**MGS 723 – Problem Set #5 – Moving to 2-D…**

*Assigned: 11/03/21*

*Due: 11/24/21*

**Problem background: Conductive cooling and advection of a plume in 2-D**

We will consider the conductive cooling and advection of a thermal plume head in the asthenosphere. This will involve implementing the heat and advection equations in 2-D. As setup in the skeleton code, the plume head will have an initial radius of 50 km, an initial temperature of 1800 C, and will be placed within background mantle of temperature 1000 C (Tmantle).

**Questions:**

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

1. **Explicit solution of the 2-D heat equation:**

In 2-D, and for constant thermal diffusivity, the conduction equation is:

(1)

The simplest explicit discretization (FTCS) of Equation 1 is:

(2)

1. Download the skeleton Python script “2D\_Heat\_Explicit\_Skeleton.py”. Using the above discretization, fill in the blanks (‘???’) to complete the code. The plume is placed beneath the lithosphere and all thermal boundary conditions should be constant temperature (T = Tmantle at the base and edges, T = Tsurface = 0 C = at the upper boundary)
2. Run the model for a total time of 4 Myrs. Save a plot that zooms into the plume/near-surface region. Using this, i) Describe how the plume is affecting the thermal structure of the lithospheric plate (i.e., the cold upper thermal boundary layer); ii) Describe the thermal structure of the plate at the model boundary (Is this realistic? How could you modify the boundary conditions to fix this?)
3. Vary the parameters (e.g., more/less grid points, larger/smaller time step) and calculate the stability parameter: . For what values of is the method stable? Is this equivalent to what you found for the 1-D model (Prob. Set #3)?
4. **Implicit solution of the 2-D heat equation:**

The simplest implicit discretization (FTCS) of Equation 1 is:

(3)

Rearranging Equation 3 for the temperatures at the next time-step (*n* + 1) gives:

(4)

with

As in Problem Set #3, this corresponds to a linear set of equations () with the solution . However, in 2-D, (which contains the coefficients on the left hand-side) is more complex. It contains the and coefficients and has dimensions nx\*nz by nx\*nz. is a vector containing the old temperatures (i.e., at ) and has length nx\*nz (i.e., the 2-D temperature field has been unwrapped into a vector). Similarly, also has length nx\*nz and contains the temperatures that are to be solved for (i.e., at ).

A picture containing text, clock

Description automatically generatedIn 2-D, an example of what will look like is:

1. Download the skeleton Python script “2D\_Heat\_Implicit\_Skeleton.py”. Using the above discretization for the explicit solution, fill in the blanks (‘???’) to complete the code. Plot the model temperature evolution.

*Note: In 2-D, keeping track of the temperatures (and coefficient locations) is quite tricky and best understood if you sketch the matrix and vector structures out on a piece of paper (and compare with the discretization, Eq. 4). In my skeleton code, the first thing I do is set up a 2-D matrix that contains the vector index numbers of each temperature. This helps to map the temperatures into the vector (for the matrix inversion) and then back to a 2-D matrix (to plot).*

1. Again, assess the stability of the solution with respect to . How does this compare to the stability of the explicit scheme? Is this scheme faster (per timestep? Overall?).
2. **Calculating plume rise velocities:**

1. In Section 4, we will solve the advection equation for a plume that is rising through the mantle. Let’s first a simple calculation to estimate out how fast the plume should rise.

In class, we derived the Stokes velocity (in an approximate way) by balancing sphere buoyancy () with viscous drag () to get:

(5)

Using the plume parameters from Parts 1-2 (i.e., in the code) and Eq. 5 to estimate the rising velocity of the plume head. Use reasonable values for the parameters not contained within the code (i.e., and thermal expansivity).

1. If the plume is rising in a power-law mantle, is still proportional to ? To address this, again balance and . will not change, but will now incorporate (where is a scaling viscosity, as actual viscosity now depends on ).

What is the power-law form of the Stokes velocity? As a sanity check, verify that this form is equivalent to Equation 5 when plug in = 1.

1. **Explicit solution of the 2-D advection equation:**

We will now use the LAX method to implement plume advection. LAX is chosen as it is simple to implement while generally behaving better than FTCS. In 2-D, the advection equation is:

(6)

Discretizing this with the LAX strategy produces:

(7)

(8)

1. Download the skeleton Python script “2D\_Advection\_Lax\_Skeleton.py” and fill in the blanks (‘???’) to complete the code using the LAX method (Eqns. 7-8). Include plots showing the computed temperature evolution (the plume should rise towards the surface).
2. As before, vary the numerical parameters (e.g., use more/less grid points, larger/smaller time step). Calculate the relevant stability parameter for advection – the *Courant number*: – and describe how the solution changes with higher/lower . Is there a trade-off between two (undesirable) numerical effects? Settle on your preferred and plot this model evolution.
3. **Bonus question:**

Here, we have solved the 2-D heat and advection equations. To rigorously model plume dynamics, we need to combine the conduction and advection solutions, and also calculate a dynamically-consistent velocity by solving the conservation of momentum (i.e., Stokes Eq.) while ensuring mass is conserved (i.e., Continuity Eq.)

Considering these steps, outline the contents of a thermo-mechanical convection code that would be suitable to model this problem. Consider this to be the initial sketch of a code that you would (hypothetically) develop. Detail the solution steps/workflow (as an ordered list) and, for each step, the numerical strategy (e.g., discretization method).