# LOV3D

Version 1.0

M.Rovira-Navarro

November 23, 2023

# Contents



# List of Code Listings

# 1 Introduction

`LOV3D` is a code developed to obtain the tidal response of bodies with lateral variations of interior properties. The tidal response is obtained using the semi-analytical approach detailed in Rovira-Navarro et al. (2023). This version of `LOV3D` computes the Love numbers, radial functions and energy dissipation spectra for a 3 layer interior model consisting of (1) a solid inner core, (2) a liquid core and (3) a solid envelope with laterally varying properties.

References to equations correspond to Rovira-Navarro et al. (2023).

Section 2 details the input, outputs of the code, Section 3 introduces the main subroutines and functions and Section 4 describes the sample script `example.mat` containing a test-case to reproduce the results in Figure 7. Section 5 lists ongoing work to extend the software.

# 2 Inputs & Outputs

The inputs and outputs produced by the code are listed here. For individual functions, inputs and outputs are described in the header.

## Outputs

- `Interior_Model` Structure containing information about the interior. Both dimensional and non-dimensional quantities can be provided. Dimensional quantities are indicated with a 0 and non-dimensional without. If dimensional quantities are provided, they are non-dimensionalized inside the code. The structure contains the following fields.

```
% Interior_Model.R0: radius
    % R0(1) core radius (solid+liquid core)
    % R0(2) surface radius
% Interior_Model.rho0: layers density
    % rho0(1) core density (solid+liquid core)
    % rho0(2) density of the outermost solid layer
% Interior_Model.Delta_rho0: Density difference between the liquid
    core and overlying solid layer. If not provided it is computed
    assuming that the two innermost layers have rho0(1).
% Interior_Model.mu0: shear modulus of the the outermost layer
% Interior_Model.Ks0: bulk modulus of the outermost layer
% Interior_Model.eta0: viscosity of the outermost layer
% Interio_Model.mu_variable: shear modulus variations
    % mu_variable(:,1): degree of variation
    % mu_variable(:,2): order of variation
    % mu_variable(:,3): amplitude of the variation (mu_l^m/mu^0_0)
% Interio_Model.K_variable: bulk modulus variations
    % K_variable(:,1): degree of variation
    % K_variable(:,2): order of variation
    % K_variable(:,3):  amplitude of bulk modulus variations (K_l^
       m/K^0_0)
% Interio_Model.eta_variable: viscosity
```

```
    % eta_variable(:,1): degree of variation
    % eta_variable(:,2): order of variation
    % eta_variable(:,3):  amplitude of viscosity variations (eta_l
       ^m/eta^0_0)
% Interior_Model.rheology_variable: complex shear and bulkd
   modulus lateral variations (assigned inside the code)
    % rheology_variable(:,1): degree of variation
    % rheology_variable(:,2): order of variation
    % rheology_variable(:,3):  amplitude of bulk modulus
       variations (K_l^m/K^0_0) --- this has not yet been used and
        tested
    % rheology_variable(:,4):  amplitude of complex shear modulus
       variations (mu_l^m/mu^0_0)
% Interior_Model.muC: complex shear modulus (assigned inside the
   code)
```

- `Forcing` Structure containing information about the forcing.

```
% Forcing.Td: forcing period
% Forcing.n: degree of the forcing
% Forcing.m: order of the forcing
% Forcing.F: amplitude of the component
```

- `Numerics` Structure containing numerical information.

```
% Numerics.Nr: number of radial points
% Numerics.perturbation_order: maximum order of the perturbation.
   Default is 2
% Numerics.rheology_cutoff: determines which terms of the rheology
    are include, (only relevant for viscoelastic rheology:  terms
   with log10(mu_n^m)-log10(mu_n^m(leading))>=-Numerics.
   rheology_cutoff are included. Default 0 (only leading terms)
% Numerics.load_couplings: indicates if the coupling coefficients
   are computed or loaded from existing file
    % (0) compute coupling coefficients from scratch
    % (1) load coupling coefficients from file that contains the
       coupling coefficintes for the specific rheology variations
       considered. If the file does not exist, it is generated and
        stored in Files_Coupling
    % (2) load coupling coefficintes from a file that contains all
        couplung coefficients up to a given degree. If the file
       does not exist, it is generated and stored in
       Files_Coupling
```

