



earth-tides

User Manual version 1.0

Machiel S. Bos

2026

TEROMOVIGO

Contents

1	Introduction	3
1.1	How to cite earth-tides	4
1.2	Main features	5
1.3	License	5
2	Installation	5
2.1	Quick start	5
2.2	Command-line tools	6
3	Conventions	6
3.1	Love numbers of a flattened Earth	6
3.2	The six choices behind the parameters	7
3.3	Units and signs	9
4	How the solver works	10
4.1	Chebyshev collocation in one variable	11
4.2	Galerkin projection in one variable	13
4.3	Jets and the chain rule in one variable	16
4.4	The equations of a deforming Earth	19
4.5	The solver module	22
4.6	The unknowns	22
4.7	The physics at the nodes	24
4.8	Galerkin projection in the angular direction	25
4.9	Assembling the linear system	25
4.10	Boundary and interface conditions	26
4.11	Liquid–solid boundaries and the Longman paradox	28
4.12	Rigid rotations and the reference frame	29
4.13	Degree 1	29
4.14	Solving the system	30
4.15	Why this is not one headache traded for another	30
5	Tutorial	31
5.1	Example 1: the pumped swing	32
5.2	Example 2: the loaded elliptical drum	33
5.3	Example 3: homogeneous spheres	34
5.4	Example 4: a flattened homogeneous Earth	35
5.5	Example 5: the free core nutation of Hough’s Earth	36
5.6	Love numbers of PREM	37
5.7	A station time series	39
5.8	What the choices change	41
6	The programs	43
6.1	earthtides-love	43
6.2	etide	44
7	API reference	45

7.1	earthtides.model	45
7.2	earthtides.geometry	46
7.3	earthtides.solver	46
7.4	earthtides.iers	49
7.5	earthtides.love	49
7.6	earthtides.etide	50
7.7	Recipes	50
8	File formats	51
8.1	Earth model	51
8.2	Wave groups	55
8.3	Love-number table	57
9	Appendix: formulas of the supporting modules	59
9.1	Hydrostatic flattening (Clairaut)	60
9.2	Anelasticity (absorption band)	60
9.3	Reference potential consistent with the density	60
9.4	Fit of the IERS parameters	61
9.5	Rigid rotations: the Tisserand row and the mantle frame	61
9.6	The synthesis of etide	61
	References	62

1 Introduction

The Moon and the Sun deform the Earth. Their tidal potential, up to about half a metre when divided by gravity, raises the solid Earth by up to about 30 cm and moves it horizontally by several centimetres, twice a day. Space-geodetic positions are corrected for this body tide, and with millimetre accuracy as the goal, the correction must be right to a fraction of a millimetre. The response of the Earth to a potential of degree n is expressed by the Love numbers h , l and k (Love 1911): the radial displacement, the horizontal displacement and the additional potential, each as a fraction of the potential divided by gravity.

For a spherical, layered Earth the Love numbers follow from a set of ordinary differential equations in radius. Takeuchi (1950) integrated them numerically for an Earth with realistic density and elasticity; Alterman et al. (1959) wrote them as the system of six first-order equations that is still the standard formulation; Pekeris and Accad (1972) added the dynamics of the liquid core. The real Earth rotates and is flattened by about 1/300, and this couples the degrees and makes the response depend on latitude: a few tenths of a millimetre at the surface, but larger in the diurnal band, where the free core nutation resonates. The classical treatment of these effects is a perturbation theory around a spherical Earth (Wahr 1981; Dehant 1987; Mathews et al. 1995; Dehant et al. 1999); numerical methods that work on the flattened or laterally heterogeneous Earth itself, such as the spectral-element model of Métivier et al. (2006), are more recent.

earth-tides is a Python package for solid Earth tides. It computes the Love numbers of a rotating, flattened, layered, anelastic and self-gravitating Earth, and the displacement of a station that follows from them. Three programs are installed with it:

- `earthtides-love` computes Love numbers for the tidal waves of a catalogue (one per wave group by default, or one for every wave) and writes them to a JSON table;
- `etide` computes the body-tide displacement of a station as a time series, from the tidal potential of Tamura (1987) and a Love-number table: the IERS model or one written by `earthtides-love`. It replaces the solid-tide routine DEHANTTIDEINEL of the IERS Conventions (Petit and Luzum 2010, Sect. 7.1.1) and, like it, includes the permanent tide. It gives the displacement only: no ocean tide loading, gravity, tilt or strain;
- `earthtides-examples` copies the worked examples and this manual into a new directory.

The aim of earth-tides is transparency. In a perturbation theory every effect of the flattening is a separate formula, derived by hand and then programmed. Here the equations of motion and Poisson's equation are written once, in physical space, and the program sets up the linear equations itself. Chapter [How the solver works](#) explains how, with tutorials in one variable for each numerical device, because the point of the package is that the reader can check it. The package is written in Python with NumPy and SciPy only.

The framework. The method rests on one property of the Earth model: within each layer the density and the elastic parameters are *polynomials* in radius, and they jump only at the boundaries between layers (PREM and the other models of the package are given in this form; Chapter [File formats](#)). Inside a layer the fields are then smooth, and the solver represents them by short series of polynomials whose derivatives are known exactly. Two numerical devices do the work, one per direction:

- in *radius*, **Chebyshev collocation**: in every layer each field is a sum of Chebyshev polynomials; their derivatives follow from a recurrence formula, so the equations are evaluated exactly

at a set of nodes. The discontinuities of the model never fall inside a series: they sit at the ends of the layers, where the interface conditions are imposed;

- in *latitude*, **Galerkin projection**: along each level surface the residual of the equations is projected on the kept spherical harmonics by Gauss–Legendre sums. The coupling between degrees, which perturbation theory derives by hand as *coupling coefficients*, comes out of the program as these sums.

In between, *jets* and the *chain rule* carry the derivatives through products and through the map from the flattened level surfaces to the radius. Writing this framework took careful programming, but once written it needs no human intervention: more Chebyshev polynomials (N) or more spherical-harmonic degrees (K) only make the linear system larger, no formula changes, and the result converges exponentially. And because nothing is derived by hand for a particular case, the program can be checked against analytical solutions – homogeneous spheres, an exact elastic spheroid, the flattened homogeneous Earth of Greff-Lefftz et al. (2005) – and against itself, a spherical Earth described with distorted coordinates. When these agree to many digits, the same machinery is right for PREM.

How to read this manual. The chapters are of three kinds, and each says at its start which kind it is:

- *Using the package*: **Installation**, **Conventions** (what the Love numbers mean), the last two sections of the **Tutorial** (Love numbers of PREM, a station time series), **The programs**, the **API reference** and the **File formats**.
- *The method*: **How the solver works**. It opens with three short tutorials, each a toy problem small enough to follow by hand; then come the equations of a deforming Earth and an overview of the solver module (the big picture), and then the solver step by step.
- *Verification*: Examples 1 to 5 of the **Tutorial**, a ladder of problems with known answers, each testing one more part of the method.

A reader who only needs Love numbers or station displacements can read the Installation, the Conventions and the last two sections of the Tutorial; a reader who wants to understand or check the method reads How the solver works and Examples 1 to 5.

Further reading. Agnew (2007) is a readable overview of Earth tides: the tidal potential, the response of the Earth, the ocean loading and the measurements. The book chapter of Bos and Scherneck (2013) derives the radial equations of a spherical Earth, solves them for homogeneous spheres in closed form and explains the numerical integration and the load Love numbers; Example 3 reproduces its tables. The classical papers Takeuchi (1950), Alterman et al. (1959) and Pekeris and Accad (1972) remain very readable, and Métivier et al. (2006) describes the elastic-gravitational equations of an Earth that is not spherically symmetric, close in spirit to this package.

1.1 How to cite earth-tides

If you use earth-tides in your research, please cite:

Bos, M.S. (2026). What the IERS body-tide model assumes: Love numbers of a flattened, rotating, anelastic Earth. *Journal of Geodesy* (submitted).

and the software with its version, for example: Bos, M.S. (2026). earth-tides, version 1.0.0 (software). A DOI (Zenodo) is assigned at the first public release and given on the PyPI page; `earth-`

`tides.__version__` and the `metadata` of every Love-number table record the version used.

1.2 Main features

- Love numbers for tidal forcing and surface loads of any degree n and order m , for a layered Earth model in the PREM file format.
- Hydrostatic flattening of all level surfaces from Clairaut's equation; the interfaces are coordinate surfaces.
- A fluid outer core that slides along its boundaries, with inertial and Coriolis forces in every layer, so that the free core nutation appears in the solution.
- Anelasticity by an absorption-band model (complex shear modulus).
- Love numbers in the parametrization of the IERS Conventions ([Petit and Luzum 2010](#)), with every choice behind it (latitude, direction, surface, normalization, frame, reference gravity) as an option.
- Station displacements from the 1200 waves of the Tamura potential (degrees 2 to 4), verified against the IERS routine DEHANTTIDEINEL [the solid-tide routine of the IERS Conventions software; Petit and Luzum ([2010](#))] to 0.1 mm rms.

1.3 License

earth-tides is distributed under a source-available license. It may be used free of charge for academic, research, and other non-commercial purposes. Commercial use is not permitted under this license and requires a separate agreement with TeroMovigo – Earth Innovation Lda. The complete license terms are provided in the `LICENSE` file.

2 Installation

earth-tides needs Python 3.10 or newer, NumPy and SciPy. Use a virtual environment:

```
python3 -m venv ~/venvs/earthtides
source ~/venvs/earthtides/bin/activate
pip install earth-tides
```

From the source repository (development install):

```
git clone https://github.com/machielbos8/earth-tides.git
cd earth-tides
python3 -m venv .venv
.venv/bin/pip install -e ".[test]"
.venv/bin/python -m pytest -q           # tests, about 4 minutes
```

2.1 Quick start

After `pip install earth-tides`, get the examples and this manual:

```
earthtides-examples
```

This copies five worked examples and the PDF manual into a new directory `earthtides-examples` (give another name as argument if you prefer). Each example runs from its own directory and prints the numbers of its `.out` file:

```
cd earthtides-examples/ex1
python swing.py
```

The body tide at Onsala for one day, hourly, with the Love numbers of the IERS Conventions (columns: MJD, up, north, east in metres):

```
etide 57.395 11.926 --start 2026-01-01 --days 1
```

The Love numbers of PREM for every wave group of the tidal catalogue (about 12 seconds), and the same body tide computed with them:

```
earthtides-love -o love.json
etide 57.395 11.926 --start 2026-01-01 --days 1 --love love.json
```

Chapter **Tutorial** explains the examples in order; Chapter **The programs** describes the options of `etide` and `earthtides-love`.

2.2 Command-line tools

program	what it does
<code>earthtides-love</code>	Love numbers per wave group or per wave, all conventions as options, to JSON
<code>etide</code>	body-tide displacement of a station (up, north, east) as a time series
<code>earthtides-examples</code>	copies the examples and this manual to a new directory

All print their options with `--help`.

3 Conventions

Using the package. This chapter says what the computed numbers mean: the Love numbers of a flattened Earth and the choices hidden in their parametrization. It can be read without the next chapter, which explains how the numbers are computed.

3.1 Love numbers of a flattened Earth

On a sphere. A tidal potential of degree n , $V(r, \theta, \lambda) = V_n(r/a)^n \bar{Y}_n^m(\theta) \cos(m\lambda + \chi)$, deforms a spherical Earth in the same pattern. Inside the Earth the response is

$$u_r = h(r) \frac{V_a}{g}, \quad \mathbf{u}_h = l(r) \frac{\nabla_1 V_a}{g}, \quad \phi_1 = k(r) V_a, \quad (3.1)$$

with V_a the potential at the surface $r = a$, ∇_1 the gradient on the unit sphere and g the surface gravity. The Love numbers are therefore *functions of radius*: $h(r)$, $l(r)$ and $k(r)$ are the radial functions $U_n(r)$, $V_n(r)$ and $P_n(r)$ of the solver (Section **The unknowns**), scaled by the potential. Usually only their values at the surface, $h = h(a)$, $l = l(a)$ and $k = k(a)$, are quoted, and the dependence on r is dropped from the notation; it matters for strain and tilt below the surface, in boreholes and mines, and the Chebyshev coefficients of the solver (`TidalSolver.coef`) give the whole field.

On a flattened Earth. The flattening couples the forcing degree n to the degrees $n \pm 2, n \pm 4, \dots$ (Section **Galerkin projection in the angular direction**), so the response is no longer proportional to the pattern of the potential: the ratio of the radial displacement to V/g depends on latitude as well as on depth, $h = h(r, \varphi)$. At the surface the latitude dependence is described with the parameters of Mathews et al. (1995), adopted in the IERS Conventions:

$$h(\varphi) = h^{(0)} + h^{(2)}P_2(\sin \varphi), \quad l(\varphi) = l^{(0)} + l^{(2)}P_2(\sin \varphi), \quad (3.2)$$

plus $l^{(1)}$ (northward and eastward Love numbers differ), l' (a rotation-like term in the diurnal band) and h' (a uniform radial term for the zonal tides). The latitude terms are small, $h^{(2)} \approx -0.0006$ against $h^{(0)} \approx 0.6$. Here too the numbers are surface values and the dependence on r is left out of the notation.

Because $h(\varphi)$ and $l(\varphi)$ are now functions, their parameters depend on how the function is described: at which latitude, along which direction, on which surface. A displacement field has many parametrizations, and the next section lists the choices that select one.

3.2 The six choices behind the parameters

The parametrization does not fix everything. Six choices are hidden in it, and each changes the small terms by amounts comparable to the terms themselves (Bos 2026, Sect. 5.1, which quantifies each for PREM). Five of them are *descriptions*: they change how one computed displacement field is written in parameters, not the field itself. The sixth is *physics*: it changes the calculation.

choice	options (IERS first)	earthtides-love	largest effect (M_2, O_1)
latitude in the basis functions	geocentric / geodetic	--latitude	$h^{(2)} + 0.0053$
direction of the components	geocentric radius / ellipsoid normal	--direction	$l^{(1)} - 0.0020$
surface	flattened surface / mean sphere	--surface	$l^{(2)} - 0.0011$
normalization of the potential	equatorial radius / mean radius	--norm	all, +0.45 %
frame (diurnal, zonal)	mantle (Tisserand) / best-fit surface rotation	--frame	$l' + 0.00035$
reference gravity (physics)	with / without centrifugal force	--no-centrifugal	$h^{(0)} - 0.0008$

Latitude. The basis functions $P_2(\sin \varphi)$ and those of the $l^{(1)}$ and l' terms can be evaluated at the geocentric latitude, as in the IERS Conventions, or at the geodetic latitude, which is larger by about $f \sin 2\varphi$. Because $h^{(0)}$ is a thousand times larger than $h^{(2)}$, a shift of order f in the argument of the main term moves $h^{(2)}$ by ten times its own value. The program fits the same surface displacement with the basis functions at the chosen latitude. *Option:* `--latitude geocentric` (default, IERS) or `--latitude geodetic`.

Direction. The components can refer to the geocentric radius and its perpendiculars (IERS) or to the ellipsoid normal. The two directions differ by the angle $f \sin 2\varphi$, which mixes a fraction f of the large radial displacement into the horizontal terms and vice versa. The program rotates the components by this angle before the fit, to first order in f : $u'_r = u_r - f \sin 2\varphi u_\theta$, $u'_\theta = u_\theta - f \sin 2\varphi u_r$. *Option:* `--direction radial` (default, IERS: geocentric radius) or `--direction normal` (ellipsoid normal).

Surface. The displacement can be evaluated on the flattened surface, where the stations are (IERS), or on a sphere of mean radius, as in perturbation theories that work from a mean sphere. The two differ by the radial gradient of the displacement times the height of the bulge, $\delta = R(a, \theta) - a = -\frac{2}{3}fa P_2(\cos \theta)$, up to 14 km. The program moves the field to the mean sphere by a first-order Taylor step, $u(a) = u(R) - \delta \partial_r u$, with the derivative of the solution in the top layer (the rigid rotation, whose derivative is its value divided by r , is treated separately). *Option:* `--surface deformed` (default, IERS: the flattened surface) or `--surface mean` (the mean sphere).

Normalization. The tidal potential can be referred to the equatorial radius, as in the IERS Conventions and the tidal catalogues, or to the mean radius a . The displacement is h times the potential divided by gravity. Referring the degree-2 potential, which grows as r^2 , to R_e instead of a gives a factor $(R_e/a)^2$, and dividing by the equatorial gravity GM/R_e^2 instead of GM/a^2 another $(R_e/a)^2$: for degree 2 all numbers scale by $(R_e/a)^4 \approx 1 + 4f/3$, and the large terms $h^{(0)}$ and $l^{(0)}$ change by 0.45 per cent. The program divides by this factor. *The Love number k is not rescaled:* the tables give the solver's k , the ratio of the potentials at the mean radius. Because the potential of the deformation decreases as r^{-3} and the tidal potential grows as r^2 , k referred to the equatorial radius is $k(a/R_e)^5 = k(1 + f/3)^{-5}$; `etide` does not use k . *Option:* `--norm equatorial` (default, IERS) or `--norm mean`.

Frame. For the diurnal tides the whole Earth also tilts rigidly (the forced nutation), and for the zonal tides its rotation rate changes (Section [Rigid rotations and the reference frame](#)). A rigid rotation is not a deformation, and it is removed from the displacement; the choice is which rotation. The IERS Conventions and Mathews et al. (1995) use the mean rotation of the mantle (its Tisserand rotation, `frame='mantle'`), Wahr (1981) the rotation that fits the surface displacement best (`frame='surface'`). The program computes the field with the rotation included and subtracts the chosen rigid rotation afterwards (`_remove_mantle_rotation` or `_remove_rigid_rotation`). Only l' , which has the shape of a rotation, is affected. *Option:* `--frame mantle` (default, IERS) or `--frame surface`.

Reference gravity. The gravity of the undeformed Earth enters the forces on a displaced parcel (Section [The physics at the nodes](#)). It can include the centrifugal force, as it must for a rotating Earth in equilibrium, or leave it out, as some analytical theories do (Greff-Lefftz et al. 2005). This is a change of the problem, not of its description: the program solves it again with a different reference state, and $h^{(0)}$ drops by 8×10^{-4} . *Option:* without a flag the centrifugal force is included (IERS);

--no-centrifugal leaves it out.

What the program does. For every wave `earthtides-love` solves the problem once, with the physics options (Earth model, anelasticity, rotation, flattening, reference gravity) and the resolution. It then evaluates the surface displacement at 600 colatitudes on the flattened surface, removes the rigid rotation in the chosen frame, moves the field to the mean sphere if asked, rotates the components if asked, scales it by the normalization, and fits the parameters with the basis functions at the chosen latitude (`love.parameters`). The five descriptions are thus transformations of one solution, and the same solution can be reduced in all of them:

```
from earthtides.love import Options, solve_wave, parameters
S, r = solve_wave(Options(), m=2, f_cpd=1.9322736)           # M2, one solve
iers = parameters(S, r, Options())                          # h2 = -0.00052
geodetic = parameters(S, r, Options(latitude="geodetic"))    # same field: h2 = +0.0049
```

The frame is set when the solver is constructed (`TidalSolver(..., frame=...)`), but it too acts only on the output. `etide` uses only tables in the IERS conventions (first option in every row) and refuses others; tables in other conventions are for comparison. `etide` takes the geodetic latitude of a station, as given by GNSS, converts it to the geocentric latitude of the IERS basis functions, and returns the displacement along the geocentric radius and its perpendiculars; the up component along the ellipsoid normal differs from it by $f \sin 2\varphi$ times the horizontal displacement, a few tenths of a millimetre at most.

3.3 Units and signs

Frequencies are in cycles per day unless stated otherwise (cycles per sidereal day where the IERS tables use them). Displacements are in metres, positive up, north and east. The tidal potential is $\phi_T = -g a (r/a)^n \bar{Y}_n^m(\theta) \cos m\lambda$ in the code, with $\bar{Y}_n^m$ the Schmidt semi-normalized associated Legendre functions. Complex Love numbers $L = L_R + iL_I$ multiply $e^{i\chi(t)}$, so that $L \cos \chi \rightarrow L_R \cos \chi - L_I \sin \chi$ as in the IERS Conventions; a negative L_I is a lag.

Signs of the potential. Three conventions meet in this manual. (i) In geodesy, and in the IERS Conventions, the tidal potential V is positive where the attraction is strongest and the force per unit mass is $+\nabla V$; the Love numbers are defined with this V (Section **Love numbers of a flattened Earth**). (ii) In the solver, potentials are potential *energies* per unit mass, Φ_0, ϕ_1, ϕ_T , with force $-\nabla\phi$ (the convention of Bos and Scherneck (2013)); hence $\phi_T = -V$, the minus sign in the code's tidal potential above. (iii) The closed-form solutions of Bos and Scherneck (2013) used in Example 3 are for a unit potential of the opposite sign, so there $h = -gU(a)$ and $l = -gV(a)$. All three describe the same deformation; the Love numbers of the package are those of convention (i).

Notation. Some symbols have two meanings, kept because both are standard; the context decides:

symbol	meaning	also
P_n	Legendre polynomial	radial function of ϕ_1 in the solver, $P_n(s)$
μ	shear modulus	$\cos \theta$ in the projection tutorial and the integrals $d\mu$

symbol	meaning	also
λ	Lamé parameter	longitude
f	flattening of the surface	frequency f (cycles per day) of a tidal wave
$\epsilon(s)$	flattening of the level surface s	ϵ : the coupling coefficient of the projection tutorial
k	Love number	Chebyshev index T_k ; $k = \sqrt{n(n+1)}$ in Example 3
h	Love number	step size of a finite difference (jets tutorial)
L	number of layers	a complex Love number $L_R + iL_I$; $L = \rho V_{sv}^2$ in the Voigt average
s	level coordinate (mean radius of a level surface)	–
θ, φ	colatitude, latitude (geocentric unless stated)	–
a, R_e	mean radius, equatorial radius	–
n, m, n_f	degree, order, degree of the forcing	–
K, N	kept degrees per field, Chebyshev polynomials per layer	–

4 How the solver works

The method. This chapter explains how `earthtides/solver.py` computes the Love numbers. The program is not a translation of long formulas: it evaluates the physics at a set of points and lets linear algebra do the rest. The chapter has four parts:

1. **Tutorials** (the first three sections): Chebyshev collocation, Galerkin projection, and jets with the chain rule, each on a problem in one variable that can be followed by hand.
2. **The big picture** (the next two): the equations of a deforming Earth, and how the solver module puts the three devices together (Figure 5).
3. **The solver step by step** (from The unknowns to Boundary and interface conditions): the blocks of the solver module in the order in which they act.
4. **Special cases and closing** (from Liquid–solid boundaries to the end): what is not simply more of the same – a liquid core, rigid rotations, degree 1 – and why the method is worth it.

