

Rayleigh Hackathon 2022 Report

Logistics	2
Introduction	3
Land Acknowledgement	4
Timeline	5
Participants and areas of interest	5
Resources	7
Git Tutorial:	7
Report on projects the participants worked on	8
Modernize the design of Rayleigh's documentation page	8
Rene Gassmoeller	8
Restructure Rayleigh's documentation	8
Nick Featherstone, Nick Nelson, ..., Rene Gassmoeller	8
Improvements to Python Analysis Routines	9
Nicholas Nelson, Philipp Edelmann	9
Exploring an Alternative Chebyshev Formulation	9
Kyle Augustson	9
Toward open INCITE data to public	10
Hiroaki Matsui	10
Rayleigh Co-Author Network	10
Lorraine Hwang	10
Rename Rayleigh's default branch name to 'main'	12
Rene Gassmoeller, Philipp Edelmann, Nick Featherstone	12
Testing the extra scalar field on Rayleigh (Boussinesq version)	12
Maria Camisassa, Rafael Fuentes, Cian Wilson, Nick Featherstone	12
Allowing arbitrary numbers of extra active and passive scalar fields	13
Cian Wilson, Nick Featherstone, Maria Camisassa, Rafael Fuentes	13
Bug fixes	13
Everyone	13
2021 Statistics about Rayleigh's growth during the hackathon	14

Logistics

The hackathon occurs in person in Breckenridge, Colorado. Participants that have to self-isolate due to Covid will use the following Zoom Link:

<https://ufl.zoom.us/j/94646279339?pwd=SzhoUIFwSUNxWmZOTFRZMGtnTGVIZz09>

Slack:

https://join.slack.com/t/rayleighworkspace/shared_invite/zt-1fy7the9g-l80zezN~0bf64mFp2vKRiW

Logistics spreadsheet:

<https://docs.google.com/spreadsheets/d/1ugQTVp4ZR1K-ptcSg7bBgIEKMuJi4TjGq4jqSuV0sfM/edit#gid=0>

Rental agreement:

https://drive.google.com/drive/u/0/folders/1um2Ox_TMsHnBig3ctfAQz9SsWf6PIfQs

Dataset Listing:

<https://osf.io/j275z/wiki/Dataset%20Listing/>

Link to last year's report:

https://geodynamics.org/resources/1929/download/Rayleigh_Hackathon_2021_Log.pdf

A Miro Board for Documentation Flowcharting/Brainstorming:

<https://app.conceptboard.com/board/ksgb-gmck-7011-phmc-2ysh>

Notes on Documentation Discussion

https://docs.google.com/document/d/17TthEITd0y4WZw0uriRj96YpolaKP6u-Cu6xi30y_gQ/edit?usp=sharing

Documentation Outline

<https://docs.google.com/document/d/1qxnBepvC8UMeh98dMDcbBtHp8yIC9YM4RXpJREB1Dms/edit?usp=sharing>

Introduction

For the 2022 Rayleigh hackathon, 11 user-developers of Rayleigh worked in an in-person hackathon over a 6-day period. In contrast to past hackathons, which were held on the campus of the University of Colorado, or remotely due to Covid-19, this year's hackathon was held in Breckenridge, CO. Moose and fox citings offered a welcome change of scenery from that otherwise offered by the windowless, basement meeting room used for prior workshops! Several improvements to the source code, documentation, and supporting analysis routines were completed and/or initiated during this year's week-long workshop.

One major focus of this year's workshop was to revisit the structure and content of Rayleigh's online documentation. Rayleigh's online documentation began as an online index of the output quantity codes. From there, it grew piecewise over the years, incorporating sections from various early Rayleigh user manuals in the process. As a result, the design of the webpage was not particularly user-friendly, and its content was not always organized in a logical fashion. This year's workshop addressed these two outstanding issues. The webpage design was revised, and it now has a more modern appearance and user-friendly feel. A significant reorganization of existing content was also undertaken, and a plan was constructed to fill in those pieces that are missing.

