RAYLEIGH OVERVIEW

SECTION 1: BUILDING AND PORTING THE CODE

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

IN THIS SECTION:

- Obtaining the code
- Code Dependencies
- Compilation
- Running an Accuracy Benchmark

OBTAINING THE CODE:

- Rayleigh is available at:

  https://github.com/geodynamics/Rayleigh
- Clone repository or download the tarball

RAYLEIGH DEPENDENCIES:

(REQUIRED TO BUILD AND RUN RAYLEIGH)

- GNU Make
- MPI (Message Passing Interface)
  - So far, tested on MPICH, MVAPICH, OpenMPI, Intel MPI, SGI MPT
- Fortran 2003/MPI compiler
- C++ compiler
- Math libraries:
  - BLAS, LAPack, and FFTW 3.x or later

  OR

  - Intel's Math Kernel Library (MKL*)

* Intel's MKL provides optimized interfaces to BLAS, LAPack, and FFTW

* It is HIGHLY recommended that you use MKL if running on Intel processors

BUILDING THE CODE (OVERVIEW)

- Build via the “configure / make / make install” process

From within the *Rayleigh* directory …

1. Run *./configure {options}*
   -- generates make.inc needed for compilation
2. Run *make*
   -- compiles optimized and debugging versions of Rayleigh
3. Run *make install*
   -- place executables and documentation in desired location

THE RAYLEIGH CONFIGURE SCRIPT

- Most important step in the build process
- Invocation assigns machine-dependent values to variables usd by Rayleigh's Makefile.
- Variables defined in *Rayleigh/src/build/Machine_Definitions*
- Some examples are shown on the following slides
- Obtaining further help:
  - Run ./configure --help (enumerates all configure options)
  - Read Rayleigh/INSTALL (various configure examples)

RUNNING CONFIGURE

- Usage:

```
./configure --option1=value1 … --option{N}=value{N}
```

Important Options

- prefix={Rayleigh Root}
  - Defines root directory of Rayleigh installation
  - Executables placed in {Rayleigh Root}/bin
  - Documentation placed in {Rayleigh Root}/doc
  - Defaults to directory where configure is run

- FC={Fortran MPI Compiler command}

- CC={C++ Compiler command}

PORTING TO NEW MACHINES:

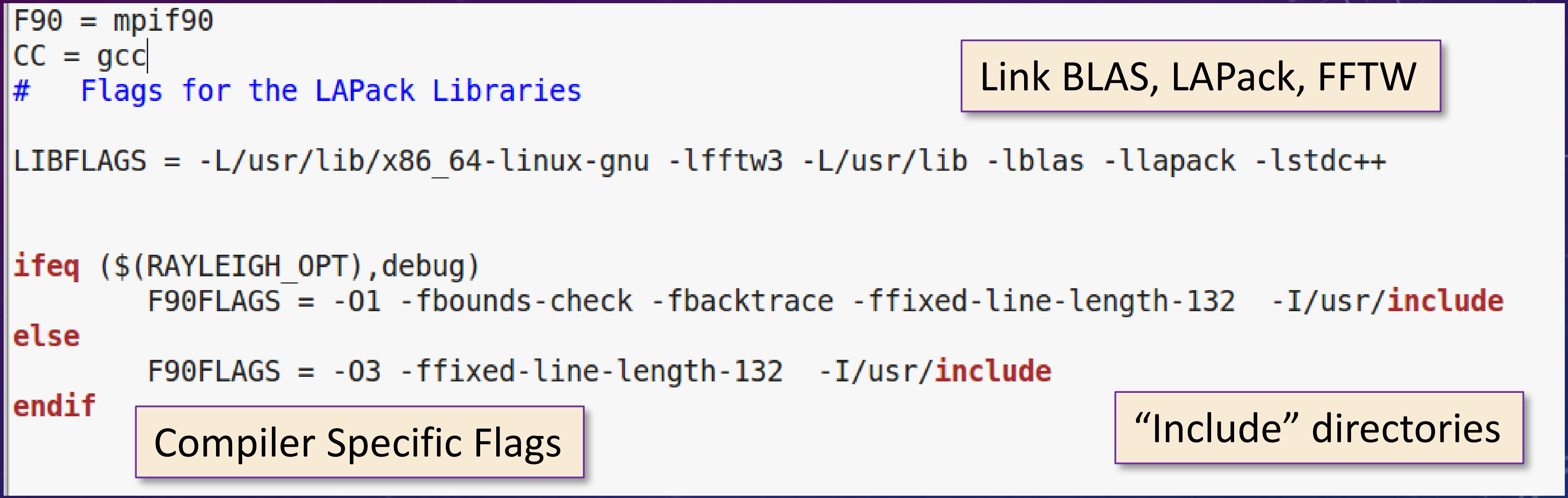
- Create a new Makefile_NAME and place it in the rayleigh/Makefiles directory

```
$ ./build_rayleigh NAME
```

