

MATLAB Solution for the Toroidal Modes of a Vibrating Spherical Shell

Draft 2.6

Charles J. Ammon, Saint Louis University

One advantage of studying theoretical seismology today is the availability of computer tools that we can use to explore equations and their solutions. Tools such as MATLAB create new opportunities for deeper understanding – with a modest effort we can investigate the nature of problems that have long been fundamental to the interpretation and understanding of seismograms. In these notes, I outline a MATLAB solution for computing toroidal normal modes in radially symmetric earth models. To simplify the discussion, I work with simplest model - a uniform shell (Figure 1). This physical model is actually a surprisingly good approximation to Earth at the longest periods of oscillation. Although Earth has a very dense, mostly iron core, the fluid outer core results in a zero toroidal stress at the core-mantle boundary, much like the elastic shell model shown in Figure 1.

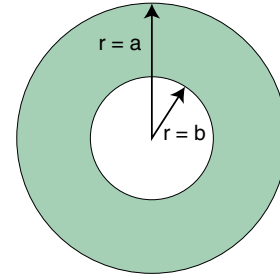


Figure 1. Model cross section. Since the outer core is fluid, a shell is representative of Earth for toroidal vibrations.

To compute the eigenfrequencies and eigenfunctions of the model we must solve two problems. First, we must develop a tool to integrate two coupled first-order ordinary differential equations (ODE's), and second we must develop a tool to identify those values of frequency that produce a zero surface stress when the ODE's are integrated.

The Equations of Motion

A separation of variables solution to the homogeneous elastodynamic equations of motion results in three equations for the three space variables (r , θ , ϕ). The longitudinal (ϕ) equation is a simple second order harmonic ordinary differential equation. The equation containing colatitude, (θ), reduces to the a form of Legendre's equation with solutions that are the Associated Legendre functions. The equation in radius is a form of Bessel's equation, which has solutions in terms of spherical Bessel functions. Alternatively, Bessel's second order differential equation can be reduced to a set of first-order, coupled ODE's. Specifically, for the toroidal modes of a radially symmetric model the radial equations are equivalent to the eigenvalue problem

$$\frac{d}{dr} \begin{bmatrix} W \\ T \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{\mu(r)} \\ -\omega^2 \rho(r) + \frac{(l+2)(l-1)\mu(r)}{r^2} & \frac{-3}{r} \end{bmatrix} \begin{bmatrix} W \\ T \end{bmatrix}. \quad (1)$$

$W(r)$ and $T(r)$ are the displacement and the stress, ω is the frequency, $\rho(r)$ is the density, $\mu(r)$ is the shear modulus, and l is the radial order (a separation of variables constant). You can find the derivations of the equations in Dahlen and Tromp (1999) (our equation (1) corresponds to equations 8.114 and 8.115). For low frequencies and low radial orders, l , these equations can be integrated numerically using a Runge-Kutta approach (e.g. Press et al, 1988).

Of course before you integrate (1), you must specify values of l , the density and shear modulus, and the frequency, ω . Integration produces two functions of radius, $\hat{W}(r)$ and $\hat{T}(r)$. If $\hat{T}(a) = 0$ and $\hat{T}(b) = 0$, the assumed value of ω is an eigenfrequency (ω^2 is an eigenvalue) and $\hat{W}(r)$ and $\hat{T}(r)$ are eigenfunctions. We don't know *a priori* the eigenfrequencies that satisfy the zero-stress boundary condition on the top and bottom of the shell. However, if we start the integration at the bottom of the shell ($r = b$), we can guarantee that the stress at the bottom is zero by using initial values for the displacement and the stress equal to 1.0 and 0.0 respectively. The arbitrary amplitude of the displacement can be accommodated in normalization terms.

In general, there are an infinite number of frequencies that satisfy equation (1) for each value of l . A systematic search must be performed to identify all the roots in a specific frequency range for a particular value of l . In essence, we must solve an equation that has the form

$$F(l, \omega_n) = 0. \quad (2)$$

The form of the function, F , is described in equation (1), and we must solve those equations for the surface stress, $\hat{T}(a)$ for each assumed value of l and ω . Finding solutions that have zero surface stress is a numerical root finding problem - you assume a value of ω and integrate the equations to see if it produces zero stress at the outer surface.