In terms of the source code, one major effort of this year's workshop was the addition of new functionality in Rayleigh allowing *multiple additional scalar fields* to be evolved. This functionality was originally planned during the 2021 workshop, and is essentially complete (the lead developer, Cian Wilson is tracking down the source of one outstanding bug at the time of this writing). These new fields act analogously to the temperature field and may be used for studies exploring the convection of a multi-compositional fluid. Such systems are of wide interest across the full spectrum of Rayleigh's user base, with applications in the Earth's core, planetary interiors, and stellar interiors.

Several other efforts came to fruition during the course of the workshop, such as the organization of INCITE data and its preparation for publication on the Rayleigh documentation page. Moreover, a more user-friendly python interface, capable of easily dealing with a series of output files, was added to the Rayleigh repository, along with documentation.

Finally, with an eye toward the future, the team began exploring an alternative formulation for the radial discretization scheme employed in Rayleigh. This alternative formulation employs Chebyshev polynomials of the 2nd kind and holds the potential to offer enhanced accuracy in the radial derivatives. It can also be used to reformulate the implicit time-stepping matrices such that their structure is sparse, rather than dense as in the current scheme. The team will continue to assess the utility of this new approach throughout the coming months.

The new additions to the codebase resulting from this workshop will be wrapped into the Rayleigh 1.2 release later this fall. 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.

Land Acknowledgement

We acknowledge the indigneous people and land in which we are gathered. Breckenridge, Colorado has been home to the Núu-agma-tʼv̥v̥-p̥ʼ (Ute) peoples who have provided stewardship of this land over many centuries. We are honored and grateful to be here today on their traditional lands.

Timeline

Day	Scheduled items
Sunday 09/11	Arrival, Introductions
Monday, 09/12	9 am: Morning rounds 10 am: Git + pull request tutorial
Tuesday, 09/13	9 am: Morning rounds 10 a.m. Documentation 11 a.m. INCITE 1 pm: Intro to new Python post processing 3 pm: Small group documentation reorganization
Wednesday, 09/14	9 am: Morning rounds
Thursday, 05/27	9 am: Morning rounds Afternoon: Hike / Half-day off Dinner on your own
Friday, 05/28	9 am: Morning rounds 10 am: Forums/Slack + renaming Rayleigh's default branch (Rene)
Saturday, 05/29	Departure

Participants and areas of interest

Name, affiliation, email	Goals and interests for this hackathon
Rene Gassmoeller, University of Florida rene.gassmoeller@mailbox.org	<ol style="list-style-type: none">1. Help others with their goals2. Review pull requests3. Improve documentation structure4. Create docker and singularity container5. Test non-homogeneous boundary conditions
Lorraine Hwang UC Davis ljhwang@ucdavis.edu	<ol style="list-style-type: none">1. Logistics2. Documentation3. INCITE Data