Each device appears in the tutorials, is used in the solver and is tested by the examples of the Tutorial chapter:

device	tutorial	in the solver	tested by
Chebyshev collocation (radius)	collocation in one variable	unknowns, assembly, boundary rows	Examples 2, 3
Galerkin projection (latitude)	projection in one variable	projection in the angular direction	Examples 1, 2, 4
level-surface map, chain rule, jets	jets and the chain rule	physics at the nodes	Examples 2, 4; distorted mesh
liquid–solid boundaries	–	liquid–solid boundaries	Examples 3 (model a), 5
rotation and reference state	–	rigid rotations, reference potential	Example 5

4.1 Chebyshev collocation in one variable

Tutorial 1 of 3. Before the Earth, the radial half of the method on the simplest possible problem: a boundary value problem in one variable with a known solution ([Trefethen 2000](#), Program 13),

$$u''(x) = e^{4x} \quad \text{on } [-1, 1], \quad u(-1) = u(1) = 0, \quad (4.1)$$

whose exact solution is $u(x) = (e^{4x} - x \sinh 4 - \cosh 4)/16$.

The trial function. We write the unknown function as a sum of N Chebyshev polynomials, each with its own amplitude:

$$u(x) \approx \sum_{k=0}^{N-1} a_k T_k(x), \quad (4.2)$$

$$T_0 = 1, \quad T_1 = x, \quad T_2 = 2x^2 - 1, \quad T_3 = 4x^3 - 3x, \quad \dots,$$

with $T_k(\cos t) = \cos kt$. The N amplitudes a_k are the unknowns.

Derivatives are known exactly. Each T_k is a polynomial, so its derivatives are known analytically, and they are again sums of Chebyshev polynomials. For example $T_2'' = 4$ and $T_3'' = 24x$; in general the coefficients b_k of the derivative of $\sum a_k T_k$ follow from the recurrence $b_{k-1} = b_{k+1} + 2k a_k$ ([Boyd 2001](#)). The second derivative of the trial function at any point x is therefore $u''(x) = \sum_k a_k T_k''(x)$, a linear combination of the unknown amplitudes with known numbers $T_k''(x)$.

$N - 2$ equations from the differential equation. We require the differential equation to hold at $N - 2$ points inside the interval, the interior Gauss–Lobatto nodes

$$x_j = \cos \frac{\pi j}{N-1}, \quad j = 1, \dots, N-2. \quad (4.3)$$

At each node this gives one linear equation for the amplitudes: $\sum_k a_k T_k''(x_j) = e^{4x_j}$.

2 equations from the boundary conditions. The two remaining equations are the boundary conditions, $u(1) = \sum_k a_k T_k(1) = \sum_k a_k = 0$ and $u(-1) = \sum_k (-1)^k a_k = 0$.

N linear equations for N unknowns. Together this is a square linear system. For $N = 4$, with interior nodes $x_1 = \frac{1}{2}$ and $x_2 = -\frac{1}{2}$, it reads

$$\begin{pmatrix} 0 & 0 & 4 & 12 \\ 0 & 0 & 4 & -12 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} e^2 \\ e^{-2} \\ 0 \\ 0 \end{pmatrix}, \quad (4.4)$$

the first two rows being the differential equation at $x = \pm\frac{1}{2}$ ($T_2'' = 4$, $T_3'' = 24x = \pm 12$) and the last two the boundary conditions. Its solution is $a = (-0.941, -0.302, 0.941, 0.302)$, a rough approximation with four polynomials. Each further polynomial adds one node and one unknown, and the error falls exponentially (Figure 1): from 0.5 with $N = 4$ to 4×10^{-3} with $N = 8$, 10^{-9} with $N = 16$ and round-off with $N = 22$. This is what makes spectral methods economical: a few dozen coefficients per layer give the Love numbers to many digits.

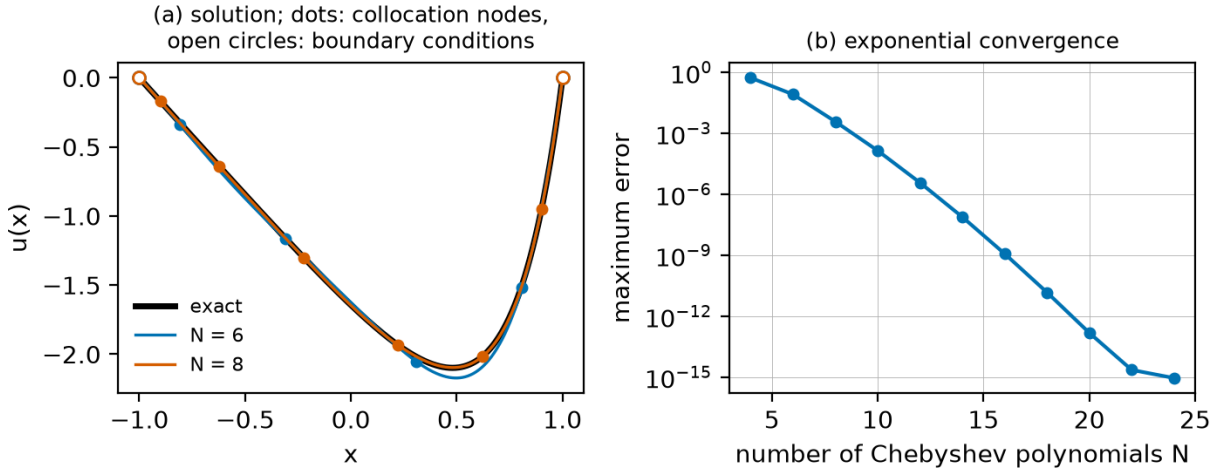


Figure 1: Chebyshev collocation for $u'' = e^{4x}$, $u(\pm 1) = 0$ (docs/manual/make_figures.py). (a) Exact solution and the collocation solutions with 6 and 8 polynomials; dots are the interior nodes where the equation is imposed, open circles the boundary conditions. (b) Maximum error against the number of polynomials N .

From one variable to the Earth. The solver does exactly this in every layer and for every field, with three differences. There are four fields (three displacement components and the potential) for every kept spherical degree instead of one function, so each node carries four equations per degree and each layer end as many conditions. The equations couple the fields, and on a flattened Earth also neighbouring spherical degrees, through the Galerkin projection in the angular direction (the pumped swing, Example 1, shows that projection in one variable; the drum, Example 2, combines both). And the boundary conditions at the interior layer ends are interface conditions that link neighbouring layers, which turns the separate layer systems into one block-banded system (Figure 6).

4.2 Galerkin projection in one variable

Tutorial 2 of 3. The angular half of the method, on a problem that has the structure of a flattened Earth: a function $u(\theta)$ on the sphere, symmetric about the axis, that satisfies

$$-\nabla_S^2 u + (1 + \varepsilon P_2(\mu)) u = P_2(\mu), \quad \mu = \cos \theta, \quad \varepsilon = 0.3. \quad (4.5)$$

∇_S^2 is the Laplacian on the unit sphere, for which $\nabla_S^2 P_n = -n(n+1)P_n$, and the right-hand side is a degree-2 forcing, like the tide. Without the term εP_2 the answer is a pure degree 2, $u = P_2/7$. The term $\varepsilon P_2(\mu) u$ plays the role of the flattening: a property that varies with latitude as P_2 , which makes the response depend on latitude and couples degree n to the degrees $n \pm 2$.

Where εP_2 comes from on the Earth. The solver uses level-surface coordinates (s, θ) : s is the mean radius of the level surface through a point, and

$$r = R(s, \theta) = s \left[1 - \frac{2}{3} \varepsilon(s) P_2(\cos \theta) \right] \quad (4.6)$$

with $\varepsilon(s)$ the flattening of that surface. Density and elastic moduli depend on s only, so along a level surface $s = s_j$, the row of collocation nodes, they are constant. The latitude dependence enters instead through the map: the equations need derivatives at fixed r and θ , and the chain rule converts them with the derivatives of the inverse map $s = S(r, \theta)$ (Section [Jets and the chain rule in one variable](#)). To first order in ε ,

$$\begin{aligned} s_r &= \frac{1}{R_s} \approx 1 + \frac{2}{3} (\varepsilon + s \varepsilon') P_2(\cos \theta), & s_\theta &= -\frac{R_\theta}{R_s} \approx -s \varepsilon \sin 2\theta, \\ r &\approx s \left[1 - \frac{2}{3} \varepsilon P_2(\cos \theta) \right]. \end{aligned} \quad (4.7)$$

So every radial derivative, $u_r = u_s s_r$, is multiplied by a coefficient of the form $1 + \text{const} \cdot P_2(\cos \theta)$, and the factors $1/r$ of the spherical components and the extra term $u_s s_\theta$ in the latitude derivative vary in the same way. The term $\varepsilon P_2(\mu) u$ of the toy problem stands for these factors: a coefficient that is not constant along the surface on which the residual is projected. In the solver nobody writes them as P_2 terms: they appear because the chain-rule factors (4.7) are evaluated at every colatitude θ_q of the quadrature.

The trial function. We write u as a sum of Legendre polynomials over K kept degrees, each with its own amplitude:

$$u(\mu) \approx \sum_n a_n P_n(\mu), \quad n = 0, 2, 4, \dots \quad (4.8)$$

(only even degrees are excited: the flattening couples a degree n only to $n \pm 2$, so a degree-2 forcing reaches the degrees 0, 2, 4, ... and never the odd ones; on the Earth the same holds for the spheroidal fields of a tide of degree n_f , which contain only degrees n with $n - n_f$ even).

The residual. Substituting the trial function into the equation and subtracting the right-hand side leaves the residual

$$E(\mu) = \sum_n a_n [n(n+1) + 1 + \varepsilon P_2(\mu)] P_n(\mu) - P_2(\mu), \quad (4.9)$$

a function of latitude. With a finite number of degrees it cannot vanish everywhere: the product $P_2 P_n$ contains the degrees $n-2$, n and $n+2$, so there is always a degree beyond the kept ones.

K equations from the projection. Instead we require that the residual contains *none of the kept degrees*, in the same way as a Fourier coefficient measures how much of a sine wave a signal contains:

$$\int_{-1}^1 E(\mu) P_k(\mu) d\mu = 0 \quad \text{for every kept degree } k. \quad (4.10)$$

This is the Galerkin projection: the test functions are the trial functions themselves. It gives one linear equation per kept degree, K equations for K amplitudes:

$$\sum_n a_n \left[(n(n+1) + 1) \frac{2\delta_{nk}}{2k+1} + \varepsilon G_{nk} \right] = \frac{2\delta_{2k}}{5}, \quad (4.11)$$

$$G_{nk} = \int_{-1}^1 P_2 P_n P_k d\mu.$$

The integrals G_{nk} are the *coupling coefficients*. For three Legendre polynomials they have a closed form, the Gaunt integral,

$$\int_{-1}^1 P_l P_n P_k d\mu = 2 \begin{pmatrix} l & n & k \\ 0 & 0 & 0 \end{pmatrix}^2, \quad (4.12)$$

with a Wigner $3j$ symbol, which vanishes unless $l+n+k$ is even and $|n-k| \leq l \leq n+k$. For $l=2$ only $k=n-2$, n and $n+2$ survive: this is why the flattening couples a degree only to its neighbours two apart. Here $G_{00} = 0$, $G_{02} = \frac{2}{5}$, $G_{22} = G_{24} = \frac{4}{35}$, $G_{44} = \frac{40}{693}$. Perturbation theory derives such coefficients by hand, for every term of the equations; the program never sees these formulas. It computes the same numbers by quadrature, which for these polynomial integrands is exact.

The projection by hand. Take the two degrees 0 and 2. The machinery has three steps, and they are exactly what the solver does at every node.

1. *The residual of each trial function at the quadrature points.* Apply the operator of the equation to each trial function alone, without the forcing: $\mathcal{L}P_0 = (1 + \varepsilon P_2)$ and $\mathcal{L}P_2 = (7 + \varepsilon P_2)P_2$, and evaluate these at the Gauss–Legendre points μ_q . Four points suffice here, because the integrands below are polynomials of degree at most 6 and a four-point rule is exact up to degree 7:

q	μ_q	weight w_q	$P_2(\mu_q)$	$\mathcal{L}P_0(\mu_q)$	$\mathcal{L}P_2(\mu_q)$
1	-0.86114	0.34785	0.61233	1.18370	4.39882
2	-0.33998	0.65215	-0.32662	0.90201	-2.25433
3	0.33998	0.65215	-0.32662	0.90201	-2.25433
4	0.86114	0.34785	0.61233	1.18370	4.39882

2. *Project each residual on each test function.* Multiply by the test function $P_k(\mu_q)$ and the weight, and sum over the points. The projection of $\mathcal{L}P_n$ on P_k is the matrix element in row k (the equation) and column n (the unknown a_n). For example, row 0, column 2 (with $P_0 = 1$):

$$A_{02} = \sum_q w_q \mathcal{L}P_2(\mu_q) P_0(\mu_q) = 2(0.34785 \cdot 4.39882) + 2(0.65215 \cdot (-2.25433)) = 0.12, \quad (4.13)$$

and row 2, column 2: $A_{22} = \sum_q w_q \mathcal{L}P_2(\mu_q) P_2(\mu_q) = 2.8343$. In the same way $A_{00} = 2$ and $A_{20} = 0.12$. The forcing P_2 is projected in the same way and gives the right-hand side, $(0, 0.4)$.

3. *Solve.* The system is

$$\begin{pmatrix} 2 & 0.12 \\ 0.12 & 2.8343 \end{pmatrix} \begin{pmatrix} a_0 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.4 \end{pmatrix}, \quad a_0 = -0.00849, \quad a_2 = 0.14149. \quad (4.14)$$

The off-diagonal element $0.12 = \varepsilon G_{02}$ is the coupling: it exists only because the coefficient $1 + \varepsilon P_2$ varies along the surface. With $\varepsilon = 0$ the residual of P_2 would be $7P_2$, whose projection on P_0 vanishes, and the degrees would decouple as on a sphere. With degree 2 alone, $a_2 = 0.4/2.8343 = 0.14113$.

In the solver the same three steps happen at every radial node s_j and for every trial field: `operator` evaluates the residual of the trial field at the $2n_{\max} + 24$ Gauss–Legendre colatitudes, with the chain-rule factors (4.7) at each of them (step 1); `_project` multiplies by the test functions and the weights and sums (step 2), which gives one column of the matrix; and the assembled system is solved at the end (step 3).

Exponential convergence. Each further degree removes the largest part of the residual (Figure 2a), and the error falls exponentially (Figure 2b): 7×10^{-2} with degree 2 alone, 8×10^{-3} with degrees 0 and 2, 2.5×10^{-5} with degrees 0, 2, 4 and 4×10^{-14} with six degrees. First-order perturbation theory in ε is limited to 4×10^{-3} : the projected system contains all orders of ε within the kept degrees.

From one variable to the Earth. On the Earth the residual at each radial node is a function of colatitude for each of the four equations (three momentum components and Poisson’s equation), and it is projected on the Schmidt functions $\bar{Y}_k^m$ and their derivatives (the vector test functions of Section [Galerkin projection in the angular direction](#)). The latitude dependence that couples the degrees does not come from one prescribed term εP_2 but from everything that varies along a

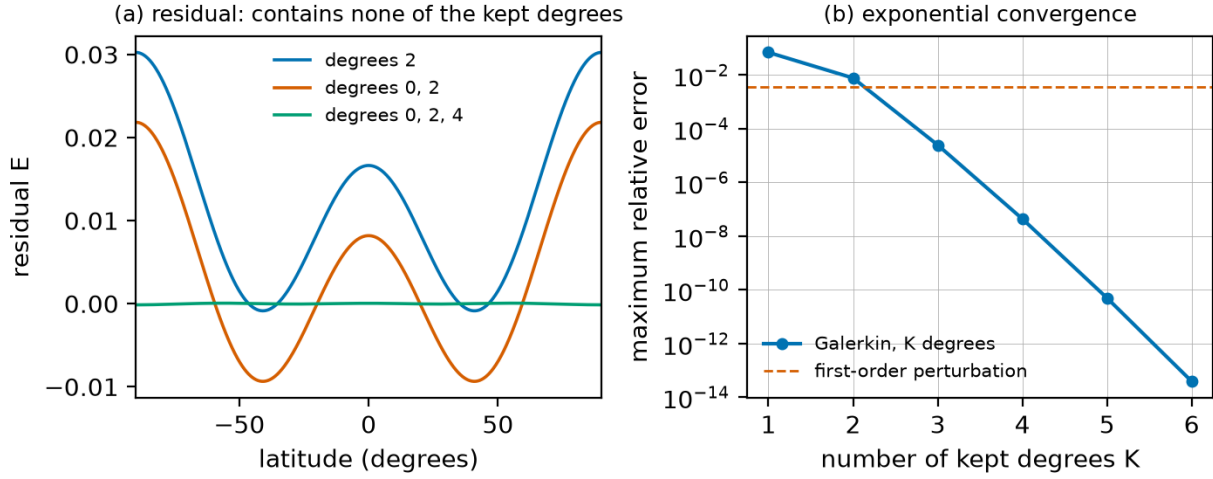


Figure 2: Galerkin projection for $-\nabla_S^2 u + (1 + 0.3P_2)u = P_2$ (docs/manual/make_figures.py). (a) The residual E for one, two and three kept degrees: it contains none of the kept degrees, only higher ones, and becomes small fast. (b) Maximum relative error against the number of kept degrees, and the error of first-order perturbation theory.

sphere of fixed radius in a flattened Earth: density, elastic moduli, gravity and the shape of the interfaces.

4.3 Jets and the chain rule in one variable

Tutorial 3 of 3. The equations need derivatives of quantities that are themselves built from derivatives: the momentum equation contains the divergence of the stress, and the stress contains the strain and the elastic moduli, which vary with position. A small problem shows the two devices the program uses for this. Take an elastic sphere, without gravity, in which every point moves only radially, $u_r = u(r)$.

Newton's law for a shell. The momentum equation is Newton's second law for a small piece of the sphere: mass times acceleration equals the net force on it,

$$\rho \ddot{u} = \frac{d\tau_{rr}}{dr} + \frac{2}{r}(\tau_{rr} - \tau_{\theta\theta}). \quad (4.15)$$

The right-hand side is the net elastic force per unit volume, the radial component of $\nabla \cdot \tau$ for radial motion. The first term is the difference between the radial stress on the outer and on the inner face of the piece; the second comes from its curved sides, where the hoop stress $\tau_{\theta\theta}$ pulls along the shell and, because the shell is curved, has a radial component. The stresses follow from the strain through Hooke's law,

$$\tau_{rr} = \lambda D + 2\mu u', \quad (4.16)$$

$$\tau_{\theta\theta} = \lambda D + 2\mu \frac{u}{r}, \quad (4.17)$$

$$D = u' + \frac{2u}{r}, \quad (4.18)$$

with D the relative change of volume. If the deformation is slow, as for a static load, the acceleration is negligible, $\rho\ddot{u} = 0$, and Newton's law becomes a balance of forces: the net force on every piece must vanish. For a tide the acceleration is kept: with $u \propto e^{i\omega t}$ it is $\rho\ddot{u} = -\rho\omega^2 u$, which adds $\rho\omega^2 u$ to the force side (eq. (4.27) of the solver). Here we keep the static case and define the residual

$$E = \frac{d\tau_{rr}}{dr} + \frac{2}{r}(\tau_{rr} - \tau_{\theta\theta}), \quad (4.19)$$

the net force per unit volume that a displacement leaves unbalanced; a solution has $E = 0$. To test the derivatives we take a displacement that is *not* a solution, $u = r^2$, with $\lambda = 2 + r$ and $\mu = 1 + r^2$ (dimensionless), and compute the force it leaves. By hand, $d\tau_{rr}/dr = \lambda'D + \lambda D' + 2\mu'u' + 2\mu u''$, and the result is $E = 16 + 8r + 16r^2$.

The chain rule through the map. In the program the fields are not stored as functions of r but of the level coordinate s : the Chebyshev polynomials are polynomials in s . In one variable the map is a stretch, here $r = R(s) = s(1 + 0.1s^2)$; on the Earth it is $R(s, \theta) = s[1 - \frac{2}{3}\epsilon(s)P_2(\cos \theta)]$. The derivatives with respect to s are converted with the derivatives of the inverse map, $s_r = 1/R'$ and $s_{rr} = -R''/R'^3$:

$$u_r = u_s s_r, \quad u_{rr} = u_{ss} s_r^2 + u_s s_{rr}. \quad (4.20)$$

At $s = 0.75$, $r = 0.79219$, the field $u(s) = R(s)^2$ has $u_s = 2RR'$ and $u_{ss} = 2R'^2 + 2RR''$, and the two formulas return $u_r = 1.58437 = 2r$ and $u_{rr} = 2.00000$, as they should. These two lines (with the θ -derivatives and mixed terms added) are all the program knows about the flattening.

Jets. The second device removes the hand work in $d\tau_{rr}/dr$. A *jet* is a pair: a number together with its derivative, (a, a') . Arithmetic on jets is defined so that the second entry of every result is the derivative of the first, which are the familiar rules of differentiation:

$$\begin{aligned} (a, a') + (b, b') &= (a + b, a' + b'), \\ (a, a') \cdot (b, b') &= (ab, a'b + ab'), \\ \frac{(a, a')}{(b, b')} &= \left(\frac{a}{b}, \frac{a' - (a/b)b'}{b} \right). \end{aligned} \quad (4.21)$$

A constant c is the jet $(c, 0)$, and the variable r itself is $(r, 1)$, because $dr/dr = 1$.

