]Rayleigh Hackathon 2021 Report

Introduction

To further develop the spherical convection/dynamo code Rayleigh and to grow and foster its open source community, 15 user-developers of Rayleigh worked in a virtual hackathon over a 7 day period. Due to the ongoing Covid-19 pandemic, this year's workshop was held virtually, using the Wonder and Slack collaboration tools. While the remote aspect of this workshop offered new challenges for what is normally a very in-person, collaborative working experience, several improvements to the source code, documentation, and supporting analysis routines were completed and/or initiated.

Some efforts begun at the workshop have yet to make it into the main branch but are now in their near-complete stages. This includes the addition of new functionality in Rayleigh allowing *multiple additional scalar fields* to be evolved. These new fields act analogously to the temperature field and may be used for studies exploring the convection of a multi-compositional fluid. We also introduced the ability to model a *conducting inner core*, such as that which lies below the Earth's molten outer core and which may impact the long-term evolution of Earth's magnetic field.

Several code additions came to fruition during the workshop, such as the addition of a new spectral derivative library, new plotting routines for the display of spherical data, cleanup of source code through the removal of unused variables, and various bug fixes.

A separate major outcome of the workshop was the preparation of a Rayleigh 1.0 release. Additionally, an informal working group was formed to design a new, more user-friendly interface to Rayleigh's custom-reference state functionality. That group (composed of Hindman, Korre, Matilsky & Blume) has corresponded regularly over email and held two virtual meetings since the completion of the workshop.

During the course of the hackathon, almost every participant contributed source code to the project. Together, users and developers changed a total of more than 500 lines of source code. This number is expected to change as the additional-scalar and conducting-core functionality have not yet been merged into the code.

Below is the timeline and a log of the individual contributions. Many of these contributions are discussed in greater detail following the table of participants' interests.

Timeline

Day	**Scheduled items**
Monday, 05/24	12 pm Mountain: Git tutorial

Tuesday, 05/25	Noon discussion: Version 1.0 and versioning. Citation/Code Authorship
Wednesday, 05/26	Docker: Rene Presentation: MariaCamisassa, Peter Driscoll & Cian Wilson (scalar fields)
Thursday, 05/27	Discussion about future development plans Presentation: Rene Gassmoeller
Friday, 05/28	

Participants and areas of interest

Name, affiliation, email	**Goals and interests for this hackathon**
Rene Gassmoeller, UC Davis, rene.gassmoeller@mailbox.org	1. Help others with their goals 2. Review pull requests 3. Migrate tester 4. Improve documentation structure
Lorraine Hwang UC Davis ljhwang@ucdavis.edu	1. Logistics 2. Documentation
Nick Featherstone Southwest Research Institute nicholas.featherstone@colorado.edu	1. Introducing people to Rayleigh's design 2. Helping others
Cian Wilson Carnegie Science cwilson@carnegiescience.edu	1. Goal: Adding (at least one) extra scalar field 2. Interested: Conductive inner core 3. Interested/Goal: Post-processing scripts 4. Interested: Namelist lists
Ryan Orvedahl UC Davis ryan.orvedahl@colorado.edu	1. Add post-processing library to do spectral transforms/derivatives 2. Review pull requests
Maria Camisassa CU Boulder maria.camisassa@colorado.edu	1. Learning about how Rayleigh is programed 2. Improve documentation in a way that it would make it easier for new users 3. Review pull requests
Philipp Edelmann	1. Designing a new post-processing Python

Los Alamos National Laboratory pedelmann@lanl.gov	framework using memory mapping 2. Docker based development environment 3. Improve the scripts to use stellar models from MESA as custom reference states
Krista Soderlund UT Austin krista@ig.utexas.edu	1. Help implement and test the addition of a conducting inner core
Lydia Korre LASP, CU Boulder lydia.korre@lasp.colorado.edu	1. Update post-processing files with newer and more efficient routines/examples 2. Improve documentation
Anne Hedlund NMSU/LANL ahedlund@nmsu.edu	1. Documentation: New user, machine specific, Jupyter notebooks
Catherine Blume CU Boulder catherine.blume@colorado.edu	1. Help rewrite the custom reference states user interface
Bradley Hindman CU Boulder hindman@colorado.edu	1. Help rewrite the custom reference states user interface 2. Compile and document a list of all the possible namelist variables

Resources

Git Tutorial:

- The slides from the first day presentation: https://www.dropbox.com/s/6xvb4pyq7mefxp7/Git-Github-introduction.pdf?dl=0
- Git commands cheat sheet: https://education.github.com/git-cheat-sheet-education.pdf
- Github workflow: https://guides.github.com/introduction/flow/
- Git tutorial: https://swcarpentry.github.io/git-novice/

