AMiGO: documentation

Timothy Van Reeth

1 General

AMiGO (**A**symptotic **M**odelling of **G**-mode **O**scillations) is a python package to *(i)* calculate theoretical asymptotic g-mode period-spacing patterns for rotating stars and *(ii)* measure near-core rotation rates of observed stars by analysing their g-mode pulsations.

In AMiGO, *(i)* any mode trapping caused by the chemical structure of the star is ignored, and *(ii)* it is assumed that the g-mode pulsations are in the asymptotic regime, that is, have frequencies $\omega \ll N$, where ω is the angular pulsation frequency in the corotating frame and N is the Bruntäisälä frequency. Moreover, the Traditional Approximation of Rotation (TAR) is used: the horizontal component of the rotation vector is ignored in the equation of motion. Finally, unless otherwise specified, the star is assumed to be uniformly rotating and spherically symmetric.

AMiGO combines multiple algorithms, which have been described in separate scientific publications (listed in Section 5 below). We refer the user to these publications and the references therein for a more detailed of the scientific framework(s).

AMiGO allows the user to:

- determine the mode identification of observed g-mode period spacing patterns.
- measure (uniform) near-core rotation rates of observed stars by fitting their observed g-mode period spacing patterns.
- measure (uniform) near-core rotation rates of observed stars by fitting individual observed g-mode periods.
- calculate the effects of radially differential rotation on g-mode period-spacing patterns.
- account for the effects of the weak centrifugal acceleration on g-mode pulsation periods in a uniformly rotating star.

2 Installation instructions

Prerequisites

AMiGO is a python package that requires:

- Python ≥ 3.9 and < 3.13,
- Poetry for python package dependency management,
- the virtual Python environment manager of your choice (e.g., Conda),
- GYRE v6.x or newer.

Installing AMiGO

- When the prerequisites are met, the git repository can be cloned into a directory `<dir>` of your choice by typing these commands into a terminal:

```
$ cd <dir>
$ git clone https://github.com/TVanReeth/amigo.git amigo
```

- Add the directory `<dir>` as a Python path to your `~/.bashrc` file:

```
$ export PYTHONPATH="${PYTHONPATH}:<dir>"
```

- Activate the python virtual environment in which you want to install the required python packages. To avoid possible conflicting dependencies, we advice to build and activate a custom environment. E.g., with conda this can be done by typing:

```
$ conda create -n amigo_py python=3.9
$ conda activate amigo_py
```

- Use Poetry to install the required Python packages with all their dependencies.

```
$ cd amigo
$ poetry install
```

- Modify the parameters in the configuration file `<dir>/amigo/defaults/config.dat` as needed.

- **Optional:** include the following alias in your `~/.bashrc` file:

```
alias amigo='python <dir>/amigo/amigo/compute_rotation.py'
```

  Throughout the rest of this documentation, it is assumed that this alias command has been defined.

3 Tutorials

Various use cases of AMiGO are demonstrated in the tutorials below. In tutorials 1 to 3, AMiGO is run as a standalone software package, while in tutorials 4 to 7 selected subroutines are called from a python script. **It is assumed that the python environment in which AMiGO was installed, is activated. The directory `<dir>` mentioned below is the (same) directory in which AMiGO was installed in Section 2.**

Throughout these tutorial descriptions it is also assumed that the user has read and understands the associated publications. The tutorials are not meant to teach about the physical processes inside these stars.

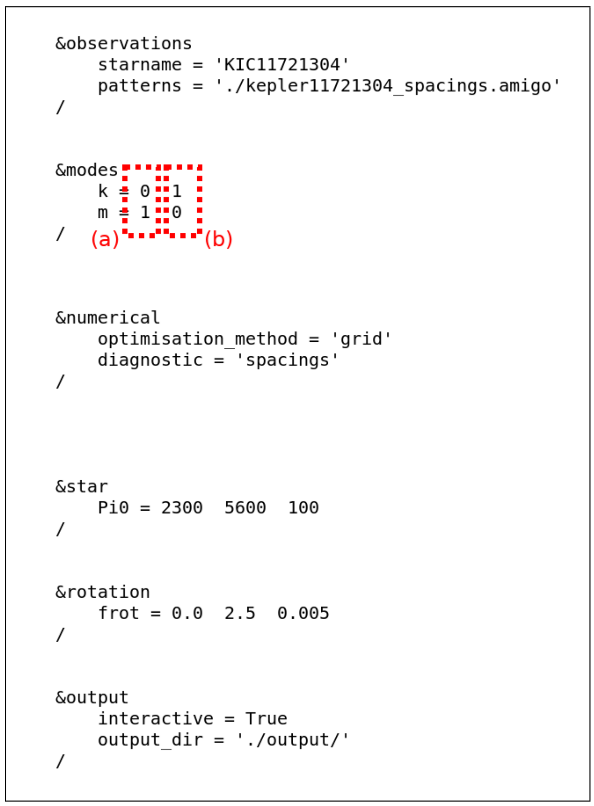
Tutorial 1: mode identification of observed g modes

The algorithms used in this tutorial are explained in Van Reeth et al. 2016.

In this tutorial, we identify the mode geometry of the observed period-spacing pattern of the γ Doradus star KIC 11721304, by fitting theoretical asymptotic period-spacings to it. It is assumed that the star is uniformly rotating.

Here we run AMiGO as a standalone package by typing

```
$ cd <dir>/amigo/tutorials/tutorial1_mode-identification/
$ amigo ./tutorial1_inlist.dat
```



```
&observations
    starname = 'KIC11721304'
    patterns = './kepler11721304_spacings.amigo'
/

&modes
    k = 0 1
    m = 1 0
/

&numerical
    optimisation_method = 'grid'
    diagnostic = 'spacings'
/

&star
    Pi0 = 2300  5600  100
/

&rotation
    frot = 0.0  2.5  0.005
/

&output
    interactive = True
    output_dir = './output/'
/
```


Figure 1: Inlist of tutorial 1. The different variables are explained in the text.

