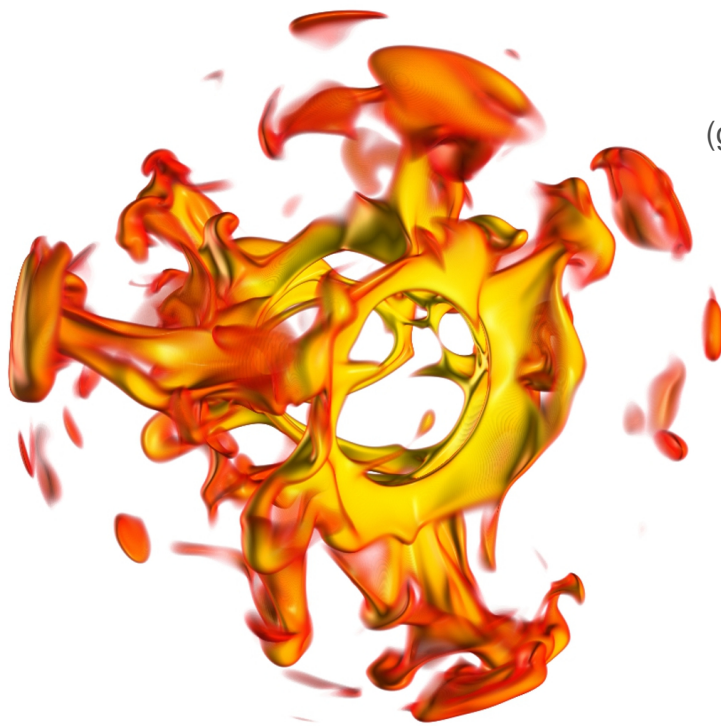


# RAYLEIGH



## User Manual

Version 1.0.1

(generated March 28, 2022)

Nicholas Featherstone

with contributions by:

Kyle Augustson, Wolfgang Bangerth, Rene Gassmöller, Sebastian Glane, Brad Hindman, Lorraine Hwang, Hiro Matsui, Ryan Orvedahl, Krista Soderlund, Cian Wilson, Maria Weber, Rakesh Yadav

[geodynamics.org](http://geodynamics.org)

**CONTENTS**

<b>1</b>	<b>Rayleigh User Manual</b>	<b>1</b>
1.1	Grid Specification . . . . .	1
<b>2</b>	<b>Main_Input Namelists</b>	<b>5</b>
2.1	Problemsize . . . . .	5
2.2	Numerical Controls . . . . .	6
2.3	Physical Controls . . . . .	6
2.4	Temporal Controls . . . . .	7
2.5	IO Controls . . . . .	8



## RAYLEIGH USER MANUAL

### 1.1 Grid Specification

Rayleigh solves the fluid equations in spherical-shell geometry. As the poles are included, the grid is fully specified by providing four pieces of information:

- The coordinates of the computational domain's radial boundaries of the domain,  $r_{\min}$  and  $r_{\max}$
- The number of radial grid points,  $N_r$
- The number of latitudinal grid points,  $N_\theta$

The number of longitudinal grid points,  $N_\phi$ , is always twice  $N_\theta$ . The total number of gridpoints for a Rayleigh simulation is then given by  $2N_r N_\theta^2$ . Note that both  $N_r$  and  $N_\theta$  must be even. Rayleigh's computational grid is specified using the problemsize namelist in the main\_input file. A quick reference for all problemsize-namelist variables is provided in the [namelist documentation](#). In this section, we discuss in detail how to define Rayleigh's grid using these variables.

#### 1.1.1 Standard grid specification

We begin by discussing how to define a grid employing a single Chebyshev domain in radius, meaning that a single Chebyshev expansion is carried out over the domain  $r_{\min} \leq r \leq r_{\max}$ . This is probably the most common grid setup employed in Rayleigh.

The problemsize variables `n_r` and `n_theta` provide values for  $N_r$  and  $N_\theta$  respectively. Similarly, `rmin` and `rmax` define the value for  $r_{\min}$  and  $r_{\max}$ . If we wanted to define a spherical shell extending from  $r=1.0$  to  $r=2.0$ , with  $N_r = 48$  and  $N_\theta = 96$ , our problemsize namelist should look like:

```
&problemsize_namelist
  n_r = 48
  n_theta = 96
  rmin = 1.0
  rmax = 2.0
/
```

Note that  $N_r$  and  $N_\theta$  may also be specified at the command line using the flags `-nr` and `-ntheta`, e.g.:

```
mpiexec -np 8 ./rayleigh.opt -nr 48 -ntheta 96
```

Doing so will override any values supplied via main\_input. This can be particularly useful when scripting

performance analyses on a new machine.

