GYRE

Rich Townsend & The GYRE Team

Aug 09, 2023

USER GUIDE

GYRE is a *stellar oscillation code*. Given an input stellar model, GYRE calculates the eigenfrequencies and eigenfunctions for the normal oscillation modes of the model. These data can be put to a variety of uses; the most common is to compare them against observed oscillation frequencies of a star, allowing constraints on the star's fundamental parameters (mass, radius, etc.) to be established — the discipline of *asteroseismology*.

GYRE also supports other, related kinds of calculation. One example is evaluating the response of a star to tidal disturbances produced by an orbiting companion; because this is an instance of *forced* stellar oscillations, similar numerical techniques can be brought to bear.

CHAPTER

ONE

PRELIMINARIES

1.1 Intended Audience

This manual is aimed at a broad audience — whether you're a GYRE novice or a seasoned veteran, it provides you with the information you'll need to get the most out of GYRE. However, it does presume some experience with Unix command-line environments, and likewise some basic familiarity with the subject of stellar oscillations. If you need the former, then the Internet is your oyster; and for the latter, we recommend the following online resources:

- Jørgen Christensen-Dalsgaard's Lecture Notes on Stellar Oscillations;
- Gerald Handler's Asteroseismology article.

1.2 Obtaining GYRE

The source code for GYRE is hosted in the https://github.com/rhdtownsend/gyre git repository on GitHub. GYRE is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.

1.3 Citing GYRE

If you use GYRE in your research, please cite one or more of the relevant 'instrument' papers:

- Townsend & Teitler (2013) describes the basic operation of the code;
- Townsend et al. (2018) outlines capabilities for *non-adiabatic oscillation* calculations;
- Goldstein & Townsend (2020) describes the *contour method* for finding non-adiabatic modes;
- Sun et al. (2023) introduces the *gyre_tides frontend* for evaluating tidal responses.

If you find yourself using GYRE on a regular basis, you might consider contributing to the project to ensure its long-term success. Options include

- contributing code to the project (e.g., via GitHub pull requests), to extend GYRE's capabilities;
- contributing documentation and tutorials to the project, to make GYRE more user-friendly;
- inviting the GYRE team to be co-authors on relevant papers;
- inviting the GYRE team to be co-investigators on relevant grant applications.

1.4 Development Team

GYRE remains under active development by the following team:

- Rich Townsend (University of Wisconsin-Madison); project leader
- Warrick Ball (University of Birmingham)
- Earl Bellinger (MPIA Garching)
- Zhao Guo (Cambridge University)
- Mathias Michielsen (KU-Leuven)
- Joel Ong (University of Hawaii-Manoa)
- Jarret Rosenberg (University of Wisconsin-Madison)
- Meng Sun (Northwestern University)
- Vincent Vanlaer (KU-Leuven)

Former developers include:

- Jacqueline Goldstein (MIT)
- Aaron Lopez

Also, the following people have made valuable contributions toward testing GYRE:

- Siemen Burssens (KU Leuven)
- Timothy Van Reeth (KU Leven)

1.5 Related Links

- The GYRE discussion forums, the place to post feature requests and bug reports (don't send emails!).
- The MESA Software Development Kit (SDK), which provides the compilers and supporting libraries needed to build GYRE.
- The MESA Stellar Evolution Code, which can generate stellar models readable by GYRE.

1.6 Acknowledgments

GYRE has been developed with financial support from the following grants:

- NSF awards AST-0908688, AST-0904607, ACI-1339606, ACI-1663696, and AST-1716436;
- NASA awards NNX14AB55G, NNX16AB97G, and 80NSSC20K0515.

GYRE has also benefited greatly from contributions (code, bug reports, feature requests) from the academic community. Thanks, folks!

CHAPTER

TWO

QUICK START

To get started with GYRE, follow these five simple steps:

- install the MESA Software Development Kit;
- download the GYRE source code;
- unpack the source code using the **`tar`** utility;
- set the `GYRE_DIR` environment variable to point to the newly created source directory;
- compile GYRE using the command **`make -C $GYRE_DIR`**.

For a more in-depth installation guide that covers alternative use-cases, refer to the *Installation* chapter. If the code doesn't compile properly, consult the *Troubleshooting* chapter. Otherwise, proceed to the next chapter where you'll put together your first GYRE calculation.

CHAPTER

THREE

EXAMPLE WALKTHROUGH

This chapter provides a walkthrough of a example GYRE project, to illustrate the typical steps involved. For this example, we'll use **gyre** (the *frontend* focused on stellar oscillations) to find eigenfrequencies and eigenfunctions of dipole and quadrupole gravity modes for a MESA model of slowly pulsating B (SPB) star.

3.1 Making a Place to Work

When starting a new project, it's a good idea to create a dedicated work directory to contain the various input and output files that **gyre** operates on. These commands will make a new directory beneath your home directory with the name `work`, and then set this directory as the current working directory:

```
$ mkdir ~/work
$ cd ~/work
```

3.2 Grabbing a Stellar Model

The next step is to grab the stellar model. There are a number of example models provided in the `$GYRE_DIR/models` directory; the following commands will copy a MESA model for a $5\,\mathrm{M}_\odot$ SPB star into your work directory:

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

3.3 Assembling a Namelist File

Now comes the fun part: assembling an input file containing the various parameters which control a **gyre** run. Using a text editor, create the file `gyre.in` in your work directory with the following content cut-and-pasted in:

```
&constants
/

&model
  model_type = 'EVOL'  ! Obtain stellar structure from an evolutionary model
  file = 'spb.mesa'    ! File name of the evolutionary model
  file_format = 'MESA' ! File format of the evolutionary model
/

&mode
  l = 1 ! Harmonic degree
```

(continues on next page)

(continued from previous page)

```
/

&mode
  l = 2 ! Harmonic degree
/

&osc
  outer_bound = 'VACUUM' ! Assume the density vanishes at the stellar surface
/

&rot
/

&num
  diff_scheme = 'COLLOC_GL4' ! 4th-order collocation scheme for difference equations
/

&scan
  grid_type = 'INVERSE' ! Scan grid uniform in inverse frequency
  freq_min = 0.5        ! Minimum frequency to scan from
  freq_max = 1.5        ! Maximum frequency to scan to
  n_freq = 100          ! Number of frequency points in scan
/

&grid
  w_osc = 10 ! Oscillatory region weight parameter
  w_exp = 2  ! Exponential region weight parameter
  w_ctr = 10 ! Central region weight parameter
/


&ad_output
  summary_file = 'summary.h5'                          ! File name for summary file
  summary_item_list = 'l,n_pg,freq,freq_units,E_norm' ! Items to appear in summary file
  detail_template = 'detail.l%l.n%n.h5'                      ! File name template for␣
↪detail files
  detail_item_list = 'l,n_pg,omega,x,xi_r,
                      xi_h,c_1,As,V_2,Gamma_1'               ! Items to appear in detail␣
↪files
  freq_units = 'CYC_PER_DAY'                                 ! Units of freq output items
/

&nad_output
/
```

This file is in namelist format, containing multiple namelist groups. Detailed information on the groups can be found in the *Namelist Input Files* chapter; for now, let's just focus on some of the more-important aspects of the file above:

- the `&constants` namelist group is empty, telling **gyre** to use default values for fundamental constants;
- the `&model` namelist group tells **gyre** to read an evolutionary model, in *MESA format*, from the file `spb.mesa`;
- the two `&mode` namelist groups tells **gyre** to search first for dipole ($\ell = 1$) and then quadrupole ($\ell = 2$) modes;
- the `&osc` namelist group tells **gyre** to assume, when setting up the outer boundary conditions in the oscillation

equations, that the density vanishes at the stellar surface;

- the `&scan` namelist group tells **gyre** to scan a region of dimensionless angular frequency space typically occupied by gravity modes;
- the `&grid` namelist group tells **gyre** how to refine the model spatial grid;
- the `&ad_output` namelist group tells **gyre** what adiabatic data to write to which output files; summary data to the file `summary.h5`, and individual mode data to files having the prefix `mode.`;
- the `&nad_output` namelist group is empty, telling **gyre** not to write any non-adiabatic data.

3.4 Running gyre

With the hard work done, it's now trivial to run **gyre**:

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

As the frontend runs (on multiple cores, if you have a multi-core machine; see the *FAQ* for more details), it will print lots of data to the screen. Let's break down this output, chunk by chunk.

First, **gyre** prints out its version number, tells us (in OpenMP threads) how many cores it is running on, and indicates which file it is reading parameters from (here, `gyre.in`):

```
gyre [master]
-------------

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

Next, **gyre** loads the stellar model from the file `spb.mesa`. This model comprises 1814 points and extends from the surface all the way to the center (which is why **gyre** decides not to add a central point).

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

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

gyre then prepares to search for modes with harmonic degree $\ell = 1$ and azimuthal order $m = 0$ (not specified in `gyre.in`, but assumed by default), by building a frequency grid and a spatial grid:

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

Mode parameters
   l : 1
   m : 0

Building frequency grid (REAL axis)
   added scan interval :  0.5000E+00 ->  0.1500E+01 (100 points, INVERSE)

Building spatial grid
```

(continues on next page)

(continued from previous page)

```
   Scaffold grid from model
   Refined 0 subinterval(s) in iteration 1
   Final grid has 1 segment(s) and 1814 point(s):
      Segment 1 : x range 0.0000 -> 1.0000 (1 -> 1814)
```

(The concepts of spatial and frequency grids are explored in greater detail in the *Numerical Methods* and *Understanding Grids* chapters). Next, **gyre** attempts to bracket roots of the discriminant function (again, see the *Numerical Methods* chapter) by searching for changes in its sign:

```
Starting search (adiabatic)

Evaluating discriminant
  Time elapsed :     0.886 s
```

Finally, for each bracket found **gyre** uses a root solver to converge to the eigenfrequency. Each row of output here corresponds to a mode that **gyre** has successfully found:

```
Root Solving
   l    m    n_pg    n_p    n_g       Re(omega)       Im(omega)        chi n_iter
   1    0      -9      0      9  0.50907836E+00  0.00000000E+00 0.6678E-14      7
   1    0      -8      0      8  0.58398491E+00  0.00000000E+00 0.1352E-13      6
   1    0      -7      0      7  0.66078111E+00  0.00000000E+00 0.1666E-13      7
   1    0      -6      0      6  0.73734087E+00  0.00000000E+00 0.3141E-13      6
   1    0      -5      0      5  0.89820448E+00  0.00000000E+00 0.1363E-13      6
   1    0      -4      0      4  0.11322842E+01  0.00000000E+00 0.6270E-13      7
   1    0      -3      0      3  0.13377876E+01  0.00000000E+00 0.6789E-13      6
  Time elapsed :      0.382 s
```

The columns appearing are as follows:

l
: harmonic degree ℓ

m
: azimuthal order m

n_pg
: radial order n (in the Eckart-Osaki-Scuflaire-Takata scheme)

n_p
: acoustic-wave winding number n_{p}

n_g
: gravity-wave winding number n_{g}

Re(omega)
: real part of dimensionless eigenfrequency ω

Im(omega)
: imaginary part of dimensionless eigenfrequency ω (zero here because we've performed an adiabatic calculation)

chi
: convergence parameter

n_iter
: number of iterations required for convergence

These values are printed to screen primarily to give an idea of **gyre**'s progress. Some things to watch out for:

- The convergence parameter `chi`, defined as the ratio of discriminant values before and after the root finding, should small (on the order of 1E-9 to 1E-15). If it is significantly larger than this, the mode may not be properly converged; and if it is significantly smaller than this, there may be numerical issues with the discretization scheme.
- The number of iterations `n_iter` should be moderate; values above 20 or so indicate that **gyre** is having problems converging.
- The mode radial order `n_pg` should be monotonic-increasing. Departures from this behavior can happen for a number of reasons, that are discussed in the *Troubleshooting* chapter.

After processing the dipole modes, **gyre** repeats the search steps for the quadrupole modes. Once the overall run is complete, a number of output files are written:

- A summary file with the name `summary.h5`
- For each mode found, a detail file with the name `detail.l`*L*`.n`*N*`.h5`, where *L* and *N* are the harmonic degree and radial order of the mode, respectively.

The *Output Files* chapter discusses how to read and analyze these files.

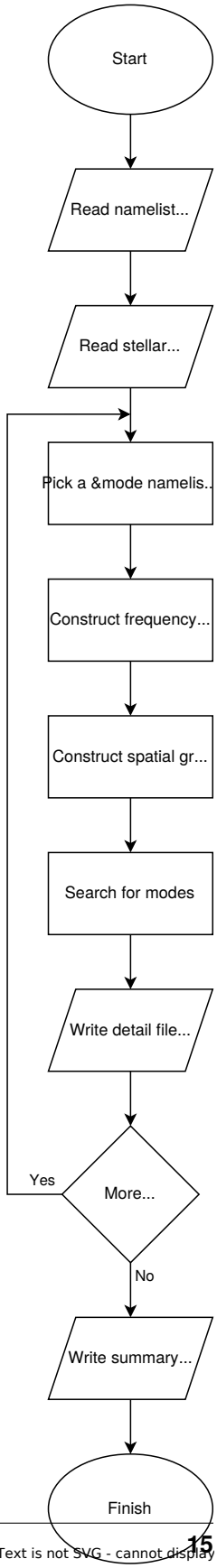
CHAPTER

FOUR

FRONTENDS

This chapter summarizes the frontends provided by GYRE — the executables programs that users run to perform calculations. Although the *Example Walkthrough* chapter focuses on the **`gyre`** frontend, there are others available focused on different kinds of task.

4.1 gyre


Start
Read namelist...
Read stellar...
Pick a &mode namelis...
Construct frequency...
Construct spatial gr...
Search for modes
Write detail file...
More...
Yes
No
Write summary...
Finish
Text is not SVG - cannot display

The **gyre** frontend calculates the free-oscillation modes of a stellar model. The general flow of execution is outlined in the chart to the right. After reading the *namelist input file* and the *model*, **gyre** loops over `&mode` namelist groups, processing each in turn.

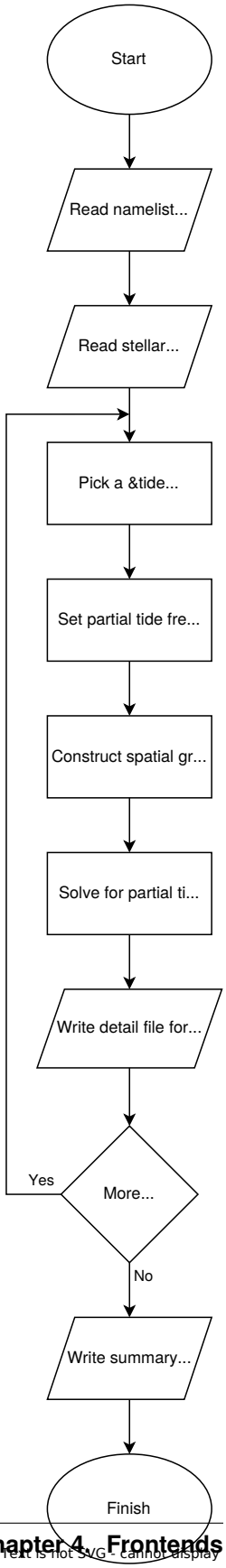
For a given group, **gyre** searches over a range of oscillation frequencies for modes with a specific harmonic degree ℓ and azimuthal order m. With each mode found, the eigenfrequency, eigenfunctions and other data are optionally written to a *detail file*. At the end of the run, response data from all modes found (across all `&mode` groups) are optionally written to a *summary file*.

The table below lists which namelist groups, and in what number, should appear in namelist input files for **gyre**.

Description	Namelist group name	Count
Constants	`&constants`	1
Grid Parameters	`&grid`	≥ 1[1]
Mode Parameters	`&mode`	≥ 1
Stellar Model Parameters	`&model`	1
Numerical Parameters	`&num`	≥ 1[1]
Oscillation Parameters	`&osc`	≥ 1[1]
Output Parameters	`&ad_output`	1
	`&nad_output`	1
Rotation Parameters	`&rot`	≥ 1[1]
Frequency Scan Parameters	`&scan`	≥ 1

[1] While the input file can contain one or more of the indicated namelist group, only the last (*tag-matching*) one is used.

4.2 gyre_tides



The **gyre_tides** frontend calculates the response of a stellar model to tidal forcing by a orbiting point-mass companion. The general flow of execution is outlined in the chart to the right. After reading the *namelist input file* and the *model*, **gyre_tides** loops over `&tide` namelist groups, processing each in turn.

For a given group, **gyre_tides** solves for the response of the star to the superposition of partial tidal potentials $\Phi_{\mathrm{T};\ell,m,k}$ (see the *Tidal Effects* section). The response wavefunctions and other data associated with an individual partial potential are optionally written to a *detail file*. At the end of the run, response data from all partial potentials (across all `&tide` groups) are optionally written to a *summary file*.

The table below lists which namelist groups, and in what number, should appear in namelist input files for **gyre_tides**.

Description	Namelist group name	Number
Constants	`&constants`	1
Grid Parameters	`&grid`	≥ 1[1]
Stellar Model Parameters	`&model`	1
Numerical Parameters	`&num`	≥ 1[Page 19, 1]
Orbital Parameters	`&orbit`	≥ 1[1]
Oscillation Parameters	`&osc`	≥ 1[1]
Output Parameters	`&tides_output`	1
Rotation Parameters	`&rot`	≥ 1[1]
Tidal Parameters	`&tide`	≥ 1

[1] While the input file can contain one or more of the indicated namelist group, only the last (*tag-matching*) one is used.

CHAPTER

FIVE

NUMERICAL METHODS

This chapter explains the numerical methods used by the **gyre** frontend to solve the oscillation equations (similar approaches are followed in other *frontends*). Although it aims to be user-friendly, **gyre** is nevertheless a complex piece of software; thus, getting it to produce the 'best' results requires some degree of insight into the algorithms it uses to calculate mode eigenfrequencies and eigenfunctions.

5.1 The Stretched String Problem

We'll start our discussion of numerical methods by considering the problem of finding normal-mode eigenfrequencies and eigenfunctions for waves on a stretched string clamped at both ends. Let the string have mass per unit length ρ and tension T; then, the wave equation describing the transverse string displacement $y(x,t)$ at spatial position x and time t is

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

with $c \equiv (T/\rho)^{1/2}$. If the string is clamped at $x = 0$ and $x = L$, then the wave equation together with the boundary conditions

$$y(0,t) = 0 \qquad y(L,t) = 0$$

comprises a two-point boundary value problem (BVP).

5.1.1 Analytic Solution

The stretched-string BVP is straightforward to solve analytically. General solutions of the wave equation take the form of traveling waves,

$$y(x,t) = A\exp[\mathrm{i}(kx - \sigma t)],$$

where A an arbitrary constant, and the frequency σ and wavenumber k are linked by the dispersion relation

$$\sigma^2 = c^2 k^2.$$

The phase velocity of these waves is $\sigma/k = \pm c$.

To satisfy the boundary condition at $x = 0$, we combine traveling-wave solutions with opposite-sign wavenumbers

$$y(x,t) = A\exp[\mathrm{i}(kx - \sigma t)] - A\exp[\mathrm{i}(-kx - \sigma t)] = B\sin(kx)\exp(-\mathrm{i}\sigma t),$$

where $B = 2A$. For the boundary condition at $x = L$ to be satisfied simultaneously,

$$\sin(kL) = 0,$$

and so

$$kL = n\pi$$

where n is a non-zero integer (we exclude $n = 0$ because it corresponds to the trivial solution $y(x,t) = 0$). Combining this with the dispersion relation, we find that the normal-mode eigenfrequencies of the stretched-string BVP are

$$\sigma = n\frac{\pi c}{L}, \tag{5.1}$$

and the corresponding eigenfunctions are

$$y_n(x,t) = B\sin\left(\frac{n\pi x}{L}\right)\exp(-\mathrm{i}\sigma t). \tag{5.2}$$

The index n uniquely labels the modes, and $y_n(x,t)$ exhibits $n-1$ nodes in the open interval $x \in (0, L)$.

5.1.2 Separation

Now let's see how we might go about solving the stretched-string BVP numerically. We begin by performing a separation of variables on the wave equation, assuming trial solutions of the form

$$y(x;t) = \tilde{y}(x)\,\exp(-\mathrm{i}\sigma t), \tag{5.3}$$

where $\tilde{y}(x)$ is a function of x alone. Then, the wave equation reduces to an ordinary differential equation (ODE) for $\tilde{y}$,

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

5.1.3 Discretization

To solve the ODE, we discretize it to establish a set of difference equations. The discretization involves transforming the continuous function $\tilde{y}(x)$ into a finite set of N values $\{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_N\}$, representing the function sampled on the discrete spatial grid $\{x_1, x_2, \ldots, x_N\}$.

For simplicity let's assume the grid is uniform, so that

$$x_{j+1} - x_j \equiv \Delta x = \frac{L}{N-1} \qquad (1 \le j \le N-1).$$

Then, the second derivative of $\tilde{y}$ can be approximated to second order in Δx as

$$\left.\frac{\mathrm{d}^2\tilde{y}}{\mathrm{d}x^2}\right|_{x=x_j} \approx \frac{\tilde{y}_{j+1} - 2\tilde{y}_j + \tilde{y}_{j-1}}{\Delta x^2} \qquad (2 \le j \le N-1).$$

This allows us to replace the ODE with $N-2$ difference equations

$$\frac{\tilde{y}_{j+1} - 2\tilde{y}_j + \tilde{y}_{j-1}}{\Delta x^2} = -\frac{\sigma^2}{c^2}\tilde{y}_j \qquad (2 \le j \le N-1).$$

Together with the two boundary conditions

$$\tilde{y}_1 = 0 \qquad \tilde{y}_N = 0,$$

we thus have a linear system of N algebraic equations and N unknowns.

5.1.4 Linear System

To find solutions to the linear system, we first write it in matrix form as

$$\mathbf{S}\mathbf{u} = \mathbf{0}, \tag{5.4}$$

where $\mathbf{u}$ is the vector with components

$$\mathbf{u} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N-1} \\ \tilde{y}_N \end{pmatrix}$$

and the 'system matrix' $\mathbf{S}$ is an $N \times N$ tridiagonal matrix with components

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & \sigma^2\tau^2 - 2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \sigma^2\tau^2 - 2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

Here we've introduced

$$\tau \equiv \frac{\Delta x}{c}$$

as the sound crossing time of a single cell.

Equation (5.4) is a homogeneous linear system, meaning that it has non-trivial solutions $\mathbf{u}$ only when the determinant of $\mathbf{S}$ vanishes. With this in mind, we formulate the characteristic equation for the BVP,

$$\mathcal{D}(\sigma) = 0$$

where $\mathcal{D}(\sigma) \equiv \det(\mathbf{S})$ is a discriminant function whose roots are the characteristic frequencies (*eigenfrequencies*) of the stretched-string BVP.

Fig. 5.1 plots the discriminant function for the BVP discretized on a spatial grid of $N = 50$ points. The roots (zeros) of the function are highlighted by the orange markers; they fall very close to the values $\sigma = \pi c/L, 2\pi c/L, \ldots$ predicted by the analytic solutions.

5.1.5 Scanning for Eigenfrequencies

While Fig. 5.1 is useful for visualizing $\mathcal{D}$, it's not the best way to find eigenfrequencies. Instead, we can rely on well-established techniques for isolating and refining roots of monovariate functions.

First, we evaluate a set of M values $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$, representing the discriminant function sampled on the discrete frequency grid $\{\sigma_1, \sigma_2, \ldots, \sigma_M\}$. Then, we scan through these data looking for sign changes between adjacent discriminant values. If $\mathcal{D}_i\mathcal{D}_{i+1} < 0$, we know that a root of the discriminant function must lie in the interval (σ_i, σ_{i+1}) — we have *bracketed* a root. Fig. 5.2 illustrates the process of bracket scanning for a frequency grid comprising $M = 32$ points, distributed uniformly in σ across the same range as plotted in Fig. 5.1. This figure highlights five brackets containing the five roots identified previously.

Once a bracket is established for a given root, it can be narrowed through a process of iterative refinement until the root is converged upon. There are a variety of well-known root-finding algorithms that perform this refinement; the bisection method is conceptually the simplest, but approaches such as Brent's method can be much more efficient. For

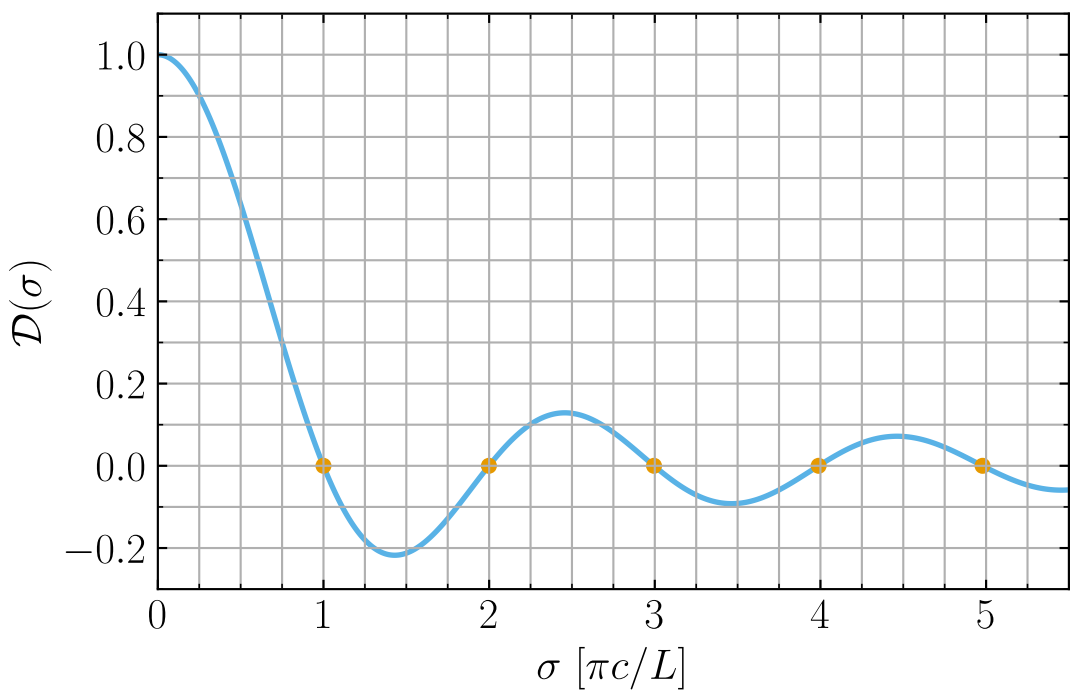


Fig. 5.1: Plot of the discriminant function $\mathcal{D}(\sigma)$ as a function of the frequency σ, for the stretched-string BVP with $N = 50$. The orange dots highlight where $\mathcal{D} = 0$. The function has been scaled so that $\mathcal{D}(0) = 1$. (`Source`)

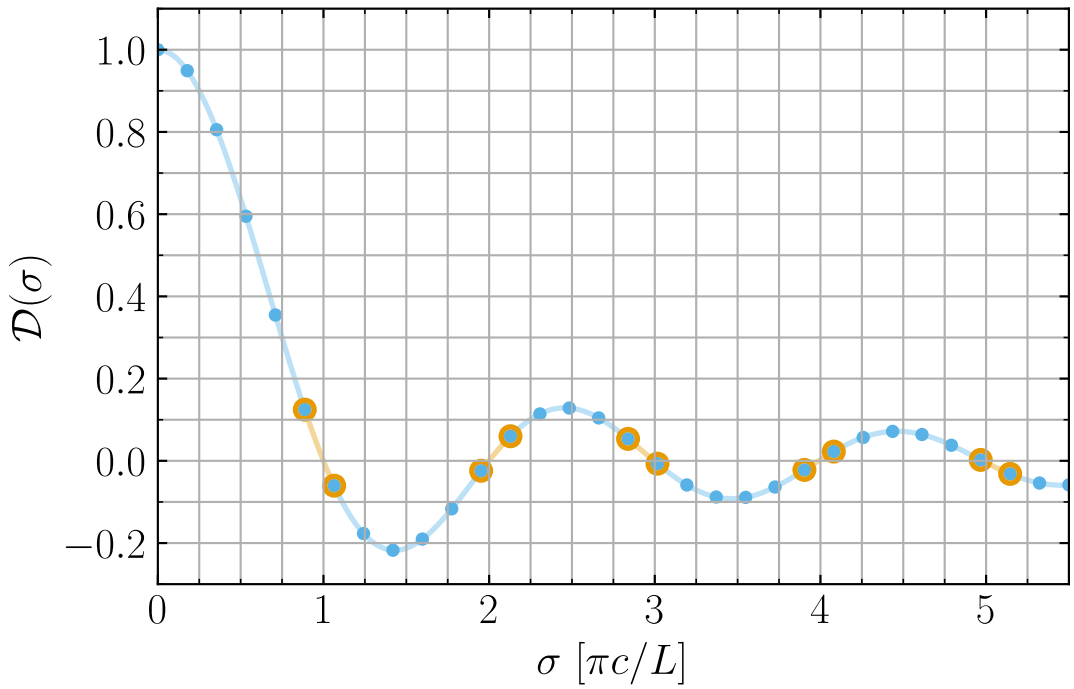


Fig. 5.2: Plot of the discriminant values $\{\mathcal{D}\}$ on the discrete frequency grid $\{\sigma\}$ (distributed uniformly in σ), for the stretched-string BVP with $N = 50$ and $M = 32$. The orange-haloed segments highlight adjacent points that bracket a root $\mathcal{D} = 0$. (`Source`)

the brackets plotted in Fig. 5.2, Table 5.1 compares the eigenfrequencies found using Python's `scipy.optimize.brentq()` function, against the analytic values predicted by eqn. (5.1).

Table 5.1: Numerical and analytic eigenfrequencies, in units of $\pi c/L$, for the stretched-string BVP with $N = 50$. (`Source`)

n	numerical	analytic
1	0.999829	1.000000
2	1.998630	2.000000
3	2.995378	3.000000
4	3.989047	4.000000
5	4.978618	5.000000

5.1.6 Eigenfunction Reconstruction

For each of the eigenfrequencies found, we reconstruct the corresponding eigenfunction by solving the linear system (5.4). Because $\det(\mathbf{S})$ is now zero, this system is guaranteed to have a non-trivial solution. The solution vector $\mathbf{u}$ resides in the null space of $\mathbf{S}$, and we can use standard numerical techniques (e.g., singular value decomposition) to evaluate it. Then, the j'th element of $\mathbf{u}$ corresponds to the eigenfunction sampled at the j'th spatial grid point:

$$(\mathbf{u})_j = \tilde{y}_j \equiv \tilde{y}(x_j)$$

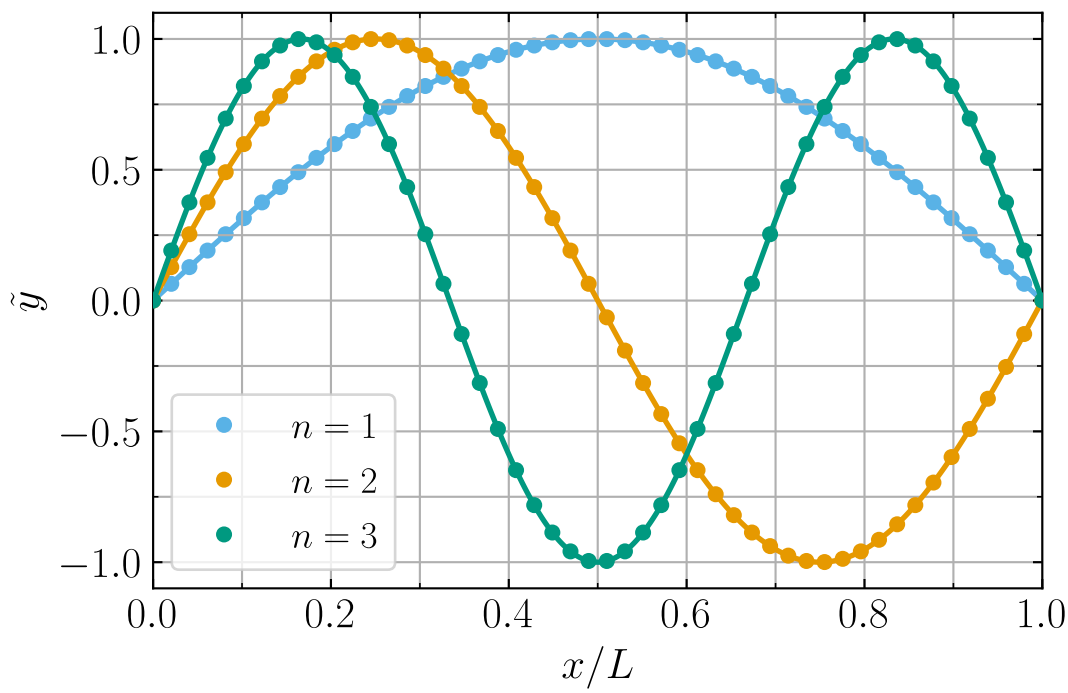


Fig. 5.3: Plot of the eigenfunctions $\tilde{y}$ as a function of spatial coordinate x, for the first three modes of the stretched-string BVP with $N = 50$. The discrete points show the numerical functions, and the solid lines the corresponding analytic functions. In all cases, the eigenfunctions have been normalized to have a maximum $|\tilde{y}|$ of unity. (`Source`)

Fig. 5.3 plots the eigenfunctions found in this way for the first three modes ($n = 1, 2, 3$) given in Table 5.1. Also shown are the corresponding analytic solutions given by eqn. (5.2). The agreement between the two is good.

