RAYLEIGH TUTORIAL CIG TNG 2025

MODULE 1: BUILDING AND PORTING THE CODE

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

IN THIS MODULE:

- Obtaining a copy of Rayleigh
- Important Resources
- Code requirements/porting considerations
- Compile the code
- Verify the code is working

Obtaining Rayleigh

Stable Releases (latest is v 1.2.0)

- CIG Website: https://geodynamics.org/software
- Github: v/releases
- Zendo: https://zenodo.org/records/11391213

Latest Development Version

https://github.com/geodynamics/Rayleigh

FOR TODAY: Copy of 1.2.0 provided in tutorial disk image file

Important Resources

- Github:
  - https://github.com/geodynamics/Rayleigh/issues
  - Feel free to post an issue!
- Documentation:
  - https://rayleigh-documentation.readthedocs.io/en/latest/index.html

CODE REQUIREMENTS:

(Necessary to build and run Rayleigh)

- Fortran compiler (2003 or higher)
- C++ compiler
- BLAS
- LAPack
- FFTW version 3.x or later
- MPI
- Python 3, Numpy, Matplotlib (for output analysis only)

* Intel’s Math Kernel Library (MKL) provides interfaces to BLAS, LAPack, and FFTW

* Conda environment setup describe in the docs

BUILDING THE CODE (preparation)

- Open a terminal window
- Set Intel-compiler-relevant environment variables for future login shells

```
$ echo “source /opt/intel/oneapi/setvars.sh” >> ~/.profile
```

- Close and reopen terminal window
- Copy and extract the source code

```
$ cp /rayleigh_tutorial/Rayleigh-1.2.0.tgz  ~/.
$ cd ~
$ tar –xvf Rayleigh-1.2.0.tgz
$ cd Rayleigh-1.2.0
```

BUILDING THE CODE

- Once you’re run “make install,” look in the bin directory:

```
$ ls bin
```

- rayleigh.opt
  - compiled using optimization flags
  - designed for production runs
  - No debugging info provided on crash
- rayleigh.dbg
  - Compiled without optimization flags
  - Provides filename and line number when crash encountered
  - Intended for development/debugging purposes

Running the Code: Overview

Every Rayleigh simulation starts with the same three steps…

1. Create a unique directory in which to store the simulation data
2. Copy or soft-link the rayleigh executables into the directory
3. Place a Rayleigh “main_input” file into the directory

- Several “main_input” examples can be found in Rayleigh/input_examples
- Let’s walk through these steps together to run our first model

OUR FIRST RUN: Accuracy Benchmark

- Always verify your install by running an accuracy benchmark!

- Create a unique directory for the model

```
$ cd ~
$ mkdir module1
$ cd module 1
```

- Softlink the executable.

```
$ ln –s ~/Rayleigh-1.2.0/bin/rayleigh.opt .
```

- Place a main_input file into the directory

```
$ cp ~/Rayleigh-1.2.0/input_examples/c2001_case0_minimal    main_input
```