## Outputs

- `Love` Love number spectra (see Eq. (29)). Output of `get_Love`

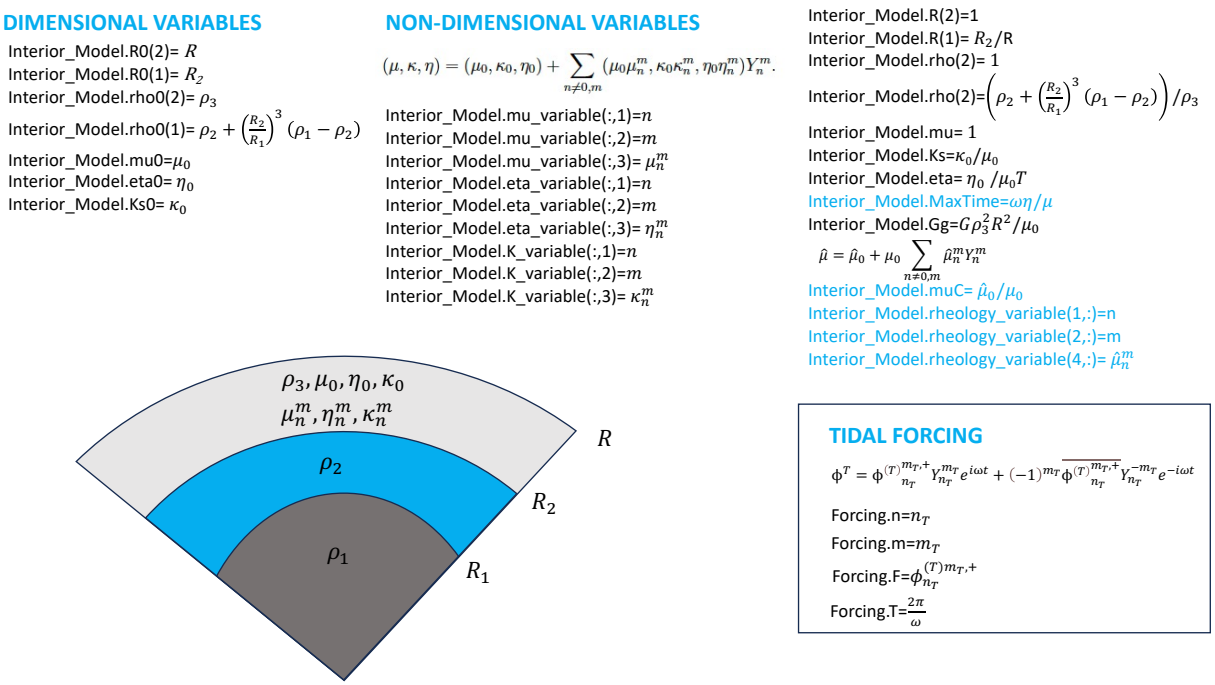


Figure 1

Figure 2: Sketch of interior model and relation to inputs. Fields indicated in blue are computed inside `get_Love`.

```
% Love_Spectra.nf: degree of the forcing
% Love_Spectra.mf: order of the forcing
% Love_Spectra.n: degree of the solution mode
% Love_Spectra.m: order of the solution mode
% Love_Spectra.k: gravity Love numbers
```

