NumPDE Final Project Write Up

Devon Kesseli
December 18th 2018

Part 1: Firedrake

My original plan was to use Firedrake FEM to model a phase change material. I was planning on starting with a Rayleigh-Benard model (provided in a demo), and add a Temperature dependent viscosity term. This proved a lot harder than I expected. The major challenges and problems I came across are documented below.

Installing Firedrake

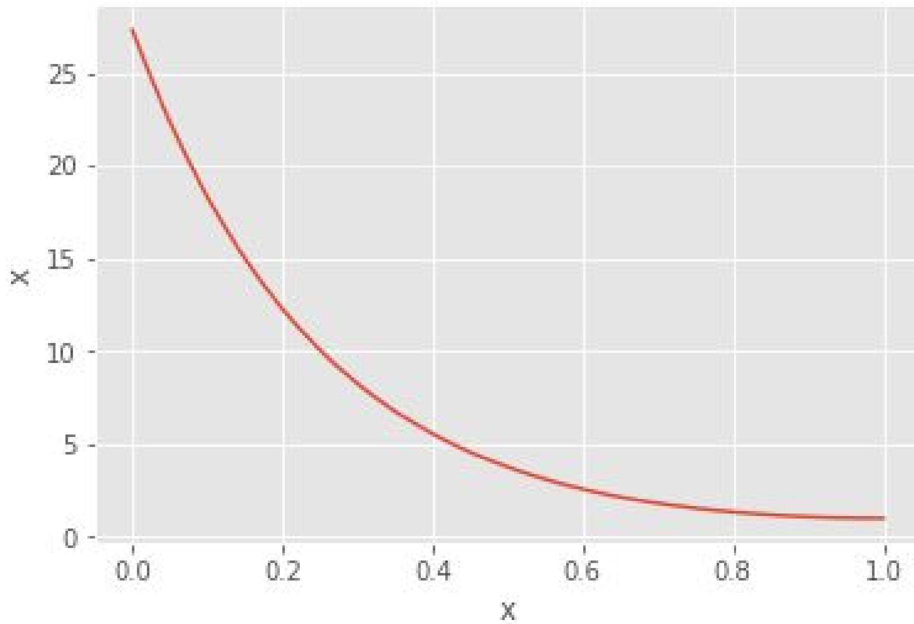
This was a struggle on my Mac computer. Partly because my PETSc install was using the Anaconda distribution of python, which apparently doesn't work in the Firedrake virtual environment. Even after realizing this, many hours of troubleshooting were required. Eventually I collected all of the versions of Python I needed, and reinstalled my faulty fortran compiler, and installed a lot of other required pieces. And behold! The demo worked!

Using Firedrake

One of the biggest challenges here was learning how to do things in the Firedrake environment. Even after a few tutorials, I still don't feel very comfortable manipulating the UFL VectorFunctionSpace and FunctionSpace values. I initially had wanted to do a time-dependent model, and made a few stabs at this with the results in this paper: (https://aip.scitation.org/doi/pdf/10.1063/1.5012653).

