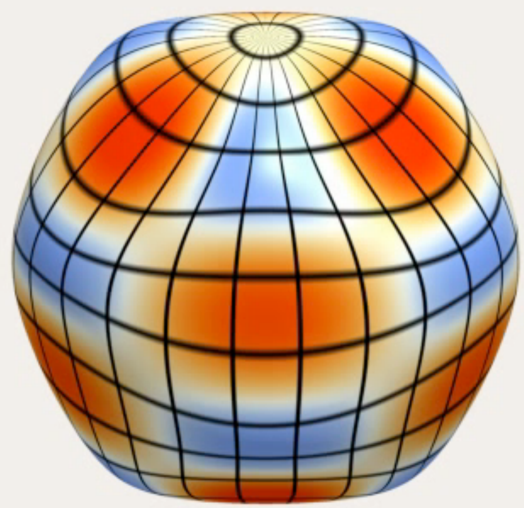
Asteroseismology with MESA

Rich Townsend

University of Wisconsin-Madison



MESA

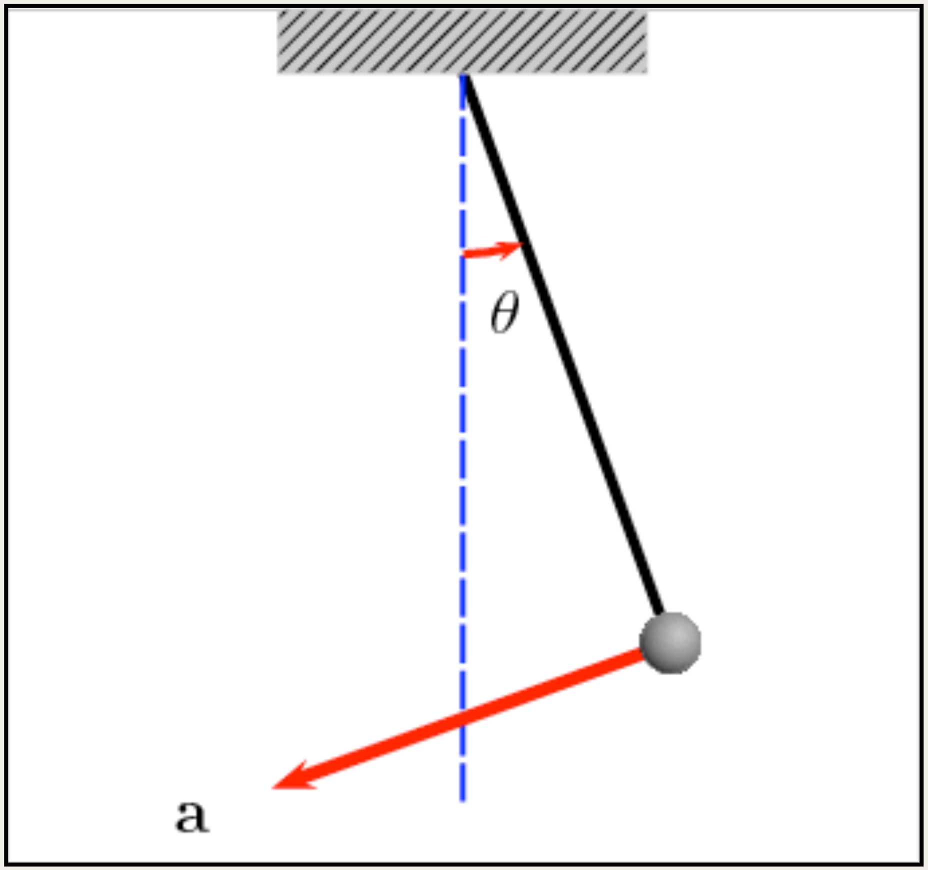
The MESA Software Development Kit (SDK)

- What's it for?
  - Hassle-free compilation of MESA
  - Works on Linux and Mac OS X (Intel-based)
- What's in it?
  - gcc/gfortran 4.7 compilers (good support for Fortran 2003)
  - BLAS/LAPACK libraries (linear algebra)
  - PGPLOT library (graphics)
  - HDF5 library (file storage)
- Where do I get it from?
  - http://www.astro.wisc.edu/~townsend/static.php?ref=mesasdk
- How do I install?
  - Linux: unpack tar archive (anywhere; don't need to be root user)
  - OS X: drag package into Applications folder
- Should I stop using ifort etc?
  - No! Diversity improves software
  - MESA follows *standards* conformance, not *compiler* conformance