If desired, a user may instead specify the radial domain bounds in terms of the shell aspect ratio  $\chi = r_{\min}/r_{\max}$ , and the shell depth  $r_{\max} - r_{\min}$ . This is accomplished using the `aspect_ratio` and `shell_depth` problemsize variables. The example below describes a grid equivalent to the one described above.

```
&problemsize_namelist
  n_r = 48
  n_theta = 96
  aspect_ratio = 0.5
  shell_depth = 1.0
/
```

Rayleigh's horizontal resolution ( $N_\theta \times N_\phi$ ) may alternatively be described in terms of spherical harmonics. The maximum Legendre degree employed in Rayleigh's truncated spherical harmonic expansion is denoted by  $\ell_{\max}$ , and the total number of degrees by  $N_\ell$ . These two variables are related to  $N_\theta$  via

$$N_\ell = \ell_{\max} + 1 = \frac{2}{3}N_\theta,$$

and they are described by the problemsize variables `n_l` and `l_max`. Thus, the examples

```
&problemsize_namelist
  n_r = 48
  l_max = 63
  rmin = 1.0
  rmax = 2.0
/
```

and

```
&problemsize_namelist
  n_r = 48
  n_l = 64
  aspect_ratio = 0.5
  shell_depth = 1.0
/
```

both describe a grid extending from  $r=1.0$  to  $r=2.0$ , with  $N_r = 48$  and  $N_\theta = 96$ .

### 1.1.2 Defining multiple Chebyshev domains

In some instances, it may be advantageous to describe the radial grid using multiple Chebyshev domains. The most common use case probably occurs when the system under consideration is characterized by layers subject to different physical conditions. For instance, models that include regions that are both superadiabatically and subadiabatically stratified might employ a different Chebyshev expansion within each domain. Similarly so for geodynamo models that include the solid inner core.

When describing a grid with  $N$  Chebyshev domains, the `main_input` file must first supply  $N+1$  points  $r_i$  that define the bounds of these domains. The  $i$ th Chebyshev domain will span the interval  $r_i \leq r \leq r_{i+1}$ , and

the global domain bounds are defined such that

$$r_0 \equiv r_{\min} \quad \text{and} \quad r_{N+1} \equiv r_{\max}.$$

I was here. The next example is good. Note that we use `ncheby`. Note that a radial point will be repeated.

It is possible to run Rayleigh with multiple, stacked domains in the radial direction. Each of these is discretized using their own set of Chebyshev polynomials. The boundaries and number of polynomials can be set for each domain individually, which makes it possible to control the radial resolution at different radii.

To use this feature the problem size has to be specified using `domain_bounds` and `ncheby` instead of `rmin`, `rmax`, and `n_r`. `ncheby` takes a comma-separated list of the number of radial points to use in each domain. `domain_bounds` takes a comma-separated list of the radii of the domain boundaries, starting with the smallest radius. It has one element more than the number of domains. This is an example of two radial domains, one covering the radii 1 to 2 with 16 radial points, the other the radii 2 to 4 with 64 radial points.

```
&problemsize_namelist
  domain_bounds = 1.0, 2.0, 4.0
  ncheby = 16, 64
/
```

Radial values in the diagnostic output will be repeated at the inner domain boundaries. Most quantities are forced to be continuous at these points.

### 1.1.3 Controlling radial dealiasing



## MAIN\_INPUT NAMELISTS

### 2.1 Problemsize

This namelist is used to specify the grid.

**n\_r** Number of radial points in model grid

**rmin** Radius of the inner domain boundary,  $r_{\min}$

**rmax** Radius of the outer domain boundary,  $r_{\max}$

**aspect\_ratio**  $r_{\min}/r_{\max}$

**shell\_depth**  $r_{\max} - r_{\min}$

**n\_theta** Number of theta points in the model grid,  $N_{\theta}$

**l\_max** Truncation degree  $\ell_{\max}$  used in the spherical harmonic expansion

**n\_l**  $\ell_{\max} + 1$

**nprow** Number of MPI ranks within each row of the 2-D process grid

**npcol** Number of MPI ranks within each column of the 2-D process grid

**ncheby** Comma-separated list indicating number of Chebyshev polynomials used in each radial subdomain (e.g., 16, 32, 16). Default: n\_r [ single domain]

**dealias\_by** Comma-separated list indicating number of Chebyshev modes dealiased to zero. Default is 2/3 ncheby.

**domain\_bounds** The domain bounds defining each Chebyshev subdomain

**n\_uniform\_domains** Number of uniformly-sized Chebyshev domains spanning the depth of the shell. Default: 1

**uniform\_bounds** When set to .true., each chebyshev subdomain will possess the same radial extent. Default: .false.



## 2.2 Numerical Controls

This namelist provides access to Rayleigh’s run-time optimization options.

