COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)

RAYLEIGH

User Manual

Version 1.0.1

(generated April 5, 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

CONTENTS

CHAPTER

ONE

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_θ

The number of longitudinal grid points, N_ϕ , is always twice N_θ. The total number of gridpoints for a Rayleigh simulation is then given by $2N_rN_\theta^2$. Note that both N_r and N_θ 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 $r_{\mathrm{min}} \leq r \leq r_{\mathrm{max}}$. 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_θ respectively. Similarly, *rmin* and *rmax* define the value for r_{min} and *r_mathrm{max}*. If we wanted to define a spherical shell extending from r=1.0 to r=2.0, with $N_r = 48$ and $N_\theta = 96$, out 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_θ 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 $\chi = r_{\mathrm{min}}/r_{\mathrm{max}}$, and the shell depth $r_{\mathrm{max}} - r_{\mathrm{min}}$. This is accomplished using the 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 ℓ_{max}, and the total number of degrees by N_ℓ. These two variables are related to N_θ via

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

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 *ith* 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 indiviadually, 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

CHAPTER

TWO

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 $r_{\mathrm{min}}/r_{\mathrm{max}}$

shell_depth $r_{\mathrm{max}} - r_{\mathrm{min}}$

n_theta Number of theta points in the model grid, N_θ

l_max Truncation degree ℓ_{max} used in the spherical harmonic expansion

n_l $\ell_{\mathrm{max}} + 1$

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_tranpose=.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 my 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., $\boldsymbol{v}\cdot\boldsymbol{\nabla}\boldsymbol{v}=0$. This flag is primarily for debugging purposes. The default value is .true.

inertia When set to .false., the material derivative of velocity is omitted ($\frac{D\boldsymbol{v}}{Dt}=0$). 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 $v_r \frac{\partial \overline{S}}{\partial r}$ 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 peformed 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 α 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 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 contraint 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.

clfmax Used for adaptive timestep control. Rayleigh ensures that the time-step size never exceeds $\mathrm{cflmax} \times t_{\mathrm{CFL}}$, 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 ignored. The default value is 0.

2.5 IO Controls

This namelist provides various 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 if 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 sdout 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 particpate 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 then decimal point for those portions of Rayleigh's text output that displayed in scientific notation. The default value is 3.

2.6 Boundary Conditions

This namelist provides those options necessary to determine the boundary conditions employed in a Rayleigh model.

fix_tvar_top Logical flag indicating whether thermal variable (T,S) should be fixed on the upper boundary. Default = .true.

fix_tvar_bottom Logical flag indicating whether thermal variable (T,S) should be fixed on the lower boundary. Default = .true.

fix_dtdr_top Logical flag indicating whether the radial derivative of thermal variable (T,S) should be fixed on the upper boundary. Default = .false.

fix_dtdr_bottom Logical flag indicating whether the radial derivative of thermal variable (T,S) should be fixed on the lower boundary. Default = .false.

T_top Value of thermal variable (T,S) at the upper boundary. Default = 0.

T_bottom Value of thermal variable (T,S) at the lower boundary. Default = 1.

dTdr_top Value of radial derivative of thermal variable (T,S) at the upper boundary. Default = 0.

dTdr_bottom Value of radial derivative of thermal variable (T,S) at the lower boundary. Default = 0.

adjust_dTdr_top Logical flag indicating that dTdr_top should be set based on the values of heating_integral (or luminosity) and the value of dTdr_bottom. Default value is .false. When .true., this flag only has an effect when fix_dtdr_top = .true. and heating_type > 0. When active, dTdr_top is set such that the integrated flux passing through the upper boundary is equal to the sum of those due to internal heating and any flux passing through the lower boundary due to fixed dTdr_bottom.

no_slip_top When .true., a no-slip condition on the horizontal velocity field is enforced at the upper boundary. Default = .false.

no_slip_bottom When .true., a no-slip condition on the horizontal velocity field is enforced at the lower boundary. Default = .false.

stress_free_top When .true., a stress-free condition on the horizontal velocity field is enforced at the upper boundary. Default = .true.

stress_free_bottom When .true., a stress-free condition on the horizontal velocity field is enforced at the lower boundary. Default = .true.

no_slip_boundaries When .true., both no_slip_top and no_slip_bottom are set to .false. Default = .false.

strict_L_Conservation In some cases, typically rotating models employing MHD or thick shells, angular momentum can leak into/out of the domain even when using stree-free boundaries. When .true., this flag replaces the upper boundary condition with an integral constraint on the $\ell = 1$ toroidal streamfunction that enforces strict conservation of angular momentum. Note that the upper boundary is neither stress-free nor no-slip in this case. Default = .false.