A: A dynamically stable self-gravitating gaseous sphere

Dynamical Stability → Oscillation

Ideal Pendulum



Restoring Force: brings displaced mass back to equilibrium point

+

Inertia: causes mass to overshoot equilibrium point

=

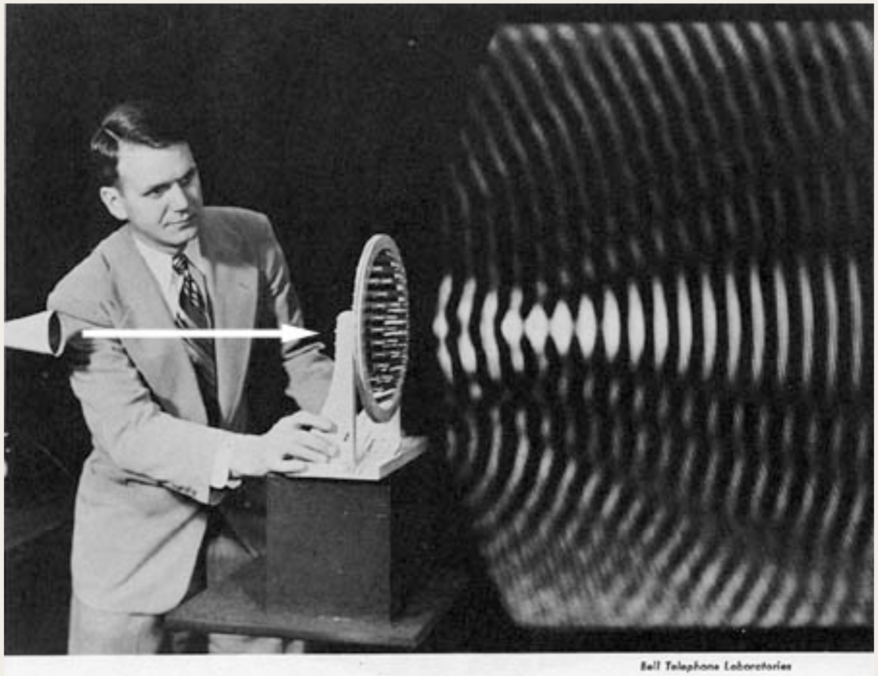
Oscillation at fixed frequency $\omega = \sqrt{(g/L)}$

Oscillations in Stars

- Star in hydrostatic equilibrium: $\nabla p = \varrho\, \mathbf{g}$
- Restoring force*s* due to local departure from HE
- LHS: force due to under/over-pressure
- RHS: force due to under/over-bouyancy
- Restoring forces from neighbors' pressure
  - coupled oscillators
  - wave propagation
  - free oscillation at any frequency ω
  - wavelength/wavenumber depends on ω via dispersion relation

Acoustic and Gravity Waves

Pressure → acoustic waves



$$\omega^2 = c^2 |\mathbf{k}|^2$$

Buoyancy → gravity waves

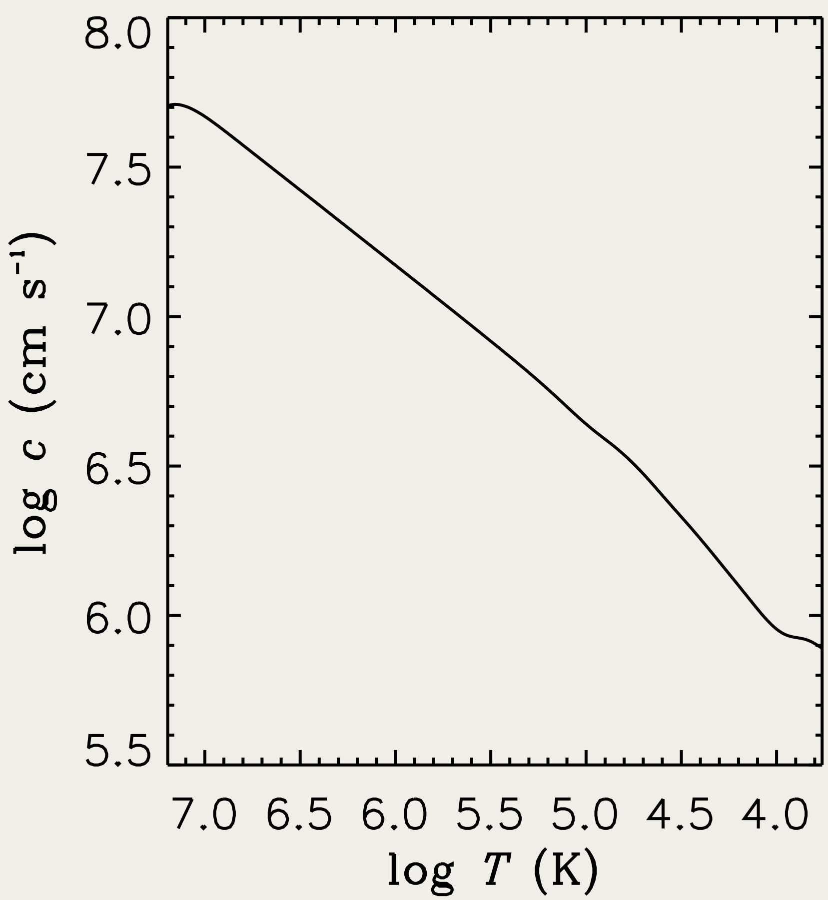
$$\omega^2 = N^2 \frac{|\mathbf{k}_\mathrm{h}|^2}{|\mathbf{k}|^2}$$

$$\mathbf{k} = k_r \hat{\mathbf{r}} + \mathbf{k}_h$$

The Sound Speed

$$c^2 = \frac{\Gamma_1 p}{\varrho}$$

- Local propagation speed of adiabatic acoustic waves
- For ideal gas, depends only on temperature & composition
- Large in the core, small in the envelope