5.2 From Stretched String to gyre

The numerical technique demonstrated in the *The Stretched String Problem* section provides a powerful analog to how **gyre** solves the oscillation equations. The full details of **gyre**'s approach are laid out in Townsend & Teitler (2013); in this section we briefly summarize it, highlighting similarities and differences with the stretched-string problem.

5.2.1 Separation

Similar to the stretched-string problem, **gyre** begins by separating variables in space and time. For the radial displacement perturbation ξ_r, trial solutions take the form

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

(this is taken from the *Separated Equations* section). In addition to the same sinusoidal time dependence as in eqn. (5.3), a spherical harmonic term Y_ℓ^m appears because we are separating in three (spherical) spatial coordinates rather than one.

5.2.2 Discretization

As with the stretched-string problem, **gyre** discretizes the ODE governing $\tilde{\xi}_r(r)$ and related quantities on a spatial grid $\{x_1, x_2, \ldots, x_N\}$. However, a couple of important differences arise at this juncture. First, the oscillation equations are fourth order (sixth, in the non-adiabatic case). Rather than employing finite-difference approximations to high-order differential operators, **gyre** instead decomposes the problem into a system of coupled first-order equations. This system is written generically as

$$x\frac{\mathrm{d}\tilde{\mathbf{y}}}{\mathrm{d}x} = \mathbf{A}\,\tilde{\mathbf{y}},$$

where $\tilde{\mathbf{y}}$ is a vector of n_e dependent variables, and $\mathbf{A}$ is a $n_\mathrm{e} \times n_\mathrm{e}$ Jacobian matrix. In the adiabatic case, $n_\mathrm{e} = 4$; in the non-adiabatic case, $n_\mathrm{e} = 6$.

Second, while the above equation system can be discretized using a simple finite-difference approximation to the left-hand side, **gyre** offers more-sophisticated approaches with higher orders of accuracy. These include the Magnus schemes described in Townsend & Teitler (2013), and implicit Runge-Kutta schemes mentioned in Townsend et al. (2018). The choice of scheme is set by the `diff_scheme` parameter of the `&num` namelist group. The discretization leads to difference equations of the form

$$\tilde{\mathbf{y}}_{j+1} = \mathbf{Y}_{j+1;j}\,\tilde{\mathbf{y}}_j,$$

relating the dependent variable vector at adjacent grid points. The $n_\mathrm{e} \times n_\mathrm{e}$ fundamental solution matrix $\mathbf{Y}_{j+1,j}$ is evaluated from the value(s) of $\mathbf{A}$ within the interval $[x_j, x_{j+1}]$ using the discretization scheme.

There are $N - 1$ of these sets of difference equations. They are augmented with the boundary conditions

$$\mathbf{B}_\mathrm{i}\,\tilde{\mathbf{y}}_1 = 0, \qquad \mathbf{B}_\mathrm{o}\,\tilde{\mathbf{y}}_N = 0,$$

where $\mathbf{B}_\mathrm{i}$ is a $n_\mathrm{i} \times n_\mathrm{e}$ matrix representing the n_i inner boundary conditions, and $\mathbf{B}_\mathrm{o}$ is a $n_\mathrm{o} \times n_\mathrm{e}$ matrix representing the outer boundary conditions (note that $n_\mathrm{i} + n_\mathrm{o} = n_\mathrm{e}$). Together, the difference equations and boundary conditions comprise a linear system of $n_\mathrm{e}\,N$ algebraic equations and $n_\mathrm{e}N$ unknowns.

5.2.3 Linear System

The linear system can be written in the same form (cf. eqn. 5.4) as with the stretched-string problem . However, now $\mathbf{u}$ is the vector with components

$$\mathbf{u} = \begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \vdots \\ \tilde{\mathbf{y}}_{N-1} \\ \tilde{\mathbf{y}}_N \end{pmatrix}$$

and the system matrix S is an $\neq N \times n_{\rm e} N$ block-staircase matrix with components

$$\mathsf{S} = \begin{pmatrix} \mathsf{B}_{\rm i} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\mathsf{Y}_{2;1} & \mathsf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathsf{Y}_{N;N-1} & \mathsf{I} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathsf{B}_{\rm o} \end{pmatrix}.$$

As before, the linear system (5.4) has non-trivial solutions only when the determinant of S vanishes. Thus, **gyre** finds eigenvalues of the oscillation equation by solving the characteristic equation

$$\mathcal{D}(\omega) \equiv \det(\mathsf{S}) = 0,$$

where the dimensionless frequency

$$\omega \equiv \sqrt{\frac{R^3}{GM}}\,\sigma,$$

is the product of the star's dynamical timescale and the oscillation frequency σ. (Internally, **gyre** works extensively with such *dimensionless quantities*, as it improves the stability of the numerical algorithms).

5.2.4 Scanning for Eigenfrequencies

In the adiabatic case, **gyre** searches for roots of the discriminant function $\mathcal{D}$ using the same bracketing and refinement strategies as the stretched-string problem.

In the non-adiabatic case, a complication is that the discriminant function and the dimensionless frequency are both complex quantities. Solving the characteristic equation in the complex plane is computationally challenging because there is no equivalent to bracketing and refinement. **gyre** implements a couple of different approaches to the problem, as discussed in the *Non-Adiabatic Oscillations* section.

5.3 Limitations of the Numerical Method

The numerical method used both in the stretched string problem and in GYRE generally performs very well; however, it has a couple of failure scenarios that are important to understand. These scenarios arise through poor choices of the spatial grid used to discretize the governing differential equation(s), and/or the frequency grid used to search for roots of the discriminant function.

5.3.1 Insufficient Spatial Resolution

The cost of evaluating the determinant of the system matrix **S** scales proportionally to the number of grid points N used for the discretization. Therefore, in the interests of computational efficiency, we want to make N as small as possible.

However, things go wrong when N becomes too small. Fig. 5.4 demonstrates this by plotting the discriminant function for the stretched-string BVP with $N = 7$. Compared against Fig. 5.1, we see that toward larger σ the roots of the discriminant function become progressively shifted toward lower frequencies; and, above $\sigma \approx 3.5\pi c/L$, they disappear altogether.

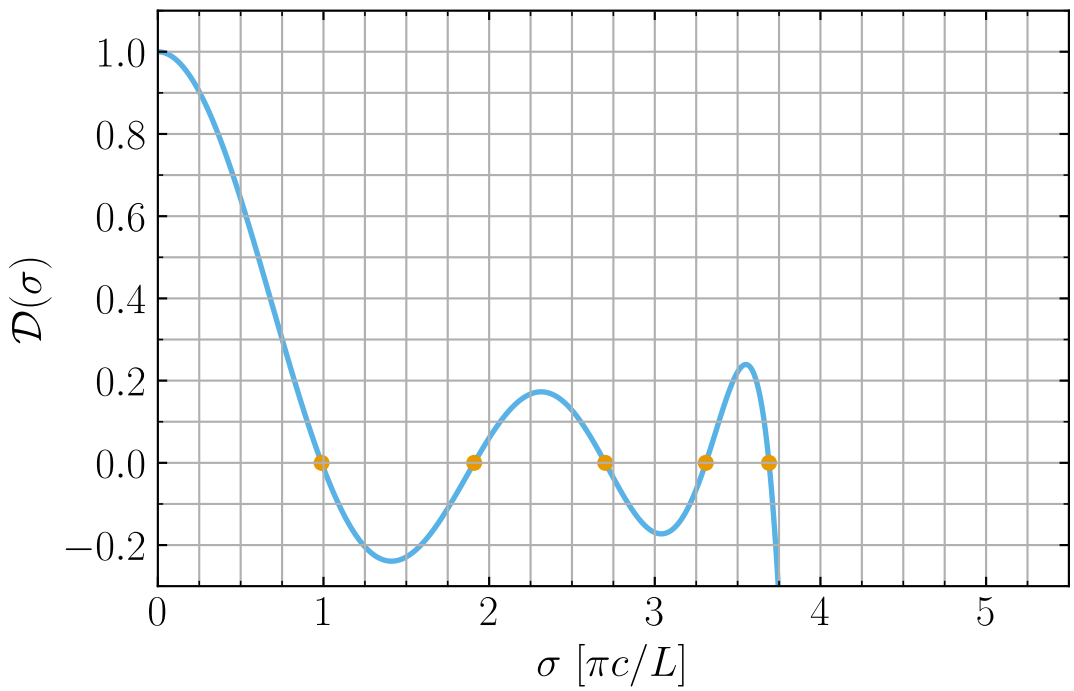


Fig. 5.4: Plot of the discriminant function $\mathcal{D}(\sigma)$ as a function of the frequency σ, for the stretched-string BVP with $N = 7$. The orange dots highlight where $\mathcal{D} = 0$. The function has been scaled so that $\mathcal{D}(0) = 1$. (`Source`)

To understand this behavior, recall that the determinant of an $N \times N$ matrix can be expressed (via Laplace expansion) as the sum of N terms; and each term itself involves the product of N matrix elements, picked so that each row/column is used only once in the construction of the term. With these points in mind, we can see from the definition (5.4) of **S** that its determinant (i.e., the discriminant function) must be a polynomial in σ^2 of order $N - 2$; and as such, it can have at most $N - 2$ (in this case, 5) roots. This leads us to important lesson #1:

> **Attention:** The number of points adopted in the discretization limits the number of modes that can be found. With a spatial grid of N points, there are only $\sim N$ distinct numerical solutions.

Returning to Fig. 5.4, the shift in eigenfrequency for the modes that *are* found occurs due to inadequate resolution of the eigenfunctions. We can see this in Fig. 5.5, which reprises Fig. 5.3 for $N = 7$. Clearly, the spatial oscillations of the modes are poorly resolved; the $n = 3$ mode, for instance, is sampled with only one point per quarter wavelength. It's little wonder that the corresponding eigenfrequencies are off. This brings us to important lesson #2 (closely related to #1):

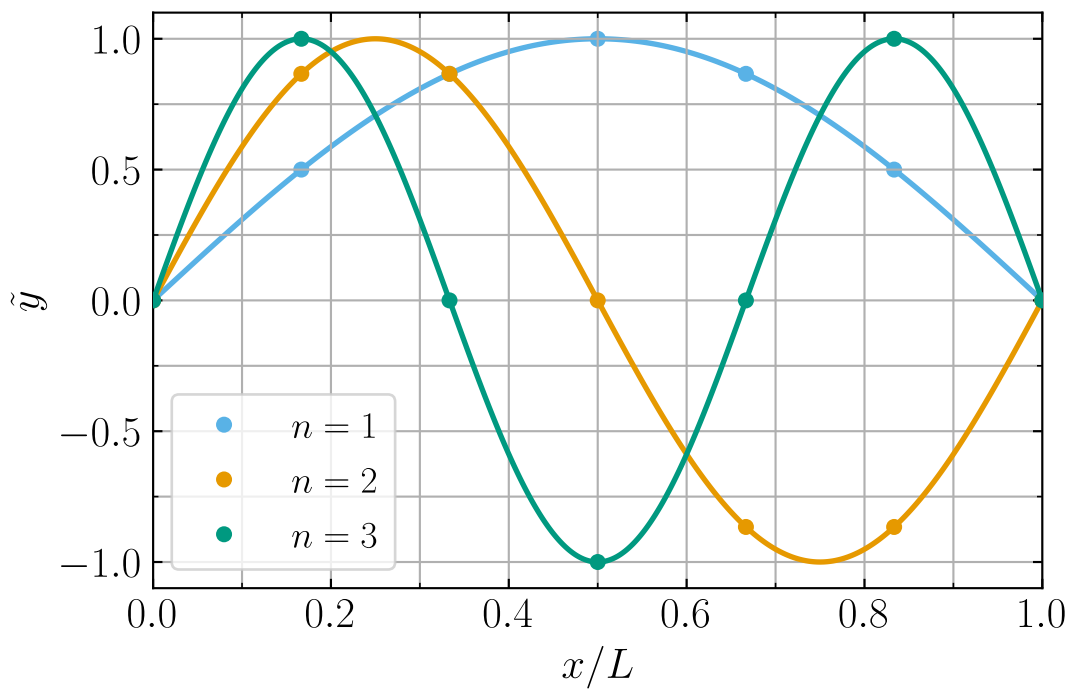


Fig. 5.5: Plot of the eigenfunctions $\tilde{y}$ as a function of spatial coordinate x, for the first three modes of the stretched-string BVP with $N = 7$. The discrete points show the numerical functions, and the solid lines the corresponding analytic functions. (`Source`)

Attention: The spatial resolution adopted in the discretization determines the accuracy of the modes found. A given eigenfrequency will be accurate only when the grid spacing is appreciably smaller than the spatial variation scale of its corresponding eigenfunction.

5.3.2 Insufficient Frequency Resolution

When searching for root brackets, we have to evaluate the discriminant function a total of M times. Therefore, as with the spatial grid, computational efficiency dictates that we want to make M as small as possible. Again, however, things go wrong if M is too small. Fig. 5.6 reprises Fig. 5.2, but adopting a much coarser frequency grid with only $M = 5$ points.

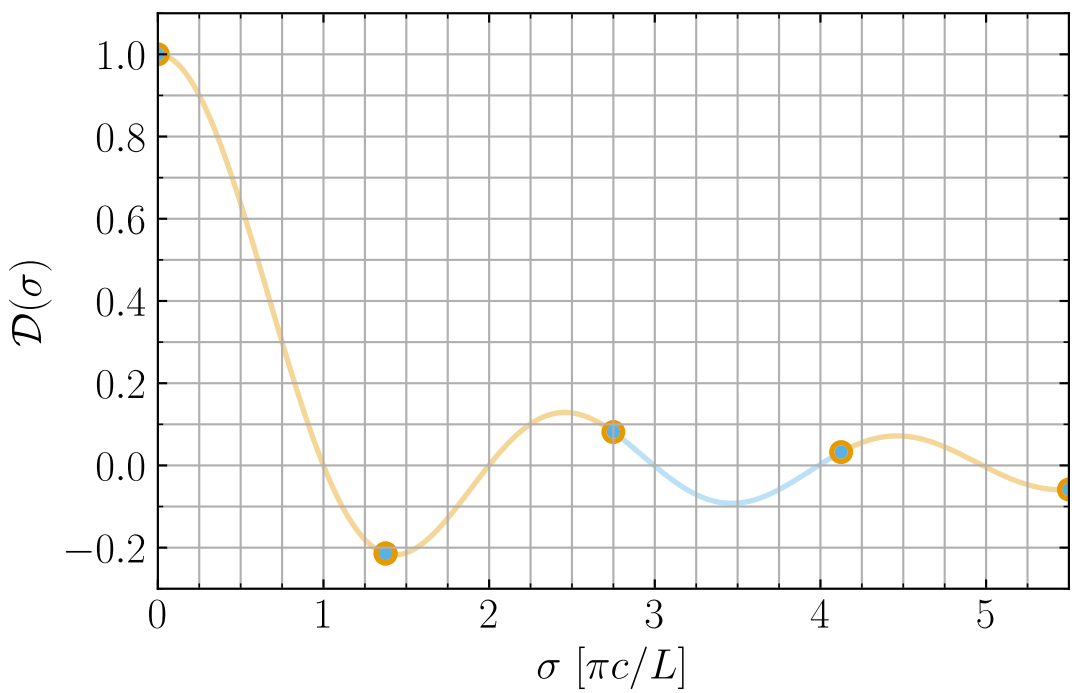


Fig. 5.6: Plot of the discriminant values $\{\mathcal{D}\}$ on the discrete frequency grid $\{\sigma\}$ (distributed uniformly in σ), for the stretched-string BVP with $N = 50$ and $M = 5$. The orange-haloed segments highlight adjacent points that bracket a root $\mathcal{D} = 0$. (`Source`)

Clearly, a pair of adjacent roots (corresponding to the $n = 3$ and $n = 4$ modes) is missed in the bracketing process, as a direct result of the too-coarse grid.

Even when many points are included in the frequency grid, issues can still arise when the distribution of points doesn't match the distribution of roots. An example of this is provided in Fig. 5.7, which reprises Fig. 5.2 with the same number $M = 32$ of points in the grid, but now distributed uniformly in σ^{-1}.

Now it's the roots corresponding to the $n = 4$ and $n = 5$ mode pair that are missed. As with the case in Fig. 5.6, the failure ultimately arises because the spacing between adjacent frequency grid points is (in at least some parts of the grid) larger than the spacing between adjacent roots. This can be summarized in important lesson #3:

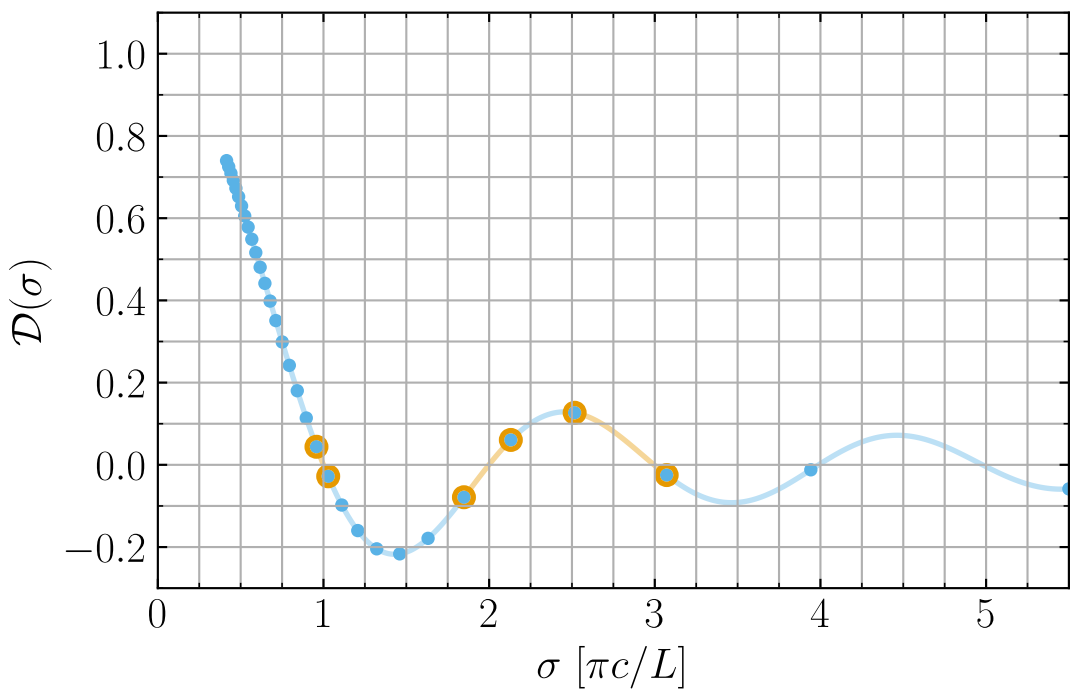


Fig. 5.7: Plot of the discriminant values $\{\mathcal{D}\}$ on the discrete frequency grid $\{\sigma\}$ (distributed uniformly in σ^{-1}), for the stretched-string BVP with $N = 50$ and $M = 32$. The orange-haloed segments highlight adjacent points that bracket a root $\mathcal{D} = 0$. (`Source`)

Attention: The frequency resolution adopted in the root bracketing influences the completeness of the modes found. All modes will be found only when the grid spacing is smaller than the eigenfrequency separation of adjacent modes, across the full range of the grid.

CHAPTER

SIX

INTERPRETING OUTPUT FILES

This chapter demonstrates using Python to read and plot the summary and detail output files written by the GYRE *frontends*. Further information about these files is provided in the *Output Files* chapter.

6.1 PyGYRE

PyGYRE is a Python package maintained separately from GYRE, that provides a set of routines that greatly simplify the analysis of summary and detail files. Detailed information about PyGYRE can be found in the full documentation; here, we demonstrate how to use it to read and plot the output files from the *Example Walkthrough* section.

As a preliminary step, you'll need to install PyGYRE from the Python Package Index (PyPI). This can be done using the **pip** command,

```
$ pip install pygyre
```

If PyGYRE is already installed, you can upgrade to a more-recent version via

```
$ pip install --upgrade pygyre
```

6.2 Analyzing a Summary File

To analyze the summary file written by **gyre** during the *example walkthrough*, change into your *work directory* and fire up your preferred interactive Python environment (e.g., Jupyter). Import PyGYRE and the other modules needed for plotting:

```
# Import modules

import pygyre as pg
import matplotlib.pyplot as plt
import numpy as np
```

(you may want to directly cut and paste this code). Next, read the summary file into the variable *s*:

```
# Read data from a gyre summary file

s = pg.read_output('summary.h5')
```

The `pygyre.read_output()` function is able to read files in both *TXT and HDF formats*, returning the data in an `astropy.table.Table` object. To inspect the data on-screen, simply evaluate the table:

```
# Inspect the data

s
```

From this, you'll see that there are three columns in the table, containing the harmonic degree `l`, radial order `n_pg` and frequency `freq` of each mode found during the GYRE run.

Next, plot the frequencies against radial orders via

```
# Plot the data

plt.figure()

plt.plot(s['n_pg'], s['freq'].real)

plt.xlabel('n_pg')
plt.ylabel('Frequency (cyc/day)')
```

(the values in the `freq` column are complex, and we plot the real part). The plot should look something like Fig. 6.1.

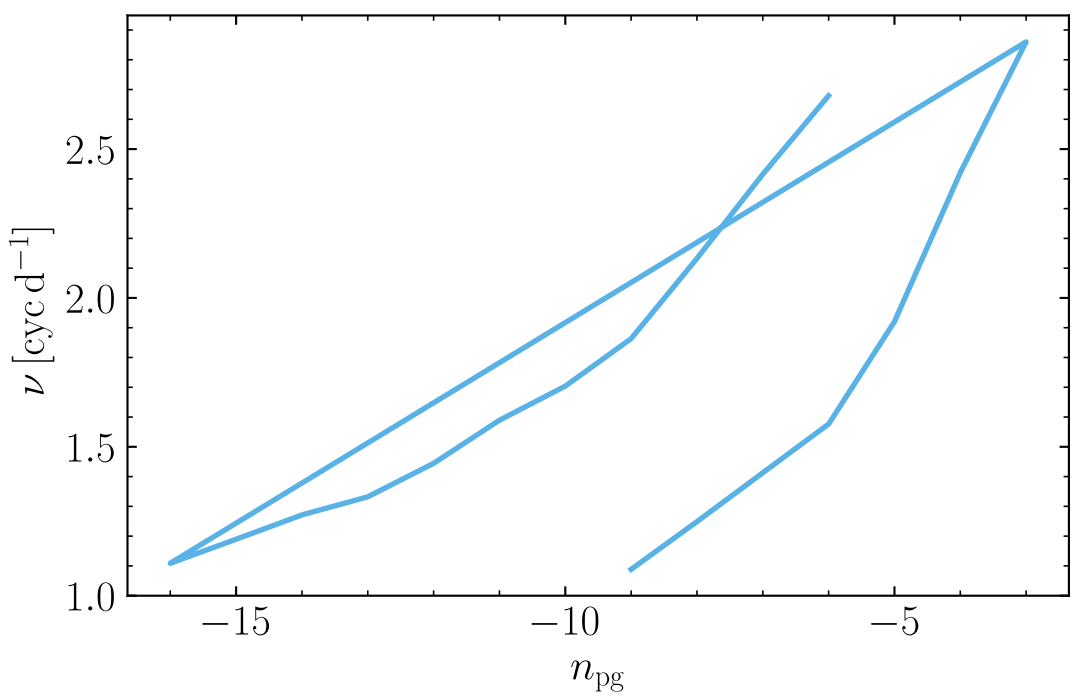


Fig. 6.1: The frequency ν of $\ell = 1$ and $\ell = 2$ modes, plotted against their radial order n_{pg}. (`Source`)

The straight line connecting the two curves occurs because we are plotting both the dipole and quadrupole modes together. To separate them, the table rows can be grouped by harmonic degree:

```
# Plot the data, grouped by harmonic degree

plt.figure()
```

(continues on next page)

(continued from previous page)

```
sg = s.group_by('l')

plt.plot(sg.groups[0]['n_pg'], sg.groups[0]['freq'].real, label=r'l=1')
plt.plot(sg.groups[1]['n_pg'], sg.groups[1]['freq'].real, label=r'l=2')

plt.xlabel('n_pg')
plt.ylabel('Frequency (cyc/day)')

plt.legend()
```

The resulting plot, in Fig. 6.2, looks much better.

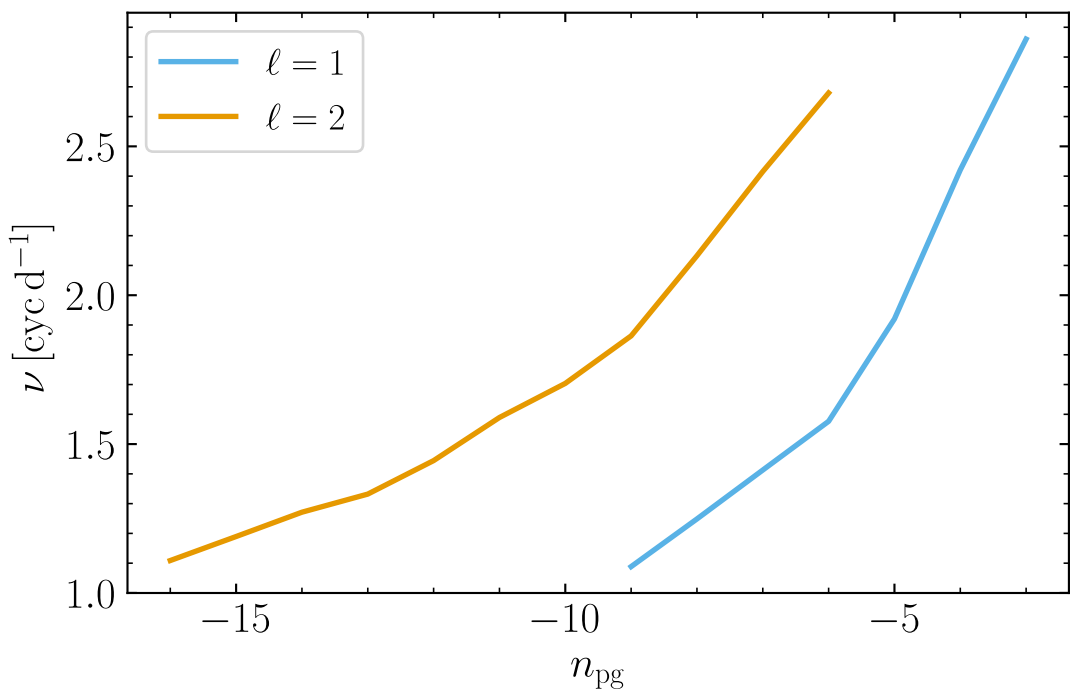


Fig. 6.2: The frequency *nu* of $\ell = 1$ and $\ell = 2$ modes, grouped by ℓ and plotted against their radial order n_{pg}. (`Source`)

6.3 Analyzing a Detail File

Now let's take a look at one of the detail files, for the mode with $\ell = 1$ and $n_{\mathrm{pg}} = -7$. As with the summary file, `pygyre.read_output()` can be used to read the file data into an `astropy.table.Table` object:

```
# Read data from a GYRE detail file

d = pg.read_output('detail.l1.n-7.h5')
```

Inspecting the data using

```
# Inspect the data

d
```

shows there are 7 columns: the fractional radius `x`, the radial displacement eigenfunction `xi_r`, the horizontal displacement eigenfunction `xi_h`, and 4 further columns storing structure coefficients (see the *Detail Files* section for descriptions of these data). Plot the two eigenfunctions using the code

```
# Plot displacement eigenfunctions

plt.figure()

plt.plot(d['x'], d['xi_r'].real, label='xi_r')
plt.plot(d['x'], d['xi_h'].real, label='xi_h')

plt.xlabel('x')

plt.legend()
```

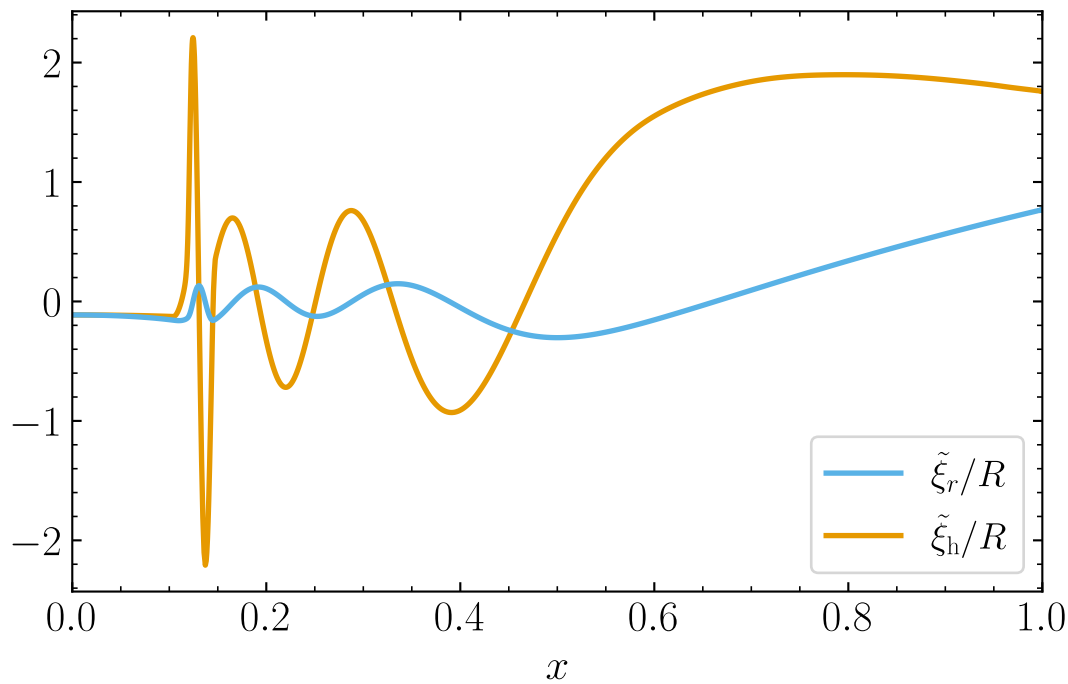


Fig. 6.3: The radial ($\tilde{\xi}_r$) and horizontal ($\tilde{\xi}_\mathrm{h}$) displacement eigenfunctions of the $\ell = 1$, $n_\mathrm{pg} = -7$ mode, plotted against the fractional radius x. (`Source`)

The plot should look something like Fig. 6.3. From this figure , we see that the radial wavelengths of the eigenfunctions become very short around a fractional radius $x \approx 0.125$. To figure out why this is, we can take a look at the star's propagation diagram:

```
# Evaluate dimensionless characteristic frequencies
```

(continues on next page)

(continued from previous page)

```
l = d.meta['l']
omega = d.meta['omega']

x = d['x']
V = d['V_2']*d['x']**2
As = d['As']
c_1 = d['c_1']
Gamma_1 = d['Gamma_1']

d['N2'] = d['As']/d['c_1']
d['Sl2'] = l*(l+1)*Gamma_1/(V*c_1)

# Plot the propagation diagram

plt.figure()

plt.plot(d['x'], d['N2'], label='N^2')
plt.plot(d['x'], d['Sl2'], label='S_l^2')

plt.axhline(omega.real**2, dashes=(4,2))

plt.xlabel('x')
plt.ylabel('omega^2')

plt.ylim(5e-2, 5e2)
plt.yscale('log')
```

Note how we access the mode harmonic degree `l` and dimensionless eigenfrequency `omega` through the table metadata dict `d.meta`. The resulting plot (Fig. 6.4) reveals that the Brunt-Väisälä frequency squared is large around $x \approx 0.125$; this feature is a consequence of the molecular weight gradient zone outside the star's convective core, and results in the short radial wavelengths seen there in Fig. 6.3.