- See examples in Rayleigh/Makefiles:
  - Intel Compiler : Makefile_Pleiades
  - IBM Compiler : Makefile_Mira
  - GNU Compiler : Makefile_CIG

MAKEFILE CUSTOMIZATION

From rayleigh/Makefiles/Makefile_CIG



```
F90 = mpif90
CC = gcc
#   Flags for the LAPack Libraries

LIBFLAGS = -L/usr/lib/x86_64-linux-gnu -lfftw3 -L/usr/lib -lblas -llapack -lstdc++


ifeq ($(RAYLEIGH_OPT),debug)
        F90FLAGS = -O1 -fbounds-check -fbacktrace -ffixed-line-length-132  -I/usr/include
else
        F90FLAGS = -O3 -ffixed-line-length-132  -I/usr/include
endif
```

Link BLAS, LAPack, FFTW

Compiler Specific Flags

“Include” directories

- RAYLEIGH_OPT1 is passed through build_rayleigh
- Also RAYLEIGH_OPT2 and RAYLEIGH_OPT3

USING BUILD_RAYLEIGH FLAGS:

TRY THIS:

```
$ ./build_rayleigh CU debug
```

Sets $RAYLEIGH_OPT1 to debug

Enables debugging flags in Makefile_CIG

NO DEBUGGING FOR NOW PLEASE!

RERUN THIS:

```
$ ./build_rayleigh CU
```

OUR FIRST RUN: PREPWORK (WINDOW 2)

- We will run the code in WINDOW 2 from the scratch directory.
- Change to scratch directory and create module1 subdirectory:

```
$ cd /scratch/summit/user00XX
$ mkdir module1
$ cd module 1
```

Summit Scratch

- 10 TB default (can be increased)
- Wiped every 90 days (PERIOD)
- No touching …   or whining

- Softlink the executable.

```
$ export RADIR=/projects/user00XX/rayleigh
$ ln –s $RADIR/build/rayleigh .
```

OUR FIRST RUN: PREPWORK (WINDOW 2)

- Each simulation requires an input file (run parameters)
- Grab this file from input_examples:

```
$ cd /scratch/summit/user00XX/module1
$ cp $RADIR/input_examples/c2001_case0_minimal   .
```

- Rayleigh expects its input to be named “main_input”
- Rename the file to “main_input”

```
$ mv  c2001_case0_minimal    main_input
```

