Instalation guide

In order to compile the code a 'build' directory should be created. Then the user should open a terminal inside this directory and run the command 'cmake ..' to create a Makefile for compilation. The default compiler is gfortran but the user can also choose an Intel compiler by doing 'cmake -DCMAKE_Fortran_COMPILER=ifx ..' (one can also use ifort instead of ifx). The cmake will default to use MKL and turn to LAPACK if not found. The default build is 'release' but this can be changed to 'debug' by doing 'cmake -DCMAKE_BUILD_TYPE=debug ..'. Once the Makefile is created, the user should run 'make -j' on the terminal and the executable spectral_code_pmmh is ready to use.

Code options

The objective of this part is to outline the different options that we can use when running the code. We can perform time-stepping simulations or use the Newton or the Continuation solvers.

- Time-stepping simulations:
  - delta_t (float): time step for the simulation.
  - NTS (integer): total number of iterations.
  - save_every (integer): Files (output and restart files) will be saved every *save_every* iterations.
  - solver (string): for these simulations it should be either *convective_explicit* or *convective_implicit* for explicit or implicit treatment of the Coriolis term.
  - time_step (string): time stepping scheme for the simulation. It may be *pc, cn, fbe* or *bdf2* for Predictor-Corrector, Crank-Nicolson, Forward-Backward Euler or BDF2.
  - restart (string): it can be *yes, y, no, n*. With this we will indicate the code to begin the simulation by reading the fields from binary files.
  - restart_filename (string): file name for restart file containing fields. If no name is specified the code defaults to *Restart.b* as filename.
  - dim_filename (string): file name for restart file containing dimensions. If no name is specified the code defaults to *Dim.b* as filename.
  - dealiasing (string): it can be *yes, y, no, n*. With this we will indicate the code to use dealiasing techniques for the transformations to real space and back.
  - directory (string): indicate the directory where the simulation will be executed. If doing a restart from binary files, these have to be in that directory.
  - KK (integer): number of Chebyshev modes.
  - LL (integer): number of Legendre modes.
  - MM (integer): number of Fourier modes.
  - mres (integer): ϕ will span an angle range between 0 and 2π/mres.
  - Rin and Rout (floats): inner and outer radii. Default values are set to 7/13 and 20/13, respectively.
  - Pr (float): Prandtl number.
  - Ek (float): Ekman number.
  - Ra (float): Rayleigh number.
  - IER (float): implicit to explicit ratio. Only available in Predictor-Corrector and Crank-Nicolson time-stepping schemes. For more details see R. Hollerbach *A spectral solution of the magneto-convection equations in spherical geometry* in Int. J. Numer. Meth. Fluids 2000.
  - init (string): initial condition. Can be set to *Christensen* for the U.R. Christensen *et al. A numerical dynamo benchmark*, in Physics of the Earth and Planetary Interiors 128 (2001) initial condition. Otherwise, it can be set to *symmetric*, for an M-fold symmetry initial condition. Should this last option be chosen, two other parameters are to be set:
    - sym (integer): M-fold symmetry of choice.
    - init_amp (float): amplitude of the initial condition. The resulting initial condition will be init_amp $\cdot \cos(\text{sym} \cdot \phi)$

An example of a command line to begin a simulation from a symmetric initial condition would be:

./build/spherical_code_pmmh -delta_t 1.0e-4 -NTS 5000 -save_every 5000 -KK 30 -LL 40 -MM 10

-mres 4 -Pr 1. -Ek 1.0e-3 -Ra 65. -time_step bdf2 -directory 'my_directory' -restart no -init symmetric -sym 4 -init_amp 4 -dealiasing yes -solver convective_explicit

- Newton Solver:

  All the options stated for the time stepping simulations are valid except for: NTS, save_every, init, init_amp and sym and restart. The restart filenames should be specified unless they go by the default names indicated above.

  - delta_t (float): time step for the simulation. Should be set delta_t $\approx 10^2$.
  - solver (string): for these simulations it should be either *newton_convective_explicit* or *newton_convective_implicit* for explicit or implicit treatment of the Coriolis term.
  - time_step (string): time stepping scheme for these simulations may be *cn* or *fbe* for Crank-Nicolson or Forward-Backward Euler. The latter is the default one.
  - max_newt (integer): maximum amount of Newton iterations allowed. Usual values: 10-15.
  - max_gmres (integer): maximum amount of GMRES iterations allowed in each Newton step. Usual values: 500-1000.
  - restart_gmres (integer): in case GMRES restart is desired by the user. Otherwise we recommend to set it equal to max_gmres for no restart.
  - newt_eps (float): tolerance for the Newton method. Usual values: 1×10^{-7}
  - newt_delta (float): stagnation criterion to evaluate how much the Newton solution is changing. Usual values: 1×10^{-16}.
  - tol_gmres (float): GMRES tolerance. Usual values: 1×10^{-10}.
  - M_wave (integer): symmetry of the rotating wave to be found.

