Applied Numerical Computing for Scientists and Engineers

Dr. Ford Versypt

2D Heat Conduction GUI

1. In-class problem: MATLAB GUIDE

Download and install the app titled `2D Heat Conduction.mlappinstall` from the course bitbucket repository or website. The app feature is a quick way to package GUIs for sharing and installation. After installation, find the directory on your computer where the app's code is located. You will be asked to edit some of the code in this exercise.

1.1. Background

The app demonstrates two-dimensional conduction in a rectangular slab. The x and y dimensions are the same but can be scaled along with the Δx and Δy grid spacing. The α value is the thermal diffusivity of the material. Δt is the time step for the numerical method. The GUI was developed for junior level chemical engineering undergraduates learning about conduction and the finite difference method for solving ODEs. The first learning objective was for students to visualize the impact of changing the Fourier number (Fo) above its stable maximum of 0.25

$$\mathrm{Fo} = \frac{\alpha \Delta t}{(\Delta x)^2} \tag{1}$$

for $\Delta x = \Delta y$. All three terms in (1) can be manipulated in the GUI. The second learning objective was for students to simulate the effects of changing left, right, top, and bottom boundary conditions for the rectangular domain and the impact of changing the magnitude of the uniform initial condition.

This GUI was used as a homework assignment for CHE 3013 in Fall 2015 taught by Dr. Ford Versypt.

1.2. Challenge

On Oct. 11, 2016. Dr. Ford Versypt, Minu Pilvankar, and Ye Nguyen demonstrated GUIs from the research group and chemical engineering classes to middle school and high school students. Observations gleamed from using the heat transfer GUI in this context:

- While useful for undergraduate in a major course, the learning objective related to the Fourier number and numerical stability is beyond the scope of what can be explained to younger students, particularly in a 5 minute informal learning experience.
- The term "boundary condition" is jargon to middle school and high school students. A more approachable term for the constant temperature boundary condition uniform along a side of the rectangular domain is "edge temperature".
- We had a rectangular object at our booth. We had small pieces of paper for students to label the edges of the domain and make predictions. A similar virtual demonstration built into the GUI instead of a physical demonstration placed on the time may be more engaging while serving the same explanatory purpose.
- Temperature ranges of a qualitative type are desired for students to choose from: hot, warm, cool, cold instead of exact temperatures in °C. Also, American teenagers don't have intuition about Celcius temperatures.
- The current GUI uses a mesh plot with lines only along the spatial grids. A surface plot would be superior for being able to see the shape of the output more clearly and with less focus on the grid resolution.

1.3. In-class problem

Using GUIDE in MATLAB and the .m file, edit the GUI `2D Heat ... Conduction.mlappinstall` to meet the following objectives.

1. Change the mesh plot displayed to be a surface plot.

2. Remove the display of the Current Fourier Number and the Number of time intervals displayed in the plot.

3. Instead of editable text boxes for each edge temperature and initial condition, insert drop down menus for four options: Hot T = 100°F, Warm T = 80°F, Cool T = 60°F, Cold T = 40°F. When an option is selected, the corresponding boundary condition should be updated. Don't worry about unit conversions. We are theoretically just altering the material to maintain Fo = 0.25, and all the rest of the calculations are with respect to temperature differences.

1.4. Additional assignment requirements for 2016 CA5

In Fall 2016, CHE 5110 finished the in-class problem as Computational Assignment 5. Two sample student solutions are available as

- `CA5_minu_pilvankar_ANFV.mlappinstall`
- `CA5_Steve_Ruggiero.mlappinstall`

The following instructions were given to the students for modifying `2D ... Heat Conduction.mlappinstall` for their assignment.

Using GUIDE in MATLAB and editing the .m file for the GUI, edit the GUI to meet the following objectives. A good rule of thumb is to commit your changes after each item is completed. You do not have to work on the items in order, except item 1 should happen first chronologically and item 15 should occur last.

1. Commit the initial versions of the original GUI codes before you make any changes, so that you can go back to them in the future and so that differences between the original and your edited files are distinguishable easily on Bitbucket.

2. Remove the static and editable text boxes for Length of X = Length of Y, Alpha Value, Delta Time, Delta X = Delta Y. Note that the code still needs these quantities to run. They should just remain constant at their initialized values.

3. Remove the display of the Current Fourier Number and the Number of time intervals displayed in the plot. *You may want to simply hide the box that displays the number of time intervals so that the* `Counter_of_clicks` *variable can still function properly. Otherwise, you'll need to code a work around for that variable that is independent of displaying it in the static text box.*

4. Add a plot to the left side of the GUI that displays a 1x1 square. Use the built-in `rectangle` function with bottom left corner at the origin. Use the `FaceColor` option to set the inside initial temperature. The documentation for the `rectangle` function on the Mathworks website will be very helpful for completing this item.

5. Plot lines of `LineWidth` 3 and set the `Color` to visualize the edge temperatures for each edge separately. The lines should be plotted onto the figure of the square created in item 4.

6. Display a color key for your hot, warm, cool, cold temperatures and their values in Fahrenheit somewhere on your GUI. This might be text on the figures, a legend, colors appearing in the popup menu, or a set of 4 static text boxes with the temperatures and qualitative description ("hot") displayed as text and the text boxes colored to match the colors used on the figure of the square.

7. Arrange the boundary condition labels and boxes around the perimeter of the figure of the square adjacent to the corresponding sides (top with top, right on the right, etc.)

8. Arrange the initial condition label and box to be *above* the figure of the square.

9. Change the text "Boundary Condition" to "Edge Temperature" in all instances.

10. Change the text "Uniform Initial Condition" to "Initial Temperature Inside the Square"

11. Change the text on the button from "Calculate next time interval of 3 Delta t" to "Calculate the next time interval"

12. Add a button to reset the simulation to time zero even if the boundary conditions aren't edited (currently changing any editable field will result to time zero).

13. Instead of editable text boxes for each edge temperature and initial condition, insert **popup menus** (not a list, not a slider, but a popup menu) for four options: Hot T = 100°F, Warm T = 80°F, Cool T = 60°F, Cold T = 40°F. *When an option is selected, the corresponding boundary condition should be updated and the color of the corresponding edge or interior region should change.* Don't worry about unit conversions. We are theoretically just altering the material to maintain Fo = 0.25, and all the rest of the calculations are with respect to temperature differences.

14. Change the mesh plot displayed to be a surface plot.

15. Package a MATLAB app titled CA5_firstname_lastname for your completed assignment. Make sure to submit your .mlappinstall file on the course website, but it does not need to be tracked with version control.