- Let’s edit main_input

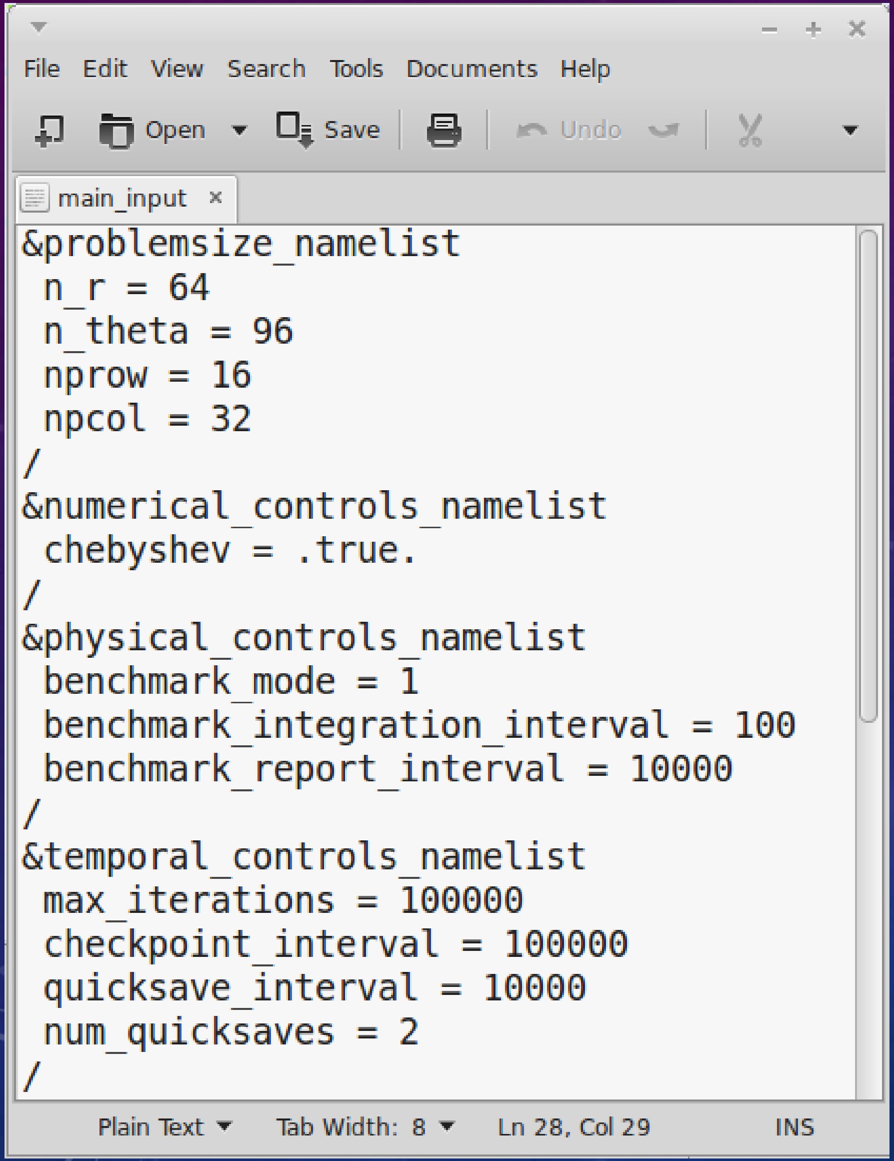
```
$ nano main_input
```

..etc..

QUICK NANO SURVIVAL TIPS

- We will use nano as our editor of choice
- To open a file from shell prompt: nano filename
- Some useful commands from within nano:
  - ctrl + o   - save changes
  - ctrl + x   - exit
  - ctrl + k   - cut
  - ctrl + u   - paste