<p>Nick Featherstone Southwest Research Institute nicholas.featherstone@colorado.edu</p>	<ol style="list-style-type: none"> 1. Introducing people to Rayleigh's design 2. Helping others
<p>Cian Wilson Carnegie Science cwilson@carnegiescience.edu</p>	<ol style="list-style-type: none"> 1. Goal: Interface to scalar fields 2. Goal: Improving generic input
<p>Nicholas Nelson California State University, Chico njonelson@csuchico.edu</p>	<ol style="list-style-type: none"> 1. Develop 3D rendering pipeline 2. Re-familiarize with Rayleigh
<p>Philipp Edelmann Los Alamos National Laboratory pedelmann@lanl.gov</p>	<ol style="list-style-type: none"> 1. Scalar fields for composition 2. Finish new Python post processing 3. Provide more examples for custom reference states from MESA
<p>Daniele Thallner University of Florida d.thallner@ufl.edu</p>	<ol style="list-style-type: none"> 1. Reproduce a model run with LSD code 2. Compare Rayleigh and LSD speed/results 3. (Use QPM to score Rayleigh output)
<p>Kyle Augustson Northwestern University kyle.augustson@northwestern.edu</p>	<ol style="list-style-type: none"> 1. Implement sparse methods for the implicit solve. 2. Try other timesteppers (RK222, SBDF2) 3. Look at the performance of LU solvers.
<p>Hiro Matsui UC Davis CIG hrmatsui@ucdavis.edu</p>	<ol style="list-style-type: none"> 1. Documentation for INCITE data 2. Data conversion tools for INCITE data for the current release version.
<p>Maria Camisassa University of Colorado Boulder maria.camisassa@colorado.edu</p>	<ol style="list-style-type: none"> 1. Goal: Test the new version of Rayleigh with the scalar field and try to use it 2. Help on improving the documentation
<p>Rafael Fuentes University of Colorado Boulder jofu5477@colorado.edu</p>	<ol style="list-style-type: none"> 1. Goal: Learn to use the Rayleigh code 2. Test the new version of Rayleigh including an extra scalar field that represents a compositional gradient.

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 fix_version_number'
 - 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 fix_version_number'
 - 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

Modernize the design of Rayleigh's documentation page

Rene Gassmoeller

The Sphinx documentation system that Rayleigh uses to generate its documentation offers different themes that improve the design of the online documentation. Using experiences from other CIG projects we have improved the design of Rayleigh's documentation using a better and more modern theme.

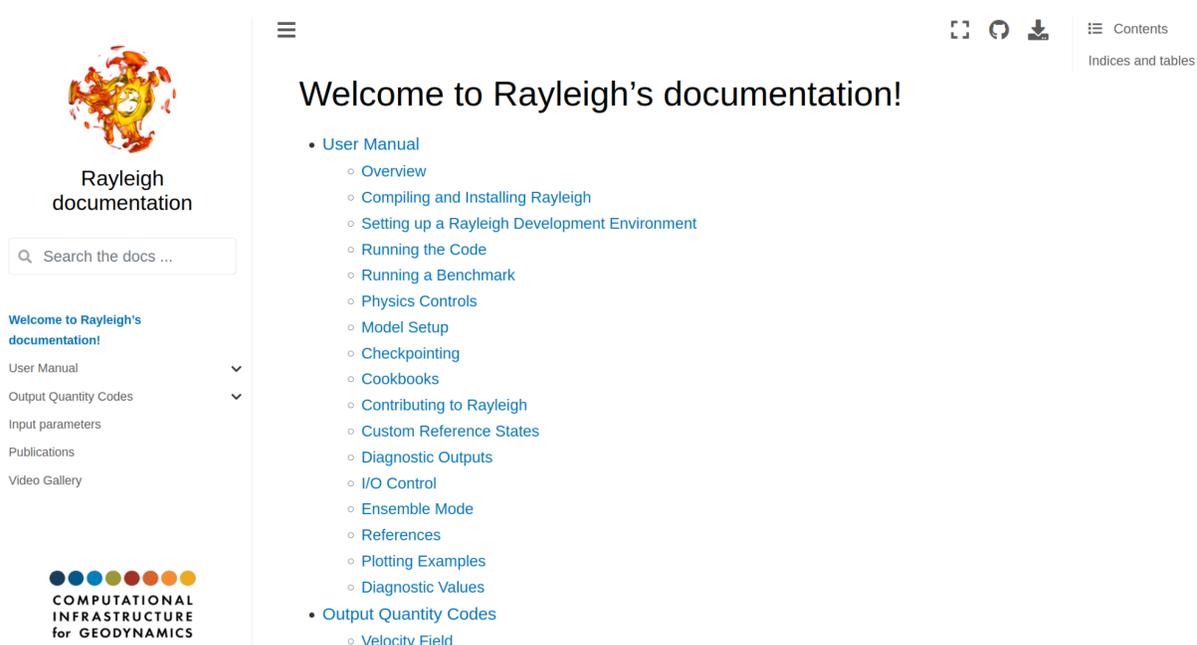


Figure 1: Screenshot of the new Rayleigh documentation design.