- `y_rad`: Radial Functions (see Eq. (40)). Output of `get_Love`

```
% y_rad.nf: degree of the forcing
% y_rad.mf: order of the forcing
% y_rad.n: degree of the solution mode
% y_rad.m: order of the solution mode
% y_rad.y(r,X,mode); where r stands for radial point and X
    % y_rad.y(r,1,mode): r radial position
    % y_rad.y(r,2,mode): U radial displacement
    % y_rad.y(r,3,mode): V tangential displacement
    % y_rad.y(r,4,mode): R normal stress
    % y_rad.y(r,5,mode): S tangential stress
    % y_rad.y(r,6,mode): \phi gravitational potential
    % y_rad.y(r,7,mode): \dot\phi potential stress
    % y_rad.y(r,8,mode): W toroidal displacement
    % y_rad.y(r,9,mode): T toroidal stress
    % y_rad.y(r,10,mode): u_{n,n-1}
```

```
    % y_rad.y(r,11,mode): u_{n,n}
    % y_rad.y(r,12,mode): u_{n,n+1}
    % y_rad.y(r,13,mode): \sigma_{n,n,0}
    % y_rad.y(r,14,mode): \sigma_{n,n-2,2}
    % y_rad.y(r,15,mode): \sigma_{n,n-1,2}
    % y_rad.y(r,16,mode): \sigma_{n,n,2}
    % y_rad.y(r,17,mode): \sigma_{n,n+1,2}
    % y_rad.y(r,18,mode): \sigma_{n,n+2,2}
    % y_rad.y(r,19,mode): \epsilon_{n,n,0}
    % y_rad.y(r,20,mode): \epsilon_{n,n-2,2}
    % y_rad.y(r,21,mode): \epsilon_{n,n-1,2}
    % y_rad.y(r,22,mode): \epsilon_{n,n,2}
    % y_rad.y(r,23,mode): \epsilon{n,n+1,2}
    % y_rad.y(r,24,mode): \epsilon_{n,n+2,2}
```

- `y_LatLon` `y` transformed to the spatial domain. Output of `get_map`

```
% y_LatLon.nf: degree of the forcing
% y_LatLon.mf: order of the forcing
% y_LatLon.lon: longitude
% y_LatLon.lat: latitude
% y_LatLon.r: radial point at which y functions are evaluated
% y_LatLon.mu(lon,lat): shear modulus
% y_LatLon.forcing(lon,lat): forcing
% y_LatLon.y(lon,lat,r,1):  Gravitational Potential
% y_LatLon.y(lon,lat,r,2):  Displacement e_r component
% y_LatLon.y(lon,lat,r,3):  Displacement e_theta component
% y_LatLon.y(lon,lat,r,4):  Displacement e_phi component
% y_LatLon.y(lon,lat,r,5):  stress   e_r e_r component
% y_LatLon.y(lon,lat,r,6):  stress   e_r e_theta component
% y_LatLon.y(lon,lat,r,7):  stress   e_r e_phi component
% y_LatLon.y(lon,lat,r,8):  stress   e_theta e_r component
% y_LatLon.y(lon,lat,r,9):  stress   e_theta e_theta component
% y_LatLon.y(lon,lat,r,10): stress   e_theta e_phi component
% y_LatLon.y(lon,lat,r,11): stress   e_phi e_r component
% y_LatLon.y(lon,lat,r,12): stress   e_phi e_theta component
% y_LatLon.y(lon,lat,r,13): stress   e_phi e_phi component
% y_LatLon.y(lon,lat,r,14): strain   e_r e_r component
% y_LatLon.y(lon,lat,r,15): strain   e_r e_theta component
% y_LatLon.y(lon,lat,r,16): strain   e_r e_phi component
% y_LatLon.y(lon,lat,r,17): strain   e_theta e_r component
% y_LatLon.y(lon,lat,r,18): strain   e_theta e_theta component
% y_LatLon.y(lon,lat,r,19): strain   e_theta e_phi component
% y_LatLon.y(lon,lat,r,20): strain   e_phi e_r component
% y_LatLon.y(lon,lat,r,21): strain   e_phi e_theta component
% y_LatLon.y(lon,lat,r,22): strain   e_phi e_phi component
```

- `Energy_Spectra` Energy dissipation spectra (Eq. (32)). Output of `get_energy`

```
%Energy_Spectra.n: degrees with non-zero energy
%Energy_Spectra.m: orders with non-zero energy
%Energy_Spectra.n_v: degrees from 0 to Numerics.Nenergy
%Energy_Spectra.n_v: orders from 0 to Numerics.Nenergy
%Energy_Spectra.energy(r,mode): radial profile of energy spectra
%Energy_Spectra.energy_integral(mode): radially integrated energy
   for all non-zero degrees an orders (n,m)
%Energy_Spectra.energy_integral_v(mode): radially integrated
   energy for all degrees an orders (n_v,m_v)
```