The input parameter values are all given in the inlist, as shown in Fig. 1. We consider two possible pulsation mode geometries, $(k, m) = (0, 1)$ and $(1, 0)$, as indicated in Fig. 1 with (a) and (b). Because the observed period spacings decrease as a function of pulsation period, as shown in Fig. 2, and the spacings have values up to 1500 s, these are the most likely mode geometries. We calculate asymptotic g-mode patterns assuming uniform rotation rates $f_{\rm rot}$ between $0\,{\rm d}^{-1}$ and $2.5\,{\rm d}^{-1}$ and buoyancy radii Π_0 with values between 2300 s and 5600 s, which are typical for γ Doradus stars. This parameter space is evaluated in a grid-based approach, whereby the spacings between consecutive $f_{\rm rot}$ and Π_0 values are $0.005\,{\rm d}^{-1}$ and 100 s, respectively. The model patterns are compared to the observations by fitting the spacings between consecutive pulsation periods as a function of the periods. The results are shown interactively and saved in the directory "./output/KIC11721304/". A more detailed description of the AMiGO inlist parameters is given in Section 4.

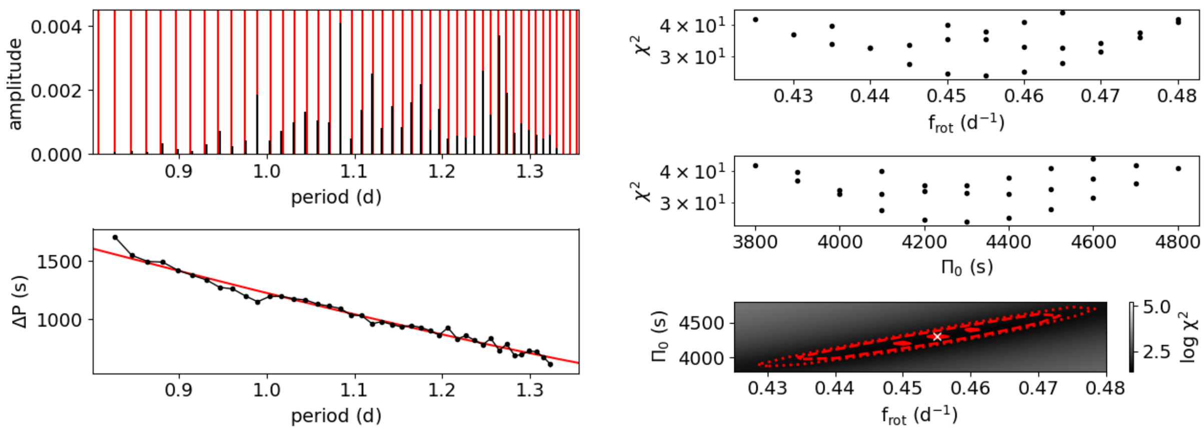


Figure 2: Best-fitting g-mode model for the observed pattern of KIC 11721304 *Top left:* best-fitting asymptotic g-mode pattern (red) with the observed g modes (black). *Bottom left:* spacings between consecutive pulsation periods in the patterns as a function of the pulsation period. *Top right:* marginalised $chi^2_{\rm red}$ values as a function of the best evaluated rotation frequencies. *Middle right:* marginalised $chi^2_{\rm red}$ values as a function of the best evaluated buoyancy radii. *Bottom right:* $chi^2_{\rm red}$ values for the the best evaluated Π_0 and $f_{\rm rot}$ values, with the $1-$, $2-$, and $3-\sigma$ contours shown in red.

The best-fitting pattern, found for $(k, m) = (0, 1)$, is shown in Fig. 2. It is found for

$$
\begin{aligned}
f_{\rm rot} &= 0.455 \pm 0.010\,{\rm d}^{-1}, \\
\Pi_0 &= 4300 \pm 170\,s, \\
{\rm estimated}\alpha &= 0.521, {\rm with} \\
\chi^2_{\rm red} &= 25.095.
\end{aligned}
$$

By contrast, the best-fitting solution for $(k, m) = (1, 0)$ has $\chi^2_{\rm red} = 120.788$, showing that $(k, m) = (0, 1)$ is the true mode identification and the corresponding near-core rotation and buoyancy radius values are true. The phase term α_g, dependent on the g-mode cavity boundaries, is not taken into account during the model optimisation, because the 'spacings' diagnostic is used. However, it is calculated a posteriori for the best-fitting model and used in Fig. 2.

Tutorial 2: measuring near-core rotation from multiple period-spacing patterns

The algorithms used in this tutorial are explained in Van Reeth et al. (2016).

In this tutorial, we measure the near-core rotation rate and buoyancy radius of the γ Doradus star KIC 3228863 by fitting theoretical asymptotic period-spacings to the observed period-spacing patterns. It is assumed that the star is uniformly rotating, and to speed up the tutorial, it is assumed that the mode geometries $(k, m) = (0, 1)$ and $(-2, -1)$ are already known.

Here we run AMiGO as a standalone package by typing

```
$ cd <dir>/amigo/tutorials/tutorial2_fitting-multiple-patterns/
$ amigo ./tutorial2_inlist.dat
```

```
&observations
    starname = 'KIC03228863'
    patterns = './kepler03228863_spacings.amigo'
/

&modes              pattern 1
    k =  0
    m =  1
/

&modes              pattern 2
    k = -2
    m = -1
/

&numerical
    optimisation_method = 'lmfit'
    diagnostic = 'spacings'
/



&star
    Pi0 = 2300  5600  100
/


&rotation
    frot = 0.0  2.5  0.005
/


&output
    interactive = True
    output_dir = './output/'
/
```

Figure 3: Inlist of tutorial 2. The different variables are explained in the text.