T_top_file Generic-input file containing a custom, fixed (T,S) upper boundary condition.

T_bottom_file Generic-input file containing a custom, fixed (T,S) lower boundary condition.

dTdr_top_file Generic-input file containing a custom, fixed ($\partial T/\partial r$, $\partial S/\partial r$) upper boundary condition.

dTdr_bottom_file Generic-input file containing a custom, fixed ($\partial T/\partial r$, $\partial S/\partial r$) lower boundary condition.

C_top_file Generic-input file containing a custom upper boundary condition for the poloidal flux function C.

C_bottom_file Generic-input file containing a custom lower boundary condition for the poloidal flux function C.

2.7 Initial Conditions

All variables necessary to initialize velocity, temperature, pressure, and magnetic field are supplied here.

init_type

Integer value indicating how nonmagnetic variables should be initialized.

- type -1: Restart from a checkpoint
- type 1: Hydro Boussinesq benchmark init (Christensen et al. 2001). The temperature field is initialized with an $\ell = 4$, m=4 perturbation on top of a conductive profile. Velocity/pressure are zero.
- type 6: Hydro anelastic benchmark init (Jones et al. 2011). The entropy field is initialized with an $\ell = 19$, m=19 and $\ell = 1$, m=1 perturbation on top of a conductive profile. Velocity/pressure are zero.
- type 7: A randomized temperature/entropy field is initialized. Velocity and pressure are set to zero.
- type 8: Velocity, entropy/temperature, and pressure are initialized to zero, or if an associated filename is provided, they are initialized using the generic input interface.

magnetic_init_type

Integer value indicating how magnetic field should be initialized.

- type -1: Initialize magnetic field from a checkpoint.
- type 1: Magnetic initialization for Christensen et al. (2001), case 1. The poloidal flux function is initialized using an $\ell = 1, m = 0$ mode. THe toroidal flux function is initialized with an $\ell = 2, m = 0$ mode.
- type 7: The poloidal and toroidal flux functions are initialized to randomized values.
- type 8: The poloidal and toroidal flux functions are intialized to zero, and then if a corresponding generic input file is specified, their initial state is read from that file.

restart_iter Iteration number indicating the checkpoint to restart from when init_type and magnetic_init_type equal 1.

temp_amp Amplitude of randomized temperature/entropy perturbations to use with init_type = 7.

mag_amp Amplitude of randomized magnetic perturbations to use with magnetic_init_type = 7.

t_init_file Name of generic input file that, if init_type=8, will be used to initialize temperature/entropy.

p_init_file Name of generic input file that, if init_type=8, will be used to initialize pressure.

w_init_file Name of generic input file that, if init_type=8, will be used to initialize the poloidal stream function W.

z_init_file Name of generic input file that, if init_type=8, will be used to initialize the toroidal stream function Z.

c_init_file Name of generic input file that, if init_type=8, will be used to initialize the poloidal stream function C.

a_init_file Name of generic input file that, if init_type=8, will be used to initialize the toroidal stream function A.

rescale_velocity Logical variable indicating that the velocity field should be rescaled upon restart. Default = .false.

velocity_scale Factor by which to rescale the velocity field upon restart.

rescale_pressure Logical variable indicating that the pressure field should be rescaled upon restart. Default = .false.

pressure_scale Factor by which to rescale the pressure field upon restart.

rescale_tvar Logical variable indicating that the temperature/entropy field should be rescaled upon restart. Default = .false.

tvar_scale Factor by which to rescale the temperature/entropy field upon restart.

rescale_bfield Logical variable indicating that the magnetic field should be rescaled upon restart. Default = .false.

bfield_scale Factor by which to rescale the magnetic field upon restart.

2.8 Reference

This namelist provides options to control the properties of Rayleigh's background state.

reference_type

Determines the fluid approximation and background state used by Rayleigh.