Restructure Rayleigh's documentation

Nick Featherstone, Nick Nelson, Rene Gassmoeller, many others

Rayleigh's documentation was initially created by combining several earlier manuals and guides, producing a somewhat confusing structure in the process. We restructured the entire documentation, moved sections, and wrote new sections to make Rayleigh's documentation more coherent and useful. Empty sections corresponding to anticipated content have also been created to facilitate the piecemeal incorporation of that content throughout the coming months. A screenshot of the new structure is below.

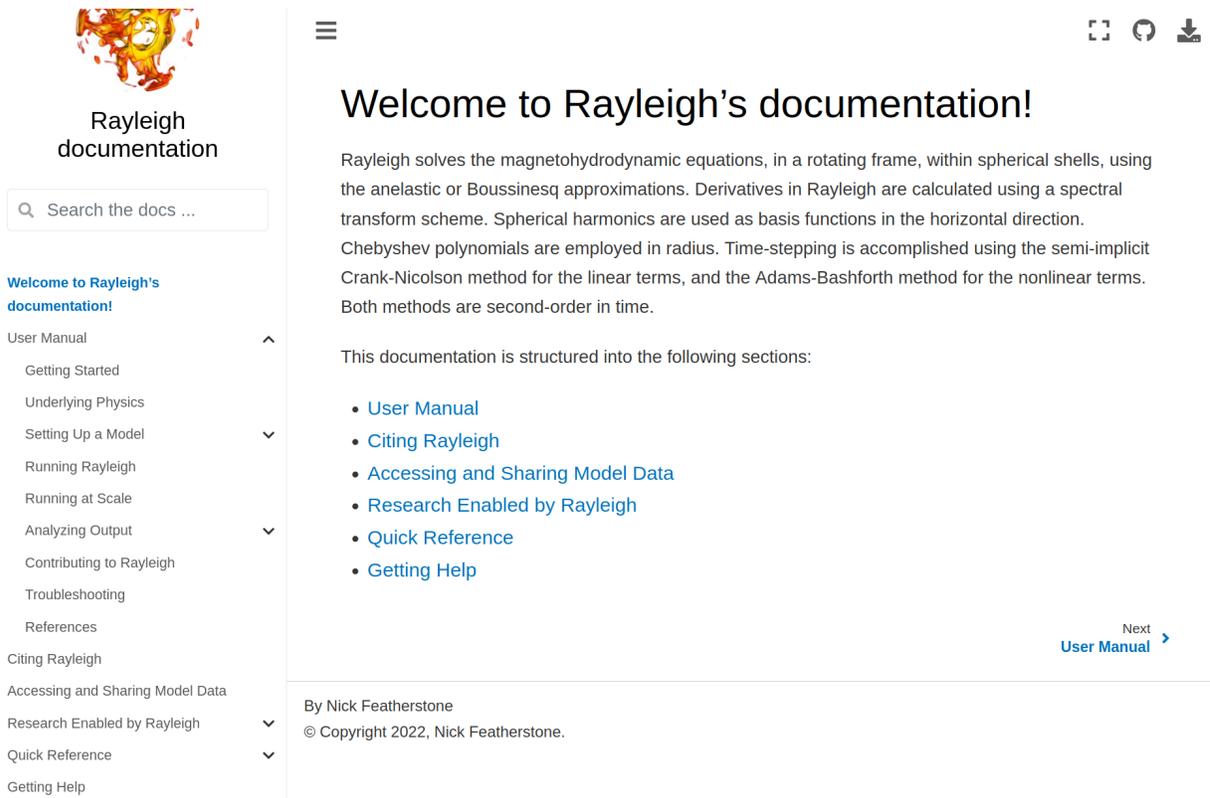


Figure 2: Screenshot of the new Rayleigh documentation structure.