The MATLAB Solution

Our solution will involve three different M-scripts, two are needed to solve the equations of motion (1), and the third is used to find the values of ω that are eigenfrequencies (Figure 2).

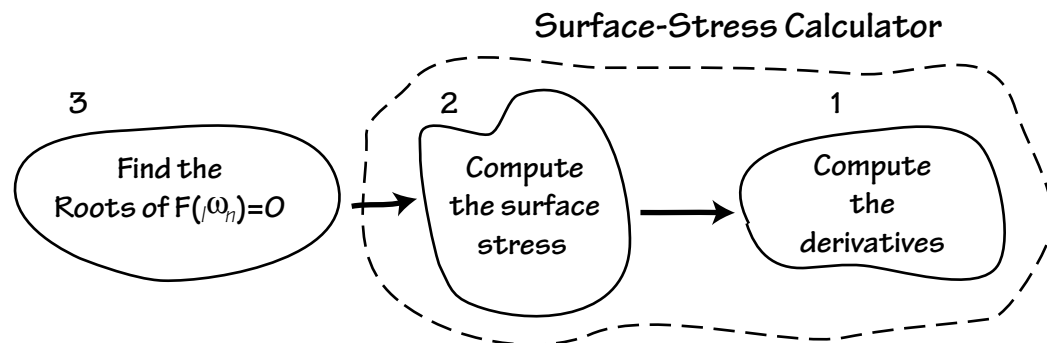


Figure 2. Solving the eigenvalue problem for the vibrating shell is accomplished using three scripts. The main purpose of each script is described in the above chart, which also shows the dependencies (the third “calls” the second, which requires the first).

Script Listing 1: stress_disp_tor.m

```

function dsdd = stress_disp_tor(r,s)
global mu rho omega l c0;
%
% The input values of s(1) and s(2) are W(r) and T(r) respectively
% The returned derivatives are stored in dsdd()
%
% The displacement
dsdd(1,1) = s(1) / r + s(2) / mu;
%
% the stress
dsdd(2,1) = ((1-1)*(1+2)*mu/(r*r) - rho*omega*omega)*s(1) - 3*s(2)/r;

```

The first two M-files handle the computation of the surface stress using the Runge-Kutta routines in MATLAB (the surface-stress calculator). The third script repeatedly calls the surface-stress calculator as part of a simple root finding scheme that uses a fixed step size in frequency to identify regions containing the roots, and then relies on the MATLAB root-finding routine *fzero()* to refine the value of the eigenfrequency. I describe each script in more detail below, and conclude with some results for specific values of l .

Computing the Surface Stress

To integrate the equations and evaluate the surface traction I created two M-files. The first computes the stress and displacement derivatives using the differential equations (1). The input arguments and return values of this M-file are of the special form required by the Runge-Kutta functions in MATLAB and the name of this M-file is an argument to the MATLAB Runge-Kutta function. The first input argument is the independent variable of integration (radius for our problem), the second is a vector that contains the dependent variables in the coupled first-order differential equations (displacement (1) and stress (2)). I set up μ , ρ , ω , and l as global variables so that they can be set and changed from the main script (listed later). The same global values are used in all the scripts. It's a good idea to keep this script as simple and short as you can, since it's executed often. Listing 1 contains the details.

The second M-file contains a script that is used to calculate the surface stress and is shown in Listing 2. For a given value of ω , the script *surface_stress.m* performs the computations needed to compute the stress and displacement eigenvectors. Note the brevity and simplicity of the two scripts used to solve equation (1). Since the Runge-Kutta routines are implemented in MATLAB, we need only a two-line script to compute the derivatives and a one-line call to *ode45()* to integrate the equations. Once the integration is complete, we look up the surface stress (at $r = 6371$ km), which after execution is stored in the last element of array *sd*.