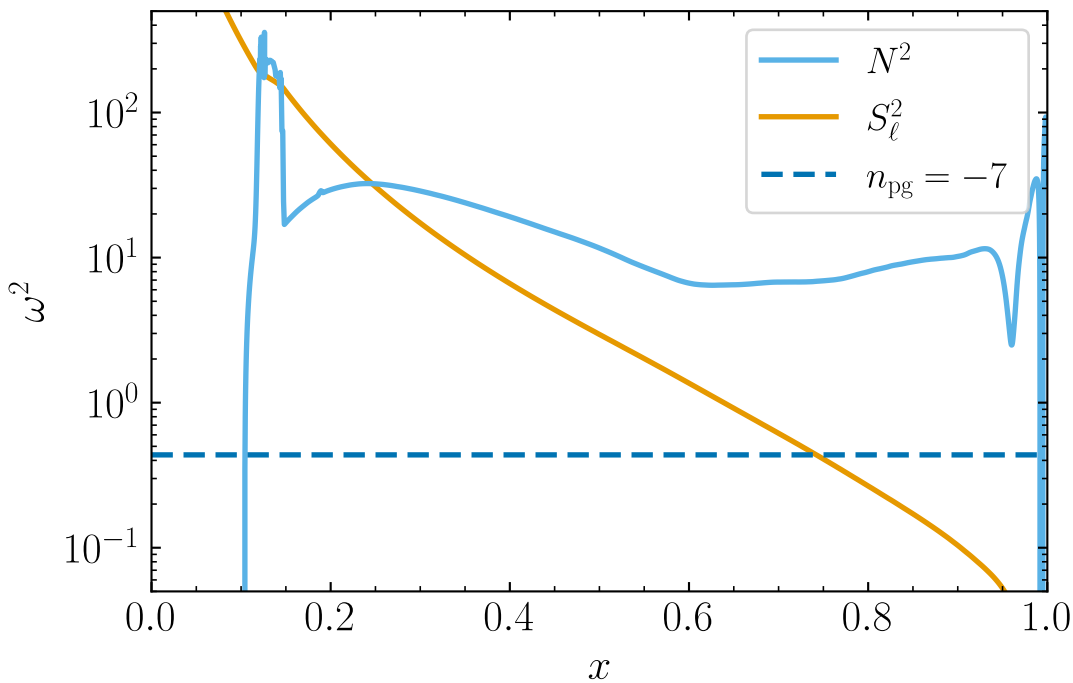


Fig. 6.4: Propagation diagram for the $5\,\mathrm{M}_\odot$ model, plotting the squares of the Brunt-Väisälä (N^2) and Lamb (S_ℓ^2) frequencies versus fractional radius x. The horizontal dashed line shows the frequency squared ω^2 of the $\ell = 1$, $n_{\rm pg} = -7$ mode shown in Fig. 6.3. Regions where ω^2 is smaller (greater) than both N^2 and S_ℓ^2 are gravity (acoustic) propagation regions; other regions are evanescent. (`Source`)

CHAPTER

SEVEN

UNDERSTANDING GRIDS

This chapter describes how the GYRE *frontends* set up frequency and spatial grids, and discusses some strategies for ensuring these grids are optimal.

7.1 Spatial Grids

The various GYRE *frontends* all discretize their equations on a spatial grid $\{x_1, x_2, \ldots, x_N\}$ in the dimensionless radial coordinate $x \equiv r/R$. The computational cost of a calculation scales with the total number of points N in this grid, while the grid's resolution — i.e., the spacing between adjacent points — impacts both the accuracy of solutions, and in the case of the **gyre** frontend, the number of solutions that can be found. (The *Limitations of the Numerical Method* section discusses these behaviors in the context of the stretched string BVP).

7.1.1 Scaffold Grid

A fresh spatial grid is constructed for each iteration of the main computation loop (see the flow-charts in the *Frontends* chapter). This is done under the control of the `&grid` namelist groups; there must be at least one of these, subject to the tag matching rules (see the *Working With Tags* chapter). If there is more than one matching `&grid` namelist group, then the final one is used.

Each grid begins as a *scaffold grid*, comprising the following points:

- an inner point x_i;
- an outer point x_o;
- the subset of points of the source grid satisfying $x_\mathrm{i} < x < x_\mathrm{o}$

The source grid can be either the input model grid, or a grid read from file; this choice is determined by the `scaffold_src` parameter of the `&grid` namelist group. By default, x_i and x_o are obtained from the source grid as well (as its inner-most and outer-most point). However, either or both can be overridden using the `x_i` and `x_o` parameters.

7.1.2 Iterative Refinement

Scaffold grids are refined via a sequence of iterations. During a given iteration, each subinterval $[x_j, x_{j+1}]$ is assessed against various criteria (discussed in greater detail below). If any criteria match, then the subinterval is refined by bisection, inserting an additional point at the midpoint

$$x_{j+1/2} = \frac{x_j + x_{j+1}}{2}.$$

The sequence terminates if no refinements occur during a given iteration, or if the number of completed iterations equals the value specified by the `n_iter_max` parameter of the `&grid` namelist group.

Mechanical Criterion

The wave criterion involves a local analysis of the mechanical parts of the oscillation equations, with the goal of improving resolution where the displacement perturbation $\boldsymbol{\xi}$ is rapidly varying. Within the subinterval $[x_j, x_{j+1}]$, the y_1 and y_2 solutions (see the *Dimensionless Formulation* section) take the approximate form

$$y_{1,2}(x) \sim \exp[\chi\,(\ln x - \ln x_{j+1/2})],$$

where χ is one of the two eigenvalues of the mechanical (upper-left) 2×2 submatrix of the full Jacobian matrix $\mathbf{A}$, evaluated at the midpoint $x_{j+1/2}$.

In propagation zones the imaginary part χ_i of the eigenvalue gives the local wavenumber in $\ln x$ space, and $2\pi\chi_\mathrm{i}^{-1}$ the corresponding wavelength; while in evanescent zones the real part χ_r gives the local exponential growth/decay rate, and χ_r^{-1} the corresponding e-folding length.

Based on this analysis, the criterion for refinement of the subinterval is

$$(\ln x_{j+1} - \ln x_j)\ \max(w_\mathrm{osc}|\chi_\mathrm{i}|, w_\mathrm{exp}|\chi_\mathrm{r}|) > 2\pi,$$

where w_osc and w_exp are user-definable weighting parameters. This causes refinement if the subinterval width (in $\ln x$ space) exceeds w_osc^{-1} times the local wavelength, or $2\pi w_\mathrm{exp}^{-1}$ times the local e-folding length.

Because there are two possible values for χ, the above refinement criterion is applied twice (once for each). Moreover, because χ depends implicitly on the oscillation frequency, the criterion is applied for each frequency in the grid $\{\omega_1, \omega_2, \ldots, \omega_M\}$ (see the *Frequency Grids* section).

Thermal Criterion

Similar to the wave criterion discussed above, the thermal criterion involves a local analysis of the energetic parts of the oscillation equation, with the goal of improving resolution where the thermal timescale is very long and perturbations are almost adiabatic. Within the subinterval $[x_j, x_{j+1}]$, the y_5 and y_6 perturbation take the approximate form

$$y_{5,6}(x) \sim \exp[\pm\tau\,(\ln x - \ln x_{j+1/2})],$$

where $\pm\tau$ are the eigenvalues of the matrix formed from the energetic (bottom-right) 2×2 submatrix of the full Jacobian matrix $\mathbf{A}$, evaluated at the midpoint $x_{j+1/2}$.

Based on this analysis, the criterion for refinement of the subinterval is

$$(\ln x_{j+1} - \ln x_j)\, w_\mathrm{thm}|\tau| > 1,$$

where w_thm is a user-definable weighting parameter.

Because τ depends implicitly on the oscillation frequency, this criterion is applied for each frequency in the grid $\{\omega_1, \omega_2, \ldots, \omega_M\}$.

Structural Criteria

The structural criteria have the goal of improving resolution where the stellar structure coefficients are changing rapidly. For a given coefficient C, the criterion for refinement of the subinterval $[x_j, x_{j+1}]$ is

$$(\ln x_{j+1} - \ln x_j)\, w_{\rm str} \left| \frac{\partial \ln C}{\partial \ln x} \right| > 1,$$

where $w_{\rm str}$ is a user-definable weighting parameter. This criterion is applied separately to the $V_2 \equiv V/x^2$, U, A^*, c_1 and Γ_1 coefficients (see the *Structure Coefficients* section).

Central Criteria

All of the above criteria depend on the logarithmic subinterval width $(\ln x_{j+1} - \ln x_j)$, and cannot be applied to the first subinterval $[x_1, x_2]$ if it extends to the center of the star, $x = 0$. In such cases, the `resolve_ctr` parameter of the `&grid` namelist group determines whether the subinterval is refined. If set to `.FALSE.`, then no refinement occurs; while if set to `.TRUE.`, then the refinement criteria are

$$\chi_{\rm i} > 0$$

or

$$w_{\rm ctr} |\chi_{\rm r}| > 1$$

where χ is the eigenvalue from the local analysis (see the *Mechanical Criterion* section) corresponding to the solution that remains well-behaved at the origin, and $w_{\rm ctr}$ is a user-definable weighting parameter. The first criterion causes refinement if the subinterval is in a propagation zone, and the second if the solution slope $|{\rm d}y/{\rm d}\ln x| \sim |\chi_{\rm r}|$ exceeds $w_{\rm ctr}^{-1}$.

Because χ depends implicitly on the oscillation frequency, these criteria are applied for each frequency in the grid $\{\omega_1, \omega_2, \ldots, \omega_M\}$.

7.1.3 Limiting Controls

A couple of additional controls affect the iterative refinement described above. Refinement of the $[x_j, x_{j+1}]$ subinterval *always* occurs if

$$x_{j+1} - x_j > \Delta x_{\rm max},$$

and *never* occurs if

$$x_{j+1} - x_j < \Delta x_{\rm min},$$

where both $\Delta x_{\rm max}$ and $\Delta x_{\rm min}$ are user-definable.

7.1.4 Namelist Parameters

The full set of parameters supported by the `&grid` namelist group is listed in the *Grid Parameters* section. However, the table below summarizes the mapping between the user-definable controls appearing in the expressions above, and the corresponding namelist parameters.

Symbol	Parameter
$w_{\rm osc}$	`w_osc`
$w_{\rm exp}$	`w_exp`
$w_{\rm thm}$	`w_thm`
$w_{\rm str}$	`w_str`
$w_{\rm ctr}$	`w_ctr`
$\Delta x_{\rm max}$	`dx_max`
$\Delta x_{\rm min}$	`dx_min`

7.1.5 Recommended Values

While `w_exp`, `w_osc` and `w_ctr` all default to zero, it is highly recommended to use non-zero values for these parameters, to ensure adequate resolution of solutions throughout the star. Reasonable starting choices are `w_osc = 10`, `w_exp = 2` and `w_ctr = 10`.

7.2 Frequency Grids

The **gyre** frontend evaluates its discriminant function $\mathcal{D}(\omega)$ on a grid $\{\omega_1, \omega_2, \ldots, \omega_M\}$ in the dimensionless frequency, and scans for changes in the sign of $\mathcal{D}(\omega)$ that are indicative of a bracketed root. The computational cost of a calculation scales with the total number of points M in this grid, while the grid's resolution — i.e., the spacing between adjacent points — impacts the completeness of the modes found by **gyre**. (See the *Limitations of the Numerical Method* section for a discussion of these behaviors).

A fresh frequency grid is constructed for each iteration of the main computation loop (see the flow-chart in the *gyre* section). This is done under the control of the `&scan` namelist groups; there must be at least one of these, subject to the tag matching rules (see the *Working With Tags* chapter). Each `&scan` group creates a separate frequency grid; these are then combined and the discriminant function is evaluated on the merged grid.

7.2.1 Grid Types

The `grid_type` parameter of the `&scan` namelist group controls the overall distribution of points in a frequency grid. There are currently three options:

Linear Grid

When `grid_type = 'LINEAR'`, **gyre** first evaluates a sequence of dimensionless angular frequencies in the grid reference frame according to the formula

$$\omega_i^{\rm g} = \frac{1}{M-1}\left[(M-i)\,\omega_{\rm min}^{\rm g} + (i-1)\,\omega_{\rm max}^{\rm g}\right] \qquad i = 1, 2, \ldots, M.$$

(here, the superscript 'g' indicates that these are frequencies in the grid reference frame). Then, it transforms from the grid frame to the inertial reference frame via

$$\omega_i = \omega_i^{\rm g} + \Delta\omega$$

where $\Delta\omega$ is the frequency shift that transforms from the grid frame to the inertial frame. The actual value of this shift depends on the `grid_frame` parameter:

- When `grid_frame = 'INERTIAL'`, the shift is $\Delta\omega = 0$; the grid frame and the inertial frame coincide.

- When `grid_frame = 'COROT_I'`, the shift is $\Delta\omega = m\,\Omega_{\rm rot}^{\rm i}\sqrt{R^3/GM}$, where $\Omega_{\rm rot}^{\rm i}$ is the rotation angular frequency at the inner boundary of the spatial grid; the grid frame coincides with the local co-rotating frame at that boundary.
- When `grid_frame = 'COROT_O'`, the shift is $\Delta\omega = m\,\Omega_{\rm rot}^{\rm o}\sqrt{R^3/GM}$, where $\Omega_{\rm rot}^{\rm o}$ is the rotation angular frequency at the outer boundary of the spatial grid; the grid frame coincides with the local co-rotating frame at that boundary.

The range spanned by the frequency grid, in the grid frame, is set by $\omega_{\rm min}^{\rm g}$ and $\omega_{\rm max}^{\rm g}$. These are evaluated via

$$\omega_{\rm min}^{\rm g} = \frac{f_{\rm min}}{\widehat{f}_{\rm min}} + \delta\omega - \Delta\omega, \qquad \omega_{\rm max}^{\rm g} = \frac{f_{\rm max}}{\widehat{f}_{\rm max}} + \delta\omega - \Delta\omega,$$

where $f_{\rm min,max}$ are user-definable, $\widehat{f}_{\rm min,max}$ will be discussed below in the *Frequency Units* section, and $\delta\omega$ is the frequency shift that transforms from the frame in which $f_{\rm min,max}$ are defined to the inertial frame. The actual value of this shift depends on the `freq_frame` parameter, which behaves analogously to the `grid_frame` parameter discussed above.

Inverse Grid

When `grid_type = 'INVERSE'`, **gyre** first evaluates a sequence of dimensionless angular frequencies in the grid reference frame according to the formula

$$\omega_i^{\rm g} = (M-1)\left[\frac{(M-i)}{\omega_{\rm min}^{\rm g}} + \frac{(i-1)}{\omega_{\rm max}^{\rm g}}\right]^{-1} \qquad i = 1, 2, \ldots, M.$$

The grid creation then proceeds as described above in the *Linear Grid* section.

File Grid

When `grid_type = 'FILE'`, **gyre** first reads a sequence of dimensioned frequencies $\{f_1, f_2, \ldots, f_M\}$ from an external file named by the `grid_file` parameter. This file is a single-column ASCII table; the number of points M is determined implicitly from the number of lines in the file. Then, it transforms these frequencies via

$$\omega_i = \frac{f_j}{\widehat{f}} + \delta\omega,$$

where $\widehat{f}$ will be discussed below in the *Frequency Units* section, and $\delta\omega$ is the frequency shift that transforms from the frame in which f is defined to the inertial frame. The actual value of this shift depends on the `freq_frame` parameter, which behaves analogously to the `grid_frame` parameter discussed above.

7.2.2 Frequency Units

In the expressions above, terms of the form $f/\widehat{f}$ are used to transform a dimensioned frequency f into a dimensionless one ω. The scale factor $\widehat{f}$ depends on the `freq_units` parameter. Thus, for example, if `freq_units = 'UHZ'`, then f is treated as a linear frequency expressed in μHz, and the scale factor is set by

$$\widehat{f} = \sqrt{\frac{GM}{R^3}}\,\frac{1}{2\pi\,\mu{\rm Hz}}$$

(the factor of 2π comes from the transformation between linear and angular frequency).

The full set of values supported by the `freq_units` parameter is listed in the *Frequency Scan Parameters* section.

7.2.3 Namelist Parameters

The full set of parameters supported by the `&scan` namelist group is listed in the *Frequency Scan Parameters* section. However, the table below summarizes the mapping between the user-definable controls appearing in the expressions above, and the corresponding namelist parameters:

Symbol	Parameter
$f_{\rm min}$	`freq_min`
$f_{\rm max}$	`freq_max`
M	`n_freq`

7.2.4 Recommended Values

The default values `freq_min=1`, `freq_max=10`, `n_freq=10`, together with `grid_type='LINEAR'` are sufficient to find *some* modes — although unlikely the modes that you want. Choosing good values for these parameters requires some degree of judgment, but here are some suggestions:

- The number of points in the frequency grid should be a factor of 2–3 larger than the number of modes you expect **gyre** will find. This is to ensure that the frequency spacing of the grid is everywhere smaller than the anticipated eigenfrequency spacing between adjacent modes (see the *Limitations of the Numerical Method* section for further discussion).
- The distribution of points in the frequency grid should follow anticipated distribution of mode frequencies; this again is to ensure adequate frequency resolution. For p modes, which tend toward a uniform frequency spacing in the asymptotic limit of large radial order, you should chose `grid_type = 'LINEAR'`; likewise, for g modes, which tend toward a uniform period spacing in the asymptotic limit, you should choose `grid_type = 'INVERSE'`.
- When modeling rotating stars, you should choose `grid_frame = 'COROT_I'` or `grid_frame = 'COROT_O'`, because the asymptotic behaviors mentioned above apply in the co-rotating reference frame rather than the inertial one.

CHAPTER

EIGHT

WORKING WITH TAGS

This chapter uses a worked example to demonstrate tags — a simple yet powerful feature allowing a much greater degree of control over GYRE calculations.

8.1 Example Tag Usage

Consider applying **gyre** to calculate the eigenfrequencies of a red giant branch (RGB) stellar model. Because non-radial p-modes in the convective envelope couple with high-order g-modes in the radiative core, the frequency spacing of the non-radial modes is *much* smaller than that of the radial modes. In such cases, we ideally want to use a coarse frequency scan for the radial modes and a fine frequency scan for the non-radial modes.

The following input file, which is designed to work with the $2\,\mathrm{M}_\odot$ RGB model in `$GYRE_DIR/models/mesa/rgb/rgb.mesa`, achieves this goal using tags:

```
&constants
/

&model
  model_type = 'EVOL'  ! Obtain stellar structure from an evolutionary model
  file = 'rgb.mesa'    ! File name of the evolutionary model
  file_format = 'MESA' ! File format of the evolutionary model
/

&mode
  l = 0          ! Harmonic degree
  tag = 'radial' ! Matching tag
/

&mode
  l = 1              ! Harmonic degree
  tag = 'non-radial' ! Matching tag
/

&mode
  l = 2              ! Harmonic degree
  tag = 'non-radial' ! Matching tag
/

&osc
  outer_bound = 'VACUUM' ! Assume the density vanishes at the stellar surface
```

(continues on next page)

(continued from previous page)

```
/

&rot
/

&num
  diff_scheme = 'COLLOC_GL4' ! 4th-order collocation scheme for difference equations
/

&scan
  grid_type = 'LINEAR' ! Scan grid uniform in frequency
  freq_min = 41        ! Minimum frequency to scan from
  freq_max = 43        ! Maximum frequency to scan to
  freq_units = 'UHZ'   ! Frequency units
  n_freq = 10          ! Number of frequency points in scan
  tag_list = 'radial'  ! Pair only with 'radial' &mode groups
/

&scan
  grid_type = 'LINEAR'           ! Scan grid uniform in frequency
  freq_min = 41                  ! Minimum frequency to scan from
  freq_max = 43           ! Maximum frequency to scan to
  freq_units = 'UHZ'      ! Frequency units
  n_freq = 100            ! Number of frequency points in scan
  tag_list = 'non-radial' ! Pair only with 'non-radial' &mode groups
/

&grid
  w_osc = 10  ! Oscillatory region weight parameter
  w_exp = 2   ! Exponential region weight parameter
  w_ctr = 100 ! Central region weight parameter
/

&ad_output
  summary_file = 'summary.h5'                   ! File name for summary file
  summary_item_list = 'l,n_pg,freq,freq_units' ! Items to appear in summary file
  freq_units = 'UHZ'                            ! Units of freq output items
/

&nad_output
/
```

Observe that each `&mode` namelist groups has a `tag` parameter. When processing a given `&mode`, **gyre** pairs it up with other namelist groups that match one of the following criteria:

- The namelist group doesn't have a `tag_list` parameter;
- The namelist does have a `tag_list` parameter, *and* the parameter value (a comma-separated list) contains the tag value defined in the `&mode` group.

In the example given above, the `&osc` namelist group doesn't have a `tag_list` parameter; therefore, it is paired with all three `&mode` namelist groups, irrespective of their `tag` values. However, the two `&scan` namelist groups each have `tag_list` parameters. In the first group the `radial` tag appears, and so this group is paired with the first `&mode` namelist group (i.e., the $\ell = 0$ mode). Likewise, in the second group the `non-radial` tag appears, and so this group

is paired with the second and third `&mode` namelist groups (i.e., the $\ell = 1$ and $\ell = 2$ modes).

8.2 Tag Rules

In addition to the matching criteria given above, there are a couple of rules that must be obeyed by tags:

- Tag names can't contain commas (however, they can be otherwise arbitrary);
- If a `&mode` namelist group doesn't have a `tag` parameter, then only namelists without a `tag_list` parameter will be paired with it;
- The `&constants`, `&model`, `&ad_output` and `&nad_output` namelist groups don't support tags.

CHAPTER

NINE

ADVANCED USAGE

This chapter covers a number of topics that fall under the umbrella of 'advanced' GYRE usage.

9.1 Non-Adiabatic Oscillations

This section discusses how to undertake *non-adiabatic* oscillation calculations using the **gyre** frontend. Asteroseismic studies typically rely on adiabatic calculations, because the frequencies of oscillation modes are the primary focus. However, for heat-driven modes the linear growth or damping rates can also be of interest — and evaluating these requires that non-adiabatic effects are included in the oscillation equations.

Note: Not all types of stellar mode include the necessary data (e.g., thermodynamic coefficients, opacity partial derivatives) to undertake non-adiabatic calculations. The *Model Capabilities* section summarizes this information.

9.1.1 Overview

To include non-adiabatic effects **gyre** augments the linearized mass, momentum and Poisson equations with the linearized heat and radiative diffusion equations (see the *Linearized Equations* section for full details). With these additions, the equations and their solutions become complex quantities. The assumed time dependence for perturbations is $\propto \exp(-\mathrm{i}\sigma t)$; therefore, the real part σ_R and imaginary part σ_I of the eigenfrequency are related to the mode period Π and growth e-folding time τ, respectively, via

$$\Pi = \frac{2\pi}{\sigma_\mathrm{R}}, \qquad \tau = \frac{1}{\sigma_\mathrm{I}}.$$

Solving the non-adiabatic equations proceeds using the same general approach as in the adiabatic case, by searching for the roots of a discriminant function $\mathcal{D}(\omega)$ (see the *Numerical Methods* chapter for more details). However, a challenge is that there is no simple way to bracket roots in the complex plane. Instead, **gyre** must generate initial trial roots that are close to the true roots, and then refine them iteratively. Currently, **gyre** offers three methods for establishing the trial roots.

9.1.2 Adiabatic Method

The adiabatic method involves adopting the (real) roots found from adiabatic calculations as the initial trial roots for the non-adiabatic problem. This works well as long as the adiabatic and non-adiabatic roots lie close together in the complex plane — typically, when the oscillation modes are only weakly non-adiabatic, with $|\sigma_{\rm I}/\sigma_{\rm R}| \ll 1$.

To perform non-adiabatic calculations with the adiabatic method, set the following parameters in the `&osc` namelist group:

- `nonadiabatic=.TRUE.`
- `adiabatic=.TRUE.`[1]

and the following parameters in the `&num` namelist group:

- `ad_search='BRACKET'`Page 52, 1
- `nad_search='AD'`[1]

You may also wish to use the following setting in the `&num` namelist group:

- `diff_scheme='MAGNUS_GL2'`

This tells **gyre** to evaluate the finite-difference equations using the 2nd order Magnus scheme; experience suggests that this gives the most reliable convergence for the root refinement.

An example of the adiabatic method in action can be found in the *$GYRE_DIR*`/test/nad/mesa/bcep/gyre.in` namelist input file, which is set up to find $\ell = 0, \ldots, 3$ modes of a $20\,{\rm M}_\odot$ β Cephei model using the adiabatic method. The important parts are as follows:

```
&osc
  nonadiabatic = .TRUE.
/

&num
  diff_scheme = 'MAGNUS_GL2'
  restrict_roots = .FALSE.
/

&scan
  grid_type = 'LINEAR'
  freq_min = 3.0
  freq_max = 10.0
  n_freq = 50
/
```

Note the `nonadiabatic` parameter in the `&osc` namelist group, and the `diff_scheme` parameter in the `&num` namelist group. The `restrict_roots=.FALSE.` setting, also in the `&num` namelist group, tells **gyre** not to reject any modes that have $\sigma_{\rm R}$ outside the frequency range specified by the `&scan` namelist group; this ensures that modes whose non-adiabatic frequencies fall just outside the frequency grid are still found.

[1] This is the default setting; you don't need to include it explicitly.

9.1.3 Minmod Method

The minmod method involves evaluating the discriminant function along the real-ω axis, and then adopting local minima in its modulus $|\mathcal{D}|$ as the initial trial roots for the non-adiabatic problem. The method is described in full in Goldstein & Townsend (2020); as shown there, it does not perform significantly better than the adiabatic method, and is included in **gyre** for the sake of completeness.

To perform non-adiabatic calculations with the minmod method, set the following parameters in the `&osc` namelist group:

- `nonadiabatic=.TRUE.`
- `adiabatic=.FALSE.`[2]

and the following parameters in the `&num` namelist group:

- `nad_search='MINMOD'`

As with the adiabatic method, you may also wish to use the following setting in the `&num` namelist group:

- `diff_scheme='MAGNUS_GL2'`

An example of the minmod method in action can be found in the *$GYRE_DIR*`/test/nad/mesa/bcep-minmod/gyre.in` namelist input file, which is equivalent to *$GYRE_DIR*`/test/nad/mesa/bcep/gyre.in` but using the minmod method. The important parts are as follows:

```
&osc
  adiabatic = .FALSE.
  nonadiabatic = .TRUE.
/

&num
  diff_scheme = 'MAGNUS_GL2'
  nad_search = 'MINMOD'
  restrict_roots = .FALSE.
/

&scan
  grid_type = 'LINEAR'
  freq_min = 3.0
  freq_max = 10.0
  n_freq = 250
/
```

Note the additional `nad_search='MINMOD'` parameter in the `&num` namelist group, which stipulates that the minmod method should be used.

[2] This is optional; leave it out if you want **gyre** to perform adiabatic calculations as well.

9.1.4 Contour Method

The contour method involves evaluating the discriminant function on a grid in the complex-ω plane, and then adopting intersections between the real zero-contours $\mathcal{D}_\mathrm{R} = 0$, and the corresponding imaginary ones $\mathcal{D}_\mathrm{I} = 0$, as the initial trial roots for the non-adiabatic problem. The method is described in full in Goldstein & Townsend (2020); it is very effective even for strongly non-adiabatic modes with $|\sigma_\mathrm{I}/\sigma_\mathrm{R}| \sim 1$, although there is an increased computational cost (see *here* for one strategy for mitigating this cost).

To perform non-adiabatic calculations with the contour method, set the following parameters in the `&osc` namelist group:

- `nonadiabatic=.TRUE.`
- `adiabatic=.FALSE.` Page 53, 2

and the following parameters in the `&num` namelist group:

- `nad_search='CONTOUR'`

You must also ensure that at least one `&scan` namelist group with `axis='REAL'` is present, and likewise at least one with `axis='IMAG'`. Together, these groups define the real and imaginary axes of the discriminant grid in the complex-ω plane. As a rule of thumb, the resolution along the imaginary axis should be comparable to that along the real axis; this ensures that the contour-tracing algorithm behaves well.

Finally, as with the adiabatic method, you may also wish to use the following setting in the `&num` namelist group:

- `diff_scheme='MAGNUS_GL2'`

Note: Because g modes are spaced uniformly in period (in the asymptotic limit of large radial order), it would seem sensible to set `grid_type='INVERSE'` in the `&scan` namelist group(s) that correspond to the real axis (i.e., `axis='REAL'`). However, this typically results in a mismatch between the resolution of the real and imaginary axes, and the contour method doesn't perform well. A fix for this issue will be forthcoming in a future release of GYRE, but in the meantime it's probably best to avoid the contour method for g modes.

An example of the minmod method in action can be found in the *$GYRE_DIR*`/test/nad/mesa/bcep-contour/gyre.in` namelist input file, which is equivalent to *$GYRE_DIR*`/test/nad/mesa/bcep/gyre.in` but using the minmod method. The important parts are as follows:

```
&osc
  adiabatic = .FALSE.
  nonadiabatic = .TRUE.
/

&num
  diff_scheme = 'MAGNUS_GL2'
  restrict_roots = .FALSE.
  nad_search = 'CONTOUR'
/

&scan
  axis = 'REAL'
  grid_type = 'LINEAR'
  freq_min = 3.0
  freq_max = 10.0
  n_freq = 50
/
```

(continues on next page)

(continued from previous page)

```
&scan
  axis = 'IMAG'
  grid_type = 'LINEAR'
  freq_min = -0.28
  freq_max = 0.28
  n_freq = 5
/
```

Note the additional `nad_search='CONTOUR'` parameter in the `&num` namelist group, which stipulates that the contour method should be used; and, the fact that there are now two `&scan` namelist groups, one with `axis='REAL'` and the other with `axis='IMAG'`.

9.2 Tidal Forcing

This section discusses how to evaluate the stellar response (fluid displacements and perturbations) to tidal forcing, using the **gyre_tides** frontend. The response data can be used to calculate the secular rates-of-change of orbital elements, or to synthesize a light curve for a tidally distorted star.

9.2.1 Overview

As discussed in the *Tidal Effects* section, the tidal gravitational potential (16.12) of an orbiting companion can be expressed as a superposition of partial potentials $\Phi_{\mathrm{T};\ell,m,k}$. For a given `&tide` namelist group appearing in the namelist input file, **gyre_tides** solves for the response of the star to each term in the superposition.