Improvements to Python Analysis Routines

Nicholas Nelson, Philipp Edelmann

Recent changes to the Matplotlib python library allowed for improvements in the plotting routines. Specifically, shell slices can now be easily plotted in Mollweide projection. Also, some general updates were applied.

In addition to the Python analysis routines in `rayleigh_diagnostics.py`, we developed a new framework stored in `rayleigh.py`. This new framework makes use of memory mapping to make data at all timesteps accessible via a uniform interface that hides the fact that the data are split into individual files. This also comes with a set of convenience plotting functions for 2D slices and 1D line plots, which automatically add labels, and tools for time-series Fourier transforms.

Exploring an Alternative Chebyshev Formulation

Kyle Augustson

Kyle Augustson created an example illustrating how the use of Chebyshev polynomials of the second kind can lead to greater sparsity of the matrices. This is discussed in an Overleaf

document [here](#) and is accompanied by an example python program uploaded to the workshop slack channel. There were two parts to the effort: 1. To examine the error growth of chebyshev derivative representations, and find one for which this is no longer an issue and 2. Test Galerkin spectral type methods that have multi-polynomial representations (first and second Chebyshev polynomials). These efforts will be continued throughout the coming year and will likely form the basis for some of next year's hackathon activities.

Toward open INCITE data to public

Hiroaki Matsui

We have performed large scale numerical simulations for geodynamo and Jovian atmosphere on ALCF Mira under the INCITE project. We have planned to open these results to the public, and started to make a documentation page for the INCITE data. The page consists of the list of parameters of the simulations, and examples of visualization data of the results such as flow and magnetic field pattern of equatorial plane or meridional plane.

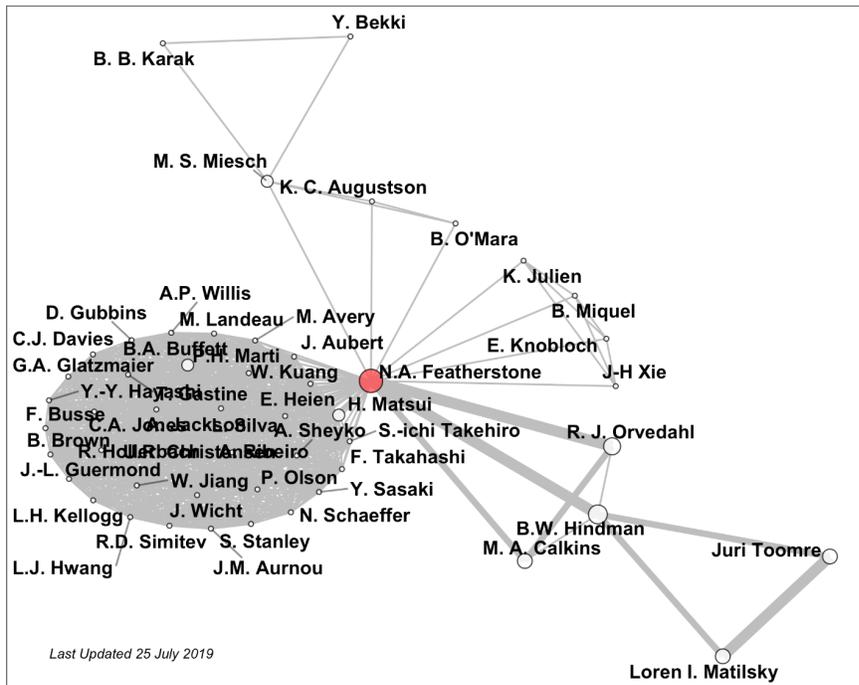
The simulations under the INCITE project are performed by using the pre-released version of Rayleigh, which has a different code from the released version for some fields. In addition, INCITE data consists of big-endian ordering while most today's processors use the little-endian. Consequently, we need to construct tools of data conversion programs to adapt the current version of Rayleigh. Dr. Philipp Edelmann developed Python tools for data conversion of the parameter file ("main_input") and Diagnostic data files, and Dr. Hiroaki Matsui developed the data conversion programs for the checkpoint data by Fortran and C. These tools will also be released in the INCITE data documentation page.