The input parameter values are all given in the inlist, as shown in Fig. 3. We consider the pulsation mode geometries $(k, m) = (0, 1)$ and $(-2, -1)$ for the two observed patterns in the file "kepler03228863_spacings.amigo", as indicated in Fig. 1. The two &modes environments correspond to the two observed patterns in the file. We calculate asymptotic g-mode patterns assuming uniform rotation rates f_{rot} between $0\,\mathrm{d}^{-1}$ and $2.5\,\mathrm{d}^{-1}$ and buoyancy radii Π_0 with values between $2300\,s$ and $5600\,s$, which are typical for γ Doradus stars. This parameter space is evaluated using the Levenberg-Marquardt algorithm, as implemented in the Python package lmfit. The model patterns are compared to the observations by fitting the spacings between consecutive pulsation periods as a function of the periods. The results are shown interactively and saved in the directory "./output/KIC03228863/". A more detailed description of the AMiGO inlist parameters is given in Section 4.

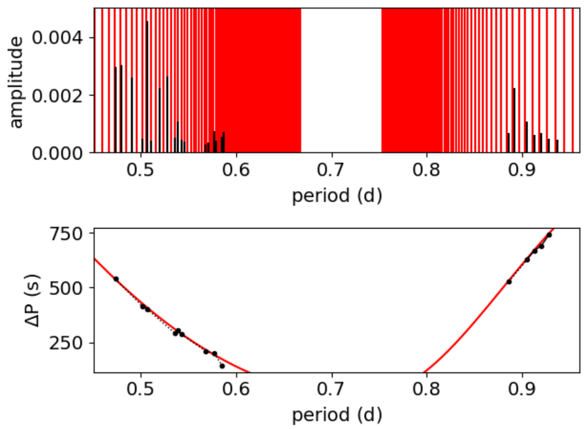


Figure 4: Best-fitting g-mode model for the observed pattern of KIC 11721304 *Top:* best-fitting asymptotic g-mode patterns (red) with the observed g modes (black) for $(k, m) = (0, 1)$ (left) and $(k, m) = (-2, -1)$ (right). *Bottom:* spacings between consecutive pulsation periods in the patterns as a function of the pulsation period.

The best-fitting patterns, shown in Fig. 4, are found for

$$\begin{aligned} f_{\mathrm{rot}} &= 1.369 \pm 0.002\,\mathrm{d}^{-1}, \\ \Pi_0 &= 4451 \pm 30\,s, \\ \mathrm{estimated}\alpha_1 &= 0.885, \\ \mathrm{estimated}\alpha_2 &= 1.182, \mathrm{with} \\ \chi^2_{\mathrm{red}} &= 311.469. \end{aligned}$$

The phase term α_g, dependent on the g-mode cavity boundaries, is not taken into account during the model optimisation, because the 'spacings' diagnostic is used. However, it is calculated a posteriori for the best-fitting model and used to calculate the model shown in the top left panel of Fig. 4.

Tutorial 3: measuring near-core rotation from sparse g modes

The algorithms used in this tutorial are explained in Van Reeth et al. 2022.

In this tutorial, we measure the near-core rotation rate and buoyancy radius of the γ Doradus star HD 112429 by fitting theoretical asymptotic period-spacings to the observed period-spacing patterns. It is assumed that the star is uniformly rotating, and to speed up the tutorial, it is assumed that the mode geometries $(k, m) = (0, 1)$ and $(-2, -1)$ are already known. The difference with KIC 3228863, analysed in tutorial 2, is that the detected g-mode patterns of HD112429 are too sparse to calculate and model the period spacings. Instead, we estimate the radial orders of the observed g-mode pulsations, and model the pulsation frequencies as a function of the estimated radial orders.

Here we run AMiGO as a standalone package by typing

```
$ cd <dir>/amigo/tutorials/tutorial3_fitting-sparse-patterns/
$ amigo ./tutorial3_inlist.dat
```

```
&observations
    starname = 'HD_112429'
    patterns = './HD112429_spacings.amigo'
/


&modes
    k =  0
    m =  1
/


&modes
    k = -2
    m = -1
/


&numerical
    optimisation_method = 'iterative'
    diagnostic = 'frequency'
    use_sequence = False
    sigma_sampling = 5
    grid_scaling = 10

    cvg_rate = 1.5
factor.
/

&star
    Pi0 = 2300  5600  50
/


&rotation
    frot = 1.4  1.9  0.01
/


&output
    interactive = True
    output_dir = './output/'
/
```

Figure 5: Inlist of tutorial 3. The different variables are explained in the text.

The input parameter values are all given in the inlist, as shown in Fig. 5. We consider the pulsation mode geometries $(k, m) = (0, 1)$ and $(-2, -1)$ for the two observed patterns in the file "HD112429_spacings.amigo", as indicated in Fig. 1. The two &modes environments correspond to the two observed patterns in the file. To speed up the calculations, we calculate asymptotic g-mode patterns assuming uniform rotation rates $f_{\rm rot}$ between $1.6\,\mathrm{d}^{-1}$ and $1.9\,\mathrm{d}^{-1}$, where the boundaries are determined by the observed g- and r-modes, and buoyancy radii Π_0 with values between $2300\,s$ and $5600\,s$, which are typical for γ Doradus stars. This parameter space is evaluated using the "iterative grid" approach. First, the parameter space is sampled coarsely, using the parameter steps given in the inlist. This grid sampling is then iteratively refined at the rate set by the "cvg_rate" parameter. At each step, the grid range ΔX for parameter X is given by

$$\Delta X = 2 \times \text{sigma\_sampling} \times \text{grid}_{\text{s}}\text{caling} \times \mathrm{d}X,$$

where $\mathrm{d}X$ is the current grid step size. The iterative resampling is stopped when the uncertainty

$$\sigma_X \geq \text{sigma\_sampling} \times \mathrm{d}X.$$

The model patterns are compared to the observations by fitting the model frequencies corresponding to the estimated radial orders to the observed frequencies. The results are shown interactively and saved in the directory "./output/HD_112429/". A more detailed description of the AMiGO inlist parameters is given in Section 4.