1. Explain and set up Git:
   a. https://swcarpentry.github.io/git-novice/01-basics/index.html
   b. https://swcarpentry.github.io/git-novice/02-setup/index.html
2. Explain Github Workflow:
   a. https://guides.github.com/introduction/flow/
   b. Ensure forked repositories
   c. Ensure proper remotes
3. Walkthrough
   a. Create Branch
      i. ‘git checkout master’
      ii. ‘git pull upstream master’
      iii. ‘git checkout -b remove_dealii_compatibility_fix’
   b. Create commit
      i. ‘git add FILE’
      ii. ‘git commit -m ‘A short message describing the change’
   c. Push and open PR
      i. ‘git push origin remove_dealii_compatibility_fix’
      ii. Open PR on github (CTRL-Click on shown link)
   d. Wait for review
   e. Address review (repeat steps b,c,d)
   f. Success!

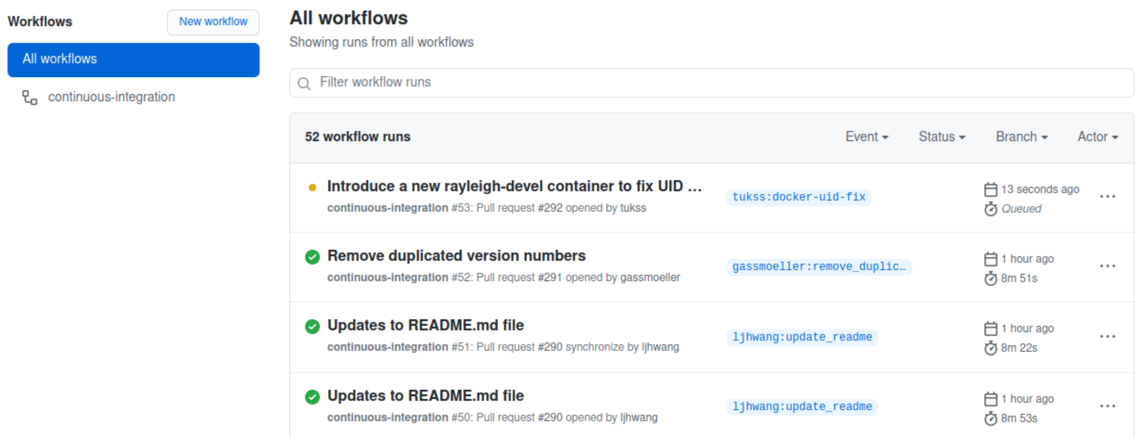
Now repeat the steps in 3. on your own.

Report on projects the participants worked on

Build a Github Action that replaces the official Jenkins tester

Rene Gassmoeller

Rayleigh used the CIG Jenkins tester for it’s CI, but perspectively we want to move to Github Actions or some other hosted CI service to improve the reliability of the system. I generated a Github action workflow that executes the identical tests as the Jenkins tester as a Github workflow.



52 workflow runs

Introduce a new rayleigh-devel container to fix UID …
continuous-integration #53: Pull request #292 opened by tukss
tukss:docker-uid-fix
13 seconds ago
Queued

Remove duplicated version numbers
continuous-integration #52: Pull request #291 opened by gassmoeller
gassmoeller:remove_duplic…
1 hour ago
8m 51s

Updates to README.md file
continuous-integration #51: Pull request #290 synchronize by ljhwang
ljhwang:update_readme
1 hour ago
8m 22s

Updates to README.md file
continuous-integration #50: Pull request #290 opened by ljhwang
ljhwang:update_readme
1 hour ago
8m 53s

Figure 1: Screenshot of the new workflow in Github actions that may be used to test changes to Rayleigh source code provided via new pull requests.

Automatically build and test the Rayleigh conda environment

Rene Gassmoeller

Rayleigh advertises a set of conda packages as an official development environment, but it did not have a unified environment.yml file that officially defines this environment. I created such a file and added a CI workflow that automatically builds the environment to make sure that the file is up-to-date and working.

Add a scalar field

Cian Wilson, Nick Featherstone, Peter Driscoll

In order to track composition or other tracers Rayleigh needs to solve extra scalar advection-diffusion equations. We worked on this by updating a preliminary branch created during the last hackathon. This mostly involved modifying that work for the more recent additions of custom reference states and generic input and bcs. It also required some work related to checkpointing.

Currently we have implemented and tested a passive scalar field and are working on making this active (coupling it to the buoyancy). We also have plans to generalize it beyond a single scalar field, make it optional and attempt to couple the boundary conditions with other fields.