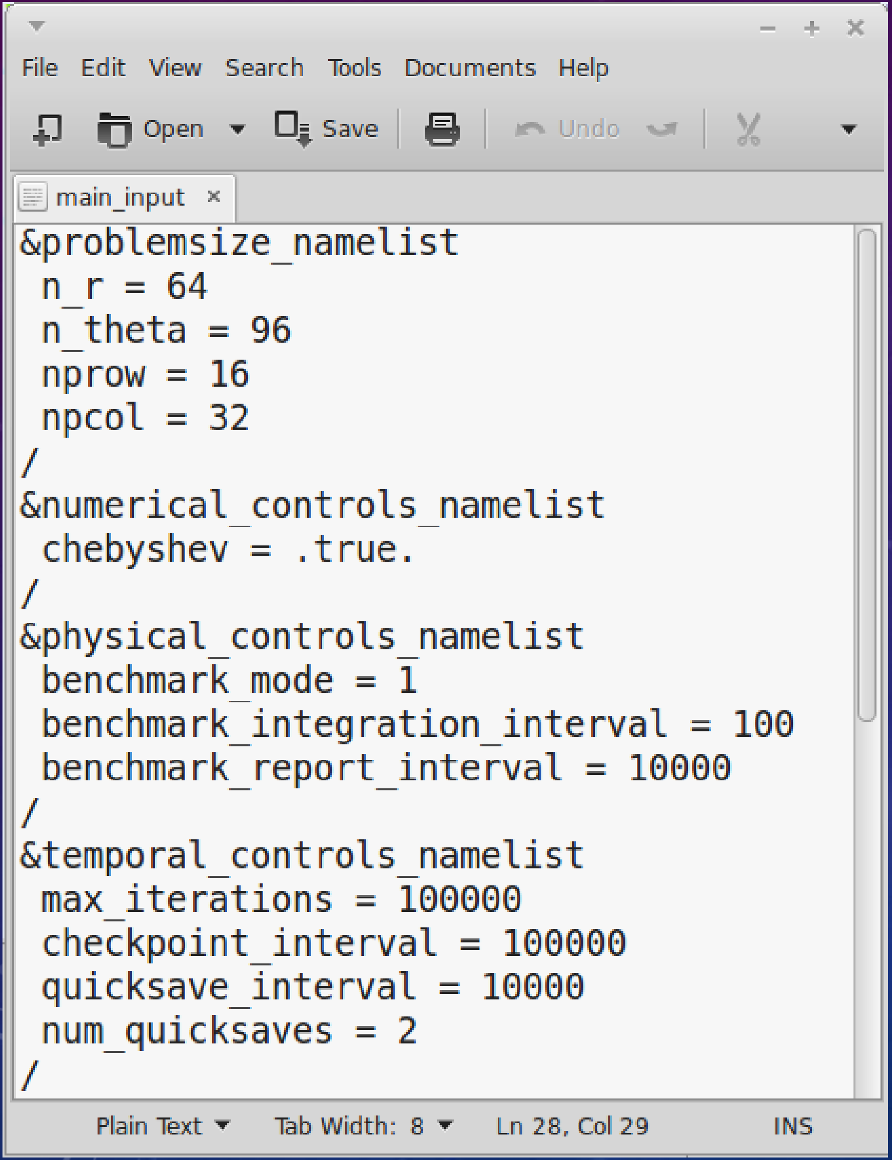
MAIN_INPUT



```
&problemsize_namelist
 n_r = 64
 n_theta = 96
 nprow = 16
 npcol = 32
/
&numerical_controls_namelist
 chebyshev = .true.
/
&physical_controls_namelist
 benchmark_mode = 1
 benchmark_integration_interval = 100
 benchmark_report_interval = 10000
/
&temporal_controls_namelist
 max_iterations = 100000
 checkpoint_interval = 100000
 quicksave_interval = 10000
 num_quicksaves = 2
/
```

- Instructions from the user
- Flags override defaults
- Grouped into namelists
- Namelists control different aspects of the simulation.

MAIN_INPUT



```
&problemsize_namelist
 n_r = 64
 n_theta = 96
 nprow = 16
 npcol = 32
/
&numerical_controls_namelist
 chebyshev = .true.
/
&physical_controls_namelist
 benchmark_mode = 1
 benchmark_integration_interval = 100
 benchmark_report_interval = 10000
/
&temporal_controls_namelist
 max_iterations = 100000
 checkpoint_interval = 100000
 quicksave_interval = 10000
 num_quicksaves = 2
/
```

Modify these values …

nprow = 8
npcol = 8
max_iterations = 30000

… and save.