A jet differentiates nothing by itself. The rules only *combine* derivatives: to get the jet of λD one needs the jets of λ and of D , and so on back to the quantities that are not computed from others, the inputs. The derivatives of the inputs must be known in advance. What makes jets work in this solver is that every input is a function whose derivative is known exactly:

- *the coordinate*, $r = (r, 1)$, and the geometric factors $1/r$, $\cot \theta$, $1/\sin \theta$, whose derivatives are elementary;

- *the trial fields*: a Chebyshev polynomial in s times an angular function of θ , both differentiated exactly (the recurrence of the collocation tutorial, and the trigonometric form of $\bar{Y}_n^m$), and converted to derivatives at fixed r and θ by the chain rule (4.20);
- *the material parameters*: in the Earth model (the PREM file format of astyanax) density and seismic velocities are given in each layer as polynomials in radius, so ρ , λ and μ are polynomials in each layer too, and their derivatives are the derivatives of these polynomials (EarthModel, using NumPy's `Chebyshev.deriv`). On the flattened Earth they are functions of the level coordinate s , and the chain rule turns $d\mu/ds$ into $\partial\mu/\partial r$ and $\partial\mu/\partial\theta$ (props);
- *the reference state*: gravity follows from the integrated density polynomial, and the reference potential and its derivatives from gravity, again exactly.

Everything else, the strain, the volume change D , the stresses and the products in them, is *computed* from these inputs, and that is where the jets take over: each arithmetic operation carries the derivative along by the product or quotient rule. The derivative of a compound expression such as $\tau_{rr} = \lambda D + 2\mu u'$ is therefore never written down. In one variable that saves one line, $d\tau_{rr}/dr = \lambda' D + \lambda D' + 2\mu' u' + 2\mu u''$; on the flattened Earth the divergence of the stress in spherical components needs the r - and θ -derivatives of six stress components, each a combination of λ , μ , three displacement components and trigonometric factors, and written out it runs to dozens of terms.

One product by hand. The term $\mu u'$ of τ_{rr} , with $\mu = 1 + r^2$ and $u' = 2r$: the input jets are $(\mu, \mu') = (1 + r^2, 2r)$, from the model polynomial, and $(u', u'') = (2r, 2)$, from the trial function. The product rule gives $(\mu u', \mu' u' + \mu u'') = (2r + 2r^3, 6r^2 + 2)$, and the second entry is indeed the derivative of $2r + 2r^3$, although nobody differentiated $\mu u'$.

The example in numbers. At $s = 0.75$, $r = R(s) = 0.7921875$. The chain rule (4.20) gives $u_r = 1.584375$ and $u_{rr} = 2$ (for $u = r^2$ these are $2r$ and 2 , which checks it). The input jets and the jets computed from them are

jet	value	derivative	obtained from
r	0.7921875	1	the coordinate
λ	2.7921875	1	model, $2 + r$
μ	1.6275610	1.584375	model, $1 + r^2$
u	0.6275610	1.584375	trial field, chain rule
u_r	1.584375	2	trial field, chain rule
$D = u_r + 2u/r$	3.16875	4	jet arithmetic
$\tau_{rr} = \lambda D + 2\mu u_r$	14.005078	25.868232	jet arithmetic
$\tau_{\theta\theta} = \lambda D + 2\mu u/r$	11.426411	20.102866	jet arithmetic

and eq. (4.19) gives $E = 25.868232 + 2(14.005078 - 11.426411)/0.7921875 = 32.378477$, the exact $16 + 8r + 16r^2$. In the program a jet is the class `J` of `earthtides/jets.py`: a value and two derivatives (r and θ), with the operators $+$, $-$, $*$ and $/$ defined by eq. (4.21), about 40 lines. The solver builds the input jets of the trial fields in `_fields_at` and of the material in `props`, and evaluates the strain and stress with them in `_stress`.

The jet result differs from the exact value only by rounding, 2×10^{-16} relative. A finite difference of τ_{rr} would do the same job without hand work, but it trades truncation error against rounding error:

the best step gives 10^{-11} here, the wrong step much worse (Figure 3), and a second difference, which the radial equation of a flattened Earth needs, loses twice as many digits. Jets have no step to choose: they are exact up to rounding. This is first-order automatic differentiation.

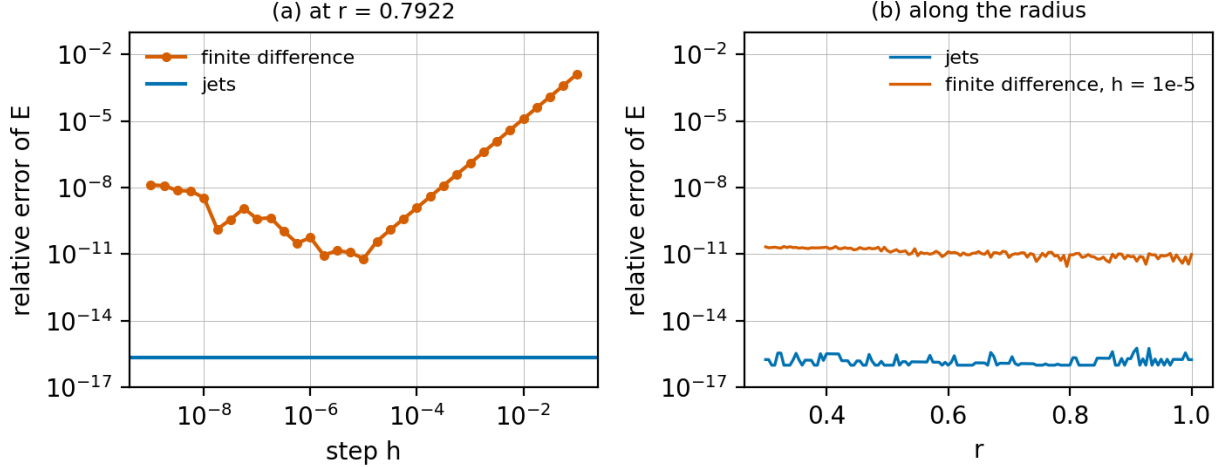


Figure 3: Jets and the chain rule for the radial equation of an elastic sphere, $u = r^2$, $\lambda = 2 + r$, $\mu = 1 + r^2$, fields stored on s with $r = s(1 + 0.1s^2)$ (docs/manual/make_figures.py). (a) Relative error of E against the step of a central difference of τ_{rr} , and with jets. (b) The same along the radius, finite difference with $h = 10^{-5}$, near the best step.

From one variable to the Earth. On the Earth a jet carries two derivatives, with respect to r and to θ at fixed r , and the divergence of the stress in spherical components takes both. The trigonometric factors are jets too: $\cot \theta$ enters as $(\cot \theta, 0, -1/\sin^2 \theta)$. The displacement components and the moduli enter exactly as above, with one derivative more, which is why the chain rule supplies second derivatives of the trial fields.

4.4 The equations of a deforming Earth

The big picture. With the three devices introduced, the chapter turns to the Earth. This section states the equations that are solved; the next one shows how the solver module is organized.

The reference state. Before the tide, every parcel of rock carries the weight of the rock above it. This load is carried by a pressure p_0 , as in a fluid at rest (hydrostatic equilibrium):

$$\nabla p_0 = -\rho_0 \nabla \Phi_0, \quad \nabla^2 \Phi_0 = 4\pi G \rho_0 - 2\Omega^2, \quad (4.22)$$

with ρ_0 the density and Φ_0 the potential energy per unit mass of gravity and centrifugal force together. The force per unit mass is $\mathbf{g} = -\nabla \Phi_0$, the gravity (including the centrifugal force), pointing down; this is the sign convention of Bos and Scherneck (2013). Density, pressure and the elastic parameters are constant on the level surfaces $\Phi_0 = \text{const}$, which rotation flattens into spheroids (Clairaut's equation), and pressure and density increase with depth.

The four forces. The tide moves a parcel from its reference position $\mathbf{x}$ to $\mathbf{x} + \mathbf{u}$. The displacement $\mathbf{u}$ is labelled by the reference position (Lagrangian), while the potential perturbation ϕ_1 , which also exists outside the Earth, and the density change at a fixed point,

$$\rho_1 = -\nabla \cdot (\rho_0 \mathbf{u}), \quad (4.23)$$

are Eulerian. After the displacement four forces per unit volume no longer balance (Figure 4). The first two act on the surface of the parcel, the last two, the body forces, on every part of its volume:

1. *the elastic force of the change of shape*, $\nabla \cdot \tau$, with the incremental stress of a linearly elastic, isotropic material

$$\tau = \lambda(\nabla \cdot \mathbf{u}) \mathbf{I} + \mu[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]; \quad (4.24)$$

2. *the pre-stress carried to a new level*, $-\nabla(\rho_0 \mathbf{u} \cdot \nabla \Phi_0)$: a parcel moved downward sits in a region of higher reference pressure, which pushes it back;
3. *the weight of the density change*, $-\rho_1 \nabla \Phi_0 = \nabla \cdot (\rho_0 \mathbf{u}) \nabla \Phi_0$: the parcel keeps its mass, so if its volume shrinks its density rises (eq. (4.23), the conservation of mass), and the extra mass has weight;
4. *the gravity of the moved mass and of the tide*, $-\rho_0 \nabla(\phi_1 + \phi_T)$, with ϕ_T the tidal potential of the Moon and the Sun.

Newton's second law and Poisson's equation. In a frame that rotates with the Earth, mass times acceleration plus the Coriolis force equals the sum of the four forces, and the moved mass ρ_1 is the source of ϕ_1 :

$$\begin{aligned} \rho_0 \ddot{\mathbf{u}} + 2\rho_0 \Omega \times \dot{\mathbf{u}} = & \nabla \cdot \tau - \nabla(\rho_0 \mathbf{u} \cdot \nabla \Phi_0) \\ & + \nabla \cdot (\rho_0 \mathbf{u}) \nabla \Phi_0 - \rho_0 \nabla(\phi_1 + \phi_T), \end{aligned} \quad (4.25)$$

$$\nabla^2 \phi_1 = -4\pi G \nabla \cdot (\rho_0 \mathbf{u}). \quad (4.26)$$

The change of the centrifugal force is contained in the terms with Φ_0 . Gravity and elasticity are equally important: their ratio, $\rho_0 g a / \mu$ (the weight of a column of rock as tall as the Earth against its rigidity), is about 2. Within a layer $\nabla \rho_0$ is parallel to $\nabla \Phi_0$, and the two middle terms can be written without a derivative of the density, $-\rho_0 \nabla(\mathbf{u} \cdot \nabla \Phi_0) + \rho_0 (\nabla \cdot \mathbf{u}) \nabla \Phi_0$; jumps of the density are handled by the interface conditions. For a spherical, non-rotating Earth eqs (4.25) and (4.26) are eqs (1.13) and (1.10) of Bos and Scherneck (2013).

One frequency at a time. A tidal wave varies harmonically, and the program looks for the steady response at its angular frequency ω in the rotating frame. With $\mathbf{u} \propto e^{i\omega t}$, $\ddot{\mathbf{u}} = -\omega^2 \mathbf{u}$ and $\dot{\mathbf{u}} = i\omega \mathbf{u}$; for a static calculation both are zero.

The residual E . Moving everything in eqs (4.25) and (4.26) to one side defines the residuals

$$\begin{aligned} \mathbf{E}_u = & \nabla \cdot \tau - \rho_0 \nabla(\mathbf{u} \cdot \nabla \Phi_0) + \rho_0 (\nabla \cdot \mathbf{u}) \nabla \Phi_0 - \rho_0 \nabla(\phi_1 + \phi_T) \\ & + \rho_0 \omega^2 \mathbf{u} - 2i\omega \rho_0 \Omega \times \mathbf{u}, \\ E_\phi = & \nabla^2 \phi_1 + 4\pi G \nabla \cdot (\rho_0 \mathbf{u}). \end{aligned} \quad (4.27)$$

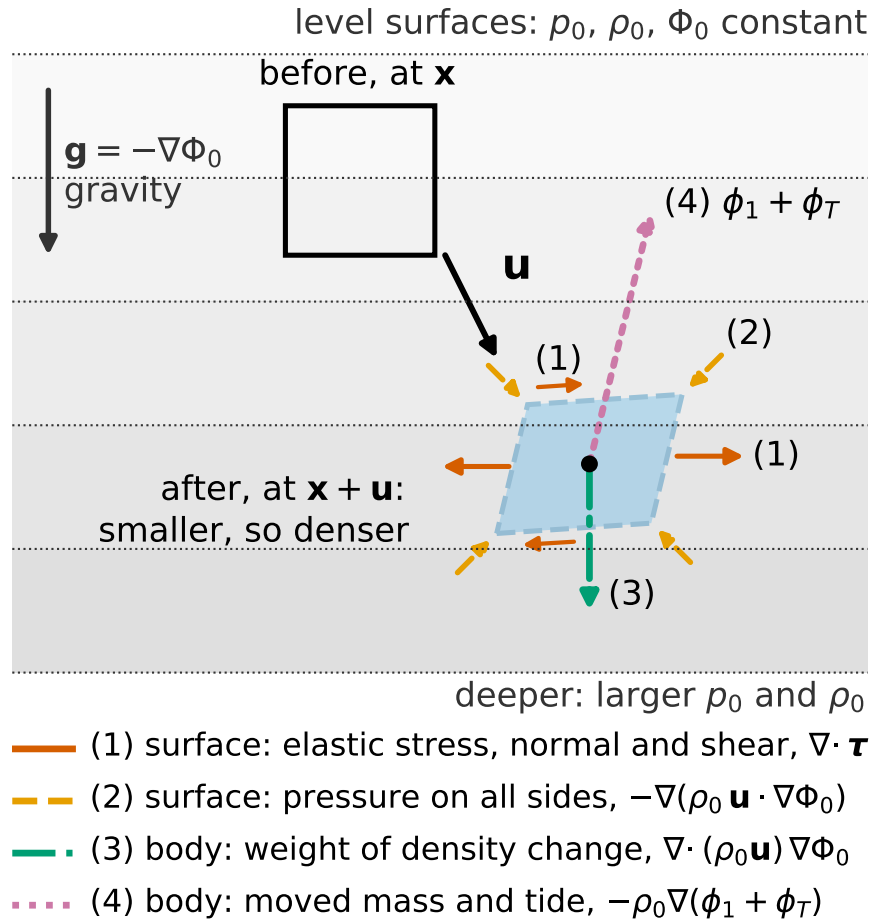


Figure 4: The four forces on a parcel of rock moved by $\mathbf{u}$ (`docs/manual/make_figures.py`). Dotted lines are level surfaces of the reference state; pressure and density increase downward (darker). After the move the parcel is sheared and compressed. Two forces act on its surface: the elastic stress, across the faces (normal) and along them (shear), (1), and the higher reference pressure of its new level, on all sides, (2). Two are body forces, acting on the whole volume and drawn from its centre: the weight of the density change, (3), and the gravity of all moved mass, ϕ_1 , and of the Moon and the Sun, ϕ_T , (4). Colours (Okabe–Ito) and line styles distinguish the four forces.

The true solution makes $\mathbf{E}_u = 0$ and $E_\phi = 0$ everywhere: that is what the equations say. For a trial field, a finite sum of Chebyshev polynomials and angular functions, they do not vanish, and $\mathbf{E}_u$ and E_ϕ measure by how much the trial field fails, exactly as $E = u'' - e^{4x}$ in the collocation tutorial and the residual of the Galerkin tutorial. Every term except the tidal force is linear in the unknown fields; the tidal force $-\rho_0 \nabla \phi_T$ does not depend on them and becomes the right-hand side of the linear system. The program requires the residuals to vanish at the radial nodes (collocation) and to contain none of the kept angular functions (projection). In spherical components $\mathbf{E}_u = (E_r, E_\theta, E_\lambda)$.

4.5 The solver module

The equations and the three devices come together in the module `solver.py` (class `TidalSolver`, about 830 lines, with the basis functions in `basis.py` and the jets in `jets.py`). Figure 5 shows its structure: each block is a group of functions with one task, and the arrows show which blocks use which.

Before reading on, it helps to map the tutorials onto these blocks.

- **set-up** (`__init__`) chooses the kept degrees, builds the angular test functions and the Gauss–Legendre points once, and finds the liquid layers.
- **basis** makes the trial fields. The Chebyshev polynomials and their exact derivatives of the collocation tutorial come from `cheb_basis`, scaled to a layer by `_radial`; the angular functions of the Galerkin tutorial come from `ybar`; `_fields_at` multiplies the two and, with the chain rule of the jets tutorial (`geometry.chain`, through `_comp`), delivers every trial field with its physical derivatives at fixed r and θ .
- **physics at the nodes** evaluates the residuals (4.27) of every trial field at the nodes (`operator`, with the stress (4.24) built from jets in `_stress`: the numbers $T_k''(x_j)$ of the collocation tutorial), and the boundary quantities traction and gravity flux (`interface`). The material parameters, the map and the reference potential at the nodes come from `props`.
- **Galerkin projection** (`_project`) turns a residual, a function of colatitude, into one equation per kept degree, as in the Galerkin tutorial.
- **assembly** (`solve` and the `_..._rows` functions) puts every row in its place: the interior rows, and the boundary and interface conditions at the two ends of every layer – the two extra rows per field of the collocation tutorial.
- **interface rows** handles the special cases: the liquid–solid interfaces, the tidal force and the rigid-rotation (Tisserand) rows.
- **linear system** (`_solve_linear`) scales and solves the system.
- **output** turns the solution into surface fields and Love numbers.

The following sections take these blocks in order: the unknowns (basis), the residuals at the nodes (physics at the nodes), their projection, the assembly of the linear system and the boundary and interface conditions.

4.6 The unknowns

The solver step by step. The Earth is divided into layers (12 for PREM: inner core, outer core, eight mantle layers and two crustal layers, the ocean merged into the upper crust). In every layer the four fields – the three components of the displacement $\mathbf{u}$ and the perturbation ϕ_1 of the gravitational

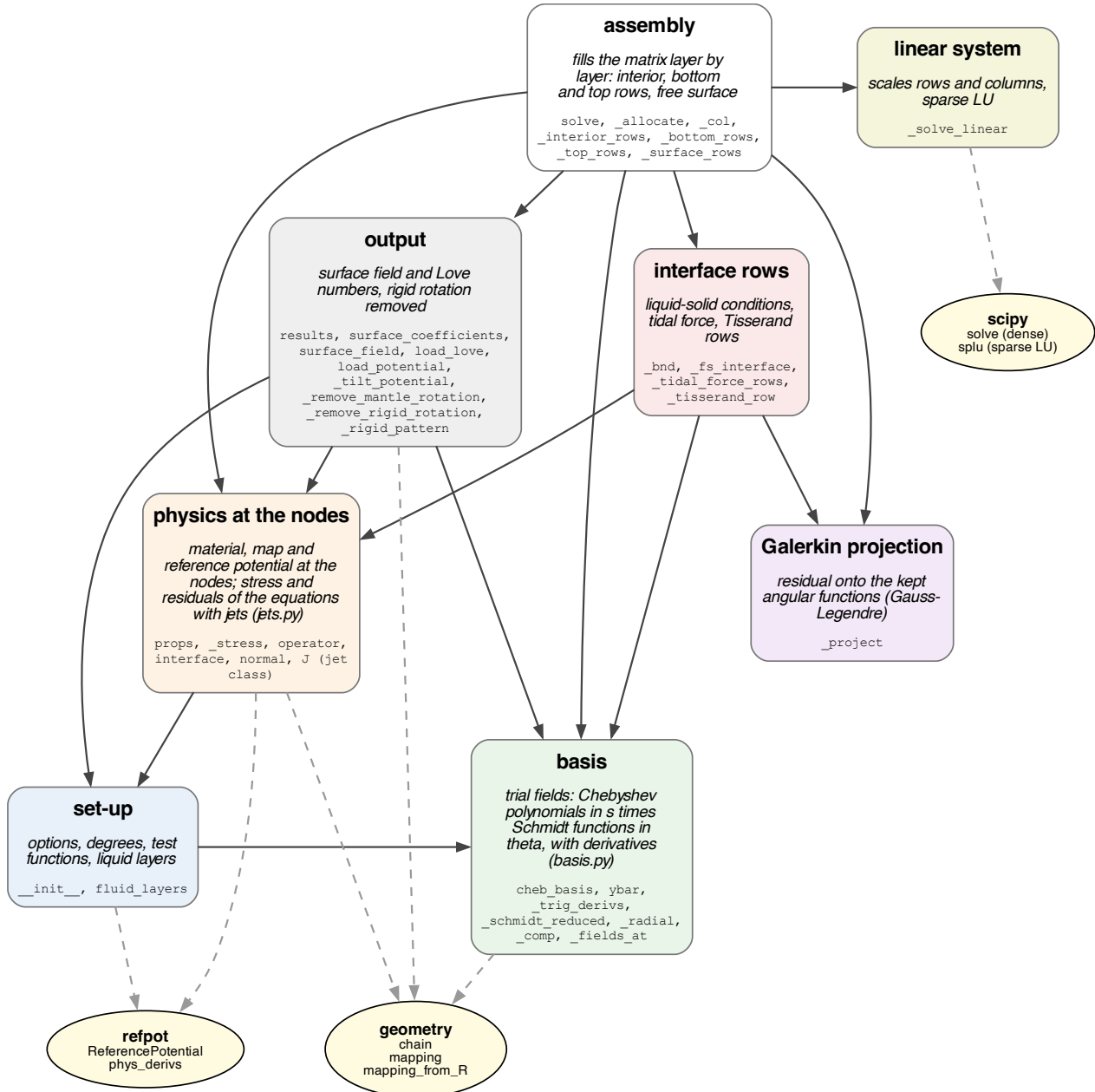


Figure 5: The structure of the solver (`solver.py`, `basis.py`, `jets.py`), generated from the source by `docs/manual/make_figures.py`. Each block is a group of functions with one task (its functions in small type); an arrow means that functions of one block call functions of the other (two-headed if both ways), aggregated from the call graph; dashed lines lead to functions imported from other modules (yellow). `solve` in the assembly block fills the matrix layer by layer, using the trial fields of the basis, the residuals of the physics block, the projection and the interface rows, and hands the system to the linear solver; the output block turns the solution into surface fields and Love numbers.

potential – are written as sums over a few spherical degrees n of a radial function times an angular function:

$$\begin{aligned}
 u_r &= \cos m\lambda \sum_n U_n(s) \bar{Y}_n^m(\theta), \\
 u_\theta &= \cos m\lambda \left[\sum_n V_n(s) \partial_\theta \bar{Y}_n^m - \sum_{n'} W_{n'}(s) \frac{m \bar{Y}_{n'}^m}{\sin \theta} \right], \\
 u_\lambda &= \sin m\lambda \left[- \sum_n V_n(s) \frac{m \bar{Y}_n^m}{\sin \theta} + \sum_{n'} W_{n'}(s) \partial_\theta \bar{Y}_{n'}^m \right], \\
 \phi_1 &= \cos m\lambda \sum_n P_n(s) \bar{Y}_n^m(\theta).
 \end{aligned} \tag{4.28}$$

Here s labels the level surfaces of the flattened Earth (its mean radius) and θ is colatitude; the components are the ordinary spherical components at the point $r = R(s, \theta)$. U, V, P (spheroidal) run over the degrees n with $n - n_f$ even, W (toroidal) over those with $n - n_f$ odd; for the M_2 tide ($n_f = 2, m = 2$) and $K = 3$ these are $n = 2, 4, 6$ and $n' = 3, 5, 7$. Each radial function is a Chebyshev series in the layer,

$$U_n(s) = \sum_{k=0}^{N-1} a_{nk} T_k(x), \quad x = \frac{2s - s_0 - s_1}{s_1 - s_0} \in [-1, 1]. \tag{4.29}$$

The unknowns of the problem are the coefficients a_{nk} : N per field, per degree and per layer. As in the collocation tutorial their derivatives are exact, with $d/ds = 2/(s_1 - s_0) d/dx$ (`_radial`), and the equations are imposed at the $N - 2$ interior Gauss–Lobatto nodes of each layer, which cluster towards the layer boundaries, where the fields change fastest; the two end points are left for the boundary and interface conditions. For PREM with $K = 4$ and $N = 24$ there are 4612 unknowns. The basis is built by `cheb_basis` (module `basis`; values and first two derivatives of the T_k) and `ybar` (the angular functions and their θ -derivatives, computed exactly from a trigonometric polynomial); `_fields_at` combines them into the trial fields.