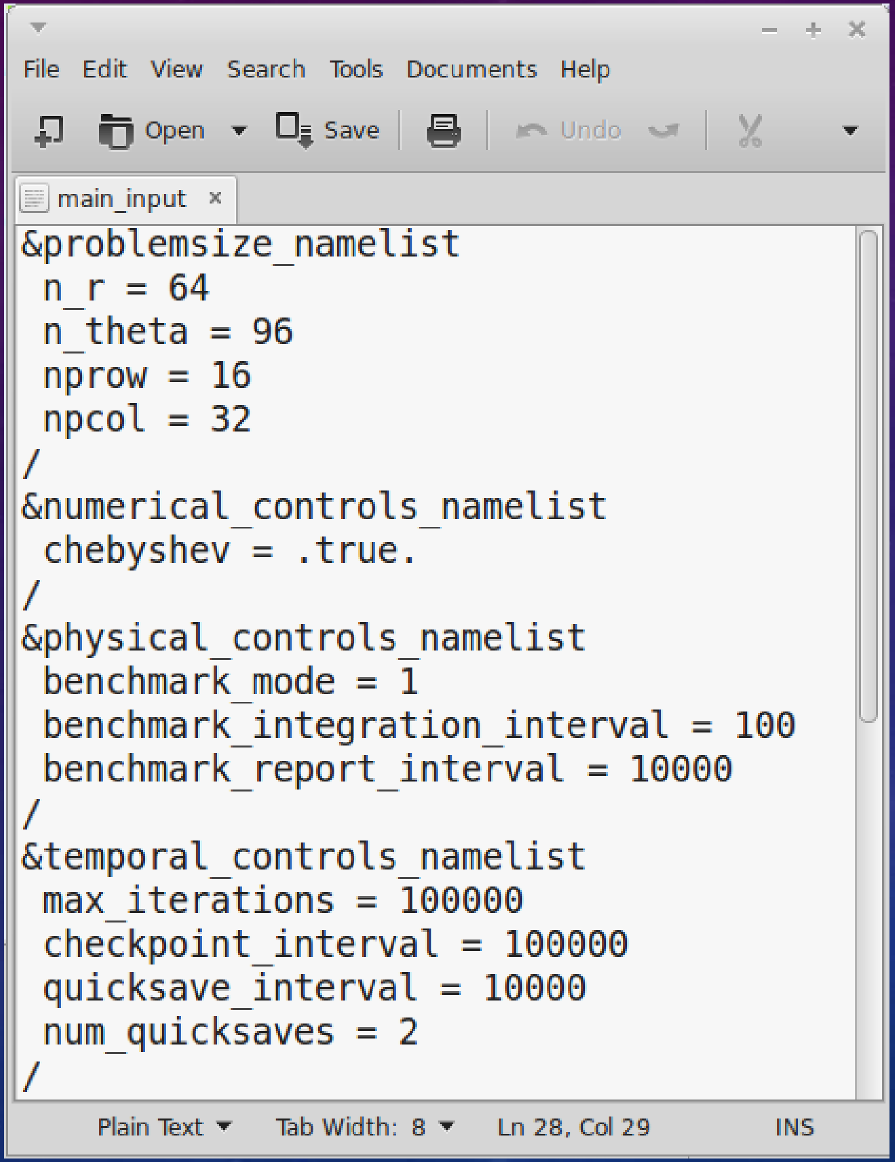
- Let’s edit main_input

```
$ nano main_input
```

QUICK NANO SURVIVAL TIPS

- If you don’t have a preferred editor, nano works well in the terminal
- To open a file from shell prompt: nano filename
- The only commands you really need:
  - ctrl + o - save changes
  - ctrl + x - exit
  - ctrl + k - cut
  - ctrl + u - paste