MESA variable: `csound`

The Brunt-Väisälä Frequency

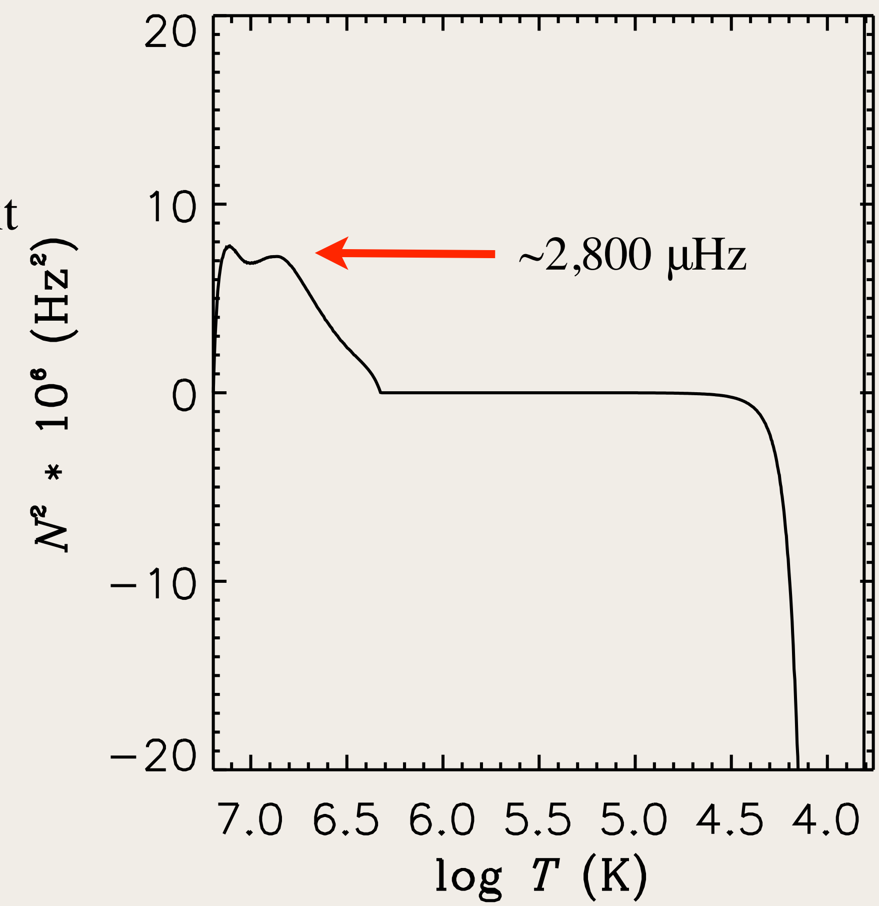
$$N^2 = -\frac{g}{r}\left(\frac{1}{\Gamma_1}\frac{\mathrm{d}\ln p}{\mathrm{d}\ln r} - \frac{\mathrm{d}\ln \varrho}{\mathrm{d}\ln r}\right)$$

- Oscillation frequency of vertically displaced fluid element

Text

- Closely related to Schwarzschild criterion:
  - $N^2 > 0 \rightarrow$ convectively stable
  - $N^2 < 0 \rightarrow$ convectively unstable

MESA variable: `brunt_N2`
(must set `calculate_Brunt_N2` flag)