4.7 The physics at the nodes

At each collocation node s_j and at the Gauss–Legendre colatitudes θ_q the program evaluates, for every trial field, the residuals E_r, E_θ, E_λ and E_ϕ of eq. (4.27) (`operator`). These are the same expressions, in the same spherical components, as for a spherical Earth; nothing in them refers to the flattening.

The derivatives that these expressions need are obtained as in the tutorial **Jets and the chain rule in one variable**: the trial fields arrive at the nodes with their derivatives at fixed r and θ through the chain rule of the map (`geometry.mapping, geometry.chain`; the material parameters, gravity and reference potential depend on s only and get their gradients the same way, `props`), and the strain, stress and stress divergence are evaluated with jets in two variables (`_stress`). The flattening enters through $\epsilon(s)$ in the map and nowhere else.

The result of `operator` for one trial field is an array of residuals E_r , E_θ , E_λ , E_ϕ at the nodes (s_j, θ_q) : the contribution of that basis field to the equations.

4.8 Galerkin projection in the angular direction

No integral in the radial direction. The two directions are treated differently: in s the equations are imposed *pointwise*, at the collocation nodes s_j of each layer (Chebyshev collocation, nothing is integrated over s); in θ they are *projected*, by an integral over colatitude at fixed s_j . The Chebyshev polynomials are only evaluated and differentiated at the nodes. At a node s_j the residual is a function of θ . With a finite number of degrees it cannot vanish at every θ ; instead it is required to contain none of the angular functions that are kept (`_project`):

$$\begin{aligned} \int E_r \bar{Y}_k^m d\mu &= 0, & \int E_\phi \bar{Y}_k^m d\mu &= 0, \\ \int \left(E_\theta \partial_\theta \bar{Y}_k^m - E_\lambda \frac{m \bar{Y}_k^m}{\sin \theta} \right) d\mu &= 0, & \int \left(-E_\theta \frac{m \bar{Y}_{k'}^m}{\sin \theta} + E_\lambda \partial_\theta \bar{Y}_{k'}^m \right) d\mu &= 0, \end{aligned} \quad (4.30)$$

with $d\mu = \sin \theta d\theta$, one condition per field and kept degree. The integrals are Gauss–Legendre sums over $2n_{\max} + 24$ colatitudes. On a sphere the integrands are products of spherical harmonics and these sums give the closed forms (4.12) exactly. On a flattened Earth the integrands also contain the chain-rule factors (4.7) and $1/\sin \theta$, which are not polynomials in $\cos \theta$, and there is no simple closed form; but they are smooth, and the sums are converged to round-off. This is a Galerkin projection: the test functions are the trial functions themselves.

On a sphere a degree- n trial field produces a residual of degree n only, and the projection picks out the classical radial equation of that degree. On a flattened Earth the residual also contains degrees $n \pm 2$, because the coefficients of the equations vary along the level surface – the chain-rule factors (4.7) and the geometric factors, as the term εP_2 in the projection tutorial; the projection on degree $n \pm 2$ is then the *coupling coefficient* of perturbation theory, obtained here as a number from a quadrature instead of a formula.

4.9 Assembling the linear system

`solve` builds one matrix row per projected condition and one column per unknown coefficient (Figure 6), layer by layer:

1. **Interior rows** (`_interior_rows`). For every trial field, the projected residuals at the $N - 2$ interior nodes fill the field's N columns. The tidal force $\rho_0 \nabla \phi_T$ goes to the right-hand side.
2. **Boundary rows** (`_bottom_rows`, `_top_rows`). The two remaining rows per field of every layer carry the conditions at its bottom and top (next section); the top rows of the last layer hold the free surface (`_surface_rows`).
3. **Bordering rows and columns** (`_surface_rows`, `_tisserand_row`). The exterior potential coefficients c_n (from which the Love number k follows) and, for static runs with $m = 0$ or 1 , one unknown rigid-rotation force per region with its Tisserand condition (below) add a few dense rows and columns.

The number of conditions equals the number of unknowns, so the system is square.

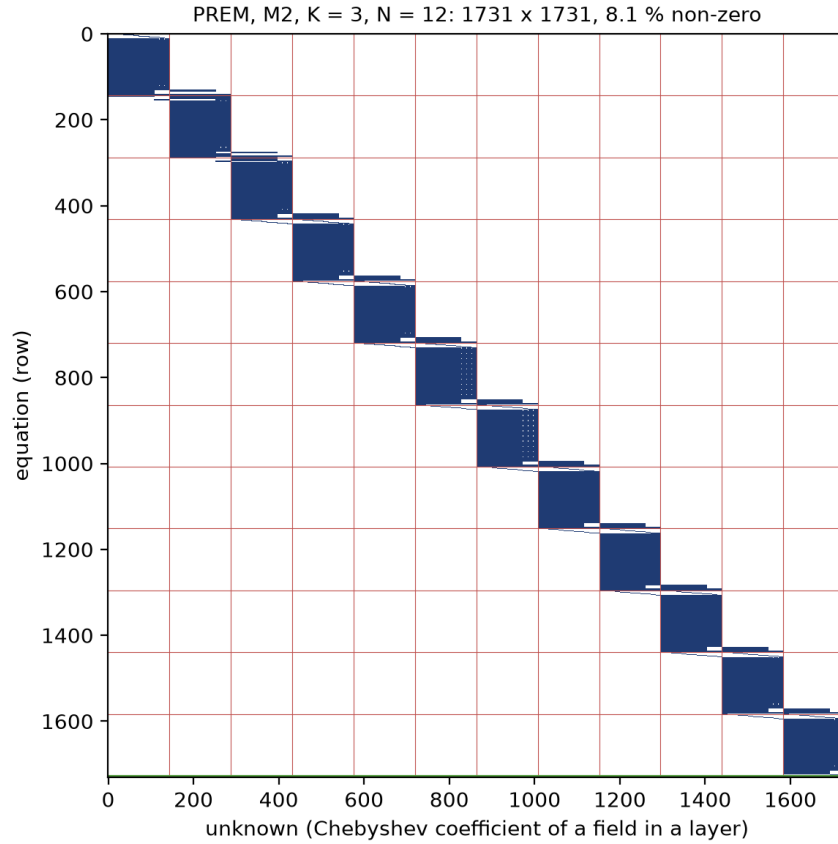


Figure 6: Non-zero entries of the system matrix for PREM at M_2 with $K = 3$, $N = 12$ (`make_figures.py`). Each dense block is one layer (all fields and degrees of that layer are coupled by the projection); the short horizontal strips between blocks are the interface rows that link a layer to its neighbour. Only 8 per cent of the entries are non-zero.

4.10 Boundary and interface conditions

How many conditions. Every field is governed by an equation of second order in s : the momentum equation for U_n , V_n , W_n and Poisson's equation for P_n . As in the one-variable tutorial, the N Chebyshev coefficients of a field in a layer are fixed by the $N - 2$ collocation equations at the interior nodes and *two* further conditions, one at the bottom and one at the top of the layer. With n_{fld} fields per layer,

$$n_{\text{fld}} = K_U + K_V + K_W + K_P, \quad (4.31)$$

the numbers of kept degrees of U , V , W and P ($K_U = K_P = K$; $K_V = K$ unless degree 0 is kept, which has no V ; K_W toroidal degrees), each layer has $n_{\text{fld}}N$ unknowns, $n_{\text{fld}}(N - 2)$ interior rows and $2n_{\text{fld}}$ boundary rows: n_{fld} at the centre or at the interface below, n_{fld} at the interface above or the surface. In the language of the radial equations of a spherical Earth: per spheroidal degree

a layer has six constants of integration (U , V , P , two each), per toroidal degree two. For M_2 with $K = 3$ ($n = 2, 4, 6$; $n' = 3, 5, 7$) $n_{\text{fld}} = 12$; for PREM with $K = 4$ and $N = 24$ the 12 layers have $12 \cdot 16 \cdot 24 = 4608$ unknowns, plus the $K = 4$ exterior coefficients c_n introduced at the free surface.

Because the level surfaces are coordinate surfaces $s = \text{const}$, every interface is a surface on which $x = \pm 1$ in the two layers it separates. The conditions are therefore imposed where the interfaces actually are, without the correction terms that perturbation theory needs to transfer them from a mean sphere. Each condition is a function of θ and is projected on the kept angular functions exactly like the interior equations, which turns one physical condition into one row per kept degree.

The quantities at a boundary (interface). With $\hat{\mathbf{n}}$ the unit normal of the flattened level surface (normal; $\hat{\mathbf{n}} \propto (1, -R_\theta/R)$ in (r, θ) components), the traction is the stress on the surface,

$$\begin{aligned} \mathbf{t} &= \hat{\mathbf{n}} \cdot \boldsymbol{\tau}, \\ t_r &= \tau_{rr}n_r + \tau_{r\theta}n_\theta, \quad t_\theta = \tau_{r\theta}n_r + \tau_{\theta\theta}n_\theta, \quad t_\lambda = \tau_{r\lambda}n_r + \tau_{\theta\lambda}n_\theta, \end{aligned} \quad (4.32)$$

its normal part is $t_n = \hat{\mathbf{n}} \cdot \mathbf{t}$ and its tangential part $\mathbf{t} - t_n \hat{\mathbf{n}}$. The gravity condition concerns the flux

$$q = \hat{\mathbf{n}} \cdot \nabla \phi_1 + 4\pi G \rho_0 \mathbf{u} \cdot \hat{\mathbf{n}}, \quad (4.33)$$

whose second term is the mass that the displaced boundary carries across a density jump: a surface density $\rho_0 \mathbf{u} \cdot \hat{\mathbf{n}}$, which makes $\hat{\mathbf{n}} \cdot \nabla \phi_1$ jump by $4\pi G$ times that density. Projected, t_r gives one row per spheroidal degree (on $\bar{Y}_k^m$), (t_θ, t_λ) one row per V degree and one per W degree (on the vector test functions), and q one row per P degree: n_{fld} rows in all. On a sphere these are the classical conditions on τ_{rr} (normal stress), $\tau_{r\theta}$, $\tau_{r\lambda}$ (shear stress) and on $\partial_r \phi_1 + 4\pi G \rho_0 u_r$; on the flattened Earth the tilted normal mixes them, and the projection couples the degrees.

Centre. Nothing is integrated from the centre, so the centre does not supply *initial conditions*. In the classical method (the propagator or Runge–Kutta integration of the radial equations of a spherical Earth; see Bos and Scherneck (2013)) one starts at a small radius with the three regular solutions, known analytically in a small homogeneous sphere, integrates them outward and combines them at the surface. Here the centre is an ordinary boundary of the first layer, where n_{fld} conditions select the solutions that are regular there. Near $r = 0$ a regular field of degree n behaves as

$$U_n, V_n \propto r^{n-1}, \quad W_n \propto r^n, \quad P_n \propto r^n \quad (4.34)$$

(degree 0: $U_0 \propto r$, $P_0 \rightarrow \text{const}$), so the conditions at $s = 0$ are $U_n = V_n = W_n = 0$ and $dP_n/ds = 0$; for $n = 1$ they are different (Section Degree 1). The coordinates are singular at the centre (the equations contain $1/r$ and $1/r^2$), but they are never evaluated there: the collocation nodes are interior points and only the regularity conditions sit at $s = 0$. The map $R = s[1 - \frac{2}{3}\epsilon P_2]$ shrinks to a point at the centre, whatever $\epsilon(0)$.

Welded interface (solid–solid, or between two fluid layers). Four physical conditions, each for all kept degrees:

- *the displacement is continuous*: the two layers stay in contact, with no holes opening and no overlap ($[U_n] = [V_n] = [W_n] = 0$);
- *the potential is continuous*: $[\phi_1] = 0$;
- *the traction is continuous*: $[\mathbf{t}] = 0$ (Newton's third law for the interface);
- *the gravity flux is continuous*: $[q] = 0$.

The bottom rows of the upper layer hold the continuity of the field values ($n_{\text{fld}} \text{ rows, } \text{_bottom_rows}$) and the top rows of the lower layer the continuity of traction and flux ($n_{\text{fld}} \text{ rows, } \text{_top_rows}$): $2n_{\text{fld}}$ conditions for the $2n_{\text{fld}}$ boundary rows that the two layers bring to the interface.

Free surface (_surface_rows). Outside the Earth ϕ_1 is a potential field, $\phi_1 = \sum_n c_n (a/r)^{n+1} \bar{Y}_n^m \cos m\lambda$, with K unknown coefficients c_n . At $s = a$, that is on the flattened surface $r = R(a, \theta)$:

$$\begin{aligned} \mathbf{t} = 0 \quad (\text{load: } \mathbf{t} = -g\sigma\hat{\mathbf{n}}), \quad \phi_1 = \sum_n c_n \left(\frac{a}{R}\right)^{n+1} \bar{Y}_n^m, \\ q = \hat{\mathbf{n}} \cdot \nabla \left[\sum_n c_n \left(\frac{a}{r}\right)^{n+1} \bar{Y}_n^m \right] \quad (\text{load: } q - \hat{\mathbf{n}} \cdot \nabla[\dots] = -4\pi G\sigma), \end{aligned} \quad (4.35)$$

where σ is the surface density of a load. The traction gives $K_U + K_V + K_W$ rows and the continuity of ϕ_1 K rows, which fills the n_{fld} top rows of the last layer; the continuity of the flux gives K more rows, placed after the layers, for the K unknowns c_n . The system is square: $L n_{\text{fld}} N + K$ rows and unknowns for L layers. The Love numbers are read off at the surface, $h = U_{n_f}(a)/a$, $l = V_{n_f}(a)/a$, $k = -c_{n_f}/(ga)$. Because $(a/R)^{n+1}$ is evaluated on the flattened surface, the exterior coefficients of all kept degrees are coupled too.

4.11 Liquid–solid boundaries and the Longman paradox

Special cases. The sections so far describe a solid Earth. This one and the next two treat what needs more than the general machinery: a liquid core, rigid rotations and degree 1.

The outer core is a liquid without viscosity: it carries pressure but no shear stress. At its boundaries with the inner core and the mantle the liquid may therefore slide along the boundary, and the tangential displacement need not be continuous. For a hydrostatic Earth the linearized conditions are (Tromp and Mitrovica 1999, eq. 3.7):

- the normal displacement $\mathbf{u} \cdot \hat{\mathbf{n}}$ is continuous (no holes, no overlap);
- the normal traction $\hat{\mathbf{n}} \cdot \boldsymbol{\tau} \cdot \hat{\mathbf{n}}$ is continuous;
- the tangential traction vanishes on the solid side;
- ϕ_1 and the gravity flux are continuous as before.

These are fewer conditions than at a welded interface: the continuity of the two tangential displacement components is dropped. To keep the system square, the freed rows of the liquid layer carry the *tangential momentum equation* at the boundary node instead, that is, the equation of motion of the liquid itself, evaluated on the boundary ($\text{_bnd, } \text{_fs_interface}$).

A liquid layer has a consequence that a solid does not. In a solid the shear stress resists every tangential displacement. In an inviscid liquid nothing does, except inertia (and the Coriolis force):

the tangential motion of the liquid is determined by $\rho_0 \omega^2 \mathbf{u}$. In the static limit $\omega \rightarrow 0$ the tangential displacement in the core is not determined by the equations at all; this is the static core paradox (Longman 1963). Numerically the system becomes singular as $\omega \rightarrow 0$: its condition number grows, first in the highest kept degrees of the core flow, which are held only by inertia. At M_2 the condition number is already $\sim 10^{14}$, and two equally valid solutions differ by $\sim 10^{-10}$ in the Love numbers; at K_1 , exactly at the frequency of the free tilt-over mode, by $\sim 10^{-6}$.

earth-tides therefore always solves the liquid core at the frequency of the tide, with inertia (and, by default, the Coriolis force), and never statically. For waves with frequencies below 0.005 cycles per day (periods longer than 200 days: the 18.6-year tide and the long nodal waves), `earthtides-love` solves the core dynamics at 0.005 cycles per day while taking the anelasticity at the wave's own frequency. Static calculations can use the model `PREMs`, in which the outer core is a weak solid (shear velocity 1 m/s); it is unusable with inertia, because its shear waves would be only tens of kilometres long.

4.12 Rigid rotations and the reference frame

A tide of order $m = 1$ exerts a torque on the equatorial bulge and tilts the Earth (the forced nutation); a zonal tide ($m = 0$) changes its rotation rate. With inertia and Coriolis forces in every layer these rotations follow from the equations, including the free core nutation, and the station displacement is defined in a frame attached to the mantle: the mean rotation of the mantle (its Tisserand rotation) is removed afterwards (`_remove_mantle_rotation`; the alternative `frame='surface'` removes the rotation that fits the surface best). In static runs there is no dynamics to fix the rotations; there each region (inner core, outer core, mantle) gets an unknown rigid-rotation force and the condition that its mean rotation vanishes (`_tisserand_row`).

4.13 Degree 1

Spheroidal fields of degree 1 occur only for odd forcing degrees with $m \leq 1$ (a degree-3 tide on the flattened Earth, a degree-1 load, the response of a laterally heterogeneous Earth); the Love numbers of degree 2 and the spherical degree-3 Love numbers of `earthtides-love` contain none. Toroidal degree 1 (a rigid rotation) occurs in every run with $m \leq 1$ and was treated in the previous section. Degree 1 is special in three ways.

The centre moves. For $n = 1$ the regular solutions have $U_1, V_1 \rightarrow \text{const}$ at the centre: the displacement of the centre itself, one vector, so $U_1(0) = V_1(0)$. The potential grows as r , so the regular condition is $P_1(0) = 0$, not the condition $dP_n/ds = 0$ of the other degrees; with the latter, the degree-1 load Love numbers of a homogeneous sphere would be off by 7×10^{-4} .

A translation costs nothing. A rigid translation T of the whole Earth together with its field, $U_1 = V_1 = T$ and $\phi_1 = -Tg(r)$, satisfies the static equations without any force. Static degree-1 results are therefore defined only up to this translation, that is, up to the choice of frame: centre of mass of the whole system (CM), of the solid Earth (CE) or of the surface figure (CF) (Blewitt 2003). A translation shifts h'_1, l'_1 and k'_1 by the same amount, so $h'_1 - k'_1$ and $l'_1 - k'_1$ do not depend on the frame. The solver fixes the frame by $U_1(0) = V_1(0) = 0$: the centre of the Earth stays where it is. To change the frame, subtract the same constant from all three: CE has $k'_1 = 0$, CM has $k'_1 = -1$. For PREM (static):

frame	h'_1	l'_1	k'_1
centre fixed (solver)	-0.8899	-0.5004	-0.6040
CE	-0.2858	+0.1036	0
CM	-1.2858	-0.8964	-1

With inertia the centre is not free. At the frequency of a tide a translation costs $\rho_0 \omega^2 T$, and the motion of the centre is part of the solution. Pinning the centre is then a constraint that the true solution does not obey: for the homogeneous sphere at M_2 it changes $h'_1 - k'_1$ by 2×10^{-5} and $l'_1 - k'_1$ by 2×10^{-4} , for PREM $h'_1 - k'_1$ by 9×10^{-4} . The option `centre='free'` replaces the pinning by the regular behaviour of a moving centre, $U_1, V_1 = T + O(s^2)$:

$$U_1(0) = V_1(0), \quad \frac{dU_1}{ds}(0) = 0, \quad P_1(0) = 0, \quad (4.36)$$

and leaves the translation T to inertia. It needs `omega_f` or `omega_t` (a static run with a free centre is singular, and the solver refuses it). For the homogeneous sphere it reproduces the exact solution, which contains the moving centre (`examples/ex3/analytic.py`), to seven digits at every frequency: for a load the solid Earth's centre of mass then stays at rest and $k'_1 = 0$ comes out by itself, the CE frame; a degree-1 tidal potential, a uniform force, moves the whole sphere rigidly by the amount of the exact solution ($h = l = k = -78.09$ at M_2).

The free centre gives the motion in an *inertial* frame. That is the right frame for a load (CE), but not for a body tide on a flattened Earth: the tide pulls on the equatorial bulge, the net force accelerates the whole Earth (the force of the tide on J_2 , part of the orbital motion), and the free centre follows it. For the degree-3 O_1 tide ($m = 1$) of PREM the translation is 140 times the equilibrium tide, because in inertial space this force varies with a period of 13.7 days, and the translated J_2 field changes k_3 from +0.096 to -0.283. Geodesy wants the geocentric frame, in which that acceleration is removed together with the degree-1 term of the tidal potential; for body tides keep the default `centre='fixed'`.

