**MGS 723 – Problem Set #3 – Solving the conduction equation with finite differences**

*Assigned: 10/06/21*

*Due: 10/20/21*

**Problem background: Intrusion of a dike into the surrounding/“country” rock**

Diagram

Description automatically generatedWe will consider the instantaneous intrusion of a uniform temperature dike (*T* = 1200 °C), with width *W*, into surrounding rock (*T* = 300 °C) of length *L*. We are interested in determining the time and spatially-dependent temperature – *T(x,t)* – that satisfies the 1-D dimensional conduction equation (Equation 1 below). While this is a very simple setup, it should provide you with intuition about how the temperature equation behaves and introduce you to solving equations using finite differences. Below is a sketch showing the physical setup of the model (left panel) and how we discretize the problem (right). The discretization uses *n* as the time index (the vertical direction in the figure) and *i* as the space/grid index (the horizontal direction in the figure)

**Technical background: Explicit and implicit discretizations of the conduction Eq.:**

For constant thermal diffusivity/conductivity, the 1-D conduction equation is:

(1)

Whereis the thermal diffusivity (for rocks 10-6 m2/s). To solve this numerically, we need to “discretize” this equation into the temperature values at different grid nodes and different times (e.g., for the temperature at timestep *n* and grid node *i*) so that we can solve for the temperature in the future (e.g., ). In explicit finite difference schemes, the temperature at time *n* + 1 depends only on the already known temperature at time at the previous time, *n*. The simplest explicit discretization of Equation 1 is:

(2)

Since we know the temperature structure of the current time (, , and ), we can calculate the temperature at the next time by simply rearranging Equation 2:

(3)

In calculating the explicit solution, we use Equation 3 to compute the new temperature for every time-step (*n* = 1 to NT) and at every grid node (*i* = 1 to NX). The implicit solution is sightlier trickier to deal with as the spatial derivative is evaluated at the new timestep (*n* + 1). The simplest implicit discretization of the Equation 1 is:

(4)

Rearranging Equation 4 for the temperatures at the next time-step (*n* + 1) gives:

with (5)

Where the left-hand side (LHS) contains the unknown (*T* at *n* + 1) and the right-hand side contains the known quantities (*T* at *n*). Because we have more than one unknown on the LHS, we need to solve a system of a linear equations. Hence, Equation 5 can be written in matrix form:

(6)

A picture containing text, clock

Description automatically generatedWhere is the coefficient matrix (i.e., containing and coefficients), is the unknown temperature vector (i.e., at all grid points), and is the known temperature vector (i.e., at all grid points). For a 6-node grid, the matrix and vectors would look like:

In this example, the upper left and upper right-most coefficients in (= 1) set the boundary conditions to be temperatures fixed at specific values (i.e., and ). To solve for the new temperatures, at any given time, we need to construct and and then do a matrix inversion for :

(7)

This is easy to do in one line using either Python [x = numpy.linalg.solve(A, b)] or Matlab [x = A\b].

**Questions:**

*Note: In addition to answering the questions, please provide your code as an appendix (i.e., attached at the end) and include Python plots to justify your answers.*

1. **Explicit solution:**
2. Download the skeleton Python script “1D\_Explicit\_Skeletion.py”. Using the above discretization for the explicit solution, fill in the blanks (‘???’) to complete the code.
3. Uncomment the plotting portion of the script (lines 45-53). Save a plot showing the time-evolution of the temperature profile. What parameter determines the relationship between two spatial solutions at different times?

Now let’s to plot the solution differently: First, modify the vertical (*T*) axis of the plot to have an upper limit corresponding to the maximum temperature in the profile at each time. Second, divide (/non-dimensionalize) the distance plotted on the horizontal axis by a characteristic diffusion length given by (Recall that this is a length because has units of m2/s). Save this plot. Comment on how (/if) the temperature evolves when plotted along this re-scaled horizontal axis. Why does normalizing the horizontal axis like this produce this effect?

1. Return to the original, simpler way of plotting the solution (i.e., without normalization). Vary the parameters (e.g., use more grid points, a larger or smaller time step). Compare the results for small and with those for larger and . How are these solutions different?

If the time step is increased beyond a certain value, the numerical method becomes unstable and does not converge (i.e., it grows without bounds and exhibits non-physical features). For the original (0.5 m), increase until the solution becomes unstable. At this stable-unstable transition, calculate the value of the stability parameter: . For what values of is the solution method stable? This is the Courant-Friedrichs-Lewy stability criterion. is a non-dimensional number; Can you describe what it means physically?

1. **Implicit solution:**
2. Download the Python script “1D\_Implicit\_Skeleton.py”. Using the above discretization for the Implicit Solution, fill in the blanks to complete the code.
3. As for the explicit method, compare the results for variable and . In particular, assess whether the solution remains stable above the stability criterion for the explicit method (see your answer to 1C). Does it? What does this tell us about this solution method?
4. Now modify the constant-temperature boundary conditions to Tleft = 300 °C and Tright = 600 °C. Describe how this affects the temperature evolution.

Now, compute the steady state solution (i.e., with ) for these conditions. This can be done in two ways:

1. Wait until the time-dependent solution converges (i.e., run for large times)
2. Write out the discretization with and modify your finite difference scheme to solve this directly.

Describe the steady-state temperature profile. Recalling that heat flux is proportional to the temperature gradient (), comment on how heat flux varies along the steady-state temperature profile.

**Bonus question:**

Derive and implement an explicit finite-difference scheme for variable *k* (conductivity). Test the solution for the case of *k* = 12 W/mK inside the dike and *k* = 2 W/mK in the country rock, and vice-versa (use reasonable constant density and specific heat values. What’s the effect of having higher conductivity within the dike on the thermal profile evolution? How about the other way around?

*Hint: You will need to i) set up a vector containing the conductivity values; ii) solve (/discretize) the following form of the 1-D heat equation:*

Note: These problems are modified from Exercises #4.2.1 and #4.4.4, Becker and Kaus, Numerical Modeling of Earth Systems