Rayleigh Co-Author Network

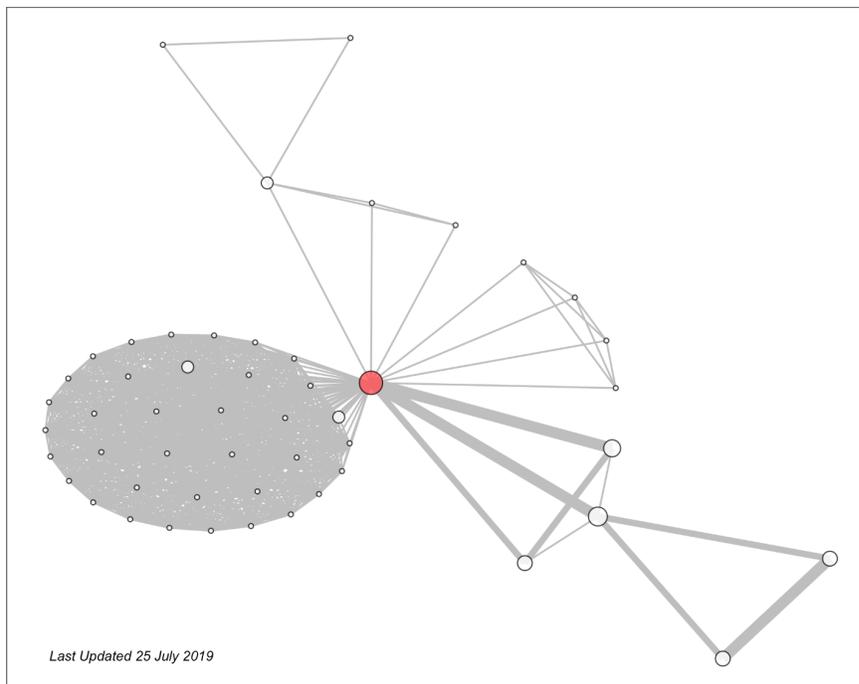
Lorraine Hwang

Network analysis of relationships between coAuthors of peer reviewed publications.

Rayleigh Co-author Relationships



Rayleigh Co-author Relationships



Without labels.

Rename Rayleigh's default branch name to 'main'

Rene Gassmoeller, Philipp Edelmann, Nick Featherstone

We have renamed Rayleigh's default git branch to 'main'. This was done to avoid the use of exclusionary wording and to follow the standards set by github and gitlab for new projects. The rationale and instructions were published in a forum post (<https://community.geodynamics.org/t/renaming-rayleighs-default-branch-to-main/2578>) and in a Google slide presentation (https://docs.google.com/presentation/d/1Rhnz2WPMm6tzbAHi5AspN2s598Fd_mvHvdPOPrJYZlg).

Testing the extra scalar field on Rayleigh (Boussinesq version)

Maria Camisassa, Rafael Fuentes, Cian Wilson, Nick Featherstone

We tested the new version of Rayleigh that includes an extra-scalar field. In particular, we ran simulations of double diffusive (thermal + compositional) convection in a binary mixture of fluid (under the Boussinesq approximation). This benchmark comes from the website of the MagIC code and is described in detail in Breuer et al. 2010, GJI, 183, 150-162¹. It is worth mentioning that in this test the extra-scalar field corresponds to the lighter element of the mixture. The figure below shows the time series of the kinetic energy density for runs using the MagIC code (left panel) and the Rayleigh code (right panel). Both sets of runs consider the same Rayleigh, Prandtl and Ekman numbers and the variable delta indicates the fraction of thermal contribution to the total Rayleigh number in percent. We can see agreement between both codes.

