Problem Set 5: Toroidal modes of a spherically symmetric earth[1]

GEOS 626: Applied Seismology, Carl Tape

Assigned: February 22, 2016 — Due: February 29, 2016

Last compiled: October 16, 2017

Overview and instructions

- This problem set provides a hands-on introduction to how normal modes of a particular earth model can be computed. The goal is to gain a better understanding for the concepts of normal modes, eigenfrequencies, eigenfunctions, and dispersion.
- The necessary starting Matlab scripts and data are available from `seis2016` directory on the linux network; type `git pull` to obtain the latest version of files. The relevant files are:
  - `spshell_template.m` (Section 2.1)
  - `surf_stress.m` (Section 2.2)
  - `stress_disp_tor.m` (Section 2.3)
- Background reading:

  Stein and Wysession (2003, Section 2.9), *Dahlen and Tromp* (1998, Ch. 8)

  Note the differences in nomenclature listed in Table 1.
- See Table 2 for a summary table of wave parameters.
- The Preliminary Reference Earth Model is from *Dziewonski and Anderson* (1981). It descibes the Earth's structure as a set of 1D radial functions (e.g., $\rho(r)$).

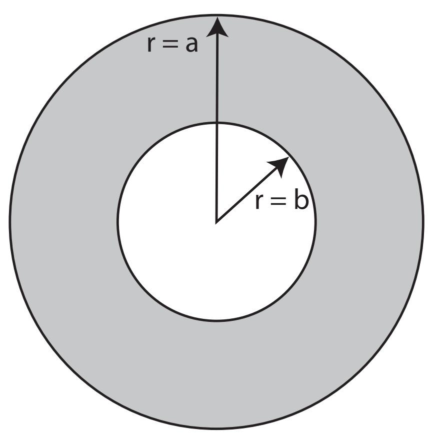

Figure 1: Cross section of the spherical shell model assumed in this problem. Since the outer core is fluid, a shell is representative of earth for toroidal vibrations. We assume this shell can have radial variations in density, $\rho(r)$, and rigidity, $\mu(r)$.

[1] This problem set was designed and written by Charles Ammon, Penn State University. I have made some modifications.

Table 1: Differences in nomenclature between *Dahlen and Tromp* (1998) (DT) and *Stein and Wysession* (2003) (SW).

symbol	range	DT label	SW label
n	$0 \leq n \leq \infty$	overtone number	radial order
l	$0 \leq l \leq \infty$	(angular) degree	angular order
m	$-l \leq m \leq l$	(azimuthal) order	azimuthal order
${}_n\omega_l{}^m$	$0 < {}_n\omega_l{}^m \leq \infty$	eigenfrequency for (n, l, m)	eigenfrequency for (n, l, m)

Table 2: Wave parameters and some related equations. Note that $\lim_{l\to\infty} \sqrt{l(l+1)} = l + \frac{1}{2}$, so in the high-frequency limit, $l + \frac{1}{2}$ is appropriate. We have listed "rad" for radians, which have no units; however, it is helpful to think of angles for these quantities.

earth radius earth circumference	a $2\pi a$	6.371×10^6 m 4.003×10^7 m		
degree	l			
period	T	$T = \frac{1}{f} = \frac{2\pi}{\omega} = \frac{\lambda}{c} = \frac{2\pi}{ck}$	$T = \frac{2\pi}{c\sqrt{l(l+1)}}$ [s]	$T = \frac{2\pi a}{c'\sqrt{l(l+1)}}$ [s]
frequency	f	$f = \frac{\omega}{2\pi} = \frac{1}{T} = \frac{c}{\lambda} = \frac{ck}{2\pi}$	$f = \frac{c\sqrt{l(l+1)}}{2\pi}$ [1/s]	$f = \frac{c'\sqrt{l(l+1)}}{2\pi a}$ [1/s]
angular frequency	ω	$\omega = 2\pi f = \frac{2\pi}{T} = ck = \frac{2\pi c}{\lambda}$	$\omega = c\sqrt{l(l+1)}$ [rad/s]	$\omega = \frac{c'\sqrt{l(l+1)}}{a}$ [rad/s]
wavelength	λ	$\lambda = \frac{c}{f} = cT = \frac{2\pi}{k} = \frac{2\pi c}{\omega}$	$\lambda = \frac{2\pi}{\sqrt{l(l+1)}}$ [rad]	$\lambda' = \frac{2\pi a}{\sqrt{l(l+1)}} = \lambda\, a$ [m]
wavenumber	k	$k = \frac{\omega}{c} = \frac{2\pi}{\lambda} = \frac{2\pi f}{c} = \frac{2\pi}{Tc}$	$k = \sqrt{l(l+1)}$ []	$k' = \frac{\sqrt{l(l+1)}}{a} = k/a$ [1/m]
phase speed	c	$c = \frac{\omega}{k} = \frac{\lambda}{T} = f\lambda$	$c = \frac{2\pi}{T\sqrt{l(l+1)}}$ [rad/s]	$c' = \frac{2\pi a}{T\sqrt{l(l+1)}} = c\, a$ [m/s]

