

NonLinearKit

Version 1.0

Daniel Reese

December 2015

Acknowledgements

This program was derived from InversionKit by Daniel Reese during his postdoctoral position at the University of Liège and later improved during his postdoctoral position at the University of Birmingham while funded by the SPACEINN network. The SPACEINN network is a major international collaboration funded by the European Commission's Seventh Framework Programme. Marc-Antoine Dupret indirectly contributed to this code by explaining the Levenberg-Marquardt algorithm to Daniel Reese.

Contents

1	Getting started	4
1.1	About the program	4
1.2	Running the program	4
1.3	Using the program	4
1.4	Preliminary remark	5
2	File formats	11
2.1	File with observed frequencies	11
2.2	Stellar models	11
2.2.1	MOD (binary/ascii)	12
2.2.2	CLES format	13
2.2.3	LOSC format	13
2.2.4	Output text format from the Structural Inversion tab	14
2.2.5	Comparison of different model formats	14
2.2.6	Useful formulas	15
2.3	GZIP compression	15
3	Treatment of pulsation calculations	16
3.1	Pulsation calculations	16
3.2	JCD's equations	16
3.2.1	Non-radial modes	16
3.2.2	Radial modes	17
3.3	Lagrangian pressure perturbations	18
3.3.1	Non-radial modes	18
3.3.2	Radial modes	19
3.4	Eulerian pressure and density perturbations	20
3.4.1	Non-radial modes	20
3.4.2	Radial modes	21
3.5	Reconstructing missing variables	21
4	Inversions	22
4.1	General description	22
4.2	Comparison with other methods	22
4.3	Levenberg-Marquardt algorithm	23
4.4	Various useful formulas	24
4.4.1	Mode inertia	24
4.4.2	c2, rho kernels	25
4.4.3	Gamma1, rho kernels	25
5	Known bugs	26
6	Copyright notices	26
6.1	Source code for reading fortran binary files	26
6.2	Supplementary notices	26

1 Getting started

1.1 About the program

`NonLinearKit` is an experimental non-linear inversion program used to deduce stellar structure from a set of observed frequencies. Given its experimental nature, this program is expected to evolve into more sophisticated versions in the future. Currently, the program only adjusts the density profile of the star while keeping the Γ_1 profile fixed as it has been found that modifying the Γ_1 profile can quickly lead to nonphysical solutions. A similar approach has been taken by I. Roxburgh and S. Vorontsov in their completely different inversion techniques.

1.2 Running the program

`NonLinearKit` runs under Java 5.0 or later versions. If Java is not installed on your computer, or is not sufficiently up-to-date, it can be downloaded from:

<http://www.java.com/en/>

JRE (Java Runtime Environment) allows you to run Java programs but not to compile your own. JDK (Java Development Kit) allows you to run and compile Java programs.

To run the program download the file `NonLinearKit.jar` from the following website:

<http://bison.ph.bham.ac.uk/~dreese/NonLinearKit/index.html>

then type the following command in a command window, in the directory that contains `NonLinearKit.jar`:

```
java -jar NonLinearKit.jar
```

If you are planning to do calculations involving large data and kernels sets, you may need to allocate a larger amount of memory to run the program. To allocate, for example, 500 MB of memory, use the following command:

```
java -Xmx500m -jar NonLinearKit.jar
```

Note: the option `-Xmx` is nonstandard and may change according to the release installed on your computer.

1.3 Using the program

Once `NonLinearKit` is running, there are several tabs which follow a logical progression:

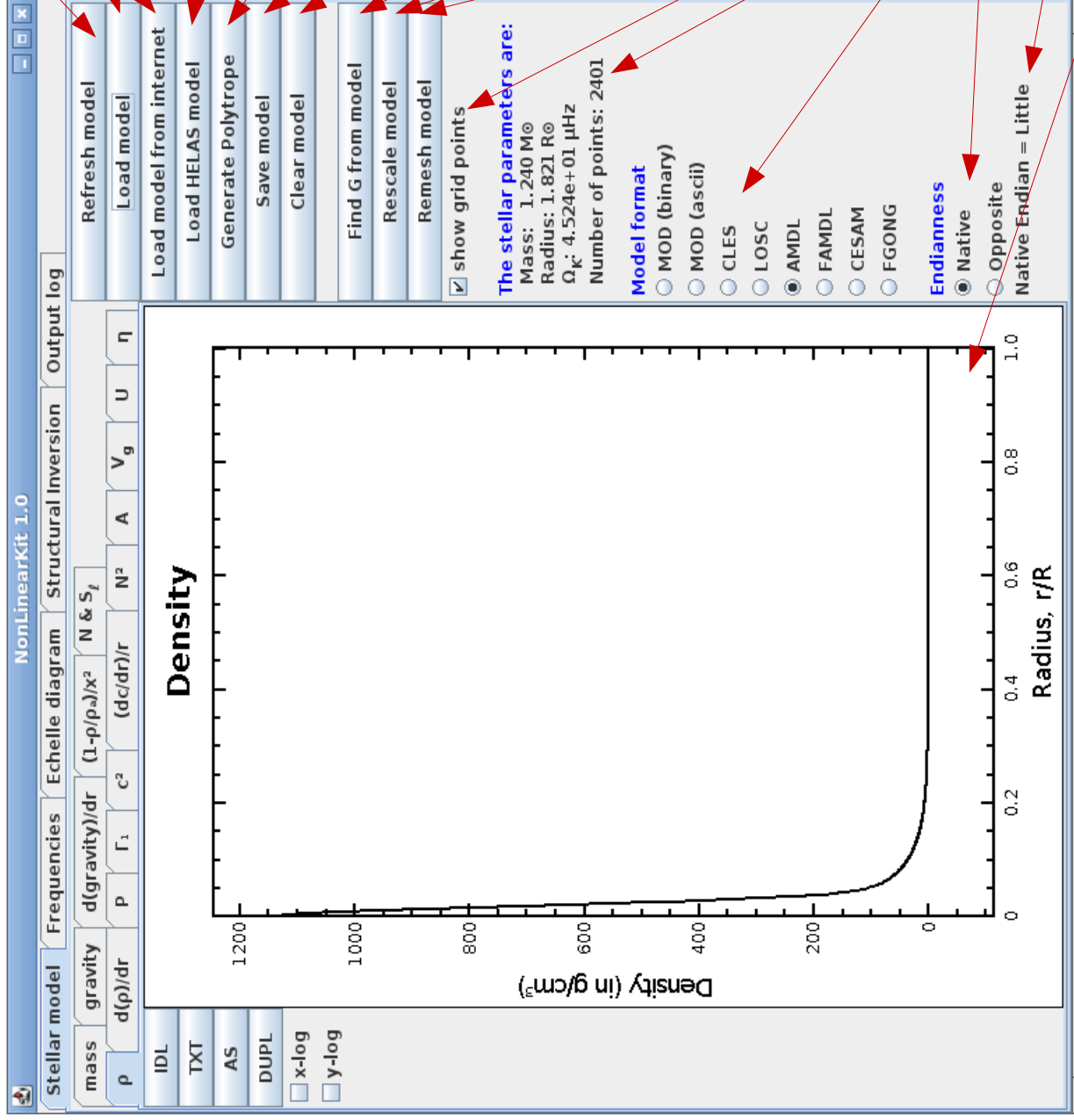
- **Stellar model tab:** this allows the user to load an initial model
- **Frequencies tab:** this allows the user to load observed frequencies (without specifying their radial orders)
- **Echelle diagram tab:** this can be used to visually compare the frequencies from the current model with the observed frequencies

- **Structural Inversion tab:** this is where the user can carry out non-linear structural inversions
- **Output log tab:** this is a log of the different operations carried out (primarily related to the non-linear inversion)

The next few pages show a set of screen captures along with a brief description of the various buttons and options which appear in the different tabs. The actual appearance of these tabs and windows may vary from one platform to another depending on the Java installation. This section can therefore be viewed as a quick and easy guide for `NonLinearKit`. For more detailed information on the file formats and the different equations which intervene, we refer the reader to the following sections.