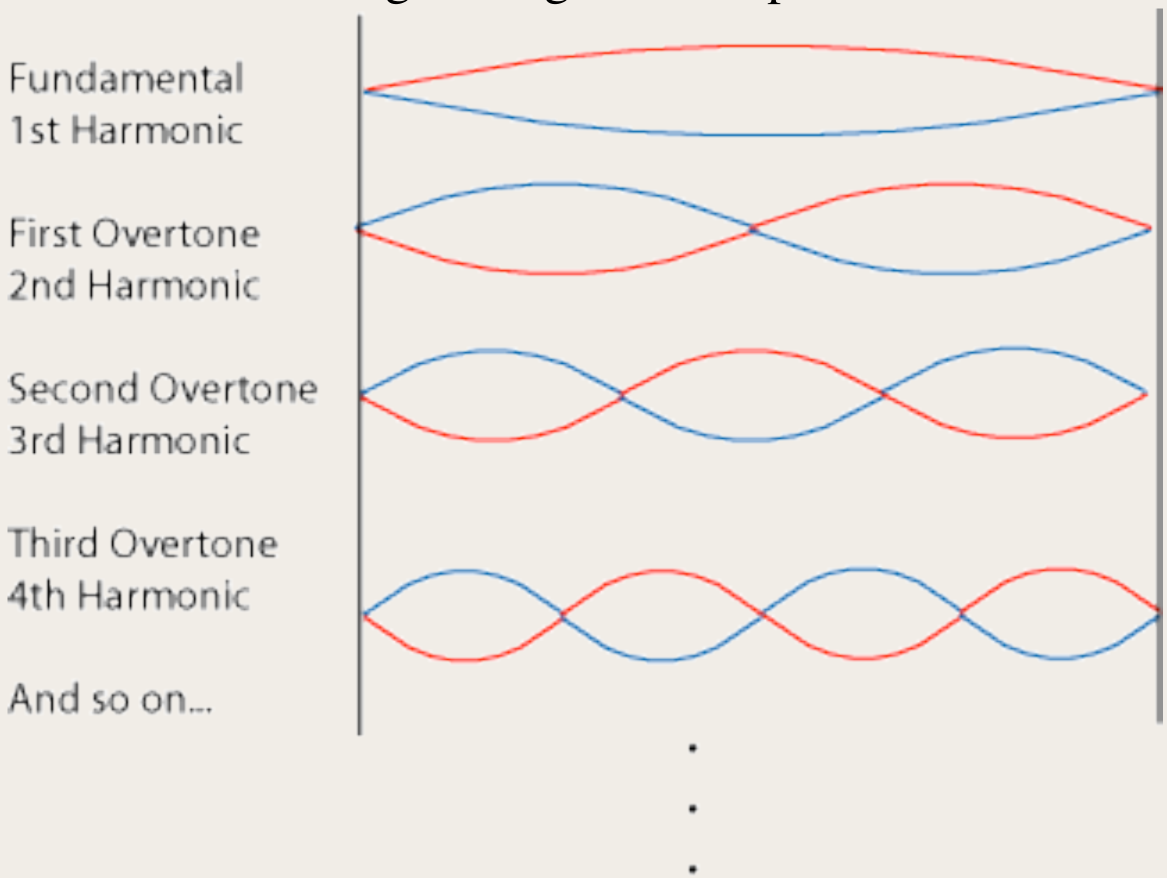


Boundary Conditions...

...restrict the ‘allowed’ oscillation frequencies to a discrete mode spectrum
(ask Mike M!)

Normal Modes 101

Stretched string of length L clamped at each end

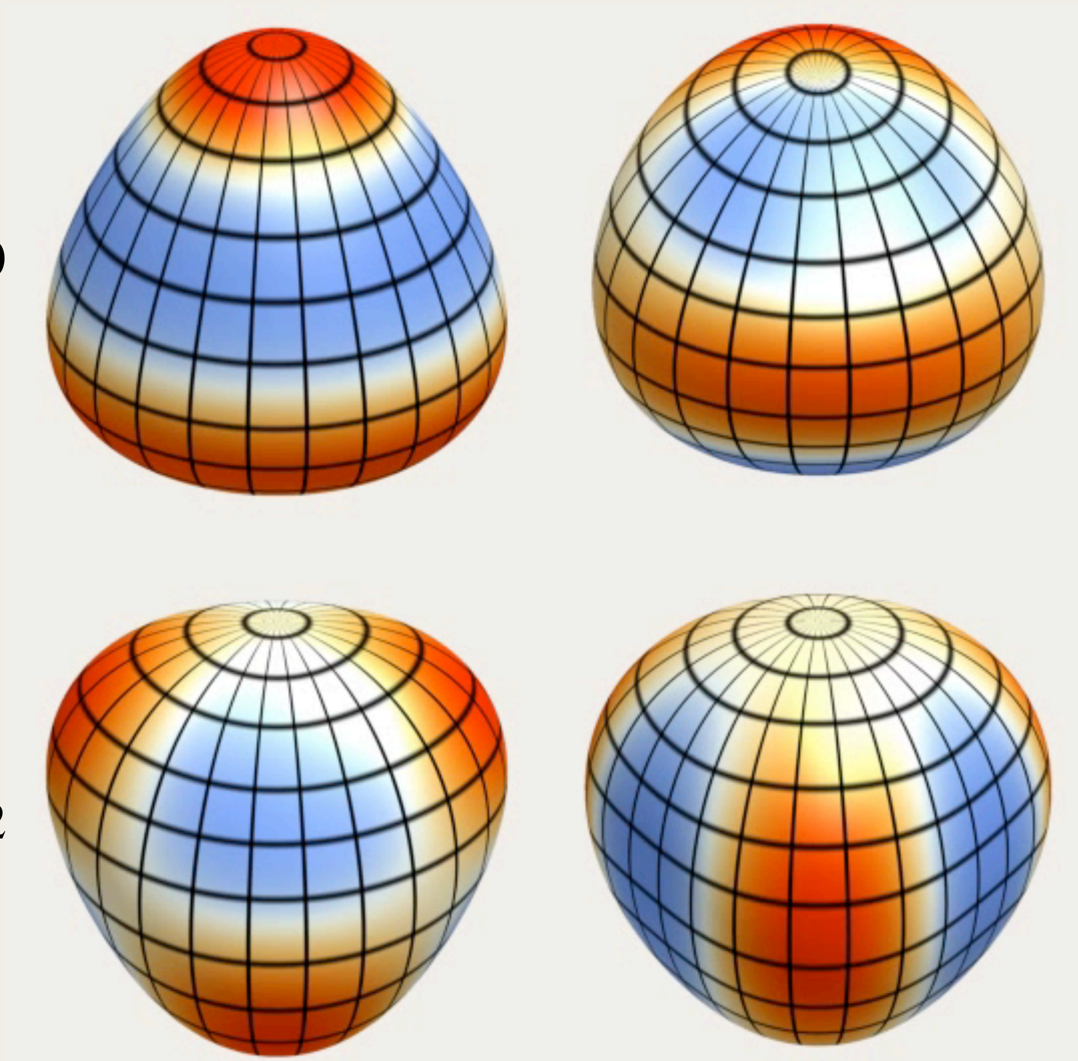


$$\left.\begin{aligned} kL &= n\pi \quad (n = 1, \ldots) \\ \omega &= c\,k \end{aligned}\right\} \longrightarrow \quad \omega = \frac{n\pi c}{L}$$