### A note about the LOV3D use of nondimensional units

LOV3D uses nondimensional units. The spatial and temporal timescales are the surface radiii and the orbital period $R$ and $T$. Stress and rheological parameteres (e.g., shear and bulk modulii) are non-dimensionalized with the shear modulus of the solid envelope $\mu_0$ and density with the density of the solid envelope ($\rho_3$). The following relations between dimensional and nondimensional quantities —indicated with a $'$— can be derived:

$$\mu_0' = 1 \quad \eta' = \frac{1}{T\mu_0}\eta_0 \quad \kappa_0' = \frac{1}{\mu_0}\kappa_0$$

$$r' = \frac{1}{R}r \quad u' = \frac{1}{R}\mathbf{u} \quad \boldsymbol{\sigma}' = \frac{1}{\mu_0}\boldsymbol{\sigma}$$

$$\dot{e}' = \frac{T}{\mu_0}\dot{e} \quad \dot{E}' = \frac{T}{R^3\mu_0}\dot{E}$$

$$G' = \frac{\rho_0^2 R^2}{\mu_0}G \quad \phi' = \frac{\rho_0}{\mu_0}\phi \quad g' = \frac{\rho_0 R}{\mu_0}g$$

Here, $\boldsymbol{\sigma}$, $\mathbf{u}$, $\dot{e}$, $\dot{E}$, $g$ and $\phi$ stand for the stress tensor, the displacement field, the volumetric and total energy dissipation, and the gravitational acceleration and potential. Note that LOV3D computes the tidal response assuming a unit tidal forcing ${\phi'}^{T m_T}_{n_T} = 1$. Thus in order to recover the tidal response one needs to multiply the solution `y` by the factor $A\rho_0/\mu_0$ where $A$ is the amplitude of the tidal force in dimensional units. For example, for a moon in an eccentric synchrnous orbit $A = (\omega R)^2 e$, where $e$ is the moon eccentricity (see Appendix D and Table 2). In such a case, the dimensional volumetric energy dissipation follows from:

$$\dot{e} = \frac{\rho_0^2 \omega^4 R^4 e^2}{\mu_0 T}\dot{e}' \tag{2}$$

# 3 Functions & Subroutines

The functions are stored in `/Functions`. The main functions are:

- `get_Love.m` computes the tidal response of the body for a given `Interior_Model` and `Forcing` using the numerical information specified in `Numerics`. It employs the subroutines: `get_solution`, `get_couplings`.
- `get_couplings` determines the modes that intervene in the tidal response and the coupling coefficients (see Appendix B). It employs spherical harmonic functions contained in `Functions/SPH_Tools`. The coupling coefficients are stored in `Files_Coupling` for future use.
- `get_solution` computes the tidal response by numerically-integrating the governing equations using a Runge-Kutta integrator (see Section 4 for more details).
- `get_energy` obtains the energy dissipation spectra (`Energy_Spectra`) given the tidal response of the body (`y`). It uses the function `couplings_energy`.
- `couplings_energy` computes the energy dissipation integrals defined in Appendix B. The energy integrals are stored in `Files_Coupling` for future use.
- `get_map` computes the required tensor spherical harmonics and transforms the spectral solution, `y`, to the spatial domain `y_LatLon` using the transformation of Appendix A
- `plot_map` plots `y_LatLon`

Additionally, the directory `/SPH_Tools` contains several functions used for Spherical Harmonics related computations, including:

- `Legendre.m` computes the normalized associated Legendre functions.
- `Wigner3j.m`, `Wigner6j.m` and `Wigner9j.m` obtains the Wigner symbols required to obtain the coupling integrals. These functions were developed by Vladimir Sovkov and are available at MATLAB Central File Exchange
- `LatLon_SPH` converts a field in the spatial domain to the spectral domain (real spherical harmonics).
- `SPH_LatLon` converts a field from the spectral domain (real spherical harmonics) to the spatial domain.

The relation between the different functions is illustrated in the flow diagram of Figure 3.