Script Listing 2: surface_stress.m

```

function value = surf_stress(f)

global mu rho omega l c0 radius sd;
%
rspan = [2891000, 6371000]; % radius limits in meters
sd0 = [1.0; 0.0]; % the initial values of [displacement stress]
%
omega = 2 * pi * f; % angular frequency
%
% the integration is performed here, by MATLAB
% the first argument is the name of our routine that calculates
% the derivatives
% on return the vectors radius and sd contain the values of radius
% and the displacement and stress eigenfunctions
%
[radius,sd] = ode45('stress_disp_tor',rspan,sd0);
%
nr = size(sd(:,2)); % number of points need in the integration
value = sd(nr(1),2); % pull out the top value of stress
% (surface, r = 6371 km).

```

Finding the Roots

The third and final script implements a simple root-finding search and provides visual feedback on the process. You should have no trouble finding lucid descriptions of the ideas behind searching for roots, or zeros, of a function (e.g. Press et al, 1988). The basic idea in this script is to begin a search at a small value of frequency (I chose $f = 1/3600$ corresponding to a period of one hour) and march forward with a small step size (df), looking for steps that produce a change in sign of the surface stress estimate, $\hat{T}(a)$. If the sign of the surface stress changes then we know that we've bracketed at least one zero of the function. Of course we have to be careful, a step size too large may jump more than one root in one step. If we choose too small a value for df , then we spend much more time searching. More clever approaches are more appropriate when dealing with more complicated models. For this example I opted to use a small step size rather than getting too fancy with something adaptive.

Once I know two frequency values that bracket a root, I use the MATLAB function `fzero()` to find that root. `fzero()` uses a bisection-interpolation procedure to find the root of the function. You must pass the name of the M-file used to calculate the surface stress and the bounding values of frequency to the `fzero()` routine, and it returns the value of the root. MATLAB allows you to control the options in the search for roots, but I found the default values suitable for this problem.

Script Listing 3: spshell.m

```

clear all;
global mu rho omega l c0 radius sd;
rho = 4380; % a rough average of earth's mantle
mu = 5930*5930*rho; % properties of the shell 5.93 km/s shear velocity
%
l = 25; % the angular order - change this and execute spshell
i = 1; j = 1; % counters, j will be the root number (starting at one)
%
f = 1/3600; % the initial frequency to start (T = one hour)
df = 0.0002; % the frequency step size (chosen by trial and error)
oldvalue = 0;
%
while(f < 0.08) % stopping at 0.08 is arbitrary...
    %
    % compute the surface stress for this frequency
    % this is calling the scripts listed above
    surface_stress = surf_stress(f);
    %
    % here we see if the value of the surface-stress has changed sign
    % which would indicate that we passed at least one root
    % If we did cross a root, I call the matlab function fzero to
    % refine the root.
    % Then I store the root in the vector root() and plot the results.
    %
    if(oldvalue * surface_stress < 0)
        theRoot = fzero('surf_stress', [oldf f]);
        root(j) = theRoot;
        subplot(3,3,j); % this is where the graphics begin
        mytitle = sprintf(' = %.2f mHz, T = %.0f min',root(j)*1000,1/theRoot/60);
        plot(sd(:,1)/max(abs(sd(:,1))),radius/1000,'b',sd(:,2)/ ...
            max(abs(sd(:,2))),radius/1000,'r');
        axis([-1.2 1.2 2890 6800]);grid;
        title(['\omega' mytitle]);
        txt = sprintf('%d',j-1); text(-1,6371,['n = ' txt]); %root label
        if mod(j-1,3) == 0
            ylabel('radius (km)');
        end % this is the end of the graphics commands
        if( j = 9 ) f = 1e6; end % this halts the search after nine plots
        j = j + 1;
    end
    %
    % save the old values of the stress and frequency
    oldvalue = surface_stress;
    oldf = f;
    %
    % step forward in frequency
    %
    f = f + df;
    i = i + 1;
    % provide some feedback on the surface-stress value
    % (which is huge for large l)
    disp(sprintf('%d %.4f %f %.1f %.2f', i, surface_stress, f, 1/f, 1/f/60))
end

```

The Uniform Shell Model

As mentioned earlier, the shell is a better approximation to Earth than you might think. Approximate values of mantle parameters for the PREM (Preliminary Earth Reference Model) are $4,380 \text{ kg/m}^3$ and 5.93 km/s for the density and shear velocity respectively. We can compute the shear modulus using $\mu = \beta^2 \rho$.