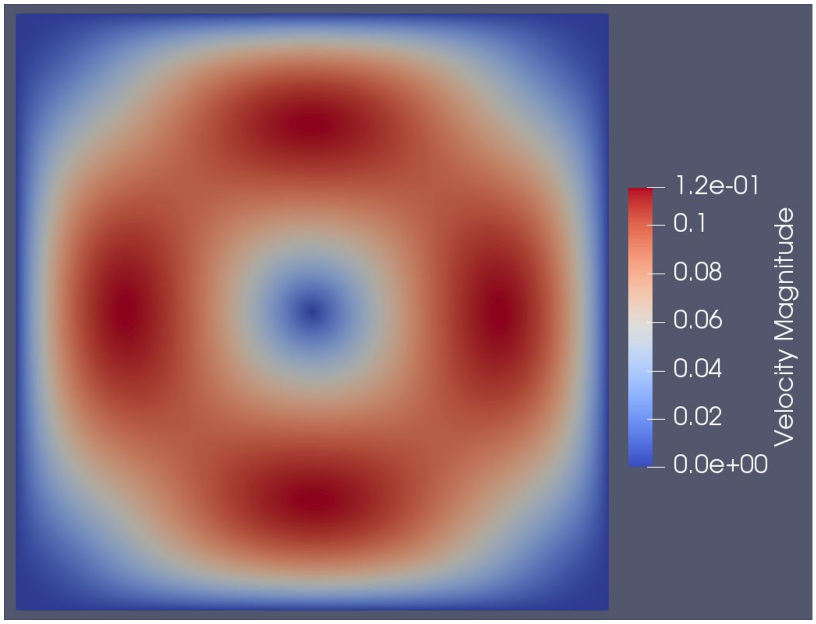
Making a Phase Change

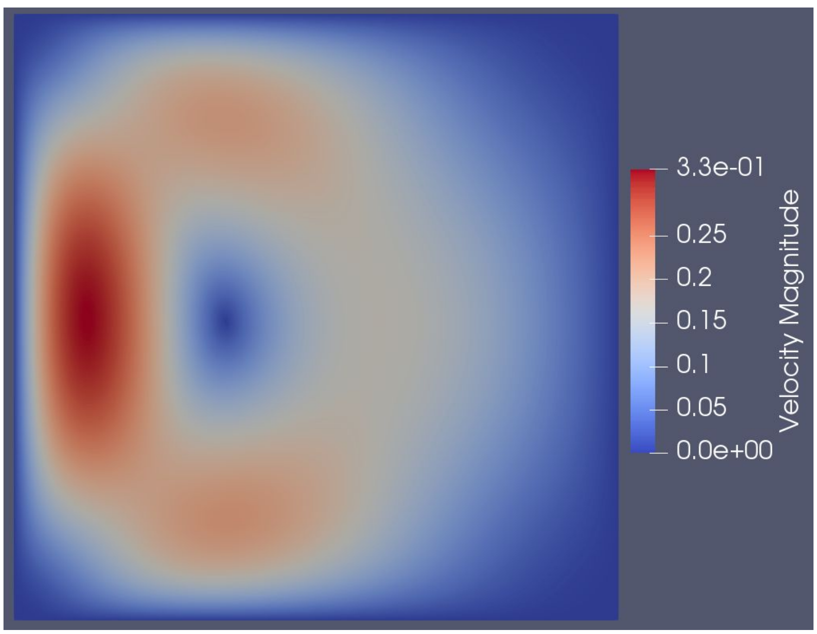
After getting Part 2 done, I took another stab at Firedrake with more reasonable goals. I wanted to add in a temperature dependent viscosity. This is represented in the Prandtl Number (Pr), which is proportional to viscosity. In the example, Pr is set to 6.8. I essentially wanted a function of T that is very high for low T values (symbolizing a pseudo-solid section), and around 6.8 for lower T. I also needed to use operations I could apply to T, which is a UFL FunctionSpace on the 128x128 mesh. After some trial and error, I decided on cosh. As temperature is between 0 and 1, Something like cosh(4*T - 4) gave me an adequate function:

