GYRE: Yet another oscillation code, why we need it and how it works

Rich Townsend & Seth Teitler
University of Wisconsin-Madison
What might one want from a new code?

- Improved flexibility to handle new problems
  - oscillations with differential rotation & magnetic fields
  - dynamic tides in binary stars
- Greater accuracy and robustness
  - “hands-off” asteroseismic analyses
  - integrated oscillation & stellar evolution simulations
- Higher performance
  - Take advantage of multiple cores / nodes
GYRE: A new oscillation code suite

- Programmatic motivation: developed as part of “Wave transport of angular momentum: a new spin on massive-star evolution” (NSF grant #AST 0908688)

- Personal motivations:
  - why does the BOOJUM code (Townsend 2005) work in cases x and y, but not in case z?
  - I enjoy programming!
Statement of the problem

- Stellar oscillation is a linear two-point boundary-value problem (BVP):

\[
\frac{dy}{dx} = A(x) y
\]

\[
B_a y_a \equiv B_a y(x_a) = 0
\]

\[
B_b y_b \equiv B_b y(x_b) = 0
\]

- The problem specifics are defined by the Jacobian matrix \( A \) and the boundary conditions \( B \)
Alternative approaches to solving BVPs

**Shooting**

Smeyers (1966, 1967)

**Relaxation**

Castor (1970)
Alternative approaches to solving BVPs

Shooting

Smeyers (1966, 1967)

Relaxation

Castor (1970)

At a fundamental level, both approaches are the same!
• Replace the differential equations by finite differences on a discrete grid \( x = x^k \) \( (k = 1, \ldots, N) \):

\[
\frac{y^{k+1} - y^k}{x^{k+1} - x^k} = A \left( \frac{x^{k+1} + x^k}{2} \right) \frac{y^{k+1} + y^k}{2}
\]

• Combine the difference equations with the boundary conditions to form a large, sparse linear system for \( y^k \)
Shooting via superposition

- Use initial-value problem (IVP) integrator to solve
  \[
  \frac{dY}{dx} = A(x)Y, \quad Y(x_a) = I
  \]

- The fundamental solution \( Y \) relates \( y^b \) back to \( y^a \):
  \[
  y^b = Y(x^b) y^a
  \]

- The BVP becomes a linear system for \( y^a \):
  \[
  B^a y^a = 0 \\
  B^b Y(x^b) y^a = 0
  \]
Multiple shooting: the best of both worlds

- Apply shooting across multiple intervals of a discrete grid \( x = x^k (k = 1, \ldots, N) \):
  \[
  y^{k+1} = Y(x^{k+1}; x^k) y^k
  \]

- Combine with the boundary conditions to form large, sparse linear system for \( y^k \)

- Stability is improved vs. single/double shooting

- Depending on how we evaluate \( Y^{k+1, k} = Y(x^{k+1}; x^k) \), accuracy is improved vs. relaxation

- Multiple shooting is easy to parallelize
Calculating the fundamental solution matrices

- Simple approach following Gabriel & Noels (1976): assume the Jacobian matrix $A(x)$ is constant in each interval $x^k \leq x \leq x^{k+1}$

- The fundamental solution matrix is then a matrix exponential:

$$Y^{k+1; k} = \exp \left\{ \left[ x^{k+1} - x^k \right] A \right\}$$

- This approach has \textit{arbitrarily high resolution} of eigenfunction oscillations

- However, it is only second-order accurate
Higher-order approaches using the Magnus method

- Magnus (1954): solutions to the IVP

\[
\frac{dY}{dx} = A(x)Y, \quad Y(x_a) = I
\]

can be written as

\[
Y = \exp \{ M(x) \}
\]

- The Magnus matrix $M$ can be expanded as an infinite series, with leading terms

\[
M(x) = \int_{x_a}^{x} A(x_1) \, dx_1 - \frac{1}{2} \int_{x_a}^{x} \left[ \int_{x_a}^{x_1} A(x_2) \, dx_2, A(x_1) \right] \, dx_1 + \ldots
\]
Magnus methods in GYRE

- Integrals in the Magnus expansion are evaluated using Gauss-Legendre quadrature
- Matrix exponentials are evaluated via a spectral decomposition of $M$:
  $$\exp M = U (\exp \Lambda) U^{-1}$$
- Three choices in GYRE:
  - MAGNUS_GL2 – 2nd order (Gabriel & Noels approach)
  - MAGNUS_GL4 – 4th order
  - MAGNUS_GL6 – 6th order
Stellar oscillation is an eigenproblem

- The oscillation equations appear to be overdetermined:
  - 4 differential equations (adiabatic case)
  - 4 boundary conditions
  - 1 arbitrary normalization condition

- The BVP can only be solved at discrete values of the oscillation frequency $\omega$ appearing in the Jacobian matrix