4.14 Solving the system

Rows and columns are scaled to unit maximum first (`_solve_linear`) (the Tisserand rows are $\sim 10^{30}$ times larger than the others), and the system is solved with a sparse LU factorization (SuperLU with COLAMD ordering). One solution takes about one second; the dense LAPACK solver is available as `linear_solver='dense'` and gives the same Love numbers within the precision quoted above.

4.15 Why this is not one headache traded for another

Closing. This section returns to the framework of the Introduction.

A perturbation theory needs, for every term of the equations, the formulas of its coupling to the neighbouring degrees, derived by hand to first order in the flattening and programmed term by term; a new term means new formulas. Here a developer writes:

1. the field equations as pointwise expressions in spherical components (`operator`, about 35 lines), the same as for a spherical Earth;
2. the stress, the boundary tractions and fluxes (`_stress`, `interface`, `_bnd`);
3. the conditions at the boundaries (which rows hold which condition).

Everything else is generic and does not depend on the physics: the basis, the chain rule through the map, the jets, the projection and the linear algebra. Adding the Coriolis force, for example, took a few lines in `operator`, added to the residuals E_r , E_t , E_p of the radial, colatitude and longitude components of the momentum equation:

```
# excerpt of TidalSolver.operator (solver.py): inertia and Coriolis force of a rotating run
if self.coriolis or self.omega_t:
    # the fields are Re[(A, B, -iC) exp(i(omega t + m lambda))], with A, B, C the amplitudes of
    # u_r, u_theta, u_lambda; then rho omega^2 u - 2 i omega rho Omega x u is real in (A, B, C)
    rs_, wt = rho * p.solid, self.omega_t      # density (zero in a liquid unless dyn_fluid), omega
    cw = 2.0 * wt * self.coriolis              # 2 omega Omega
    sn, cs = p.sin, p.cos                     # sin and cos of colatitude at the nodes
    Er = Er + rs_ * (wt**2 * e.A.v + cw * sn * e.C.v)
    Et = Et + rs_ * (wt**2 * e.B.v + cw * cs * e.C.v)
    Ep = Ep + rs_ * (wt**2 * e.C.v + cw * (sn * e.A.v + cs * e.B.v))
```

Here $e.A.v$, $e.B.v$, $e.C.v$ are the values of the jets of the three displacement components of the trial field (Section [Jets and the chain rule in one variable](#)), and `p` holds the material and geometry at the nodes (`props`).

Errors are caught by tests with exact answers rather than by comparison with another implementation of the same algebra. The strongest is the *distorted mesh*: a spherical Earth described with deliberately bulging level surfaces must give the spherical Love numbers, which exercises the map, the chain rule, the jets, the projection and the interface conditions at once [Bos (2026), Sect. 4.3; `tests/test_mapping.py`]. The price of the method is computing time (seconds instead of milliseconds per tide) and the conditioning of the liquid core discussed above.

5 Tutorial

Verification, then use. The first five sections are the examples in `examples/` (copied by `earthtides-examples`). They form a ladder of increasing complexity: each adds one part of the method to the previous one and is checked against a solution known independently – analytical where possible – so that an error would show up at the rung where it enters. Each example has a README, a script and the expected output; run the scripts from their own directory. The test suite runs them all. The last two sections use the programs for the real Earth: the Love numbers of PREM and the body tide at a station.

section	topic	tests	runtime
Example 1 (ex1)	the pumped swing	Galerkin projection, in time instead of latitude	1 s
Example 2 (ex2)	the loaded elliptical drum	level-curve map, chain rule, collocation and projection together	1 s

section	topic	tests	runtime
Example 3 (ex3)	homogeneous spheres	the elastic-gravitational equations on a sphere; a liquid core	2 s
Example 4 (ex4)	a flattened homogeneous Earth	the flattening with self-gravitation and the centrifugal force	6 min
Example 5 (ex5)	Hough's Earth	rotation, Coriolis force and a liquid core: the free core nutation	20 s
Love numbers of PREM	earthtides-love	– (use)	12 s
A station time series	etide	– (use)	1 s

5.1 Example 1: the pumped swing

Problem. A swing pushed once per period while its rider pumps it standing up: crouching at the ends lowers her centre of mass and weakens the restoring force, twice per period. With time scaled so that the push has frequency one,

$$x'' + \omega_0^2 (1 + \varepsilon \cos 2t) x = \cos t, \quad \omega_0^2 = 2, \varepsilon = -0.1. \quad (5.1)$$

This is a forced Mathieu equation ([Mathieu 1868](#)). The term $\varepsilon \cos 2t$ plays the role of the flattening: it is even, like $P_2(\cos \theta) = (1 + 3 \cos 2\theta)/4$, and couples harmonic n to $n \pm 2$, as the flattening couples degree n to $n \pm 2$.

Solution. The periodic solution has no closed form; the reference is the periodic solution obtained by numerical integration, $a_1 = 1.1093501285$ (the amplitude of $\cos t$). The method of the package – a trial function $x = \sum a_n \cos nt$ over K odd harmonics, and the requirement that the residual contains none of them – is harmonic balance ([Nayfeh and Mook 1979](#)). It gives closed forms for small K :

$$K = 1 : \quad a_1 = \frac{1}{\omega_0^2(1 + \varepsilon/2) - 1} = 1.1111, \quad (5.2)$$

$$K = 2 : \quad a_1 = \frac{\omega_0^2 - 9}{[\omega_0^2(1 + \varepsilon/2) - 1](\omega_0^2 - 9) - (\varepsilon\omega_0^2/2)^2} = 1.109350, \quad (5.3)$$

the off-diagonal $\varepsilon\omega_0^2/2$ being the coupling coefficient. First-order perturbation theory in ε gives $a_1 \approx 1/(\omega_0^2 - 1) - (\varepsilon\omega_0^2/2)/(\omega_0^2 - 1)^2 = 1.1$.

Result. The error of a_1 is 1.8×10^{-3} , 1.1×10^{-7} and 10^{-12} for $K = 1, 2, 3$, against 9×10^{-3} for first-order perturbation theory: the truncated system contains all orders of ε within the kept harmonics.

5.2 Example 2: the loaded elliptical drum

Problem. A membrane stretched over an elliptical frame $x^2/a^2 + y^2/b^2 = 1$ ($a = 1, b = 1 - f$), fixed at the rim and loaded:

$$-\nabla^2 u = q \quad \text{inside,} \quad u = 0 \quad \text{on the rim.} \quad (5.4)$$

It is solved with exactly the steps used for the Earth (Figure 7): level-curve coordinates $r = sR(\theta)$, in which the rim is $s = 1$ and every curve $s = \text{const}$ is a similar ellipse; a trial function $u = \sum_k U_k(s) \cos 2k\theta$; Galerkin projection in θ ; Chebyshev collocation in s with $U_k(1) = 0$ at the rim and regularity at the centre. In these coordinates ($q_R = R'/R$)

$$R^2 \nabla^2 u = (1 + q_R^2) u_{ss} + (1 + q_R^2 - q'_R) \frac{u_s}{s} + \frac{u_{\theta\theta}}{s^2} - 2q_R \frac{u_{s\theta}}{s}, \quad (5.5)$$

and the flattening enters only through $q_R \simeq -f \sin 2\theta$.

Elliptical drum (flattening 0.2): level curves and nodes

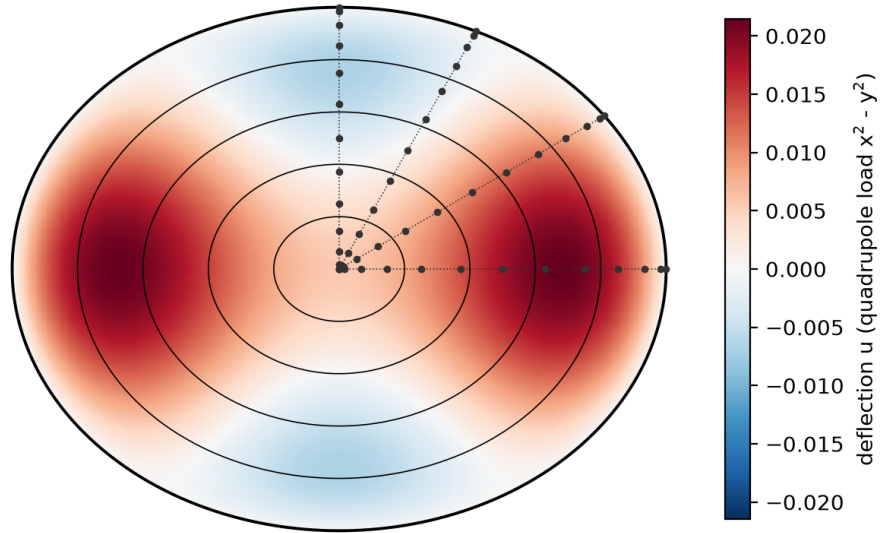


Figure 7: Example 2 with an exaggerated flattening of 0.2: the frame, level curves $s = \text{const}$, the Chebyshev nodes on four rays, and the exact solution for the quadrupole load.

Solution. For polynomial loads the solution is a polynomial times $g = 1 - x^2/a^2 - y^2/b^2$, which vanishes on the rim. With $A = a^{-2} + b^{-2}$:

- uniform load $q = 1$: $u = g/(2A)$. This is the classical solution for an elliptical membrane, the same function as Prandtl's stress function for the torsion of a bar of elliptical cross section

(Timoshenko and Goodier 1951). On the level curves $g = 1 - s^2$: it contains no θ at all, and the method reproduces it to round-off with a single harmonic;

- quadrupole load $q = x^2 - y^2$ (the analogue of the degree-2 tide): $u = g(\alpha + \beta x^2 + \gamma y^2)$, where $\nabla^2 u = -q$ gives, for the constant, x^2 and y^2 terms,

$$\begin{pmatrix} -2A & 2 & 2 \\ 0 & -2A - 10/a^2 & -2/a^2 \\ 0 & -2/b^2 & -2A - 10/b^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \quad (5.6)$$

(derived for this package, Bos (2026), Sect. 3.5). This solution needs all harmonics.

Result. For the quadrupole load the error falls exponentially with the number of harmonics K : 4×10^{-2} , 10^{-4} and 9×10^{-10} for $K = 2, 4, 8$ at $f = 0.1$; with the Earth's flattening $K = 6$ reaches round-off. First-order perturbation theory is limited to errors of order f^2 .

5.3 Example 3: homogeneous spheres

Model b: a homogeneous sphere ($\rho = 5517 \text{ kg m}^{-3}$, $\mu = 146 \text{ GPa}$, $\lambda = 347 \text{ GPa}$, radius 6371 km, $G = 6.673 \times 10^{-11}$) at the M_2 frequency $\omega = 1.40526 \times 10^{-4} \text{ rad/s}$.

Solution. Gravity is proportional to r and the radial equations have an exact solution in spherical Bessel functions (Love 1911; Pekeris and Jarosch 1958; Dahlen and Tromp 1998, Sect. 8.7); we follow Bos and Scherneck (2013) (Appendix 2). For degree n there are three solutions regular at the centre. With $k = \sqrt{n(n+1)}$, v_α , v_β the compressional and shear velocities,

$$\gamma_\pm^2 = \frac{\omega^2}{2v_\beta^2} + \frac{\omega^2 + \frac{16}{3}\pi G\rho}{2v_\alpha^2} \pm \frac{1}{2} \left[\left(\frac{\omega^2}{v_\beta^2} - \frac{\omega^2 + \frac{16}{3}\pi G\rho}{v_\alpha^2} \right)^2 + \left(\frac{8\pi G k \rho}{3v_\alpha v_\beta} \right)^2 \right]^{1/2}, \quad (5.7)$$

$$\zeta = \frac{3}{4}(\pi G\rho)^{-1}v_\beta^2(\gamma^2 - \omega^2/v_\beta^2), \quad \xi = \zeta - (n+1), \quad (5.8)$$

the two Bessel solutions (one for each $\gamma_\pm$) and the r^n solution are

$$\begin{aligned} U^\pm &= \frac{n\xi}{r} j_n(\gamma r) - \zeta \gamma j_{n+1}(\gamma r), & U^\oplus &= n r^{n-1}, \\ V^\pm &= \frac{\xi}{r} j_n(\gamma r) + \gamma j_{n+1}(\gamma r), & V^\oplus &= r^{n-1}, \\ P^\pm &= -4\pi G\rho \zeta j_n(\gamma r), & P^\oplus &= (\omega^2 - \frac{4}{3}\pi G\rho n) r^n, \end{aligned} \quad (5.9)$$

with the radial stress R , tangential stress S and gravity Q of Bos and Scherneck (2013) (eqs 1.138–1.143). Three amplitudes follow from the surface conditions for a unit potential ($R = S = 0$, $Q = (2n+1)/a$ for the body tide; $R = (2n+1)g/(4\pi Ga)$ in addition for the load), and then $h = -gU(a)$, $l = -gV(a)$, $k = P(a) - 1$. Note: eq. 1.132 of Bos and Scherneck (2013) as printed lacks the ω^2 in the second term inside the square root; only the form given here reproduces their Table 1.3 (at degree 2 the difference in h is 6×10^{-4} at M_2). `examples/ex3/analytic.py`

evaluates these formulas and reproduces all twelve numbers of their Table 1.3 for degrees 2 and 10.

Result. The solver reproduces Table 1.3 to the five digits given, for the body tide and the load, at degrees 2, 10 and 100.

Model a: a homogeneous mantle over a liquid core (Figure 8).

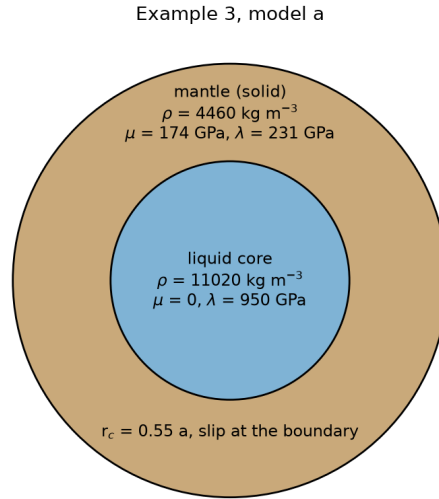


Figure 8: Example 3, model a: a homogeneous solid mantle over a homogeneous liquid core of radius $0.55 a$. The core slides along the boundary (Chapter [How the solver works](#)).

Solution. Gravity is no longer proportional to r and there is no closed form. Bos and Scherneck (2013) (Table 1.4) used the Bessel solutions layer by layer by assuming $g \propto r$ in each layer, as Alterman et al. (1959); their numbers are therefore approximate (4 per cent in h , 15 per cent in l). The reference is an independent solver of the radial equations with exact gravity (astyanax, described in Bos and Scherneck (2013)).

Result. Agreement with the radial-equation solver to 10^{-9} at periods of 6, 12 and 24 h, for the body tide and the load. The model tests the liquid–solid slip conditions on a sphere.

5.4 Example 4: a flattened homogeneous Earth

Problem. A homogeneous, incompressible, self-gravitating elastic Earth with its hydrostatic flattening, under the semidiurnal tide (Greff-Lefftz et al. 2005): $\rho = 5520 \text{ kg m}^{-3}$, $a = 6371 \text{ km}$, the rigidity expressed by $\bar{\mu} = 19\mu/(2\rho ga) = 3.15592$, incompressibility approximated by $\lambda = 10^5\mu$. Their flattening is $\alpha = \frac{5}{4}q_0 = 1/232$, with $q_0 = \Omega^2 a/g$.

Solution. On a sphere the Love numbers are Kelvin's (Thomson 1863; Love 1911):

$$h_2 = \frac{5/2}{1 + \bar{\mu}}, \quad l_2 = \frac{3/4}{1 + \bar{\mu}}, \quad k_2 = \frac{3/2}{1 + \bar{\mu}}. \quad (5.10)$$

Greff-Lefftz et al. (2005) derived the corrections of first order in α , in a basis of the ellipsoid (their eqs 38, 48 and 52): with

$$\delta h_2 = \frac{\alpha}{399} \frac{653\bar{\mu} + 1349}{(1 + \bar{\mu})^2}, \quad \delta h_4 = -\frac{4\alpha}{35} \frac{68\bar{\mu} + 95}{(1 + \bar{\mu})(51\bar{\mu} + 38)}, \quad (5.11)$$

the displacement normal to the ellipsoid has the degree-2 and degree-4 parameters

$$h_2 + \Delta h_o = h_2 + \delta h_2 - \frac{11}{21} \frac{\alpha}{1 + \bar{\mu}} = 0.603140, \quad \Delta h_+ = \delta h_4 - \frac{1}{7} \frac{\alpha}{1 + \bar{\mu}} = -0.000332, \quad (5.12)$$

and the tangential displacement

$$\begin{aligned} l_2 + \Delta l_o &= l_2 + \frac{\alpha}{1995} \frac{1825\bar{\mu} + 2869}{(1 + \bar{\mu})^2} = 0.181544, \\ \Delta l_+ &= -\frac{\alpha}{1260} \frac{3825\bar{\mu} + 3382}{(1 + \bar{\mu})(51\bar{\mu} + 38)} = -0.000064, \\ \Delta l_* &= -\frac{13}{60} \frac{\alpha}{1 + \bar{\mu}} = -0.000224 \quad (\text{toroidal, degree 3}). \end{aligned} \quad (5.13)$$

For the geoid on the deformed surface they find $k_2 + \Delta k_o = 0.363824$ and $\Delta k_+ = -0.000633$ (their eq. 53). These closed forms reproduce their quoted numbers to the last digit.

Result. All seven parameters agree to 5×10^{-6} or better when, as Greff-Lefftz et al. (2005) do, the reference gravity in the perturbation equations is purely gravitational. With the centrifugal force on the displaced parcel included, as in the rest of the package, the degree-2 parameters change by 2.5×10^{-4} (l) to 8×10^{-4} (h).

5.5 Example 5: the free core nutation of Hough's Earth

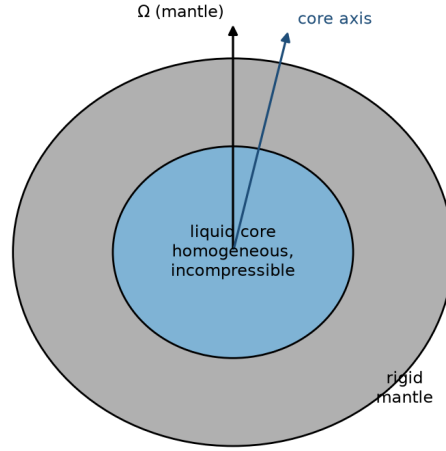
Problem. A rigid, flattened mantle containing a homogeneous, incompressible liquid core (Figure 9). Because the core–mantle boundary is flattened, the core can rotate about an axis slightly inclined to that of the mantle, coupled to it by the pressure on the boundary; this free mode is the free core nutation.

Solution. For a rigid mantle the frequency of the mode in the rotating frame is (Hough 1895; in modern form Requier 2022)

$$\omega_{\text{FCN}} = \Omega \left[1 + \frac{A}{A_m} e_f \right], \quad (5.14)$$

with $e_f = (C_f - A_f)/A_f$ the dynamical ellipticity of the core and A, A_m the equatorial moments of inertia of the Earth and of the mantle; the period in space is $1/[(A/A_m)e_f]$ sidereal days. The

Example 5: the free core nutation of Hough (1895)



similar Maclaurin spheroids (flattening exaggerated)

Figure 9: Example 5: Hough's Earth. A rigid mantle and a homogeneous, incompressible liquid core, both similar Maclaurin spheroids (flattening exaggerated). In the free core nutation the axis of the core is inclined to the rotation axis of the mantle.

body is homogeneous ($\rho = 5500 \text{ kg m}^{-3}$) with the exact equilibrium shape for its rotation rate, a Maclaurin spheroid of eccentricity e given by (Chandrasekhar 1969)

$$\frac{\Omega^2}{\pi G \rho} = \frac{2\sqrt{1-e^2} (3-2e^2) \arcsin e}{e^3} - \frac{6(1-e^2)}{e^2}, \quad f = 1 - \sqrt{1-e^2}, \quad (5.15)$$

and for similar homogeneous layers $e_f = (a^2 - c^2)/(a^2 + c^2)$ and $A_m/A = 1 - (r_c/a)^5$, so the prediction has no free parameter. For the Earth's rotation rate $f = 1/231$ and the period is 219.36 sidereal days.

Result. The solver finds the mode as the frequency at which its linear system becomes singular: 219.41 sidereal days, a relative difference of 2×10^{-4} ; for a body four times as flattened 9×10^{-4} , as expected for a formula of first order in e_f . Without the reference potential computed from the model's density the frequency is wrong by 2.5 per cent: the free core nutation tests the whole reference state, not only the Coriolis force.

5.6 Love numbers of PREM

The command

```
earthtides-love -o love.json
```

computes the Love numbers of PREM with the default options: anelastic (absorption band), rotating (Coriolis force, fluid outer core with inertia), hydrostatically flattened, with the centrifugal force in the reference gravity, and in the IERS conventions. It uses the default wave groups (Section [Wave groups](#)): one solution per group, at the frequency of the group's largest wave, spread over the

processor cores. The default set has 22 groups: 21 with degree-2 waves, each needing one solution, and a last group, HIGH, that contains only waves of degrees 3 and 4 and needs none. On a laptop this takes about 12 seconds on eight cores, and the program reports

```
1/21 solved (8 s)
22 groups: 21 solved, 1 with only degrees 3 and 4 (no solve)
1200 entries -> love.json (12 s)
```

The file `love.json` has an entry for each of the 1200 waves of the Tamura catalogue, so that `etide` can use it (Section [Love-number table](#)). The table below lists the degree-2 values of every group, at the frequency where they were computed. The columns $h^{(2)}$, $l^{(1)}$ and l' , h' are in units of 10^{-4} ; the last column holds h' for the zonal groups and l' for the diurnal ones. Only the real parts are listed except for $h^{(0)}$; the imaginary parts of the others are of the same relative size.