9.2.2 Truncating the Sums

Although the sums appearing in eqn. (16.12) are formally infinite, the terms with large harmonic degree ℓ and/or orbital harmonic k typically produce a negligible response. **gyre_tides** offers a couple of approaches for truncating the sums by dropping these terms. The simplest is to set limits on the maximum values of the indices, through the `l_max`, `k_min` and `k_max` parameters of the `&tide` namelist group (if desired, minimum values can also be set using the corresponding `l_min` and `k_min` parameters).

A slightly more sophisticated approach is to set these parameters to large-ish values (say, a few hundred), and then also set one or both of the `y_T_thresh_abs` and `y_T_thresh_rel` parameters. These establish a threshold on the magnitude of

$$y_{\mathrm{T}} \equiv \frac{\tilde{\Phi}_{\mathrm{T};\ell,m,k}}{GM/R}$$

for a given tidal partial potential $\tilde{\Phi}_{\mathrm{T};\ell,m,k}$ (see eqn. 16.14) to be included in calculations; if $|y_{\mathrm{T}}|$ does not meet this threshold, it is ignored.

9.2.3 Optimizing Grids

During the *iterative refinement* process used in setting up spatial grids, the refinement criteria are evaluated for every partial tide under consideration. If the co-rotating forcing frequency

$$\sigma_{\rm c} \equiv k\Omega_{\rm orb} - m\Omega_{\rm rot}$$

associated with a specific partial tidal potential is small compared to the dynamical frequency of the star, many levels of refinement will occur. While this is exactly what one wants in oscillation calculations (because low-frequency modes have short spatial wavelengths), it often isn't necessary in tidal calculations because the response of a star to low-frequency forcing is the long-wavelength equilibrium tide.

One way of preventing over-refinement due to low-frequency partial potentials is to set the `omega_c_thresh` parameter in the `&tide` namelist group. This establishes a threshold on the dimensionless frequency $\omega_{\rm c} \equiv \sigma_{\rm c}\sqrt{R^3/GM}$; partial potentials with $|\omega_{\rm c}|$ below this threshold are treated as static tides ($\omega_{\rm c} = 0$), and are not considered during the iterative refinement process.

An alternative approach is to avoid iterative refinement altogether, instead obtaining the spatial grid from an external file (see the `'FILE'` option of the `scaffold_src` parameter, in the *Grid Parameters* section). This is the most flexible approach, but creating a grid that will adequately resolve the response to each partial potential requires some care.

9.2.4 Output Files

`gyre_tides` writes response data to summary and detail files. One detail file is created for each partial potential evaluated, and the summary file collects together global data for all partial potentials across all `&tide` namelist groups. The `id` output item can be used to determine which group a given response belongs to.

The following Python code demonstrates how the summary data might be used to evaluate the secular rates-of-change of orbital semi-major axis, eccentricity, and argument of periastron, and the stellar torque. The expression for `e_dot` mirrors eqn. (23) of Sun et al. (2023), and for `J_dot` eqn. (25) *ibid.*

```
import pygyre as pg
import numpy as np

# Read summary file from gyre_tides

s = pg.read_output('summary.h5')

# Extract the first set of responses

sg = s.group_by('id').groups[0]

Omega_orb = sg['Omega_orb']
R_a = sg['R_a']
q = sg['q']

eps_T = R_a**3*q

l = sg['l']
m = sg['m']
k = sg['k']

cbar = sg['cbar']
```

(continues on next page)

(continued from previous page)

```
Fbar = -0.5*sg['eul_Phi_ref']/(cbar*eps_T)

x = sg['x_ref']

Gbar_1 = sg['Gbar_1']
Gbar_2 = sg['Gbar_2']
Gbar_3 = sg['Gbar_3']
Gbar_4 = sg['Gbar_4']

# Evaluate secular rates-of-change

kap = np.empty(len(l))

for i in range(len(kap)):
    if k[i] == 0:
        if m[i] == 0:
            kap[i] = 0.5
        elif m[i] > 0 and m[i] <= l[i]:
            kap[i] = 1.
        else:
            kap[i] = 0.
    elif k[i] > 0:
        kap[i] = 1.
    else:
        kap[i] = 0.

# Semi-major axis (units of R per dynamical timescale)

a_dot = np.sum(4. * Omega_orb * q / R_a *
    (R_a)**(l+3) * (x)**(l+1) * kap * Fbar.imag * Gbar_2)

# Eccentricity (units of per dynamical timescale)

e_dot = np.sum(4. * Omega_orb * q *
    (R_a)**(l+3) * (x)**(l+1) * kap * Fbar.imag * Gbar_3)

# Argument of periastron (units of radians per dynamical timescale)

pom_dot = np.sum(4. * Omega_orb * q *
    (R_a)**(l+3) * (x)**(l+1) * kap * Fbar.real * Gbar_1)

# Angular momentum (units of GM^2/R)

J_dot = np.sun(4. * q**2 * R_a *
    (R_a)**(l+3) * (x)**(l+1) * kap * Fbar.imag * Gbar_4)
```

9.3 Including Rotation

This section discusses how to include the effects of rotation in GYRE calculations. See the *Rotation Effects* section for further details of GYRE's rotation treatment.

9.3.1 Setting the Rotation Rate

There are two different ways to define the rotation angular frequency $\Omega_{\rm rot}$, via parameters in the `&rot` namelist group.

- If `Omega_rot_source = 'MODEL'`, then differential rotation is assumed with a spatially varying $\Omega_{\rm rot}$ obtained from the stellar model. If the model doesn't have this capability (see the *Model Capabilities* section), then $\Omega_{\rm rot}$ is set to zero throughout the star.
- If `Omega_rot_source = 'UNIFORM'`, then uniform rotation is assumed with a spatially constant $\Omega_{\rm rot}$ set by the `Omega_rot` and `Omega_rot_units` parameters.

9.3.2 Incorporating Doppler Effects

The *Doppler Shift* effect is incorporated automatically whenever calculations are performed with non-zero $\Omega_{\rm rot}$ and mode azimuthal order m. To disable this effect, set `m= 0` in the `&mode` namelist group.

9.3.3 Incorporating Coriolis Effects

Incorporating the effects of the Coriolis force can be done using a *perturbative treatment* or a *non-perturbative treatment*. In the former case the effects are be applied as a post-calculation correction to non-rotating eigenfrequencies (see the `domega_rot` output item in the *Summary Files* and *Detail Files* sections). In the latter case, the traditional approximation of rotation (TAR) can be enabled by setting `coriolis_method='TAR'` in the `&rot` namelist group.

The *TAR solution family* is controlled by the `rossby` parameter of the `&rot` namelist group; set to `.TRUE.` for the Rossby family, and to `.FALSE.` for the gravito-acoustic family.

CHAPTER

TEN

FREQUENTLY ASKED QUESTIONS

This is a list of Frequently Asked Questions about GYRE. Suggestions for new entries are always most welcome!

10.1 How Do I…

…obtain the GYRE source code?
See the *Download* section.

…compile GYRE?
See the *Compile* section.

…compile a statically linked version of GYRE?
Set the `STATIC` environment variable to `yes` prior to compilation. Note that this currently only works on Linux platforms, and when then `CRMATH` environment variable is set to `no`.

…run GYRE on multiple cores?
GYRE can take advantage of multiple processors in a shared-memory (multicore) computer through its use of OpenMP. This functionality should be enabled by default, but you can nevertheless force it by setting the `OMP` environment variable to `yes` prior to compilation. Then, set the `OMP_NUM_THREADS` environment variable to the number of threads you want to use.

…run GYRE on a cluster?
GYRE can take advantage of multiple nodes in a computer cluster through its use of MPI. To enable this functionality, set the `MPI` environment variable to `yes` prior to compilation. Note that you'll need to use a version of the MESA Software Development Kit built with MPI support for your cluster's specific networking technology (e.g., Infiniband); contact the GYRE team for more details.

…pronounce GYRE?
With a soft 'g' rhyming with 'wire', like `this`.

…cite GYRE?
See the *Citing GYRE* section.

…get support for a problem I'm having?
Post a message to one of the GYRE discussion forums.

…contact the GYRE team?
Send an email to Rich Townsend.

…access the documentation for older releases?
Click on the 'v:…' dropdown menu at the bottom of the left-hand panel. Note that this menu is only available when viewing the documentation on Read the Docs; it isn't available in a local build within the *$GYRE_DIR/*`doc/sphinx` directory. Also, the legacy documentation is a work-in-progress, and remains incomplete.

10.2 What Does…

…'Failed during deflate narrow

[out-of-domain frequency' mean?] This is an indication that GYRE's root solver wandered out of bounds when trying to find a complex root of the discriminant function. Try running with a different choice of `diff_scheme` parameter (`MAGNUS_GL2` seems to be the most robust), and/or using contour method instead (see the *Non-Adiabatic Oscillations* chapter).

10.3 Why Does…

…the error 'Illegal Instruction' arise on MacOS when running with large grid sizes?

This behavior is typically caused by overflow of the OpenMP stack (for more info see here). Try setting the `OMP_STACKSIZE` environment variable to 500K or 1M.

CHAPTER

ELEVEN

TROUBLESHOOTING

This chapter discusses various problems that can arise during normal GYRE operation, and steps that can be taken to resolve them.

11.1 Missing Modes

For adiabatic oscillation calculations using **gyre**, the radial order $n_{\rm pg}$ of modes found should be monotonic-increasing[1]. Departures from this behavior can occur for a number of reasons.

11.1.1 Insufficient Frequency Resolution

If the *frequency grid* has insufficient resolution, then **gyre** can skip modes during the bracketing phase, as discussed in the *Limitations of the Numerical Method* section. The signature of insufficient frequency resolution is that an even number of consecutive modes is missed — most often, an adjacent pair of modes.

To fix this problem, first check that the distribution of points in the frequency grids matches (approximately) the expected distribution of mode eigenfrequencies:

- In the asymptotic limit of large radial order, p modes are uniformly distributed in frequency (see, e.g., Aerts et al., 2010). Hence, to search for these modes set `grid_type='LINEAR'` in the `&scan` namelist group(s).
- Likewise, in the asymptotic limit of large radial order, g modes are uniformly distributed in period. Hence, to search for these modes set `grid_type='INVERSE'` in the `&scan` namelist group(s).
- For rotating stars, the asymptotic behaviors mentioned apply in the co-rotating reference frame, not in the inertial reference frame. So, be sure to also set `grid_frame ='COROT_I'|'COROT_O'` in the `&scan` namelist group.

Next, try increasing the number of points in the frequency grids, simply by increasing the `n_freq` parameter in the `&scan` namelist group(s).

Tip: A good rule of thumb is that `n_freq` should be around 5 times larger than the number of modes expected to be found.

[1] The sole exception is $\ell = 1$ modes, where $n_{\rm pg} = 0$ is skipped due to the way the Takata 2006b classification scheme is set up.

11.1.2 Insufficient Spatial Resolution

If the *spatial grid* has insufficient resolution, then certain modes can simply be absent from the (finite) set of distinct numerical solutions, as discussed in the *Limitations of the Numerical Method* section. The signature of insufficient spatial resolution is that modes that *are* found have radial orders comparable to the number of grid points N in the grid; and that the eigenfunctions of these modes are barely resolved (cf. Fig. 5.5).

To fix this problem, first check that the `w_osc`, `w_exp` and `w_ctr` weighting parameters in the `&grid` namelist group are set to reasonable values (see the *Recommended Values* section). If that doesn't improve things, try gradually increasing both `w_osc` and `w_ctr`.

11.1.3 Non-adiabatic Effects

When undertaking *non-adiabatic calculations*, modes can be mis-classified or completely missed. The former situation arises because the expectation of monotonic-increasing $n_{\rm pg}$ formally applies only to adiabatic oscillations; while it can also work reasonably well for weakly non-adiabatic cases, there are no guarantees. If mis-classification does occur, then it must be fixed manually by determining which adiabatic mode the problematic non-adiabatic mode corresponds to.

Missing modes occur for a different reason: if a mode has a large growth rate, then the usual *adiabatic method* for establishing initial trial roots can fail to find it. In such cases, the alternative *contour method* performs very well.

11.2 Duplicated Modes

Sometimes two oscillation modes with the same $n_{\rm pg}$ are found during a **`gyre`** calculation. This violates the expectation that $n_{\rm pg}$ be monotonic-increasing, and can happen for a few reasons.

11.2.1 Bad Stellar Model

If the input stellar doesn't conserve mass properly, then one or more bogus (unphysical) modes can appear with the same $n_{\rm pg}$ as an existing mode, but a significantly different frequency. Such modes can also be spotted because their radial order is very different from the adjacent-in-frequency modes.

Bogus modes arise because the input stellar model doesn't conserve mass. GYRE` assumes that the density ρ and interior mass M_r are related by equation 16.1. Given that there are many different ways to discretize this equation, there is a certain amount of numerical 'slop' that arises when going from discrete form assumed in the stellar evolution code that generated the model (e.g., MESA) through to the discrete form implicitly assumed by GYRE. This slop can sometimes produce bogus modes, and the fix is to re-create the model with a higher spatial resolution.

Another possibility arises if the input stellar model contains density discontinuities that aren't properly marked using *double points*. Often, these discontinuities can be diagnosed by plotting the I_0 or I_1 first integrals (for radial and dipole modes, respectively) as a function of radius. These first integrals should vanish everywhere (as shown by Takata 2006a), but will typically show abrupt jumps to non-zero values at the location of unmarked discontinuities. The fix is to re-create the model with double points inserted as necessary; in the case of MESA models, this can be achieved using the `add_double_points_to_pulse_data` parameter of the `&controls` namelist group.

11.2.2 Non-adiabatic Effects

When non-adiabatic effects cause a mode to be mis-classified (as discussed in the *Missing Modes* section), often the incorrect $n_{\rm pg}$ value duplicates that of another mode. As before, a mis-classification must be fixed manually by determining which adiabatic mode the problematic non-adiabatic mode corresponds to.

11.3 Long Runtimes

11.3.1 gyre

Long runtimes with the **`gyre`** frontend occur when the *spatial grid* and/or *frequency grid* contain many points. The execution time to process a single `&mode` namelist group can be approximated by

$$\tau \approx C_{\rm b} N M + C_{\rm s} N N_{\rm m},$$

where N is the number of spatial points, M is the number of frequency points, $N_{\rm m}$ is the number of modes found, and $C_{\rm b}$ and $C_{\rm s}$ are constants. The first ($C_{\rm b}$) term represents the time take to bracket roots of the discriminant function, and the second ($C_{\rm s}$) the time taken to solve for these roots (see the *Numerical Methods* chapter for details).

The key to ensuring reasonable runtimes lies in judicious choice of parameters in the `&scan` namelist group(s). The `n_freq` parameter obviously has an impact on τ, as it directly sets M. However, the `freq_min` and `freq_max` parameters also influence τ, due to the way the spatial grid is constructed. If the frequency scan includes parts of the star's oscillation spectrum containing modes with very large radial orders (whether p modes or g modes), then GYRE's *iterative refinement algorithm* will insert many grid points in order to resolve the modes' wavefunctions. This can ultimately lead to huge N and very long runtimes.

11.3.2 gyre_tides

The narrative is similar with the **`gyre_tides`** frontend, although there are no frequency grids involved. The execution time to process a single `&orbit` namelist group can be approximated by

$$\tau \approx C_{\rm t} N$$

where $C_{\rm t}$ is a constant.

CHAPTER

TWELVE

INSTALLATION

This chapter discusses GYRE installation in detail. If you just want to get up and running, have a look at the *Quick Start* chapter.

12.1 Pre-Requisites

To compile and run GYRE, you'll need the following software components:

- A modern (2003+) Fortran compiler
- The BLAS linear algebra library
- The LAPACK linear algebra library
- The LAPACK95 Fortran 95 interface to LAPACK
- The HDF5 data management library
- The crlibm correctly rounded math library
- The crmath Fortran 2003 interface to crlibm
- An OpenMP-aware version of the ODEPACK differential equation library (optional)

On Linux and MacOS platforms, these components are bundled together in the MESA Software Development Kit (SDK), which can be downloaded from the MESA SDK homepage. Using this SDK is strongly recommended.

12.2 Building GYRE

12.2.1 Download

Download the GYRE source code, and unpack it from the command line using the **tar** utility:

```
$ tar xf gyre-7.1.tar.gz
```

Set the GYRE_DIR environment variable with the path to the newly created source directory; this can be achieved, e.g., using the **realpath** command[1]:

```
$ export GYRE_DIR_DIR=$(realpath gyre-7.1)
```

[1] The **realpath** command is included in the GNU CoreUtils package. Mac OS users can install CoreUtils using MacPorts or Homebrew.

12.2.2 Compile

Compile GYRE using the **make** utility:

```
$ make -j -C $GYRE_DIR
```

(the **-j** flags tells **make** to use multiple cores, speeding up the build).

12.2.3 Test

To check that GYRE has compiled correctly and gives reasonable results, you can run the calculation test suite via the command

```
$ make -C $GYRE_DIR test
```

The initial output from the tests should look something like this:

```
TEST numerics (OpenMP)...
 ...succeeded
TEST numerics (band matrix)...
 ...succeeded
TEST numerics (*_DELTA frequency units)...
 ...succeeded
TEST numerics (rotation, Doppler shift)...
 ...succeeded
TEST numerics (rotation, traditional approximation)...
 ...succeeded
```

If things go awry, consult the *Troubleshooting* chapter.

12.3 Custom Builds

Custom builds of GYRE can be created by setting certain environment variables, and/or variables in the file *$GYRE_DIR*/src/build/Makefile, to the value yes. The following variables are currently supported:

DEBUG
: Enable debugging mode (default no)

OMP
: Enable OpenMP parallelization (default yes)

MPI
: Enable MPI parallelization (default no)

DOUBLE_PRECISION
: Use double precision floating point arithmetic (default yes)

CRMATH
: Use correctly rounded math functions (default yes)

IEEE
: Use Fortran IEEE floating point features (default no)

FPE
: Enable floating point exception checks (default yes)

HDF5
: Include HDF5 support (default yes)

EXPERIMENTAL
: Enable experimental features (default `no`)

If a variable is not set, then its default value is assumed.

12.4 Git Access

Sometimes, you'll want to try out new features in GYRE that haven't yet made it into a formal release. In such cases, you can check out GYRE directly from the https://github.com/rhdtownsend/gyre git repository on GitHub:

```
$ git clone https://github.com/rhdtownsend/gyre.git
```

However, a word of caution: GYRE is under constant development, and features in the main (`master`) branch can change without warning.

footnote

CHAPTER

THIRTEEN

NAMELIST INPUT FILES

This chapter describes the various groups that can appear in the input files read by the *GYRE frontends*. These files are in Fortran's namelist format, a simple text-based format containing one or more namelist groups. Each group begins with the line `&name` (where `name` is the name of the group); a list of parameter-value pairs then follows, and the group ends with a slash `/`.

13.1 Constants

The `&constants` namelist group defines various physical constants, as follows:

G_GRAVITY
: Gravitational constant G

C_LIGHT
: Speed of light *in vacuo* c

A_RADIATION
: Radiation constant a

M_SUN
: Solar mass $\mathrm{M}_\odot$

R_SUN
: Solar radius $\mathrm{R}_\odot$

L_SUN
: Solar luminosity $\mathrm{L}_\odot$

GYRE_DIR
: Top-level GYRE directory; overrides the `GYRE_DIR` environment variable

All of these constants are in cgs units (where applicable), and the default values are defined in *$GYRE_DIR*`/src/common/gyre_constants.fpp`.

13.2 Grid Parameters

The `&grid` namelist group defines the parameters used to set up the spatial grid, as follows:

`scaffold_src` (default `'MODEL'`)
Source for scaffold grid; one of:

- `'MODEL'` : Obtained from the stellar model
- `'FILE'` : Read from a file (see the `file` and `file_format` parameters)

`x_i` (default based on model grid)
Inner boundary coordinate of calculation grid

`x_o` (default based on model grid)
Outer boundary coordinate of calculation grid

`w_osc` (default `0`)
Oscillatory weighting parameter $w_{\rm osc}$

`w_exp` (default `0`)
Exponential weighting parameter $w_{\rm exp}$

`w_ctr` (default `0`)
Center weighting parameter $w_{\rm ctr}$

`w_thm` (default `0`)
Thermal weighting parameter $w_{\rm thm}$

`w_str` (default `0`)
Structural weighting parameter $w_{\rm str}$

`dx_min` (default `SQRT(EPSILON(1._WP))`)
Minimum spacing of grid points

`dx_max` (default `HUGE(0._WP)`)
Maximum spacing of grid points

`n_iter_max` (default `32`)
Maximum number of refinement iterations

`resolve_ctr` (default `.TRUE.`)
Flag to resolve central evanescent region

`file` (default `''`)
Name of file containing scaffold grid data (when `scaffold_src='FILE'`)

`file_format` (default `''`)
Format of file containing scaffold grid data (when `scaffold_src='FILE'`); one of:

- `'TEXT'`: text file with one abscissa value per line
- `'DETAIL'`: *detail* file with abscissa values provided in `x` dataset

`tag_list` (default `''`, which matches all)
Comma-separated list of `&mode` tags to match

See the *Spatial Grids* section for further details, in particular a discussion of how the weighting (`w_*`) parameters work.

13.3 Stellar Model Parameters

The `&model` namelist group defines stellar model parameters, as follows:

`model_type`
Type of model to use; one of:

- `'HOM'` : Homogeneous compressible model
- `'POLY'` : Polytropic model read from external file
- `'EVOL'` : Evolutionary model read from external file

`file`
Name of file (when `model_type= 'POLY'|'EVOL'`)

`file_format`
Format of file (when `model_type= 'EVOL'`); one of

- `'AMDL'` : AMDL-format binary file
- `'B3'` : B3-format HDF5 file
- `'FAMDL'` : FAMDL-format text file
- `'FGONG'` : FGONG-format text file
- `'GSM'` : GSM-format HDF5 file
- `'LOSC'` : LOSC-format text file
- `'MESA'` : MESA GYRE-format text file
- `'OSC'` : OSC-format text file
- `'WDEC'` : WDEC-format text file

`data_format` (default `''`, indicates auto-select)
Fortran format specifier for data read from OSC-, FGONG- and FAMDL-format files

`deriv_type` (default `'MONO'`)
Cubic interpolation derivatives type (when `model_type='POLY'|'EVOL'`); one of

- `'NATURAL'` : Natural (spline) derivatives
- `'FINDIFF'` : Finite-difference derivatives
- `'MONO'` : Monotonized derivatives (default)

`Gamma_1` (default `5/3`)
First adiabatic exponent (when `model_type='HOM'`)

`grid_type` (default `'UNI'`)
Model grid type (when `model_type='HOM'`); one of

- `'UNI'` : Uniform spacing
- `'GEO'` : Geometric spacing
- `'LOG'` : Logarithmic spacing

`n` (default `10`)
Number of points in model grid (when `model_type='HOM'`)

`s` (default `1`)
Skewness parameter for model grid (when `model_type='HOM'` and `grid_type='GEO'|'LOG'`)

x_i (default 0)
Inner boundary coordinate of model grid (when `model_type='HOM'`)

x_o (default 1)
Outer boundary coordinate of model grid (when `model_type='HOM'`)

dx_snap (default 0)
Threshold for snapping model points together, when `model_type` is `'EVOL'`. If a pair of points are separated by less than `dx_snap`, they are snapped together.

add_center (default .TRUE.)
Flag to add a center point to the model (when `model_type='EVOL'|'POLY'`). If a point does not already exist at the origin, then one is added

repair_As (default .FALSE.)
Flag to repair inaccuracies in the dimensionless Brunt-Väisälä frequency at density discontinuities

13.4 Mode Parameters

The `&mode` namelist group defines mode parameters, as follows:

l (default 0)
Harmonic degree ℓ

m (default 0)
Azimuthal order m

n_pg_min (default -HUGE)
Filter for minimum radial order (applies only to adiabatic calculations)

n_pg_max (default +HUGE)
Filter for maximum radial order (applies only to adiabatic calculations)

tag
Tag for controlling selection of other parameters

13.5 Numerical Parameters

The `&num` namelist group defines numerical method parameters; the input file can contain one or more, but only the last (tag-matching) one is used. Allowable fields are:

diff_scheme (default 'COLLOC_GL2')
Difference equation scheme; one of:

- `'COLLOC_GL2'` : Second-order Gauss-Legendre collocation
- `'COLLOC_GL4'` : Fourth-order Gauss-Legendre collocation
- `'COLLOC_GL6'` : Sixth-order Gauss-Legendre collocation
- `'MAGNUS_GL2'` : Second-order Gauss-Legendre Magnus
- `'MAGNUS_GL4'` : Fourth-order Gauss-Legendre Magnus
- `'MAGNUS_GL6'` : Sixth-order Gauss-Legendre Magnus
- `'MIRK'` : Fourth-order mono-implicit Runge-Kutta (experimental)
- `'TRAPZ'` : Trapezoidal, with the prescription by Sugimoto (1970) for non-adiabatic cases

`r_root_solver` (default `'BRENT'`)
Root solver for real arithmetic; one of:

- `'BRENT'` : Brent's method

`c_root_solver` (default `'RIDDERS'`)
Root solver for complex arithmetic; one of

- `'RIDDERS'` : Complex Ridders' method
- `'SECANT'` : Secant method
- `'SIMPLEX'` : Simplex method

`n_iter_max` (default `50`)
Maximum number of iterations in root-finding algorithm

`matrix_type` (default `'BLOCK'`)
Storage type of system matrix; one of

- `'BAND'` : Band-structured
- `'BLOCK'` : Block-structured

`deflate_roots` (default `.TRUE.`)
Flag to use root deflation, which can avoid the same eigenfrequency being found multiple times

`restrict_roots` (default `.TRUE.`)
Flag to check each roots found lies within the bounds of the frequency scan

`ad_search` (default `'BRACKET'`)
Initial search method for adiabatic calculations; one of

- `'BRACKET'` : Bracket sign changes in the discriminant function

`nad_search` (default `'AD'`)
Initial search method for non-adiabatic calculations; one of

- `'AD'` : Use adiabatic eigenfrequencies
- `'MINMOD'` : Find minima in the modulus of the discriminant function, along the real-ω axis
- `'CONTOUR'` : Find intersections between real and imaginary zero-contours of the discriminant function

See the *Non-Adiabatic Oscillations* chapter for more details about these search methods.

`tag_list` (default `''`, which matches all)
Comma-separated list of `&mode` tags to match

13.6 Orbital Parameters

The `&orbit` namelist group defines orbital parameters, as follows:

`Omega_orb` (default `1`)
Orbital angular frequency

`Omega_orb_units` (default `'NULL'`)
Units of `Omega_orb`; one of:

- `'NONE'` : Dimensionless angular frequency
- `'HZ'` : Linear frequency in Hz[1]

[1] This option is available only for stellar models with *D capability*

- `'UHZ'` : Linear frequency in μHz[Page 73, 1]
- `'RAD_PER_SEC'` : Angular frequency in radians per second[Page 73, 1]
- `'CYC_PER_DAY'` : Linear frequency in cycles per day[Page 73, 1]
- `'CRITICAL'` : Fraction of the Roche critical rate[Page 73, 1]

`q` (default `1`)
Ratio of secondary mass to primary mass

`e` (default `0`)
Orbital eccentricity

`tag_list` (default `''`, which matches all)
Comma-separated list of `&tide` tags to match

13.7 Oscillation Parameters

The `&osc` namelist group defines oscillation parameters, as follows:

`inner_bound` (default `'REGULAR'`)
Inner boundary conditions; one of:

- `'REGULAR'` : Regularity-enforcing (only valid when inner grid point is at $x = 0$)
- `'ZERO_R'` : Zero radial displacement (only valid when inner grid point is at $x \neq 0$)
- `'ZERO_H'` : Zero horizontal displacement (only valid when inner grid point is at $x \neq 0$)

`outer_bound` (default `'VACUUM'`)
Outer boundary conditions; one of:

- `'VACUUM'` : Vanishing surface density
- `'DZIEM'` : Formulation following Dziembowski (1971)
- `'UNNO'` : Formulation following Unno et al. (1989)
- `'JCD'` : Formulation following Jørgen Christensen-Dalsgaard (ADIPLS)
- `'ISOTHERMAL'` : Formulation based on local dispersion analysis for isothermal atmosphere
- `'GAMMA'` : Vanishing displacement and derivative at outer boundary, intended for use with γ modes (isolated g modes; see Ong & Basu, 2020)

`outer_bound_cutoff` (default `''`)
Outer boundary conditions to use when evaluating cutoff frequencies (see `freq_units`); same options as `outer_bound`, and if left blank then takes its value from `outer_bound`

`outer_bound_branch` (default `'E_NEG'`)
Dispersion relation solution branch to use for outer boundary conditions (when `outer_bound='UNNO'|'JCD'|'ISOTHERMAL'`); one of

- `'E_NEG'` : Outward-decaying energy density
- `'E_POS'` : Outward-growing energy density
- `'F_NEG'` : Outward energy flux
- `'F_POS'` : Inward energy flux
- `'V_NEG'` : Outward phase velocity
- `'V_POS'` : Inward phase velocity

variables_set (default 'GYRE')
Dependent variables in oscillation equations; one of:

- 'GYRE' : GYRE formulation, as described in the *Dimensionless Formulation* section
- 'DZIEM' : Formulation following Dziembowski (1971)
- 'JCD' : Formulation following Jørgen Christensen-Dalsgaard (ADIPLS)
- 'MIX' : Mixed formulation ('JCD' for $y_{3,4}$, 'DZIEM' for $y_{1,2}$)
- 'LAGP' : Lagrangian pressure perturbation formulation

alpha_grv (default 1.)
Scaling factor for gravitational potential perturbations (see the $\alpha_{\rm grv}$ entry in the *Physics Switches* section)

alpha_thm (default 1.)
Scaling factor for the thermal timescale (see the $\alpha_{\rm thm}$ entry in the *Physics Switches* section)

alpha_hfl (default 1.)
Scaling factor for horizontal flux perturbations (see the $\alpha_{\rm hfl}$ entry in the *Physics Switches* section)

alpha_gam (default 1.)
Scaling factor for g-mode isolation (see the α_γ term in entry in the *Physics Switches* section)

alpha_pi (default 1.)
Scaling factor for p-mode isolation (see the α_π term in entry in the *Physics Switches* section)

alpha_kar (default 1.)
Scaling factor for opacity density partial derivative (see the $\alpha_{\kappa,\rho}$ entry in the *Physics Switches* section)

alpha_kat (default 1.)
Scaling factor for opacity temperature partial derivative (see the $\alpha_{\kappa,T}$ entry in the *Physics Switches* section)

alpha_rht (default 0.)
Scaling factor for time-dependent term in radiative heat equation (see the $\alpha_{\rm rht}$ entry in the *Physics Switches* section)

alpha_trb (default 0.)
Scaling factor for the turbulent mixing length (see the $\alpha_{\rm trb}$ entry in the *Physics Switches* section)

inertia_norm (default 'BOTH')
Inertia normalization factor; one of

- 'RADIAL' : Radial amplitude squared, $|\xi_{\rm r}|^2$, evaluated at x_ref
- 'HORIZ' : Horizontal amplitude squared, $|\lambda||\xi_{\rm h}|^2$, evaluated at x_ref
- 'BOTH' : Overall amplitude squared, $|\xi_{\rm r}|^2 + |\lambda||\xi_{\rm h}|^2$, evaluated at x_ref

time_factor (default 'OSC')
Time-dependence factor in pulsation equations; one of:

- 'OSC' : Oscillatory, $\propto \exp(-{\rm i}\sigma t)$
- 'EXP' : Exponential, $\propto \exp(-\sigma t)$

conv_scheme (default 'FROZEN_PESNELL_1')
Scheme for treating convection; one of:

- 'FROZEN_PESNELL_1' : Freeze convective heating altogether; case 1 described by Pesnell (1990)
- 'FROZEN_PESNELL_4' : Freeze Lagrangian perturbation of convective luminosity; case 4 described by Pesnell (1990)

`zeta_scheme` (default `'KAWALER'`)
Scheme for evaluating dimensionless frequency weight function $\mathrm{d}\zeta/\mathrm{d}x$ and integral eigenfrequency ω_{int}; one of:

- `'PESNELL'` : Evaluate using eqn. (A5) of Pesnell (1987)
- `'KAWALER'` : Evaluate using eqn. (7) of Kawaler et al. (1985), as corrected by Townsend & Kawaler (2023)
- `'KAWALER_GRAV'`: Evaluate using the g-mode part in eqn. (7) of Kawaler et al. (1985)
- `'DUPRET'` : Evaluate using eqn. (1.71) of Dupret (2002, PhD thesis)

`deps_scheme` (default `'MODEL'`)
Scheme for calculating nuclear energy generation partials $\epsilon_{\mathrm{n},\rho}$ and $\epsilon_{\mathrm{n},T}$; one of:

- `'MODEL'` : Use values from model
- `'FILE'` : Use complex (phase-lagged) values from separate file

`deps_file` (default `''`)
Name of epsilon partial derivatives file (when `deps_scheme='FILE'`)

`deps_file_format` (default `'WOLF'`)
Format of epsilon partial derivative file (when `deps_scheme='FILE'`); one of:

- `'WOLF'` : Format used in preparation of Wolf et al. (2018)

`x_ref` (default 1 or outer grid point, whichever is smaller)
Reference fractional radius for photosphere, normalizations etc.