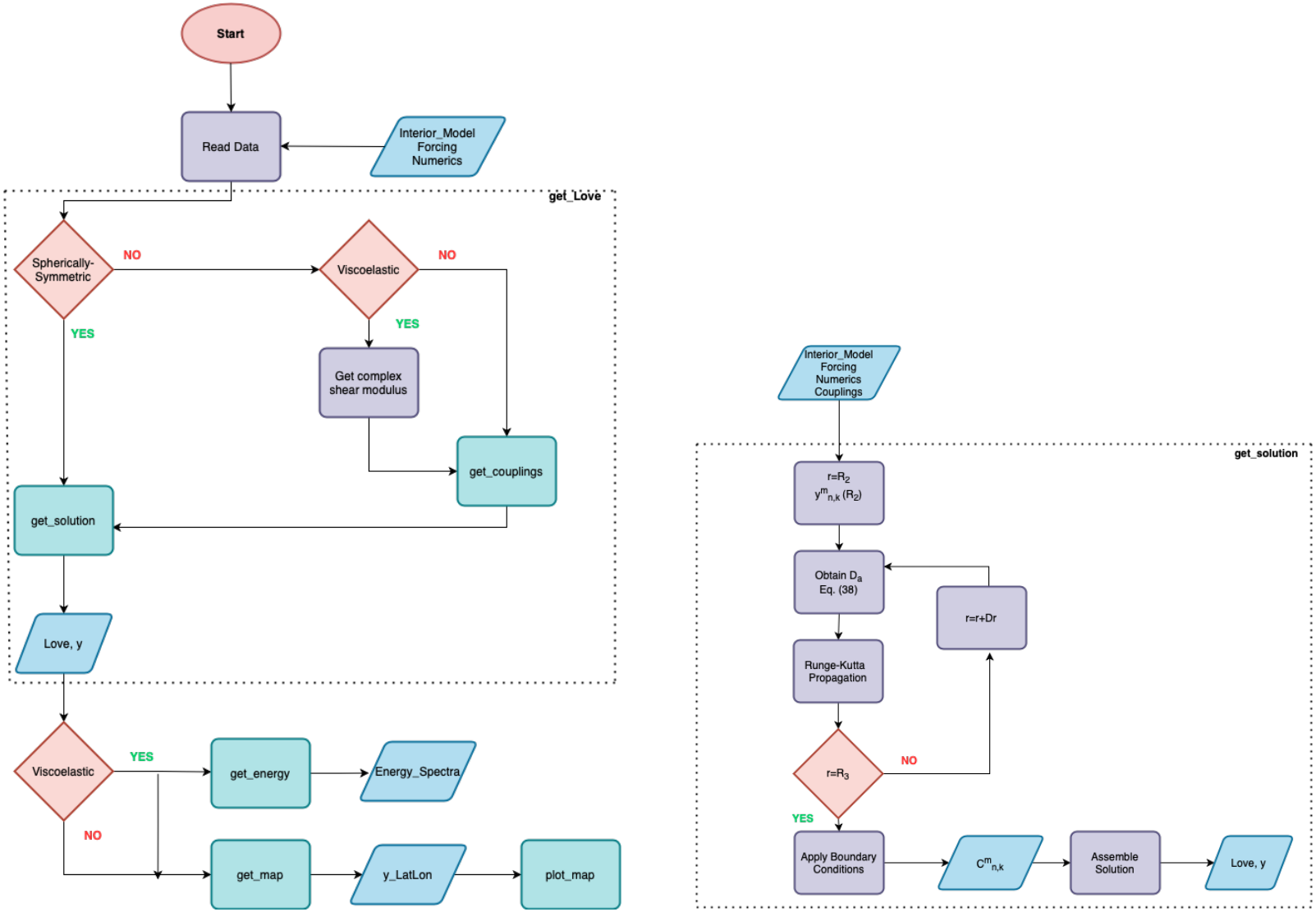


Figure 3: Code Flow Diagram (a) and detail of the function get_solution flow diagram (b)

# 4 Script examples

The script `/Scripts/example` provides a test-run for the code. It can be used to reproduce the results in Section 7 of Rovira-Navarro et al. (2023). For this example:

- We consider the reference Io model of Table 1 with monochromatic lateral viscosity variations
- The Love numbers and energy spectra for the model and a spherically-symmetric counterpart are computed
- Gravitational potential, surface displacements, stress and strain tensors, and tidal heating maps are plotted.
- In all cases, non-dimensional units are used. See `get_Love.m` L161 for a reference to non-dimensional units.

# 5 Work in Progress

The authors of this code are currently working on:

- Including radial variations of interior properties.
- Extending boundary conditions to consider other types of loads (e.g., surface loads).

Please check this github repository for an updated version of the code.

# References

Rovira-Navarro, M., Matsuyama, I., & Berne, A. 2023, (in preparation)