1.4 Preliminary remark

The non-linear inversion process can take time. Accordingly, when the program is calculating, it is best just to wait and let it finish what it is doing. Furthermore, in the current version, the program is unable to refresh its appearance while calculating. Hence, the only way to have an idea of whether progress is being made is to look at the messages printed in the terminal window (note that these messages will also appear in the output log once the calculations are finished).



update model using inversion results

load model from a file or from internet

load model from CoRoT/ESTAR/HELAS website

generate polytropic model

save model

clear model

find gravitational constant from hydrostatic equilibrium in model

rescale mass and/or radius of model

calculate new mesh points for model

show grid points in model

information concerning model

model formats

endianness of model

native endianness of computer

plots showing structural profiles

NonLinearKit 1.0

Stellar model	Frequencies	Echelle diagram	Structural inversion	Output log
n	ℓ	f_{obs} (in μHz)	Δf_{obs} (in μHz)	f_{theo} (in μHz)
10	0	700.304	0.135	697.342
11	0	759.825	0.357	756.499
12	0	819.858	0.155	817.387
13	0	881.263	0.246	879.396
14	0	943.429	0.342	941.460
15	0	1 004.540	0.296	1 002.539
16	0	1 065.520	0.195	1 063.695
17	0	1 126.970	0.180	1 125.854
18	0	1 189.570	0.181	1 188.864
19	0	1 252.470	0.191	1 252.333
20	0	1 315.010	0.273	1 315.617
21	0	1 376.960	0.407	1 378.855
22	0	1 439.110	1.104	1 442.030
23	0	1 503.030	0.775	1 505.562
24	0	1 564.660	2.939	1 569.293
25	0	1 628.220	1.636	1 633.245
26	0	1 693.510	5.136	1 697.203
27	0	1 759.310	2.427	1 761.191
10	1	726.591	0.149	735.812
11	1	786.019	0.187	787.368
12	1	846.646	0.125	847.002
13	1	908.447	0.166	908.353
14	1	970.170	0.219	969.277
15	1	1 031.280	0.183	1 029.923
16	1	1 091.970	0.139	1 091.027
17	1	1 154.000	0.123	1 153.425
18	1	1 216.760	0.139	1 216.415
19	1	1 279.620	0.143	1 279.673
20	1	1 342.570	0.169	1 342.721
21	1	1 405.440	0.222	1 405.752
22	1	1 467.510	0.325	1 468.939
23	1	1 532.020	0.349	1 532.417
24	1	1 593.690	0.609	1 596.189
25	1	1 656.750	1.257	1 660.029
26	1	1 718.260	2.349	1 723.968
27	1	1 787.950	2.257	1 787.881
8	2	754.833	0.388	751.222
9	2	815.388	0.344	812.053
12	2	938.883	0.542	937.027
13	2	1 000.200	0.544	997.603
14	2	1 061.440	0.289	1 058.735
15	2	1 122.750	0.207	1 120.761
16	2	1 184.740	0.222	1 183.741

Calculate eigenfunctions

Plot kernels

Add row

Remove selected row(s)

Clear

Sort

Load file

Append file

Write file

Number of frequencies: 51

observed frequencies,
associated error bars,
and ℓ values

radial orders and calculated
frequencies (these are
obtained from the model,
not the observations)

calculate pulsations

plot kernels of modes
selected in table

add blank row

remove selected row(s)