**band\_solve** For use with models employing at least three Chebyshev domains. In those models, the rows of the normally dense matrices used in the Crank-Nicolson scheme may be rearranged into a block-banded form. Setting this variable to `.true.` will perform this rearrangement, and Rayleigh will execute a band, rather than dense, solve during each timestep. Using the band-solve approach can help save memory and may yield performance gains. No benefit is gained for models using one or two Chebyshev domains. The default behavior is to use a dense solve (`band_solve = .false.`).

**static\_transpose** When set to `.true.`, buffer space used during Rayleigh’s transposes is allocated once at runtime. The default behavior (`static_transpose=.false.`) is to allocate and deallocate buffer space during each transpose. On some machines, avoiding this cycle of allocation/deallocation has led to minor performance improvements.

**static\_config** When set to `.true.`, sphericalbuffer configurations (e.g., `p3a`, `s2b`) are allocated once at runtime. The default behavior (`static_config=.false.`) is to save memory by deallocating memory associated with the prior configuration space following a transpose. If memory is not an issue, this may lead to minor performance improvements on some systems.

**pad\_alltoall** When set to `.true.`, transpose buffers are padded throughout with zeros to enforce uniform message size, and a standard alltoall is used for each transpose. The default behavior (`pad_alltoall=.false.`) uses alltoallv and variable message sizes. Depending on the underlying alltoall algorithms in the MPI implementation used, performance may differ between these two approaches.

## 2.3 Physical Controls

This namelist controls the physical effects used in a Rayleigh simulation.

**magnetism** When set to `.true.`, the MHD approximation is employed. The default (`magnetism=.false.`) is to omit the effects of magnetism.

**nonlinear** When set to `.false.`, all nonlinear terms are omitted in the model. The default (`nonlinear=.true.`) is to include those terms.

**momentum\_advection** When set to `.false.`,  $\mathbf{v} \cdot \nabla \mathbf{v} = 0$ . This flag is primarily for debugging purposes. The default value is `.true.`

**inertia** When set to `.false.`, the material derivative of velocity is omitted ( $\frac{D\mathbf{v}}{Dt} = 0$ ). This option is primarily intended for mantle convection models. The default value is `.true.`

**rotation** When set to `.true.`, the Coriolis term is included in the momentum equation. The default behavior is to omit rotation in a Rayleigh model (`rotation = .false.`).

**lorentz\_forces** Set this debugging/development flag to `.false.` to disable the Lorentz force. Default value is `.true.`, but this flag is ignored entirely when `magnetism = .false.`

**viscous\_heating** Determines whether viscous heating is included in the thermal energy equation. Default value is `.true.` Note that the user-supplied value of this variable is ignored entirely for Boussinesq models run with `reference_type = 1`. In those models, `viscous_heating` is set to `.false.`

**ohmic\_heating** Determines whether ohmic heating is included in the thermal energy equation. Default value is `.true.` Note that the user-supplied value of this variable is ignored entirely for Boussinesq

models run with `reference_type = 1`. In those models, `ohmic_heating` is set to `.false`.

**advect\_reference\_state** Determines whether the reference-state entropy is advected. The default is `.true`. When set to `.false`., the  $v_r \frac{\partial \bar{S}}{\partial r}$  term is omitted in the thermal energy equation. Note that this variable has no impact on models with an adiabatic background state.

**benchmark\_mode** When set to a positive value in the interval  $[1,4]$ , an accuracy benchmark will be performed. The default is 0 (no benchmarking). Boussinesq benchmarks are performed for values of 1 (nonmagnetic) and 2 (magnetic). Anelastic benchmarks are performed if `benchmark_mode` has a value of 3 (nonmagnetic) or 4 (magnetic).

**benchmark\_integration\_interval** Determines the interval (in timesteps) between successive benchmark snapshot analyses.

**benchmark\_report\_interval** Determines the interval (in timesteps) between successive benchmark report outputs. Each output contains an average over all benchmark snapshot analyses performed since the previous report.

## 2.4 Temporal Controls

This namelist controls timing, time-stepping, and checkpointing in Rayleigh.