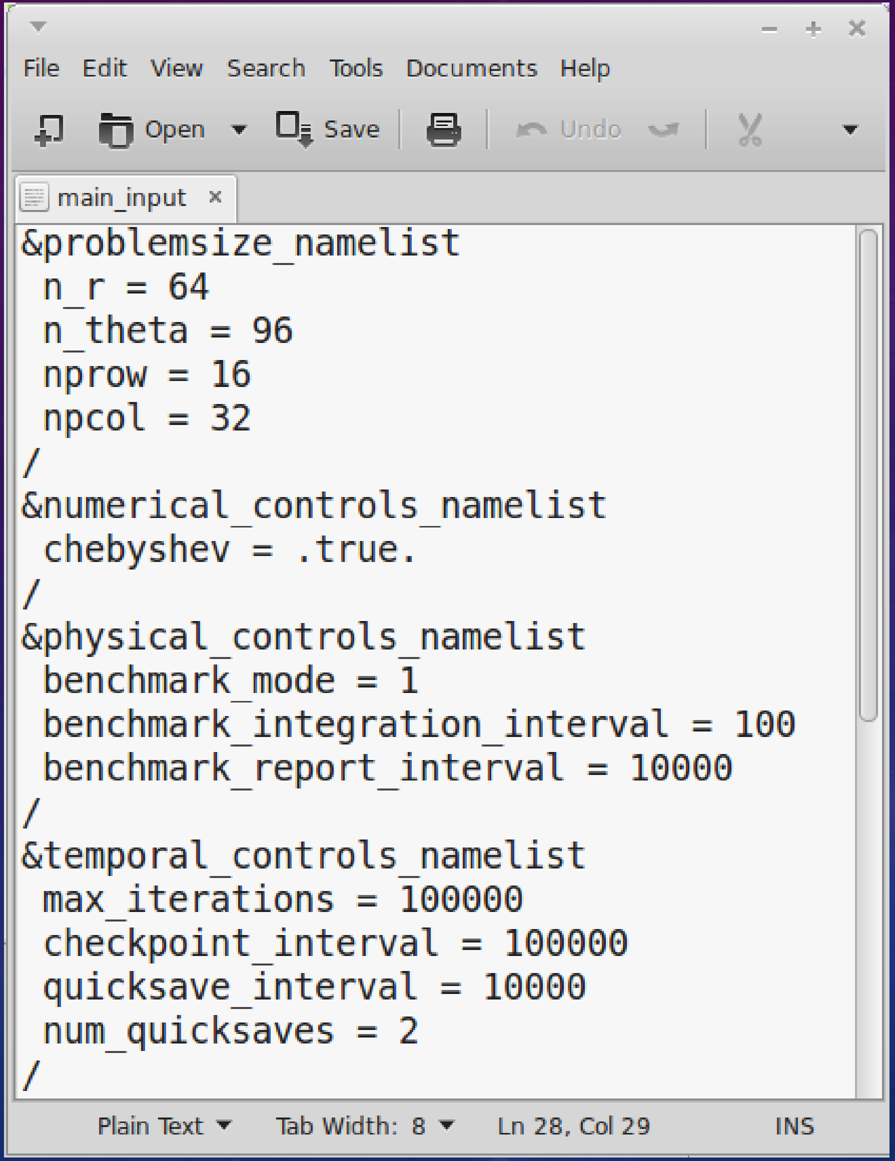
MAIN_INPUT



```
&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 = 2
npcol  = 2
max_iterations = 40000
```

… and save.