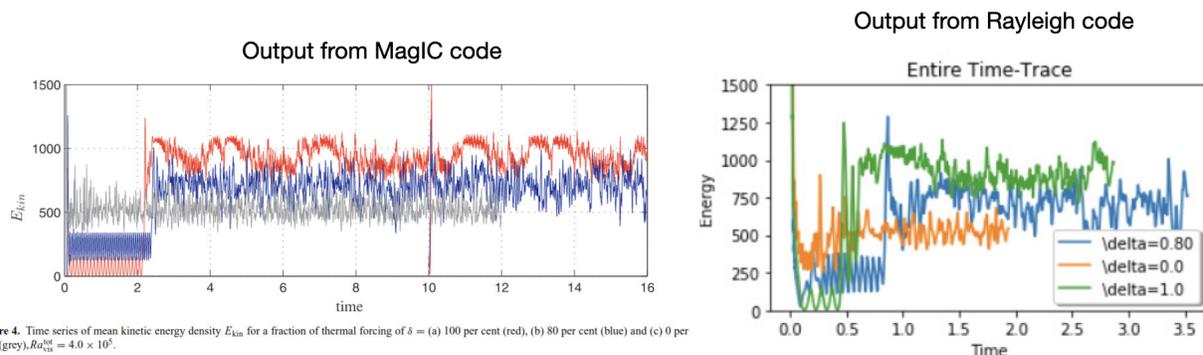


Figure 4. Time series of mean kinetic energy density E_{kin} for a fraction of thermal forcing of $\delta =$ (a) 100 per cent (red), (b) 80 per cent (blue) and (c) 0 per cent (grey). $Ra_{eff} = 4.0 \times 10^7$.

¹ <https://doi.org/10.1111/j.1365-246X.2010.04722.x>

Allowing arbitrary numbers of extra active and passive scalar fields

Cian Wilson, Nick Featherstone, Maria Camisassa, Rafael Fuentes

In order to simulate processes like thermochemical convection it is necessary to have the ability to model extra scalar fields. This had previously been hard-coded into a branch of Rayleigh (used above). Led by Cian Wilson, our team generalized this to both make the inclusion of an extra field optional and to make it possible to simulate more than one field. Doing this involved generalizing several assumptions in Rayleigh about the number of fields to be solved for, establishing new protocols for the output codes of now arbitrary fields (several initial codes have been added), modifying checkpointing to allow more files to be written, linking active fields into the coupled matrix and establishing new matrices for any passive fields. A test has been added for this functionality and a pull request proposed (to be merged next week). Near future work involves fully documenting the new functionality (especially the sign conventions for the active fields' buoyancy contributions), further testing the branch and properly linking the new field coefficients to the custom reference states (ensuring that sufficient space is left in the arrays for expansion).

Bug fixes

Everyone

Various bugs were fixed:

- G_Avgs diagnostic routines produced errors when only one quantity code was included
- G_Avgs diagnostic routines produced an error when reading multiple files with different numbers of records
- The sign of the buoyancy term for extra scalar fields was reversed.

2022 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 Sep 12 2022, commit 93574ab0300c235ffd77ed29699b0f8676965cbb and Sep 27 2022, commit 463e70f9e3ded8cc4b76caab2f1b32063d8d2162 to allow for late merges):

- Number of source files in Rayleigh before/after: 160 -> 163 +2
- Lines of code in Rayleigh before/after: 65195 -> 68465 +3270
- Number of merged pull requests before/after: 342 -> 377 +35
- Commits in github before/after: 1030 -> 1167 +137
- Number of tests before/after: 6 -> 6 +0

Statistics from 2021:

- 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