clear table

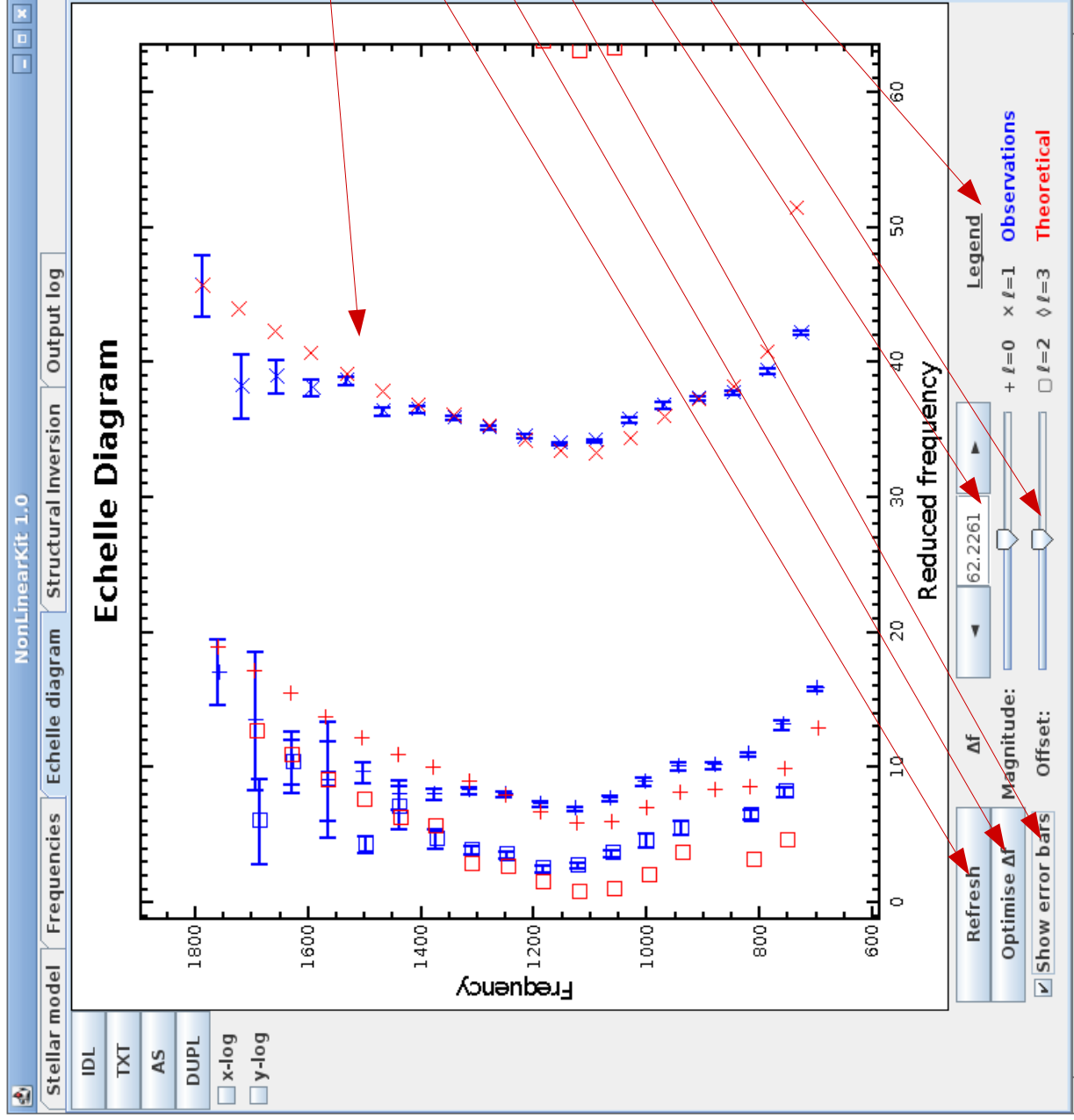
sort table

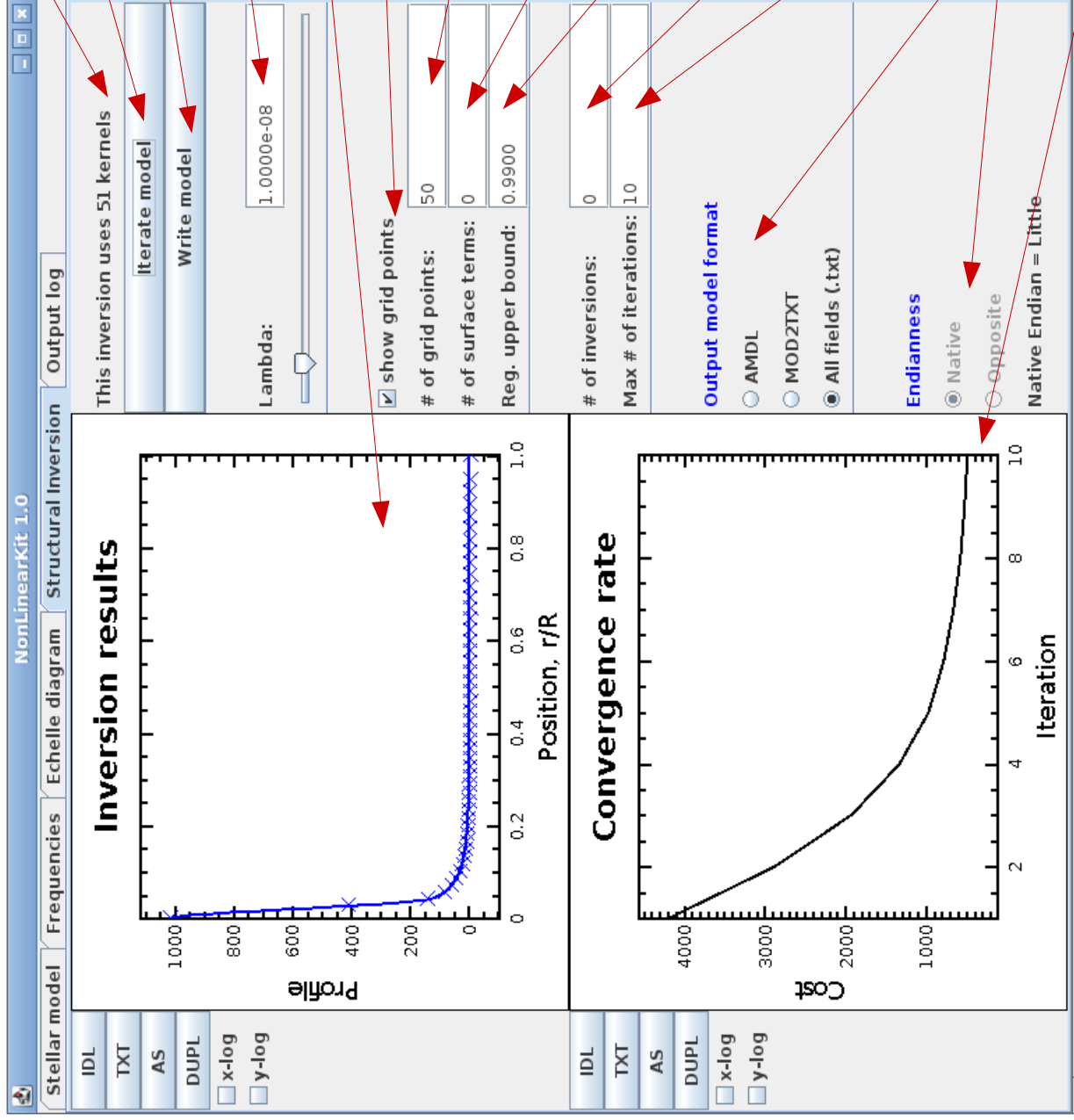
load file with observed data
(erase previous data)

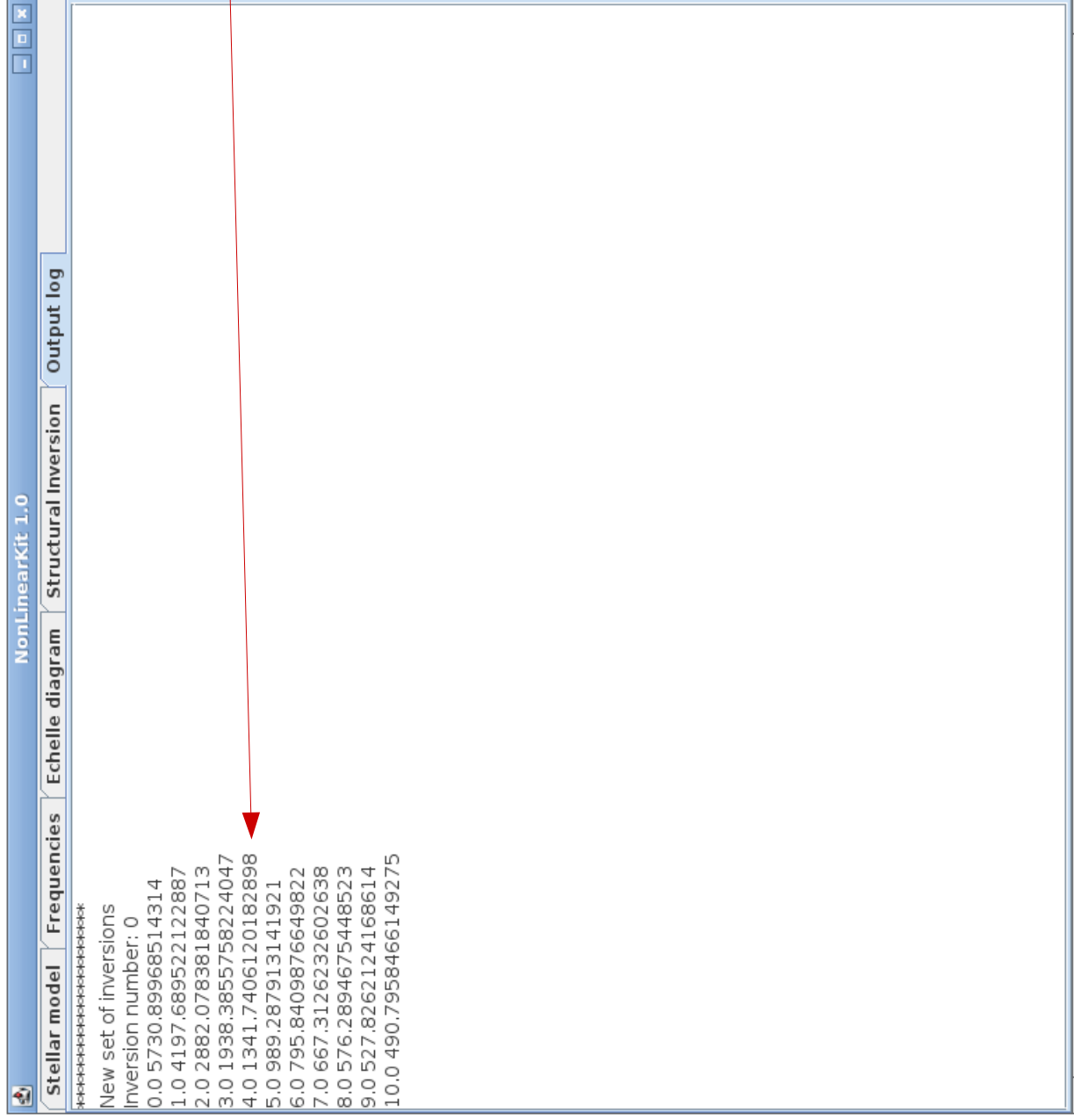
load file with observed data
(do not erase previous data)

write file with observed data

number of frequencies







log of the different operation or results from the non-linear inversion; similar outputs are also written in the command window from which is launched NonLinearKit

2 File formats

`NonLinearKit` deals primarily with two types of files: files with observed frequencies, and files with stellar models. These are described in the following sections.