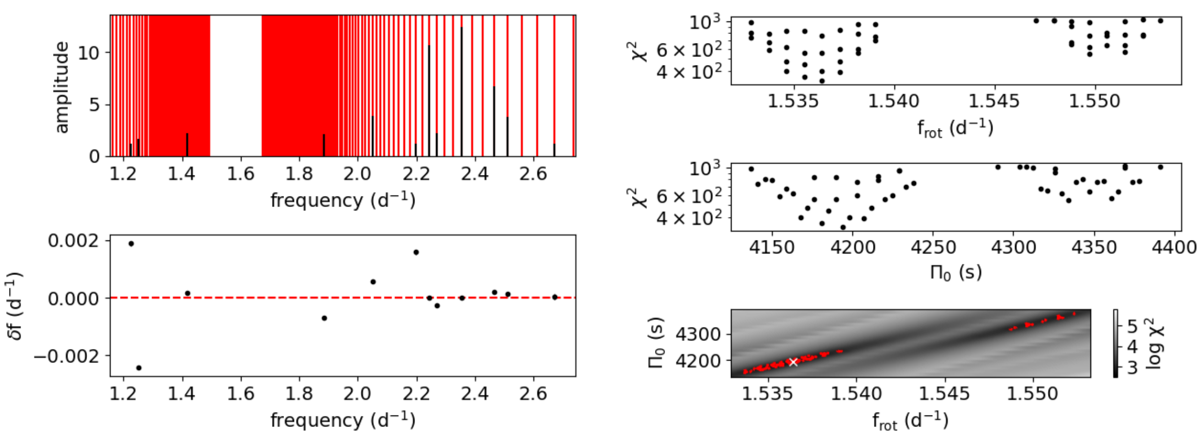


Figure 6: Best-fitting g-mode model for the observed patterns of HD 112429 *Top left:* best-fitting asymptotic g-mode patterns (red) with the observed g modes (black) for $(k, m) = (0, 1)$ (left) and $(k, m) = (-2, -1)$ (right). *Bottom left:* residuals between the observed pulsation frequencies and model frequencies that correspond to the estimated radial orders of the observed pulsations. *Top right:* marginalised $chi^2_{\rm red}$ values as a function of the best evaluated rotation frequencies. *Middle right:* marginalised $chi^2_{\rm red}$ values as a function of the best evaluated buoyancy radii. *Bottom right:* $chi^2_{\rm red}$ values for the the best evaluated Π_0 and $f_{\rm rot}$ values, with the $1-$, $2-$, and $3-\sigma$ contours shown in red.

The best-fitting patterns, shown in Fig. 6, are found for

$$\begin{aligned}
f_{\rm rot} &= 1.536 \pm 0.005\,{\rm d}^{-1},\\
\Pi_0 &= 4194 \pm 61\,s,\\
\alpha_1 &= 0.90 \pm 0.50,\\
\alpha_2 &= 1.14 \pm 0.04, \text{with}\\
\chi^2_{\rm red} &= 331.167.
\end{aligned}$$

Here, the phase terms α_1 and α_2, dependent on the g-mode cavity boundaries, correspond to the two observed patterns. They are taken into account and calculated during the model optimisation, because the 'frequency' diagnostic is used.

As can be seen in the right-hand side panels of Fig. 6, the $\chi^2_{\rm red}$-distribution has a local minimum at higher $f_{\rm rot}$ and Π_0 values. This second solution corresponds to different estimated radial orders then those assigned for the best solution.

Tutorial 4: calculating asymptotic g-mode models for a uniformly rotating spherical star

The algorithms used in this tutorial are explained in Van Reeth et al. (2016).

In this tutorial, we calculate g-mode model patterns (with (k,m) = (0,1)) for a uniformly rotating star, given an input value for the stellar buoyancy radius Π_0 and assuming different rotation frequencies $f_{\rm rot}$.

This tutorial can be run by typing

```
$ cd <dir>/amigo/tutorials/tutorial4_uniform-rotation-gmodes/
$ python tutorial4_uniform-rotation-gmodes.py
```

The results of this tutorial are shown interactively, and illustrated in Fig. 7. The python subroutines that are called in this python script are also part of the back-end that was used in tutorials 1, 2 and 3.

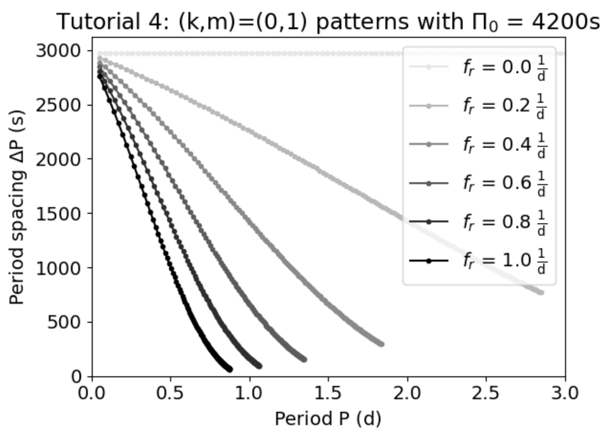


Figure 7: Asymptotic g-mode period-spacing patterns, calculated for mode identification $(k,m) = (0,1)$ and buoyancy radius $\Pi_0 = 4200\,s$, and for increasing near-core rotation rates $f_{\rm rot}$.