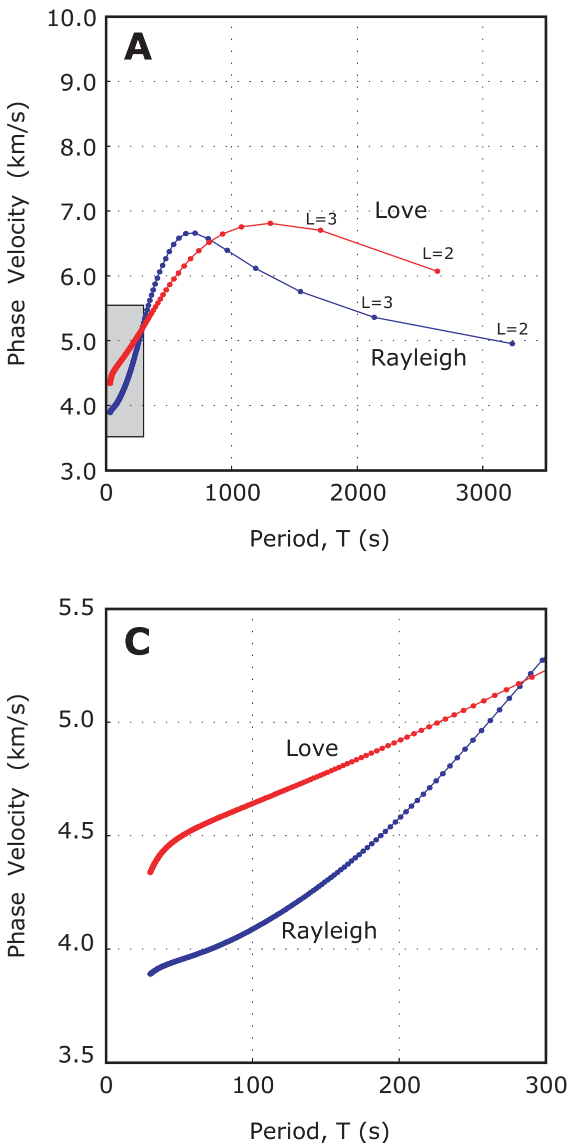

Figure 2: Dispersion of surface waves, plotted as phase velocity $c(T)$ and group velocity $U(T)$. The curves are calculated using normal-modes summation with the 1D earth model PREM (*Dziewonski and Anderson*, 1981); each point corresponds to a normal mode with harmonic degree l. A couple points are labeled in A. These curves are for the fundamental spheroidal (Rayleigh) and toroidal (Love) modes ($n = 0$); dispersion curves of overtone branches ($n \geq 1$) can be seen in Figures 11.5 and 11.6 of *Dahlen and Tromp* (1998). **A.** Phase velocity dispersion $c(T)$. Gray box shows region of C. **B.** Group velocity dispersion $U(T)$. Gray box shows region of D. **C-D.** Same as A-B, only with different axes ranges. With periods $T < 300$s, Love waves travel faster than Rayleigh waves. Calculations courtesy of Ana Ferreira.

1 Background

With the power of computing, 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. 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. 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.

1.1 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 first-order coupled system of equations (see Section 1.3 for details)

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

- $W(r)$ is the displacement
- $T(r)$ is the stress
- ω is the angular frequency
- $\rho(r)$ is the density
- $\mu(r)$ is the shear modulus
- l is the angular order

For low frequencies and low angular orders, l, these equations can be integrated numerically using a Runge-Kutta approach (e.g., *Press et al.*, 1988).

Before you integrate Equation (1), you must specify values of l, the density, the shear modulus, and the frequency ω. Integration produces two functions of radius, $\hat{W}(r)$ and $\hat{T}(r)$. If the stress is zero at the top of the shell (earth surface) and the bottom of the shell (core-mantle boundary),

$$\hat{T}(a) = 0 \tag{2}$$
$$\hat{T}(b) = 0, \tag{3}$$

then the assumed value of ω is an **eigenfrequency** (ω^2 is an **eigenvalue**) and $\hat{W}(r)$ and $\hat{T}(r)$ are **eigenfunctions**. (See note [2].) 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 ($r = b$), then 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

$$F({}_n\omega_l) = 0. \tag{4}$$

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 n. Finding solutions that have zero surface stress is a numerical root-finding problem: you assume a value of and integrate the equations to see if it produces zero stress at the outer surface.

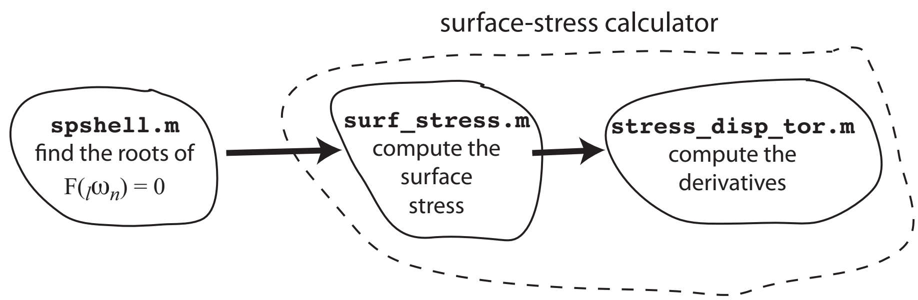

Figure 3: Solving the eigenvalue problem for the vibrating shell is accomplished using three scripts. The main purpose of each script is described in the above chart, which also shows the dependencies: `spshell_template.m` calls `surf_stress.m`, which requires `stress_disp_tor.m`.