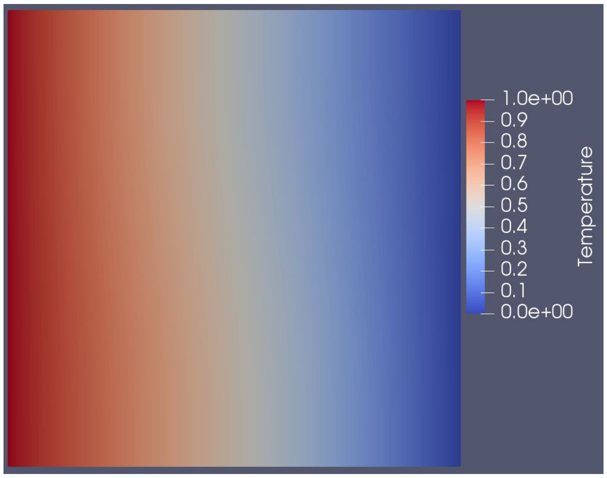


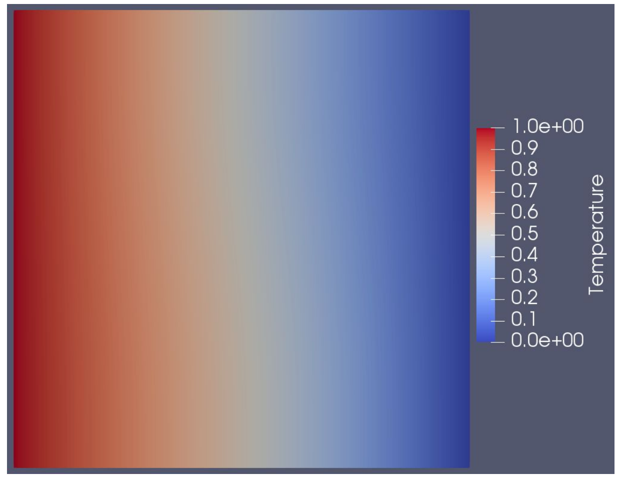
I scaled up the T=1 end to 6.8. Multiplying each term by the new temperature-scaled Pr gave me an unconverging solution, but adding a constant Pr to two of the terms allowed it to converge quickly while still showing the qualitative behavior I was looking for. I decided to stick with the direct solver from the example, since the fieldsplit preconditioner and recursive fieldsplitting methods (also included in the example).

The model is a 2-dimensional steady-state Rayleigh-Benard system, using the the Boussinesq approximation for a buoyant term. In this system heat the left wall is heated, and the right wall cooled. The Below, the velocity magnitude with a constant Pr and temperature dependent Pr are shown. Both show convection, but on the cold-wall side (right), the much-higher Prandtl number leads to low or no velocity. The result is the doughnut being squished against the left wall. A sharper Pr change would be more representative of a real solid/liquid interface, and look like a more cleanly compressed circle, instead of the stretched right side shown here.