Results!

Much of the third script is actually devoted to plotting the results. What do they look like? Figure 3 is a plot of the results of running the scripts described above after setting $l = 2$, and Figure 4 and Figure 5 are similar plots for $l = 25$, and $l = 100$, respectively. In each illustration, the eigenfrequency in millihertz and the eigenperiod (to the nearest minute) are listed in the subplot

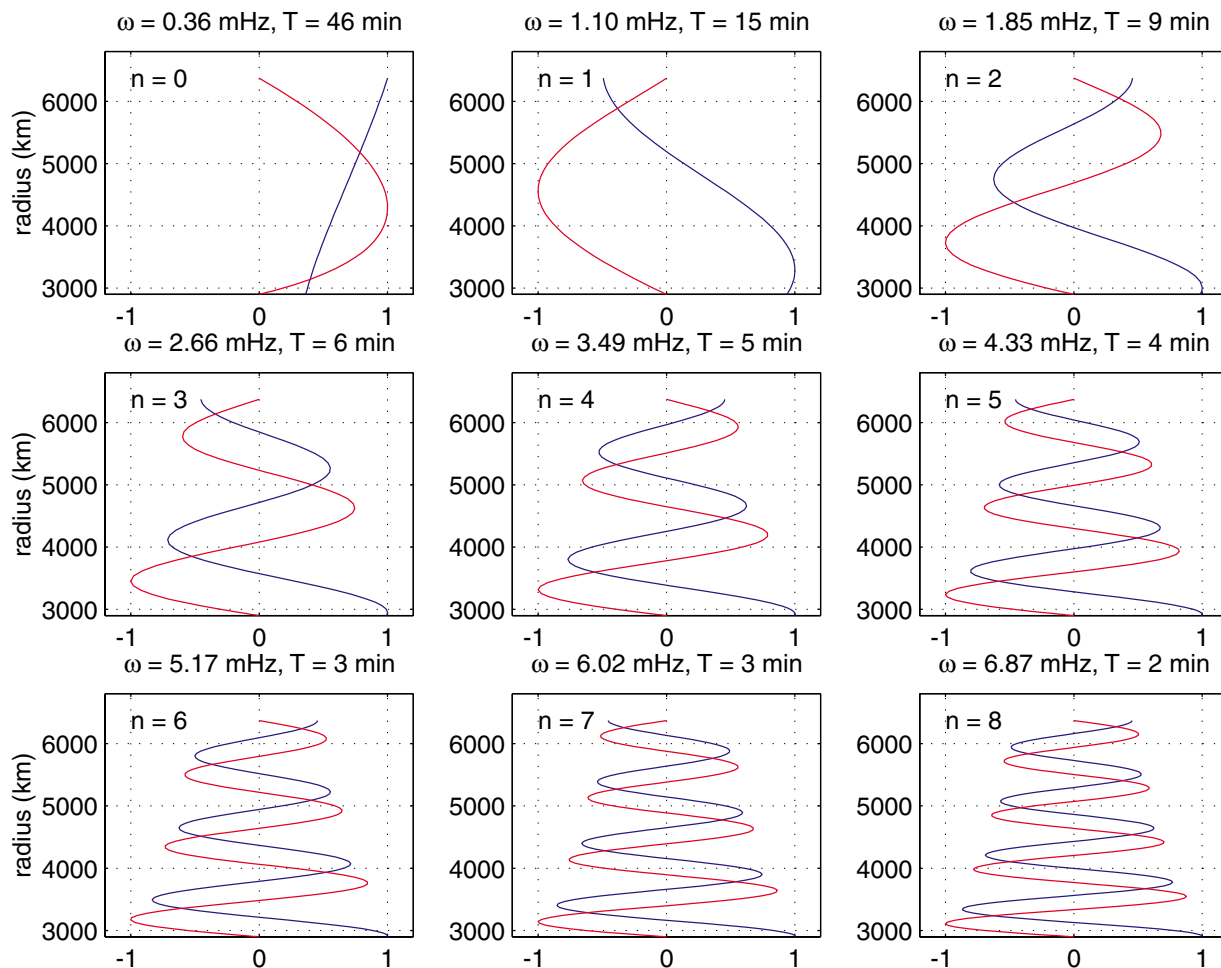