Table 1: Love numbers of anelastic, hydrostatic PREM per wave group, from `earthtides-love -o love.json` (IERS conventions; k at the mean radius). Mn is the 18.6-year nodal tide; K_1^- and K_1^+ are the nodal sidebands of K_1 .

group	wave	f (cpd)	Re $h^{(0)}$	Im $h^{(0)}$	$h^{(2)}$	$l^{(0)}$	$l^{(1)}$	l', h'	k
L01	Mn	0.000147	0.64357	-0.01188	-6.18	0.09531		1.19	0.32199
L02	Sa	0.002738	0.62676	-0.00715	-5.67	0.09045		1.08	0.31234
L03	Ssa	0.005476	0.62386	-0.00637	-5.59	0.08964		1.06	0.31068
L04	Mm	0.036292	0.61743	-0.00467	-5.41	0.08786		1.02	0.30700
L05	Mf	0.073202	0.61551	-0.00417	-5.35	0.08733		1.00	0.30590
L06	Mtm	0.109494	0.61450	-0.00391	-5.32	0.08705		1.00	0.30532
D01	Q_1	0.893244	0.60753	-0.00281	-5.25	0.08583	4.93	-3.06	0.30186
D02	O_1	0.929536	0.60678	-0.00280	-5.29	0.08584	4.98	-3.04	0.30148
D03	M_1	0.966446	0.60452	-0.00279	-5.37	0.08591	5.00	-3.02	0.30032
D04	π_1	0.994524	0.59198	-0.00282	-5.87	0.08632	4.81	-2.95	0.29395
D05	P_1	0.997262	0.58590	-0.00284	-6.12	0.08651	4.69	-2.92	0.29086
D06	S_1	1.000000	0.57406	-0.00289	-6.71	0.08689	4.46	-2.87	0.28476
D07	K_1^-	1.002591	0.53111	-0.00294	-8.20	0.08834	3.65	-2.69	0.26313
D08	K_1	1.002738	0.52590	-0.00294	-8.39	0.08851	3.55	-2.67	0.26049
D09	K_1^+	1.002885	0.51993	-0.00294	-8.64	0.08871	3.43	-2.64	0.25746
D10	ψ_1	1.005476	0.93283	0.00053	7.51	0.07503	11.42	-4.42	0.46680
D11	ϕ_1	1.008214	0.66391	-0.00248	-3.03	0.08393	6.22	-3.26	0.33044
D12	J_1	1.039030	0.61452	-0.00272	-4.96	0.08556	5.32	-3.03	0.30538
D13	OO_1	1.075940	0.61181	-0.00272	-5.01	0.08564	5.35	-3.01	0.30401
S01	M_2	1.932274	0.61180	-0.00251	-5.19	0.08609	10.29		0.30495
S02	S_2	2.000000	0.61188	-0.00250	-5.16	0.08607	10.42		0.30498

Three effects stand out.

- **Anelasticity.** The mantle is softer at long periods: $h^{(0)}$ grows from 0.612 at the semidiurnal frequencies to 0.644 at the 18.6-year tide, and the imaginary part, a lag, from -0.0025 to -0.012 .

- **The free core nutation.** In the diurnal band $h^{(0)}$ falls from 0.607 at O_1 to 0.526 at K_1 , jumps to 0.933 at ψ_1 and returns to 0.664 at ϕ_1 . The resonance lies between K_1 and ψ_1 (at $1 + 1/457$ cycles per sidereal day for PREM), and beyond it the imaginary part changes sign. $h^{(2)}$, $l^{(1)}$, l' and k follow the same resonance. Even the two nodal sidebands of K_1 , only 0.000147 cycles per day away, differ by 0.005 in $h^{(0)}$, which is why the default groups give them groups of their own.
- **The latitude terms** $h^{(2)}$ and $l^{(1)}$ are of the order of the flattening times the main terms, l' exists only in the diurnal band and h' only for the zonal tides, as in the IERS parametrization (Section [Love numbers of a flattened Earth](#)).

The last group (HIGH, above 2.45 cycles per day) contains only waves of degrees 3 and 4. These, like the degree-3 and degree-4 waves of every other group, get spherical Love numbers at the M_2 frequency, and the permanent tide gets the nominal IERS values.

Other examples:

```
earthtides-love --all-waves -o all.json          # every Tamura wave, about 5 minutes on 8 cores
earthtides-love --elastic -o elastic.json        # no anelasticity
earthtides-love --frame surface --latitude geodetic -o test.json # other conventions
```

A table in other conventions than those of the IERS is for comparison only; `etide` refuses it.

5.7 A station time series

`etide` computes the body-tide displacement of a station as a time series. For the space-geodetic station at Onsala, Sweden (geodetic latitude 57.395° , east longitude 11.926°), 30 days from 1 January 2026, hourly:

```
etide 57.395 11.926 --start 2026-01-01 --days 30 --step 3600 > onsala.txt
```

The file starts with

#	MJD (UTC)	up [m]	north [m]	east [m]
61041.000000	0.099463	-0.040228	-0.058562	
61041.041667	0.040177	-0.023449	-0.067201	
61041.083333	-0.022182	-0.007244	-0.066342	
61041.125000	-0.079094	0.005336	-0.057307	
61041.166667	-0.124150	0.012244	-0.042945	

and has 720 lines of data: the modified Julian date (UTC) and the displacement up, north and east in metres. For every epoch `etide` computes the astronomical arguments of the 1200 waves of the Tamura catalogue (degrees 2 to 4), with UTC converted to dynamical time through the table of leap seconds; converts the geodetic latitude of the station to the geocentric latitude of the IERS basis functions; and sums the waves, each multiplied by its Love numbers. “Up” is along the geocentric radius (Section [The six choices behind the parameters](#)). Without `--love` the Love numbers are those of the IERS Conventions ([Petit and Luzum 2010](#)).

Figure 10 shows the result. The up component varies between -17 and $+21$ cm. At this latitude the diurnal waves are large: their pattern, $\sin 2\varphi$, is largest at 45° , that of the semidiurnal waves, $\cos^2 \varphi$, at the equator. When the Moon is far from the equator (days 0 to 5, 15 to 20 and 27 to 30) the diurnal waves dominate and the station rises once a day, with a small second maximum;

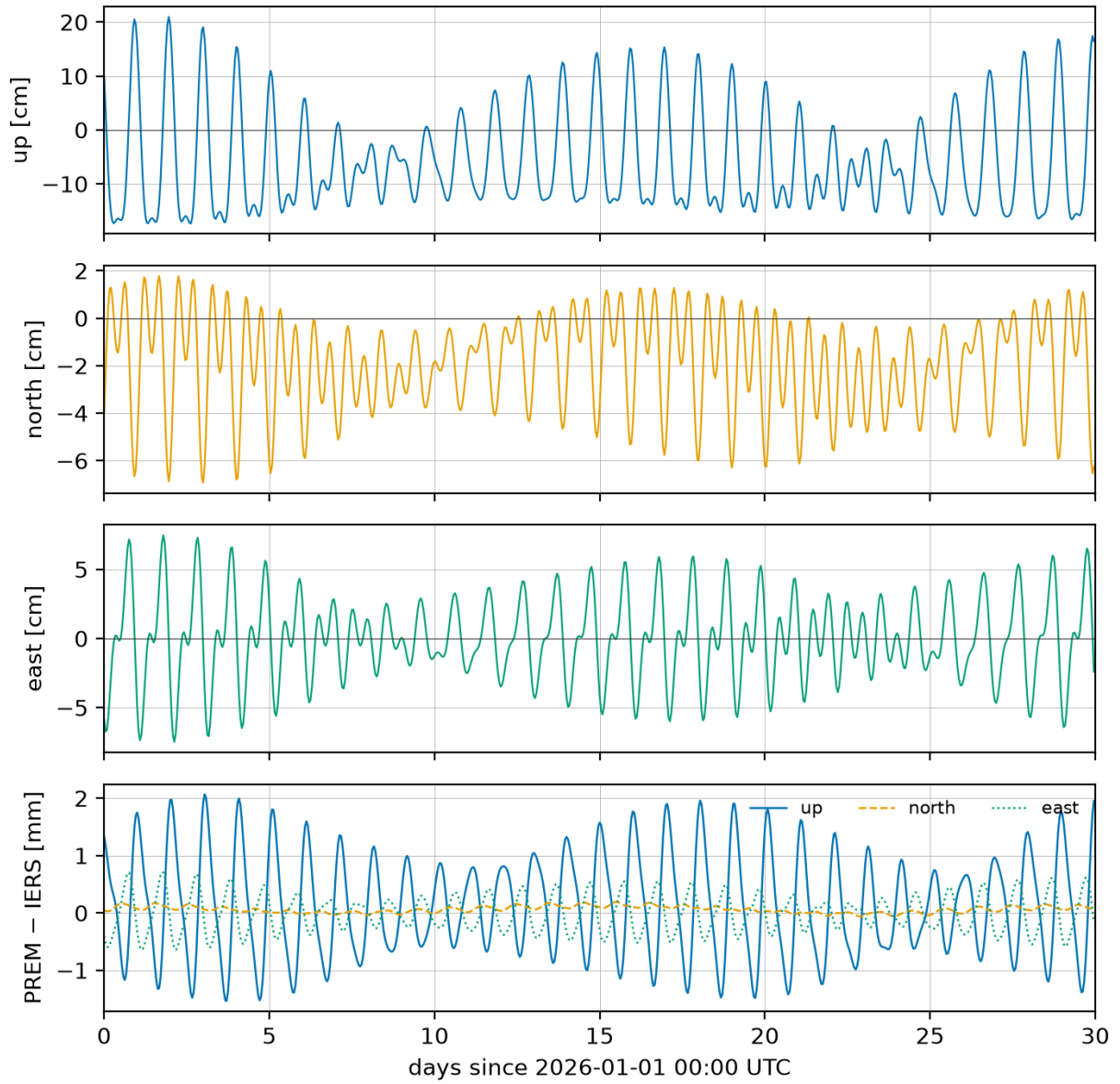


Figure 10: The body tide at Onsala, 30 days from 1 January 2026, hourly, computed by `etide` with the IERS Love numbers (`docs/manual/make_figures.py`). Upper three panels: up, north and east. Bottom: the difference made by the Love numbers of anelastic PREM (`--love prem`).

when it crosses the equator (around days 9 and 23) the diurnal waves vanish and a smaller semidiurnal signal remains. North and east are smaller, ± 7 cm. The averages are not zero: the series includes the permanent tide, about -7 cm up and -2 cm north at this latitude, because the IERS Conventions define station positions as “conventional tide free” and leave the permanent part in the tidal displacement.

The bottom panel shows what the Earth model changes. With the Love numbers of anelastic PREM computed by earth-tides,

```
etide 57.395 11.926 --start 2026-01-01 --days 30 --step 3600 --love prem > onsala_prem.txt
```

the up component differs by up to 2.1 mm (0.9 mm rms), east by up to 0.7 mm and north by up to 0.2 mm. The difference varies with a period of about one day, but the period of the free core nutation (457 days for PREM, 430 for the IERS model) is not the cause: at K_1 the values hardly depend on it. The next section traces it to three choices. A table of your own is used in the same way, `--love love.json`, provided it is in the IERS conventions.

5.8 What the choices change

Using the package. Love numbers are not what a geodesist needs; the displacement of a station is. Between the equations and the numbers in a body-tide model lie a few choices that the theory leaves open, and each changes the displacement by tenths of a millimetre to millimetres. They were measured by running `etide` twice, with and without the choice, for four stations over 2026 (`scripts/displacement_budget.py`); Figure 11 resolves them by tidal line and latitude.

choice	radial rms	maximum
absorption band between 1 s and 200 s (PREM with and without)	0.6–0.7 mm	2.2 mm
out-of-phase part of the diurnal resonance (IERS formula against ours)	0.25–0.36 mm	0.6 mm
resonant ocean-loading correction in the IERS Table 7.2	0.3–0.4 mm	0.7 mm
its in-phase sign (IERS empirical admittance against the ocean models)	0.4–0.55 mm	0.8 mm
the IERS model evaluated at geodetic instead of geocentric latitude	0.34–0.41 mm	1.2 mm
all together: PREM against IERS	0.8–0.9 mm	2.6 mm

The diurnal rows vanish at the equator and are largest at 45° latitude.

- **The absorption band.** PREM gives the elastic moduli at a period of 1 s. Between 1 s and 200 s its quality factor is constant and the moduli soften logarithmically; beyond 200 s they

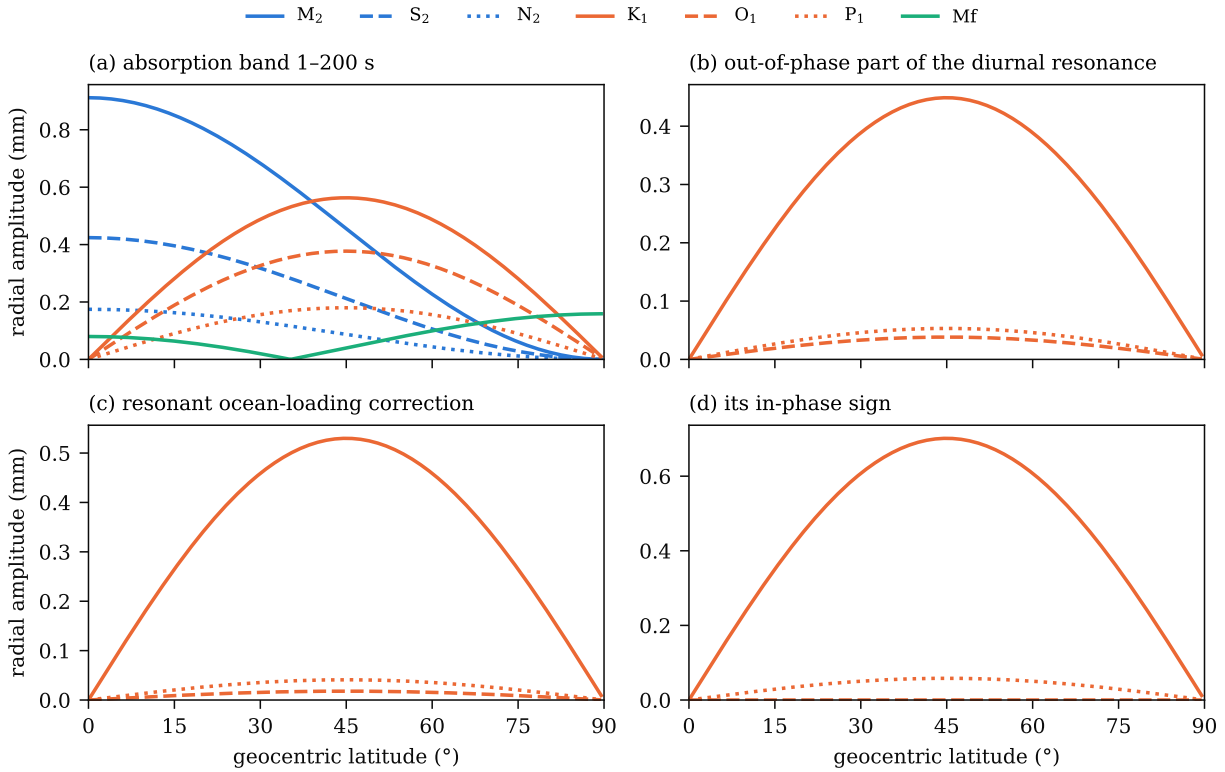


Figure 11: Radial amplitude of each choice for the main tidal lines, against latitude (`paper/make_figures.py`). (a) The absorption band between 1 s and 200 s: PREM with and without it. (b) The out-of-phase part of the diurnal resonance formula of the IERS Conventions, against ours. (c) The resonant ocean-loading correction of the IERS Table 7.2. (d) The difference between the in-phase signs of the empirical admittance and of the ocean models. Colour gives the band, line style the constituent.

follow the power law. earth-tides includes the band; the IERS values leave it out (the moduli are taken at 200 s). Because it scales the whole body tide, it is the largest choice, at all latitudes, and M_2 carries most of it at the equator.

- **The out-of-phase part of the resonance.** The resonance formula of the IERS Conventions has complex residues and eigenfrequencies fitted to the nutations (anelasticity and the coupling at the core boundaries). The damping of the free core nutation alone is only 0.1 mm rms.
- **The ocean tide.** The ocean tide loads the Earth at the same frequencies and resonates with the same free mode, but only through its degree-2, order-1 part: for degrees 3 to 5 the load Love numbers of the rotating, flattened Earth differ from the spherical ones by a constant 0.2 per cent, the same at O_1 , K_1 and next to the free core nutation. Ocean tide loading at K_1 can therefore be computed with the usual Green's functions of a spherical Earth, as long as the degree-2, order-1 resonance is added once. The IERS Conventions put it in the body tide (Table 7.2); `etide --love iers` contains it, `--love mhb` and `--love prem` do not. Do not add it a second time if your loading software already uses frequency-dependent load Love numbers. The in-phase part of the IERS correction has the opposite sign to that implied by the ocean models themselves; tidal gravity observations support the sign of the ocean models.
- **Latitude.** The IERS basis functions use geocentric latitude; `etide` converts the geodetic latitude of a station before evaluating them. Software that forgets the conversion makes an error as large as the physical choices above.

6 The programs

Using the package (reference). The options of the two programs; their use is shown in the last two sections of the Tutorial.

6.1 earthtides-love

`earthtides-love` computes a Love-number table (Section [Love-number table](#)). Without options: anelastic PREM, rotating and flattened, in the IERS conventions, one solution per wave group of the default set, all processor cores, output file name built from the options.

option	meaning
<code>--groups default eterna eterna-split FILE</code>	one Love number per wave group (default: 21 groups plus HIGH, < 0.03 mm from per wave)
<code>--all-waves</code>	a Love number for every wave of the Tamura catalogue
<code>--freq F [F ...] --order 0 1 2</code>	a list of frequencies (cycles per day) for one order instead of the catalogue
<code>--model NAME FILE</code>	Earth model: PREM (default), PREMs, HOMOG, or a model file
<code>--elastic, --no-rotation, --sphere, --no-centrifugal</code>	switch off anelasticity, rotation, flattening, centrifugal reference gravity

option	meaning
--dispersion cot astyanax	anelastic dispersion beyond 200 s: $\cot(\alpha\pi/2)$ (default, as the IERS) or $2/(\pi\alpha)$ (astyanax); see the Appendix
--frame mantle surface	rigid rotation removed (default mantle, IERS)
--latitude geocentric geodetic	latitude of the basis functions (default geocentric, IERS)
--direction radial normal	direction of the components (default radial, IERS)
--surface deformed mean	surface on which the displacement is fitted (default deformed, IERS)
--norm equatorial mean	radius of the normalization of the potential (default equatorial, IERS)
-K INT, -N INT	degrees per field (default 4) and Chebyshev polynomials per layer (default 24)
--jobs INT	parallel worker processes (default: all cores)
-o FILE	output file (default: built from the options)

6.2 etide

`etide` computes the body-tide displacement of one station. **Scope:** the solid Earth body tide of degrees 2 to 4 from the 1200 waves of Tamura (1987), including the permanent tide (as the IERS Conventions require for “conventional tide free” positions; at mid-latitudes it is several centimetres, Section A station time series). Not included: ocean tide loading, atmospheric tides, the pole tide, and gravity, tilt or strain. With the default Love numbers `etide` reproduces the IERS routine DEHANTTIDEINEL (Petit and Luzum 2010, Sect. 7.1.1) to 0.1 mm rms and can replace it.

```
usage: etide [-h] [--start START] [--days DAYS] [--step STEP] [--love LOVE]
           lat lon [h]
```

positional arguments:

```
lat          geodetic latitude [deg]
lon          east longitude [deg]
h            ellipsoidal height [m] (default 0; enters only the geocentric
           latitude)
```

options:

```
--start START  UTC start, YYYY-MM-DD[THH:MM] (default 2026-01-01)
--days DAYS   length [days] (default 1)
--step STEP    sampling [s] (default 3600)
--love LOVE    iers (default), mhb, ddw, prem, prem-allwaves, or a JSON
           table of earthtides-love
```

Input. The latitude is geodetic (GRS80), as given by GNSS; `etide` converts it to the geocentric latitude of the IERS functions. The height changes the result by less than a micrometre (the potential is evaluated at the equatorial radius, as in the IERS model). Times are UTC; they are converted to terrestrial time with the table of leap seconds for the astronomical arguments.

Love numbers (`--love`): `iers`, the IERS Conventions 2010 (resonance formula and Table 7.2 in the diurnal band, the anelastic formula for the zonal waves); `mhb`, the resonance formula alone (Mathews et al. 1995); `ddw`, diurnal values of Dehant et al. (1999); `prem` and `prem-allwaves`, anelastic PREM computed by earth-tides (per wave group and per wave); or a table written by `earthtides-love`.

Output to the standard output: a header line and one line per epoch with the modified Julian date (UTC) and the displacement up, north and east in metres; “up” is along the geocentric radius, north and east perpendicular to it.

Errors. `etide` stops with a message if the `--love` table is not an `earthtides-love` table, is not in the IERS conventions, or lacks a wave of the Tamura catalogue.

7 API reference

Using the package (reference). The command-line programs cover the common tasks. The modules can also be used from Python; this chapter lists the public classes and functions of the first release. Units are SI unless stated otherwise (frequencies of tidal waves in cycles per day).