2.1 File with observed frequencies

The file with the observed frequencies obeys the following rules:

- On a given line, anything following a “#” is treated like a comment and ignored.
- Lines should contain at least three entries, or they will be ignored and will provoke a warning message. The three entries should be:
 - an integer entry with the harmonic degree (or order) ℓ . Lines with a negative value for ℓ are ignored and cause a warning message.
 - the observed frequency, ν_{obs} (in μHz)
 - the error bar on the observed frequency, $\sigma\nu_{\text{obs}}$ (in μHz)
 - anything after the third entry is ignored
- for the floating point entries, non-numerical entries can be used instead, such as “null”, to indicate an unknown value (this will appear as a “NaN”, *i.e.* “Not a Number”, entry in the table)

2.2 Stellar models

`NonLinearKit` accepts the following file formats for stellar models:

1. **AMDL**: FORTRAN binary files generated by ASTEC
2. **FAMDL**: text (or ascii) version of the AMDL files generated by ASTEC
3. **FGONG**: exchange format under the GONG model comparison scheme, and official output format from ASTEC
4. **CESAM**: .osc text files generated by CESAM
5. **MOD (binary)**: FORTRAN binary file from CLES
6. **MOD (ascii)**: text file from CLES
7. **CLES**: another text file format from CLES
8. **LOSC**: a text file from LOSC but in which the model is embedded

`NonLinearKit` is also able to write files in the AMDL, FAMDL, MOD (binary), and MOD (ascii) formats in the **Stellar model** tab. In the **Structural Inversion** tab, it is possible to save models in an additional ascii format which is specific to `NonLinearKit`. The choice of the format (for reading or saving a model) is given by a menu in the **Stellar model** tab (see previous section). `NonLinearKit` will attempt to distinguish between models generated by CESAM2k and earlier versions of CESAM by searching for

“CESAM2k” in the header. If need be, MODCONV can be used to convert models from one format to another. This tool is available at:

<http://www.astro.up.pt/corot/ntools/modconv/>

A description of the first four file formats can be found at:

http://www.astro.up.pt/corot/ntools/docs/CoRoT_ESTA_Files.pdf

and within the instructions to the ADIPLS pulsation code:

http://www.phys.au.dk/~jcd/adipack.n/notes/adiab_prog.ps.gz

The last four file formats will be briefly described in the following sections. This will then be followed by a table which summarises the different structural variables available in the various formats, some useful formulas which relate these variables, and finally a brief description of the format of the file with a list of models for the HR diagram.

2.2.1 MOD (binary/ascii)

These contain a 5 line header (made up of 80 characters in the binary version), followed by a line which contains either 3 or 4 parameters. These are:

- the number of grid points
- the stellar radius (in cm)
- the stellar mass (in g)
- if present, the gravitational constant (in cgs)

This is then followed by a section with 6 or 7 columns (in the ascii version) or a list of entries with 5 or 6 values (in the FORTRAN binary version) which contain the following quantities:

- index of grid point (only in the ascii version)
- the radial coordinate (in cm)
- the cumulative mass (in g)
- the pressure (in cgs)
- the density (in g/cm³)
- the Γ_1 profile
- if present, an unidentified quantity (NonLinearKit does not make use of this column)

2.2.2 CLES format

These contain a header which ends with the expression “%%beginoscddata”. This is followed by a line which contains the following parameters:

- the stellar radius (in cm)
- the stellar mass (in g)
- the gravitational constant (in cgs)

This is followed by a line with the number of domains (in case there are double points), n_{domains} , as well as a section made up of n_{domains} lines which gives information on these domains. The following line contains the number of grid points. This is then followed by a section with 6 columns which contain the following quantities:

- the radial coordinate (in cm)
- the quantity m/r^3 (in g/cm³)
- the pressure (in cgs)
- the density (in g/cm³)
- the Γ_1 profile
- the quantity $-\frac{A}{r^2}$ (see Table 1)

2.2.3 LOSC format

These contain a 5 line header which is skipped. This is followed by a line with the following parameters:

- the number of grid points
- the stellar radius (in cm)
- the stellar mass (in g)
- the gravitational constant (in g)

The next three lines contain a header with column names and a list of various quantities related to the pulsation mode contained with the file. This is followed by a section with many columns. The first six columns are:

- the radial coordinate normalised by the radius
- the quantity $\eta = \frac{R^3 m}{M r^3}$
- the quantity $\frac{R}{GM} \frac{P}{\rho}$
- a normalised density, $\frac{4\pi R^3}{M} \rho$
- the Γ_1 profile
- the quantity $-\frac{A}{r^2}$ (see Table 1)

2.2.4 Output text format from the Structural Inversion tab

This format, which is specific to `NonLinearKit`, is a more extensive output format for inverted models used for testing and plotting purposes. It cannot, for instance, be read by `NonLinearKit`. Files can only be written to this format. It is described by the following formatting rules:

- the first part is an seven line header with the following information:
 - filename (line 1)
 - number of grid points (line 2)
 - radius in cm (line 3)
 - mass in g (line 4)
 - reference pressure in dyn.cm^{-2} (line 5)
 - reference density in g.cm^{-3} (line 6)
 - reference cyclic frequency in μHz (line 7)
- the next line is a header with column names
- this is followed by a table with the following dimensionless quantities (reference values may be deduced from the header):
 - radial coordinate
 - density
 - radial derivative of the density
 - cumulative mass
 - pressure
 - gravity
 - radial derivative of the gravity
 - Γ_1
 - square of the sound velocity
 - square of the Brunt-Väisälä

The comments and headers in the file help to make it self-explanatory.

2.2.5 Comparison of different model formats

Table 1 gives the different variables from the different models and allows an easy comparison between them.

Table 1: Table which lists variables from different model formats. The first column gives the variables which are stored in **NonLinearKit**.

Variable	(F)AMDL	FGONG	CESAM	MOD	LOSC	CLES
$x = \frac{r}{R}$	$\frac{r}{R}$	r	r	r	$\frac{r}{R}$	r
Γ_1	Γ_1	Γ_1	Γ_1	Γ_1	Γ_1	Γ_1
$\frac{R^3}{M}\rho$	—	ρ	ρ	ρ	$\frac{4\pi R^3}{M}\rho$	ρ
$\frac{R^4}{M}\frac{d\rho}{dr}$	—	—	—	—	—	—
$\frac{R^4}{GM^2}P$	—	P	P	P	$\frac{R}{GM}\frac{P}{\rho}$	P
$\frac{R}{GM}c^2$	—	—	—	—	—	—
$\frac{R^3}{GM}N^2$	—	—	—	—	—	—
$\frac{m}{M}$	—	$\ln\left(\frac{m}{M}\right)$	$\ln\left(\frac{m}{M}\right)$	m	—	—
$\frac{R^2}{GM}g$	—	—	—	—	—	—
$\frac{R^2}{GM}\frac{dg}{dr}$	—	—	—	—	—	—
$\frac{R^2}{r^2}\left(1 - \frac{\rho}{\langle\rho\rangle}\right)$	—	—	—	—	—	—
$U = \frac{4\pi\rho r^3}{3}$	U	—	—	—	—	—
$A = \frac{1}{\Gamma_1}\frac{d\ln P}{d\ln r} - \frac{m}{d\ln r} = \frac{rN^2}{g}$	A	A	A	—	$-\frac{A}{r^2}$	$-\frac{A}{r^2}$
$V_g = -\frac{1}{\Gamma_1}\frac{d\ln P}{d\ln r} = \frac{Gm\rho}{\Gamma_1 pr}$	V_g	—	—	—	—	—
$\eta = \frac{R^3}{M}\frac{m}{r^3} = \frac{R^3}{GM}\frac{g}{r}$	η	—	—	—	η	$\frac{m}{r^3}$

2.2.6 Useful formulas

The following formulas can be useful for finding one structural variable from other variables:

$$c^2 = \frac{\Gamma_1 P}{\rho} \quad (1)$$

$$g = \frac{Gm}{r^2} \quad (2)$$

$$\frac{dg}{dr} = 4\pi G\rho - \frac{2Gm}{r^3} \quad (3)$$

$$\frac{R^3}{M}\rho = \frac{\eta U}{4\pi} \quad (4)$$

$$\frac{R^4}{GM}P = \frac{x^2\eta^2 U}{4\pi V_g \Gamma_1} \quad (5)$$

$$A = -r \left(\frac{g}{c^2} + \frac{1}{\rho} \frac{d\rho}{dr} \right) \quad (6)$$

$$\frac{R^3}{GM}N^2 = \frac{R^3}{GM}\frac{g}{r}A = \eta A \quad (7)$$

$$\langle\rho\rangle = \frac{m}{\frac{4}{3}\pi r^3} \quad (8)$$

2.3 GZIP compression

NonLinearKit can read “gzipped” text files. In order to determine whether or not a file is gzipped, **NonLinearKit** looks at the filename to see if it ends with “.gz”. The same rule also applies when saving a model from the HELAS website <http://www.astro.up.pt/helas/>

onto the hard disk – choosing a filename which ends with “.gz” causes `NonLinearKit` to save a compressed file, whereas any other ending produces an uncompressed file.

3 Treatment of pulsation calculations

`NonLinearKit` calculates pulsation modes in much the same way as `InversionKit`. The main difference is that the current version of `NonLinearKit` does not handle models with discontinuities, as opposed to the latest version of `InversionKit`.