`x_atm` (default `-1`, implying outer grid point)
Fractional radius for convection-zone crossover point of π/γ modes (isolated p and g modes; see Ong & Basu, 2020)

`adiabatic` (default `.TRUE.`)
Flag to perform adiabatic calculations

`nonadiabatic` (default `.FALSE.`)
Flag to perform non-adiabatic calculations

`quasiad_eigfuncs` (default `.FALSE.`)
Flag to calculate quasi-adiabatic entropy/luminosity eigenfunctions during adiabatic calculations

`reduce_order` (default `.TRUE.`)
Flag to reduce the order of the *adiabatic* radial-pulsation equations from 4 to 2

`tag_list` (default `''`, which matches all)
Comma-separated list of `&mode` tags to match

13.8 Output Parameters

The `&ad_output`, `&nad_output` and `&tides_output` namelist groups determine the output produced at the end of a run (the first two for the adiabatic and non-adiabatic calculation stages of **`gyre`**; the third for **`gyre_tides`**). Parameters are as follows:

`summary_file` (default `''`)
Name of summary file

`summary_file_format` (default `'HDF'`)
Format of summary file; one of

- `'HDF'` : HDF5 file

- `'TXT'` : Text file

`summary_item_list` (default `'l,n_pg,omega,freq'`)
Comma-separated list of output items to write to summary file; see the *Summary Files* section for possible choices

`summary_filter_list` (default `''`)
Comma-separated list of filter criteria for summary files; see the *Output Filters* section for possible choices

`detail_template` (default `''`)
Name template of detail files. Names are generated using the following pattern substitutions:

- `'%ID'` : Unique mode index, formatted in fixed-width field
- `'%id'` : Same as `'%ID'`, but formatted in variable-width field
- `'%L'` : Harmonic degree ℓ, formatted in fixed-width field
- `'%l'` : Same as `'%L'`, but formatted in variable-width field
- `'%M'` : Azimuthal order m, formatted in fixed-width field
- `'%m'` : Same as `'%M'`, but formatted in variable-width field
- `'%N'` : Radial order $n_{\rm pg}$, formatted in fixed-width field
- `'%n'` : Same as `'%N'`, but formatted in variable-width field
- `'%P'` : Acoustic wave winding number $n_{\rm p}$, formatted in fixed-width field
- `'%p'` : Same as `'%P'`, but formatted in variable-width field
- `'%G'` : Gravity wave winding number $n_{\rm g}$, formatted in fixed-width field
- `'%g'` : Same as `'%G'`, but formatted in variable-width field

`detail_file_format` (default `'HDF'`)
Format of detail files; one of

- `'HDF'` : HDF5 file
- `'TXT'` : Text file

`detail_item_list` (default `'l,n_pg,omega,freq,x,xi_r,xi_h'`)
Comma-separated list of output items to write to detail files; see the *Detail Files* section for possible choices

`detail_filter_list` (default `''`)
Comma-separated list of filter criteria for detail files; see the *Output Filters* section for possible choices

`freq_units` (default `NONE`)
Units of `freq` output item; one of:

- `'NONE'` : Dimensionless angular frequency
- `'HZ'` : Linear frequency in Hz[1]
- `'UHZ'` : Linear frequency in μHz[Page 77, 1]
- `'RAD_PER_SEC'` : Angular frequency in radians per second[1]
- `'CYC_PER_DAY'` : Linear frequency in cycles per day[1]
- `'ACOUSTIC_DELTA'` : Fraction of the asymptotic acoustic large frequency separation $\Delta\nu$
- `'GRAVITY_DELTA'` : Fraction of the asymptotic inverse gravity period separation $(\Delta P)^{-1}$
- `'UPPER_DELTA'` : Greater of $\Delta\nu$ and $(\Delta P)^{-1}$

[1] This option is available only for stellar models with *D capability*

- `'LOWER_DELTA'` : Lesser of $\Delta\nu$ and $(\Delta P)^{-1}$
- `'ACOUSTIC_CUTOFF'` : Fraction of the acoustic cutoff frequency[1]
- `'GRAVITY_CUTOFF'` : Fraction of the gravity cutoff frequency[Page 77, 1]
- `'ROSSBY_I'` : Fraction of Rossby frequency at inner boundary
- `'ROSSBY_O'` : Fraction of Rossby frequency at outer boundary

`freq_frame` (default `INERTIAL`)
Frame of `freq` output item; one of:

- `'INERTIAL'` : Inertial frame
- `'COROT_I'` : Co-rotating frame at inner boundary
- `'COROT_O'` : Co-rotating frame at outer boundary

`label` (default `''`)
Textual label to add to all output files

13.9 Rotation Parameters

The `&rot` namelist group defines rotational parameters, as follows:

`coriolis_method` (default `'NULL'`)
Method used to treat the Coriolis force; one of:

- `'NULL'` : Neglect the Coriolis force
- `'TAR'` : Use the traditional approximation of rotation

`Omega_rot_source` (default `'MODEL'`)
Source for rotational angular frequency $\Omega_{\rm rot}$; one of:

- `'MODEL'` : Differential rotation, with a spatially varying $\Omega_{\rm rot}$ obtained from the stellar model
- `'UNIFORM'` : Uniform rotation, with a spatially constant $\Omega_{\rm rot}$ set by the `Omega_rot` and `Omega_rot_units` parameters

`Omega_rot` (default `0`)
Rotation angular frequency (when `Omega_rot_source='UNIFORM'`)

`Omega_rot_units` (default `'NULL'`)
Units of `Omega_rot` (when `Omega_rot_source='UNIFORM'`); one of:

- `'NONE'` : Dimensionless angular frequency
- `'HZ'` : Linear frequency in Hz[1]
- `'UHZ'` : Linear frequency in μHz[1]
- `'RAD_PER_SEC'` : Angular frequency in radians per second[1]
- `'CYC_PER_DAY'` : Linear frequency in cycles per day[1]
- `'CRITICAL'` : Fraction of the Roche critical rate[1]

`rossby` (default `.FALSE.`)
Flag to use Rossby solution family in TAR (when `coriolis_method='TAR'`)

[1] This option is available only for stellar models with *D capability*

complex_lambda (default .FALSE.)
Flag to use complex arithmetic when evaluating the TAR angular eigenvalue λ (when `coriolis_method='TAR'`)

tag_list (default '', which matches all)
Comma-separated list of `&mode` tags to match

13.10 Frequency Scan Parameters

The `&scan` namelist group defines frequency grid parameters, as follows:

grid_type (default 'LINEAR')
Distribution of frequency points; one of:

- `'LINEAR'` : Uniform in frequency
- `'INVERSE'` : Uniform in inverse frequency
- `'FILE'` : Read from file

grid_frame (default 'INERTIAL')
Reference frame in which `grid_type` applies; one of:

- `'INERTIAL'` : Inertial frame
- `'COROT_I'` : Co-rotating frame at inner boundary
- `'COROT_O'` : Co-rotating frame at outer boundary

freq_min (default 1)
Minimum frequency, when `grid_type` is `'LINEAR'` or `'INVERSE'`

freq_max (default 10)
Maximum frequency, when `grid_type` is `'LINEAR'` or `'INVERSE'`

n_freq (default 10)
Number of frequency points, when `grid_type` is `'LINEAR'` or `'INVERSE'`

freq_units (default NONE)
Units of `freq_min` and `freq_max`, when `grid_type` is `'LINEAR'` or `'INVERSE'`; units of read frequencies when `grid_type` is `'FILE'`

- `'NONE'` : Dimensionless angular frequency
- `'HZ'` : Linear frequency in Hz[1]
- `'UHZ'` : Linear frequency in μHzPage 79, 1
- `'RAD_PER_SEC'` : Angular frequency in radians per second[1]
- `'CYC_PER_DAY'` : Linear frequency in cycles per day[1]
- `'ACOUSTIC_DELTA'` : Fraction of the asymptotic acoustic large frequency separation $\Delta\nu$
- `'GRAVITY_DELTA'` : Fraction of the asymptotic inverse gravity period separation $(\Delta P)^{-1}$
- `'UPPER_DELTA'` : Greater of $\Delta\nu$ and $(\Delta P)^{-1}$
- `'LOWER_DELTA'` : Lesser of $\Delta\nu$ and $(\Delta P)^{-1}$
- `'ACOUSTIC_CUTOFF'` : fraction of the acoustic cutoff frequency[1]
- `'GRAVITY_CUTOFF'` : fraction of the gravity cutoff frequency[1]

[1] This option is available only for stellar models with *D capability*

- `'ROSSBY_I'` : fraction of Rossby frequency (see eqn. 16.11) at inner boundary
- `'ROSSBY_O'` : fraction of Rossby frequency (see eqn. 16.11) at outer boundary

`freq_min_units` (default `''`)
Units of `freq_min`; same options as `freq_units` and overrides it if set

`freq_max_units` (default `''`)
Units of `freq_max`; same options as `freq_units` and overrides it if set

`freq_frame` (default `'INERTIAL'`)
Reference frame in which `freq_min` and `freq_max` are defined, when `grid_type` is `'LINEAR'` or `'INVERSE'`; one of:

- `'INERTIAL'` : Inertial frame
- `'COROT_I'` : Co-rotating frame at inner boundary
- `'COROT_O'` : Co-rotating frame at outer boundary

`file`
File to read frequencies from, when `grid_type` is `'FILE'`

`axis` (default `'REAL'`)
Axis that scan applies to; one of

- `'REAL'` : Real axis
- `'IMAG'` : Imaginary axis

`tag_list` (default `''`, which matches all)
Comma-separated list of `&mode` tags to match

An input file can contain one or more `&scan` namelist group; the points defined by each (tag-matching) group are merged together to build the frequency grid. See the *Frequency Grids* section for further details.

13.11 Tidal Parameters

The `&tide` namelist group defines tidal parameters, as follows:

`y_T_thresh_abs` (default `0.`)
Absolute threshold on dimensionless tidal potential y_{T} for a partial tide to contribute

`y_T_thresh_rel` (default `0.`)
Relative threshold on dimensionless tidal potential y_{T} for a partial tide to contribute

`omega_c_thresh` (default `0.`)
Threshold on dimensionless co-rotating frequency ω_{c} for a partial tide to be treated as dynamic (rather than static)

`alpha_frq` (default `1.`)
Scaling parameter α_{frq} for tidal forcing frequency

`l_min` (default `2`)
Minimum harmonic degree ℓ in spatial expansion of tidal potential

`l_max` (default `2`)
Maximum harmonic degree ℓ in spatial expansion of tidal potential

`m_min` (default `-HUGE`)
Minimum azimuthal order m in spatial expansion of tidal potential

`m_max` (default `HUGE`)
Maximum azimuthal order m in spatial expansion of tidal potential

`k_min` (default `-10`)
Minimum orbital harmonic k in temporal expansion of tidal potential

`k_max` (default `10`)
Maximum orbital harmonic k in temporal expansion of tidal potential

`tag`
Tag for controlling selection of other parameters

CHAPTER

FOURTEEN

OUTPUT FILES

This chapter describes the *summary* and *detail* output files written by the GYRE *frontends*.

14.1 Summary Files

Summary files collect together global properties, such as eigenfrequencies and radial orders, of all solutions (modes, tidal responses, etc.) found during a run. The specific data written to a summary file are controlled by the `summary_item_list` parameters of the `&ad_output` and `&nad_output` namelist groups (**`gyre`** adiabatic and non-adiabatic calculations, respectively) and the `&tides_output` namelist group (**`gyre_tides`** calculations). These parameters specify the items to be written, via a comma-separated list.

The following subsections describe the items that may appear in `summary_item_list`, grouped together by functional area. For each item, the corresponding math symbol is given (if there is one), together with the datatype, and a brief description. Units (where applicable) are indicated in brackets [].

14.1.1 Solution Data

Item	Symbol	Datatype	Description
`n_row`	$N_{\rm row}$	integer	number of rows in summary file, each corresponding to a mode found (**`gyre`**) or a tidal response evaluated (**`gyre_tides`**)
`n`	N	integer(`n_row`)	number of spatial grid points
`omega`	ω	complex(`n_row`)	dimensionless eigenfrequency

14.1.2 Observables

Item	Symbol	Datatype	Description
`freq`	—	complex(`n_row`)	dimensioned frequency; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`freq_units`	—	string	`freq_units` parameter
`freq_frame`	—	string	`freq_frame` parameter
`f_T`	f_T	real(`n_row`)	Effective temperature perturbation amplitude; evaluated using eqn. 5 of Dupret et al. (2003)
`f_g`	f_g	real(`n_row`)	Effective gravity perturbation amplitude; evaluated using eqn. 6 of Dupret et al. (2003)
`psi_T`	ψ_T	real(`n_row`)	Effective temperature perturbation phase; evaluated using eqn. 5 of Dupret et al. (2003)
`psi_g`	ψ_g	real(`n_row`)	Effective gravity perturbation phase; evaluated using eqn. 6 of Dupret et al. (2003)

14.1.3 Classification & Validation

Item	Symbol	Datatype	Description
`id`	—	integer(`n_row`)	unique mode index
`l`	ℓ	integer(`n_row`)	harmonic degree
`l_i`	ℓ_i	complex(`n_row`)	effective harmonic degree at inner boundary
`m`	m	integer(`n_row`)	azimuthal order
`n_p`	n_p	integer(`n_row`)	acoustic-wave winding number
`n_g`	n_g	integer(`n_row`)	gravity-wave winding number
`n_pg`	n_pg	integer(`n_row`)	radial order within the Eckart-Scuflaire-Osaki-Takata scheme (see Takata, 2006b)
`omega_int`	ω_int	complex(`n_row`)	dimensionless eigenfrequency; evaluated as $\omega_\mathrm{int} = \sqrt{\zeta/E}$
`zeta`	ζ	complex(`n_row`)	integral of $\mathrm{d}\zeta/\mathrm{d}x$ with respect to x

14.1.4 Perturbations

Item	Symbol	Datatype	Description
`x_ref`	$x_{\rm ref}$	real	fractional radius of reference location
`xi_r_ref`	$\tilde{\xi}_{r,\rm ref}$	complex(`n_row`)	radial displacement perturbation at reference location $[R]$
`eul_Phi_ref`	$\tilde{\Phi}'_{\rm ref}$	complex(`n_row`)	Eulerian potential perturbation at reference location $[GM/R]$
`deul_Phi_ref`	$(\mathrm{d}\tilde{\Phi}'/\mathrm{d}x)_{\rm ref}$	complex(`n_row`)	Eulerian potential gradient perturbation at reference location $[GM/R^2]$
`lag_S_ref`	$\delta\tilde{S}_{\rm ref}$	complex(`n_row`)	Lagrangian specific entropy perturbation at reference location $[R]$
`lag_L_ref`	$\delta\tilde{L}_{\rm R,ref}$	complex(`n_row`)	Lagrangian radiative luminosity perturbation at reference location $[L]$

14.1.5 Energetics & Transport

Item	Symbol	Datatype	Description
`eta`[1]	η	real(`n_row`)	normalized growth rate η; evaluated using expression in text of page 1186 of Stellingwerf (1978)
`E`	E	real(`n_row`)	mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$
`E_p`	$E_{\rm p}$	real(`n_row`)	acoustic mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$ where $\varpi = 1$
`E_g`	$E_{\rm g}$	real(`n_row`)	gravity mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$ in regions where $\varpi = -1$
`E_norm`	$E_{\rm norm}$	real(`n_row`)	normalized inertia; evaluation controlled by `inertia_norm` parameter
`E_ratio`	—	real(`n_row`)	ratio of mode inertia outside reference location, to total inertia
`H`	H	real(`n_row`)	mode energy $[GM^2/R]$; evaluated as $\frac{1}{2}\omega^2 E$
`W`[Page 85, 1]	W	real(`n_row`)	mode work $[GM^2/R]$; evaluated by integrating $\mathrm{d}W/\mathrm{d}x$
`W_eps`[1]	W_ϵ	real(`n_row`)	mode work $[GM^2/R]$; evaluated by integrating $\mathrm{d}W_\epsilon/\mathrm{d}x$
`tau_ss`	$\tau_{\rm ss}$	real(`n_row`)	steady-state torque $[GM^2/R]$; evaluated by integrating $\mathrm{d}\tau_{\rm ss}/\mathrm{d}x$
`tau_tr`	$\tau_{\rm tr}$	real(`n_row`)	steady-state torque $[GM^2/R]$; evaluated by integrating $\mathrm{d}\tau_{\rm tr}/\mathrm{d}x$

14.1.6 Rotation

Item	Symbol	Datatype	Description
`Omega_rot_ref`	$\Omega_{\rm rot,ref}$	real(`n_row`)	rotation angular frequency at reference location $[\sqrt{GM/R^3}]$
`domega_rot`	$\Delta\omega$	real(`n_row`)	dimensionless first-order rotational splitting; evaluated using eqn. 3.355 of Aerts et al. (2010)
`dfreq_rot`	—	real(`n_row`)	dimensioned first-order rotational splitting; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`beta`	β	real(`n_row`)	rotation splitting coefficient; evaluated by integrating $\mathrm{d}\beta/\mathrm{d}x$

[1] This item is available only for stellar models with *N capability*

14.1.7 Stellar Structure

Item	Symbol	Datatype	Description
`M_star`[2]	M	real(`n_row`)	stellar mass [g]
`R_star`[Page 86, 2]	R	real(`n_row`)	stellar radius [cm]
`L_star`[2]	L	real(`n_row`)	stellar luminosity [$\mathrm{erg\,s^{-1}}$]
`Delta_p`	$\Delta\nu$	real(`n_row`)	asymptotic p-mode large frequency separation [$\sqrt{GM/R^3}$]
`Delta_g`	$(\Delta P)^{-1}$	real(`n_row`)	asymptotic g-mode inverse period separation [$\sqrt{GM/R^3}$]

14.1.8 Tidal Response

Note that these items are available only when using **`gyre_tides`**.

Item	Symbol	Datatype	Description
`k`	k	integer(`n_row`)	Fourier harmonic
`eul_Psi_ref`	$\tilde{\Psi}'_{\mathrm{ref}}$	complex(`n_row`)	Eulerian total potential perturbation at reference location [GM/R]
`Phi_T_ref`	$\tilde{\Phi}_{\mathrm{T,ref}}$	real(`n_row`)	tidal potential at reference location [GM/R]
`Omega_orb`	Ω_{orb}	real(`n_row`)	orbital angular frequency; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`q`	q	real(`n_row`)	ratio of secondary mass to primary mass
`e`	e	real(`n_row`)	orbital eccentricity
`R_a`	R/a	real(`n_row`)	ratio of primary radius to orbital semi-major axis
`cbar`	$\overline{c}_{\ell,m,k}$	real(`n_row`)	tidal expansion coefficient; see eqn. A1 of Sun et al. (2023)
`Gbar_1`	$\overline{G}^{(1)}_{\ell,m,k}$	real(`n_row`)	secular orbital evolution coefficient; equivalent to $G^{(1)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_2`	$\overline{G}^{(2)}_{\ell,m,k}$	real(`n_row`)	secular orbital evolution coefficient; equivalent to $G^{(2)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_3`	$\overline{G}^{(3)}_{\ell,m,k}$	real(`n_row`)	secular orbital evolution coefficient; equivalent to $G^{(3)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_4`	$\overline{G}^{(4)}_{\ell,m,k}$	real(`n_row`)	secular orbital evolution coefficient; equivalent to $G^{(4)}_{\ell,m,-k}$ (see Willems et al., 2003)

14.2 Detail Files

Detail files store spatial quantities, such as eigenfunctions and differential inertias, for an individual solution (mode, tidal response, etc.) found during a run. The specific data written to detail files are controlled by the `detail_item_list` parameters of the `&ad_output` and `&nad_output` namelist groups (**`gyre`** adiabatic and non-adiabatic calculations, respectively) and the `&tides_output` namelist group (**`gyre_tides`** calculations). These parameters specify the items to be written, via a comma-separated list.

The following subsections describe the items that may appear in `detail_item_list`, grouped together by functional area. For each item, the corresponding math symbol is given (if there is one), together with the datatype, and a brief description. Units (where applicable) are indicated in brackets [].

[2] This item is available only for stellar models with *D capability*

14.2.1 Solution Data

Item	Symbol	Datatype	Description
`n`	N	integer	number of spatial grid points
`omega`	ω	complex	dimensionless eigenfrequency (**`gyre`**) or forcing frequency (**`gyre_tides`**)
`x`	x	real(`n`)	independent variable $x = r/R$
`dx_min`	$\Delta x_{\rm min}$	real	minimum spacing of spatial grid
`dx_max`	$\Delta x_{\rm max}$	real	maximum spacing of spatial grid
`dx_rms`	$\Delta x_{\rm rms}$	real	root-mean-square spacing of spatial grid
`x_ref`	$x_{\rm ref}$	real	fractional radius of reference location
`y_1`	y_1	complex(`n`)	dependent variable
`y_2`	y_2	complex(`n`)	dependent variable
`y_3`	y_3	complex(`n`)	dependent variable
`y_4`	y_4	complex(`n`)	dependent variable
`y_5`	y_5	complex(`n`)	dependent variable
`y_6`	y_6	complex(`n`)	dependent variable

The definitions of the dependent variables $\{y_1, \ldots, y_6\}$ are provided in the *Oscillation Equations* chapter.

14.2.2 Observables

Item	Symbol	Datatype	Description
`freq`	—	complex	dimensioned frequency; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`freq_units`	—	string	`freq_units` parameter
`freq_frame`	—	string	`freq_frame` parameter
`f_T`	f_T	real	Effective temperature perturbation amplitude; evaluated using eqn. 5 of Dupret et al. (2003)
`f_g`	$f_{\rm g}$	real	Effective gravity perturbation amplitude; evaluated using eqn. 6 of Dupret et al. (2003)
`psi_T`	ψ_T	real	Effective temperature perturbation phase; evaluated using eqn. 5 of Dupret et al. (2003)
`f_g`	$\psi_{\rm g}$	real	Effective gravity perturbation phase; evaluated using eqn. 6 of Dupret et al. (2003)

14.2.3 Classification & Validation

Item	Symbol	Datatype	Description
`id`	—	integer	unique mode index
`l`	ℓ	integer	harmonic degree
`l_i`	ℓ_i	complex	effective harmonic degree at inner boundary
`m`	m	integer	azimuthal order
`n_p`	n_p	integer	acoustic-wave winding number
`n_g`	n_g	integer	gravity-wave winding number
`n_pg`	n_pg	integer	radial order within the Eckart-Scuflaire-Osaki-Takata scheme (see Takata, 2006b)
`omega_int`	ω_int	complex	dimensionless eigenfrequency; evaluated as $\omega_\mathrm{int} = \sqrt{\zeta/E}$
`dzeta_dx`	$\mathrm{d}\zeta/\mathrm{d}x$	complex(`n`)	dimensionless frequency weight function; controlled by `zeta_scheme` parameter
`zeta`	ζ	complex	integral of $\mathrm{d}\zeta/\mathrm{d}x$ with respect to x
`Yt_1`	$\mathcal{Y}_1$	complex(`n`)	primary eigenfunction for Takata classification; evaluated using a rescaled eqn. 69 of Takata (2006b)
`Yt_2`	$\mathcal{Y}_2$	complex(`n`)	secondary eigenfunction for Takata classification; evaluated using a rescaled eqn. 70 of Takata (2006b)
`I_0`	I_0	complex(`n`)	first integral for radial modes; evaluated using eqn. 42 of Takata (2006a)
`I_1`	I_1	complex(`n`)	first integral for dipole modes; evaluated using eqn. 43 of Takata (2006a)
`prop_type`	ϖ	integer(`n`)	propagation type; $\varpi = 1$ in acoustic-wave regions, $\varpi = -1$ in gravity-wave regions, and $\varpi = 0$ in evanescent regions

14.2.4 Perturbations

Item	Symbol	Datatype	Description
`xi_r_ref`	$\tilde{\xi}_{r,\mathrm{ref}}$	complex	radial displacement perturbation at reference location $[R]$
`xi_h_ref`	$\tilde{\xi}_{\mathrm{h,ref}}$	complex	radial displacement perturbation at reference location $[R]$
`eul_Phi_ref`	$\tilde{\Phi}'_{\mathrm{ref}}$	complex	Eulerian potential perturbation at reference location $[GM/R]$
`deul_Phi_ref`	$(\mathrm{d}\tilde{\Phi}'/\mathrm{d}x)_{\mathrm{ref}}$	complex	Eulerian potential gradient perturbation at reference location $[GM/R^2]$
`lag_S_ref`	$\delta\tilde{S}_{\mathrm{ref}}$	complex	Lagrangian specific entropy perturbation at reference location $[R]$
`lag_L_ref`	$\delta\tilde{L}_{\mathrm{R,ref}}$	complex	Lagrangian radiative luminosity perturbation at reference location $[L]$
`xi_r`	$\tilde{\xi}_r$	complex(n)	radial displacement perturbation $[R]$
`xi_h`	$\tilde{\xi}_{\mathrm{h}}$	complex(n)	radial displacement perturbation $[R]$
`eul_Phi`	$\tilde{\Phi}'$	complex(n)	Eulerian potential perturbation $[GM/R]$
`deul_Phi`	$\mathrm{d}\tilde{\Phi}'/\mathrm{d}x$	complex(n)	Eulerian potential gradient perturbation $[GM/R^2]$
`lag_S`	$\delta\tilde{S}$	complex(n)	Lagrangian specific entropy perturbation $[c_P]$
`lag_L`	$\delta\tilde{L}_{\mathrm{R}}$	complex(n)	Lagrangian radiative luminosity perturbation $[L]$
`eul_P`	$\tilde{P}'$	complex(n)	Eulerian total pressure perturbation $[P]$
`eul_rho`	$\tilde{\rho}'$	complex(n)	Eulerian density perturbation $[\rho]$
`eul_T`	$\tilde{T}'$	complex(n)	Eulerian temperature perturbation $[T]$
`lag_P`	$\delta\tilde{P}$	complex(n)	Lagrangian total pressure perturbation $[P]$
`lag_rho`	$\delta\tilde{\rho}$	complex(n)	Lagrangian density perturbation $[\rho]$
`lag_T`	$\delta\tilde{T}$	complex(n)	Lagrangian temperature perturbation $[T]$

14.2.5 Energetics & Transport

Item	Symbol	Datatype	Description
`eta`	η	real	normalized growth rate η; evaluated using expression in text of page 1186 of Stellingwerf (1978)
`E`	E	real	mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$
`E_p`	E_p	real	acoustic mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$ where $\varpi = 1$
`E_g`	E_g	real	gravity mode inertia $[MR^2]$; evaluated by integrating $\mathrm{d}E/\mathrm{d}x$ in regions where $\varpi = -1$
`E_norm`	E_norm	real	normalized inertia; evaluation controlled by `inertia_norm` parameter
`E_ratio`		real	ratio of mode inertia outside reference location, to total inertia
`H`	H	real	mode energy $[GM^2/R]$; evaluated as $\frac{1}{2}\omega^2 E$
`W`	W	real	mode work $[GM^2/R]$; evaluated by integrating $\mathrm{d}W/\mathrm{d}x$
`W_eps`	W_ϵ	real	mode work $[GM^2/R]$; evaluated by integrating $\mathrm{d}W_\epsilon/\mathrm{d}x$
`tau_ss`	τ_ss	real	steady-state torque $[GM^2/R]$; evaluated by integrating $\mathrm{d}\tau_\mathrm{ss}/\mathrm{d}x$
`tau_tr`	τ_tr	real	steady-state torque $[GM^2/R]$; evaluated by integrating $\mathrm{d}\tau_\mathrm{tr}/\mathrm{d}x$
`dE_dx`	$\mathrm{d}E/\mathrm{d}x$	real(n)	differential inertia $[MR^2]$; evaluated using eqn. 3.139 of Aerts et al. (2010)
`dW_dx`[1]	$\mathrm{d}W/\mathrm{d}x$	real(n)	differential work $[GM^2/R]$; evaluated using eqn. 25.9 of Unno et al. (1989)
`dW_eps_dx`[Page 90,1]	$\mathrm{d}W_\epsilon/\mathrm{d}x$	real(n)	differential nuclear work $[GM^2/R]$; evaluated using eqn. 25.9 of Unno et al. (1989)
`dtau_ss_dx`	$\mathrm{d}\tau_\mathrm{ss}/\mathrm{d}x$	real(n)	steady-state differential torque [*G M^{2}/R*]
`dtau_tr_dx`	$\mathrm{d}\tau_\mathrm{tr}/\mathrm{d}x$	real(n)	transient differential torque [*G M^{2}/R*]
`alpha_0`	α_0	real(n)	excitation coefficient; evaluated using eqn. 26.10 of Unno et al. (1989)
`alpha_1`	α_1	real(n)	excitation coefficient; evaluated using eqn. 26.12 of Unno et al. (1989)

14.2.6 Rotation

Item	Symbol	Datatype	Description
`Omega_rot_ref`	$\Omega_\mathrm{rot,ref}$	real	rotation angular frequency at reference location[$\sqrt{GM/R^3}$]
`Omega_rot`	Ω_rot	real(n)	rotation angular frequency [$\sqrt{GM/R^3}$]
`domega_rot`	$\Delta\omega$	real	dimensionless first-order rotational splitting; evaluated using eqn. 3.355 of Aerts et al. (2010)
`dfreq_rot`	—	real	dimensioned first-order rotational splitting; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`beta`	β	real	rotation splitting coefficient; evaluated by integrating $\mathrm{d}\beta/\mathrm{d}x$
`dbeta_dx`	$\mathrm{d}\beta/\mathrm{d}x$	complex(n)	unnormalized rotation splitting kernel; evaluated using eqn. 3.357 of Aerts et al. (2010)
`lambda`	λ	complex(n)	tidal equation eigenvalue

[1] This option is available only for stellar models with *N capability*

14.2.7 Stellar Structure

Item	Symbol	Datatype	Description
`M_star`[2]	M	real	stellar mass [g]
`R_star`[Page 91, 2]	R	real	stellar radius [cm]
`L_star`[2]	L	real	stellar luminosity [$\mathrm{erg\,s^{-1}}$]
`Delta_p`	$\Delta\nu$	real	asymptotic p-mode large frequency separation [$\sqrt{GM/R^3}$]
`Delta_g`	$(\Delta P)^{-1}$	real	asymptotic g-mode inverse period separation [$\sqrt{GM/R^3}$]
`V_2`	V_2	real(n)	structure coefficient; defined in *Structure Coefficients* section
`As`	A^*	real(n)	structure coefficient; defined in *Structure Coefficients* section
`U`	U	real(n)	structure coefficient; defined in *Structure Coefficients* section
`c_1`	c_1	real(n)	structure coefficient; defined in *Structure Coefficients* section
`Gamma_1`	Γ_1	real(n)	adiabatic exponent; defined in *Linearized Equations* section
`nabla`[Page 90, 1]	∇	real(n)	temperature gradient; defined in *Structure Coefficients* section *Dimensionless Formulation* section
`nabla_ad`[Page 90, 1]	∇_{ad}	real(n)	adiabatic temperature gradient; defined in *Linearized Equations* section
`dnabla_ad`[Page 90, 2]	$\partial\nabla_{\mathrm{ad}}$	real(n)	derivative of adiabatic temperature gradient
`upsilon_T`[Page 90, 1]	υ_T	real(n)	thermodynamic coefficient; defined in *Linearized Equations* section
`c_lum`[Page 90, 1]	c_{lum}	real(n)	structure coefficient; defined in *Structure Coefficients* section
`c_rad`[Page 90, 1]	c_{rad}	real(n)	structure coefficient; defined in *Structure Coefficients* section
`c_thn`[Page 90, 1]	c_{thn}	real(n)	structure coefficient; defined in *Structure Coefficients* section
`c_thk`[Page 90, 1]	c_{thk}	real(n)	structure coefficient; defined in *Structure Coefficients* section
`c_eps`[Page 90, 1]	c_ϵ	real(n)	structure coefficient; defined in *Structure Coefficients* section
`kap_rho`[Page 90, 1]	κ_ρ	real(n)	opacity partial; defined in *Linearized Equations* section
`kap_T`[Page 90, 1]	κ_T	real(n)	opacity partial; defined in *Linearized Equations* section
`eps_rho`[Page 90, 1]	$\epsilon_{\mathrm{n},\rho}$	real(n)	nuclear energy generation partial; defined in *Linearized Equations* section
`eps_T`[Page 90, 1]	$\epsilon_{\mathrm{n},T}$	real(n)	nuclear energy generation partial; defined in *Linearized Equations* section
`M_r`[2]	M_r	real(n)	interior mass [g]
`P`[2]	P	real(n)	total pressure [Ba]
`rho`[2]	ρ	real(n)	density [$\mathrm{g\,cm^{-3}}$]
`T`[2]	T	real(n)	temperature [K]

14.2.8 Tidal Response

Note that these items are available only when using **`gyre_tides`**.