Ctrl+o

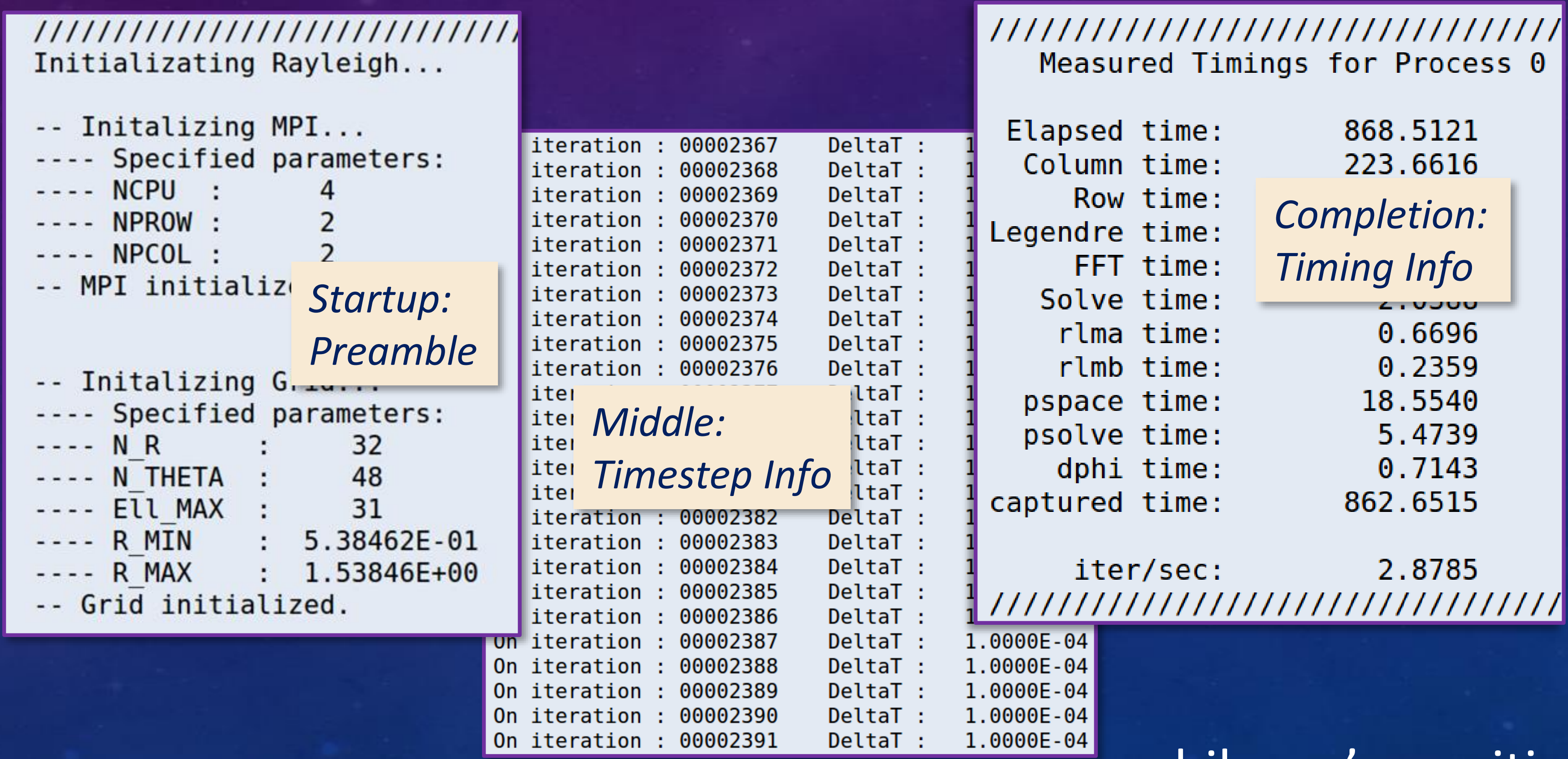
Ctrl+x

OUR FIRST RUN

- Run the code…

```
$ mpiexec –np 4 ./rayleigh.opt
```

- You will see:



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

-- Initalizing MPI...
---- Specified parameters:
---- NCPU  :        4
---- NPROW :        2
---- NCOL  :        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
```

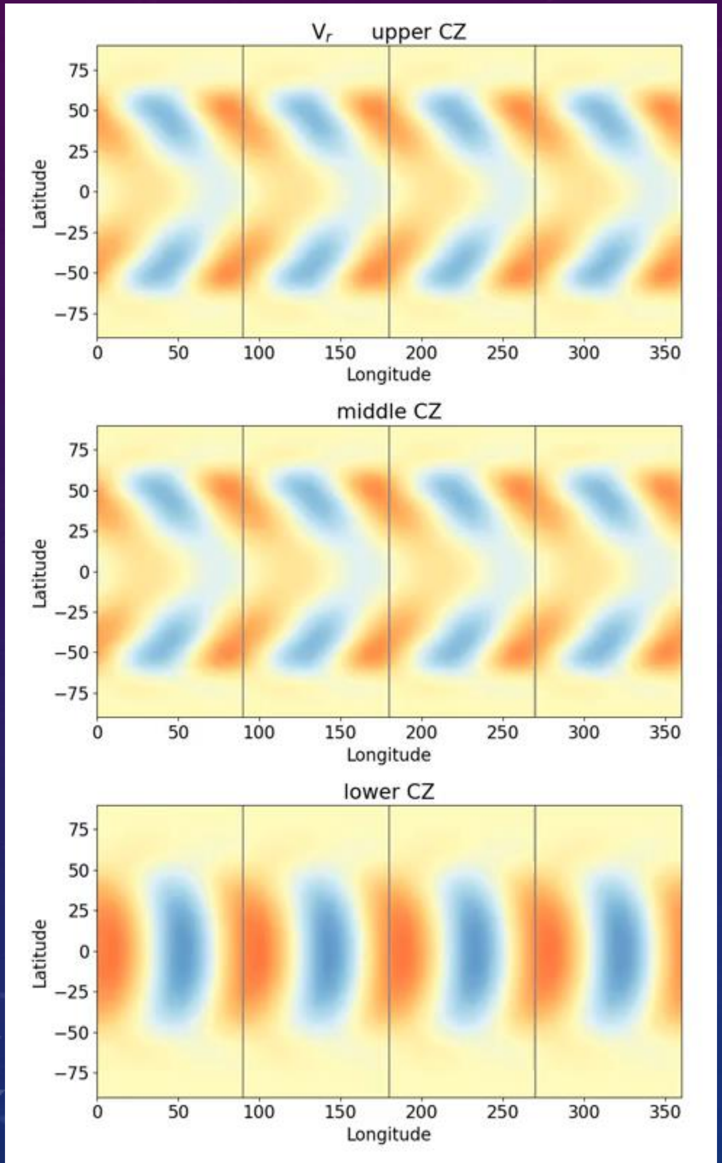
Middle: Timestep Info