3.1 Pulsation calculations

In order to calculate theoretical modes, the pulsation equations are discretised into matrix form, using the finite-difference scheme described in Reese (2013, A&A 555, A148), and solved using `jlpack` and `jarpack`, a java translation of `lapack` and `arpack`. `NonLinearKit` uses the observed frequencies as target values for the pulsation calculations, thereby ensuring that the nearest theoretical frequencies are obtained.

There are three options for the set of equations. The first option is based on the system of equations used in ADIPLS (Christensen-Dalsgaard, 2008, ApSS 316, 113). The second option uses the Lagrangian pressure perturbation. The third one makes use of the Eulerian pressure and density perturbations. In all cases, a reduced set of equations is used for radial modes. The associated sets of equations are described in the following sections.

3.2 JCD’s equations

3.2.1 Non-radial modes

System of equations The following system of equations is derived from the one used in the ADIPLS code. The variables have been scaled by the appropriate power of r so as to have an $\mathcal{O}(1)$ behaviour at the centre. Furthermore, the differential system has been expressed in terms of r^2 to ensure that the solutions are a power series in r^2 rather than r :

$$2r^2 \frac{dy_1}{dr^2} = (V_g - \ell - 1)y_1 + \left(1 - \frac{\omega^2 V_g}{\ell(\ell + 1)\eta}\right) y_2 - V_g y_3 \quad (9)$$

$$2r^2 \omega^2 \frac{dy_2}{dr^2} = \ell(\ell + 1) (\omega^2 - \eta A) y_1 + \omega^2 (A - \ell) y_2 + \ell(\ell + 1) \eta A y_3 \quad (10)$$

$$2r^2 \frac{dy_3}{dr^2} = (2 - \ell) y_3 + y_4 \quad (11)$$

$$2r^2 \frac{dy_4}{dr^2} = -AU y_1 - \frac{\omega^2 U V_g}{\ell(\ell + 1)\eta} y_2 + [\ell(\ell + 1) + U(A - 2)] y_3 + (3 - \ell - 2U) y_4 \quad (12)$$

where

$$y_1 = \frac{\xi_r}{r^{\ell-1}} \quad (13)$$

$$y_2 = \ell(\ell+1) \frac{\xi_h}{r^{\ell-1}} = \frac{\ell(\ell+1)}{\omega^2 r^\ell} \left(\frac{P}{\rho_0} + \psi \right) \quad (14)$$

$$y_3 = -\frac{\psi}{r^{\ell-1} g_0} \quad (15)$$

$$y_4 = -r^{3-\ell} \frac{d}{dr} \left(\frac{\psi}{r g_0} \right) = r^{3-\ell} \frac{dr^{\ell-2} y_3}{dr} \quad (16)$$

$$\xi_r = \text{the Lagrangian vertical displacement} \quad (17)$$

$$\xi_h = \text{the Lagrangian horizontal displacement} \quad (18)$$

$$\psi = \text{the Eulerian perturbation to the gravitational potential} \quad (19)$$

$$V_g = -\frac{1}{\Gamma_1} \frac{d \ln P_0}{d \ln r} = \frac{m_0 \rho_0}{\Gamma_1 P_0 r} = \frac{m_0}{r c_0^2} \quad (20)$$

$$A = \frac{1}{\Gamma_1} \frac{d \ln P_0}{d \ln r} - \frac{d \ln \rho_0}{d \ln r} \quad (21)$$

$$U = \frac{4\pi \rho_0 r^3}{m_0} \quad (22)$$

$$\eta = \frac{m_0}{r^3}, \quad (23)$$

the above variables being non-dimensionalised with respect to R , M/R^2 , and GM^2/R^4 as units of length, density and pressure, respectively. *NOTE*: in JCD's notes for ADIPLS, the convention $\Delta\psi = -4\pi G\rho$ is used, thereby explaining the sign differences with the following sets of equations.

Boundary conditions In the centre, the boundary conditions are:

$$y_2 = (\ell+1)y_1 \quad (24)$$

$$y_4 = (\ell-2)y_3 \quad (25)$$

At the surface, the boundary conditions are:

$$\omega^2 y_2 = \ell(\ell+1)\eta(y_1 - y_3) \quad (26)$$

$$y_4 = -(\ell+U)y_3 + U y_1 \quad (27)$$

3.2.2 Radial modes

System of equations In the radial case, the above equations are not appropriate, given the $1/[\ell(\ell+1)]$ factor which appears in some of the terms. Furthermore, one can eliminate the perturbations to the gravitational potential by integrating Poisson's equation (see, e.g., Aerts et al. 2010, *Asteroseismology*, p. 195):

$$\frac{d\psi}{dr} = -\Lambda \rho_0 \xi_r \quad (28)$$

Hence, the following reduced system is used instead:

$$2r^2 \frac{d\tilde{y}_1}{dr^2} = (V_g - 3) \tilde{y}_1 - \frac{\omega^2}{c_0^2} \tilde{y}_2, \quad (29)$$

$$2\omega^2 \frac{d\tilde{y}_2}{dr^2} = [\omega^2 - \eta(A - U)] \tilde{y}_1 + \omega^2 \frac{A}{r^2} \tilde{y}_2 \quad (30)$$

where

$$\tilde{y}_1 = \frac{\xi_r}{r} \quad (31)$$

$$\tilde{y}_2 = \frac{P}{\omega^2 \rho_0} \quad (32)$$

Boundary conditions In the centre, the boundary condition is:

$$0 = 3\tilde{y}_1 + \frac{\omega^2}{c_0^2} \tilde{y}_2 \quad (33)$$

At the surface, the boundary condition is:

$$0 = r^2 \eta \tilde{y}_1 - \omega^2 \tilde{y}_2 \quad (34)$$

3.3 Lagrangian pressure perturbations

The main advantage of using the Lagrangian pressure perturbation is that the Brunt-Väisälä drops out from the equations. Nonetheless, this set of equations is not a good choice when dealing with red giants given that the Lagrangian pressure perturbation is dominated by the $\xi \cdot \vec{\nabla} P_0$ term.

3.3.1 Non-radial modes

System of equations As was done above, the variables have been scaled by the appropriate power of r so as to have an $\mathcal{O}(1)$ behaviour at the centre, and the differential system has been expressed in terms of r^2 (the entity $\left(\frac{d\tilde{\psi}}{dr}\right)$ is treated as a variable in its own right, so is not derived with respect to r^2):

$$0 = \frac{r^2}{\Gamma_1 P_0} \delta \tilde{P} + 2r^2 \frac{d\tilde{\xi}_r}{dr^2} + (\ell + 1)\tilde{\xi}_r - \ell(\ell + 1)\tilde{\xi}_h \quad (35)$$

$$\begin{aligned} \omega^2 \tilde{\xi}_r &= \frac{2r^2 P_0}{\rho_0} \frac{d}{dr^2} \left(\frac{\delta \tilde{P}}{P_0} \right) + \left[\frac{\ell P_0}{\rho_0} + \left(\frac{1}{\Gamma_1} - 1 \right) g_0 r \right] \frac{\delta \tilde{P}}{P_0} \\ &\quad + \left(\frac{d\tilde{\psi}}{dr} \right) + 2r g_0 \frac{d\tilde{\xi}_r}{dr^2} + \left(\frac{(\ell - 1)g_0}{r} + \frac{dg_0}{dr} \right) \tilde{\xi}_r \end{aligned} \quad (36)$$

$$\omega^2 \tilde{\xi}_h = \frac{P_0}{\rho_0} \frac{\delta \tilde{P}}{P_0} + \tilde{\psi} + \frac{g_0}{r} \tilde{\xi}_r \quad (37)$$

$$0 = 2r^2 \frac{d}{dr^2} \left(\tilde{\psi} \right) + \ell \tilde{\psi} - \left(\frac{d\tilde{\psi}}{dr} \right) \quad (38)$$

$$0 = 2r^2 \frac{d}{dr^2} \left(\frac{d\tilde{\psi}}{dr} \right) + (\ell + 1) \left(\frac{d\tilde{\psi}}{dr} \right) - \ell(\ell + 1)\tilde{\psi} - \Lambda \left(\frac{r^2 \rho_0}{\Gamma_1} \frac{\delta \tilde{P}}{P_0} - r \frac{d\rho_0}{dr} \tilde{\xi}_r \right) \quad (39)$$

where

$$\begin{aligned}\delta\tilde{P} &= r^{-\ell}\delta P \\ \tilde{\xi}_r &= r^{1-\ell}\xi_r \\ \tilde{\xi}_h &= r^{1-\ell}\xi_h \\ \tilde{\psi} &= r^{-\ell}\psi \\ \left(\frac{d\tilde{\psi}}{dr}\right) &= r^{1-\ell}\frac{d\psi}{dr}\end{aligned}$$

The above equations are solved in non-dimensional form, where R , M/R^3 and GM^2/R^4 are units of length, density and pressure, respectively. This leads to $\Lambda = 4\pi$ (as opposed to $4\pi G$ in dimensional form). Once the above system is solved, the original non-scaled variables are obtained by multiplying the scaled variables by the appropriate power of r .