1.2 The Matlab Solution

Our solution will involve three different Matlab scripts (Figure 3); two are needed to solve the equations of motion (Eq. 1), and the third is used to find the values of that are eigenfrequencies. The first two scripts 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.

[2]These are not eigenvalues and eigenfunctions of Equation (1) but rather of (*Dahlen and Tromp*, 1998, Eq. 4.8-4.9)

$$\mathcal{H}\mathbf{s} = \omega^2\mathbf{s}.$$

where $\mathcal{H}$ is a linear operator with the eigensolution ω^2 (eigenvalue) and $\mathbf{s}$ (eigenfunction). It is best to think of *solution pairs* of ${}_n\omega_l$ associated with functions ${}_nW_l(r)$ and ${}_nT_l(r)$.

1.2.1 Computing the Surface Stress

To integrate the equations and evaluate the surface traction I created two scripts. The first script (Section 2.3) computes the stress and displacement derivatives using the differential equations (Eq. 1). The input arguments and return values of this script are of the special form required by the Runge-Kutta functions in Matlab, and the name of this script 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 and stress). 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.

The second script (Section 2.2) calculates the surface stress. For a given value of ω, the `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 $r = a = 6371$ km), which after execution is stored in the last element of array `sd`.

1.2.2 Finding the Roots

The third and final script (Section 2.1) 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 (Δf), 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 weve 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 Δf, 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 script 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.

1.3 More from *Dahlen and Tromp* (1998) on Equation (1)

A derivation of Equation (1) is presented in *Dahlen and Tromp* (1998); our Equation (1) corresponds to their equations 8.114 and 8.115. As stated in (*Dahlen and Tromp*, 1998, p. 270), the traction, $\mathbf{t}$, exerted on any spherical surface is given in terms of the displacment scalars U, V, and W by

$$\mathbf{t} = \hat{\mathbf{r}} \cdot \mathbf{T} = R\,\mathbf{P}_{lm} + S\,\mathbf{B}_{lm} + T\,\mathbf{C}_{lm}, \tag{5}$$

where $\mathbf{T}$ is the stress tensor and

$$R(r) = (\kappa + \tfrac{4}{3}\mu)\frac{dU}{dr} + (\kappa - \tfrac{2}{3}\mu)r^{-1}(2U(r) - kV(r)) \tag{6}$$

$$S(r) = \mu\left(\frac{dV}{dr} - r^{-1}V(r) + kr^{-1}U(r)\right) \tag{7}$$

$$T(r) = \mu\left(\frac{dW}{dr} - r^{-1}W\right) \tag{8}$$

where $k = \sqrt{l(l+1)}$ (Table 2).

We see that the radial stress function, $T(r)$ is defined in terms of the radial displacement function, $W(r)$. For toroidal motion, as in our problem, only $W(r)$ is non-zero while $U(r) = V(r) = 0$ (and therefore $R(r) = S(r) = 0$).

References

Dahlen, F. A., and J. Tromp (1998), *Theoretical Global Seismology*, Princeton U. Press, Princeton, New Jersey, USA.

Dziewonski, A., and D. Anderson (1981), Preliminary reference Earth model, *Phys. Earth Planet. Inter.*, *25*, 297–356.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1988), *Numerical Recipes in C*, Cambridge U. Press, New York.

Stein, S., and M. Wysession (2003), *An Introduction to Seismology, Earthquakes, and Earth Structure*, Blackwell, Malden, Mass., USA.

