A Crash Course in the GYRE Stellar Oscillation Code

Rich Townsend / MadStar Group
University of Wisconsin-Madsion

What does GYRE do?

- GYRE solves the system of equations governing small periodic perturbations to an equilibrium stellar state
- Solutions take the form

$$\xi_r(r,\theta,\phi;t) = \mathrm{Re}\left[\sqrt{4\pi}\tilde{\xi}_r(r) Y_\ell^m(\theta,\phi)\exp(-\mathrm{i}\sigma t)\right]$$

$$\boldsymbol{\xi}_\mathrm{h}(r,\theta,\phi;t) = \mathrm{Re}\left[\sqrt{4\pi}\tilde{\xi}_h(r) r \nabla_\mathrm{h} Y_\ell^m(\theta,\phi)\exp(-\mathrm{i}\sigma t)\right]$$

$$f'(r,\theta,\phi;t) = \mathrm{Re}\left[\sqrt{4\pi}\tilde{f}'(r) Y_\ell^m(\theta,\phi)\exp(-\mathrm{i}\sigma t)\right]$$

- Solutions which satisfy all boundary conditions can only found for discrete values of the frequency σ — these are the *eigenfrequencies* of the star

Installing GYRE

- Download and install MESA SDK

  http://www.astro.wisc.edu/~townsend/static.php?ref=mesasdk

- Download GYRE

  https://bitbucket.org/rhdtownsend/gyre/downloads/gyre-5.2.tar.gz

- Build and test GYRE

```
$ tar xfz gyre-5.2.tar.gz
$ cd gyre
$ make -j
$ make test
$ export GYRE_DIR=`pwd`
```

Running a Test Problem

- Make a space to work

```
$ cd $GYRE_DIR
$ mkdir work
$ cd work
```

- Grab a stellar model (MESA model of SPB star)

```
$ cp ../models/mesa/spb/spb.mesa .
```

- Create a GYRE input file (from GYRE's test suite)

```
$ cp ../test/ad/mesa/spb/gyre.in .
```

- Run GYRE

```
$ ../bin/gyre gyre.in
```

Output: Initialization & Model Read

- Initialization

```
gyre [5.2]
----------

OpenMP Threads    : 4
Input filename    : gyre.in
```

- Read the model, repair any obvious issues

```
Model Init
----------

Reading from MESA file
   File name spb.mesa
   File version 1.00
   Read 1814 points
   No need to add central point
```

Output: Mode Search

- Begin a search for modes with $\ell=1$, $m=1$

```
Mode Search
-----------

Mode parameters
   l : 1
   m : 0
```

- Search for eigenfrequencies in a specified range

```
Building frequency scan
   added scan interval :  0.1500E+00 ->  0.4500E+00 (100 points, INVERSE)
```

- Solve for eigenfunctions on a radial grid of 1842 points

```
Building x grid
   Found inner turning points, x range 0.1038 -> 0.1041
   Adding 0 inner point(s)
   Adding 28 global point(s) in iteration 1
   Adding 0 global point(s) in iteration 2
   Final grid has 1 segment(s) and 1842 point(s):
      Segment 1 : x range 0.0000 -> 1.0000 (1 -> 1842)
```

Output: Eigenfrequencies!

```
Starting search (adiabatic)

Root bracketing
  Time elapsed :      1.685 s

Root Solving
    l     m     n_pg     n_p     n_g         Re(omega)         Im(omega)        chi  n_iter
    1     0      -33       0      33   0.15311422E+00  0.00000000E+00 0.1346E-13       6
    1     0      -32       0      32   0.15817553E+00  0.00000000E+00 0.3377E-13       7
    1     0      -31       0      31   0.16396253E+00  0.00000000E+00 0.2423E-13       7
    1     0      -30       0      30   0.16875293E+00  0.00000000E+00 0.8124E-14       6
    1     0      -29       0      29   0.17374720E+00  0.00000000E+00 0.1326E-13       7
    1     0      -28       0      28   0.18079054E+00  0.00000000E+00 0.3573E-16       6
    1     0      -27       0      27   0.18784331E+00  0.00000000E+00 0.6048E-14       6
    1     0      -26       0      26   0.19305723E+00  0.00000000E+00 0.1983E-13       7
    1     0      -25       0      25   0.20112428E+00  0.00000000E+00 0.1422E-13       7
    1     0      -24       0      24   0.21039452E+00  0.00000000E+00 0.2982E-13       7
    1     0      -23       0      23   0.21728250E+00  0.00000000E+00 0.5762E-13       6
    1     0      -22       0      22   0.22628995E+00  0.00000000E+00 0.1047E-13       6
    1     0      -21       0      21   0.23798874E+00  0.00000000E+00 0.4412E-14       6
    1     0      -20       0      20   0.24766095E+00  0.00000000E+00 0.4686E-13       6
    1     0      -19       0      19   0.25810394E+00  0.00000000E+00 0.7205E-14       7
    1     0      -18       0      18   0.27401057E+00  0.00000000E+00 0.1123E-14       6
    1     0      -17       0      17   0.28880565E+00  0.00000000E+00 0.8277E-13       6
    1     0      -16       0      16   0.30124205E+00  0.00000000E+00 0.7310E-15       7
```