Horizontal BCs

Periodic boundary conditions → perturbations have spherical-harmonic angular dependence (*think QM!*)

$$\delta f \sim y(r)\, Y_\ell^m(\theta, \phi)$$



$\ell = 3, m = 0$

$\ell = 3, m = 1$

$\ell = 3, m = 2$

$\ell = 3, m = 3$

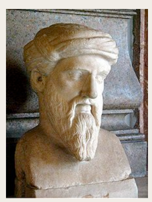
Radial BC

$$\int k_r \, \mathrm{d}r = n\,\pi$$

(cf. $k\,L = n\,\pi$)

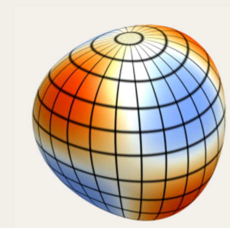
...but what is k_r in terms of ω?

$$|\mathbf{k}|^2 = k_r^2 + |\mathbf{k}_h|^2$$



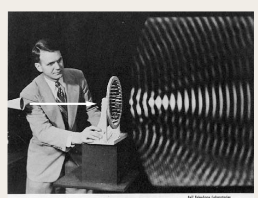
+

$$|\mathbf{k}_h|^2 = \frac{\ell(\ell+1)}{r^2}$$



+

$$\omega^2 = c^2 |\mathbf{k}|^2$$



or

$$\omega^2 = N^2 \frac{|\mathbf{k}_\mathrm{h}|^2}{|\mathbf{k}|^2}$$

After some algebra...

Acoustic Waves: $$k_r = \frac{\omega}{c}\sqrt{1 - \frac{S_\ell^2}{\omega^2}}$$

* Lamb Frequency S_ℓ:

$$S_\ell^2 \equiv \frac{\ell(\ell+1)c^2}{r^2}$$

Gravity Waves: $$k_r = \frac{\sqrt{\ell(\ell+1)}N}{\omega r}\sqrt{1 - \frac{\omega^2}{N^2}}$$

Big result I: Lamb and Brunt-Väisälä frequencies are *critical frequencies* which govern where in a star acoustic and gravity waves (resp.) of a given ω can propagate

Big result II: Combined with the radial BC, these expressions indicate that the normal-mode frequencies $\omega_{n,\ell}$ depend only on n, ℓ , and the stellar structure

Asteroseismology!!!

Introducing ADIPLS

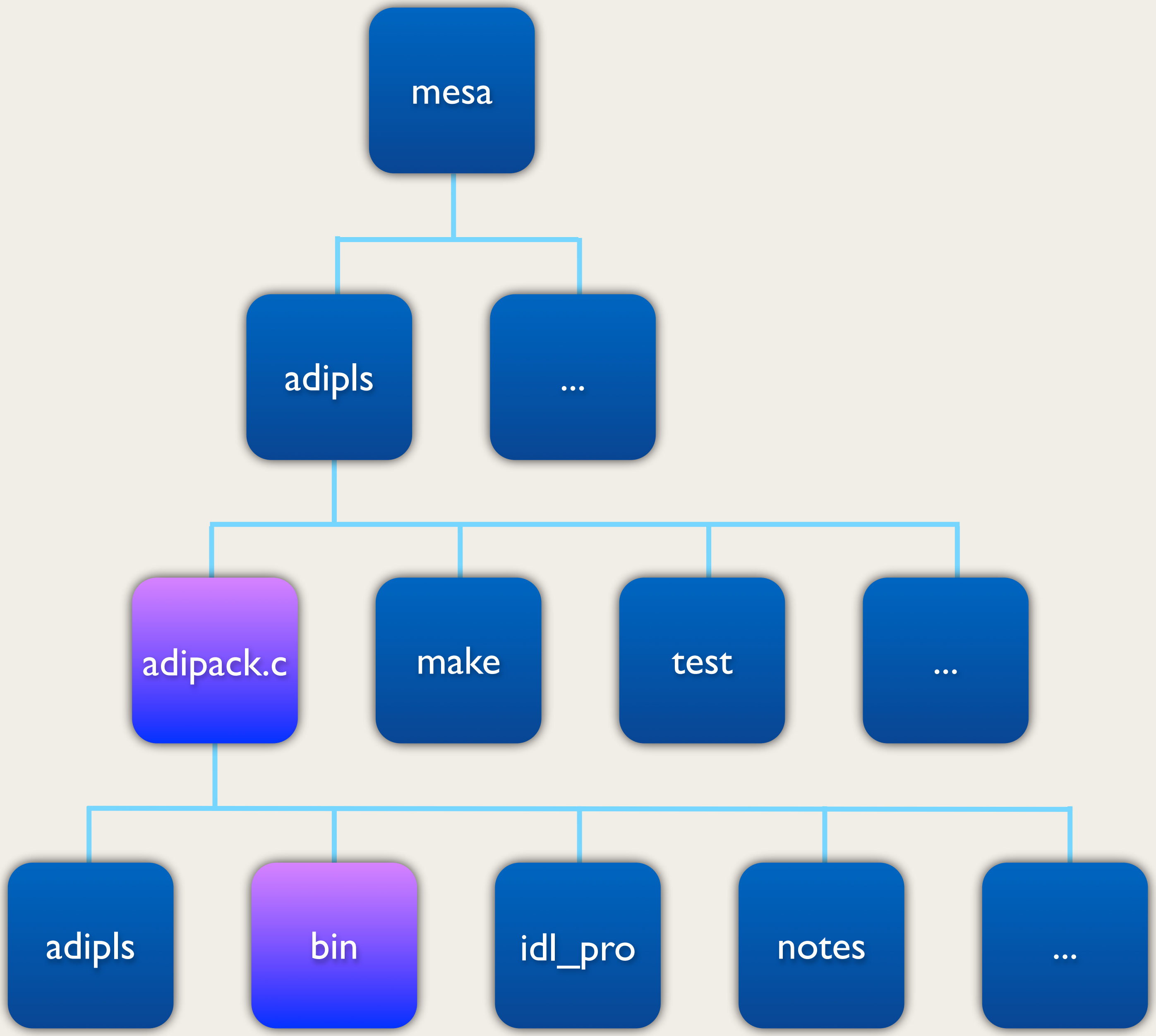
- ADIPLS - the Aarhus adiabatic oscillation package
  - Jørgen Christensen-Dalsgaard
  - Linear, adiabatic, radial/non-radial oscillations
  - Aimed at asteroseismic analyses
  - Well documented (see arXiv 0710.3106)