2 Three Matlab scripts

2.1 `spshell_template.m`

```
close all; clear all; clc

% global variables
% WARNING: DO NOT CHANGE THE DIMENSION OF ANY OF THESE VARIABLES
global l rvec WT rspan imod rho mu % omega

%-------------------------------------------------
% USER INPUT

earthr = 6371000;       % radius of earth, in meters (a)
cmbr = 3480000;         % radius of core-mantle boundary, in meters (b)
rspan = [cmbr earthr];  % [b a]

% Earth model (uniform mantle shell)
imod = 0;               % index for Earth model (see earthfun.m)
                        % =0 for homogeneous
                        % =1 for linear rho(r) and mu(r)
                        % =2 for cubic rho(r) and mu(r)
% WARNING: BECAUSE THESE ARE GLOBAL VARIABLES, THESE VALUES MAY BE
% OVER-RIDDEN BY VALUES DEFINED IN OTHER FUNCTIONS.
```

```
rho = 4380;             % density
mu  = 5930*5930*rho;    % rigidity (mu = 1.54e11 Pa)

% options for searching solution space
l = 2;                  % degree (l >= 1)
nmax = 8;               % maximum n (default = 8)
                        % nmax+1 is the max number of roots/eigenfunctions/subplots
                        % nmax=0 will return the first root (n=0)
% plotting options
iplot_eig_freqs = 1;    % plot eigenfunctions (=1) or not (=0)
iplot_all_freqs = 0;

% path to the directory containing the data file prem_Tmodes.txt
ddir = './data/';

iprint = 0; % print figures to file (=1) or not (=0)
pdir = './';

%-----------------------------------------------

% range of frequencies (note: omega = 2*pi*f), in Hz
fmin = 1/3600;      % initial frequency to start (T = one hour)
df = 0.0002;        % frequency step size (chosen by trial and error)
fmax = 0.08;        % stopping frequency (somewhat arbitrary)
%fmax = 0.003;
fvec = [fmin:df:fmax];
numf = length(fvec);
disp(sprintf('frequency vector ranges from %.3f mHz to %.3f mHz',fmin*1e3,fmax*1e3));
disp(sprintf('num frequency points is %i, df = %.3f mHz',numf,df*1e3));
disp(sprintf('--> period ranges from %.2f min to %.2f min',1/fmin/60,1/fmax/60));

% THIS BLOCK IS FOR INITIAL PLOTTING ONLY
if iplot_all_freqs==1
    for ii=1:numf
        disp(sprintf('%2i/%2i: f = %.3f mHz',ii,numf,fvec(ii)*1e3));
        % update W(r) and T(r), stored within WT
        surf_stress(fvec(ii));

        % plotting parameters
        rplot = rvec/1000;
        xmx = 1.1; ymn = rspan(1)/1000; ymx = rspan(2)/1000; dy = 100;

        % displacement for each frequency
        Wplot = WT(:,1)/max(abs(WT(:,1)));
        figure(2); hold on; plot(Wplot,rplot,'b');
        text(Wplot(end),rplot(end)+dy/2,num2str(ii));
        plot([0 0],rspan/1000,'k','linewidth',2);
        xlabel('normalized displacement, W(r)'); ylabel('radius, km');
        axis([-xmx xmx ymn-dy ymx+dy]);

        % stress for each frequency
        Tplot = WT(:,2)/max(abs(WT(:,2)));
        figure(3); hold on; plot(Tplot,rplot,'r');
        text(Tplot(end),rplot(end)+dy/2,num2str(ii));
        plot([0 0],rspan/1000,'k','linewidth',2);
        xlabel('normalized stress, T(r)'); ylabel('radius, km');
        axis([-xmx xmx ymn-dy ymx+dy]);
```

```
    end

    % print figures for HW
    if iprint==1
        figure(2); print(gcf,'-depsc',[pdir 'modes_Wr']);
        figure(3); print(gcf,'-depsc',[pdir 'modes_Tr']);
    end

    % exit
    break
end

% initial freqeuncy and corresponding surface stress
% NOTE: surf_stress.m calls stress_disp_tor.m, which depends on degree l
f = fvec(1);
Tsurf = surf_stress(f);
n = 0;             % counter for n (n=0 is the first root)

% leave gap for T(n=0,l=1), which do not exist
% note: these are useful when looping over degree l
if and(l==1,nmax>0), n = 1; end        % fill the n >= 1 entries
if and(l==1,nmax==0), continue; end     % exit loop early (mode 0T1 does not exist)

% THIS IS THE KEY LOOP OVER FREQUENCIES
froots = NaN*ones(nmax+1,1);
for ii = 2:numf-1
    % frequency interval over which we check for a root
    oldf = f;
    f = fvec(ii);

    % The function surf_stress.m will updated the key variable WT,
    % which contains the radial displacement W(r) in the first column
    % and stress T(r) in the second column.
    Tsurfold = Tsurf;          % surface stress for previous f
    Tsurf = surf_stress(f);    % surface stress for new f

    disp(sprintf('%3i %10.3e %10.3e %.2f mHz %.1f s %.2f min', ...
        ii, Tsurfold, Tsurf, f*1e3, 1/f, 1/f/60));

    % Check 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, call the matlab function fzero to refine the root.
    % Then store the root in the vector froots and plot the results.
    if (Tsurfold * Tsurf < 0)
        f0 = fzero('surf_stress',[oldf f]);
        froots(n+1) = f0;

        % update eigenfunctions (WT, rvec) for the exact frequency
        surf_stress(f0);
        disp(sprintf('n=%i %.3f mHz l=%i', n,f0*1e3,l));

        % plotting eigenfunctions (displacement and stress as a function of radius)
        if iplot_eig_freqs==1
            xmx = 1.2; ymn = rspan(1)/1000; ymx = rspan(2)/1000;
            rplot = rvec/1000;
            Wplot = WT(:,1)/max(abs(WT(:,1)));
            Tplot = WT(:,2)/max(abs(WT(:,2)));
```

```
            figure(1); if nmax==0, subplot(1,1,n+1); else subplot(3,3,n+1); end
            hold on;
            plot(Wplot,rplot,'b');    % W(r), displacement (blue)
            plot(Tplot,rplot,'r');    % T(r), stress (red)
            plot([-xmx xmx],ymn*[1 1],'k',[-xmx xmx],ymx*[1 1],'k',[0 0],[ymn ymx],'k');
            axis([-xmx xmx ymn-300 ymx+300]); %grid on;
            title(sprintf('f = %.2f mHz, T = %.2f min (l = %i)',froots(n+1)*1000,1/f0/60,l));
            text(-1,ymx+100,sprintf('n = %i',n));
            if mod(n-1,3)==0, ylabel('radius (km)'); end
        end

        % exit the loop when you reach n=nmax
        if n==nmax, break; end
        % count for the next root
        n = n + 1;
    end
end
fprintf('/// l = %i, nroots = %i (nmax = %i, fmax = %.3f mHz)\n',...
    l,sum(~isnan(froots)),nmax,fmax*1e3);
```