```
////////////////////////////////////
  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

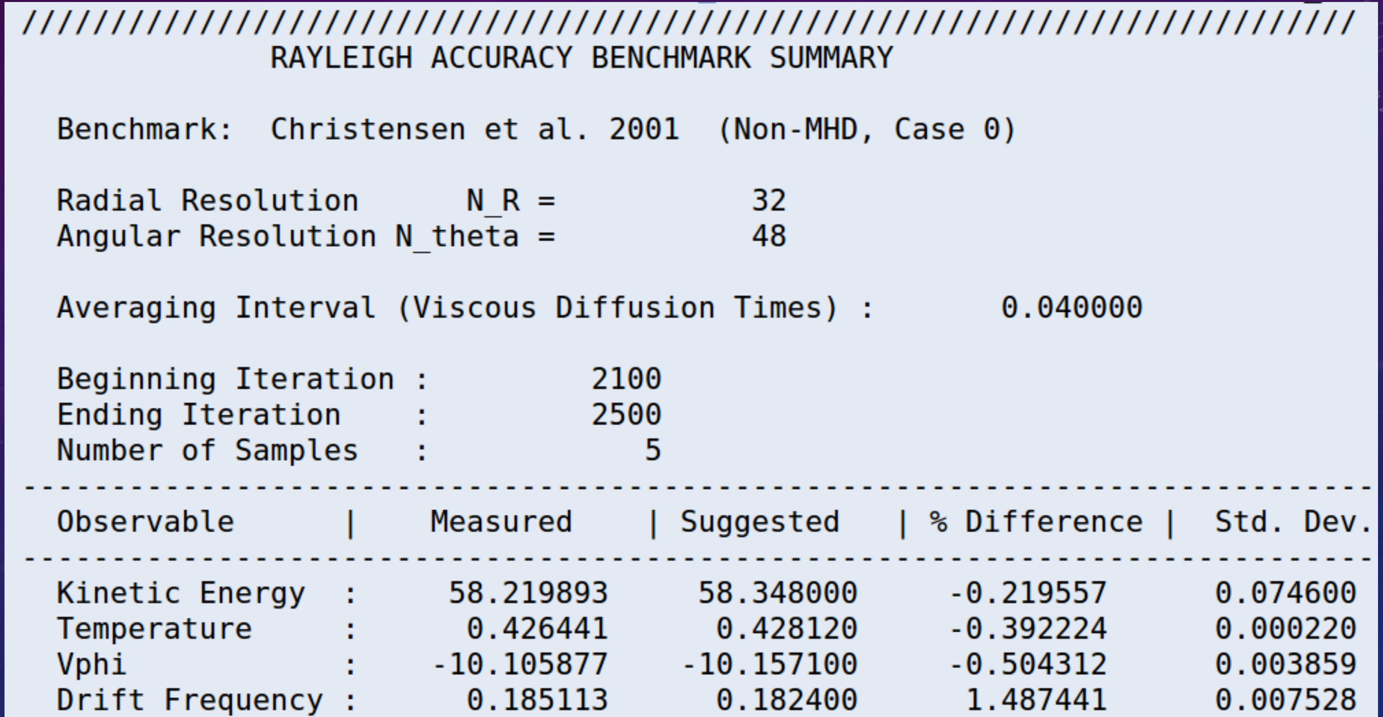
- Boussinesq: Christensen et al. 2001, PEPI, 128, 25
  - input_examples/c2001_case0_minimal (hydro)
  - Input_examples/c2001_case1_minimal (MHD)

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/00040000
```



```
///////////////////////////////////////////////////////////////////////////
             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%
- Need ~ 30,000 time steps

Questions?