Tutorial 5: calculating asymptotic r-mode models for a uniformly rotating spherical star

The algorithms used in this tutorial are explained in Van Reeth et al. (2016) and Van Reeth et al. (2018).

In this tutorial, we calculate r-mode model patterns (with (k,m) = (-2,-1)) for a uniformly rotating star, given input values for the stellar buoyancy radius Π_0 and near-core rotation frequencies $f_{\rm rot}$. This tutorial is complementary to tutorial 4, because:

- the used input value for the stellar buoyancy radius Π_0 is calculated for a given stellar structure model (with a stellar mass of $1.5\,M_\odot$, metallicity $Z = 0.014$, core overshooting $f_{\rm ov} = 0.015$ and a core hydrogen mass fraction $X = 0.5$),
- this tutorial demonstrates the calculation of Rossby modes (or r-modes) rather than "normal" gravito-inertial modes.

Tutorial 5 can be run by typing

```
$ cd <dir>/amigo/tutorials/tutorial5_uniform-rotation-rmodes/
$ python tutorial5_uniform-rotation-rmodes.py
```

The results of this tutorial are then shown interactively, and illustrated in Fig. 8. In AMiGO, pulsation frequencies $f_{\rm co}$ in the co-rotating reference frame are always positive. However, r modes are retrograde and have frequencies $f_{\rm co} < |m|f_{\rm rot}$, so that the calculated frequencies in the inertial reference frame $f_{\rm in} = f_{\rm co} + mf_{\rm rot} < 0$. Physically, this means that because the r mode travels so slowly in the direction opposite to the stellar rotation, it looks *as if the r mode is travelling in the direction of the rotation, i.e. as if it is prograde,* to an observer in the inertial reference frame. This is shown in the top panel of Fig. 8. However, pulsation frequencies (and periods) are always measured as positive values by an observer. Hence, to match a theoretical r-mode pattern to an observed one, we first need to take the absolute values of the pulsation frequencies (or periods). This is illustrated in the bottom panel of Fig. 8.

The python subroutines that are called in this python script are also part of the back-end that was used in tutorials 2 and 3.

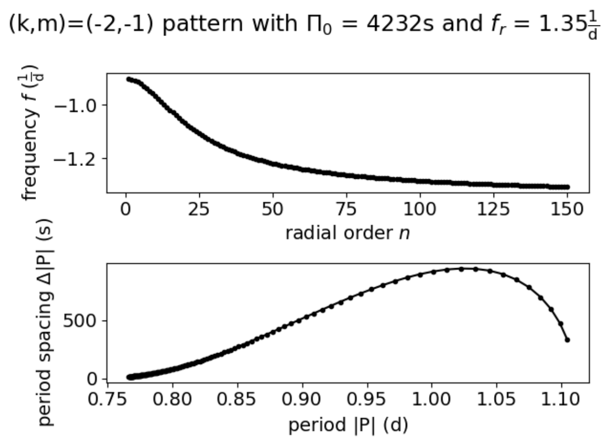


Figure 8: r-mode pulsation pattern with $(k, m) = (-2, -1)$, assuming a buoyancy radius Π_0 value of $4232\,s$ and a near-core rotation frequency $f_{\rm rot}$ of $1.35\,\mathrm{d}^{-1}$. *Top:* the calculated pulsation frequency f in the inertial reference frame, as a function of the radial order n. Note that $f < 0$. *Bottom:* period-spacings $\Delta|P|$ between consecutive pulsation periods $|P|$ in the calculated pattern, as seen by an observer. Because measured pulsation frequencies and periods are always positive, we use $|P|$ rather than P.

Tutorial 6: calculating asymptotic models for a differentially rotating spherical star

The algorithms used in this tutorial are explained in Van Reeth et al. 2018 and the corrigendum (included with the documentation of AMiGO).

In this tutorial, we calculate g-mode period-spacing patterns of a radially differential rotating star. Contrary to the calculations in the previous tutorials, where the use of an input stellar structure model was optional, it is required here: the differential rotation leads to a radial dependence of the Coriolis force, which competes with the buoyancy throughout the star. For simplicity, we use the same stellar structure model as in tutorial 5 (with a stellar mass of $1.5\,M_\odot$, metallicity $Z = 0.014$, core overshooting $f_{\rm ov} = 0.015$ and a core hydrogen mass fraction $X_c = 0.5$).

The differential rotation profile $f_{\rm rot}(r)$ is defined following Eq.(10) in Van Reeth et al. (2018), whereby we consider the stellar rotation rate at a radial coordinate r_s, where the g modes are dominant, and scale the relative differential rotation δf between r_s and the stellar centre. Subsequently, we calculate both g- and r-mode period-spacing patterns using full numerical integration (i.e., Eq.(3) in Van Reeth et al. (2018)) and using a Taylor expansion (as explained in the corrigendum and the subsection below.)

Tutorial 6 can be run by typing

```
$ cd <dir>/amigo/tutorials/tutorial6_differentially-rotating-models/
$ python tutorial6_differentially-rotating-models.py
```

The results of this tutorial are then shown interactively, and illustrated in Fig. 9. On the left-hand side, we see three differential rotation profiles in the top panel, and the corresponding g- and r-mode patterns in the middle and bottom panels, calculated using full numerical integration. However, these calculations require a known Brunt-Väisälä frequency profile $N(r)$ and a known differential rotation profile $f_{\rm rot}(r)$.

In an effort to facilitate the modelling of observed patterns, for which these patterns are not known beforehand, we also rewrote the expression for asymptotic period-spacing patterns using a Taylor expansion. The resulting expression, given below in Eq.(1), contains functions F_1 and F_2 that are evaluated in the radial coordinate r_s. The idea was that, when observed patterns are modelled, the coefficients c_1 and c_2 can be determined numerically. Unfortunately, it has become clear that the Taylor expansion is not sufficiently accurate: as shown below, c_1 and c_2 can also be computed directly from theoretical rotation profiles, but the resulting period-spacing patterns are very different from the ones calculated with full numerical integration (see Fig. 9).