$$\omega = \sqrt{(R^3/GM)}\sigma$$

Input: Constants & Model

- The gyre.in file is a Fortran namelist file, consisting of `name=value` pairs collected into named groups
- Don't override any of GYRE's physical constants (an example of an empty namelist group)

```
&constants
/
```

- Read an evolutionary model, in MESA's output format, from the file spb.mesa

```
&model
        model_type = 'EVOL'
        file = 'spb.mesa'
        file_format = 'MESA'
/
```

Input: Mode Parameters

- Separately search for modes with ℓ=1,2,3

```
&mode
        l = 1
        tag = 'l=1'
/

&mode
        l = 2
        tag = 'l=2'
/

&mode
        l = 3
        tag = 'l=3'
/
```

- The tags allow different grids, numerical schemes, etc. to be used for modes with different ℓ. They can be any textual string

Input: Oscillation & Numerical Parameters

- Set up oscillation parameters (e.g., alternative boundary conditions, treatment of rotation, non adiabatic effects)

```
&osc
/
```

- Set up numerical parameters; here, use a finite differencing scheme based on 4th order Gauss-Legendre collocation

```
&num
        diff_scheme = 'COLLOC_GL4'
/
```

Digression: The Clamped String

- Consider the linear wave PDE for a string clamped at both ends:

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 y}{\partial t^2}$$

$$y = \begin{cases} 0 & x = 0 \\ 0 & x = L \end{cases}$$

- Periodic solutions take the form

$$y(x;t) = \mathrm{Re}\left[\tilde{y}(x)\exp(-\mathrm{i}\sigma t)\right]$$

- Eigenfunctions must satisfy the ODE

$$\frac{\mathrm{d}^2\tilde{y}}{\mathrm{d}x^2} = -\frac{\sigma^2}{c^2}\tilde{y}$$

Characteristic Equation for the Clamped String

- Solutions to the ODE:

$$\tilde{y}(x) = A\sin(\sigma x/c)$$

- These solutions always satisfy the BC $y(0){=}0$, but only satisfy the BC $y(L){=}0$ when

$$\sin(\sigma x/c) = 0$$

- This is the *characteristic equation* governing which frequencies are eigenfrequencies of the clamped string

- If we didn't know *a priori* where the zeros of the cosine function are located, we would treat this as a numerical root finding problem:

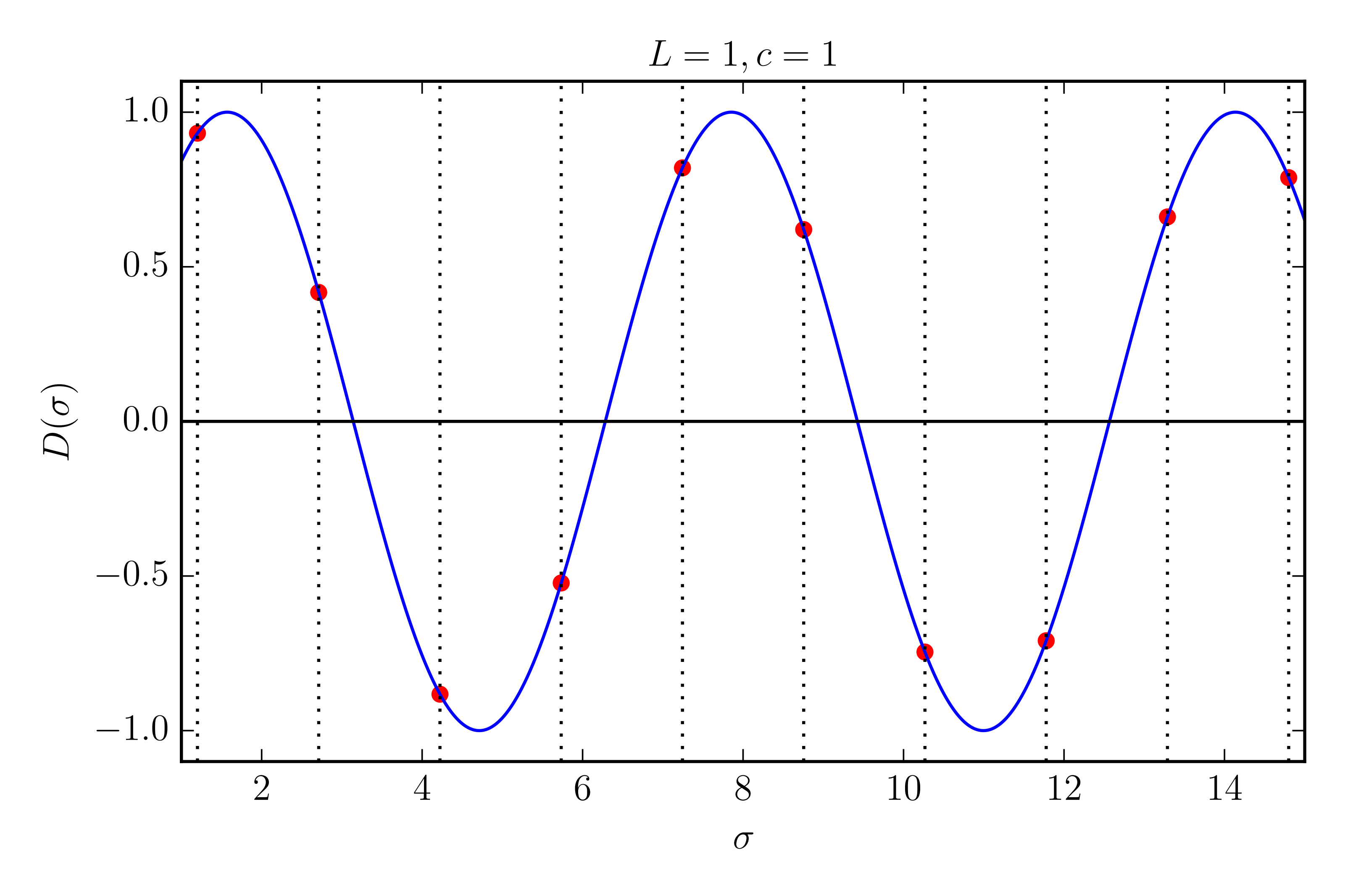
$$\text{Find all } \sigma \text{ such that } D(\sigma) \equiv \sin(\sigma x/c) = 0$$

Numerical Root Finding

- Evaluate the discriminant function $D(\sigma)$ at a discrete sequence of frequencies $\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_N$
- If the discriminant changes sign across a pair of points σ_i,σ_{i+1}, then there must be an odd number (ideally, just one) root of $D(\sigma)$ within the bracket $[\sigma_i,\sigma_{i+1}]$
- Use bisection (or a more sophisticated algorithm; see Numerical Recipes) to narrow down the bracket until the root is found

Frequency Scan Strategies

- When scanning frequencies to look for sign changes, the distribution of points $\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_N$ should adequately sample the expected distribution of roots