Boundary conditions The boundary conditions are as follows. In the centre, the continuity equation and the equation relating $\tilde{\psi}$ and $\left(\frac{d\tilde{\psi}}{dr}\right)$ become:

$$0 = \tilde{\xi}_r - \ell\tilde{\xi}_h \quad (40)$$

$$0 = \ell\tilde{\psi} - \left(\frac{d\tilde{\psi}}{dr}\right) \quad (41)$$

At the surface, the boundary conditions are:

$$0 = \frac{\delta\tilde{P}}{P_0} \quad (42)$$

$$0 = \left(\frac{d\tilde{\psi}}{dr}\right) + (\ell + 1)\tilde{\psi} + \Lambda\rho_0\tilde{\xi}_r \quad (43)$$

3.3.2 Radial modes

System of equations In the radial case, the above system can be reduced by eliminating the perturbations to the gravitational potential (see Eq. 28) and the variable ξ_h , which is undefined. The final system is, after some manipulations:

$$0 = \frac{1}{\Gamma_1} \frac{\delta P}{P_0} + 2r^2 \frac{d\hat{\xi}_r}{dr^2} + 3\hat{\xi}_r \quad (44)$$

$$\omega^2 \hat{\xi}_r = \frac{2P_0}{\rho_0} \frac{d}{dr^2} \left(\frac{\delta P}{P_0} \right) - \frac{g_0}{r} \frac{\delta P}{P_0} - 4 \frac{g_0}{r} \hat{\xi}_r \quad (45)$$

where $\xi_r = r\hat{\xi}_r$.

Boundary conditions The boundary conditions are:

$$0 = \frac{1}{\Gamma_1} \frac{\delta P}{P_0} + 3\hat{\xi}_r \quad \text{at} \quad r = 0 \quad (46)$$

$$0 = \frac{\delta P}{P} \quad \text{at} \quad r = R \quad (47)$$

3.4 Eulerian pressure and density perturbations

This is a straightforward, but somewhat naive version of the pulsation equations with the Eulerian pressure and density perturbations, P and ρ . As was the case for JCD's equations, the Brunt-Väisälä appears explicitly. However, unlike in JCD's equations, the variable P appears on its own rather than implicitly through the horizontal displacement. Accordingly, this set of equations fails to produce good results in red giants, and is also more expensive since it involves 6 rather than 4 unknowns. It is kept primarily for the purposes of carrying out tests and comparisons.

3.4.1 Non-radial modes

System of equations Once more, the variables have been scaled by the appropriate power of r so as to have an $\mathcal{O}(1)$ behaviour at the centre, and the differential system has been expressed in terms of r^2 :

$$0 = r^2 \tilde{\rho} + 2r^2 \rho_0 \frac{d\tilde{\xi}_r}{dr^2} + \left[(\ell + 1) \rho_0 + r \frac{d\rho_0}{dr} \right] \tilde{\xi}_r - \ell(\ell + 1) \rho_0 \tilde{\xi}_h \quad (48)$$

$$\omega^2 \rho_0 \tilde{\xi}_r = 2r^2 \frac{d\tilde{P}}{dr^2} + \ell \tilde{P} + r g_0 \tilde{\rho} + \rho_0 \left(\frac{d\tilde{\psi}}{dr} \right) \quad (49)$$

$$\omega^2 \rho_0 \tilde{\xi}_h = \tilde{P} + \rho_0 \tilde{\psi} \quad (50)$$

$$0 = \tilde{P} - c_0^2 \tilde{\rho} + \frac{\rho_0 N_0^2 c_0^2}{r g_0} \tilde{\xi}_r \quad (51)$$

$$0 = 2r^2 \frac{d}{dr^2} \left(\tilde{\psi} \right) + \ell \tilde{\psi} - \left(\frac{d\tilde{\psi}}{dr} \right) \quad (52)$$

$$0 = 2r^2 \frac{d}{dr^2} \left(\frac{d\tilde{\psi}}{dr} \right) + (\ell + 1) \left(\frac{d\tilde{\psi}}{dr} \right) - \ell(\ell + 1) \tilde{\psi} - \Lambda r^2 \tilde{\rho} \quad (53)$$

where

$$P = r^\ell \tilde{P} = \text{the Eulerian pressure perturbation}$$

$$\rho = r^\ell \tilde{\rho} = \text{the Eulerian pressure perturbation}$$

The above equations are solved in non-dimensional form, where R , M/R^3 and GM^2/R^4 are units of length, density and pressure, respectively. This leads to $\Lambda = 4\pi$ (as opposed to $4\pi G$ in dimensional form). Once the above system is solved, the original non-scaled variables are obtained by multiplying the scaled variables by the appropriate power of r .

Boundary conditions In the centre, the boundary conditions are:

$$0 = \tilde{\xi}_r - \ell \tilde{\xi}_h \quad (54)$$

$$0 = \ell \tilde{\psi} - \left(\frac{d\tilde{\psi}}{dr} \right) \quad (55)$$

At the surface, the boundary conditions are:

$$0 = r \tilde{P} - \rho_0 g_0 \tilde{\xi}_r \quad (56)$$

$$0 = \left(\frac{d\tilde{\psi}}{dr} \right) + (\ell + 1) \tilde{\psi} + \Lambda \rho_0 \tilde{\xi}_r \quad (57)$$

3.4.2 Radial modes

System of equations As previously, the perturbations to the gravitational potential are eliminated analytically, thanks to Eq. (28). After some manipulations, the final system is:

$$0 = \frac{1}{\Gamma_1 P_0} P + 2r^2 \frac{d\hat{\xi}_r}{dr^2} + \left(3 - \frac{rg_0}{c_0^2}\right) \hat{\xi}_r \quad (58)$$

$$\omega^2 \rho_0 \hat{\xi}_r = 2 \frac{dP}{dr^2} + \frac{g_0}{rc_0^2} P - \left(\Lambda \rho_0^2 + \frac{\rho_0 g_0^2}{c_0^2} + g_0 \frac{d\rho_0}{dr}\right) \hat{\xi}_r \quad (59)$$

where $\xi_r = r\hat{\xi}_r$.

Boundary conditions The boundary conditions are:

$$0 = \frac{1}{\Gamma_1 P_0} P + 3\hat{\xi}_r \quad \text{at} \quad r = 0 \quad (60)$$

$$0 = P - r\rho_0 g_0 \hat{\xi}_r \quad \text{at} \quad r = R \quad (61)$$

3.5 Reconstructing missing variables

Evidently, the above systems of equations do not use the same variables. When constructing kernels, however, it is helpful to have the same set of variables for all of the pulsation modes. A simple solution consists in having a maximal set of variables which include all of the above variables. This, therefore, involves reconstructing missing variables in the different cases.

