Project 6: Solving PDEs- Heat Conduction in a Rod

Due Dates

Week 10, Monday at 4:59

Your submission should be a single zip file containing all the .py files containing your code as well as a Single typed PDF document with your test results and answers to the problems/questions. Word documents will not be accepted. Your zip file should be in the form <lastname>_project_< N >.zip where you replace <lastname> with your last name and <N> with the project number. Upload your zip file via the link provided on the project webpage found at http://urminsky.ca.

In your PDF, please include your python code and any figures or graphs you were asked to plot.

No handwritten work will be accepted. Please include any mathematics in a typed format. You are welcome to use latex or word to do your write up.

Please write and hand in your own work. Work which is deemed too similar to another students work will be given an automatic zero. It is ok to work together, in fact, I would encourage you to work together. However, when too many similarities exists between two or more student's work, all work will be given automatic zeros.

Documentation: *Make sure that your code is well enough commented so that it can be easily understood by almost anyone, including yourself, if you look at it again at the end of the course (when you've forgotten how you coded it.)*

Project Briefing

The goal of this weeks project is to implement an explicit finite difference method to simulate heat conduction in an insulated metal rod. This will entail the solution of the partial differential equation that describes the change in temperature both with distance along the rod and with time. Once you have developed and tested your program, you can use it to explore the effects of changing the boundary and initial conditions.

Part 1

In class we discussed thermal conduction in an insulated metal rod whose ends are in contact with heat sources held at a fixed temperature. We saw that this case can be described by the 1-d thermal diffusion equation,

$$\frac{\partial T(x,t)}{\partial t} = \kappa \frac{\partial^2}{\partial x^2} T(x,t)$$

where $T(x,t)$ is the temperature at distance x along the rod and time t. The constant κ is the thermal diffusion coefficient of the metal.

using the lecture as a guide, recast the thermal diffusion equation above as an explicit finite difference equation. Show all steps and clearly define your time-space grid.

Design, write, and test a PYTHON program that uses this finite difference equation to compute the change with time of the temperature profile along an insulated metal rod of length L (in meters) and thermal diffusivity κ. The initial values and boundary conditions should be supplied as parameters to the program.

The program should

- Read the initial values (i.e., the value of T at each grid-point, x_i, along the rod at time $t = 0$) from an input file.
- Check that the stability condition is satisfied and terminate the program with a warning message if it isnt.
- Write out to a file the temperature profile along the rod at a specified end-time (i.e., the values of x_i and $T(x_i, t)$ at the specified time t).

The program should be named hotrod and have the following form:

```
def hotrod(tau,L,kappa,T_left,T_right, t_end, infile, outfile):
    # Your code follows
    ...
```

where

tau : the timestep, that is $t_{n+1} = t_n + \tau$

L : the length of the rod (i.e. the distance between the left and right boundaries)

Kappa : thermal diffusion coefficient

T_left : Temperature of heat source applied to left end of rod

T_right : Temperature of heat source applied to right end of rod

t_end : the time at which the calculation is terminated

infile : The name of the input file that stores the initial values (i.e. the values of $T(x_i, 0)$)

outfile : the name of the output file to which you will write the temperature at t_end

You may download and use the following files from the course website:

invals.py Function that generates a set of initial temperature values and writes them to a file

proj6_frag1.py code fragment that opens an infile, reads the initial temperatures and stores them in a list.

Test your program by running a test case with the following parameters:

$$
\begin{aligned}
\text{L} &= 1.0 \text{ m} \\
\text{Kappa} &= 0.01 \text{ m}^2/\text{s} \\
\text{T\_left} &= 1000 \text{ K} \\
\text{T\_right} &= 273 \text{ K}
\end{aligned}
$$

Initial values: assume the rod has a uniform temperature of 273 K at $t = 0$

Run the test case for several different values of tau and t_end. Does it do what you expect it to?

Part 2

When you are satisfied that the program is working properly, feel free to use it to explore as many different scenarios as you like. However, **to receive full credit you must at least investigate the following cases.**

1. For the test case parameters above, use your program to create plots of the temperature profile along the rod at $t = 5$ s, 10 s and 20 s. Approximately how long does it take for the temperature profile to reach a steady-state (i.e., when the temperature profile no longer changes significantly with time)?

2. Boundary conditions: both ends of the rod are in contact with plates whose temperature is fixed at 273 K. Initial values: the rod has a temperature of 273 K everywhere except the mid-point $x = L/2$, which at $t = 0$ has a temperature $T(L/2) = 1000$ K. What happens as time passes? Does the temperature profile reach a steadystate? If so what is it and about how long does it take to reach this steady-state?

3. Boundary conditions: both ends of the rod are in contact with plates whose temperature is fixed at 1000 K. Initial values: the rod has a uniform temperature of 273 K. What happens as time passes? Does the temperature profile reach a steadystate? If so what is it and about how long does it take to reach a steady-state?

In each case, write a brief description of the behaviour that you observe in the simulations and illustrate your answers with plots of the temperature profile at selected times. **You must submit these notes and the accompanying plots to receive full credit.**

Program design issues to consider

- The rod should be represented as a 1-dimensional array, which stores the current values of temperature at each point. You'll need two such arrays, one storing the temperature before the current update and one storing it after the update.

- How will you implement the stability check?

Part 3

Create a new program (hotrod_sine) to explore the case where the temperature of the heat source at left-hand end of the rod varies sinusoidally between values of 200 and 400 K. Try several different periods for the sinusoid. Illustrate the behaviour with a graph which shows the temperature at well spaced values of time.