[2] This option is available only for stellar models with *D capability*

Item	Symbol	Datatype	Description
`k`	k	integer	Fourier harmonic
`eul_Psi_ref`	$\tilde{\Psi}'_{\rm ref}$	complex	Eulerian total potential perturbation at reference location $[GM/R]$
`Phi_T_ref`	$\tilde{\Phi}_{\rm T,ref}$	real	tidal potential at reference location $[GM/R]$
`eul_Psi`	$\tilde{\Psi}'$	complex(`n`)	Eulerian total potential perturbation $[GM/R]$
`Phi_T`	$\tilde{\Phi}_{\rm T}$	real(`n`)	tidal potential $[GM/R]$
`Omega_orb`	$\Omega_{\rm orb}$	real	orbital angular frequency; units and reference frame controlled by `freq_units` and `freq_frame` parameters
`q`	q	real	ratio of secondary mass to primary mass
`e`	e	real	orbital eccentricity
`R_a`	R/a	real	ratio of primary radius to orbital semi-major axis
`cbar`	$\overline{c}_{\ell,m,k}$	real	tidal expansion coefficient; see eqn. A1 of Sun et al. (2023)
`Gbar_1`	$\overline{G}^{(1)}_{\ell,m,k}$	real	secular orbital evolution coefficient; equivalent to $G^{(1)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_2`	$\overline{G}^{(2)}_{\ell,m,k}$	real	secular orbital evolution coefficient; equivalent to $G^{(2)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_3`	$\overline{G}^{(3)}_{\ell,m,k}$	real	secular orbital evolution coefficient; equivalent to $G^{(3)}_{\ell,m,-k}$ (see Willems et al., 2003)
`Gbar_4`	$\overline{G}^{(4)}_{\ell,m,k}$	real	secular orbital evolution coefficient; equivalent to $G^{(4)}_{\ell,m,-k}$ (see Willems et al., 2003)

14.3 File Formats

The format of summary and detail files depends on the value of the `summary_file_format` and `detail_file_format` parameters in the `&ad_output` and `&nad_output` namelist groups (see the *Output Parameters* section). Possible choices are:

- `'HDF'` : A binary format based on the HDF5 format
- `'TXT'` : A text format modeled after MESA's profile file format

For both formats, the data stored in the files come in two flavors — scalars (a single value) and arrays (a sequence of values). Files in either format can be read in Python using the `pygyre.read_output()` function from PyGYRE (see the *Interpreting Output Files* chapter for examples).

14.3.1 HDF Format

HDF-format output files adhere to the following conventions:

- All data objects are attached to the root HDF5 group (/)
- Attributes are used to store scalar data
- Datasets are used to store array data
- Real values are written with type *H5T_IEEE_F64LE* when GYRE is compiled in double precision (the default), and type *H5T_IEEE_F32LE* otherwise
- Integer values are written with type *H5T_STD_I32LE*
- Complex values are written as a compound type, composed of a real component *re* and an imaginary component *im*; the types of these components are the same as for real values

14.3.2 TXT Format

TXT-format files adhere to the following conventions:

- The first three lines contain the scalar data:
  - The first line contains the column numbers for the scalar data, starting at 1
  - The second line contains the column labels for the scalar data
  - The third line contains the actual scalar data values
- The subsequent lines contain the array data:
  - The fourth line contains the column numbers for the array data, starting at 1
  - The fifth line contains the the column labels for the array data
  - The sixth and subsequent lines contain the actual array data (one line per array element)
- Complex values are written as two columns, with the first column containing the real component and the second the imaginary component

14.4 Output Filters

TBD

CHAPTER

FIFTEEN

STELLAR MODELS

This chapter documents the different types of stellar models that can be used with *GYRE frontends* to specify the equilibrium stellar configuration.

15.1 Evolutionary Models

Setting the `model_type` parameter of the `&model` namelist group to `'EVOL'` tells the frontend to read the equilibrium stellar model from a file created by a stellar evolution code (e.g., MESA).

15.1.1 Supported Formats

The format of the model file is specified by the `file_format` parameter of the `&model` namelist group (see the *Stellar Model Parameters* section). Possible choices are summarized in the table below.

`file_format`	Description
`'AMDL'`	Binary file describing an evolutionary model in AMDL format, as reverse engineered from the ADIPLS stellar oscillation code (Christensen-Dalsgaard, 2008)
`'B3'`	HDF5 file describing an evolutionary model in B3 format. This format is for testing purposes only, and will eventually be superseded and/or removed
`'FAMDL'`	Text file describing an evolutionary model in FAMDL format, as specified in the `CoRoT/ESTA File Formats` document
`'FGONG'`	Text file describing an evolutionary model in FGONG format, as specified in the updated `FGONG Format` document
`'GSM'`	HDF5 file describing an evolutionary model in GYRE Stellar Model (GSM) format, as specified in the *GSM File Format* section
`'MESA'`	Text file describing an evolutionary model in MESA format, as specified in the *MESA File Format* section
`'LOSC'`	Text file describing an evolutionary model in the revised LOSC format
`'OSC'`	Text file describing an evolutionary model in OSC format, as specified in the `CoRoT/ESTA File Formats` document)
`'WDEC'`	Text file describing an evolutionary model in WDEC format, as specified in Bischoff-Kim & Montgomery (2018)

15.1.2 Interpolation

Cubic spline interpolation is used to evaluate data between model grid points. The `deriv_type` parameter in the `&model` namelist group controls how the spline derivatives are set up.

15.1.3 Double Points

If a model contains a pair of adjacent points with the same radial coordinate r, this pair is treated as a double point representing a discontinuity in the density and some other thermodynamic quantities (but not the pressure or temperature). GYRE does not attempt to interpolate across double points, but instead handles them properly when solving equations through the use of *internal boundary conditions*.

15.2 Polytropic Models

Setting the `model_type` parameter of the `&model` namelist group to `'POLY'` tells the frontend to read the equilibrium stellar model from a POLY file created by the **`build_poly`** tool. See the *Composite Polytropes* and *Building POLY Models* appendices for further details.

15.3 Homogeneous Models

Setting the `model_type` parameter of the `&model` namelist group to `'HOM'` tells the frontend to create a homogeneous (uniform density) stellar model, equivalent to a polytrope with index $n = 0$. Because the structure of these model can be computed analytically, there is no need to read from an external file.

The `Gamma_1` parameter of the `&model` namelist group controls the first adiabatic index of the model, while the `n`, `s` and `grid_type` parameters control the model grid. See the *Stellar Model Parameters* section for further details.

15.4 Model Capabilities

Which data items are included in a given stellar model dictates what sorts of calculation can be performed on that model by the frontends. To this end, the different model types and file formats can be classified according to their *capabilities* (labeled using a single letter):

N
: The model supports non-adiabatic calculations.

D
: The model supports dimensioned results.

R
: The model supports differential rotational.

The table below summarizes the different capabilities of each model-type and file-format combination.

Model Type	File Format	N	D	R
EVOL	AMDL		X	
EVOL	B3	X	X	
EVOL	FAMDL		X	
EVOL	FGONG		X	
EVOL	GSM	X	X	X
EVOL	LOSC		X	
EVOL	MESA	X	X	X
EVOL	OSC	X	X	X
EVOL	WDEC		X	
POLY	POLY			
HOM	—			

15.5 MESA File Format

Files in MESA format store ASCII text data describing a MESA stellar model (note that MESA itself refers to these files as 'GYRE-format' files). To create one of these files in MESA, set the `pulse_data_format` parameter of the `&controls` namelist group to the value `'GYRE'`.

There are a number of versions of the MESA format, distinguished by the initial header line:

15.5.1 Version 0.01

The first line of version-0.01 MESA-format files is a header with the following columns:

Column	Symbol	Datatype	Definition
1	N	integer	number of grid points
2	M	real	stellar mass $[\mathrm{g}]$
3	R	real	photospheric radius $[\mathrm{cm}]$
4	L	real	photospheric luminosity $[\mathrm{erg\,s^{-1}}]$

The subsequent N lines contain the model data, one line per grid point extending from the center to the surface, with the following columns:

Column	Symbol	Datatype	Definition
1	k	integer	grid point index ($k = 1, \ldots, N$)
2	r	real	radial coordinate [cm]
3	$\frac{M_r}{M-M_r}$	real	transformed interior mass
4	L_r	real	interior luminosity [$\mathrm{erg\,s^{-1}}$]
5	P	real	total pressure [Ba]
6	T	real	temperature [K]
7	ρ	real	density [$\mathrm{g\,cm^{-3}}$]
8	∇	real	temperature gradient
9	N^2	real	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
10	c_V	real	specific heat at constant volume [$\mathrm{erg\,g^{-1}\,K^{-1}}$)
11	c_P	real	specific heat at constant pressure [$\mathrm{erg\,g^{-1}\,K^{-1}}$)
12	χ_T	real	equation-of-state partial $(\partial \ln P/\partial \ln T)_\rho$
13	χ_ρ	real	equation-of-state partial $(\partial \ln P/\partial \ln \rho)_T$
14	κ	real	opacity [$\mathrm{cm^2\,g^{-1}}$]
15	κ_T	real	opacity partial
16	κ_ρ	real	opacity partial
17	ϵ	real	total energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
18	$\epsilon_\mathrm{n}\,\epsilon_{\mathrm{n},T}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
19	$\epsilon_\mathrm{n}\,\epsilon_{\mathrm{n},\rho}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]

15.5.2 Version 0.19

The first line of version-0.19 MESA-format files is a header with the following columns:

Column	Symbol	Datatype	Definition
1	N	integer	number of grid points
2	M	real	stellar mass [g]
3	R	real	photospheric radius [cm]
4	L	real	photospheric luminosity [$\mathrm{erg\,s^{-1}}$]
5	19	integer	version number

The subsequent N lines contain the model data, one line per grid point extending from the center to the surface, with the following columns:

Column	Symbol	Datatype	Definition
1	k	integer	grid point index ($k = 1, \ldots, N$)
2	r	real	radial coordinate [cm]
3	$\frac{M_r}{M-M_r}$	real	transformed interior mass
4	L_r	real	interior luminosity [$\mathrm{erg\,s^{-1}}$]
5	P	real	total pressure [Ba]
6	T	real	temperature [K]
7	ρ	real	density [$\mathrm{g\,cm^{-3}}$]
8	∇	real	temperature gradient
9	N^2	real	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
10	Γ_1	real	adiabatic exponent
11	∇_{ad}	real	adiabatic temperature gradient
12	υ_T	real	thermodynamic coefficient
13	κ	real	opacity [$\mathrm{cm^2\,g^{-1}}$]
14	κ_T	real	opacity partial
15	κ_ρ	real	opacity partial
16	ϵ	real	total energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
17	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
18	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
19	Ω_{rot}	real	rotation angular frequency [$\mathrm{rad\,s^{-1}}$]

15.5.3 Version 1.00

The first line of version-1.00 MESA-format files is a header with the following columns:

Column	Symbol	Datatype	Definition
1	N	integer	number of grid points
2	M	real	stellar mass [g]
3	R	real	photospheric radius [cm]
4	L	real	photospheric luminosity [$\mathrm{erg\,s^{-1}}$]
5	100	integer	version number

The subsequent N lines contain the model data, one line per grid point extending from the center to the surface, with the following columns:

Column	Symbol	Datatype	Definition
1	k	integer	grid point index ($k = 1, \ldots, N$)
2	r	real	radial coordinate [cm]
3	M_r	real	interior mass [g]
4	L_r	real	interior luminosity [$\mathrm{erg\,s^{-1}}$]
5	P	real	total pressure [Ba]
6	T	real	temperature [K]
7	ρ	real	density [$\mathrm{g\,cm^{-3}}$]
8	∇	real	dimensionless temperature gradient
9	N^2	real	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
10	Γ_1	real	adiabatic exponent
11	∇_{ad}	real	adiabatic temperature gradient
12	υ_T	real	thermodynamic coefficient
13	κ	real	opacity [$\mathrm{cm^2\,g^{-1}}$]
14	$\kappa\,\kappa_T$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
15	$\kappa\,\kappa_\rho$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
16	ϵ	real	total energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
17	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
18	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
19	Ω_{rot}	real	rotation angular frequency [$\mathrm{rad\,s^{-1}}$]

15.5.4 Version 1.01

The first line of version-1.01 MESA-format files is a header with the following columns:

Column	Symbol	Datatype	Definition
1	N	integer	number of grid points
2	M	real	stellar mass [g]
3	R	real	photospheric radius [cm]
4	L	real	photospheric luminosity [$\mathrm{erg\,s^{-1}}$]
5	101	integer	version number

The subsequent N lines contain the model data, one line per grid point extending from the center to the surface, with the following columns:

Column	Symbol	Datatype	Definition
1	k	integer	grid point index ($k = 1, \ldots, N$)
2	r	real	radial coordinate [cm]
3	M_r	real	interior mass [g]
4	L_r	real	interior luminosity [$\mathrm{erg\,s^{-1}}$]
5	P	real	total pressure [Ba]
6	T	real	temperature [K]
7	ρ	real	density [$\mathrm{g\,cm^{-3}}$]
8	∇	real	dimensionless temperature gradient
9	N^2	real	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
10	Γ_1	real	adiabatic exponent
11	∇_{ad}	real	adiabatic temperature gradient
12	υ_T	real	thermodynamic coefficient
13	κ	real	opacity [$\mathrm{cm^2\,g^{-1}}$]
14	$\kappa\,\kappa_T$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
15	$\kappa\,\kappa_\rho$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
16	ϵ_{n}	real	nuclear energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
17	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
18	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
19	Ω_{rot}	real	rotation angular frequency [$\mathrm{rad\,s^{-1}}$]

15.5.5 Version 1.20

The first line of version-1.20 MESA-format files is a header with the following columns:

Column	Symbol	Datatype	Definition
1	N	integer	number of grid points
2	M	real	stellar mass [g]
3	R	real	photospheric radius [cm]
4	L	real	photospheric luminosity [$\mathrm{erg\,s^{-1}}$]
5	120	integer	version number

The subsequent N lines contain the model data, one line per grid point extending from the center to the surface, with the following columns:

Column	Symbol	Datatype	Definition
1	k	integer	grid point index ($k = 1, \ldots, N$)
2	r	real	radial coordinate [cm]
3	M_r	real	interior mass [g]
4	L_r	real	interior luminosity [$\mathrm{erg\,s^{-1}}$]
5	P	real	total pressure [Ba]
6	T	real	temperature [K]
7	ρ	real	density [$\mathrm{g\,cm^{-3}}$]
8	∇	real	dimensionless temperature gradient
9	N^2	real	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
10	Γ_1	real	adiabatic exponent
11	∇_{ad}	real	adiabatic temperature gradient
12	υ_T	real	thermodynamic coefficient
13	κ	real	opacity [$\mathrm{cm^2\,g^{-1}}$]
14	$\kappa\,\kappa_T$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
15	$\kappa\,\kappa_\rho$	real	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
16	ϵ_{n}	real	nuclear energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
17	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
18	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	real	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
19	ϵ_{grav}	real	gravothermal energy release rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
20	Ω_{rot}	real	rotation angular frequency [$\mathrm{rad\,s^{-1}}$]

15.6 GSM File Format

Files in GSM (GYRE Stellar Model) format store HDF5 data describing a stellar model. This format is intended as a portable, storage-efficient alternative to the *MESA File Format*. To create one of these files in MESA (from revision 21.12.1 onward), set the `pulse_data_format` parameter of the `&controls` namelist group to the value `'GSM'`.

The GSM format adheres to the following conventions:

- All data objects are attached to the root HDF5 group (*/*)
- Real values are written with type *H5T_IEEE_F64LE* when GYRE is compiled in double precision (the default), and type *H5T_IEEE_F32LE* otherwise
- Integer values are written with type *H5T_STD_I32LE*

There are a number of versions of the GSM format, distinguished by the `version` attribute in the root HDF5 group:

15.6.1 Version 0.00

Data items in the root HDF5 group of version-0.00 GSM-format files are as follows:

Item	Symbol	Object type	Data type	Definition
`n`	N	attribute	integer	number of grid points
`M_star`	M	attribute	real	stellar mass $[\mathrm{g}]$
`R_star`	R	attribute	real	photospheric radius $[\mathrm{cm}]$
`L_star`	L	attribute	real	photospheric luminosity $[\mathrm{erg\,s^{-1}}]$
`r`	r	dataset	real (`n`)	radial coordinate $[\mathrm{cm}]$
`w`	$\frac{M_r}{M-M_r}$	dataset	real (`n`)	transformed interior mass
`L_r`	L_r	dataset	real (`n`)	interior luminosity $[\mathrm{erg\,s^{-1}}]$
`p`	P	dataset	real (`n`)	total pressure $[\mathrm{Ba}]$
`rho`	ρ	dataset	real (`n`)	density $[\mathrm{g\,cm^{-3}}]$
`T`	T	dataset	real (`n`)	temperature $[\mathrm{K}]$
`N2`	N^2	dataset	real (`n`)	Brunt-Väisälä frequency squared $[\mathrm{s^{-2}}]$
`Gamma_1`	Γ_1	dataset	real (`n`)	adiabatic exponent
`nabla_ad`	∇_{ad}	dataset	real (`n`)	adiabatic temperature gradient
`delta`	υ_T	dataset	real (`n`)	thermodynamic coefficient
`nabla`	∇	dataset	real (`n`)	temperature gradient
`kappa`	κ	dataset	real (`n`)	opacity $[\mathrm{cm^2\,g^{-1}}]$
`kappa_T`	κ_T	dataset	real (`n`)	opacity partial
`kappa_rho`	κ_ρ	dataset	real (`n`)	opacity partial
`epsilon`	ϵ	dataset	real (`n`)	total energy generation rate $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`epsilon_T`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	dataset	real (`n`)	nuclear energy generation partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`epsilon_rho`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	dataset	real (`n`)	nuclear energy generation partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`Omega_rot`	Ω_{rot}	dataset	real (`n`)	rotation angular frequency $[\mathrm{rad\,s^{-1}}]$

15.6.2 Version 1.00

Data items in the root HDF5 group of version-1.00 GSM-format files are as follows:

Item	Symbol	Object type	Data type	Definition
`n`	N	attribute	integer	number of grid points
`version`	—	attribute	integer	100
`M_star`	M	attribute	real	stellar mass $[\mathrm{g}]$
`R_star`	R	attribute	real	photospheric radius $[\mathrm{cm}]$
`L_star`	L	attribute	real	photospheric luminosity $[\mathrm{erg\,s^{-1}}]$
`r`	r	dataset	real (`n`)	radial coordinate $[\mathrm{cm}]$
`M_r`	M_r	dataset	real (`n`)	interior mass $[\mathrm{g}]$
`L_r`	L_r	dataset	real (`n`)	interior luminosity $[\mathrm{erg\,s^{-1}}]$
`P`	P	dataset	real (`n`)	total pressure $[\mathrm{Ba}]$
`rho`	ρ	dataset	real (`n`)	density $[\mathrm{g\,cm^{-3}}]$
`T`	T	dataset	real (`n`)	temperature $[\mathrm{K}]$
`N2`	N^2	dataset	real (`n`)	Brunt-Väisälä frequency squared $[\mathrm{s^{-2}}]$
`Gamma_1`	Γ_1	dataset	real (`n`)	adiabatic exponent
`nabla_ad`	∇_{ad}	dataset	real (`n`)	adiabatic temperature gradient
`delta`	υ_T	dataset	real (`n`)	thermodynamic coefficient
`nabla`	∇	dataset	real (`n`)	temperature gradient
`kap`	κ	dataset	real (`n`)	opacity $[\mathrm{cm^2\,g^{-1}}]$
`kap_T`	$\kappa\,\kappa_T$	dataset	real (`n`)	opacity partial $[\mathrm{cm^2\,g^{-1}}]$
`kap_rho`	$\kappa\,\kappa_\rho$	dataset	real (`n`)	opacity partial $[\mathrm{cm^2\,g^{-1}}]$
`eps`	ϵ	dataset	real (`n`)	total energy generation rate $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`eps_T`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	dataset	real (`n`)	nuclear energy generation partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`eps_rho`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	dataset	real (`n`)	nuclear energy generation rate partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`Omega_rot`	Ω_{rot}	dataset	real (`n`)	rotation angular frequency $[\mathrm{rad\,s^{-1}}]$

15.6.3 Version 1.10

Data items in the root HDF5 group of version-1.10 GSM-format files are as follows:

Item	Symbol	Object type	Data type	Definition
`n`	n	attribute	integer	number of grid points
`version`	—	attribute	integer	110
`M_star`	M	attribute	real	stellar mass $[\mathrm{g}]$
`R_star`	R	attribute	real	photospheric radius $[\mathrm{cm}]$
`L_star`	L	attribute	real	photospheric luminosity $[\mathrm{erg\,s^{-1}}]$
`r`	r	dataset	real (`n`)	radial coordinate $[\mathrm{cm}]$
`M_r`	M_r	dataset	real (`n`)	interior mass $[\mathrm{g}]$
`L_r`	L_r	dataset	real (`n`)	interior luminosity $[\mathrm{erg\,s^{-1}}]$
`P`	P	dataset	real (`n`)	total pressure $[\mathrm{Ba}]$
`rho`	ρ	dataset	real (`n`)	density $[\mathrm{g\,cm^{-3}}]$
`T`	T	dataset	real (`n`)	temperature $[\mathrm{K}]$
`N2`	N^2	dataset	real (`n`)	Brunt-Väisälä frequency squared $[\mathrm{s^{-2}}]$
`Gamma_1`	Γ_1	dataset	real (`n`)	adiabatic exponent
`nabla_ad`	∇_{ad}	dataset	real (`n`)	adiabatic temperature gradient
`delta`	υ_T	dataset	real (`n`)	thermodynamic coefficient
`nabla`	∇	dataset	real (`n`)	dimensionless temperature gradient
`kap`	κ	dataset	real (`n`)	opacity $[\mathrm{cm^2\,g^{-1}}]$
`kap_kap_T`	$\kappa\,\kappa_T$	dataset	real (`n`)	opacity partial $[\mathrm{cm^2\,g^{-1}}]$
`kap_kap_rho`	$\kappa\,\kappa_\rho$	dataset	real (`n`)	opacity partial $[\mathrm{cm^2\,g^{-1}}]$
`eps`	ϵ	dataset	real (`n`)	total energy generation rate $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`eps_eps_T`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	dataset	real (`n`)	nuclear energy generation partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`eps_eps_rho`	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	dataset	real (`n`)	nuclear energy generation partial $[\mathrm{erg\,s^{-1}\,g^{-1}}]$
`Omega_rot`	Ω_{rot}	dataset	real (`n`)	rotation angular frequency $[\mathrm{rad\,s^{-1}}]$

15.6.4 Version 1.20

Data items in the root HDF5 group of version-1.20 GSM-format files are as follows:

Item	Symbol	Object type	Data type	Definition
n	n	attribute	integer	number of grid points
version	—	attribute	integer	120
M_star	M	attribute	real	stellar mass [g]
R_star	R	attribute	real	photospheric radius [cm]
L_star	L	attribute	real	photospheric luminosity [$\mathrm{erg\,s^{-1}}$]
r	r	dataset	real (n)	radial coordinate [cm]
M_r	M_r	dataset	real (n)	interior mass [g]
L_r	L_r	dataset	real (n)	interior luminosity [$\mathrm{erg\,s^{-1}}$]
P	P	dataset	real (n)	total pressure [Ba]
rho	ρ	dataset	real (n)	density [$\mathrm{g\,cm^{-3}}$]
T	T	dataset	real (n)	temperature [K]
N2	N^2	dataset	real (n)	Brunt-Väisälä frequency squared [$\mathrm{s^{-2}}$]
Gamma_1	Γ_1	dataset	real (n)	adiabatic exponent
nabla_ad	∇_{ad}	dataset	real (n)	adiabatic temperature gradient
delta	υ_T	dataset	real (n)	thermodynamic coefficient
nabla	∇	dataset	real (n)	dimensionless temperature gradient
kap	κ	dataset	real (n)	opacity [$\mathrm{cm^2\,g^{-1}}$]
kap_kap_T	$\kappa\,\kappa_T$	dataset	real (n)	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
kap_kap_rho	$\kappa\,\kappa_\rho$	dataset	real (n)	opacity partial [$\mathrm{cm^2\,g^{-1}}$]
eps	ϵ	dataset	real (n)	total energy generation rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
eps_eps_T	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},T}$	dataset	real (n)	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
eps_eps_rho	$\epsilon_{\mathrm{n}}\,\epsilon_{\mathrm{n},\rho}$	dataset	real (n)	nuclear energy generation partial [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
eps_grav	ϵ_{grav}	dataset	real (n)	gravothermal energy release rate [$\mathrm{erg\,s^{-1}\,g^{-1}}$]
Omega_rot	Ω_{rot}	dataset	real (n)	rotation angular frequency [$\mathrm{rad\,s^{-1}}$]

15.7 POLY File Format

Files in POLY format store HDF5 data describing a *composite polytrope model*. This format adheres to the following conventions:

- All data objects are attached to the root HDF5 group (/)
- Real values are written with type *H5T_IEEE_F64LE* when GYRE is compiled in double precision (the default), and type *H5T_IEEE_F32LE* otherwise
- Integer values are written with type *H5T_STD_I32LE*

Data items in the root HDF5 group are as follows:

Data Item	Variable	Object type	Data type	Definition
n	N	attribute	integer	number of grid points
n_r	$\mathcal{N}$	attribute	integer	number of regions
n_poly	n_i	attribute	real (n_r)	polytropic indices of regions
z_b	$z_{i-1/2}$	attribute	real (n_r-1)	radial coordinates of region boundaries
Delta_b	$\Delta_{i-1/2}$	attribute	real (n_r-1)	log density jump of region boundaries
Gamma_1	Γ_1	attribute	real	first adiabatic exponent
z	z	dataset	real (n)	polytropic radial coordinate
theta	θ	dataset	real (n)	Lane-Emden variable
dtheta	θ'	dataset	real (n)	Derivative of Lane-Emden variable

CHAPTER

SIXTEEN

OSCILLATION EQUATIONS

This chapter outlines how the oscillation equations solved by the GYRE frontends are obtained from the basic equations of stellar structure.

16.1 Fluid Equations

The starting point is the fluid equations, comprising the conservation laws for mass

$$\frac{\partial \rho}{\partial t} + \cdot\nabla\left(\rho \mathbf{v}\right) = 0$$

and momentum

$$\rho\left(\frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla\right)\mathbf{v} = -\nabla P - \rho\nabla\Phi;$$

the heat equation

$$\rho T\left(\frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla\right)S = \rho\epsilon_{\mathrm{n}} - \nabla\cdot\left(\mathbf{F}_{\mathrm{R}} + \mathbf{F}_{\mathrm{C}}\right);$$

and Poisson's equation

$$\nabla^2\Phi = 4\pi G\rho.$$

Here, ρ, P, T, S and $\mathbf{v}$ are the fluid density, pressure, temperature, specific entropy and velocity; Φ is the self-gravitational potential; ϵ_{n} is the specific nuclear energy generation rate; and $\mathbf{F}_{\mathrm{R}}$ and $\mathbf{F}_{\mathrm{C}}$ are the radiative and convective energy fluxes. An explicit expression for the radiative flux is provided by the radiative diffusion equation,

$$\mathbf{F}_{\mathrm{R}} = -\frac{c}{3\kappa\rho}\nabla(aT^4),$$

where κ is the opacity and a the radiation constant.

The fluid equations are augmented by the thermodynamic relationships between the four state variables (P, T, ρ and S). Only two of these are required to uniquely specify the state (we assume that the composition remains fixed over an oscillation cycle). In GYRE, P and S are adopted as these primary variables[1], and the other two are presumed to be derivable from them:

$$\rho = \rho(P, S), \qquad T = T(P, S).$$

The nuclear energy generation rate and opacity are likewise presumed to be functions of the pressure and entropy:

$$\epsilon_{\mathrm{n}} = \epsilon_{\mathrm{n}}(P, S), \qquad \kappa = \kappa(P, S).$$

[1] This may seem like a strange choice, but it simplifies the switch between adiabatic and non-adiabatic calculations

16.2 Equilibrium State

In a static equilibrium state the fluid velocity $\mathbf{v}$ vanishes. The momentum equation then becomes the hydrostatic equilibrium equation

$$\nabla P = -\rho \nabla \Phi.$$

Assuming the equilibrium is spherically symmetric, this simplifies to

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\rho \frac{\mathrm{d}\Phi}{\mathrm{d}r}.$$

Poisson's equation can be integrated once to yield

$$\frac{\mathrm{d}\Phi}{\mathrm{d}r} = \frac{G}{r^2} \int 4\pi \rho r^2 \, \mathrm{d}r = \frac{GM_r}{r^2},$$

where the second equality introduces the interior mass

$$M_r \equiv \int 4\pi \rho r^2 \, \mathrm{d}r. \tag{16.1}$$

The hydrostatic equilibrium equation thus becomes

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\rho \frac{GM_r}{r^2}.$$

The heat equation in the equilibrium state is

$$\rho T \frac{\partial S}{\partial t} = \rho \epsilon_{\mathrm{n}} - \nabla \cdot (\mathbf{F}_{\mathrm{R}} + \mathbf{F}_{\mathrm{C}}).$$

If the star is in thermal equilibrium then the left-hand side vanishes, and the nuclear heating rate balances the flux divergence term. Again assuming spherical symmetry, this is written

$$\frac{\mathrm{d}}{\mathrm{d}r} (L_{\mathrm{R}} + L_{\mathrm{C}}) = 4\pi r^2 \rho \epsilon_{\mathrm{n}},$$

where

$$L_{\mathrm{R}} \equiv 4\pi r^2 F_{\mathrm{R},r}, \qquad L_{\mathrm{C}} \equiv 4\pi r^2 F_{\mathrm{C},r}$$

are the radiative and convective luminosities, respectively.