- MESA includes recent release (adipack.c; 2010)

- Can be operated in three modes
  - stand-alone — run independently of MESA
  - integrated (pulse) — called from within MESA, to calculate frequencies
  - integrated (astero) — called from within MESA, to match observed frequencies

- Not the only option:
  - Liège Oscillation Code (OSC) - adiabatic oscillations (see arXiv 0712.3474v1)
  - Madison Code (GYRE) - non-adiabatic oscillations, differential rotation (open-source)

MESA/ADIPLS Directory Structure



ADIPLS Usage

fgong-amdl.d

- convert FGONG model to AMDL model
- syntax:
  `fgong-amdl.d XXX.fgong amdl.XXX`
- inputs:
  `XXX.fgong (FGONG model)`
- outputs:
  amdl.XXX (AMDL model)

redistrb.d

- regrid AMDL model
- syntax:
  `redistrb.d redistrb.in`
- inputs:
  `redistrb.in` (parameter file)
  `amdl.XXX` (AMDL model)
- outputs:
  amdl.YYY (AMDL model)

adipls.c.d

- pulsation code
- syntax:
  `adipls.c.d adipls.c.in`
- inputs:
  `adipls.c.in` (parameters)
  `amdl.XXX` (stellar model)
- outputs:
  `agsm.XXX` (grand summary)
  `assm.XXX` (small summary)

set-obs.d

- extract frequencies from grand summary
- syntax:
  `set-obs.d 5 agsm.XXX XXX.freq`
- inputs:
  `agsm.XXX` (grand summary)
- outputs:
  `XXX.freq` (ASCII frequencies)

Mini-Lab: A First Look at ADIPLS

- Set environment variables:

Bourne Shell

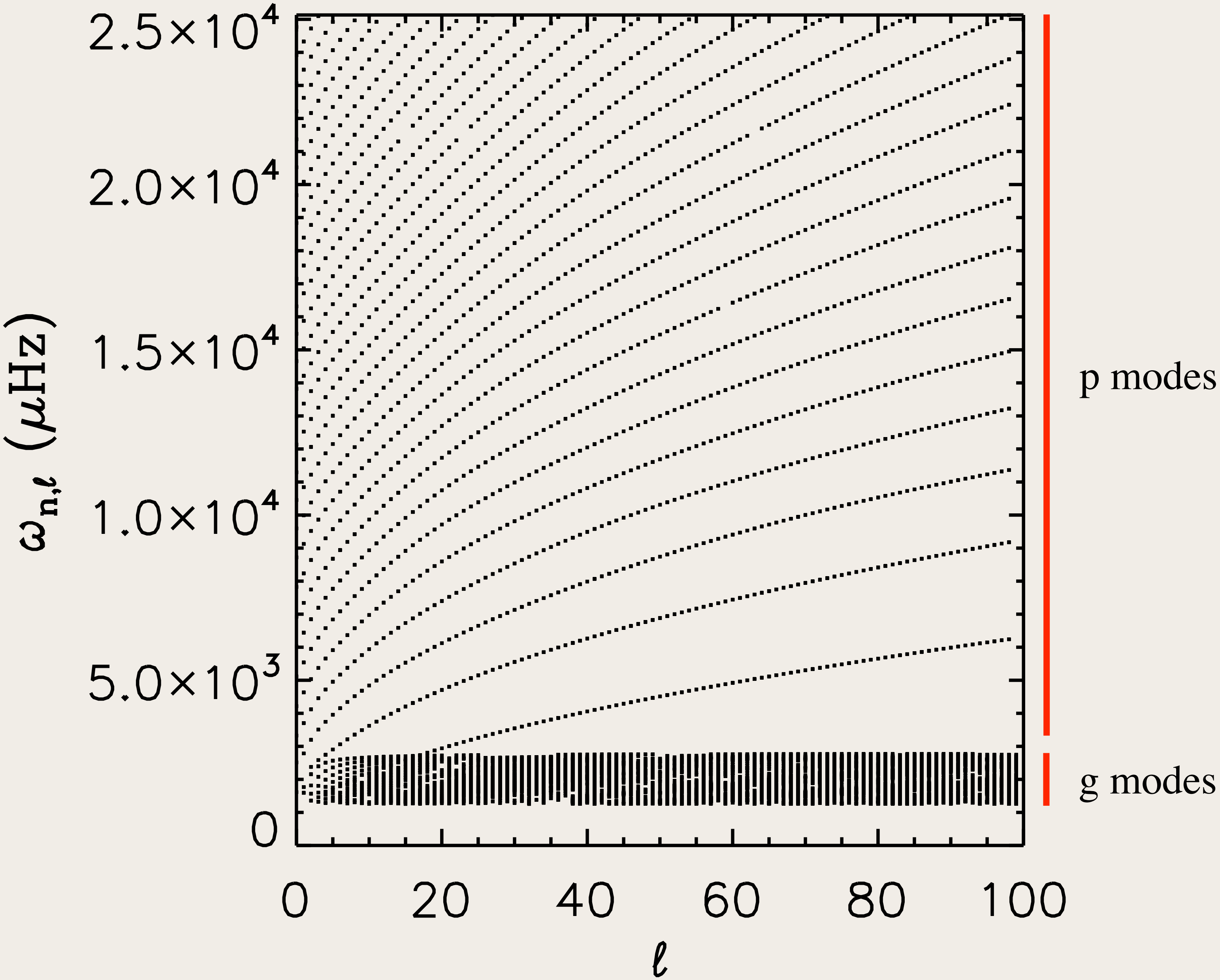
```
export aprgdir=$MESA_DIR/adipls/adipack.c
export PATH=$PATH:$aprgdir/bin
```