Ctrl+o

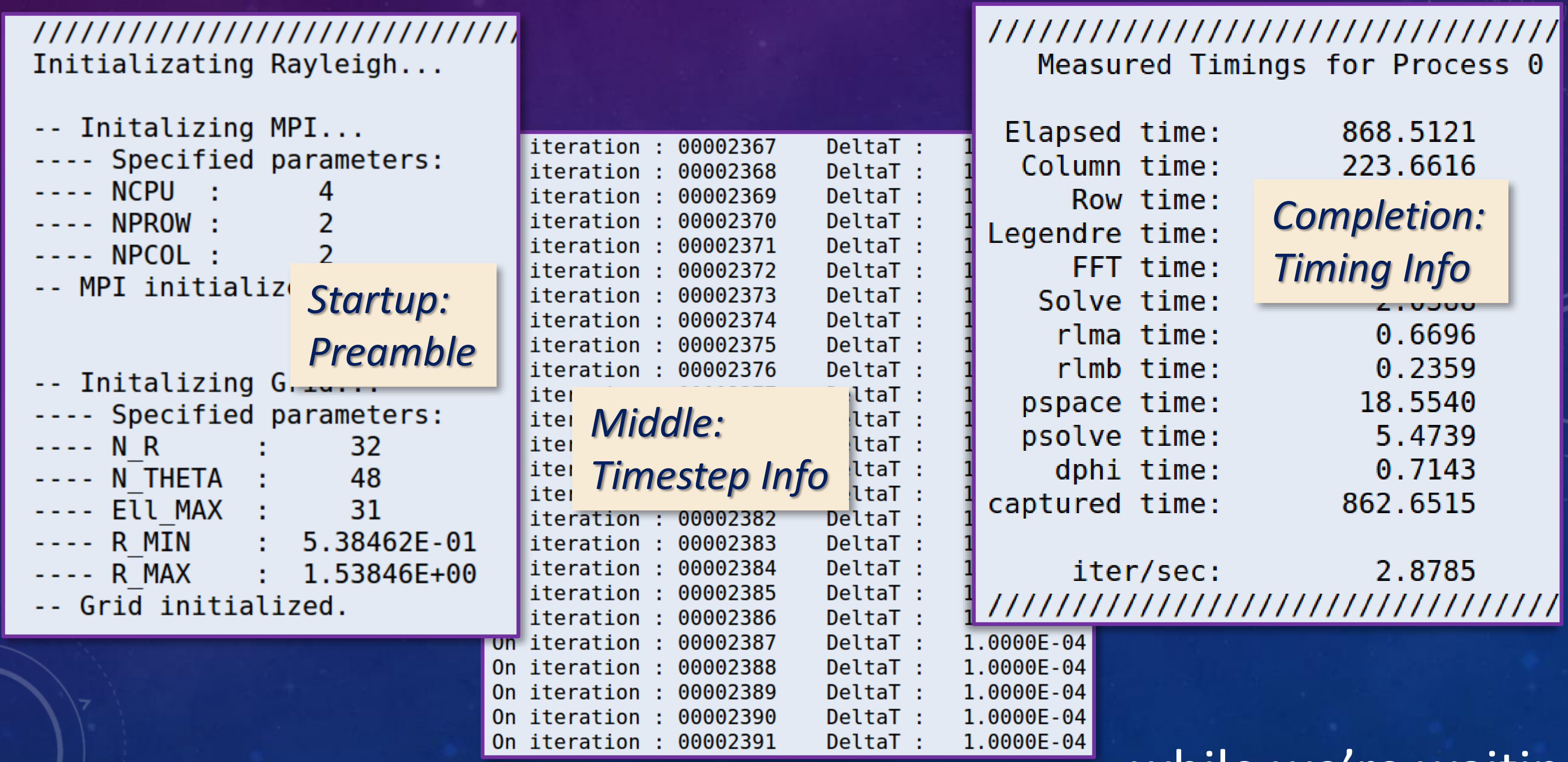
Ctrl+x

OUR FIRST RUN (WINDOW 2)

- Run the code

```
$ cp /projects/feathern/rayleigh_script .
$ sbatch rayleigh_script
$ more slurm*
```

- You will see:



```
////////////////////////////////
Initializating Rayleigh...

-- Initalizing MPI...
---- Specified parameters:
---- NCPU   :       4
---- NPROW  :       2
---- NPCOL  :       2
-- MPI initializ


-- Initalizing G
---- Specified parameters:
---- N_R        :      32
---- N_THETA    :      48
---- Ell_MAX    :      31
---- R_MIN      :   5.38462E-01
---- R_MAX      :   1.53846E+00
-- Grid initialized.
```