2.2 surf_stress.m

```
function value = surf_stress(f)

global omega rvec WT rspan % mu rho l

WT0 = [1.0 0.0];        % the initial values of [displacement stress]

omega = 2*pi*f;         % angular frequency (stress_disp_tor.m)

% integration by Matlab, calling our function stress_disp_tor.m to
% calculate derivatives (d/dr) of displacement and stress;
% on return the vectors rvec and WT contain the values of radius
% and the displacement and stress eigenfunctions
% note: the dimension of rvec and WT is the number of points needed for
%       the numerical integration -- this will vary

% try this option to use default tolerance on numerical solution
[rvec,WT] = ode45('stress_disp_tor',rspan,WT0);

% try this option if you need really accurate eigenfunctions
%opts = odeset('reltol',1e-12,'abstol',1e-12);
%[rvec,WT] = ode45('stress_disp_tor',rspan,WT0,opts);

value = WT(end,2);      % stress value at earth's surface (at r = rspan(2))
```

2.3 stress_disp_tor.m

```
function dWT = stress_disp_tor(r,WT)

global omega l imod rho mu % rspan rvec WT

% The input values of WT(1) and WT(2) are W(r) and T(r) respectively.
% The returned deriatives are stored in dWT

% structural values at radius r: density and rigidity
% note: if imod=0, then the program will use the rho and mu from spshell.m
if imod~=0, [rho,mu] = earthfun(r); end

% displacement (first row of equation 1)
dWT(1,1) = WT(1) / r + WT(2) / mu;

% stress (second row of equation 2)
dWT(2,1) = ((l-1)*(l+2)*mu/(r*r) - rho*omega*omega)*WT(1) - 3*WT(2)/r;
```

2.4 earthfun.m

This is a fourth Matlab script—a template that you can adapt in order to implement a simple (linear or cubic) profile of $\rho(r)$ and $\mu(r)$, in stead of a homogeneous model of ρ and μ.

```
function [rho,mu] = earthfun(r)
%EARTHFUN return a rho and mu value for a specified radius r
%
% called by stress_disp_tor.m

global rspan imod

switch imod
    case 1
        % linear model
        % ENTER YOUR CODE HERE
        cmbr = rspan(1);    % b
        earthr = rspan(2);  % a

        error('earthfun.m imod=1 not yet implemented');

    case 2
        % cubic model
        % ENTER YOUR CODE HERE

        error('earthfun.m imod=2 not yet implemented');

    otherwise
        error('invalid imod (=1,2)');
end
```

Problem 1 (4.0). Eigenfunctions, eigenfrequencies, and dispersion

- A key point is to recognize that "the mode" ${}_n\mathsf{T}_l$ has three parts:

$$
{}_n\mathsf{T}_l \quad \begin{cases} {}_n\omega_l & \text{eigenfrequency} \\ {}_nW_l(r) & \text{radial eigenfunction (related to } T(r)) \\ {}_nY_l(\phi,\theta) & \text{surface spherical harmonic function} \end{cases} \tag{9}
$$

  Our problem is not concerned with the surface displacement field[3]. So "the mode" or "the solution" ${}_n\mathsf{T}_l$ (to Eq. 1) has three parts: an eigenfrequency ${}_n\omega_l$ and two eigenfunctions, W and T, with the boundary condition $T(a) = T(b) = 0$:

$$
{}_n\mathsf{T}_l \quad \begin{cases} {}_n\omega_l \\ {}_nW_l(r) \\ {}_nT_l(r) \end{cases} \tag{10}
$$

- In this problem we will assume a homogeneous earth model with density $\rho = 4380$ kg/m^3 and shear modulus (rigidity) $\mu = 1.54 \times 10^{11}$ Pa. Before you proceed, make sure you follow the basic structure of the program; in particular, pay attention to the use of **global variables** within each of the scripts.

1. (0.3)

   (a) Write down the two equations represented by Equation (1) and the equations for the boundary conditions; show explicit r dependence, l dependence, and n dependence (e.g., ${}_n\omega_l$).

   (b) List and name the variables that describe the Earth structure?

   (c) What is the physical meaning of $W(r)$ and $T(r)$?

2. (0.0) Execute the script `spshell_template.m` and analyze the output. Two figures are generated, one for a set of $W(r)$, the other for a set of $T(r)$ (Figure 4).

   (a) What is the range of frequencies that you are evaluating? List $f_{\min}$ and $f_{\max}$ in mHz. Draw two lines for $f = f_{\min}$ and $f = f_{\max}$ in Figure 5a.

   (b) A key part of the problem is to identify the frequency intervals, $[f_i, f_{i+1}]$ where the surface stress, $T(a)$, changes sign. On Figure 4, highlight the pair(s) (i and $i+1$) that have $T(a)$ on opposite sides of $T(a) = 0$.

