1D Fouriers equation in Python

Files will be located in /Users/Nicholas/Documents/Python/Argon

Structure:

The Model is divided into different modules to keep it flexible and simple.

* F1D\_Parameters.py: stores the parameter values.
* F1D\_Grid.py: sets up a grid
* F1D\_BC\_Vector.py: calculates vectors defining the boundary conditions
* F1D\_Initial\_Condition.py: sets the concentration at each grid point.
* F1D\_Calculation.py: calculates and plots the final state of the system.
* F1D\_Animation.py: runs an animation of the concentration of argon as a function of time until the final timestep.
* F1D\_Enrichment\_Calc.py: animates two different concentration profiles that represent two different argon isotopes, and a third line that represents the enrichment along the x-axis. Also saves the enrichment to enrichment\_result.txt.

Run Notes:

These programs should be run in env1, but will work in the standard bash. F1D\_Calculation.py, F1D\_Animation.py, and F1D\_Enrichment\_Calc.py are the modules that should be run depending on the desired output. The animations are fast, but the loop that saves the results is slow and cannot be run for very small space and time steps. There are currently two unresolved bugs. One leads to a pathology in the second concentration line (u\_sec) near the right end of the system. The other has to do with the grid spacing, and leads to a mismatch in the size of arrays for certain ratios of nt and nx.

Current goals:

* Adjust the grid so that it is in the middle of cells rather than on the edges.
* Find the correct parameters for the experiment.
* Find the enrichment correction to the output of F1D\_Enrichment\_Calc.py.
* Make the edges “reflective” condition.
* Make sure that rounding errors do not change the total concentration, by using bill’s method to multiply the whole thing by a normalization factor to correct for the lost amount.
* Fix the edge pathology.
* Set up a “stop” when the concentration profile reaches the edge