7.1 earthtides.model

`EarthModel(filename)` reads a layered Earth model in the PREM file format (Chapter [File formats](#)). Each layer stores ρ , V_p , V_s (horizontal and vertical) and η as Chebyshev series in radius; transversely isotropic parameters are replaced by their Voigt isotropic equivalents. The radius in the file is the mean radius s of a level surface when the model is used with a flattened geometry. The package data contain `PREM.dat` (true fluid outer core, ocean merged into the upper crust), `PREMs.dat` (weak-solid outer core, for static runs only) and `HOMOG.dat`.

attribute	meaning
<code>n_layers, r0[i], r1[i]</code>	number of layers, bottom and top radius of layer i (m)
<code>rho[i], lam[i], mu[i]</code>	density and Lamé parameters of layer i : polynomials in r (call them: <code>md.mu[i](r)</code>)
<code>a, M, g_surface</code>	radius, mass and surface gravity of the model
method	returns
<code>gravity(i, r)</code>	$g(r)$ in layer i
<code>mass(i, r)</code>	mass inside radius r
<code>anelastic(omega, alpha=0.15, Tm=200, T0=1, dispersion='cot')</code>	a copy with complex, frequency-dependent $\mu(\omega)$ (absorption band; Appendix), bulk modulus unchanged; <code>dispersion='astyanax'</code> for the form of <code>astyanax</code>
<code>dmu_factor(omega, ...)</code>	the complex $\delta\mu/\mu$ per layer

7.2 earthtides.geometry

The geometry defines the level surfaces $r = R(s, \theta) = s[1 - \frac{2}{3}\epsilon(s)P_2(\cos \theta)]$.

class	level surfaces
<code>SphereGeometry()</code>	spheres, $\epsilon = 0$ (the default of <code>TidalSolver</code>)
<code>ClairautGeometry(model, omega=OMEGA_EARTH)</code>	hydrostatic flattening of the model from Clairaut's equation; attribute <code>flattening_surface</code>
<code>ExactSpheroidGeometry(flattening)</code>	exact similar spheroids (homogeneous test bodies, ex5)
<code>BumpGeometry(model, rel_amp=0.2)</code>	an artificial distortion that vanishes at the interfaces, for the distorted-mesh test

A geometry object provides `eps(i, s)` (returning $\epsilon, \epsilon', \epsilon''$); the functions `mapping`, `mapping_from_R` and `chain` turn these into the derivatives of the map and of its inverse (Chapter [How the solver works](#)). `OMEGA_EARTH = 7.292115e-5` rad/s.

7.3 earthtides.solver

`TidalSolver(model, geometry=None, m=0, K=3, N=24, n_force=2, ...)` sets up and solves the problem for one order m and one forcing degree n_f . Parameters, by purpose:

Resolution

parameter	default	meaning
<code>K</code>	3	spheroidal degrees per field ($n_f, n_f \pm 2, \dots$)
<code>KT</code>	K	toroidal degrees
<code>N</code>	24	Chebyshev polynomials per layer
<code>n_theta</code>	$2n_{\max} + 24$	Gauss–Legendre points of the projection
<code>degrees</code>	–	explicit lists (<code>[spheroidal]</code> , <code>[toroidal]</code>) instead of <code>K</code> , <code>KT</code>
<code>deg_window</code>	–	only $n_f \pm 2w$ (high-degree loads)

Forcing

parameter	default	meaning
<code>n_force, m</code>	2, 0	degree and order of the forcing

parameter	default	meaning
forcing	'tide'	'tide': tidal potential $-ga(r/a)^{n_f}\bar{Y}_{n_f}^m \cos m\lambda$; 'load': unit surface mass density $\bar{Y}_{n_f}^m \cos m\lambda$ (kg m ⁻²); 'none': matrix only

Dynamics and reference state

parameter	default	meaning
omega_f	1.4053e-4	frequency (rad/s) of the inertial term $\rho\omega^2\mathbf{u}$ in a non-rotating calculation
coriolis, omega_t	0, 0	rotation rate (rad/s) for the Coriolis force and the signed tidal frequency (rad/s) of the inertia in the rotating frame; set both for rotating runs (coriolis=OMEGA_EARTH, omega_f=0)
dyn_fluid	False	rotating runs: inertia and Coriolis force (omega_t, coriolis) also in the liquid layers, needed for a true fluid core; omega_f puts inertia in every layer regardless
omega_rot	0	rotation rate of the reference state: centrifugal part of $\nabla\Phi_0$
consistent_reference	False	Φ_0 from a Poisson solve for the model's own density plus the exact centrifugal potential (needed in the diurnal band)
gravity	True	False: no self-gravitation, no pre-stress (pure elasticity)

Frame and rotations

parameter	default	meaning
frame	'surface'	rigid rotation removed from the output: 'mantle' (Tisserand rotation of the mantle, IERS) or 'surface' (best-fit rotation of the surface). Note: the default of <code>TidalSolver</code> is 'surface', that of <code>love.Options</code> and <code>earthtides-love</code> is 'mantle' (IERS); pass <code>frame='mantle'</code> for IERS values
wahr_frame	False	regions rotate freely (with Coriolis); the rigid rotation is removed afterwards, in the chosen frame (use for $m = 0, 1$ in rotating runs)
tisserand, regions	automatic	Tisserand constraints of static runs: one per region of solid or liquid layers

Linear algebra

parameter	default	meaning
linear_solver	'sparse'	'sparse' (SuperLU) or 'dense' (LAPACK)
centre	'fixed'	spheroidal degree 1 at the centre: 'fixed' pins it ($U_1 = V_1 = 0$; static runs, body tides) or 'free' lets it move ($U_1 = V_1$, $dU_1/ds = 0$; loads with inertia, needs <code>omega_f</code> or <code>omega_t</code>); see Section Degree 1

Experimental (a laterally heterogeneous Earth; not part of this release): `lateral`, `all_degrees`, `fixed_density_layers`, `eulerian_interfaces`, `background`.

method	returns
<code>solve()</code>	solves and returns <code>results()</code>

method	returns
<code>results()</code>	dict: U, V, W (surface values divided by a , per degree), k (per degree); h, l, k_k for the forcing degree; α (rigid-rotation forces of static runs)
<code>load_love()</code>	for <code>forcing='load'</code> : dict of dicts h, l, k, W per degree, normalized by the load potential $4\pi G a / (2n_f + 1)$
<code>surface_field(theta, remove_rigid=None)</code>	amplitudes u_r, u_θ (times $\cos m\lambda$) and u_λ (times $\sin m\lambda$) on the flattened surface at colatitudes <code>theta</code>
<code>surface_coefficients()</code>	surface values of every field and the exterior coefficients c_n

On a sphere $h = U_{n_f}(a)/a$, $l = V_{n_f}(a)/a$ and $k = -c_{n_f}/(ga)$ are the ordinary Love numbers. The basis functions are in `earthtides.basis`: `cheb_basis(x, N)` and `ybar(n, m, theta)` (also importable from `earthtides.solver`); the `jet` class `J` is in `earthtides.jets`.

7.4 earthtides.iers

function	returns
<code>iers_fit(S, geom)</code>	the IERS parameters h_0, h_2, l_0, l_1, l_2 (and h_p or l_p) of a solved <code>TidalSolver</code> , normalized at the equatorial radius, with the misfit
<code>iers_fit_fields(m, th, ur, ut, up, f_geodetic=0.0)</code>	the same for given surface fields; <code>f_geodetic</code> evaluates the basis at geodetic latitude

7.5 earthtides.love

object	purpose
<code>Options(model='PREM', anelastic=True, rotation=True, flattening=True, centrifugal=True, frame='mantle', latitude='geocentric', direction='radial', surface='deformed', norm='equatorial', K=4, N=24, ...)</code>	physics, numerics and conventions of a table; <code>is_iers()</code> tells whether the conventions are the IERS ones
<code>solve_wave(opts, m, f_cpd)</code>	solves one degree-2 tide; returns the solver and its results
<code>parameters(S, r, opts)</code>	the IERS-type parameters of a solution in the conventions of <code>opts</code>
<code>love_table(opts, waves, jobs=1)</code>	a table for <code>waves = list of (n, doodson or None, f_cpd, m)</code>

object	purpose
<code>group_table(opts, groups, jobs=1)</code>	a table with one Love number per wave group (groups = list of (name, f1, f2); read_groups(path) reads a file, DEFAULT_GROUPS is the default set)
<code>write(table, path), read(path)</code>	JSON output and input (complex numbers restored)
<code>main(argv)</code>	the command earthtides-love

```

from earthtides import love
opts = love.Options(anelastic=False)           # elastic PREM, IERS conventions
S, r = love.solve_wave(opts, 2, 1.9322736)      # M2, frequency in cycles per day
print(love.parameters(S, r, opts))             # h0 = 0.60243, h2 = -0.00049, ...

```

7.6 earthtides.etide

object	purpose
<code>Tamura()</code>	the 1200 waves: n, m, dood (Doodson multipliers), phase, A, dA, f_cpd
<code>LoveModel(cat, source='iers')</code>	Love numbers per wave from 'iers', 'mhb', 'ddw', 'prem', 'prem-allwaves' or a JSON table
<code>displacement(mjd, lat, lon, h=0.0, love='iers', degrees=(2, 3, 4))</code>	up, north, east displacement (m) at UTC MJD (array) for a station at geodetic latitude and east longitude (degrees)
<code>angles(mjd, lon)</code>	Tamura's astronomical angles
<code>main()</code>	the command etide

```

import numpy as np
from earthtides import etide
mjd = 61041.0 + np.arange(0, 2, 1 / 24)        # 2026-01-01, two days hourly
up, north, east = etide.displacement(mjd, 57.395, 11.926, love='prem')

```

7.7 Recipes

Body-tide Love numbers of spherical PREM at M_2

```

from earthtides.model import EarthModel
from earthtides.solver import TidalSolver
from earthtides.love import model_path
md = EarthModel(model_path('PREM'))           # the package's PREM (or any model file)
r = TidalSolver(md, m=2, N=24, n_force=2, omega_f=1.4053e-4).solve()
print(r['h'], r['l'], r['kk'])

```

Load Love numbers of degree 10

```
S = TidalSolver(md, m=10, K=1, N=24, n_force=10, omega_f=1.4053e-4, forcing='load')
S.solve()
L = S.load_love()                                # L['h'][10], L['l'][10], L['k'][10]
```

The diurnal band on a flattened, rotating Earth (fluid core, mantle frame)

```
from earthtides.geometry import ClairautGeometry, OMEGA_EARTH
from earthtides.iers import iers_fit
g = ClairautGeometry(md)
w = 0.92700 * OMEGA_EARTH                        # 01
S = TidalSolver(md, g, m=1, K=4, N=24, omega_f=0.0, coriolis=OMEGA_EARTH, omega_t=w,
                omega_rot=OMEGA_EARTH, wahr_frame=True, dyn_fluid=True,
                consistent_reference=True, frame='mantle')

S.solve()
print(iers_fit(S, g))
```

8 File formats

Using the package (reference). earth-tides reads and writes three kinds of files: the Earth model, which the solver and earthtides-love read; the wave groups, which earthtides-love reads; and the Love-number table, which earthtides-love writes and etide reads. Each is described below with a complete example.

8.1 Earth model

An Earth model is a text file in the format of astyanax, the author's program that integrates the radial equations of a spherical Earth (used for the Green's functions of [Bos and Scherneck 2013](#)), the format of the models in the package (PREM, PREMs, HOMOG; earthtides.love.model_path gives their location).

Layout. The first line holds the name of the model, the second the number of layers L ; text after a | is a comment. Then follow seven blocks, in any order, each consisting of a line with its name, a line of dashes and L rows, one per layer, starting with the innermost layer:

block	quantity	unit
rho	density	g/cm ³
Vp_h, Vp_v	P-wave velocity, horizontally and vertically polarized	km/s
Vs_h, Vs_v	S-wave velocity, horizontally and vertically polarized	km/s
eta	anisotropy parameter η	—
Qmu	shear quality factor (nan in a liquid)	—

The first number of every row is the depth of the *bottom* of the layer in km; the layer reaches up to the bottom of the next layer, the last layer to the surface. The innermost layer therefore has the

radius of the Earth as its depth: it reaches the centre. The radius a is that first depth. Each row of the first six blocks then gives four Chebyshev coefficients $c_0, \dots, c_3$,

$$f(r) = \sum_{k=0}^3 c_k T_k(x), \quad x = \frac{2r - r_0 - r_1}{r_1 - r_0} \in [-1, 1], \quad (8.1)$$

with r_0 and r_1 the radii of the bottom and the top of the layer; the rows of `Qmu` give one constant per layer. For example, the inner-core density of PREM, $12.96667 - 0.16244 T_1 - 0.04061 T_2$, is 13.0885 g/cm^3 at the centre ($x = -1$) and 12.7636 at the inner-core boundary ($x = 1$), the values of PREM. On a flattened Earth the radius in the file is the mean radius s of a level surface.

What the program makes of it (`EarthModel`). The transversely isotropic parameters are replaced by their Voigt isotropic equivalents, as in `astyanax`: with $A = \rho V_{ph}^2$, $C = \rho V_{pv}^2$, $L = \rho V_{sv}^2$, $N = \rho V_{sh}^2$ and $F = \eta(A - 2L)$,

$$\kappa = \frac{C + 4A - 4N + 4F}{9}, \quad \mu = \frac{C + A + 6L + 5N - 2F}{15}, \quad \lambda = \kappa - \frac{2}{3}\mu. \quad (8.2)$$

Density, λ and μ are then polynomials in each layer, with exact derivatives (Section **Jets and the chain rule in one variable**), and gravity follows from the integrated density. A layer with $V_s = 0$ is a liquid that slides along its boundaries; `PREMs` replaces the liquid core by a weak solid ($V_s = 1 \text{ m/s}$) for static runs. `Qmu` is used by the anelasticity (`EarthModel.anelastic`). The constant of gravitation is $G = 6.673 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$, as in `astyanax`.

Example 1: a homogeneous sphere (`HOMOG`), the smallest complete file:

```
HOMOG | Name of Earth Model
1      | Number of layers

rho
----
6371.0    5.5000000    0.00000    0.00000    0.00000

Vp_h
----
6371.0    7.3854895    0.00000    0.00000    0.00000

Vp_v
----
6371.0    7.3854895    0.00000    0.00000    0.00000

Vs_h
----
6371.0    4.2640143    0.00000    0.00000    0.00000

Vs_v
```

```

----
6371.0    4.2640143    0.00000    0.00000    0.00000

```

```

eta

```

```

---
6371.0    1.0000000    0.00000    0.00000    0.00000

```

```

Qmu

```

```

---
6371.0    600.0

```

Example 2: PREM (_{PREM}), 12 layers: inner core, liquid outer core ($V_s = 0$, $Q_{\mu} = \text{nan}$), eight mantle layers and two crustal layers. The 3 km ocean of PREM is merged into the upper crust (the last layer, from 15 km depth to the surface), with a density that preserves the mass of the Earth:

```

PREM      | Name of Earth Model
12         | Number of layers

```

```

rho

```

```

---
6371.0    12.96667    -0.16244    -0.04061    0.00000
5149.5    11.18823    -1.12375    -0.15334    -0.00770
2891.0     5.52894    -0.03749     0.00003    -0.00000
2741.0     4.98128    -0.52129    -0.01394    -0.00285
 771.0     4.41206    -0.03123    -0.00008    -0.00000
 670.0     3.98397    -0.00815     0.00000     0.00000
 600.0     3.84978    -0.12604     0.00000     0.00000
 400.0     3.48952    -0.05374     0.00000     0.00000
 220.0     3.36710     0.00761     0.00000     0.00000
  80.0     3.37773     0.00302     0.00000     0.00000
  24.4     2.90000     0.00000     0.00000     0.00000
  15.0     2.28400     0.00000     0.00000     0.00000

```

```

Vp_h

```

```

----
6371.0    11.17447    -0.11697    -0.02924     0.00000
5149.5     9.37083    -1.12657    -0.16058    -0.01890
2891.0    13.69845    -0.01809     0.00009    -0.00000
2741.0    12.44965    -1.28280    -0.07665    -0.02461
 771.0    10.90850    -0.15714    -0.00004    -0.00000
 670.0    10.21197    -0.05421     0.00000     0.00000
 600.0     9.64587    -0.51195     0.00000     0.00000
 400.0     8.73210    -0.17315     0.00000     0.00000
 220.0     8.03298     0.04327     0.00000     0.00000
  80.0     8.09343     0.01718     0.00000     0.00000
  24.4     6.80000     0.00000     0.00000     0.00000
  15.0     6.18820     0.00000     0.00000     0.00000

```

Vp_v

6371.0	11.17447	-0.11697	-0.02924	0.00000
5149.5	9.37083	-1.12657	-0.16058	-0.01890
2891.0	13.69845	-0.01809	0.00009	-0.00000
2741.0	12.44965	-1.28280	-0.07665	-0.02461
771.0	10.90850	-0.15714	-0.00004	-0.00000
670.0	10.21197	-0.05421	0.00000	0.00000
600.0	9.64587	-0.51195	0.00000	0.00000
400.0	8.73210	-0.17315	0.00000	0.00000
220.0	8.03298	0.04327	0.00000	0.00000
80.0	8.09343	0.01718	0.00000	0.00000
24.4	6.80000	0.00000	0.00000	0.00000
15.0	6.18820	0.00000	0.00000	0.00000

Vs_h

6371.0	3.60649	-0.08174	-0.02044	0.00000
5149.5	0.00000	0.00000	0.00000	0.00000
2891.0	7.26540	0.00066	-0.00003	0.00000
2741.0	6.78548	-0.50419	-0.03232	-0.00857
771.0	6.09281	-0.14770	0.00002	0.00000
670.0	5.54311	-0.02710	0.00000	0.00000
600.0	5.22421	-0.29172	0.00000	0.00000
400.0	4.70690	-0.06300	0.00000	0.00000
220.0	4.44472	0.02580	0.00000	0.00000
80.0	4.48076	0.01025	0.00000	0.00000
24.4	3.90000	0.00000	0.00000	0.00000
15.0	3.41420	0.00000	0.00000	0.00000

Vs_v

6371.0	3.60649	-0.08174	-0.02044	0.00000
5149.5	0.00000	0.00000	0.00000	0.00000
2891.0	7.26540	0.00066	-0.00003	0.00000
2741.0	6.78548	-0.50419	-0.03232	-0.00857
771.0	6.09281	-0.14770	0.00002	0.00000
670.0	5.54311	-0.02710	0.00000	0.00000
600.0	5.22421	-0.29172	0.00000	0.00000
400.0	4.70690	-0.06300	0.00000	0.00000
220.0	4.44472	0.02580	0.00000	0.00000
80.0	4.48076	0.01025	0.00000	0.00000
24.4	3.90000	0.00000	0.00000	0.00000
15.0	3.41420	0.00000	0.00000	0.00000

eta

6371.0	1.00000	0.00000	0.00000	0.00000
5149.5	1.00000	0.00000	0.00000	0.00000
2891.0	1.00000	0.00000	0.00000	0.00000
2741.0	1.00000	0.00000	0.00000	0.00000
771.0	1.00000	0.00000	0.00000	0.00000
670.0	1.00000	0.00000	0.00000	0.00000
600.0	1.00000	0.00000	0.00000	0.00000
400.0	1.00000	0.00000	0.00000	0.00000
220.0	1.00000	0.00000	0.00000	0.00000
80.0	1.00000	0.00000	0.00000	0.00000
24.4	1.00000	0.00000	0.00000	0.00000
15.0	1.00000	0.00000	0.00000	0.00000

Qmu

6371.0	86.4
5149.5	nan
2891.0	312.0
2741.0	312.0
771.0	312.0
670.0	143.0
600.0	143.0
400.0	143.0
220.0	80.0
80.0	600.0
24.4	600.0
15.0	600.0

8.2 Wave groups

A wave-group file gives the frequency limits of the groups for `earthtides-love --groups FILE`. Each line holds one group, in one of two formats:

- `f1 f2 name`, the limits in cycles per day and a name;
- `WAVEGROUPI= f1 f2 gain phase name`, the format of the tidal analysis and prediction programs ETERNA and PREDICT ([Wenzel 1996](#)); gain and phase are read but not used.

Text after `#` is a comment. Every degree-2 wave of the Tamura catalogue with a non-zero frequency must lie in exactly one group (limits included, tolerance 10^{-9} cycles per day), otherwise `earthtides-love` stops with an error. The Love numbers of a group are computed once, at the frequency of its largest degree-2 wave, and given to every degree-2 wave of the group. Degrees 3 and 4 are always spherical values at the M_2 frequency, and the permanent tide gets the nominal IERS values; a group without degree-2 waves, such as the last group of the examples, therefore costs nothing.

Example 1: the default groups (`data/groups_default.txt`, used when `--groups` is not given). The limits were designed so that the displacement differs from the per-wave result by less than 0.03 mm (`tools/design_groups.py`); the comment gives the frequency of the main wave and the

number of waves:

```
# Default wave groups of earthtides-love (tools/design_groups.py): limits in cycles
# per day; designed from the per-wave admittance of PREM (earth-tides) and of the IERS
# model, sum of the error bounds 0.103 mm (budget 0.15 mm); main wave in
# the comment. Columns: f1 f2 name
0.000000000 0.002664232 L01      # main wave 0.000147 cpd, 7 waves
0.002664233 0.005475688 L02      # main wave 0.002738 cpd, 9 waves
0.005475689 0.031361190 L03      # main wave 0.005476 cpd, 18 waves
0.031361191 0.067652837 L04      # main wave 0.036292 cpd, 45 waves
0.067652838 0.109257942 L05      # main wave 0.073202 cpd, 56 waves
0.109257943 0.501369000 L06      # main wave 0.109494 cpd, 75 waves
0.501370000 0.929315066 D01      # main wave 0.893244 cpd, 123 waves
0.929315067 0.966372715 D02      # main wave 0.929536 cpd, 46 waves
0.966372716 0.994450765 D03      # main wave 0.966446 cpd, 21 waves
0.994450766 0.997188544 D04      # main wave 0.994524 cpd, 5 waves
0.997188545 1.000000000 D05      # main wave 0.997262 cpd, 7 waves
1.000000001 1.002517269 D06      # main wave 1.000000 cpd, 5 waves
1.002517270 1.002664363 D07      # main wave 1.002591 cpd, 1 waves
1.002664364 1.002811587 D08      # main wave 1.002738 cpd, 2 waves
1.002811588 1.005402141 D09      # main wave 1.002885 cpd, 5 waves
1.005402142 1.006608801 D10      # main wave 1.005476 cpd, 3 waves
1.006608802 1.033333097 D11      # main wave 1.008214 cpd, 17 waves
1.033333098 1.075704205 D12      # main wave 1.039030 cpd, 45 waves
1.075704206 1.470243000 D13      # main wave 1.075940 cpd, 65 waves
1.470244000 1.999926453 S01      # main wave 1.932274 cpd, 187 waves
1.999926454 2.451943000 S02      # main wave 2.000000 cpd, 94 waves
2.451944000 7.000000000 HIGH     # degrees 3 and 4 only (spherical values), no solve
```

Example 2: the ETERNA groups in ETERNA format (the built-in set `--groups eterna`). Their single K_1 group cannot follow the resonance of the free core nutation, and the displacements are off by up to 3.4 mm; `--groups eterna-split` splits it into P_1 , S_1 , K_1 , ψ_1 and ϕ_1 :

```
# ETERNA-style wave groups: WAVEGROUPI= f1 f2 gain phase name (f in cycles per day;
# gain and phase are read by ETERNA but ignored by earthtides-love)
WAVEGROUPI= 0.000000 0.501369 1.0000 0.0000 long
WAVEGROUPI= 0.501370 0.911390 1.0000 0.0000 Q1
WAVEGROUPI= 0.911391 0.947991 1.0000 0.0000 O1
WAVEGROUPI= 0.947992 0.981854 1.0000 0.0000 M1
WAVEGROUPI= 0.981855 1.023622 1.0000 0.0000 K1
WAVEGROUPI= 1.023623 1.057485 1.0000 0.0000 J1
WAVEGROUPI= 1.057486 1.470243 1.0000 0.0000 OO1
WAVEGROUPI= 1.470244 1.880264 1.0000 0.0000 2N2
WAVEGROUPI= 1.880265 1.914128 1.0000 0.0000 N2
WAVEGROUPI= 1.914129 1.950419 1.0000 0.0000 M2
WAVEGROUPI= 1.950420 1.984282 1.0000 0.0000 L2
WAVEGROUPI= 1.984283 2.451943 1.0000 0.0000 S2
WAVEGROUPI= 2.451944 7.000000 1.0000 0.0000 M3M6
```

8.3 Love-number table

earthtides-love writes a JSON file with four members:

- `format ("earthtides-love")` and `format_version (1)`;
- `metadata`: the package and its version, the creation time, the Earth-model file and its SHA-256 checksum (so that a table can be traced to its model), every option (`options`), the five conventions (`conventions`), whether they are the IERS ones (`iers_conventions`), the units, the run time and, for a grouped table, the groups with the frequency at which each was computed (`groups`, `representative_f_cpd`);
- `entries`, one per wave.