Startup: Preamble

```
   iteration : 00002367    DeltaT :   1
   iteration : 00002368    DeltaT :   1
   iteration : 00002369    DeltaT :   1
   iteration : 00002370    DeltaT :   1
   iteration : 00002371    DeltaT :   1
   iteration : 00002372    DeltaT :   1
   iteration : 00002373    DeltaT :   1
   iteration : 00002374    DeltaT :   1
   iteration : 00002375    DeltaT :   1
   iteration : 00002376    DeltaT :   1
   iter                      ltaT :   1
   iter                      ltaT :   1
   iter                      ltaT :   1
   iter                      ltaT :   1
   iter                      ltaT :   1
   iteration : 00002382    DeltaT :   1
   iteration : 00002383    DeltaT :   1
   iteration : 00002384    DeltaT :   1
   iteration : 00002385    DeltaT :   1
   iteration : 00002386    DeltaT :   1
On iteration : 00002387    DeltaT :   1.0000E-04
On iteration : 00002388    DeltaT :   1.0000E-04
On iteration : 00002389    DeltaT :   1.0000E-04
On iteration : 00002390    DeltaT :   1.0000E-04
On iteration : 00002391    DeltaT :   1.0000E-04
```

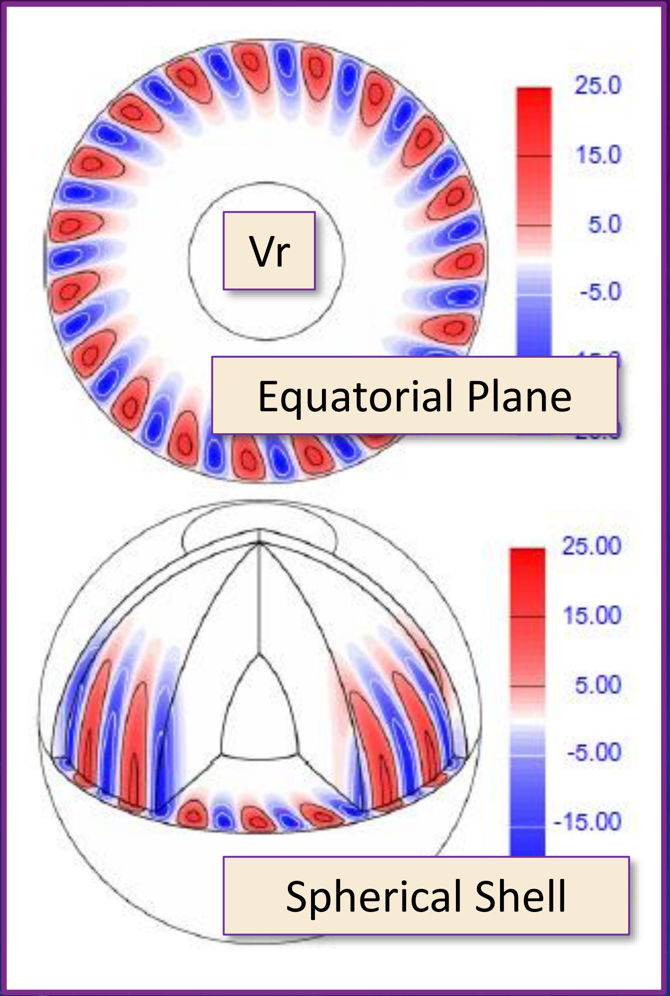
Middle: Timestep Info

```
//////////////////////////////////
   Measured Timings for Process 0

 Elapsed time:          868.5121
  Column time:          223.6616
     Row time:
Legendre time:
     FFT time:
   Solve time:
    rlma time:            0.6696
    rlmb time:            0.2359
  pspace time:           18.5540
  psolve time:            5.4739
    dphi time:            0.7143
captured time:          862.6515

     iter/sec:            2.8785
//////////////////////////////////
```

Completion: Timing Info

…while we're waiting…

IN-SITU BENCHMARKING



- Fully nonlinear, but low-Re
- Steady-state with rotating pattern
- Predefined set of analyses
- When porting: run a benchmark!

