NumPDE Final Project Write-Up

Devon Kesseli December 18, 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 (<u>provided in a demo</u>), 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, a lot of troubleshooting were required. Eventually I collected all the versions of Python I needed, 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 attempts at this with the results in this paper: (https://aip.scitation.org/doi/pdf/10.1063/1.5012653), to no avail.

Making a Phase Change

After getting Part 2 (below) 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:

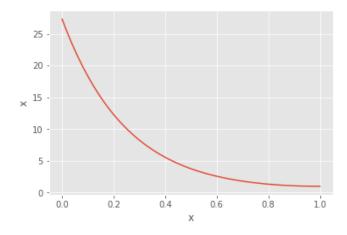


Figure 1:y = cosh(4x-4), my temperature dependent Prandtl number function (temperature ranges from 0 to 1).

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) took about the same amount of time.

The model is a 2-dimensional steady-state Rayleigh-Benard system, using the Boussinesq approximation for a buoyant term. In this system the left wall is heated, and the right wall cooled. The two plots below show the velocity magnitude with a constant Pr followed by the temperature dependent Pr. 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.

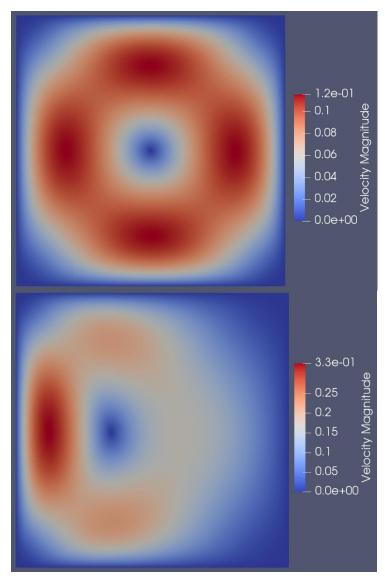
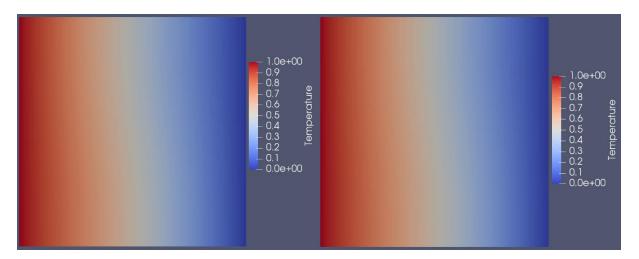


Figure 2: Resulting velocity magnitude plots for the constant Prandtl number (top) and temperature-dependent Prandtl number (bottom). In both cases the left wall is heated and right wall cooled.



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.

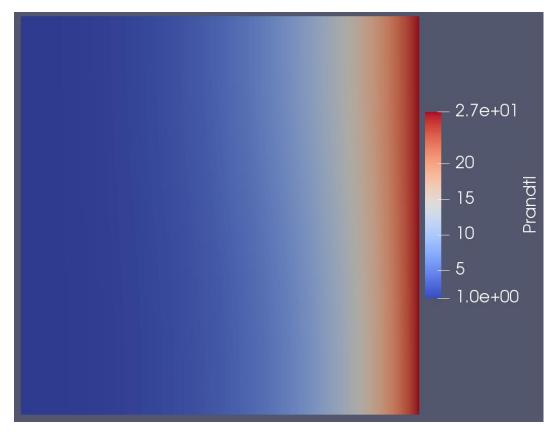


Figure 3: A plot of the temperature dependent Prandtl number

Part 2: 1D Model

At some point I realized that becoming a proficient Firedrake user over the course of this project was a bit naïve, and I had to come to terms with the fact that it all 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

- [1] S. Labrosse, A. Morison, R. Deguen and T. Alboussière, "Rayleigh—Bénard convection in a creeping solid with melting and freezing at either or both its horizontal boundaries," *Journal of Fluid Mechanics*, vol. 846, no. 5-36, 2018.
- [2] S. Abhyankar, J. Brown, E. M. Constantinescu, D. Ghosh, B. F. Smith and H. Zhang, "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," *arXiv preprint arXiv:1806.01437*, 2018.
- [3] A. T. T. McRae, G.-T. Bercea, L. Mitchell, D. A. Ham and C. J. Cotter, "Automated generation and symbolic manipulation of tensor product finite elements," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, p. S25–S47, 2016.
- [4] O. Nasreddine, B. C. Nader, B. B. Brahim and L. Taieb, "Numerical simulation of two-dimensional Rayleigh–Bénard convection in an enclosure," *Comptes Rendus Mécanique*, vol. 336, no. 5, pp. 464-470, 2008.
- [5] S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes and G. N. Wells, "The FEniCS Project Version 1.5 M.," *Archive of Numerical Software*, vol. 3, 2015.