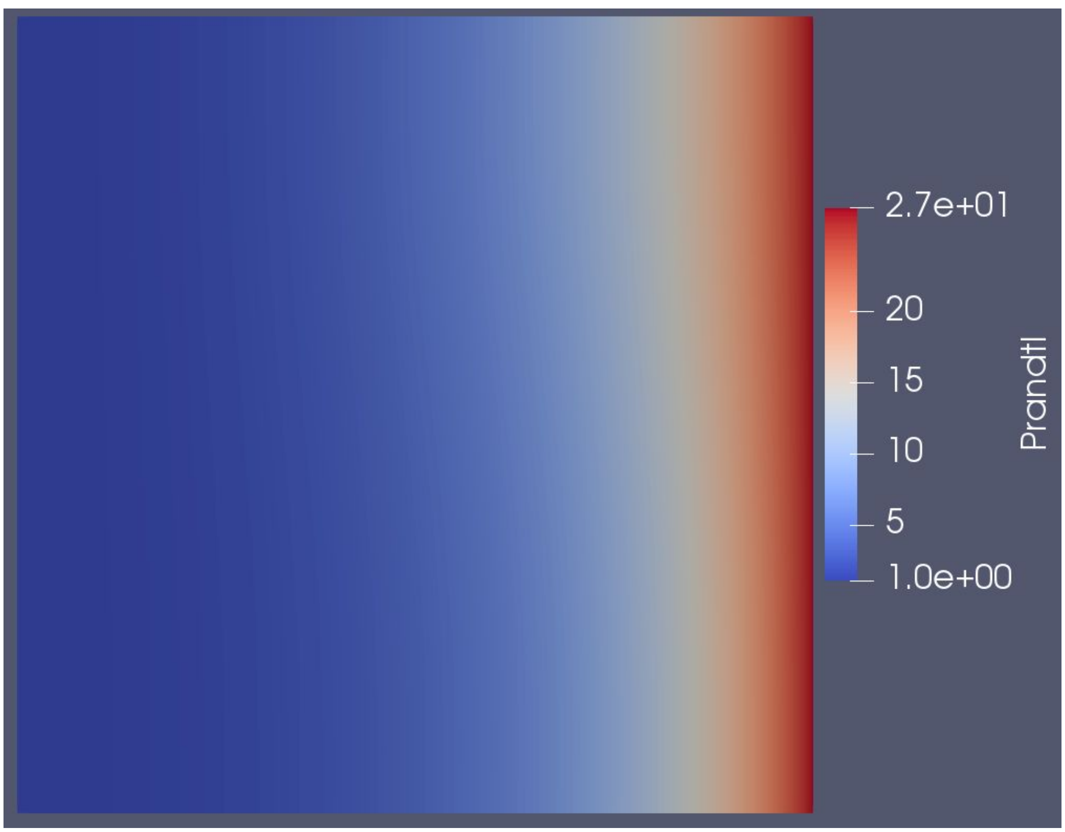

1.2e-01
0.1
0.08
0.06
0.04
0.02
0.0e+00
Velocity Magnitude


3.3e-01
0.25
0.2
0.15
0.1
0.05
0.0e+00
Velocity Magnitude





Temperature results are non-dimensional, set from 0 to 1, so are essentially the same on both cases. Below, the temperature dependent Prandtl number is plotted. The ideal case would be a blue, liquid section on the left, and evenly red solid section on the right. The transition resulting from the cosh function is much more gradual, but I think the Prandtl number does become inhibitively high in the right third of the domain.



Part 2: 1D Model

At some point I realised that becoming a proficient Firedrake user over the course of this project was more work than I had hoped, and a bit naive. After spending a long time getting it running, doing a few tutorials, and attempting to manipulate the demo problem to do what I wanted, I had to come to terms with the fact that it wasn't going to work. I decided to do a supplemental finite difference test to look at some of the phase-change phenomena I wasn't going to be able to capture in Firedrake.

The write up for this part is in the Jupyter Notebook in the project folder.

References

**Format

Papers

Labrosse, S., Morison, A., Deguen, R., & Alboussière, T. (2018). Rayleigh–Bénard convection in a creeping solid with melting and freezing at either or both its horizontal boundaries. Journal of Fluid Mechanics, 846, 5-36. doi:10.1017/jfm.2018.258

Nasreddine Ouertatani, Nader Ben Cheikh, Brahim Ben Beya, Taieb Lili,
Numerical simulation of two-dimensional Rayleigh–Bénard convection in an enclosure,
Comptes Rendus Mécanique,
Volume 336, Issue 5,
2008,
Pages 464-470,

Software

(1, 2) Andrew T. T. McRae, Gheorghe-Teodor Bercea, Lawrence Mitchell, David A. Ham, and Colin J. Cotter. Automated generation and symbolic manipulation of tensor product finite elements. *SIAM Journal on Scientific Computing*, 38(5):S25–S47, 2016. URL: http://arxiv.org/abs/1411.2940, arXiv:1411.2940, doi:10.1137/15M1021167.

The FEniCS Project Version 1.5 M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes and G. N. Wells Archive of Numerical Software, vol. 3, 2015, [DOI]

@article{abhyankar2018petsc,
author={Abhyankar, Shrirang and Brown, Jed and Constantinescu, Emil M and Ghosh, Debojyoti and Smith, Barry F and Zhang, Hong},
title={PETSc/TS: A Modern Scalable ODE/DAE Solver Library},
journal={arXiv preprint arXiv:1806.01437},
year={2018}
}