C Shell

```
setenv aprgdir $MESA_DIR/adipls/adipack.c
setenv PATH $PATH:$aprgdir/bin
```

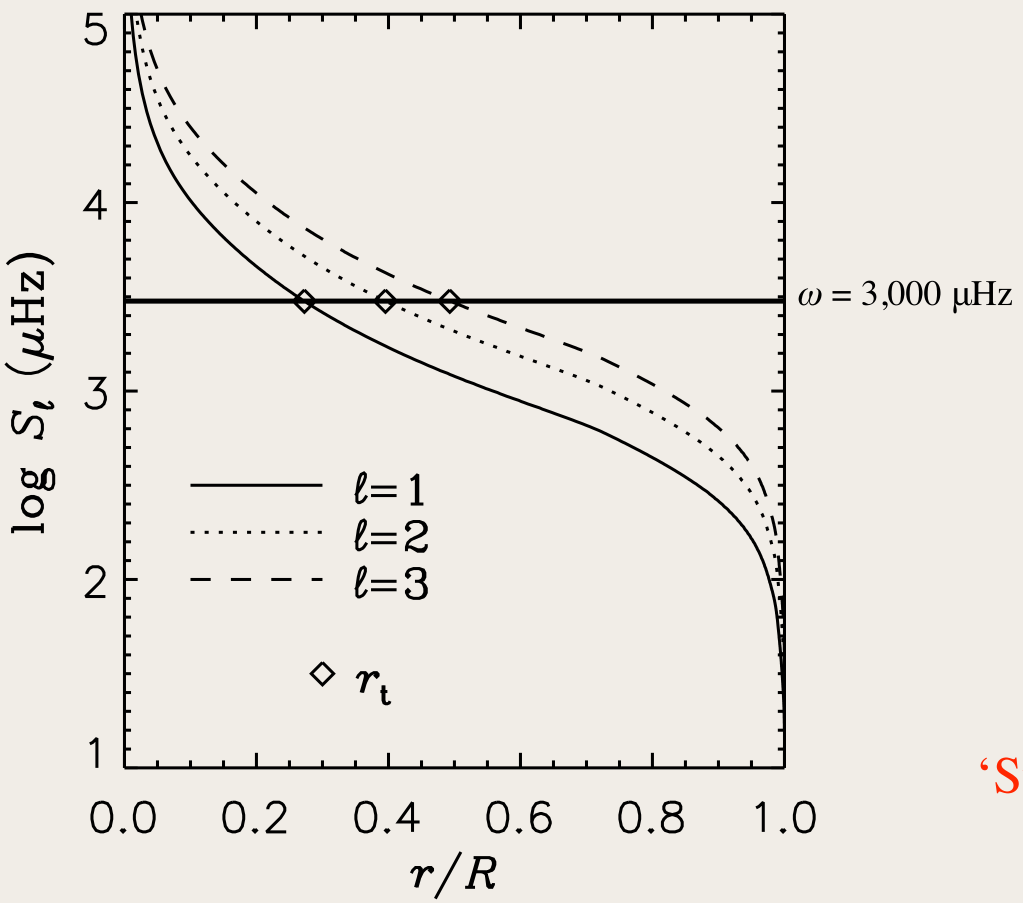
- Change into `solar` subdirectory: `cd solar`
- Create the AMDL stellar model from the FGONG file: `fgong-amdl.d solar.fgong amdl.solar`
- Regrid the AMDL stellar model: `redistrb.d redistrb.in`
- Run ADIPLS to calculate frequencies for modes with $\ell = 0,\ldots,3$: `adipls.c.d adipls.c.in`
- Extract frequencies from the resulting grand summary file: `set-obs.d 5 agsm.solar solar.freq`
- Plot the frequencies $\omega_{n,\ell} = 2\pi\,\nu_{n,\ell}$ as a function of ℓ using the software of your choice

The ‘Solar’ Oscillation Spectrum



Solving for the p-mode Frequency

$$\int_{???} k_r \, \mathrm{d}r = n\,\pi \,, \quad k_r = \frac{\omega}{c}\sqrt{1 - \frac{S_\ell^2}{\omega^2}}$$