Figure 3. Eigenfrequencies and eigenfunctions (displacement in blue, stress in red) for a vibrating shell overlying a fluid sphere. The shell density is $4,380 \text{ kg/m}^3$ and the shear velocity is 5.93 km/s . The radial order for the calculation, l , is equal to two. The eigenfrequency in millihertz, and the eigenperiod, to the nearest minute, are shown above each panel.

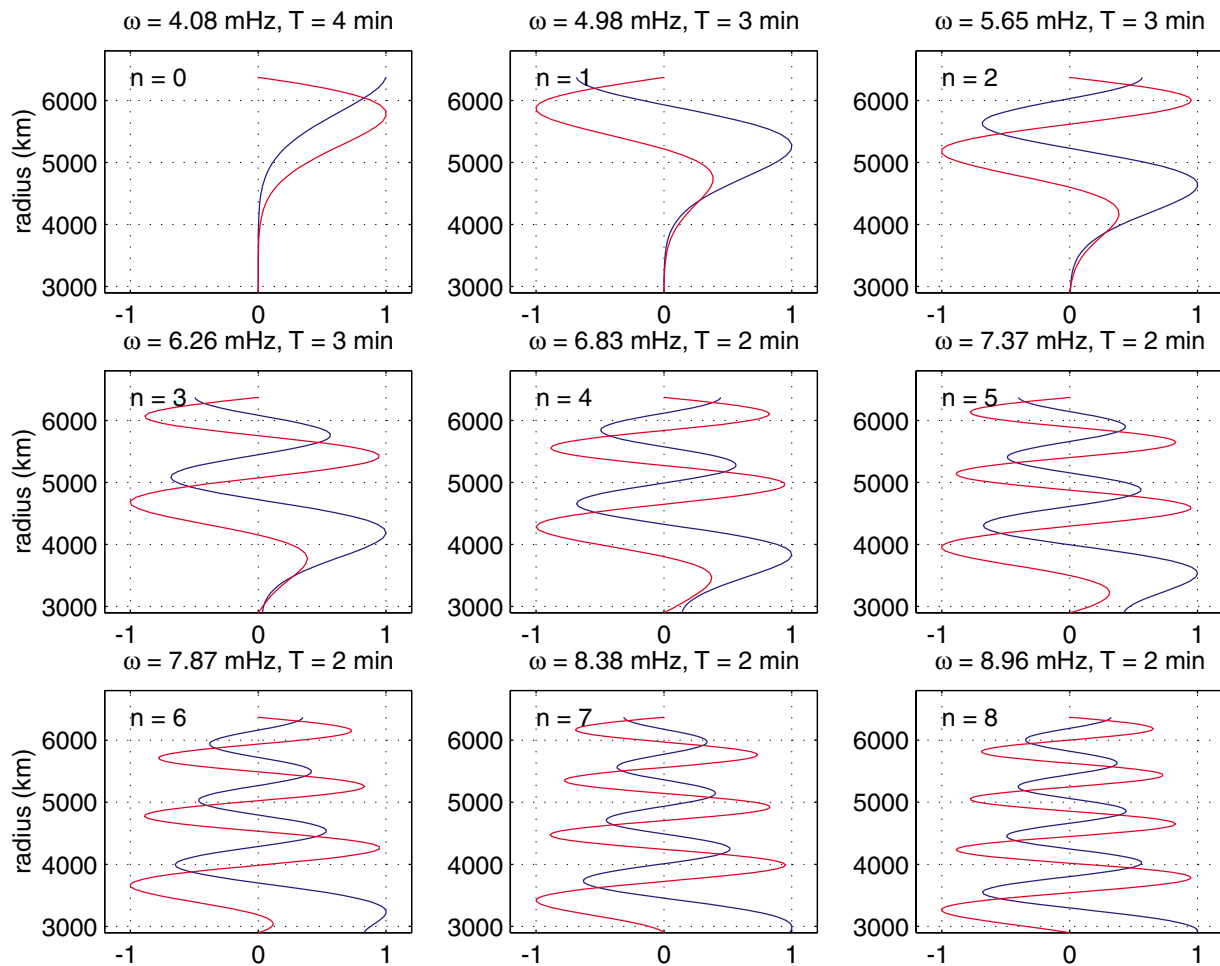


Figure 4. Same as Figure 3, but for $l = 25$.