**alpha\_implicit** Determines the value of  $\alpha$  used in the Crank-Nicolson semi-implicit time-stepping scheme employed for linear terms. The default value is 0.5, which ensures second-order accuracy of the algorithm. A value of 1 (0) describes a fully implicit (explicit) algorithm.

**max\_iterations** Maximum number of timesteps for which to evolve a single instance of Rayleigh before exiting the program. Note that this value does not describe the maximum number of timesteps a model can be run for. Instead, it determines the maximum number of timesteps Rayleigh will run for during a given session (i.e. following a single call to `mpiexec/mpirun`). The default value is 1,000,000.

**max\_time\_minutes** Maximum walltime (in minutes) for which to run a single instance of Rayleigh before exiting. As with `max_iterations`, this is specific to a given Rayleigh session. Default is  $10^8$  minutes (essentially, unlimited).

**max\_simulated\_time** The maximum time, in simulation units, for which to evolve a Rayleigh model. Restarting a model that has already reached this limit will result in running for a single time step before exiting. The default is effectively unlimited, with a value of  $10^{20}$ .

**save\_last\_timestep** When set to `.true`. (default), Rayleigh will checkpoint before exiting normally. Note that this generally occurs when the maximum time or iterations is reached. This does not apply when a job is terminated by the MPI job scheduler.

**checkpoint\_interval** Number of iterations between successive checkpoint outputs. Default value is -1 (no checkpointing).

**check\_frequency** (deprecated) Same as `checkpoint_interval`.

**quicksave\_interval** Number of iterations between successive quicksave outputs. Default value is -1 (no quicksaves).

**num\_quicksaves** Number of quicksave slots (i.e., rapid, rolling checkpoint folders) to use for a given simulation. Default value is 3.

**quicksave\_minutes** Time in minutes between successive quicksaves. If this variable is set to a positive value

(default is -1), the value of `quicksave_interval` will be ignored.

**max\_time\_step** The maximum allowed time step. This value will be respected even when if the CFL constraint admits a larger time-step size. Default value is 1.0.

**min\_time\_step** The minimum allowable time step. If the CFL constraint forces a time-step size that falls below this value, Rayleigh will exit.

**cflmin** Used for adaptive timestep control. Rayleigh ensures that the time-step size never falls below  $cflmin \times t_{CFL}$ , where  $t_{CFL}$  is the minimum timestep allowed by the CFL constraint. The default value is 0.4.

**cflmax** Used for adaptive timestep control. Rayleigh ensures that the time-step size never exceeds  $cflmax \times t_{CFL}$ , where  $t_{CFL}$  is the minimum timestep allowed by the CFL constraint. The default value is 0.6.

**new\_iteration** If desired, a simulation's iteration numbers may be reset upon restarting from a checkpoint. Set this value to the new iteration number to use (must be greater than zero), and the old iteration number contained in the checkpoint file will be ignored. The default value is 0.

## 2.5 IO Controls

This namelist provides variation options to control Rayleigh's input and output cadence and structure.

**stdout\_file** If desired, set this variable to the name of a file to which Rayleigh's text output is redirected. This can be useful for monitoring run progress and time-step size on systems that otherwise don't produce the text output until a run has complete. The default value is 'nofile,' which indicates that Rayleigh should not redirect stdout to a file.

**stdout\_flush\_interval** Number of lines to cache before writing to the `stdout_file` if used. This prevents excessive disk access while a model is evolving. The default value is 50.

**jobinfo\_file** Set this variable to the name of a file, generated during Rayleigh's initialization, that contains the values assigned to each namelist variable, along with compiler and Git hash information. The default filename is 'jobinfo.txt'

**terminate\_file** The name of a file that, if found in the top-level simulation directory, indicates Rayleigh should terminate execution. This can be useful when trying to exit a run cleanly before the scheduled wall time runs out. The default filename is 'terminate'.

**terminate\_check\_interval** Number of iterations between successive checks for the presence of the job termination file. The default value is 50.

**statusline\_interval** Number of iterations between successive outputs to stdout indicating time step number and size. The default value is 1, so that iteration number and time-step size are printed during every time step.

**outputs\_per\_row** Determines the number of process columns that participate in MPI-IO during checkpointing and diagnostic outputs. Acceptable values fall in the range  $[1, nprow]$ , with a default value of 1.

**integer\_output\_digits** Number of digits to use for all integer-based filenames (e.g., `G_Avgs/00000001`). The default value is 8.

**integer\_input\_digits** Number of digits for integer-based checkpoint names to be read during a restart. The default value is 8.

**decimal\_places** Number of digits to use after the decimal point for those portions of Rayleigh's text output

that displayed in scientific notation. The default value is 3.