   (c) How many crossings of $T(a) = 0$ are there within this set of 14 input frequencies?

   (d) The script is looping over a set of (l_i, f_i) values. In Figure 5a, plot (by hand) 14 dots that represent your values.

3. (0.0) Now set `iplot_all_freqs = 0` and rerun.

   (a) How many solutions to Equation (1) do you obtain over this range of frequencies?

   (b) Describe the key (new) step used in obtaining these solutions.

[3] The surface displacent field is a vector field that is a function of ${}_nY_l(\phi,\theta)$; see *Dahlen and Tromp* (1998, Section 8.6.1).

(c) What is the value of n for the first root that is encountered?

4. (0.7) Now set `fmax = 0.08` and rerun.

   (a) (0.1) What aspects of the curves tells you which color curve is ${}_nT_l(r)$?

   (b) (0.1) What is the relationship between the number of zero crossings of ${}_nW_l(r)$ (excluding endpoints) and n?

   (c) (0.1) What is the relationship between the number of zero crossings of ${}_nT_l(r)$ (excluding endpoints) and n?

   (d) (0.2) What is the qualitative relationship between n and ${}_nf_l$ (l fixed)? (Note [4])

   (e) (0.2) Try $l = 25$ and $l = 100$ (n=0-8).
   What is the qualitative relationship between l and ${}_nf_l$ (n fixed)?

5. (0.5) Modify your code to compute eigenfrequencies for $n = 0$, $l = 2$-9.

   (a) (0.3) In Figure 5b, plot (by hand) *all* the function evaluations (l, f) that you performed. Hint: what is f_{min}?

   (b) (0.1) List the predicted eigenperiods in Table 4.

   (c) (0.1) Among the toroidal modes in Table 4, the gravest mode is ${}_0\mathsf{T}_2$ ($n = 0$, $l = 2$).

   - List the eigenfrequency ${}_0f_2$ in mHz and eigenperiod ${}_0T_2$ in seconds and minutes. (Note [5])
   - Compare with the observed value in Table 3. What is the percent difference, $100 \ln(T/T_{\text{obs}})$, where T is your computed eigenperiod for ${}_0\mathsf{T}_2$?

6. (0.5) Generate a plot containing the two eigenfunctions ${}_nW_l(r)$ and ${}_nT_l(r)$ for $n = 0$ and $l = 40$.

   (a) (0.3) Compute various wave parameters and list your values in the left half of Table 5.

   (b) (0.1) Include your plot.

   (c) (0.1) mark the wavelength, λ, as a distance from the surface.

7. (1.5) A dispersion plot based on PREM is shown in Figure 2.9-10 of *Stein and Wysession* (2003); the inset figure is relevant to this problem.

   (a) (1.2) Reproduce a similar plot for your homogeneous earth model. Use l=1–10 and plot with axes ranges `[0 11 0 4]`, where the y-axis is frequency in mHz and the x-axis is l.

   Note: Some points $(l,\ {}_nf_l)$ will be outside the plotting range.

   (b) (0.3) On your dispersion diagram, plot the observations in Table 3 with a different symbol (or color) from the predictions. (Make sure you plot all predicted frequencies, not just the ones with matching observations.)

   Note: See code at the bottom of `spshell_template.m` to read in the observations.

[4] The eigenfrequencies are given by ${}_n\omega_l = 2\pi({}_nf_l)$. However the values in the program—and listed in dispersion plots, such as those in *Stein and Wysession* (2003)—are ${}_nf_l$, a quantity that is somewhat more intuitive.

[5] Notation might be confusing: T is period, $T(r)$ is the stress eigenfunction (for toroidal modes), and ${}_n\mathsf{T}_l$ is the label for a toroidal mode. Usually it should be clear from the context.

8. (0.5) Consider the (normalized) L2-norm misfit between observed and predicted frequencies:

$$F(\mathbf{m}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[f_i^{\mathrm{obs}} - f_i(\mathbf{m})\right]^2} \tag{11}$$

where i is the measurement index that represents some particular n-l pair, $f_i(\mathbf{m})$ is the predicted mode frequency, f_i^{obs} is the observed mode frequency, and $\mathbf{m}$ is the model representing $\mu(r)$ and $\rho(r)$.

(a) (0.1) Write the expression for $F(\mathbf{m})$ in the case of $N = 1$.

(b) (0.4) Compute $F(\mathbf{m})$ for all available observations (Table 3). These observations will be loaded within `spshell_template.m` if you comment out the last `break` statement. Tip: Initialize an $a \times b$ matrix as `Fobs = NaN(a,b)`, then fill certain entries according to n and l listed in the observations. This "observations matrix" can be differenced with a corresponding "predictions matrix"; the residual matrix can be turned into a vector as `Fres(:)`. See example code at the bottom of `spshell_template.m`.

- Show your code for the calculation.
- List the value in mHz in Table 6.

Problem 2 (3.0). Changing the earth model

1. (1.5) **Homogeneous earth.** We want to explore how different homogeneous earth models fit the toroidal mode observations. We will assume a fixed value of density ($\rho = 4380$ kg/m^3), and only look for different values of shear modulus over the range $\mu = [1.2 \times 10^{11},\ 1.9 \times 10^{11}]$ Pa (hint: use a `for` loop). It is best to think of our model vector as $\mathbf{m} = (\rho, \mu)$.