Each entry holds the degree n , the order m and the frequency f_{cpd} in cycles per day; for the waves of the catalogue the eight Doodson multipliers of Tamura's arguments `doodson` (lunar time, s , h , p , N' , p_s and two planetary arguments) and, in a grouped table, the `group`; then the Love-number parameters as complex numbers [`re`, `im`] (a negative imaginary part is a lag, Section [Units and signs](#)): `h0`, `h2`, `l0`, `l2`, `l1`, `lp` (zonal and diurnal), `hp` (zonal) and `k`; the relative `misfit` of the fit of the parametrization to the computed surface field; and a `note` for the special entries (permanent tide, spherical degrees 3 and 4).

`etide` matches the entries to the 1200 waves of its catalogue by degree and Doodson multipliers, uses `h0`, `h2`, `l0`, `l1` and `l2`, and refuses a table that lacks a wave or is not in the IERS conventions. The other parameters are there for comparison with other models.

Example: a complete table for one wave, written by `earthtides-love --freq 1.9322736 --order 2 -o m2.json` (M_2 , anelastic PREM, IERS conventions). A table made with `--freq` has no Doodson multipliers and cannot be used by `etide`; it is meant for studies of the Love numbers themselves:

```
{
  "format": "earthtides-love",
  "format_version": 1,
  "metadata": {
    "package": "earth-tides",
    "version": "0.1.0",
    "created": "2026-10-09T12:15:37+00:00",
    "model_file": "PREM.dat",
    "model_sha256": "64aeca57b67613e104653369bb314832f16810b514fd33034b1817d930a05a43",
    "options": {
      "model": "PREM",
      "anelastic": true,
      "alpha": 0.15,
      "tm": 200.0,
      "dispersion": "cot",
      "rotation": true,
      "flattening": true,
      "centrifugal": true,
      "frame": "mantle",
      "latitude": "geocentric",
      "direction": "radial",

```

```

    "surface": "deformed",
    "norm": "equatorial",
    "K": 4,
    "N": 24,
    "f_dyn": 0.005
  },
  "conventions": {
    "latitude": "geocentric",
    "direction": "radial",
    "surface": "deformed",
    "norm": "equatorial",
    "frame": "mantle"
  },
  "iers_conventions": true,
  "units": {
    "f_cpd": "cycles per day",
    "parameters": "complex [re, im]; h, l: equatorial radius; k: mean radius"
  },
  "runtime_s": 3.1
},
"entries": [
  {
    "n": 2,
    "m": 2,
    "f_cpd": 1.9322736,
    "h0": [
      0.61180310151037,
      -0.0025090160616752072
    ],
    "h2": [
      -0.0005190546985092717,
      6.752713392901831e-06
    ],
    "l0": [
      0.08609346565336735,
      -0.0006725416694198551
    ],
    "l1": [
      0.0010292205627414918,
      -1.1157135245832396e-05
    ],
    "l2": [
      8.241653617794046e-05,
      3.2187059541580805e-06
    ],
    "k": [
      0.3049508812574024,

```

```

    -0.0014383081781731723
  ],
  "misfit": 9.817256713540391e-07
}
]
}

```

Entries of a grouped table (data/love_PREM.json, 1200 entries): a zonal wave of group L03 (with h_p and l_p), O_1 in group D02, and a degree-3 wave with its spherical values:

```

{"n": 2, "m": 0, "f_cpd": 0.03052164085611073, "doodson": [0, 1, -2, -1, -2, 0, 0, 0],
 "group": "L03",
 "h0": [0.6238596971195662, -0.006370329720727849],
 "h2": [-0.000558796712766559, 1.8471046356953283e-05],
 "l0": [0.08963994867290359, -0.0017924446043052237],
 "l1": [1.065783992836489e-06, -3.965367475513343e-08],
 "l2": [5.8768378527084416e-05, 1.0185466976635981e-05],
 "hp": [0.00010570437116038353, -4.116926302720287e-06],
 "lp": [3.4136473261218824e-07, -1.2329736472391264e-08],
 "k": [0.3106761982297467, -0.003650931017359199],
 "misfit": 1.572431067822994e-06}

{"n": 2, "m": 1, "f_cpd": 0.9295357067958021, "doodson": [1, -1, 0, 0, 0, 0, 0, 0],
 "group": "D02",
 "h0": [0.6067782796194422, -0.002797818879047416],
 "h2": [-0.0005286322865246587, 7.505192518165928e-06],
 "l0": [0.08584488499782782, -0.000754061798316542],
 "l1": [0.0004978038287070359, -6.0743793311026954e-06],
 "l2": [8.12331261211025e-05, 3.6548752356265706e-06],
 "lp": [-0.0003038867216646596, 2.2403582059133455e-06],
 "k": [0.3014752331414812, -0.0015968769129258511],
 "misfit": 3.556230843637756e-06}

{"n": 3, "m": 0, "f_cpd": 0.0001623604476386037, "doodson": [0, 0, 0, 1, -1, 0, 0, 0],
 "h0": [0.2939514510175192, -0.001486557245604307],
 "l0": [0.01565143353614132, -0.0002713086953384189],
 "k": [0.09414739901300151, -0.0005346961358152061],
 "note": "spherical, at the M2 frequency"}

```

The `metadata` of a grouped table is that of the example above plus the list `groups`, for instance

```

{"name": "D02", "f1": 0.929315067, "f2": 0.966372715, "representative_f_cpd":
0.9295357067958021}.

```

9 Appendix: formulas of the supporting modules

Reference for reproducing the code. The solver chapter describes the core; this appendix gives the formulas of the modules that it only names, each as implemented.

9.1 Hydrostatic flattening (Clairaut)

`ClairautGeometry` integrates Clairaut's equation for the flattening $\epsilon(s)$ of the level surfaces, to first order in ϵ ,

$$\epsilon'' + \frac{6\rho_0}{\bar{\rho}s} \epsilon' + \frac{6}{s^2} \left(\frac{\rho_0}{\bar{\rho}} - 1 \right) \epsilon = 0, \quad \bar{\rho}(s) = \frac{3M(s)}{4\pi s^3}, \quad (9.1)$$

with $M(s)$ the mass inside s . It starts at $s = 1$ km with $\epsilon = 1$, $\epsilon' = 0$ (the regular solution), integrates outward layer by layer with ϵ and ϵ' continuous (an eighth-order Runge–Kutta method, relative tolerance 10^{-12}), and scales the solution so that at the surface

$$a \epsilon'(a) + 2\epsilon(a) = \frac{5}{2} m, \quad m = \frac{\Omega^2 a^3}{GM}. \quad (9.2)$$

For PREM this gives $1/299.90$ at the surface, $1/393$ at the core–mantle boundary and $1/413$ at the inner-core boundary. The map $r = R(s, \theta)$ is eq. (4.6).

9.2 Anelasticity (absorption band)

`EarthModel.anelastic(omega)` replaces the shear modulus by a complex, frequency-dependent one, $\mu(\omega) = \mu_0 [1 + \delta(\omega)]$, with the bulk modulus κ unchanged ($\lambda = \kappa - \frac{2}{3}\mu$). With Q_μ of the model, the seismic reference frequency $\omega_0 = 2\pi/T_0$ ($T_0 = 1$ s), $\omega_m = 2\pi/T_m$ ($T_m = 200$ s) and $x = (\omega_m/\omega)^\alpha$,

$$\delta(\omega) = \frac{1}{Q_\mu} \left[\frac{2}{\pi} \ln \frac{\omega_m}{\omega} + \cot \frac{\alpha\pi}{2} (1 - x) + ix \right], \quad \alpha = 0.15, \quad (9.3)$$

and for $\alpha \rightarrow 0$ (constant Q) $\delta = [\frac{2}{\pi} \ln(\omega/\omega_0) + i]/Q_\mu$. The factor $\cot(\alpha\pi/2)$ is the exact dispersion of a quality factor that varies as ω^α , the form of Wahr and Bergen (1986) used in the IERS Conventions; astyanax and Bos and Scherneck (2013) use its small- α approximation $2/(\pi\alpha)$, which for $\alpha = 0.15$ is 2 per cent larger and makes $h^{(0)}$ larger by 1×10^{-4} (semidiurnal) to 8×10^{-4} (18.6 years). That form is available as `dispersion='astyanax'` (`EarthModel.anelastic, earthtides-love --dispersion astyanax`). Layers with $Q_\mu = \text{nan}$ (liquids) are unchanged.

9.3 Reference potential consistent with the density

With `consistent_reference=True` (used by `earthtides-love`) the reference potential is $\Phi_0 = \phi_0 + \psi$ with the exact centrifugal potential $\psi = -\frac{1}{2}\Omega^2 r^2 \sin^2 \theta$ and ϕ_0 the solution of

$$\nabla^2 \phi_0 = 4\pi G \rho_0(s) \quad (9.4)$$

on the same mapped grid as the tidal problem (`refpot.ReferencePotential`): axisymmetric, degrees 0, 2, 4, 6, 24 Chebyshev polynomials per layer, ϕ_0 and its normal derivative continuous at

the interfaces, and $\sum_n c_n (a/r)^{n+1} P_n$ outside. The level-surface formula of $\nabla \Phi_0$ (first order in ϵ) would leave a residual force of order ϵ^2 in a rigid tilt of the Earth, which near the tilt-over mode is multiplied by the large nutation amplitude; with the consistent Φ_0 the gravitational terms cancel exactly in a rigid tilt. Without it the free core nutation of Example 5 is 2.5 per cent off.

9.4 Fit of the IERS parameters

`iers.iers_fit_fields` fits the surface displacement, normalized by R_e^4/a^3 (Section [The six choices behind the parameters](#)), at 600 colatitudes by least squares, separately for the radial and for the horizontal components (north = $-u_\theta$, east = u_λ). With φ the (geocentric) latitude, $P_2 = P_2(\sin \varphi)$, $s_\varphi = \sin \varphi$, $c_\varphi = \cos \varphi$ and C the ratio of the code's $\bar{Y}_2^m$ to the IERS function, the model is (Petit and Luzum 2010, Sect. 7.1.1):

- radial, all orders: $u_r = C [h^{(0)} + h^{(2)} P_2] Y_m$ with $Y_0 = P_2$, $Y_1 = 3s_\varphi c_\varphi$, $Y_2 = 3c_\varphi^2$, plus $C \sqrt{4\pi/5} h'$ for $m = 0$;
- $m = 0$: north = $3C s_\varphi c_\varphi [l^{(0)} + l^{(2)} P_2]$, east = $3C c_\varphi s_\varphi^2 l^{(1)} - C \sqrt{4\pi/5} c_\varphi l'$;
- $m = 1$: north = $3C \cos 2\varphi [l^{(0)} + l^{(2)} P_2] - 3C s_\varphi^2 l^{(1)} + C \sqrt{24\pi/5} l'$, east = $-3C s_\varphi [l^{(0)} + l^{(2)} P_2] + 3C s_\varphi \cos 2\varphi l^{(1)} + C \sqrt{24\pi/5} s_\varphi l'$;
- $m = 2$: north = $-6C s_\varphi c_\varphi [l^{(0)} + l^{(2)} P_2 + l^{(1)}]$, east = $-6C c_\varphi [l^{(0)} + l^{(2)} P_2] - 6C s_\varphi^2 c_\varphi l^{(1)}$.

The relative misfit of the fit is stored with every entry of a table (`misfit`); it is 10^{-6} to 10^{-5} .

9.5 Rigid rotations: the Tisserand row and the mantle frame

In a static run with $m = 0$ or 1 a rigid rotation of a region (inner core, outer core, mantle) costs no force and is undetermined. The solver adds to the momentum equation of each region an unknown force $\alpha \rho_0 \mathbf{b}$ with the pattern $\mathbf{b}$ of a rigid rotation, $\mathbf{b} = (0, r, -r \cos \theta)$ (rotation about the y axis, $m = 1$) or $(0, 0, r \sin \theta)$ (about the spin axis, $m = 0$) in (r, θ, λ) amplitudes, and the condition that the mean rotation of the region vanishes,

$$\int \rho_0 r (u_\theta - u_\lambda \cos \theta) dV = 0 \quad (m = 1), \quad \int \rho_0 r \sin \theta u_\lambda dV = 0 \quad (m = 0), \quad (9.5)$$

with $dV = R^2 R_s ds \sin \theta d\theta d\lambda$ on the mapped grid (`_tisserand_row`). In dynamic runs the rotation follows from the equations, and the output removes the mean rotation of the mantle, $\xi = \int \rho_0 \mathbf{r} \times \mathbf{u} dV / \int \rho_0 \mathbf{r} \times \mathbf{b} dV$ over the mantle, by subtracting $\xi \mathbf{b}$ from the surface displacement (`_remove_mantle_rotation; frame='surface'` fits ξ to the surface displacement instead).

9.6 The synthesis of etide

For a wave k of degree n and order m with Tamura amplitude A_k (plus its secular rate $\dot{A}_k$ times the time in Julian centuries) and argument $\chi_k = \sum_{j=1}^8 d_{kj} f_j + 90^\circ p_k$, where d_{kj} are its Doodson

multipliers, p_k its phase code and f_j the eight astronomical angles of Tamura (1987) (local lunar time, s , h , p , N' , p_s and two planetary angles, computed from terrestrial time), the displacement is

$$\begin{aligned} u_r &= \sum_k A_k \operatorname{Re}[h_k e^{i\chi_k}] W_{nm}(\varphi), & u_N &= \sum_k A_k \operatorname{Re}[l_k e^{i\chi_k}] \frac{dW_{nm}}{d\varphi}, \\ u_E &= \sum_k A_k \operatorname{Re}[i m l_k e^{i\chi_k}] \frac{W_{nm}(\varphi)}{\cos \varphi}, \end{aligned} \quad (9.6)$$

with $h_k = h_k^{(0)} + h_k^{(2)} P_2(\sin \varphi)$, $l_k = l_k^{(0)} + l_k^{(2)} P_2(\sin \varphi)$ for degree 2 (the Love numbers of degrees 3 and 4 without latitude terms), plus the $l^{(1)}$ terms of Petit and Luzum (2010) (Sect. 7.1.1) for the diurnal and semidiurnal waves. W_{nm} is the potential per unit amplitude divided by the equatorial gravity, at the equatorial radius:

$$W_{nm}(\varphi) = \frac{C_T R_e}{GM/R_e^2} \frac{P_{nm}(\sin \varphi)}{\gamma_{nm}}, \quad C_T = \frac{3}{4} \frac{M_{\text{Moon}}}{M} GM \frac{\sin^3 \pi_{\text{Moon}}}{R^2}, \quad (9.7)$$

with Tamura's Doodson-normalized functions P_{nm}/γ_{nm} , the mass ratio $M_{\text{Moon}}/M = 0.012300034$, the lunar parallax $\pi_{\text{Moon}} = 3422.448''$ and $R = 6378137$ m, the equatorial radius of Tamura's catalogue (GRS80; $R_e = 6378136.6$ m of the IERS is used in W_{nm}) (etide.F_nm, W_functions).

References

- Agnew, D. C. 2007. "Earth Tides." In *Treatise on Geophysics, Vol. 3: Geodesy*, edited by T. A. Herring. Elsevier.
- Alterman, Z., H. Jarosch, and C. L. Pekeris. 1959. "Oscillations of the Earth." *Proc. R. Soc. Lond. A* 252: 80–95.
- Blewitt, Geoffrey. 2003. "Self-Consistency in Reference Frames, Geocenter Definition, and Surface Loading of the Solid Earth." *Journal of Geophysical Research: Solid Earth* 108 (B2): 2103. <https://doi.org/10.1029/2002JB002082>.
- Bos, M. S. 2026. "Tidal Love Numbers of an Ellipsoidal, Rotating Earth Without Generalized Spherical Harmonics." *Geophys. J. Int.*
- Bos, M. S., and H.-G. Scherneck. 2013. "Computation of Green's Functions for Ocean Tide Loading." In *Sciences of Geodesy – II*, edited by G. Xu. Springer. https://doi.org/10.1007/978-3-642-28000-9_1.
- Boyd, J. P. 2001. *Chebyshev and Fourier Spectral Methods*. 2nd ed. Dover.

- Chandrasekhar, S. 1969. *Ellipsoidal Figures of Equilibrium*. Yale University Press.
- Dahlen, F. A., and J. Tromp. 1998. *Theoretical Global Seismology*. Princeton University Press.
- Dehant, V. 1987. "Integration of the Gravitational Motion Equations for an Elliptical Uniformly Rotating Earth with an Elastic Mantle." *Phys. Earth Planet. Inter.* 49: 242–58.
- Dehant, V., P. Defraigne, and J. M. Wahr. 1999. "Tides for a Convective Earth." *J. Geophys. Res.* 104 (B1): 1035–58.
- Greff-Lefftz, M., L. Métivier, and H. Legros. 2005. "Analytical Solutions of Love Numbers for a Hydrostatic Ellipsoidal Incompressible Homogeneous Earth." *Celest. Mech. Dyn. Astr.* 93: 113–46. <https://doi.org/10.1007/s10569-005-6424-3>.
- Hough, S. S. 1895. "The Oscillations of a Rotating Ellipsoidal Shell Containing Fluid." *Phil. Trans. R. Soc. Lond. A* 186: 469–506. <https://doi.org/10.1098/rsta.1895.0012>.
- Longman, I. M. 1963. "A Green's Function for Determining the Deformation of the Earth Under Surface Mass Loads, 2, Computations and Numerical Results." *J. Geophys. Res.* 68: 485–96.
- Love, A. E. H. 1911. *Some Problems of Geodynamics*. Cambridge University Press.
- Mathews, P. M., B. A. Buffett, and I. I. Shapiro. 1995. "Love Numbers for a Rotating Spheroidal Earth: New Definitions and Numerical Values." *Geophys. Res. Lett.* 22: 579–82.
- Mathieu, É. 1868. *Mémoire Sur Le Mouvement Vibratoire d'une Membrane de Forme Elliptique*. J. Math. Pures Appl.
- Métivier, L., M. Greff-Lefftz, and M. Diament. 2006. "Mantle Lateral Variations and Elastogravitational Deformations – I. Numerical Modelling." *Geophys. J. Int.* 167 (3): 1060–76. <https://doi.org/10.1111/j.1365-246X.2006.03159.x>.
- Nayfeh, A. H., and D. T. Mook. 1979. *Nonlinear Oscillations*. Wiley.
- Pekeris, C. L., and Y. Accad. 1972. "Dynamics of the Liquid Core of the Earth." *Phil. Trans. R. Soc. Lond. A* 273 (1233): 237–60.
- Pekeris, C. L., and H. Jarosch. 1958. "The Free Oscillations of the Earth." *Contributions in Geophysics in Honor of Beno Gutenberg*, 171–92.
- Petit, G., and B. Luzum. 2010. *IERS Conventions (2010)*. IERS Technical Note No. 36. Verlag des Bundesamts für Kartographie und Geodäsie.
- Rekier, J. 2022. "Free Core Nutation and Its Relation to the Spin-over Mode." *Planet. Sci. J.* 3: 133. <https://doi.org/10.3847/PSJ/ac6ce2>.
- Takeuchi, H. 1950. "On the Earth Tide of the Compressible Earth of Variable Density and Elasticity."

- Trans. Am. Geophys. Un.* 31: 651–89.
- Tamura, Y. 1987. “A Harmonic Development of the Tide-Generating Potential.” *Bull. Inf. Marées Terrestres* 99: 6813–55.
- Thomson, W. 1863. “On the Rigidity of the Earth.” *Phil. Trans. R. Soc. Lond.* 153: 573–82.
- Timoshenko, S., and J. N. Goodier. 1951. *Theory of Elasticity*. 2nd ed. McGraw-Hill.
- Trefethen, L. N. 2000. *Spectral Methods in MATLAB*. SIAM.
- Tromp, J., and J. X. Mitrovica. 1999. “Surface Loading of a Viscoelastic Earth – I. General Theory.” *Geophys. J. Int.* 137: 847–55.
- Wahr, J. M. 1981. “Body Tides on an Elliptical, Rotating, Elastic and Oceanless Earth.” *Geophys. J. R. Astr. Soc.* 64: 677–703.
- Wahr, J. M., and Z. Bergen. 1986. “The Effects of Mantle Anelasticity on Nutations, Earth Tides, and Tidal Variations of Rotation Rate.” *Geophys. J. R. Astr. Soc.* 87: 633–68.
- Wenzel, H.-G. 1996. “The Nanogal Software: Earth Tide Data Processing Package ETERNA 3.30.” *Bull. Inf. Marées Terrestres* 124: 9425–39.