- These discrete values are the eigenfrequencies; the corresponding solutions are the eigenfunctions
Castor’s method

- Replace one of the boundary conditions with the normalization condition
- The BVP can then be solved for any value of the frequency $\omega$
- Use the neglected boundary condition to define a discriminant function $D(\omega)$, such that $D$ is zero when the boundary condition is satisfied
- The roots of $D(\omega)$ then correspond to the stellar eigenfrequencies
Ill-behaved discriminants: The downfall of Castor’s method

With Cowling Approximation

This problem can affect any code which involves a single-point determinant (e.g., GraCo; PULSE; ADIPLS; NOSC)
Ill-behaved discriminants: The downfall of Castor’s method

This problem can affect any code which involves a single-point determinant (e.g., GraCo; PULSE; ADIPLS; NOSC)
Recognizing the problem

- The equations plus boundary conditions can be written as a linear, homogeneous system:

\[ Su = 0 \]

\[
S = \begin{pmatrix}
B^a & 0 & 0 & \cdots & 0 & 0 \\
-\gamma^{2;1} & 1 & 0 & \cdots & 0 & 0 \\
0 & -\gamma^{3;2} & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -\gamma^{N;N-1} & 1 \\
0 & 0 & 0 & \cdots & 0 & B^b
\end{pmatrix}, \quad u = \begin{pmatrix}
y^1 \\
y^2 \\
\vdots \\
y^N
\end{pmatrix}
\]
Solution of linear, homogeneous systems

- *Any* system of linear, homogeneous equations admits non-trivial solutions ($\mathbf{u} \neq \mathbf{0}$) when the determinant of the matrix $S$ vanishes.

- Hence, the determinant can be adopted as the discriminant function:

$$D(\omega) = \det S$$

- The determinant is a polynomial in the components of $S$; if these components are well behaved, then so is $D$.
Evaluating the determinant in GYRE

- LU decompose the system matrix
  \[ S = L U \]
- Form the determinant as the diagonal product
  \[ \det S = \prod_{i} U_{i,i} \]
Dealing with determinant overflow

“For a matrix of any substantial size, it is quite likely that the determinant will overflow or underflow your computer’s floating point dynamic range”

*Numerical Recipes in Fortran, 2nd ed., “Determinant of a Matrix”*

Solution: use extended-precision arithmetic

\[ x = f \times 2^e \]

\[ f \in \mathbb{R}, \quad 0.25 < f \leq 0.5 \]

\[ e \in \mathbb{Z}, \quad |e| \leq 2147483647 \]
Summarizing the GYRE approach

- GYRE uses a *Magnus multiple shooting* (MMS) scheme for BVPs
- Multiple shooting is used for robustness & performance
- Magnus methods are used for accuracy
- A determinant-based discriminant avoids the problems of Castor’s method
- The code is parallelized with both Open MP and MPI
Both discriminants have the same roots; but the determinant-based discriminant is well behaved
Testing convergence with the $n = 0$ polytrope

For each Magnus method, the error in the eigenfrequency has the expected scaling
Inter-comparison of the g-, f- and p-modes calculated using different oscillation codes for a given stellar model

A. Moya · J. Christensen-Dalsgaard · S. Charpinet · Y. Lebreton · A. Miglio · J. Montalbán · M.J.P.F.G. Monteiro · J. Provost · I.W. Roxburgh · R. Scuflaire · J.C. Suárez · M. Suran

In all cases, departures from ESTA results are small.
g-mode inertias in a red giant model

$M = 2.0 \, M_\odot$,
$R = 11.0 \, R_\odot$,
$L = 57.8 \, L_\odot$;

cf. Dupret et al. (2009)
Example eigenfunction of the red giant model

The Magnus method readily handles the highly oscillatory eigenfunctions in the stellar core.
Nonadiabatic eigenfrequencies for a mid-B type star

The mixed adiabatic/nonadiabatic approach is numerically more robust, without sacrificing accuracy.
Rotational splitting in the $n = 0$ polytrope

$-2 \leq m \leq 2$

Non-perturbative; 10 spherical harmonic terms

Modes with $\ell = 0, 2, 4, \ldots$ all appear together
Mode tracking uses the fact that mode frequencies evolve continuously with $\Omega$.
Differential rotation: the $n = 0$ polytrope with core/envelope shear

Simple explanation: the modes are mainly trapped in the envelope
Benchmarking the parallel performance of GYRE

- 1 thread
- 2 threads
- 4 threads
- 8 threads

Tuesday, December 11, 12
The future of GYRE

- Upcoming improvements
  - implement post-processing (e.g., mode inertias, work functions)
  - combine nonadiabatic & differential rotation functionality
  - add centrifugal force, departures from sphericity

- A full description of the code will appear in a forthcoming paper

- Scheduled for open-source release mid-2013

- Pre-release access on request