Practical notes on the Taylor expansion for differential rotation

Using the Taylor expansion, the expression for an asymptotic period-spacing pattern in a star with radially differential rotation is rewritten as

$$2 f_{\rm rot}\Pi_0 \left(n + \alpha_g\right) \approx \sqrt{\lambda_{\nu k m,s}}\nu \left\{1 + c_1 F_1 + c_2 F_2\right\}, \tag{1}$$

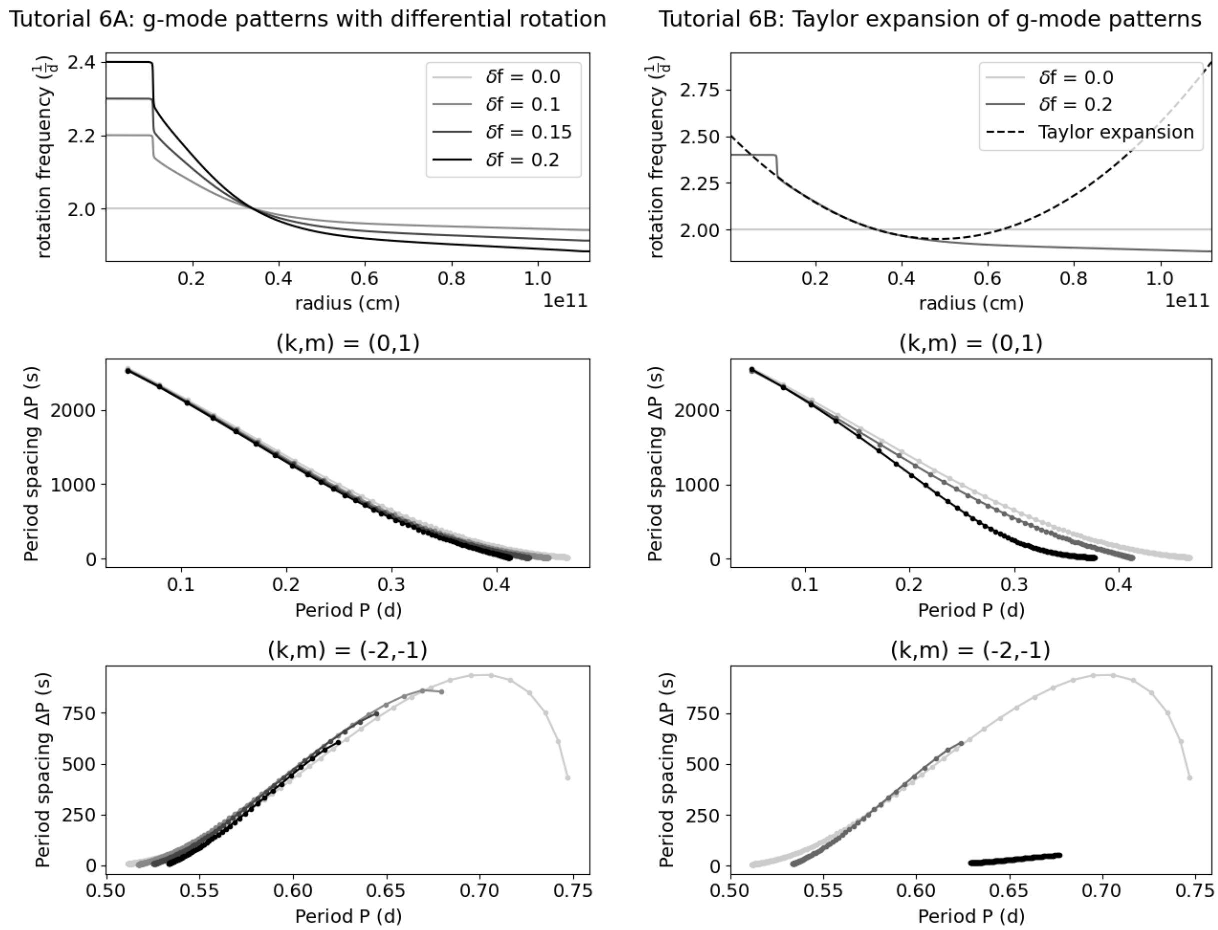


Figure 9: G-mode patterns calculated for a $1.5\,M_\odot$ star with radially differential rotation. *Left:* Results from calculations with full numerical integration. *Top left:* differential rotation profiles, following the definition in Eq.(10) of Van Reeth et al. (2018). *Middle left:* g-mode period-spacing patterns with $(k, m) = (0, 1)$ for the rotation profiles shown above. . *Bottom left:* r-mode period-spacing patterns with $(k, m) = (-2, -1)$ for the rotation profiles shown above. *Right:* Results from approximations using a Taylor expansion. *Top right:* comparison of rigid rotation (light grey), a differential rotation profile (dark grey) and the 2^{nd}-order approximation of the rotation profile. *Middle right:* g-mode period-spacing patterns with $(k, m) = (0, 1)$ for the rotation profiles and the 2^{nd}-order approximation shown above, calculated using a Taylor expansion. *Bottom right:* r-mode period-spacing patterns with $(k, m) = (-2, -1)$ for the rotation profiles shown above.

