusion equation?

(b) *Optional extra credit if you're finding the computer parts easy.* Modify `diffuse2D.pro` or `diffuse2D.m` so that it now includes a source such that energy is constantly being added into a vertial strip 10 pixels wide and the initial conditions are 50 degrees everywhere. How does the maximum temperature at the center point of energy input differ if the thermal conductivity is increased by a factor of 10?