   Hint: For this misfit analysis, note that you only have observations up to $n = 3$ (Table 3). Consider adjusting your limits for root-finding in order to save some time.

   (a) (1.0)

   - Find a value of μ that best fits all $N = 20$ observed eigenfrequencies.
   - List the value of $F(\mathbf{m})$ (Eq. 11) for the best-fitting μ.
   - Justify your choice of the new μ, either in words, Matlab output, or a figure.
   - Show a dispersion plot as before (use `axis([0 11 0 4])`) that contains all predictions and observations.

   (b) (0.2) Repeat, but now use only the $N = 9$ observations for the fundamental mode ($n = 0$). (This means that some of the observations in your plot will not have been used in determining the new μ.) **Show all predictions and all observations in your plot, not just the ones used in calculating the misfit.**

   (c) (0.3) Repeat, but now use only the observed eigenfrequency for ${}_0\mathsf{T}_2$. (This means that all but one of the observations in your plot will not have been used in determining the new μ.) **Show all predictions and all observations in your plot, not just the ones used in calculating the misfit.**
   What is the percent difference from the observed value, $100 \ln(f/f_{\mathrm{obs}})$, for ${}_0\mathsf{T}_2$?

(d) (0.0) Summarize your tests in Table 6.

2. (1.0) **Linear earth.** Adapt `earthfun.m` (Section 2.4) to create a function that inputs a radial value r (or a vector of r values) and outputs $\rho(r)$ and $\mu(r)$ described by a linear model with the "endpoint" values

$$\begin{aligned}
\rho(a) &= 2690\ \mathrm{kg/m^3} \\
\mu(a) &= 0.682 \times 10^{11}\ \mathrm{Pa} \\
\rho(b) &= 5560\ \mathrm{kg/m^3} \\
\mu(b) &= 2.938 \times 10^{11}\ \mathrm{Pa}
\end{aligned}$$

where $r = a$ is the surface and $r = b$ is the core-mantle boundary. (The values are an approximation to PREM's $\rho(r)$ and $\mu(r)$.) Thus, our Earth model $\mathbf{m}$ can be thought of as having four parameters, the intercept and slope of $\rho(r)$ and $\mu(r)$: $\mathbf{m} = (\rho_0, \rho_1, \mu_0, \mu_1)$.

Test your `earthfun.m` for a few values of r to make sure that the output values of ρ and μ are sensible (no work needed).

Note: As suggested in `earthfun.m`, it may be helpful to use `rspan` as a global variable.

(a) (0.4) Show your code for `earthfun.m`.

(b) (0.3) Generate a new dispersion plot with both predictions and observations, as in Problem 1-7.

(c) (0.1) List the misfit in Table 6.

(d) (0.2) What parameters would you search over if you wanted to obtain an optimal linear earth model?

3. (0.5) **Cubic earth.** Instead of using linear functions for $\rho(r)$ and $\mu(r)$, use the following cubic functions:

$$\begin{aligned}
\rho(r) &= (-2.84710 \times 10^{-16})r^3 + (3.84976 \times 10^{-9})r^2 - (1.76479 \times 10^{-2})r + 3.24479 \times 10^4 \\
\mu(r) &= (-8.11871 \times 10^{-9})r^3 + (9.56717 \times 10^{-2})r^2 - (4.250608 \times 10^5)r + 9.578569 \times 10^{11}
\end{aligned}$$

This cubic function is a fit to the PREM profiles, so check that the numbers are sensible before moving on (no work needed).

(a) (0.2) Show your matlab code for the cubic function.

(b) (0.2) Generate a new dispersion plot with both predictions and observations, as in Problem 1-7.

(c) (0.1) List the misfit in Table 6.

Problem 3 (3.0). Phase speed and group speed

For this problem we use the **cubic model** and we consider the fundamental mode ($n = 0$) only. In other words, ${}_nf_l$ for $n > 0$ will not be needed.

1. (0.5) Consider the set of l, `lvec = [10:30:400]`. Using the expressions in Table 2, make two dispersion plots for the fundamental mode:
   (a) (0.2) frequency (mHz) vs degree
   (b) (0.3) phase speed (km/s) vs period; use the x (period) limits `[0 300]`
   Qualitatively, how do these compare with the predictions for PREM, shown in Figure 2c?

2. (0.3)
   (a) Repeat Problem 1-6 for ${}_0\mathsf{T}_{40}$ using the cubic model. List your values in Table 5.
   (b) How has the cubic model affected the depth sensitivity, $W(r)$, of ${}_0\mathsf{T}_{40}$?
   (c) How has it affected the wavelength of ${}_0\mathsf{T}_{40}$?

3. (0.0) In the next two parts, you will need to use discretized versions of $\rho(r)$ and $\mu(r)$. This discretization is already done within `surf_stress.m`. When you plot $W(r)$ and $T(r)$, you are using discretized versions of these functions, where `rvec` is the discretized radius values associated with the eigenfunctions.

   Use the discretized radius, `rvec`, to compute discretized versions of $\mu(r)$ and $\rho(r)$. This can be done with the command

   ```
   [rhovec,muvec] = earthfun(rvec)
   ```

   if you have written `earthfun` to handle either an input vector or an input scalar.