- type 1: Boussinesq + nondimensional
- type 2: Anelastic + polytropic background state (dimensional)
- type 3: Anelastic + polytropic background state (non-dimensional)
- type 4: Custom reference-state (read from file)

poly_n The polytropic index used to describe the background state for reference types 2 and 3.

poly_Nrho Number of density scaleheights spanning the interval $r_{\mathrm{min}} \leq r \leq r_{\mathrm{max}}$ for reference types 2 and 3.

poly_mass Mass interior to r_{min}, used in defining the polytropic reference state for reference types 2 and 3.

poly_rho_i Specifies the value of density at the inner boundary $r = r_{\mathrm{min}}$ for the polytropic reference states of reference types 2 and 3.

pressure_specific_heat Determines the value of the specific heat at constant pressure, c_{p} for reference types 2 and 3.

heating_type

Integer value that determines the form of the internal heating function $Q(r)$. The default value is 0, which indi

- type 1: $Q(r) \propto \overline{\rho}(r)\overline{T}(r)$.
- type 4: $Q(r)$ is a constant function of radius.

heating_integral Determines the heating normalization L, defined such that $L = 4\pi \int_{r_\mathrm{min}}^{r_\mathrm{max}} Q(r) r^2 dr$.

luminosity Same as heating_integral. If both are specified, the value of heating_integral will be used.

angular_velocity Determines the frame rotation rate Ω for rotating models employing reference type 2.

rayleigh_number Sets the value of the Rayleigh number Ra for reference type 1.

ekman_number Sets the value of the Ekman number Ek for reference types 1 and 3.

prandtl_number Sets the value of the Prandtl number Pr for reference types 1 and 3.

prandtl_number Sets the value of the magnetic Prandtl number Pm for reference types 1 and 3.

dissipation_number Sets the value of the dissipationg number Di for reference type 3.

modified_rayleigh_number Sets the value of the modified Rayleigh number Ra^* for reference type 3.

gravity_power Specifies the value of *n* (real number) used to determine the radial variation of gravitational acceleration *g* in reference type 1, where $g \propto \left(\frac{r}{r_\mathrm{max}}\right)^n$.

ra_constants Indicates the desired value of specified constant coefficients when reading the value from main_input instead of from a custom-refernce file. For use with override_constants or override_constant flags. Syntax is:

```
&Reference_Namelist
 ...
 ra_constants( 2) = 1.0
 ra_constants(10) = 14.0
 ...
/
```

with_custom_constants Comma separated list of integers indicating which constant coefficients should be read from a custom-refernce file when with_custom_reference is true.

with_custom_functions Comma separated list of integers indicating which non-constant coefficients should be read from a custom-refernce file when with_custom_reference is true.

with_custom_reference Logical flag that indicates some constant and non-constant coefficients should be read from a custom-reference file and used to overwrite those values otherwise assigned for reference_Types 1–3. Default value is .false.

custom_reference_file Name of file from which to read custom-reference-state information when using reference_type 4 or when augmenting reference types 1–3.

override_constants When true, ALL constant coefficients specified in the custom-reference file will be ignored, and those specified in main_input will be used instead. Constant coefficients not specified in main_input will be assigned a value of zero. Default value is .false.

override_constant Indicates that particular constant coefficients, rather than all, should be overridden using main_input values when using reference_type 4. Multiple constant overrides can be specified, one per line, with the syntax:

```
&Reference_Namelist
 ...
 override_constant( 2) = T
 override_constant(10) = T
 ...
/
```

2.9 Transport

This namelist enables control of Rayleigh's diffusivities.

{nu,kappa,eta}_type

Determines the radial profile of the associated diffusion coefficient.

- type 1 : no radial variation
- type 2 : diffusivity profile varies as ρ^n for some real number n.
- type 3 : diffusivity profile is read from a custom-reference-state file

{nu,kappa,eta}_top

Specifies the value of the associated diffusion coefficient at the upper boundary. This is primarily used for dime

- reference_type 1: $\nu_{\mathrm{top}} = 1$, $\kappa_{\mathrm{top}} = 1/\mathrm{Pr}$, $\eta_{\mathrm{top}} = 1/\mathrm{Pm}$
- reference_type 3: $\nu_{\mathrm{top}} = \mathrm{Ek}$, $\kappa_{\mathrm{top}} = \mathrm{Ek}/\mathrm{Pr}$, $\eta_{\mathrm{top}} = \mathrm{Ek}/\mathrm{Pm}$

{nu,kappa,eta}_power Denotes the value of the exponent n in the ρ^n variation associated with diffusion type 2.

hyperdiffusion

Set this to variable to .true. to enable hyperdiffusion. The default value is .false. When active, diffusivities are m

- $\{\nu,\kappa,\eta\} \rightarrow \{\nu,\kappa,\eta\}\left(1+\alpha\left(\frac{\ell-1}{\ell_{\mathrm{max}}-1}\right)^{\beta}\right)$

hyperdiffusion_alpha Determines the value of α when hyper diffusion is active.

hyperdiffusion_beta Determines the value of β when hyper diffusion is active.