Benchmark Inputs

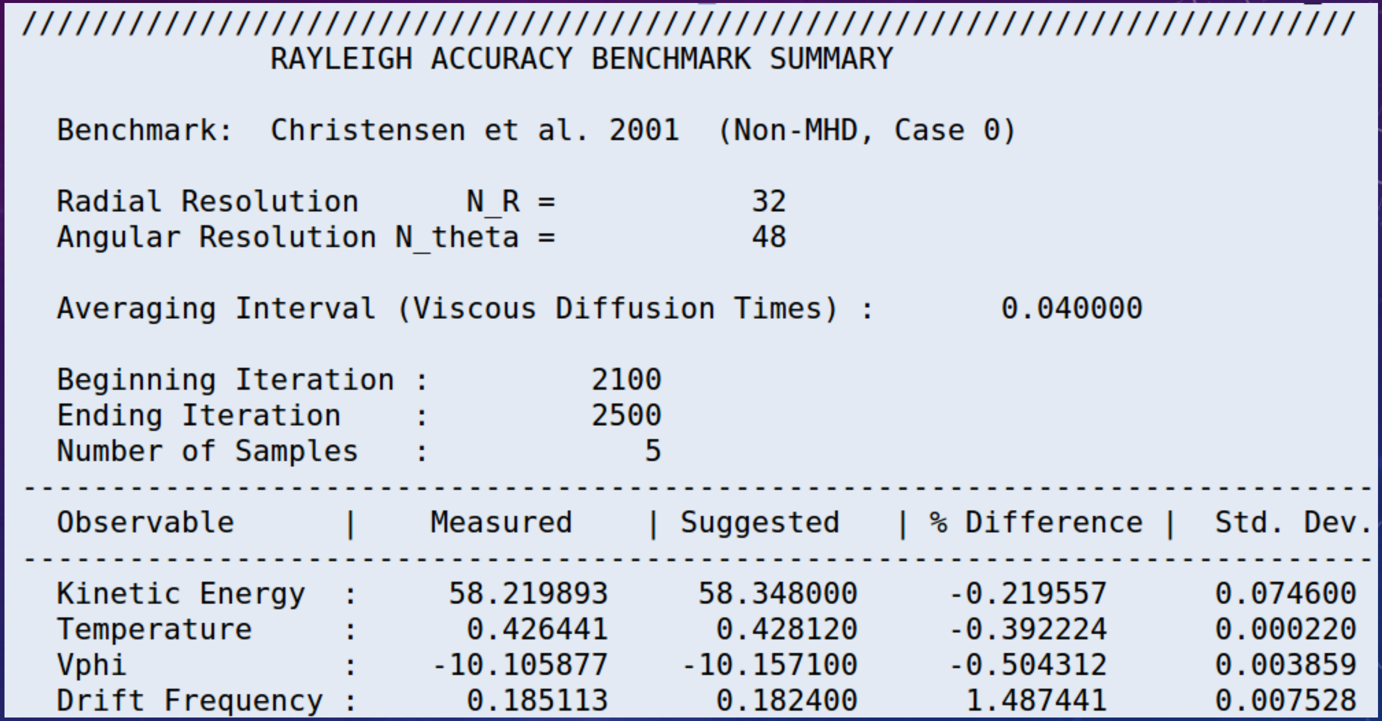
- Boussinesq: Christensen et al. 2001, PEPI, 128, 25
  - c2001_case0_minimal (hydro)
  - c2001_case1_minimal (MHD)

Cheap!

- Anelastic: Jones et al., 2011, Icarus, 216, 120
  - input_examples/j2011_hydro_steady_minimal
  - input_examples/j2011_mhd_steady_minimal

CHECK YOUR RESULTS

```
$ more Benchmark_Reports/00030000
```



```
/////////////////////////////////////////////////////////////////////////
              RAYLEIGH ACCURACY BENCHMARK SUMMARY

  Benchmark:  Christensen et al. 2001  (Non-MHD, Case 0)

  Radial Resolution      N_R =          32
  Angular Resolution N_theta =          48

  Averaging Interval (Viscous Diffusion Times) :       0.040000

  Beginning Iteration :         2100
  Ending Iteration    :         2500
  Number of Samples   :            5
--------------------------------------------------------------------------
  Observable        |     Measured     | Suggested     | % Difference |   Std. Dev.
--------------------------------------------------------------------------
  Kinetic Energy   :     58.219893     58.348000      -0.219557      0.074600
  Temperature      :      0.426441      0.428120      -0.392224      0.000220
  Vphi             :    -10.105877    -10.157100      -0.504312      0.003859
  Drift Frequency :      0.185113      0.182400       1.487441      0.007528
```

- Normally % Difference will be well under 1%
- This example is not equilibrated -- need ~ 30,000 time steps

Questions?