title. The blue line identifies the displacement eigenfunction, the red line the stress eigenfunction. Note that each of the stress eigenfunctions satisfies the zero-stress boundary conditions at the shell's inner (2891 km) and outer (6371 km) radii. You can count the number of nodes in the each displacement eigenfunction and see the integer increase in nodes associated with each higher mode (this is a consequence of the fact that we are solving a Sturm-Liouville problem - see Dahlen and Tromp, 1999). Table 1 is a list of the values with more precision than shown on the illustrations.

The $l = 2$ case includes the gravest toroidal mode ${}_0T_2$. The uniform shell with mantle-like properties predicts a vibration of 0.3638 mHz, or approximately 2,748 seconds, which is about 45.8 minutes. The value predicted by PREM is about 44.0 minutes, the value observed by Widmer et al., 1992 (GJI) is 44.25 minutes (0.3766 mHz). That means the uniform shell is predicting the period in minutes to within about 3.5%.

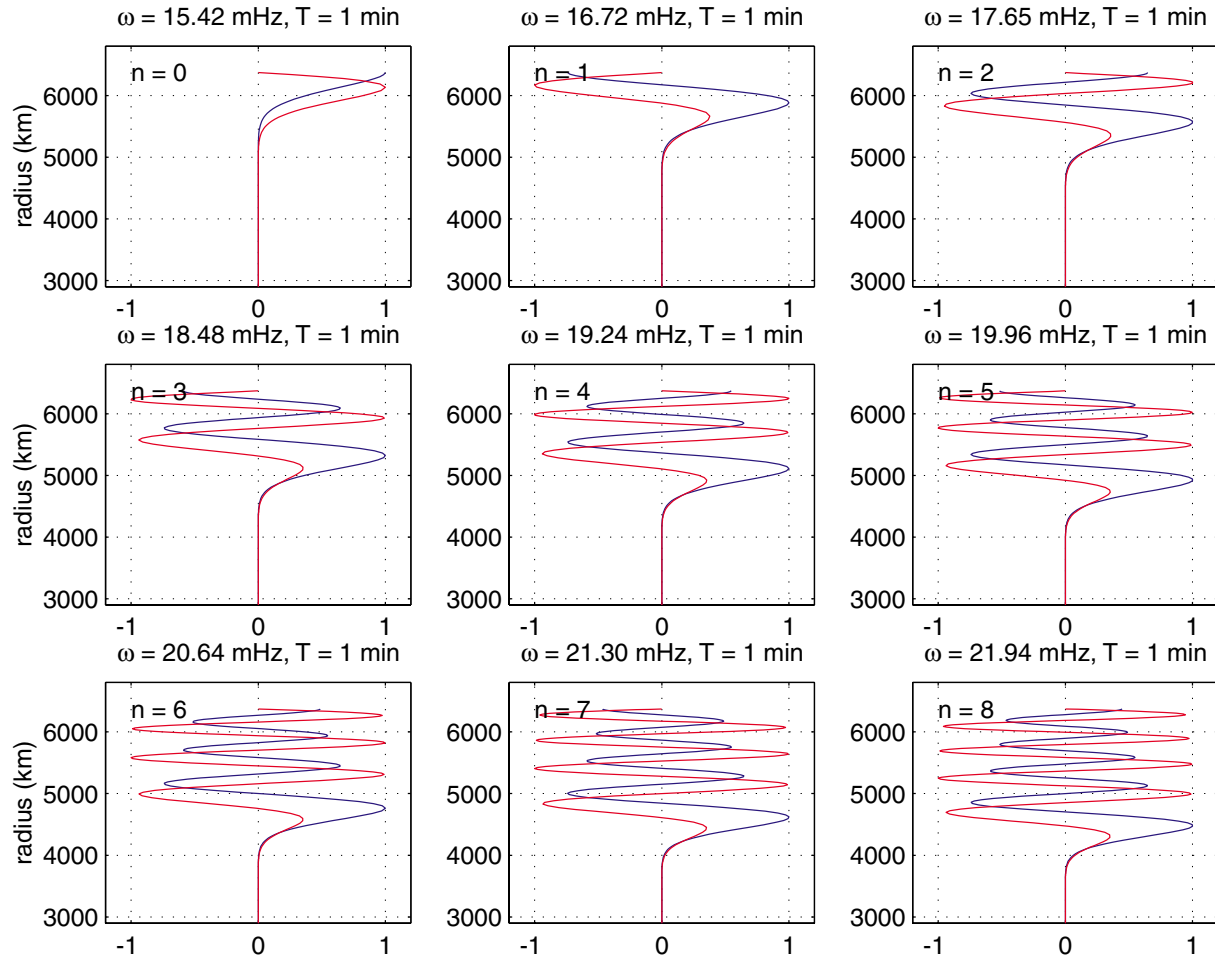


Figure 5. Same as Figure 3, but for $l = 100$. Note the eigenfunction of the fundamental mode is extending much less into the shell.

Table 1: Eigenfrequencies (mHz) of a Uniform Shell Model

l/n	0	1	2	3	4	5	6	7	8
2	0.3638	1.1046	1.8472	2.6560	3.4870	4.3271	5.1719	6.0193	6.8684
25	4.0818	4.9839	5.6526	6.2576	6.8279	7.3684	7.8742	8.3834	8.9570
100	15.4163	16.7158	17.6547	18.4788	19.2384	19.9550	20.6402	21.3015	21.9436

Table 2: Observed Eigenfrequencies (mHz) (Gilbert & Dziewonski, 1975)

l/n	0	1	2	3	4	5	6	7	8
2			2.256		2.384	2.489		2.750	2.913

Ideas for Exploring the Solution