16.3 Linearized Equations

Applying an Eulerian (fixed position, denoted by a prime) perturbation to the mass and momentum conservation equations, they linearize about the static equilibrium state as

$$\rho' + \nabla \cdot (\rho \mathbf{v}') = 0,$$

$$\rho \frac{\partial \mathbf{v}'}{\partial t} = -\nabla P' - \rho' \nabla \Phi - \rho \nabla \Phi'. \tag{16.2}$$

(in these expressions, the absence of a prime denotes an equilibrium quantity). Likewise, Poisson's equation becomes

$$\nabla^2 \Phi' = 4\pi G \rho'$$

Applying a Lagrangian (fixed mass element, denoted by a δ) perturbation to the heat equation, it linearizes about the equilibrium state as

$$T\frac{\partial \delta S}{\partial t} = \delta\epsilon_{\rm n} - \delta\left(\frac{1}{\rho}\nabla\cdot\mathbf{F}_{\rm R}\right),$$

where the heating term $\delta(\rho^{-1}\nabla\cdot\mathbf{F}_{\rm C})$ has been dropped[1] due to the continued lack of a workable theory for pulsation-convection coupling. Likewise applying a Lagrangian perturbation to the radiative diffusion equation,

$$\delta\mathbf{F}_{\rm R} = \left(4\frac{\delta T}{T} - \frac{\delta\rho}{\rho} - \frac{\delta\kappa}{\kappa}\right)\mathbf{F}_{\rm R} + \frac{\delta(\nabla\ln T)}{{\rm d}\ln T/{\rm d}r}F_{{\rm R},r}.$$

The thermodynamic relations linearize to

$$\frac{\delta\rho}{\rho} = \frac{1}{\Gamma_1}\frac{\delta P}{P} - \upsilon_T\frac{\delta S}{c_P}, \qquad \frac{\delta T}{T} = \nabla_{\rm ad}\frac{\delta P}{P} + \frac{\delta S}{c_P},$$

and the perturbations to the nuclear energy generation rate and opacity can be expressed as

$$\frac{\delta\epsilon_{\rm n}}{\epsilon_{\rm n}} = \epsilon_{\rm n,ad}\frac{\delta P}{P} + \epsilon_{{\rm n},S}\frac{\delta S}{c_P}, \qquad \frac{\delta\kappa}{\kappa} = \kappa_{\rm ad}\frac{\delta P}{P} + \kappa_S\frac{\delta S}{c_P}.$$

In these expressions, Eulerian and Lagrangian perturbations to any scalar quantity f are related via

$$\frac{\delta f}{f} = \frac{f'}{f} + \frac{\xi_r}{r}\frac{{\rm d}\ln f}{{\rm d}\ln r}.$$

Moreover, the thermodynamic partial derivatives are defined as

$$\Gamma_1 = \left(\frac{\partial\ln P}{\partial\ln\rho}\right)_S, \quad \upsilon_T = \left(\frac{\partial\ln\rho}{\partial\ln T}\right)_P, \quad c_P = \left(\frac{\partial S}{\partial\ln T}\right)_P, \quad \nabla_{\rm ad} = \left(\frac{\partial\ln T}{\partial\ln P}\right)_S,$$

and the nuclear and opacity partials are

$$\epsilon_{\rm n,ad} = \left(\frac{\partial\ln\epsilon_{\rm n}}{\partial\ln P}\right)_{\rm ad}, \quad \epsilon_{{\rm n},S} = c_P\left(\frac{\partial\ln\epsilon_{\rm n}}{\partial S}\right)_P, \quad \kappa_{\rm ad} = \left(\frac{\partial\ln\kappa}{\partial\ln P}\right)_{\rm ad}, \quad \kappa_S = c_P\left(\frac{\partial\ln\kappa}{\partial S}\right)_P.$$

The latter can be calculated from corresponding density and temperature partials via

$$\kappa_{\rm ad} = \frac{\kappa_\rho}{\Gamma_1} + \nabla_{\rm ad}\kappa_T, \qquad \kappa_S = -\upsilon_T\kappa_\rho + \kappa_T,$$
$$\epsilon_{\rm n,ad} = \frac{\epsilon_{{\rm n},\rho}}{\Gamma_1} + \nabla_{\rm ad}\epsilon_{{\rm n},T}, \qquad \epsilon_{{\rm n},S} = -\upsilon_T\epsilon_{{\rm n},\rho} + \epsilon_{{\rm n},T}.$$

16.4 Separated Equations

With a separation of variables in spherical-polar coordinates (r,θ,ϕ), and assuming an oscillatory time (t) dependence with angular frequency σ, solutions to the *linearized equations* can be expressed as

$$\begin{aligned}
\xi_r(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_r(r)\,Y_\ell^m(\theta,\phi)\,\exp(-{\rm i}\sigma t)\right],\\
\xi_\theta(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_{\rm h}(r)\,\frac{\partial}{\partial\theta}Y_\ell^m(\theta,\phi)\,\exp(-{\rm i}\sigma t)\right],\\
\xi_\phi(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_{\rm h}(r)\,\frac{{\rm i}m}{\sin\theta}Y_\ell^m(\theta,\phi)\,\exp(-{\rm i}\sigma t)\right],\\
f'(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{f}'(r)\,Y_\ell^m(\theta,\phi)\,\exp(-{\rm i}\sigma t)\right].
\end{aligned} \tag{16.3}$$

[1] This is known as a *frozen convection* approximation. GYRE offers multiple ways to freeze convection; see the *Convection Effects* section for further details.

Here, ξ_r, ξ_θ and ξ_ϕ are the radial, polar and azimuthal components of the displacement perturbation vector $\boldsymbol{\xi}$; Y_ℓ^m is the spherical harmonic with harmonic degree ℓ and azimuthal order m; and again f stands for any perturbable scalar. The displacement perturbation vector is related to the velocity perturbation via

$$\mathbf{v}' = \frac{\partial \boldsymbol{\xi}}{\partial t}.$$

Substituting the above solution forms into the linearized equations, the mechanical (mass and momentum conservation) equations become

$$\tilde{\rho}' + \frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(\rho r^2 \tilde{\xi}_r\right) - \frac{\ell(\ell+1)}{r}\rho\tilde{\xi}_\mathrm{h} = 0,$$

$$-\sigma^2 \rho \tilde{\xi}_r = -\frac{\mathrm{d}\tilde{P}'}{\mathrm{d}r} - \tilde{\rho}'\frac{\mathrm{d}\Phi}{\mathrm{d}r} - \rho\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r},$$

$$-\sigma^2 \rho r \tilde{\xi}_\mathrm{h} = -\tilde{P}' - \rho\tilde{\Phi}'.$$

Likewise, Poisson's equation becomes

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r}\right) - \frac{\ell(\ell+1)}{r^2}\tilde{\xi}_\mathrm{h} = 4\pi G\tilde{\rho}'$$

and the heat equation becomes

$$-\mathrm{i}\sigma T\delta\tilde{S} = \delta\tilde{\epsilon}_\mathrm{n} - \frac{\mathrm{d}\delta\tilde{L}_\mathrm{R}}{\mathrm{d}M_r} + \frac{\ell(\ell+1)}{\mathrm{d}\ln T/\mathrm{d}r}\frac{F_{\mathrm{R},r}}{\rho}\frac{\tilde{T}'}{T} + \ell(\ell+1)\frac{\tilde{\xi}_\mathrm{h}}{r}\frac{\mathrm{d}L_\mathrm{R}}{\mathrm{d}M_r},$$

where

$$\delta\tilde{L}_\mathrm{R} \equiv 4\pi r^2\left(\delta\tilde{F}_{\mathrm{R},r} + 2\frac{\tilde{\xi}_r}{r}F_{\mathrm{R},r}\right)$$

is the Lagrangian perturbation to the radiative luminosity. The radial part of the radiative diffusion equation becomes

$$\delta\tilde{F}_{\mathrm{R},r} = \left[4\frac{\delta\tilde{T}}{T} - \frac{\delta\tilde{\rho}}{\rho} - \frac{\delta\tilde{\kappa}}{\kappa} + \frac{\mathrm{d}(\delta\tilde{T}/T)/\mathrm{d}\ln r}{\mathrm{d}\ln T/\mathrm{d}\ln r}\right]F_{\mathrm{R},r}.$$

Finally, the thermodynamic, nuclear and opacity relations become

$$\frac{\delta\tilde{\rho}}{\rho} = \frac{1}{\Gamma_1}\frac{\delta\tilde{P}}{P} - \upsilon_T\frac{\delta\tilde{S}}{c_P}, \qquad \frac{\delta\tilde{T}}{T} = \nabla_\mathrm{ad}\frac{\delta\tilde{P}}{P} + \frac{\delta\tilde{S}}{c_P},$$

$$\frac{\delta\tilde{\epsilon}_\mathrm{n}}{\epsilon_\mathrm{n}} = \epsilon_{\mathrm{n,ad}}\frac{\delta\tilde{P}}{P} + \epsilon_{\mathrm{n},S}\frac{\delta\tilde{S}}{c_P}, \qquad \frac{\delta\tilde{\kappa}}{\kappa} = \kappa_\mathrm{ad}\frac{\delta\tilde{P}}{P} + \kappa_S\frac{\delta\tilde{S}}{c_P}.$$

16.5 Boundary Conditions

To form a closed system, the *separated equations* are augmented by algebraic relations at the inner and outer boundaries of the computational domain, and possibly at interior points as well.

16.5.1 Inner Boundary

When the inner boundary is placed at the stellar origin ($r = 0$), the requirement that solutions remain finite leads to the set of regularity conditions

$$\begin{aligned}\tilde{\xi}_r - \ell\tilde{\xi}_\mathrm{h} &= 0,\\ \ell\tilde{\Phi}' - r\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r} &= 0,\\ \delta\tilde{S} &= 0.\end{aligned}$$

Sometimes it's desirable that the inner boundary is instead placed at $r > 0$ — for instance, to excise the stellar core from the oscillation calculations. Then, there is much more flexibility in the choice of inner boundary condition. Possible options include setting $\tilde{\xi}_r = 0$ or $\tilde{\xi}_\mathrm{h} = 0$ instead of the first equation above.

16.5.2 Outer Boundary

The outer boundary typically corresponds to the stellar surface. Under the assumption that the density vanishes on and above this surface, the gravitational potential must match onto a solution to Laplace's equation that remains finite at infinity, leading to the potential boundary condition

$$(\ell + 1)\tilde{\Phi}' + r\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r} = 0.$$

Likewise, the assumption that there is no external pressure acting on the star (consistent with the vanishing surface density) gives the momentum boundary condition

$$\delta\tilde{P} = 0.$$

Finally, the thermal boundary condition can be derived from the equation

$$T^4(\tau) = \frac{4\sigma}{3}F_{\mathrm{R},r}\left(\tau + \frac{2}{3}\right)$$

describing the thermal structure of an atmosphere under the combined plane-parallel, grey, Eddington, local thermodynamic equilibrium and radiative equilibrium approximations. Here, τ is the vertical optical depth and σ_SB the Stefan-Boltzmann constant. Setting $\tau = 0$ (again, consistent with the vanishing surface density) and perturbing this equation yields the desired boundary condition

$$4\frac{\delta\tilde{T}}{T} = \frac{\delta\tilde{F}_{\mathrm{R},r}}{F_{\mathrm{R},r}}.$$

Complications arise when realistic stellar models are considered, because these typically extend only out to an optical depth $\tau = 2/3$ (or some similar value) corresponding to the photosphere. In such cases the density does not vanish at the nominal stellar surface, and the outer boundary conditions must be modified to account for the effects of the atmosphere region lying above the surface. Many stellar oscillation codes, including GYRE, can adopt more sophisticated formulations for the momentum boundary condition — for instance, based on the assumption that the outer atmosphere has an isothermal stratification. However, the atmospheric effects on the potential and thermal boundary conditions are usually neglected.

16.5.3 Internal Boundaries

Internal boundaries arise when the density and other related quantities show a radial discontinuity within the star. Across such a discontinuity $\tilde{\xi}_r$, $\delta\tilde{P}$ and $\delta\tilde{F}_{\mathrm{R},r}$ must remain continuous[1]. Internal boundary conditions on other perturbations follow from integrating the *separated equations* across the discontinuity, resulting in

$$\begin{aligned}
\tilde{P}'^{+} - \tilde{P}'^{-} &= \frac{\mathrm{d}\Phi}{\mathrm{d}r}\left(\rho^{+} - \rho^{-}\right)\tilde{\xi}_r, \\
\left.\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r}\right|^{+} - \left.\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r}\right|^{-} &= -4\pi G\left(\rho^{+} - \rho^{-}\right)\tilde{\xi}_r, \\
\tilde{T}'^{+} - \tilde{T}'^{-} &= 0.
\end{aligned}$$

Here, + (-) superscripts indicate quantities evaluated on the inner (outer) side of the discontinuity.

16.6 Dimensionless Formulation

To improve numerical stability, GYRE solves the *separated equations* and *boundary conditions* by recasting them into a dimensionless form that traces its roots back to Dziembowski (1971).

16.6.1 Variables

The independent variable is the fractional radius $x \equiv r/R$ (with R the stellar radius), and the dependent variables $\{y_1, y_2, \ldots, y_6\}$ are

$$\begin{aligned}
y_1 &= x^{2-\ell}\,\frac{\tilde{\xi}_r}{r}, \\
y_2 &= x^{2-\ell}\,\frac{\tilde{P}'}{\rho g r}, \\
y_3 &= x^{2-\ell}\,\frac{\tilde{\Phi}'}{g r}, \\
y_4 &= x^{2-\ell}\,\frac{1}{g}\frac{\mathrm{d}\tilde{\Phi}'}{\mathrm{d}r}, \\
y_5 &= x^{2-\ell}\,\frac{\delta\tilde{S}}{c_p}, \\
y_6 &= x^{-1-\ell}\,\frac{\delta\tilde{L}_{\mathrm{R}}}{L}.
\end{aligned} \tag{16.4}$$

[1] This is to ensure that the fluid doesn't 'tear', and that pressure forces and radiative heating remain finite.

16.6.2 Oscillation Equations

The dimensionless oscillation equations are

$$
\begin{aligned}
x\frac{\mathrm{d}y_1}{\mathrm{d}x} &= \left(\frac{V}{\Gamma_1} - 1 - \ell\right) y_1 + \left(\frac{\ell(\ell+1)}{c_1\omega^2} - \alpha_\gamma \frac{V}{\Gamma_1}\right) y_2 + \alpha_{\mathrm{grv}} \frac{\ell(\ell+1)}{c_1\omega^2} y_3 + \upsilon_T\, y_5, \\
x\frac{\mathrm{d}y_2}{\mathrm{d}x} &= \left(c_1\omega^2 - f_{\pi\gamma} A^*\right) y_1 + \left(3 - U + A^* - \ell\right) y_2 - \alpha_{\mathrm{grv}} y_4 + \upsilon_T\, y_5, \\
x\frac{\mathrm{d}y_3}{\mathrm{d}x} &= \alpha_{\mathrm{grv}} \left(3 - U - \ell\right) y_3 + \alpha_{\mathrm{grv}} y_4 \\
x\frac{\mathrm{d}y_4}{\mathrm{d}x} &= \alpha_{\mathrm{grv}} A^* U y_1 + \alpha_{\mathrm{grv}} \frac{V}{\Gamma_1} U y_2 + \alpha_{\mathrm{grv}} \ell(\ell+1) y_3 - \alpha_{\mathrm{grv}} (U + \ell - 2) y_4 - \alpha_{\mathrm{grv}} \upsilon_T\, U y_5, \\
x\frac{\mathrm{d}y_5}{\mathrm{d}x} &= \frac{V}{f_{\mathrm{rht}}} \left[\nabla_{\mathrm{ad}}(U - c_1\omega^2) - 4(\nabla_{\mathrm{ad}} - \nabla) + c_{\kappa,\mathrm{ad}} V\nabla + c_{\mathrm{dif}}\right] y_1 + \\
&\quad \frac{V}{f_{\mathrm{rht}}} \left[\frac{\ell(\ell+1)}{c_1\omega^2}(\nabla_{\mathrm{ad}} - \nabla) - c_{\kappa,\mathrm{ad}} V\nabla - c_{\mathrm{dif}}\right] y_2 + \\
&\quad \alpha_{\mathrm{grv}} \frac{V}{f_{\mathrm{rht}}} \left[\frac{\ell(\ell+1)}{c_1\omega^2}(\nabla_{\mathrm{ad}} - \nabla)\right] y_3 + \alpha_{\mathrm{grv}} \frac{V\nabla_{\mathrm{ad}}}{f_{\mathrm{rht}}} y_4 + \\
&\quad \left[\frac{V\nabla}{f_{\mathrm{rht}}}(4 f_{\mathrm{rht}} - c_{\kappa,S}) + \partial f_{\mathrm{rht}} + 2 - \ell\right] y_5 - \frac{V\nabla}{f_{\mathrm{rht}} c_{\mathrm{rad}}} y_6 \\
x\frac{\mathrm{d}y_6}{\mathrm{d}x} &= \left[\alpha_{\mathrm{hfl}} \ell(\ell+1) \left(\frac{\nabla_{\mathrm{ad}}}{\nabla} - 1\right) c_{\mathrm{rad}} - V c_{\epsilon,\mathrm{ad}} - \alpha_{\mathrm{egv}} c_{egv} \nabla_{\mathrm{ad}} V\right] y_1 + \\
&\quad \left[V c_{\epsilon,\mathrm{ad}} - \ell(\ell+1) c_{\mathrm{rad}} \left(\alpha_{\mathrm{hfl}} \frac{\nabla_{\mathrm{ad}}}{\nabla} - \frac{3 + \partial c_{\mathrm{rad}}}{c_1\omega^2}\right) + \alpha_{\mathrm{egv}} c_{egv} \nabla_{\mathrm{ad}} V\right] y_2 + \\
&\quad \alpha_{\mathrm{grv}} \left[\ell(\ell+1) c_{\mathrm{rad}} \frac{3 + \partial c_{\mathrm{rad}}}{c_1\omega^2}\right] y_3 + \\
&\quad \left[c_{\epsilon,S} - \alpha_{\mathrm{hfl}} \frac{\ell(\ell+1) c_{\mathrm{rad}}}{\nabla V} + \mathrm{i}\alpha_{\mathrm{thm}} \omega c_{\mathrm{thk}} + \alpha_{\mathrm{egv}} c_{egv}\right] y_5 - [1 + \ell]\, y_6,
\end{aligned}
\tag{16.5}
$$

where the dimensionless oscillation frequency is introduced as

$$
\omega \equiv \sqrt{\frac{R^3}{GM}}, \tag{16.6}
$$

(with M the stellar mass). These differential equations are derived from the separated equations, with the insertion of 'switch' terms (denoted α) that allow certain pieces of physics to be altered. See the *Physics Switches* section for more details.

For non-radial adiabatic calculations, the last two equations above are set aside and the y_5 terms dropped from the first four equations. For radial adiabatic calculations with `reduce_order=.TRUE.` (see the *Oscillation Parameters* section), the last four equations are set aside and the first two replaced by

$$
\begin{aligned}
x\frac{\mathrm{d}y_1}{\mathrm{d}x} &= \left(\frac{V}{\Gamma_1} - 1\right) y_1 - \frac{V}{\Gamma_1} y_2, \\
x\frac{\mathrm{d}y_2}{\mathrm{d}x} &= \left(c_1\omega^2 + U - A^*\right) y_1 + \left(3 - U + A^*\right) y_2.
\end{aligned}
$$

16.6.3 Boundary Conditions

Inner Boundary

When `inner_bound='REGULAR'`, GYRE applies regularity-enforcing conditions at the inner boundary:

$$\begin{aligned} c_1\omega^2 y_1 - \ell y_2 - \alpha_{\rm grv}\ell y_3 &= 0, \\ \alpha_{\rm grv}\ell y_3 - (2\alpha_{\rm grv} - 1) y_4 &= 0, \\ y_5 &= 0. \end{aligned}$$

(these are the dimensionless equivalents to the expressions appearing in the *Boundary Conditions* section).

When `inner_bound='ZERO_R'`, the first and second conditions above are replaced with zero radial displacement conditions,

$$\begin{aligned} y_1 &= 0, \\ y_4 &= 0. \end{aligned}$$

Likewise, when `inner_bound='ZERO_H'`, the first and second conditions are replaced with zero horizontal displacement conditions,

$$\begin{aligned} y_2 - y_3 &= 0, \\ y_4 &= 0. \end{aligned}$$

Outer Boundary

When `outer_bound='VACUUM'`, GYRE applies the outer boundary conditions

$$\begin{aligned} y_1 - y_2 &= 0 \\ \alpha_{\rm grv} U y_1 + (\alpha_{\rm grv}\ell + 1) y_3 + \alpha_{\rm grv} y_4 &= 0 \\ (2 - 4\nabla_{\rm ad} V) y_1 + 4\nabla_{\rm ad} V y_2 + 4 f_{\rm rht} y_5 - y_6 &= 0 \end{aligned}$$

(these are the dimensionless equivalents to the expressions appearing in the *Boundary Conditions* section).

When `outer_bound='DZIEM'`, the first condition above is replaced by the Dziembowski (1971) outer boundary condition,

$$\left\{ 1 + V^{-1} \left[\frac{\ell(\ell+1)}{c_1\omega^2} - 4 - c_1\omega^2 \right] \right\} y_1 - y_2 = 0.$$

When `outer_bound='UNNO'` or `'JCD'`, the first condition is replaced by the (possibly-leaky) outer boundary conditions described by Unno et al. (1989) and Christensen-Dalsgaard (2008), respectively. When `outer_bound='ISOTHERMAL'`, the first condition is replaced by a (possibly-leaky) outer boundary condition derived from a local dispersion analysis of waves in an isothermal atmosphere.

Finally, when `outer_bound='GAMMA'`, the first condition is replaced by the outer momentum boundary condition described by Ong & Basu (2020).

Internal Boundaries

Across density discontinuities, GYRE applies the boundary conditions

$$\begin{aligned} U^+ y_2^+ - U^- y_2^- &= y_1 (U^+ - U^-) \\ y_4^+ - y_4^- &= -y_1 (U^+ - U^-) \\ y_5^+ - y_5^- &= -V^+ \nabla_{\rm ad}^+ (y_2^+ - y_1) + V^- \nabla_{\rm ad}^- (y_2^- - y_1) \end{aligned}$$

(these are the dimensionless equivalents to the expressions appearing in the *Boundary Conditions* section). Here, + (-) superscripts indicate quantities evaluated on the inner (outer) side of the discontinuity. y_1, y_3 and y_6 remain continuous across discontinuities, and therefore don't need these superscripts.

16.6.4 Structure Coefficients

The various stellar structure coefficients appearing in the dimensionless oscillation equations and boundary conditions are defined as follows:

$$V = -\frac{\mathrm{d}\ln P}{\mathrm{d}\ln r} \qquad V_2 = x^{-2}V \qquad A^* = \frac{1}{\Gamma_1}\frac{\mathrm{d}\ln P}{\mathrm{d}\ln r} - \frac{\mathrm{d}\ln \rho}{\mathrm{d}\ln r} \qquad U = \frac{\mathrm{d}\ln M_r}{\mathrm{d}\ln r}$$

$$c_1 = \frac{r^3}{R^3}\frac{M}{M_r} \qquad f_{\pi\gamma} = \begin{cases} \alpha_\pi & A^* > 0, x < x_{\mathrm{atm}} \\ \alpha_\gamma & A^* > 0, x > x_{\mathrm{atm}} \\ 1 & \text{otherwise} \end{cases}$$

$$\nabla = \frac{\mathrm{d}\ln T}{\mathrm{d}\ln P} \qquad c_{\mathrm{lum}} = x^{-3}\frac{L_{\mathrm{R}} + L_{\mathrm{C}}}{L} \qquad c_{\mathrm{rad}} = x^{-3}\frac{L_{\mathrm{R}}}{L} \qquad \partial c_{\mathrm{rad}} = \frac{\mathrm{d}\ln c_{\mathrm{rad}}}{\mathrm{d}\ln r}$$

$$f_{\mathrm{rht}} = 1 - \alpha_{\mathrm{rht}}\frac{\mathrm{i}\omega c_{\mathrm{thn}}}{4} \qquad \partial f_{\mathrm{rht}} = -\alpha_{\mathrm{rht}}\frac{\mathrm{i}\omega c_{\mathrm{thn}}\partial c_{\mathrm{thn}}}{4 f_{\mathrm{rht}}}$$

$$c_{\kappa,\mathrm{ad}} = \frac{\alpha_{\kappa,\rho}\kappa_\rho}{\Gamma_1} + \nabla_{\mathrm{ad}}\alpha_{\kappa,T}\kappa_T \qquad c_{\kappa,S} = -\upsilon_T\alpha_{\kappa,\rho}\kappa_\rho + \alpha_{\kappa,T}\kappa_T$$

$$c_\epsilon = x^{-3}\frac{4\pi r^3 \rho \epsilon_{\mathrm{n}}}{L} \qquad c_{\epsilon,\mathrm{ad}} = c_\epsilon \epsilon_{\mathrm{n,ad}} \qquad c_{\epsilon,S} = c_\epsilon \epsilon_{\mathrm{n},S}$$

$$c_{\mathrm{dif}} = -4\nabla_{\mathrm{ad}} V \nabla + \nabla_{\mathrm{ad}}\left(V + \frac{\mathrm{d}\ln \nabla_{\mathrm{ad}}}{\mathrm{d}\ln x}\right)$$

$$c_{\mathrm{thn}} = \frac{c_P}{ac\kappa T^3}\sqrt{\frac{GM}{R^3}} \qquad \partial c_{\mathrm{thn}} = \frac{\mathrm{d}\ln c_{\mathrm{thn}}}{\mathrm{d}\ln r}$$

$$c_{\mathrm{thk}} = x^{-3}\frac{4\pi r^3 c_P T \rho}{L}\sqrt{\frac{GM}{R^3}} c_{egv} = x^{-3}\frac{4\pi r^3 \rho \epsilon_{\mathrm{grav}}}{L}$$

16.6.5 Physics Switches

GYRE offers the capability to adjust the oscillation equations through a number of physics switches, controlled by parameters in the `&osc` namelist group (see the *Oscillation Parameters* section). The table below summarizes the mapping between the switches appearing in the expressions above, and the corresponding namelist parameters.

Symbol	Parameter	Description
$\alpha_{\rm grv}$	`alpha_grv`	Scaling factor for gravitational potential perturbations. Set to 1 for normal behavior, and to 0 for the Cowling (1941) approximation
$\alpha_{\rm thm}$	`alpha_thm`	Scaling factor for local thermal timescale. Set to 1 for normal behavior, to 0 for the non-adiabatic reversible (NAR) approximation (see Gautschy et al., 1990), and to a large value to approach the adiabatic limit
$\alpha_{\rm hfl}$	`alpha_hfl`	Scaling factor for horizontal flux perturbations. Set to 1 for normal behavior, and to 0 for the non-adiabatic radial flux (NARF) approximation (see Townsend, 2003b)
α_γ	`alpha_gam`	Scaling factor for g-mode isolation. Set to 1 for normal behavior, and to 0 to isolate g modes as described by Ong & Basu (2020)
α_π	`alpha_pi`	Scaling factor for p-mode isolation. Set to 1 for normal behavior, and to 0 to isolate p modes as described by Ong & Basu (2020)
$\alpha_{\kappa,\rho}$	`alpha_kar`	Scaling factor for opacity density partial derivative. Set to 1 for normal behavior, and to 0 to suppress the density part of the κ mechanism
$\alpha_{\kappa,T}$	`alpha_kat`	Scaling factor for opacity temperature partial derivative. Set to 1 for normal behavior, and to 0 to suppress the temperature part of the κ mechanism
$\alpha_{\rm rht}$	`alpha_rht`	Scaling factor for time-dependent term in the radiative heat equation (see Unno & Spiegel, 1966). Set to 1 to include this term (Unno calls this the Eddington approximation), and to 0 to ignore the term
$\alpha_{\rm trb}$	`alpha_trb`	Scaling factor for the turbulent mixing length. Set to the convective mixing length to include the turbulent damping term (see the *Convection Effects* section), and to 0 to ignore the term

16.7 Rotation Effects

The oscillation equations presented in the preceding sections are formulated for a non-rotating star. The corresponding equations for a rotating star are significantly more complicated, and a complete treatment of rotation lies beyond the scope of GYRE. However, GYRE can include two important effects arising from rotation.

16.7.1 Doppler Shift

A lowest-order effect of rotation arises in the Doppler shift from transforming between the inertial reference frame and the local co-rotating reference frame. To incorporate this effect in the *separated equations*, all instances of the inertial-frame frequency σ are replaced by the co-rotating frequency

$$\sigma_{\rm c} \equiv \sigma - m\Omega_{\rm rot}, \tag{16.7}$$

where m is the azimuthal order of the mode and $\Omega_{\rm rot}$ is the rotation angular frequency. GYRE assumes shellular rotation (see, e.g., Meynet & Maeder, 1997), and so the latter can in principle be a function of radial coordinate r. The corresponding modifications to the *dimensionless formulation* involve replacing the dimensionless inertial-frame frequency ω with the dimensionless co-rotating frequency

$$\omega_{\rm c} \equiv \omega - m\Omega_{\rm rot}\sqrt{\frac{R^3}{GM}}.$$

16.7.2 Perturbative Coriolis Force Treatment

Another lowest-order effect of rotation arises from the Coriolis force. For slow rotation, this effect can be determined through a perturbation expansion technique (see, e.g., section 19.2 of Unno et al., 1989). To first order in $\Omega_{\rm rot}$, the frequency of a mode is shifted by the amount

$$\Delta\sigma = m \int_0^R \Omega_{\rm rot} \frac{{\rm d}\beta}{{\rm d}r} {\rm d}r,$$

where the rotation splitting kernel is

$$\frac{{\rm d}\beta}{{\rm d}r} = \frac{\left\{\tilde{\xi}_r^2 + [\ell(\ell+1)-1]\tilde{\xi}_{\rm h}^2 - 2\tilde{\xi}_r\tilde{\xi}_{\rm h}\right\}\rho r^2}{\int_0^R \left\{\tilde{\xi}_r^2 + \ell(\ell+1)\tilde{\xi}_{\rm h}^2\right\}\rho r^2 {\rm d}r}$$

In this latter expression, the eigenfunctions $\tilde{\xi}_r$ and $\tilde{\xi}_{\rm h}$ are evaluated from solutions to the oscillation equations without rotation. Therefore, the expression above for $\Delta\sigma$ can be applied as a post-calculation correction to non-rotating eigenfrequencies.

16.7.3 Non-Perturbative Coriolis Force Treatment

The perturbation expansion technique above breaks down when $\Omega_{\rm rot}/\sigma_{\rm c} \gtrsim 1$. To deal with such cases, the **gyre** frontend[1] can incorporate a non-perturbative treatment of the Coriolis force based on the 'traditional approximation of rotation' (TAR). The TAR was first introduced by Eckart (1960; *Hydrodynamics of Oceans and Atmospheres*) and has since been used extensively within the pulsation community (see, e.g., Bildsten et al., 1996; Lee & Saio, 1997; Townsend, 2003a; Bouabid et al., 2013; Townsend, 2020).