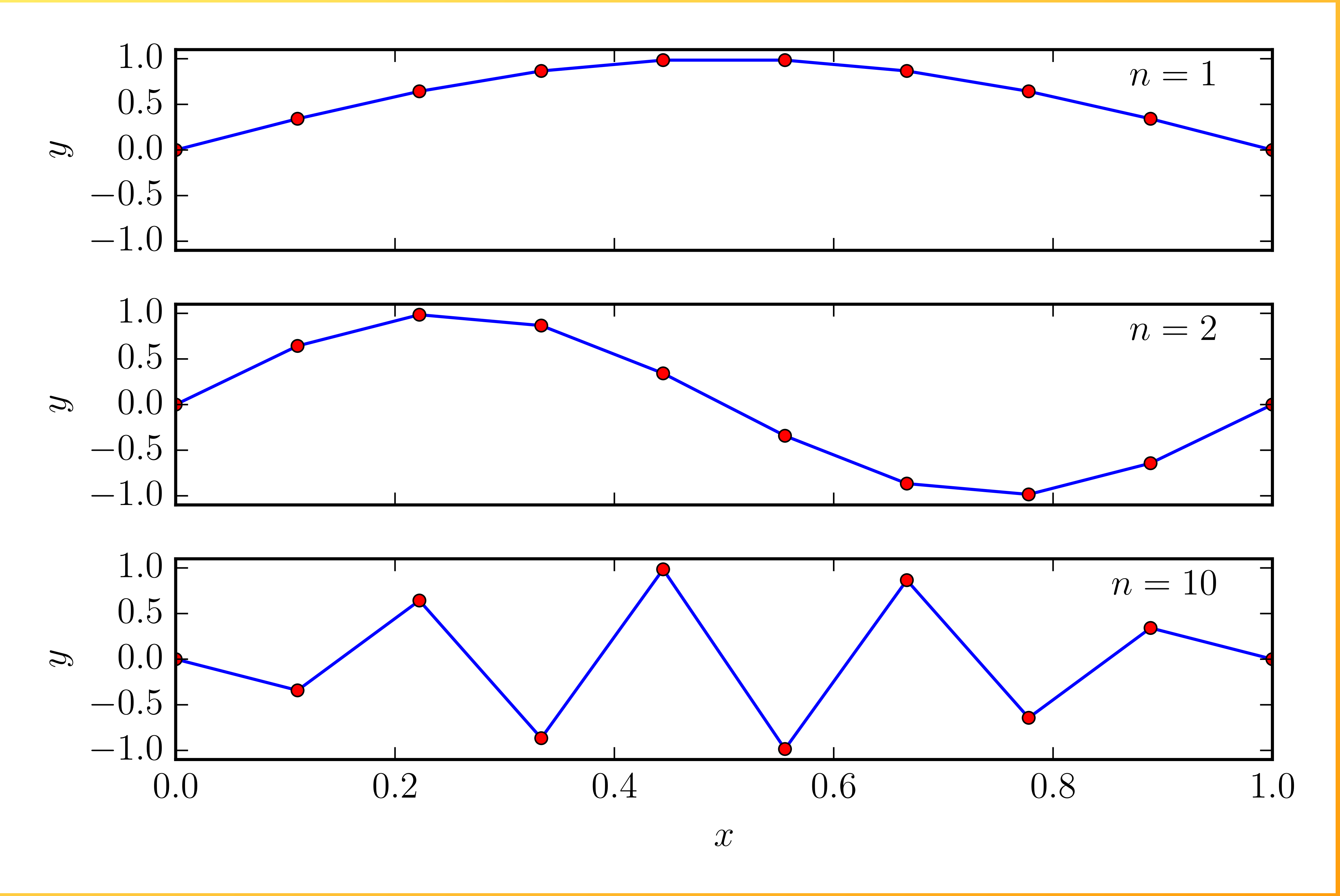


Input: Frequency Scan

- Search for ℓ=1 modes (matching the tag '`l=1`') by scanning
  - from ω=0.15 to ω=0.45
  - with 100 frequency points
  - spaced uniformly in inverse frequency (because g modes asymptote to uniform spacing in period = ω^{-1})

```
&scan
        grid_type = 'INVERSE'
        freq_min = 0.15
        freq_max = 0.45
        n_freq = 100
        tag_list = 'l=1'
/
```

The Clamped String: Eigenfunctions



Take away: spatial (x) grid must adequately resolve eigenfunctions; otherwise, eigenfrequencies will be bogus!

Input: Spatial Grid

- The ODEs governing the eigenfunctions are solved on a spatial (x) grid based on the model grid
- Additional points are inserted based on criteria

```
&grid
        alpha_osc = 10
        alpha_exp = 2
        n_inner = 5
/
```

- Add 10 points per wavelength in oscillatory regions
- Add 2 points per e-folding length in evanescent regions
- Add 5 points between the center & the inner turning point

Output

- As well as screen output, GYRE can write summary files (e.g., frequencies) and mode files (e.g., wavefunctions)
- The files & data written are controlled by two namelist groups — one adiabatic, one non-adiabatic
- The TXT file format has the same data layout as MESA's profile & history files

```
&ad_output
    summary_file = 'summary.txt'
    summary_file_format = 'TXT'
    summary_item_list = 'l,n_pg,n_p,n_g,omega'
/

&nad_output
/
```

Next Steps

- Try looking at some of the other test examples in the *test* subdirectory (incl. non-adiabatic calculations in *test/nad*)
- For running GYRE inside MESA, look at the *gyre_in_mesa_** test suite examples (next MESA release)
- If you have questions, post them at the GYRE discussion forums:

http://www.astro.wisc.edu/~townsend/gyre-forums/