  An example of a command line would be:

  ./build/spherical_code_pmmh -delta_t 200. -KK 30 -LL 40 -MM 10 -mres 4 -Pr 1. -Ek 1.0e-3 -Ra 145. -IER 0.8 -directory 'my_directory' -restart_filename Restart_Ra_140.b -dim_filename Dim_Ra_140.b -dealiasing yes -max_newt 15 -max_gmres 1000 -restart_gmres 1000 -newt_eps 1.0e-7 -newt_delta 1.0e-16 -tol_gmres 1.0e-10 -M_wave 4 -time_step fbe -solver newton_convective_implicit

- Continuation Solver

  All the options stated for the Newton Solver are valid and mandatory. The following are more options that need to be specified for the continuation solver to work:

  - Ek_final (float): final for continuation in Ekman.
  - Ra_final (float): final for continuation in Rayleigh.
  - delta_param (float): to indicate how much the parameter will change in each continuation step. Rayleigh changes in linear scale and Ekman in logarithmic scale. Usual values for Rayleigh are 1 to 5 and in Ekman around 1×10^{-2}.
  - adapt_param (string): it can be *yes, y, no, n.* The variation in the parameter will be adapted according to the optimal amount of Newton steps set by the user (Nopt). Default is *no.*
  - Nopt (integer): if adapt_param is set to *yes*, the variation in the parameter will be corrected according to equation (16) from K. Borońska & L. S. Tuckerman *Extreme multiplicity in cylindrical Rayleigh-Bénard convection. II. Bifurcation diagram and symmetry classification*, in Physical Review E (2010). Usual values for Nopt are 3 to 5.
  - gamma (float): for turning point detection, see equation (19) in Borońska and Tuckerman. Usual values for continuation in Rayleigh are 10 to 100 and in Ekman are 400 to 500.
  - grid_refine (string): it can be *yes, y, no, n.* It will indicate the code to perform grid refinement when the spectral resolution is below a threshold. Default is *no.*
  - gr_threshold (float): in case grid_refine is set to *yes* the grid will be refined if the ratio between the last mode and the mode with maximum absolute value is below this threshold. Usual values are 1×10^{-7}.

  Note: one can only set Ek_final *or* Ra_final, not both.

  An example of a command line would be:

./build/spherical_code_pmmh -delta_t 200. -KK 30 -LL 40 -MM 10 -mres 4 -Pr 1. -Ek 1.0e-2 -Ra 65 -Ek_final 1.0e-3 -directory 'my_directory' -restart_filename Restart_20000.b -dim_filename Dim_20000.b -dealiasing yes -max_newt 15 -max_gmres 1000 -restart_gmres 1000 -newt_eps 1.0e-7 -newt_delta 1.0e-16 -tol_gmres 1.0e-10 -M_wave 4 -adapt_param y -Nopt 5 -gamma 100. -delta_param 1. -time_step fbe -solver continuation_convective_implicit

Outputs and visualisation

This repository comes with a 'python' directory where there are 3 python scripts for visualisation:

- plot_data_equatorial_plane.py: takes the files called 'data_eq_plane.dat and plots them using the 'r.dat' and 'phi.dat' data. The user just needs to provide the path to the simulation results and filename of the equatorial plane data.
- plot_data_merid_plane.py: takes the files called 'data_med_plane.dat and plots them using the 'r.dat' and 'theta.dat' data. The user just needs to provide the path to the simulation results and filename of the meridional plane data.
- plot_data_mid_gap.py: takes the files called 'data_mid_gap.dat and plots them using the 'phi.dat' and 'theta.dat' data. The user just needs to provide the path to the simulation results and filename of the mid gap data.

There are other output files that the user may find useful. These can be read easily by creating simple scripts:

- Ekin_spectral.dat: Fourier spectrum of the kinetic energy density.
- T_k_spectral.dat: square of the Euclidean norm of the spectral temperature field as for all Chebyshev modes. This will allow to study the resolution in the r coordinate.
- KE_timeserie.dat (only in timestepping): timeseries of the kinetic energy density.
- Ur_mgep_timeserie.dat (only in timestepping): timestepping of the r component of the velocity field at a point places in the equatorial plane at mid gap. Allows for the computation of drifting frequencies of rotating waves from timestepping simulations.

When performing continuation, there is also the 'Continuation_params.dat' file that contains values of the control parameter, total number of Newton steps, total number of GMRES iterations, kinetic energy, drifting frequency and maximum r component of the velocity field in the equatorial plane.

Finally, there is also the 'parameters.dat' file that has a print of the parameters used in the simulation.

Testing the code

To test the code we will do a run using the timestepping solver and another one using the Newton solver. We will beign by creating a directory 'test_timestepping' which we can place anywhere. If we open a terminal in the code directory, we can run the following line to execute the test:

./build/spherical_code_pmmh -delta_t 1e-4 -KK 30 -LL 40 -MM 10 -mres 4 -NTS 5000 -save_every 2500 -Pr 1. -Ek 1.0e-3 -Ra 100.0 -IER 0.8 -directory '../test_timestepping' -restart no -init christensen -dealiasing yes -solver convective_implicit

As it can be seen, in this case we placed the test directory just outside the code directory, hence the location '../test_timestepping'. This simulation corresponds to "Christensen *et al.*, *A numerical dynamo benchmark*, Physics of the Earth and Planetary Interiors 128 (2001) 25–34". The resulting kinetic energy output on console should be approximately 58.343752.

Now, we will create another directory called 'test_newton' and copy the last two binary files from the timestepping simulations we just did, i.e., Restart_5000.b and Dim_5000.b, into it. The Newton solver will make use of these. Now we run:

./build/spherical_code_pmmh -delta_t 200. -KK 30 -LL 40 -MM 10 -mres 4 -Pr 1. -Ek 1.0e-3 -Ra 100. -IER 0.8 -directory '../test_newton' -restart yes -restart_filename Restart_5000.b -dim_filename Dim_5000.b

-dealiasing yes -max_newt 15 -max_gmres 1000 -restart_gmres 1000 -newt_eps 1.0e-7 -newt_delta 1.0e-16 -tol_gmres 1.0e-10 -M_wave 4 -solver newton_convective_implicit

We are following an M4 rotating wave hence the M_wave setting. The user can check Christensen's initial condition leads to an M4 with the python visualisation codes that are provided in this repository. Finally, the drifting frequency output on console after executing this line should be approximately 0.1824096 and the kinetic energy should be similar to the one obtained by timestepping.