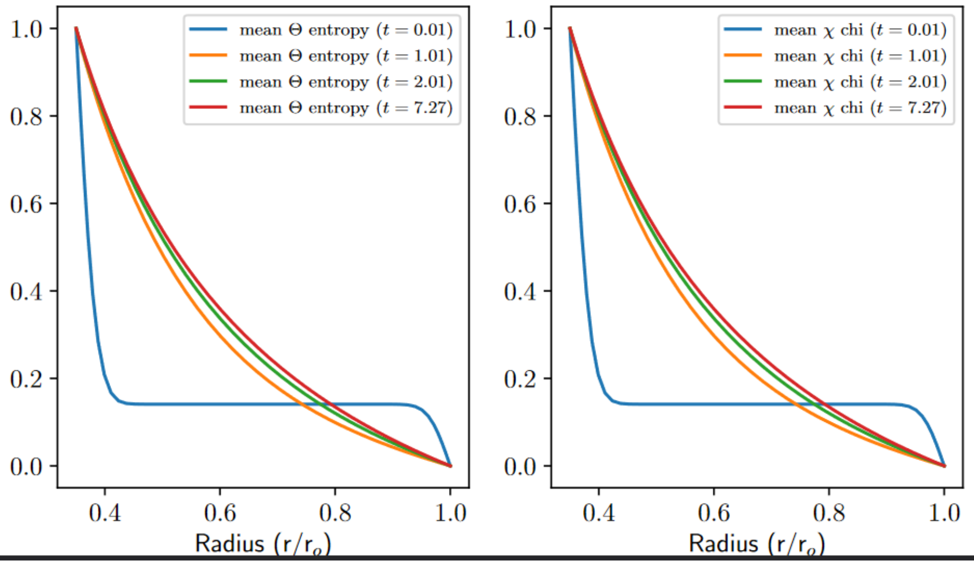


Figure 2: Testing Rayleigh's new additional-scalar-field functionality. (*Left*) Entropy field, as a function of radius, averaged over spherical surfaces, at several discrete times during a Rayleigh simulation. (*Right*) Companion scalar field, displayed in a similar fashion, at the same times. Both fields were run in passive mode (i.e., neither contributed to the buoyancy), and show identical evolutionary behavior, indicating that this functionality is working as expected.

Fix Intel Related Seg Fault

Cian Wilson, Rene Gassmoeller, Philipp Edelmann, Nick Featherstone

When using intel, copying certain arrays resulted in a seg fault. It turned out that this behavior was caused by intel allocating temporary arrays on the stack, resulting in a seg fault (expected behavior according to the compiler documentation, but not for a typical programmer!) when it ran out of stack space. The default compiler flags were modified to allocate on the heap above 65 kilobytes (mimicking the gnu compiler behavior), thus preventing the seg fault.

Fix some uninitialized variables

Cian Wilson

Using the `-check all` intel compiler flag highlighted some uninitialized variables that were being used. These were fixed by setting some (hopefully sensible) default values to them.

Python library to perform spectral derivatives

Ryan Orvedahl

In a post processing step, it is often necessary to compute derivatives and interact with the Rayleigh grids. During the hackathon, a Python module was created that allows one to compute the grids in Rayleigh and perform transforms to spectral space where derivatives can be computed. Support for Fourier, Legendre, and Chebyshev polynomials is included.

Prepare Rayleigh 1.0.0 release

Rene Gassmoeller, Nick Featherstone, and everyone participating

We prepared Rayleigh's repository for the 1.0.0 release. This in particular included fixing compiler crashes/warnings, updates to the documentation in particular for how to cite Rayleigh, automated detection of version numbers, an update to the pdf manual, improved documentation for installation. At the end of the hackathon only minor issues remained and the release was essentially prepared to be created in the week after the hackathon.

Simplify the interface for using custom reference states

Loren Matilsky, Catherine Blume, Brad Hindman

The current way in which custom reference states are implemented is fully functional, but somewhat clunky. Our goal is to redesign the namelist interface in the main_input file in such a way that is easier to use and more intuitive. Loren, Catherine, and Brad have discussed and planned how we would like the interface to operate in broad strokes. Brad will be writing up a succinct description of the plan and will make it available to the other Hackathon attendees to seek feedback. The current plan is to build and implement the new interface during the (optional) second week of the Hackathon.