where the functions

$$F_1 = \frac{\nu}{2}\left[m + \frac{1}{\lambda}\frac{\mathrm{d}\lambda}{\mathrm{d}\nu}\left(1 + \frac{m\nu}{2}\right)\right],$$

$$F_2 = \frac{\nu^2}{4}\left[2m^2 + \frac{3m}{\lambda_s}\frac{\mathrm{d}\lambda}{\mathrm{d}\nu}\left(1 + \frac{m\nu}{2}\right) + \left(\frac{2}{\lambda}\frac{\mathrm{d}^2\lambda}{\mathrm{d}\nu^2} - \left(\frac{1}{\lambda}\frac{\mathrm{d}\lambda}{\mathrm{d}\nu}\right)^2\right)\left(1 + \frac{m\nu}{2}\right)^2\right]$$

are evaluated at the radial coordinate r_s. The coefficients c_1 and c_2 can be calculated for a rotation profile $f_{\rm rot}(r)$ as

$$c_1 = \frac{f'_{\rm rot,s}}{f_{\rm rot,s}}\langle r - r_s\rangle + \frac{1}{2}\frac{f''_{\rm rot,s}}{f_{\rm rot,s}}\langle (r - r_s)^2\rangle$$

and

$$c_2 = \frac{1}{2}\left(\frac{f'_{\rm rot,s}}{f_{\rm rot,s}}\right)^2 \langle (r - r_s)^2\rangle.$$

Tutorial 7: calculating asymptotic models for a uniformly rotating, centrifugally deformed star

The algorithms used in this tutorial are explained in Mathis & Prat (2019) and Henneco et al. (2021).

In this tutorial, we calculate the effects of weak centrifugal acceleration on stellar structure (starting froma non-rotating stellar structure model), and determine the effects on asymptotic period-spacing patterns. For simplicity, we again use the same stellar structure model as in tutorials 5 and 6 (with a stellar mass of $1.5\,M_{\odot}$, metallicity $Z = 0.014$, core overshooting $f_{\rm ov} = 0.015$ and a core hydrogen mass fraction $X_c = 0.5$). In this tutorial, we consider an angular rotation frequency $\Omega_rot = 0.4\,\Omega_{\rm crit,Roche}$, where $\Omega_{\rm crit,Roche}$ is the critical rotation frequency of the stellar structure model in the Roche approximation.

Tutorial 7 can be run by typing

```
$ cd <dir>/amigo/tutorials/tutorial7_centrifugal-acceleration/
$ python tutorial7_centrifugal-acceleration.py
```

The results are shown in Fig. 10. Note that, because we only take into account weak centrifugal acceleration, these subroutines should not be used for (too) fast-rotating stars. Moreover, as we can see in the middle and bottom panels of Fig. 10, there is a numerical artefact at the high radial-order end of the spacing patterns, whereby the period spacing drastically drops. This should be excluded from the results when using these subroutines.

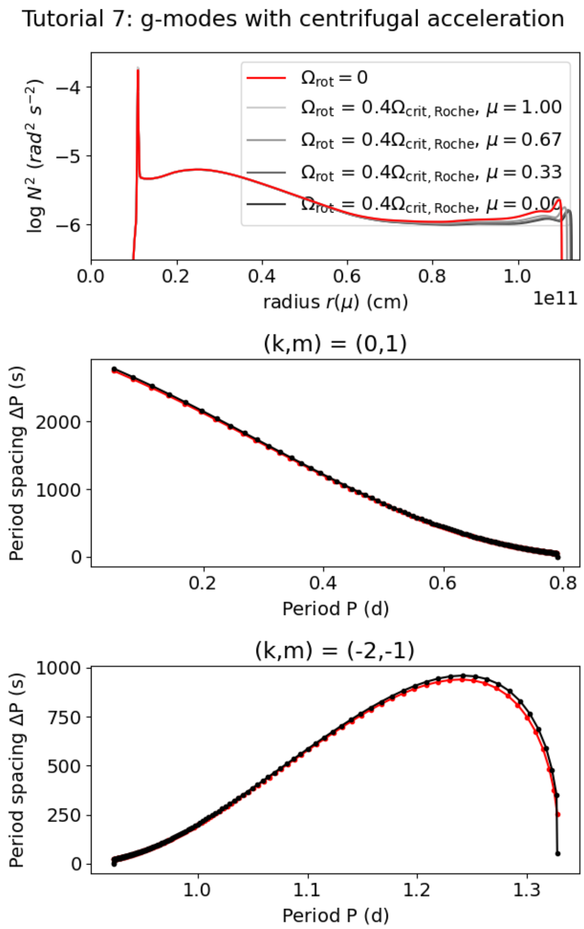


Figure 10: *Top:* The Brunt-Väisälä frequency profile $N(r)$ of the non-rotating stellar model (red) and of the centrifugally deformed model at different values of $\mu = \cos\theta$, where θ is the colatitude. *Middle:* the g-mode period-spacing pattern with $(k, m) = (0, 1)$ for both the non-rotating (red) and the centrifugally deformed (black) stellar models. *Bottom:* the r-mode period-spacing pattern with $(k, m) = (-2, -1)$ for both the non-rotating (red) and the centrifugally deformed (black) stellar models.

4 AMiGO inlist

When AMiGO is used as a standalone software package, the parameters required to run the code have to given in an inlist, as illustrated in tutorials 1, 2, and 3. Here, the variables and quantities are organised into different environments. We have:

1. **the &observations environment:** with variables describing the observed star and asteroseismic observations

   - *starname*: the name of the star, with "_" characters replacing any spaces and single quotation marks ('') indicating the beginning and the end of the string. This name will be used in the filenames of the AMiGO output.
   - *patterns*: path to the file with detected g-mode patterns. Each line corresponds to a single pulsation, and the different parameter values are sorted into columns. Per pulsation pattern, consecutive lines are sorted in order of increasing pulsation period, and a gap in the pattern is indicated by a line where individual parameter values are replaced by "–". The end of a pulsation pattern is indicated by a line filled with "*****". The pulsation parameters include:
     - per: the pulsation period
     - e_per: 1σ-uncertainty on the pulsation period
     - ampl: the pulsation amplitude
     - e_ampl: 1σ-uncertainty on the pulsation amplitude
     - phase: the pulsation phase
     - e_phase: 1σ-uncertainty on the pulsation phase
     - stopcrit: the measured S/N-value of the pulsation

2. **the &modes environment:** contains the mode identifications (k,m) that are considered for the model patterns to be fit to the observations. Each &modes environment corresponds to one observed pattern.