We used the following formulas to reconstruct variables when working with the Lagrangian pressure perturbations:

$$P = P_0 \frac{\delta P}{P_0} + \rho_0 g_0 \xi_r \quad (62)$$

$$\rho = \frac{\rho_0}{\Gamma_1} \frac{\delta P}{P_0} - \frac{d\rho_0}{dr} \xi_r \quad (63)$$

When working with Eulerian pressure and density perturbations, we applied the following formula:

$$\frac{\delta P}{P_0} = \frac{P - \rho_0 g_0 \xi_r}{P_0} \quad (64)$$

When working with JCD's formulas, the following equations were applied:

$$\psi = -r^{\ell-1} g_0 y_3 \quad (65)$$

$$\frac{d\psi}{dr} = -r^{\ell-1} \left[\frac{g_0 y_4}{r} + \left(\frac{g_0}{r} + \frac{dg_0}{dr} \right) y_3 \right] \quad (66)$$

$$P = \rho_0 (\omega^2 r \xi_h - \psi) \quad (67)$$

$$\rho = \frac{P}{c_0^2} + \frac{A\rho_0}{r} \xi_r \quad (68)$$

$$\frac{\delta P}{P_0} = \frac{P - \rho_0 g_0 \xi_r}{P_0} \quad (69)$$

When dealing with radial modes, the above formulas were used as needed. The derivative of the gravitational potential perturbations, ψ was obtained thanks to Eq. (28), and ψ was obtained via the following formula which takes into account the boundary condition on ψ and $\frac{d\psi}{dr}$:

$$\psi = \int_R^r \frac{d\psi}{dr} dr \quad (70)$$

4 Inversions

4.1 General description

NonLinearKit applies an RLS method in order to carry out a structural inversion of the star. This approach relies on an iterative Levenberg-Marquardt algorithm to minimise the following non-linear cost function:

$$J = \sum_{l=1}^L \frac{(\nu_l^{\text{obs}} - \nu_l^{\text{ref}} - \sum_{n=0}^{N-1} \frac{a_n \psi_n(\nu_l^{\text{obs}})}{E_l})^2}{\sigma_l^2} + \Lambda \int_0^{r_{\text{cut}}} \left(\frac{d^2 \ln(\rho)}{dr^2} \right)^2 dr \quad (71)$$

where

- L = the number of observed frequencies
- l = a generic index to represent individual modes
- ν_l^{obs} = an observed frequency
- ν_l^{ref} = a theoretical frequency
- ψ_n = a Legendre polynomial to represent a surface effect
- E_l = the mode inertia (an expression is given below)
- σ_l = the error bar on an observed frequency
- Λ = a regularisation parameter
- r_{cut} = a cutoff radius
- ρ = the density profile

At this point, a few comments should be made concerning the cost function. First of all, the cost function is non-linear because it depends on theoretical frequencies which depend in a non-linear way on the density profile, which is what **NonLinearKit** is trying to invert. Secondly, the regularisation term is based on the second derivative of $\ln(\rho)$ rather than ρ . This is because ρ changes by many orders of magnitude within a star. Such a regularisation term tends to favour profiles which behave in an exponential way. It was, however, found that $\ln(\rho)$ tends to decrease sharply near the surface, which could be quite penalising. To avoid this problem, a cutoff radius, r_{cut} , which can be adjusted using a text field in the **Structural Inversion** tab, was introduced. Thirdly, the surface term has been expressed in terms of the observed frequencies rather than the theoretical frequencies. This leads to simpler expressions when applying the Levenberg-Marquardt algorithm.

4.2 Comparison with other methods

NonLinearKit applies a similar approach to the one used in Antia (1996, A&A 307, 609-623). Indeed, both approaches rely on an RLS approach to iteratively correct the reference

model, and both use the variational principle, and more specifically the (ρ, Γ_1) structural kernels, to relate frequency differences to structural differences, and both.

However, there are some notable differences. In **NonLinearKit**, the same cost function is used throughout the minimisation. In contrast, the cost function in Antia (1996) includes a regularisation term based on the correction to the structural profiles leading, in effect, to a cost function which changes at each iteration. Furthermore, the present approach uses a Levenberg-Marquardt algorithm in the iterations whereas Antia (1996) applies a linear RLS at each iteration. This leads to different convergence criteria. Finally, the present approach only adjusts the ρ profile and does not modify the Γ_1 profile. The reason for this is because, **NonLinearKit** is designed to work with stars other than the sun, and hence with a much smaller set of pulsation frequencies. The method in Antia (1996) corrects both the ρ and Γ_1 profiles and was applied to the solar frequencies.

Another non-linear approach to inverting the structure of a star is the differential response technique, developed by Roxburgh and Vorontsov (see, *e.g.*, Vorontsov, 1998, IAUS 181, 135, Roxburgh & Vorontsov, 2002, Proc. 1st Eddington Workshop, 349). This method relies on internal phases of the eigenfunctions and is accordingly quite different than the approach taken here. The main similarity is that the Γ_1 profile is kept fixed and not adjusted, thereby leading to the adjustment of only one structural profile (typically the density).

4.3 Levenberg-Marquardt algorithm

In the Levenberg-Marquardt algorithm, one calculates both the gradient of the cost function J as well as an approximation of its Jacobian matrix, in which second order derivatives are neglected. In order to obtain these, it is necessary to express the density profile in a discretised form which can be adjusted by the inversion. We choose the following form:

$$\rho(r) = \exp \left[\sum_{m=1}^M c_m \phi_m(r) \right] \rho_{\text{ref}}(r) \quad (72)$$

where ρ_{ref} is the density profile from the previous iteration. This form prevent the density profile from taking on negative values, and leads to simpler expressions in the gradient and approximate Jacobian of J .

The partial derivatives of J with respect to the c_m and the a_n are:

$$\frac{\partial J}{\partial c_m} = 2 \sum_{l=1}^L \frac{\partial \nu_l^{\text{ref}}}{\partial c_m} \left(\frac{\nu_l^{\text{ref}} + \sum_{n=0}^{N-1} \frac{a_n \psi_n(\nu_l^{\text{obs}})}{E_l}}{\sigma_l^2} \right) + 2\Lambda \int_0^{r_{\text{cut}}} \frac{d^2 \ln \rho}{dr^2} \frac{\partial}{\partial c_m} \left(\frac{d^2 \ln \rho}{dr^2} \right) dr \quad (73)$$

$$\frac{\partial J}{\partial a_n} = 2 \sum_{l=1}^L \frac{\psi(\nu_l^{\text{obs}})}{E_l} \left(\frac{\nu_l^{\text{ref}} + \sum_{n=0}^{N-1} \frac{a_n \psi_n(\nu_l^{\text{obs}})}{E_l}}{\sigma_l^2} \right) \quad (74)$$

where

$$\begin{aligned} \frac{1}{\nu_l^{\text{ref}}} \frac{\partial \nu_l^{\text{ref}}}{\partial c_m} &= \int_0^R K_{\rho, \Gamma_1}^l(r) \frac{1}{\rho} \frac{\partial \rho}{\partial c_m} dr \\ K_{\rho, \Gamma_1}^l &= \text{the } (\rho, \Gamma_1) \text{ kernel associated with mode } l \text{ (expressions are given below)} \\ \frac{1}{\rho} \frac{\partial \rho}{\partial c_m} &= \frac{\partial \ln \rho}{\partial c_m} = \phi_m \\ \frac{\partial}{\partial c_m} \left(\frac{d^2 \ln \rho}{dr^2} \right) &= \frac{d^2}{dr^2} \left(\frac{\partial \ln \rho}{\partial c_m} \right) = \frac{d^2 \phi_m}{dr^2} \end{aligned}$$

Finally, the gradient is defined as:

$$\vec{\nabla} J = \begin{bmatrix} \frac{\partial J}{\partial c_1} \\ \vdots \\ \frac{\partial J}{\partial c_M} \\ \frac{\partial J}{\partial a_0} \\ \vdots \\ \frac{\partial J}{\partial a_{N-1}} \end{bmatrix} \quad (75)$$

The following terms intervene in the approximate Jacobian, \mathcal{J} :

$$\frac{\partial^2 J}{\partial c_m \partial c'_m} \simeq 2 \sum_{l=1}^L \frac{1}{\sigma_l^2} \frac{\partial \nu_l^{\text{ref}}}{\partial c_m} \frac{\partial \nu_l^{\text{ref}}}{\partial c'_m} + 2\Lambda \int_0^{r_{\text{cut}}} \frac{\partial}{\partial c_m} \left(\frac{d^2 \ln \rho}{dr^2} \right) \frac{\partial}{\partial c'_m} \left(\frac{d^2 \ln \rho}{dr^2} \right) dr \quad (76)$$

$$\frac{\partial^2 J}{\partial c_m \partial a_n} \simeq 2 \sum_{l=1}^L \frac{1}{\sigma_l^2} \frac{\partial \nu_l^{\text{ref}}}{\partial c_m} \frac{\psi_n(\nu_l^{\text{obs}})}{E_l} \quad (77)$$

$$\frac{\partial^2 J}{\partial a_n \partial a'_n} \simeq 2 \sum_{l=1}^L \frac{1}{\sigma_l^2} \frac{\psi_n(\nu_l^{\text{obs}})}{E_l} \frac{\psi'_n(\nu_l^{\text{obs}})}{E_l} dr \quad (78)$$

The Levenberg-Marquardt algorithm then consists in iteratively applying corrections in the form of $-\tilde{\mathcal{J}}^{-1} \vec{\nabla} J$ to the density profile. $\tilde{\mathcal{J}}$ is a modified version of \mathcal{J} , the approximate Jacobian, in which the diagonal elements have been multiplied by a damping parameter, λ . This parameter is large at the beginning, meaning the algorithm behaves as a gradient-descent algorithm, and progressively becomes smaller with successful iterations, thereby leading to a behaviour closer to the Gauss-Newton algorithm.