Conducting Inner Core Functionality

Nick Featherstone, Krista Soderlund, Cian Wilson, Peter Driscoll

Beneath the Earth's molten, convective outer core lies a solid iron inner core. While this region of the Earth does not harbor a convectively unstable fluid, it is expected to be electrically conducting. The interiors of Uranus and Neptune may contain thick layers of superionic ices, which are thought to behave like a solid and have significant electrical conductivity, below their convecting, ionic oceans. Magnetic fields generated in the outer core (ionic ocean) will permeate this lower conducting region, which may in turn influence the evolution of the magnetic field in the outer core. The interaction between these two regions was not previously captured by Rayleigh simulations. Using Rayleigh's multiple Chebyshev domain functionality, we have added the ability to simulate the solid inner core for the first time within Rayleigh. This is accomplished by solving the fully nonlinear MHD equations in one (upper) Chebyshev domain, and matching that solution at its inner boundary to the second (inner) Chebyshev domain. Within that domain, a reduced set of physical equations that omit nonlinear advection are solved. Both magnetic field and temperature (shown below) can be evolved in this fashion. Next steps are to add rotation to the inner core, if desired, as well as an even-mode Chebyshev expansion in that region. That latter will allow us to avoid issues related to the coordinate singularity at r=0.

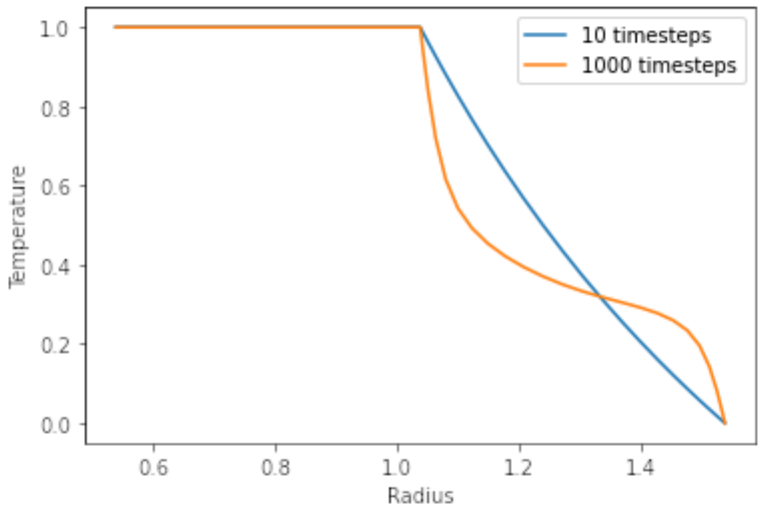


Figure 3: Testing conducting inner core functionality. Mean temperature (averaged over spherical shells) is shown as a function of radius at two different time steps in a Rayleigh simulation. In this test, the heat was not diffused in the inner core, but was diffused and advected as normal in the outer core. The temperature distribution in the solid inner core remains constant throughout the run as a result (indicating that this particular test of the new functionality was passed).

Rayleigh Dockerfile Improvements and Documentation

Rene Gassmoeller, Philipp Edelmann, Nick Featherstone

Several improvements to Rayleigh's Dockerfile were implemented. This is a work in progress, but through the workshop, we nearly arrived at a 'complete' Docker configuration script. This setup will allow a user to compile Rayleigh and the documentation (as existed before the workshop) while also allowing providing a working analysis environment within with Jupyter notebooks may be run. In addition, documentation for setting up the Rayleigh Docker image was added to Rayleigh's online documentation.

Updated Diagnostics_Plotting.ipynb post-processing notebook

Lydia Korre, Maria Camisassa

We replaced the old Basemap-based script in Diagnostics_Plotting.ipynb with the new orthographic projection-based code for plotting on spherical surfaces. The new example shows the three velocity components plotted on spherical surfaces (using the Shell_Slices output) at different latitudinal vantage points, colormaps, etc. Also we added the projection.py script needed for this, as well as the original notebook "plot_shells.ipynb" which has more details on plotting options, quantities, etc. (written by Nick Featherstone)

2021 Statistics about Rayleigh's growth during the hackathon

The following contains a number of statistics about how much Rayleigh has grown during the hackathon (between May 24 2021, commit fe54ec54 and Jun 2 2021, commit 13ec3cafd to allow for late merges):

Number of source files in Rayleigh before/after:	149 -> 151	+2
Lines of code in Rayleigh before/after:	55612 -> 56156	+544
Number of merged pull requests before/after:	240 -> 273	+33
Commits in github before/after:	755 -> 820	+65
Number of tests before/after:	6 -> 6	+0

These statistics were generated through the following commands:

```
find ./ | egrep '\.(F|F90|c|py|ipynb)$' | wc -l
cat `find ./ | egrep '\.(F|F90|c|py|ipynb)$'` | wc -l
git log --format=oneline | grep "Merge" | wc -l
git log --format=oneline | grep -v "Merge" | wc -l
```

- Tests were manually counted