Here are some ideas to help you get you started on ways that you can modify the scripts to explore these solutions or solutions to analogous problems. Once grasp the details of the examples, it's hard not to generate ideas new numerical experiments, so you might not want to stop with these. Just make sure you understand the scripts before you begin modifying.

Exercise (1). Try creating a linear shear-modulus model, $\mu(r) = ar + \mu_0$, with a surface value of 682 kbar and a core-mantle value of 2938 kbar; and $\rho(r) = br + \rho_0$, with a surface value of 2.69 g/cm³ and a core-mantle boundary value of 5.56 g/cm³.

Exercise (2). Try computing the dispersion for a two-layer mantle with an outer shell, $r = 5701$ to 6368 with a shear velocity of 5.04 km/s and a density of 3.57 g/cm³, and an inner shell from 3480 to 5701 with a shear velocity of = 6.81 km/s and density of 4.93 g/cm³. Explore the sensitivity of the eigenfrequencies to variations in shear velocity and density.

Exercise (3). A third-order polynomial fits to the density and shear modulus of PREM's crust and mantle are given in the following equations.

$$\rho(r) \approx -2.84710 \times 10^{-10} r^3 + 3.84976 \times 10^{-6} r^2 - 1.76479 \times 10^{-2} r + 32.4479$$

$$\mu(r) \approx -8.11871 r^3 + 9.56717 \times 10^4 r^2 - 4.250608 \times 10^8 r + 9.578569 \times 10^{11}$$

where $3480 \text{ km} \leq r \leq 6368 \text{ km}$ (the equations are for radius in kilometers). Modify the MATLAB scripts to compute the eigenvalues and eigenvectors for this model and compute the $l = 2$ eigenfrequencies and compare them with those of the uniform shell. Does the fit to ${}_oT_2$ improve? Does the fit to other modes improve (see Table 3)?

Exercise (4). Implement a safeguard for missed roots by counting the zero crossings in the eigenfunction to make sure that we have increased the number of nodes by one.