4.4 Various useful formulas

This section recalls various formulas which intervene in the calculations. In what follows, we will drop the l index and assume we're dealing with a particular mode.

4.4.1 Mode inertia

The mode inertia is given by the following expression

$$E = \frac{\int_0^R [\xi^2 + \ell(\ell+1)\eta^2] \rho r^2 dr}{M [\xi(R_{\text{phot.}})^2 + \ell(\ell+1)\eta(R_{\text{phot.}})^2]} \quad (79)$$

where

- ξ = the Lagrangian vertical displacement
- η = the Lagrangian horizontal displacement
- ℓ = the harmonic degree
- $R_{\text{phot.}}$ = the photospheric radius

4.4.2 c^2, ρ kernels

The expressions for structural kernels are obtained by perturbing the variational expression for the frequency, then using the the variational principle to remove terms related to the variation of the eigenfunctions. The following expressions are from Gough & Thompson (1991, in *Solar interior and atmosphere*, p. 519-561).

The relative frequency variations are given by

$$\frac{\delta\nu}{\nu} = \int_{r=0}^R \left[K_{c^2, \rho}(r) \frac{\delta c_0^2(r)}{c_0^2(r)} + K_{\rho, c^2}(r) \frac{\delta \rho_0(r)}{\rho_0(r)} \right] dr \quad (80)$$

in which the kernels take on the following expressions:

$$K_{c^2, \rho} = \frac{\rho_0 c_0^2 \chi^2 r^2}{2I\omega^2} \quad (81)$$

$$K_{\rho, c^2} = \frac{\rho_0 r^2}{2I\omega^2} \left\{ c_0^2 \chi^2 - \omega^2 (\xi^2 + \ell(\ell+1)\eta^2) - 2g_0 \xi \chi - 4\pi G \int_{s=r}^R \left(2\rho_0 \xi \chi + \frac{d\rho_0}{ds} \xi^2 \right) ds \right. \\ \left. + 2g_0 \xi \frac{d\xi}{dr} + 4\pi G \rho_0 \xi^2 + 2 \left(\xi \frac{d\psi}{dr} + \frac{\ell(\ell+1)\eta\psi}{r} \right) \right\} \quad (82)$$

where

$$\begin{aligned} \omega &= 2\pi\nu \\ \chi &= \frac{\vec{\nabla} \cdot \vec{\xi}}{Y_m^\ell} = \frac{d\xi}{dr} + \frac{2\xi}{r} - \frac{\ell(\ell+1)\eta}{r} \\ \rho &= -\frac{d\rho_0}{dr} \xi - \rho_0 \chi \\ \psi &= -\frac{4\pi G}{2\ell+1} \left[\int_{s=0}^r \rho(s) \frac{s^{\ell+2}}{r^{\ell+1}} ds + \int_{s=r}^R \rho(s) \frac{r^\ell}{s^{\ell-1}} ds \right] \\ \frac{d\psi}{dr} &= -\frac{4\pi G}{2\ell+1} \left[-(\ell+1) \int_{s=0}^r \rho(s) \frac{s^{\ell+2}}{r^{\ell+2}} ds + \ell \int_{s=r}^R \rho(s) \frac{r^{\ell-1}}{s^{\ell-1}} ds \right] \\ m_0 &= 4\pi \int_{s=0}^r \rho_0(s) s^2 ds \\ g_0 &= \frac{Gm_0}{r^2} \end{aligned}$$

and quantities with the subscript “0” refer to the equilibrium model.

4.4.3 Γ_1, ρ kernels

The relative frequency variations are given by

$$\frac{\delta\nu}{\nu} = \int_0^R \left[K_{\Gamma_1, \rho}(r) \frac{\delta \Gamma_1(r)}{\Gamma_1(r)} + K_{\rho, \Gamma_1}(r) \frac{\delta \rho_0(r)}{\rho_0(r)} \right] dr \quad (83)$$

Expressions for these kernels can be deduced from the previous set of kernels:

$$K_{\Gamma_1, \rho} = K_{c^2, \rho} = \frac{\rho_0 c_0^2 \chi^2 r^2}{2I\omega^2} \quad (84)$$

$$\begin{aligned} K_{\rho, \Gamma_1} &= K_{\rho, c^2} - K_{c^2, \rho} + \frac{Gm\rho_0}{r^2} \int_{s=0}^r \frac{K_{c^2, \rho}(s)}{p_0(s)} ds + \rho_0 r^2 \int_{s=r}^R \frac{4\pi G \rho_0}{s^2} \left(\int_{t=0}^s \frac{K_{c^2, \rho}(t)}{p_0(t)} dt \right) ds \\ &= K_{\rho, c^2} - K_{c^2, \rho} + \frac{Gm\rho_0}{r^2} \int_{s=0}^r \frac{\Gamma_1 \chi^2 s^2}{2I\omega^2} ds + \rho_0 r^2 \int_{s=r}^R \frac{4\pi G \rho_0}{s^2} \left(\int_{t=0}^s \frac{\Gamma_1 \chi^2 t^2}{2I\omega^2} dt \right) ds \end{aligned} \quad (85)$$

5 Known bugs

Here is a list of known bugs. If you find any other, please let us know by sending us an email (daniel.reese@obspm.fr).

- excessive zooming on plots can produce irregular behaviour
- the pulsation calculations becomes unreliable beyond $\ell = 20$ due to the fact that `NonLinearKit` uses scaled variables in the pulsation equations.

6 Copyright notices

Below is the copyright notice that goes with `NonLinearKit`.

Copyright (c) Daniel Reese, 2016

This file is part of `NonLinearKit`.

`NonLinearKit` 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, either version 3 of the License, or (at your option) any later version.

`NonLinearKit` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with `NonLinearKit`. If not, see <http://www.gnu.org/licenses/>.

6.1 Source code for reading fortran binary files

The source code for reading fortran binary files comes from the following web-pages:
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/NativeBinaryInputStream.html>
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/NativeBinaryOutputStream.html>
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/FortranBinaryInputStream.html>
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/FortranBinaryOutputStream.html>
and are covered by the GNU GPL2 License. They have been corrected and modified so as to meet the needs of `NonLinearKit`.

6.2 Supplementary notices

Some of the code comes from other sources. The corresponding copyright notices are reproduced below:

Notice number 1

@(#)OptionPaneDemo.java 1.9 04/07/26

Copyright (c) 2004 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Notice number 2

Copyright (c) Ian F. Darwin, <http://www.darwinsys.com/>, 1996-2002.
All rights reserved. Software written by Ian F. Darwin and others.
\$Id: LICENSE,v 1.8 2004/02/09 03:33:38 ian Exp \$

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Java, the Duke mascot, and all variants of Sun’s Java "steaming coffee cup" logo are trademarks of Sun Microsystems. Sun’s, and James Gosling’s, pioneering role in inventing and promulgating (and standardizing) the Java language and environment is gratefully acknowledged.

The pioneering role of Dennis Ritchie and Bjarne Stroustrup, of AT&T, for inventing predecessor languages C and C++ is also gratefully acknowledged.