‘String’ extends from r_t to R

Solving for the Frequency

$$\int_{r_\mathrm{t}}^{R} \frac{\omega}{c}\sqrt{1-\frac{S_\ell^2}{\omega^2}} = n\,\pi$$

↓ *trivial*

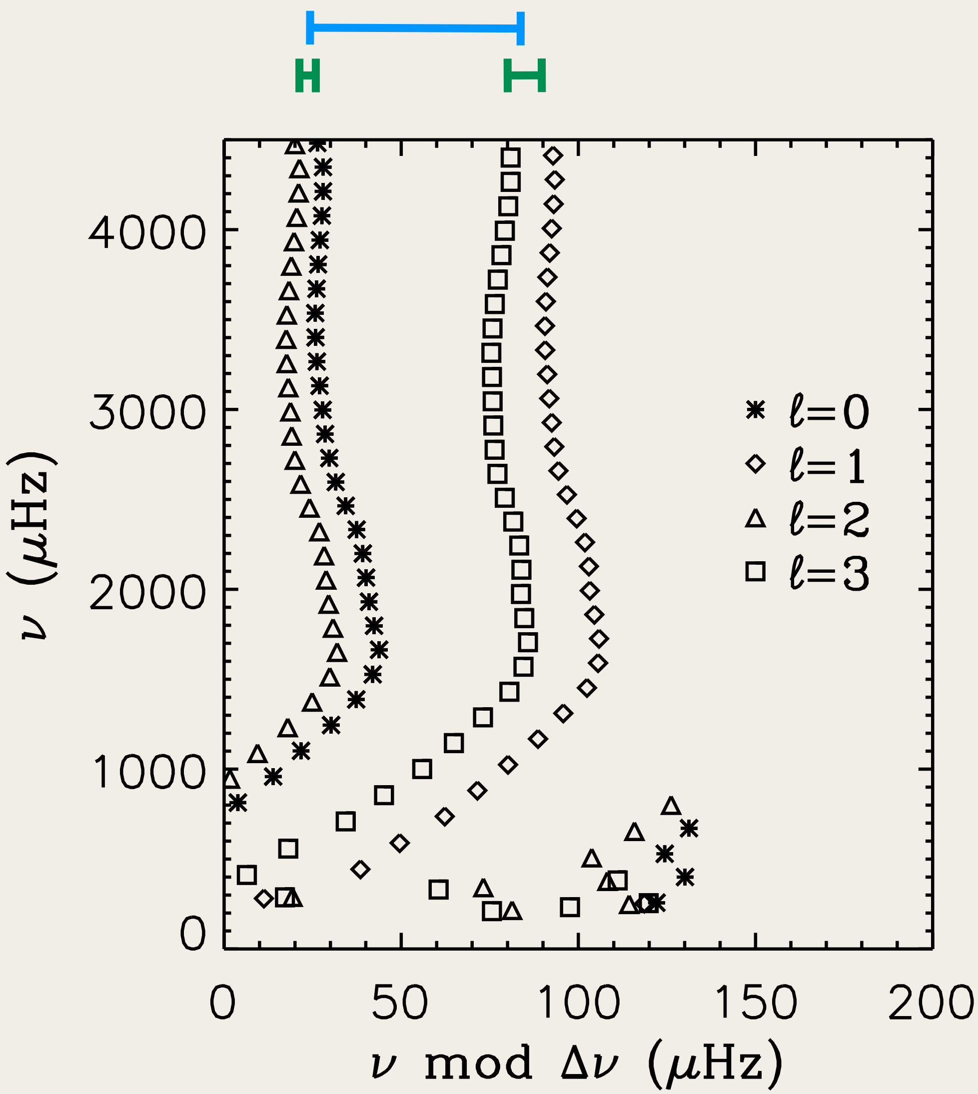
$$\omega_{n,\ell} \approx 2\pi(n+\ell/2+1/4+\alpha)\Delta\nu$$

$$\Delta\nu \equiv \left[2\int_0^R \frac{\mathrm{d}r}{c}\right]^{-1}$$

Frequency Separations

- In absolute terms, formula for $\omega_{n,\ell}$ isn't very accurate
- However, it works well for frequency separations (usually expressed in terms of linear frequency $\nu = \omega/2\pi$)
- Large separation: $\nu_{n+1,\ell} - \nu_{n,\ell} = \Delta\nu$
- Small separation: $\nu_{n,\ell} - \nu_{n-1,\ell+2} = \delta\nu_{n,\ell}$
- Our formula indicates $\delta\nu_{n,\ell} = 0$; more accurately, $\delta\nu_{n,\ell} \approx (4\ell + 6)D_0$
- D_0 depends on the sound-speed gradient in the core: $D_0 \propto \int_0^R \frac{\mathrm{d}c}{\mathrm{d}r}\frac{\mathrm{d}r}{r}$

Frequency Separations in Practice



- How do we measure frequency separations from observations?
  - peaks in power spectrum → $\nu_{?,?}$
  - we don't know n
  - we don't usually know ℓ (Sun is a special case)
- Use an *echelle diagram*
  - Make an informed guess for $\Delta\nu$
  - Plot ν mod $\Delta\nu$ (abscissa) against ν (ordinate) for each observed frequency
  - Adjust $\Delta\nu$ until points are approx. vertical
  - Measure $\Delta\nu$ and $\delta\nu$ from diagram

Lab: Diagrams, Diagrams!