   - *k*: the meridional degrees k. For a "traditional" gravito-inertial mode, i.e., one that has both the Coriolis force and the buoyancy as its dominant restoring forces and can exist in non-rotating stars, this parameter is $k = l - |m|$, where l is the spherical degree of the mode. For an inertial pulsation mode, which has the Coriolis force as its dominant restoring force and does not exist in a non-rotating star, $k < 0$.
   - *m*: the corresponding azimuthal orders m.

3. **the &numerical environment**: with variables describing the chosen numerical settings for the model optimisation.

   - *optimisation_method*: the numerical method used to sample and evaluate the parameter space. This parameter is a string, and can have the values:
     - 'grid': The evaluated parameter space is sampled as a regular (regular) grid. This method is illustrated in tutorial 1.

  - 'iterative': The evaluated parameter space is sampled as a regular (regular) grid, which is then iteratively refined around the best solution found within the grid. This method is illustrated in tutorial 2. While it is computationally much cheaper than the 'grid' method, it is more prone to get stuck in a secondary minimum of the parameter space when the 'frequency' diagnostic is used.
  - 'lmfit': The parameter space is evaluated using the Levenberg-Marquardt method, as implemented in the Python package lmfit. This method is illustrated in tutorial 3. It is quite fast and computationally cheap, but also more prone to get stuck in a secondary minimum of the paraameter space.

- *diagnostic*: the quantity used to evaluate the quality of the fit models. This parameter is a string, and can have the values:
  - 'spacings': the spacings between consecutive pulsation periods are evaluated as a function of the pulsation period. This diagnostic is numerically very stable and provides reliable results, but requires the observed pulsation patterns to be (near-)continuous and does not allow for many (large) gaps.
  - 'frequency': the pulsation frequencies are evaluated as a function of the (estimated) radial orders. This diagnostic can be used to model sparse g mode patterns with (large) gaps, but is numerically expensive and more prone to get stuck in a secondary minima in the parameter space. It may also lead to an underestimation of the parameter value uncertainties, because the dependence on the radial order identification results in a multimodal χ^2-dependence in the parameter space.

- *use_sequence*: This parameter is a boolean, and only relevant when the 'frequency' diagnostic is used. When it is set to True, any gaps in the observed g-mode pattern are sufficiently small that the number of missing g modes within each gap can be counted. If it is set to False, the gaps are (too) large, and the true number of missing g modes within each gap is unknown. In this case, AMiGO will estimate the number of missing modes based on the fit models. Unfortunately, this often leads to a high level of (systematic) degeneracy in the found solutions, making them less reliable.

- *sigma_sampling*: This parameter is only used when optimisation_method is 'iterative'. It is the minimum number of projected data points that has to be calculated per 1-σ error margin for each parameter X, before the iterative model optimisation is stopped.

- *grid_scaling*: This parameter is only used when optimisation_method is 'iterative'. It is the scaling factor for the required grid size at each iteration: at each step, the grid is centered around the current best solution, with ranges for the parameter X given by
$$\Delta X = 2 \times \text{sigma\_sampling} \times \text{grid\_scaling} \times \delta X,$$
where δX is the grid step size for the parameter X at the current iteration.

- *cvg_rate*: This parameter is only used when optimisation_method is 'iterative'. It is the convergence rate of the grid during the iteration. At each iteration, the parameter step size δX in the grid is decreased by this factor.

4. **the &star environment**: contains the variables related to the stellar structure

   - *Pi0*: the buoyancy radius or the buoyancy travel time, which can be calculated from a stellar structure model as

   $$\Pi_0 = \frac{2\pi^2}{\int_{r_1}^{r_2} \frac{N}{r} \mathrm{d}r},$$

     whereby the Brunt-Vïsälä frequency profile $N(r)$ is integrated within the boundaries of the g mode cavity r_1 and r_2. In the inlist, the minimum value, maximum value, and (initial) step size of the parameter within the grid have to be given in units of seconds.

5. **the &rotation environment**: contains the variables describing the evaluated rotation rates

   - *frot*: the (assumed uniform) near-core stellar rotation rate. In the inlist, the minimum value, maximum value, and (initial) step size of the parameter within the grid have to be given in units of d^{-1}.

6. **the &output environment**: contains the variables where the output of AMiGO is stored.

   - *output_dir*: path to the directory in which the results will be saved, which will be in a subdirectory <output_dir>/<starname>/, with the variable starname given in the &observations environment. This parameter is a string.

5 Acknowledgement requirements

Please cite the following publications when:

- determining the mode identification of observed g-mode period spacing patterns.
  - Van Reeth et al. 2016, A&A 593, A120
- measuring (uniform) near-core rotation rates of observed stars by fitting their observed g-mode period spacing patterns.
  - Van Reeth et al. 2016, A&A 593, A120
- measuring (uniform) near-core rotation rates of observed stars by fitting individual observed g-mode periods.
  - Van Reeth et al. 2022, A&A 662, A58
- calculating the effects of radially differential rotation on g-mode period-spacing patterns.
  - Van Reeth et al. 2018, A&A 618, A24
- accounting for the effects of the weak centrifugal acceleration on g-mode pulsation periods in a uniformly rotating star.
  - Mathis & Prat 2019, A&A 631, A26
  - Henneco et al. 2021, A&A 648, A97

Moreover, AMiGO relies on the GYRE, Astropy, Matplotlib, Numpy, and Scipy software packages. Please comply with their acknowledgement requirements as well.