Within the TAR, the solution forms given in eqn. (16.3) are replaced by

$$\begin{aligned}
\xi_r(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_r(r)\,\Theta(\theta)\,\exp({\rm i}m\phi - {\rm i}\sigma t)\right], \\
\xi_\theta(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_{\rm h}(r)\,\frac{\widehat{\Theta}(\theta)}{\sin\theta}\,\exp({\rm i}m\phi - {\rm i}\sigma t)\right], \\
\xi_\phi(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{\xi}_{\rm h}(r)\,\frac{\widetilde{\Theta}(\theta)}{{\rm i}\sin\theta}\,\exp({\rm i}m\phi - {\rm i}\sigma t)\right], \\
f'(r,\theta,\phi;t) &= {\rm Re}\left[\sqrt{4\pi}\,\tilde{f}'(r)\,\Theta(\theta)\,\exp({\rm i}m\phi - {\rm i}\sigma t)\right]
\end{aligned} \tag{16.8}$$

Here, the Hough functions Θ, $\widehat{\Theta}$ and $\widetilde{\Theta}$ are the eigenfunctions obtained by solving Laplace's tidal equations (TEs), a second-order system of differential equations and boundary conditions in the polar (θ) coordinate (see Townsend 2020). Together with the associated eigenvalue λ, they depend on the harmonic degree ℓ[2] and azimuthal order m, and the spin parameter

$$q \equiv \frac{2\Omega_{\rm rot}}{\sigma_{\rm c}}.$$

[1] Currently the TAR cannot be used with the **gyre_tides** frontend, because it doesn't play well with forcing by the tidal potential $\Phi_{\rm T}$.

[2] The harmonic degree isn't formally a 'good' quantum number in the TAR; however, it can still be used to identify Hough functions by considering their behavior in the limit $\Omega_{\rm rot} \to 0$, as given in eqns. (16.9) and (16.10).

Solution Families

Solutions to the TEs can be grouped into two families based on the behavior of the eigenfunctions and eigenvalue in the limit $\Omega_{\rm rot} \to 0$. For the gravito-acoustic family,

$$\left.\begin{aligned} \Theta(\theta) &\to Y_\ell^m(\theta,0) \\ \widehat{\Theta}(\theta) &\to \sin\theta \frac{\partial}{\partial\theta} Y_\ell^m(\theta,0) \\ \widetilde{\Theta}(\theta) &\to -mY_\ell^m(\theta,0) \end{aligned}\right\} \quad \text{as } \Omega_{\rm rot} \to 0. \tag{16.9}$$

and $\lambda \to \ell(\ell+1)$. With these expressions, the solution forms (16.8) reduce to those given in eqn. (16.3).

Conversely, for the Rossby family

$$\left.\begin{aligned} \Theta(\theta) &\to 0 \\ \widehat{\Theta}(\theta) &\to mY_\ell^m(\theta,0) \\ \widetilde{\Theta}(\theta) &\to -\sin\theta \frac{\partial}{\partial\theta} Y_\ell^m(\theta,0) \end{aligned}\right\} \quad \text{as } \Omega_{\rm rot} \to 0. \tag{16.10}$$

and $\lambda \to 0$. Moreover, Rossby-mode eigenfrequencies also show the limiting behavior

$$\sigma_{\rm c} = \frac{2m\Omega_{\rm rot}}{\ell(\ell+1)} \quad \text{as } \Omega_{\rm rot} \to 0, \tag{16.11}$$

which is independent of the stellar structure.

Implementing the TAR

To implement the TAR in the *separated equations* and *boundary conditions*, all instances of the term $\ell(\ell+1)$ are replaced by the TE eigenvalue λ. Then, all instances of the harmonic degree ℓ are replaced by $\ell_{\rm e}$, an effective harmonic degree found by solving

$$\ell_{\rm e}(\ell_{\rm e}+1) = \lambda.$$

Similar steps are taken in the *dimensionless formulation*, but in the definitions of the dependent variables $\{y_1, y_2, \ldots, y_6\}$, ℓ is replaced by $\ell_{\rm i}$, the effective harmonic degree evaluated at the inner boundary.

16.8 Convection Effects

The oscillation equations presented in the preceding sections neglect the thermal and mechanical effects of convection. GYRE provides functionality for controlling how the thermal effects are suppressed, and how the mechanical effects can be included in a limited way.

16.8.1 Frozen Convection

In the derivation of the *linearized equations*, a term $\delta(\rho^{-1}\nabla\cdot\mathbf{F}_{\rm C})$ is dropped from the perturbed heat equation. This is known as a *frozen convection* approximation, and is grounded in the assumption that the energy transport by convection remains unaffected affected by the pulsation. There's more than one way to freeze convection; Pesnell (1990) presents a systematic review of different approaches. GYRE currently implements a subset of these:

- Pesnell's case 1, neglecting $\delta(\rho^{-1}\nabla\cdot\mathbf{F}_{\rm C})$ in the perturbed heat equation.
- Pesnell's case 4, neglecting $\delta L_{\rm C}$ (the Lagrangian perturbation to the convective luminosity) in the perturbed heat equation.

For further details, see the `conv_scheme` parameter in the *Oscillation Parameters* section.

16.8.2 Turbulent Damping

The Reynolds number in stars is very large, and thus convection tends to be turbulent. Following the treatment by Savonije & Witte (2002), GYRE can partially incorporate the mechanical effects of this turbulence by adding a term

$$f_{r,\mathrm{visc}} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(\rho\nu r^2\frac{\partial v_r'}{\partial r}\right)$$

to the radial component of the linearized momentum equation (16.2), representing the viscous force arising from radial fluid motion. Because this term depends on v_r', it is phase-shifted by a quarter cycle relative to the other terms in the equation, and acts like a drag force that damps oscillations. The turbulent viscous coefficient ν is evaluated as

$$\nu = \frac{(\alpha_{\mathrm{trb}}H_P)^2}{\tau_{\mathrm{conv}}}\left[1+\tau_{\mathrm{conv}}\frac{\sigma}{2\pi}\right]^{-1},$$

where H_P is the pressure scale height, α_{trb} is the turbulent mixing length (in units of H_P), and τ_{conv} the convective turnover timescale. This expression is adapted from equation (18) of Savonije & Witte (2002), with an exponent $s = 1$.

In GYRE α_{trb} is implemented as a switch (see the *Physics Switches* section). A reasonable choice is to set this parameter equal to the MLT mixing length parameter α_{MLT} of the stellar model.

16.9 Tidal Effects

To simulate the effects of tidal forcing by a companion, the **`gyre_tides`** frontend solves a modified form of the linearized momentum equation (16.2), namely

$$\rho\frac{\partial \mathbf{v}'}{\partial t} = -\nabla P' - \rho'\nabla P - \rho\nabla\Phi' - \rho\nabla\Phi_{\mathrm{T}}.$$

The final term on the right-hand side represents the external force arising from the tidal gravitational potential Φ_{T}.

16.9.1 Tidal Potential

The tidal potential can be expressed via the superposition

$$\Phi_{\mathrm{T}} = \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}\sum_{k=-\infty}^{\infty}\Phi_{\mathrm{T};\ell,m,k}. \tag{16.12}$$

of partial tidal potentials defined by

$$\Phi_{\mathrm{T};\ell,m,k} \equiv -\varepsilon_{\mathrm{T}}\frac{GM}{R}\,\overline{c}_{\ell,m,k}\left(\frac{r}{R}\right)^{\ell}Y_\ell^m(\theta,\phi)\,\exp(-\mathrm{i}k\Omega_{\mathrm{orb}}t).$$

(the summation over ℓ and m comes from a multipolar space expansion of the potential, and the summation over k from a Fourier time expansion). Here,

$$\varepsilon_{\mathrm{T}} = \left(\frac{R}{a}\right)^3 = \frac{\Omega_{\mathrm{orb}}R^3}{GM}\frac{q}{1+q}$$

quantifies the overall strength of the tidal forcing, in terms of the companion's mass qM, semi-major axis a and orbital angular frequency Ω_{orb}. These expressions, and the definition of the tidal expansion coefficients $\overline{c}_{\ell,m,k}$, are presented in greater detail in Sun et al. (2023).

16.9.2 Separated Equations

Because the tidal potential (16.12) superposes many different spherical harmonics, the solution forms (16.3) must be replaced by the more-general expressions

$$\begin{aligned}
\xi_r(r,\theta,\phi;t) &= \sum_{\ell,m,k} \tilde{\xi}_{r;\ell,m,k}(r)\, Y_\ell^m(\theta,\phi)\, \exp(-\mathrm{i}k\Omega_{\mathrm{orb}}t), \\
\xi_\theta(r,\theta,\phi;t) &= \sum_{\ell,m,k} \tilde{\xi}_{\mathrm{h};\ell,\mathrm{m},\mathrm{k}}(r)\, \frac{\partial}{\partial\theta} Y_\ell^m(\theta,\phi)\, \exp(-\mathrm{i}k\Omega_{\mathrm{orb}}t), \\
\xi_\phi(r,\theta,\phi;t) &= \sum_{\ell,m,k} \tilde{\xi}_{\mathrm{h};\ell,\mathrm{m},\mathrm{k}}(r)\, \frac{\mathrm{i}m}{\sin\theta} Y_\ell^m(\theta,\phi)\, \exp(-\mathrm{i}k\Omega_{\mathrm{orb}}t), \\
f'(r,\theta,\phi;t) &= \sum_{\ell,m,k} \tilde{f}'_{\ell,m,k}(r)\, Y_\ell^m(\theta,\phi)\, \exp(-\mathrm{i}k\Omega_{\mathrm{orb}}t)
\end{aligned} \tag{16.13}$$

(the notation for the sums has been abbreviated). Substituting these solution forms into the *linearized equations*, and taking advantage of the orthonormality of the spherical harmonics, leads to a fully separated set of differential equations for each combination of ℓ, m and k. A given set resembles the regular *separated equations*, with just a couple changes:

- The perturbation $\tilde{\Phi}'$ is replaced by $\tilde{\Psi}' \equiv \tilde{\Phi}' + \tilde{\Phi}_{\mathrm{T}}$, representing the total (self + tidal) gravitational potential perturbation.
- Rather than being an eigenvalue parameter, the oscillation frequency is set by $\sigma = k\Omega_{\mathrm{orb}}$, representing the forcing frequency of the partial tidal potential in an inertial frame.

The latter change means that the dimensionless frequency (16.6) becomes

$$\omega = \alpha_{\mathrm{frq}}\, k\Omega_{\mathrm{orb}} \sqrt{\frac{R^3}{GM}},$$

where α_{frq} is an additional term introduced to allow tuning of the tidal forcing frequency (see the `alpha_frq` parameter in the *Tidal Parameters* section).

16.9.3 Boundary Conditions

The boundary conditions accompanying the separated equations for a given $\{\ell, m, k\}$ combination resemble those *presented previously*, except that the outer potential boundary condition becomes

$$(\ell+1)\tilde{\Psi}' + r\frac{\mathrm{d}\tilde{\Psi}'}{\mathrm{d}r} = (2\ell+1)\tilde{\Phi}_{\mathrm{T};\ell,m,k},$$

where

$$\tilde{\Phi}_{\mathrm{T};\ell,m,k} \equiv -\varepsilon_{\mathrm{T}}\, \frac{GM}{R}\, \bar{c}_{\ell,m,k} \left(\frac{r}{R}\right)^\ell. \tag{16.14}$$

describes the radial dependence of the partial tidal potential.

CHAPTER

SEVENTEEN

COMPOSITE POLYTROPES

Composite polytropes are an extension of standard polytrope models for stellar structure, to allow for discontinuities in the density ρ and/or the polytropic index n. This appendix lays out the mathematical formalism[1] underpinning them; it is intended to complement the *Building POLY Models* appendix, which describes how composite polytrope models can be built using the **build_poly** executable.

17.1 Equation of State

Consider a composite polytrope composed of $\mathcal{N}$ regions extending from the origin out to the stellar surface. In the i'th region ($1 \leq i \leq \mathcal{N}$), the pressure P and density ρ are related by the polytropic equation-of-state

$$\frac{P_i}{P_{i,0}} = \left(\frac{\rho_i}{\rho_{i,0}}\right)^{(n_i+1)/n_i} \tag{17.1}$$

where the normalizing pressure $P_{i,0}$ and density $\rho_{i,0}$, together with the polytropic index n_i, are constant across the region but may change from one region to the next. At the $\mathcal{N}-1$ boundaries between adjacent regions, the pressure and interior mass M_r are required to be continuous, but the density may jump.

17.2 Structure Equations

17.2.1 Lane-Emden Equation

In the i'th region, a composite polytrope satisfies the equation of hydrostatic equilibrium

$$-\frac{1}{\rho_i}\frac{\mathrm{d}P_i}{\mathrm{d}r} = \frac{\mathrm{d}\Phi_i}{\mathrm{d}r}$$

Substituting in the polytropic equation-of-state 17.1 yields

$$\frac{(n_i+1)P_{i,0}}{\rho_{i,0}^{1+1/n_i}}\frac{\mathrm{d}}{\mathrm{d}r}\left(\rho_i^{1/n_i}\right) = -\frac{\mathrm{d}\Phi_i}{\mathrm{d}r},$$

which can then be integrated with respect to r to give

$$\frac{(n_i+1)P_{i,0}}{\Phi_{i,0}\,\rho_{i,0}}\left(\frac{\rho_i^{1/n_i}}{\rho_{i,0}^{1/n_i}} - 1\right) = \left(1 - \frac{\Phi_i}{\Phi_{i,0}}\right).$$

[1] The formalism here is based on *Mixed polytrope with density discontinuities* (Christensen-Dalsgaard, 2015, unpublished), with extensions to allow for constant-density regions.

Here, the constants of integration have been chosen so that $\Phi_i = \Phi_{i,0}$ when $\rho_i = \rho_{i,0}$. Rearranging, the density follows as

$$\rho_i = \rho_{i,0}\,\theta_i^{n_i},$$

where the polytropic dependent variable is introduced as

$$\theta_i = \left[\frac{\Phi_{i,0}\,\rho_{i,0}}{(n_i+1)P_{i,0}}\left(1-\frac{\Phi_i}{\Phi_{i,0}}\right)+1\right].$$

With these expressions, Poisson's equation

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}P_i}{\mathrm{d}r}\right) = 4\pi G\rho_i$$

is recast as

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}\theta_i}{\mathrm{d}r}\right) = -\frac{1}{A_i}\theta_i^{n_i},$$

where

$$A_i \equiv \frac{(n_i+1)P_{i,0}}{4\pi G\rho_{i,0}^2}.$$

A change of variables to the polytropic independent variable $z \equiv A_1^{-1/2} r$ results in the dimensionless form

$$\frac{1}{z^2}\frac{\mathrm{d}}{\mathrm{d}z}\left(z^2\frac{\mathrm{d}\theta_i}{\mathrm{d}z}\right) = -B_i\theta_i^{n_i}, \tag{17.2}$$

where $B_i \equiv A_1/A_i$. This can be regarded as a generalization of the usual Lane-Emden equation to composite polytropes.

17.2.2 Continuity Relations

At the boundary between adjacent regions, the pressure and interior mass must be continuous. If $z_{i-1/2}$ denotes the coordinate of the boundary between the $i-1$ and i regions, then these continuity relations are expressed as

$$\left.\begin{aligned} B_i &= \frac{n_{i-1}+1}{n_i+1}\frac{\theta_i^{n_i+1}}{\theta_{i-1}^{n_{i-1}+1}}\frac{\rho_{i,0}^2}{\rho_{i-1,0}^2}\,B_{i-1}, \\ \theta_i' &= \frac{n_{i-1}+1}{n_i+1}\frac{\theta_{i-1}^{n_{i-1}+1}}{\theta_i^{n_i+1}}\frac{\rho_{i,0}}{\rho_{i-1,0}}\,\theta_{i-1}', \end{aligned}\right\} \quad \text{at } z = z_{i-1/2}$$

respectively.

17.3 Solution Method

17.3.1 Specification

The structure of a composite polytrope is specified completely by

- a set of $\mathcal{N}$ polytropic indices n_i
- a set of $\mathcal{N}-1$ boundary coordinates $z_{i-1/2}$
- a set of $\mathcal{N}$ density jumps $\Delta_{i-1/2} \equiv \ln[\rho_i(z_{i-1/2})/\rho_{i-1}(z_{i-1/2}]$

Although the normalizing densities $\rho_{i,0}$ have so far been left unspecified, it's convenient to choose them as the density at the beginning of their respective regions.

17.3.2 Solution

The *structure equations* may be solved as an initial value problem. In the first region ($i = 1$) this IVP involves integrating the Lane-Emden equation 17.2 from the center $z = 0$ to the first boundary $z = z_{3/2}$, with the initial conditions

$$\left.\begin{aligned} \theta_i &= 1, \\ \theta_i' &= 0, \\ B_1 &= 1, \\ t_1 &= 1 \end{aligned}\right\} \quad \text{at } z = 0$$

(here, $t_i \equiv \rho_{i,0}/\rho_{1,0}$).

The IVP in the intermediate regions ($i = 2, \ldots, \mathcal{N} - 1$) involves integrating from $z = z_{i-1/2}$ to $z = z_{i+1/2}$, with initial conditions established from the preceding region via

$$\left.\begin{aligned} \theta_i &= 1, \\ \theta_i' &= \frac{n_{i-1}+1}{n_i+1} \frac{\theta_{i-1}^{n_{i-1}+1}}{\theta_i^{n_i+1}} \frac{t_i}{t_{i-1}} \theta_{i-1}', \\ B_i &= \frac{n_{i-1}+1}{n_i+1} \frac{\theta_i^{n_i+1}}{\theta_{i-1}^{n_{i-1}+1}} \frac{t_i^2}{t_{i-1}^2} B_{i-1}, \\ \ln t_i &= \ln t_{i-1} + n_{i-1} \ln \theta_{i-1} - n_i \ln \theta_i + \Delta_{i-1/2}. \end{aligned}\right\} \quad \text{at } z = z_{i-1/2}$$

The IVP in the final region ($i = \mathcal{N}$) involves integrating from $z_{\mathcal{N}-1/2}$ until $\theta_{\mathcal{N}} = 0$. This point defines the stellar surface, $z = z_{\rm s}$. For some choices of n_i, $z_{i-1/2}$ and/or $\Delta_{i-1/2}$, the point $\theta = 0$ can arise in an earlier region $i = \mathcal{N}_{\rm t} < \mathcal{N}$; in such cases, the model specification must be truncated to $\mathcal{N}_{\rm t}$ regions.

17.4 Physical Variables

Once the Lane-Emden equation 17.2 has been solved, the density in each region can be evaluated by

$$\rho_i = \rho_{1,0}\, t_i\, \theta_i^{n_i}.$$

The pressure then follows from the equation-of-state 17.1 as

$$P_i = P_{1,0} \frac{n_1+1}{n_i+1} \frac{t_i^2}{B_i} \theta_i^{n_i+1}.$$

The interior mass m is evaluated by introducing the auxiliary quantity μ, which is defined in the first region by

$$\mu_1(z) = -z^2 \theta_1'(z),$$

and in subsequent regions by

$$\mu_i(z) = \mu_{i-1}(z_{i-1/2}) - \frac{t_i}{B_i} \left[z^2 \theta_i'(z) - z_{i-1/2}^2 \theta_i'(z_{i-1/2}) \right].$$

The interior mass then follows as

$$M_r = M \frac{\mu_i}{\mu_{\rm s}},$$

where $\mu_{\rm s} \equiv \mu_{\mathcal{N}}(z_{\rm s})$.

17.5 Structure Coefficients

The *structure coefficients* for composite polytropic models are evaluated using

$$V_2 = -(n_i + 1)\frac{z_\mathrm{s}^2}{z}\frac{\theta_i'}{\theta_i}, \qquad A^* = V_2\frac{z^2}{z_\mathrm{s}^2}\left(\frac{n_i}{n_i+1} - \frac{1}{\Gamma_1}\right),$$
$$U = \frac{t_i z^3}{\mu_i}\theta_i^{n_i}, \qquad c_1 = \frac{z^3}{z_\mathrm{s}^3}\frac{\mu_\mathrm{s}}{\mu_i},$$

where μ is the auxiliary mass variable introduced above.

CHAPTER

EIGHTEEN

BUILDING POLY MODELS

This appendix describes the **build_poly** executable, which builds a composite polytropic stellar model and writes it to a file in the *POLY format*.

18.1 Installation

The **build_poly** executable is automatically compiled when GYRE is built, and installed in the *$GYRE_DIR*/bin directory (see the main *Installation* chapter).

18.2 Example Walkthrough: Simple Polytrope

As the first example of **build_poly** in action, let's build a simple (i.e., single-region) $n = 3$ polytrope, that for instance describes the structure of a radiation-pressure dominated, fully convective star.

18.2.1 Assembling a Namelist File

First, let's assemble a namelist file containing the various parameters which control a **build_poly** run. Using a text editor, create the file build_poly.simple.in with the following content cut-and-pasted in:

```
&poly
        n_poly = 3.0  ! Polytropic index of single region
/

&num
        dz = 1E-2     ! Radial spacing of points
        toler = 1E-10 ! Tolerance of integrator
/

&out
        file = 'poly.simple.h5' ! Name of output file
/
```

Detailed information on the namelist groups expected in **build_poly** input files can be found in the *Input Files* section. Here, let's briefly narrate the parameters appearing in the file above:

- In the &poly namelist group, the n_poly parameter sets the polytropic index.
- In the &num namelist group, the dz parameter sets the radial spacing of points, and the toler parameter sets the tolerance of the numerical integrator.

- In the `&output` namelist group, the `file` parameter sets the name of the output file.

18.2.2 Running build_poly

To run **`build_poly`**, use the command

```
$ $GYRE_DIR/bin/build_poly build_poly.simple.in
```

There is no screen output produced during the run, but at the end the `poly.simple.h5` will be written to disk. This file, which is in *POLY format*, can be used as the input stellar model in a GYRE calculation; but it can also be explored in Python (see Fig. 18.1) using the `read_model` function from PyGYRE.

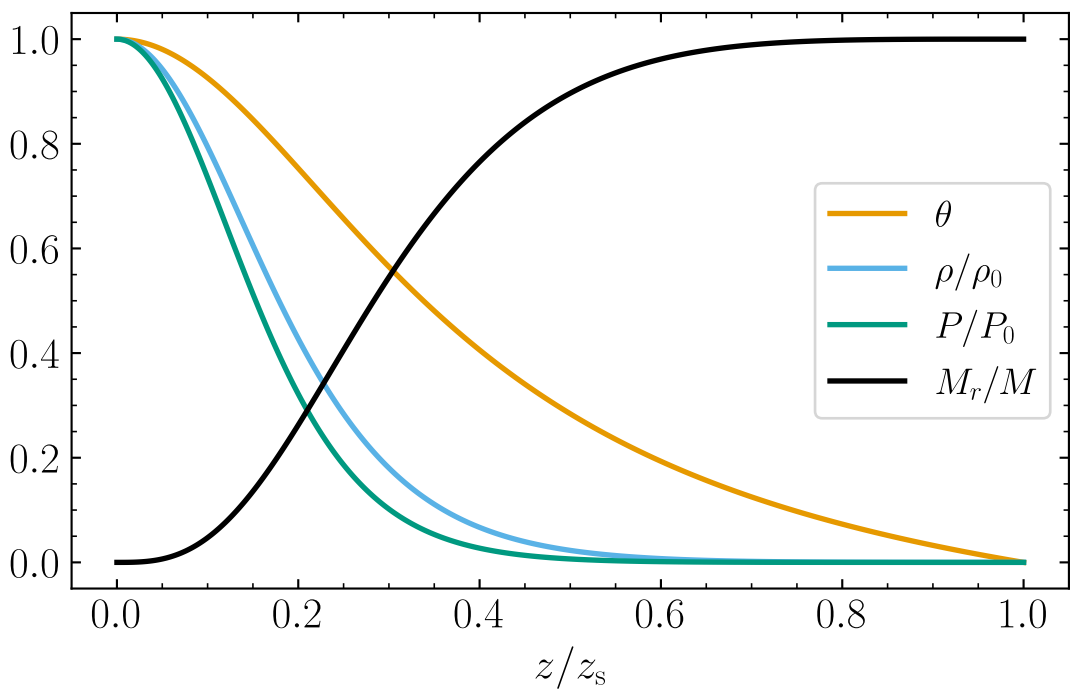


Fig. 18.1: Plot of the Lane-Emden solution variable θ, density ρ, pressure P and interior mass M_r as a function of radial coordinate, for the simple polytrope. (`Source`)

18.3 Example Walkthrough: Composite Polytrope

As the second example of **`build_poly`** in action, let's build a two-region composite polytrope. The polytropic index is $n = 3$ in the inner region, and $n = 1.5$ in the outer region. At the boundary between the regions, located at radial coordinate $z = 1.4$, the logarithmic density jump is $\Delta = -0.5$.

18.3.1 Assembling a Namelist File

Using a text editor, create the file `build_poly.composite.in` with the following content cut-and-pasted in:

```
&poly
        n_r = 2           ! Number of regions
        n_poly = 3.0, 1.5 ! Polytropic indices of regions
        z_b = 1.4         ! Radial coordinate of region boundaries
        Delta_b = -0.5    ! Logarithmic density jump at region boundaries
/

&num
        dz = 1E-2     ! Radial spacing of points
        toler = 1E-10 ! Tolerance of integrator
/

&out
        file = 'poly.composite.h5' ! Name of output file
/
```

Again, detailed information on the namelist groups expected in **`build_poly`** input files can be found in the *Input Files* section. Here, let's briefly narrate the parameters appearing in the file above:

- In the `&poly` namelist group, the `n_r` parameter sets the number of regions; the `n_poly` parameter sets the polytropic indices in the two regions; the `z_b` sets the radial coordinate of the boundary between the regions; and the `Delta_b` sets the density jump at this boundary.
- In the `&num` namelist group, the `dz` parameter sets the radial spacing of points, and the `toler` parameter sets the tolerance of the numerical integrator.
- In the `&output` namelist group, the `file` parameter sets the name of the output file.

18.3.2 Running build_poly

As before, to run **`build_poly`** use the command

```
$ $GYRE_DIR/bin/build_poly build_poly.composite.in
```

There is no screen output produced during the run, but at the end the `poly.composite.h5` will be written to disk. This file, which is in *POLY format*, can be used as the input stellar model in a GYRE calculation; but it can also be explored in Python (see Fig. 18.2) using the `read_model` function from PyGYRE.

18.4 Input Files

The **`build_poly`** executable reads parameters from an input file that defines a number of Fortran namelist groups, as described below.

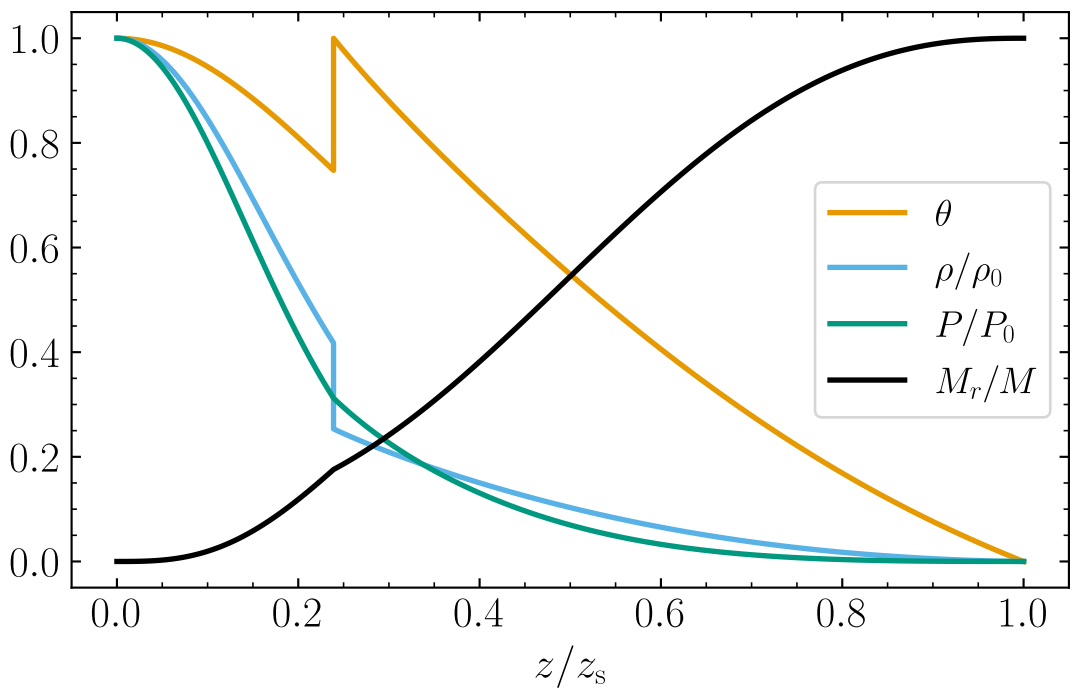


Fig. 18.2: Plot of the Lane-Emden solution variable θ, density ρ, pressure P and interior mass M_r as a function of radial coordinate, for the composite polytrope. Note the density discontinuity, and the associated discontinuities in the gradients of the pressure and interior mass. (`Source`)

18.4.1 Polytrope Parameters

The `&poly` namelist group defines polytrope parameters; the input file can contain only one. Allowable parameters are:

`n_r` (default `1`)
Number of regions

`n_poly` (default `0`)
Comma-separated list of length `n_r`, specifying polytropic indices for regions

`z_b`
Comma-separated list of length `n_r`-1, specifying radial coordinates of boundaries between regions

`Delta_b`
Comma-separated list of length `n_r`-1, specifying logarithmic density jumps at boundaries between regions

`Gamma_1` (default `5./3.`)
First adiabatic exponent

18.4.2 Numerical Parameters

The `&num` namelist group defines numerical parameters; the input file can contain only one. Allowable parameters are:

`dz` (default `1E-2`)
Spacing of grid points in polytropic radial coordinate z

`toler` (default `1E-10`)
Relative and absolute tolerance of Lane-Emden integrations

18.4.3 Output Parameters

The `&out` namelist group defines output parameters; the input file can contain only one. Allowable parameters are:

`file`
Name of *POLY-format* file to write to

CHAPTER

NINETEEN

EVALUATING TIDAL EIGENVALUES

This appendix describes the **eval_lambda** executable, which evaluates the eigenvalue λ appearing in Laplace's tidal equations (see the *Rotation Effects* section). This executable is used for the calculations presented in Townsend (2020).

19.1 Installation

eval_lambda is automatically compiled when GYRE is built, and installed in the *$GYRE_DIR*/bin directory (see the main *Installation* chapter).

19.2 Running

Unlike most other GYRE executables, the parameters for **eval_lambda** are supplied directly on the command line, with the syntax

```
$ ./eval_lambda l m q_min q_max n_q log_q rossby filename
```

This evaluates λ for harmonic degree ℓ and azimuthal order m on a grid $\{q_1, q_2, \ldots, q_N\}$ in the spin parameter, writing the results to the file `filename`. If the flag `log_q` has the value `T` then the grid is logarithmically spaced:

$$q_i = 10^{(1-w_i)\log q_{\min} + w_i \log q_{\max}},$$

where

$$w_i \equiv \frac{i-1}{N-1}.$$

Alternatively, if `log_q` has the value `F`, then the grid is linearly spaced:

$$q_i = (1-w_i)q_{\min} + w_i q_{\max}.$$

As a special case, when $n_q = 1$, $q_{\min}$ and $q_{\max}$ must match, and the single q point is set to equal them.

If the flag `rossby` has the value `T`, then the Rossby λ family is evaluated; otherwise, the gravito-acoustic family is evaluated.

The table below summarizes the mapping between the user-definable controls appearing in the expressions above, and the corresponding command-line parameters:

Symbol	Parameter
ℓ	`l`
m	`m`
$q_{\min}$	`q_min`
$q_{\max}$	`q_max`
N	`n_q`

19.3 Interpreting Output

The output file created by **`eval_lambda`** is in GYRE's *HDF Format*, with the following data:

`l` (integer scalar)
Harmonic degree ℓ

`k` (integer scalar)
Meridional order k (see Townsend, 2003a)

`m` (integer scalar)
Azimuthal order m

`rossby` (logical scalar)
Rossby family flag

`q` (real array)
Spin parameter q

`lambda` (real array)
Eigenvalue λ

INDEX