Exercise (5). Modify the scripts to solve the coupled, first-order differential equations for the spheroidal, radial oscillations of an elastic planetary model (Equations 8.149-8.150 of Dahlen and Tromp, 1999). This is slightly more involved, since you have to start the integration from the center of the planet using the analytic expressions for the solution of a uniform elastic sphere.

Exercise (6). Modify the scripts to solve the coupled, first-order differential equations for the non-radial spheroidal problem (Equations 8.135-8.140 of Dahlen and Tromp, 1999). This is much more involved, since you have to start the integration from the center of the planet using the analytic expressions for the solution of a uniform sphere.

Exercise (7). Set up a script to search for an optimal value of shear modulus and density that match the observed eigenfrequencies of Earth. That is, step through values of shear modulus and

find that which matches Earth's eigenfrequencies best. Some observed values of the fundamental mode toroidal eigenfrequencies are

Table 3: Observed Eigenfrequencies (Widmer et al., 1992)

Mode:	${}_0T_2$	${}_0T_3$	${}_0T_4$	${}_0T_5$	${}_0T_6$
f (μHz):	376.60	587.26	766.73	928.97	1079.51

Minimize the sum of the square misfits between the observed and predicted eigenfrequencies (or eigenperiods) and present your results as a contour plot of misfit as a function of shear modulus and density.

Exercise (8). Solve the analogous problem for Love waves for a flat-layered structure underlain by a half-space. The eigenvalue problem can be written as

$$\frac{d}{dz} \begin{bmatrix} W(z) \\ T(z) \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\mu(z)} \\ -\omega^2 \rho(r) + k^2 \mu(z) & 0 \end{bmatrix} \begin{bmatrix} W(z) \\ T(z) \end{bmatrix}$$

For this problem you must find the appropriate value of k for each frequency, $\omega = 2\pi/T$, where T is the period of interest.

Place the origin ($z = 0$) within the half-space and assume that z is positive above the origin. Then the layer(s) is(are) in the range of positive z values. Start the upward integration from the origin within the underlying elastic half-space. With this choice of coordinates, we have initial conditions,

$$W(z_{bottom}) = 1.0, T(z_{bottom}) = \mu v_{\beta}, \quad (1)$$

where $z_{bottom} = 0$ because of our choice of origin, and

$$v_{\beta} \equiv \sqrt{k^2 - \omega^2 / \beta^2}. \quad (2)$$

Note that you use the value of β appropriate for the half-space in the initial conditions. You integrate the equations starting from $z = 0$ up to the top of the crust. To avoid numerical overflow during the integration, place the origin within the half space, about three times a shear-wavelength in the half-space.

Try a model with a 40 km thick crust ($\beta = 3.45$ km/s, $\rho = 2800$ kg/m³) lying above the mantle ($\beta = 4.60$ km/s, $\rho = 3300$ kg/m³) and compute the eigenfunctions and values of wavenumber, k , for periods of 120, 80, 50, 25, and 10 seconds. Think about an appropriate range of k for the root search; you can place limits on the values it will have.

Exercise (9). Compare the fundamental mode eigenfrequency predictions of a spherical Earth model with those of a flat-lying model with Earth flattening. See Box 9.9 of Aki and Richards (1980) for a discussion of Earth flattening transformations.

References

Aki, K., and P. G. Richards, *Quantitative Seismology: Theory and Methods, vol. 1 and 2*, W.H. Freeman, San Francisco, pages, 1980.

Dahlen, F. A., and J. Tromp, *Theoretical Global Seismology*, Princeton University Press, Princeton, NJ, 1025 pages, 1999.

Gilbert, F. and A. M. Dziewonski, An application of normal mode theory to the retrieval of structural parameters and source mechanisms from seismic spectra, *Phil. Trans. Royal Soc. London, Mathematical and Physical Sciences*, 278, 187-269, 1975.

Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, New York, NY, 735 pages, 1988.

Widmer, R., W. Zurn, and G. Masters, Observation of low-order toroidal modes from the 1989 Macquarie Rise event, *Geophys. J. Int.*, 111, 226-236, 1992.