4. (1.0) Compute the group speed of ${}_0\mathsf{T}_{40}$ using (*Dahlen and Tromp*, 1998, Eq. 11.67):

$$U = \frac{I_2}{cI_1} \tag{12}$$

$$I_1 = \int_0^a \rho(r)\,[W(r)]^2\,r^2\,\mathrm{d}r \tag{13}$$

$$I_2 = \int_0^a \mu(r)\,[W(r)]^2\,\mathrm{d}r \tag{14}$$

   where c is phase speed (Table 2). The phase speed, c, and the group speed, U, have units of rad/s. Multiplication by a (earth radius) will give the group speed in km/s ($U' = Ua$).

   (a) Show you lines of code.
   Hint: For an adequate, crude numerical integration, use the command `diff(rvec)` to get a discretized vector of dr that can be multiplied with the discretized version of $W(r)$. (Integration is summation.)
   (b) List both U and U' in Table 5.

5. (1.0) Show numerically that each of the equations in Eq. 1 is satisfied for ${}_0\mathsf{T}_{40}$: ${}_0W_{40}(r)$, ${}_0T_{40}(r)$, ${}_0\omega_{40}$.

   Notes:

   - Again you will need the discretized versions of $\rho(r)$ and $\mu(r)$.
   - To check the solution, use the quantity `norm(a - b)/norm(a)` where `a` is the left-hand side and `b` is the right-hand side of Equation (1). You want this value to be $< 10^{-3}$; you may need to lower the tolerance in the `ode45` solver in `surf_stress.m`.
   - Use the command `gradient(h,r)` to obtain dh/dr where `h` and `r` are equal-length vectors. This numerical approximation of dh/dr will be good if `r` is densely sampled.

6. (0.2) How do the calculations of ${}_0f_{40}$ compare for with and without the lowered tolerance for the `ode45` solver? Let f_a denote the root obtained with the default numerical tolerance in `ode45`. Let f_b denote the root obtained with the lowered numerical tolerance.

   (a) List f_a and f_b in mHz.

   Note: Use `format long` to list the output with as many digits as you need to make the comparison.

   (b) Use $|\ln(f_a/f_b)|$ to quantify the difference between f_a and f_b.

   (c) What does this imply about the accuracy of ${}_nf_l$ relative to the accuracy of ${}_nW_l(r)$ and ${}_nT_l(r)$ in the numerical solutions?

Problem

Approximately how much time *outside of class and lab time* did you spend on this problem set? Feel free to suggest improvements here.

Table 3: Observed toroidal modes eigenperiods (in seconds) for earth (*Dziewonski and Anderson*, 1981). These data can be found in the text file `prem_Tmodes.txt`.

	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$	$l = 8$	$l = 9$	$l = 10$
$n = 0$	NA	2636.38	1705.95	1305.92	1075.98	925.84	819.31	736.86	671.80	618.97
$n = 1$	xxxx	756.57	695.18	xxxx	xxxx	519.09	475.17	438.49	407.74	381.65
$n = 2$	xxxx	xxxx	xxxx	420.46	xxxx	xxxx	363.65	343.34	xxxx	xxxx
$n = 3$	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	259.26	xxxx

Table 4: Toroidal modes eigenperiods (in seconds) predicted for a homogeneous earth model with $\rho = 4380$ kg/m^3, $\mu = 1.54 \times 10^{11}$ Pa. See Table 3 for the corresponding observed periods. List your values with 0.1 precision.

	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=7$	$l=8$	$l=9$	$l=10$
$n=0$	NA									
$n=1$	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx
$n=2$	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx
$n=3$	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx

Table 5: Wave parameters (Table 2) for ${}_0\mathsf{T}_{40}$ ($n=0$, $l=40$) addressed in Problems 1-6 and 3-2. The group speed calculation is from Problem 3-4. List your values with three significant figures.

		homogeneous model (#1-6)		cubic model (#3-2)	
period	T	s	xxxx	s	xxxx
frequency	f	mHz	xxxx	mHz	xxxx
angular frequency	ω	mHz	xxxx	mHz	xxxx
wavelength	λ	rad	km	rad	km
wavenumber	k		1/km		1/km
phase speed	c	rad/s	km/s	rad/s	km/s
group speed (#3-2)	U	xxxx	xxxx	rad/s	km/s

Table 6: Summary of misfit tests for different earth models. The first misfit value is based on using only N observations. The second misfit value is the same model, but computing $F(\mathbf{m})$ for all $N_{\text{total}} = 20$ observations in Table 3. List your values of $F(\mathbf{m})$ (in mHz) with 0.001 precision.

problem	model index	$\rho(r)$ kg m^{-3}	$\mu(r)$ 10^{11} Pa	N	$F(\mathbf{m})$ (N) mHz	$F(\mathbf{m})$ (N_{total}) mHz
#1-8	1	4380	1.54	20	NA	
#2-1	3	4380		9		
#2-1	4	4380		1		
#2-1	2	4380		20	NA	
#2-3	5	linear	linear	20	NA	
#2-3	6	cubic	cubic	20	NA	

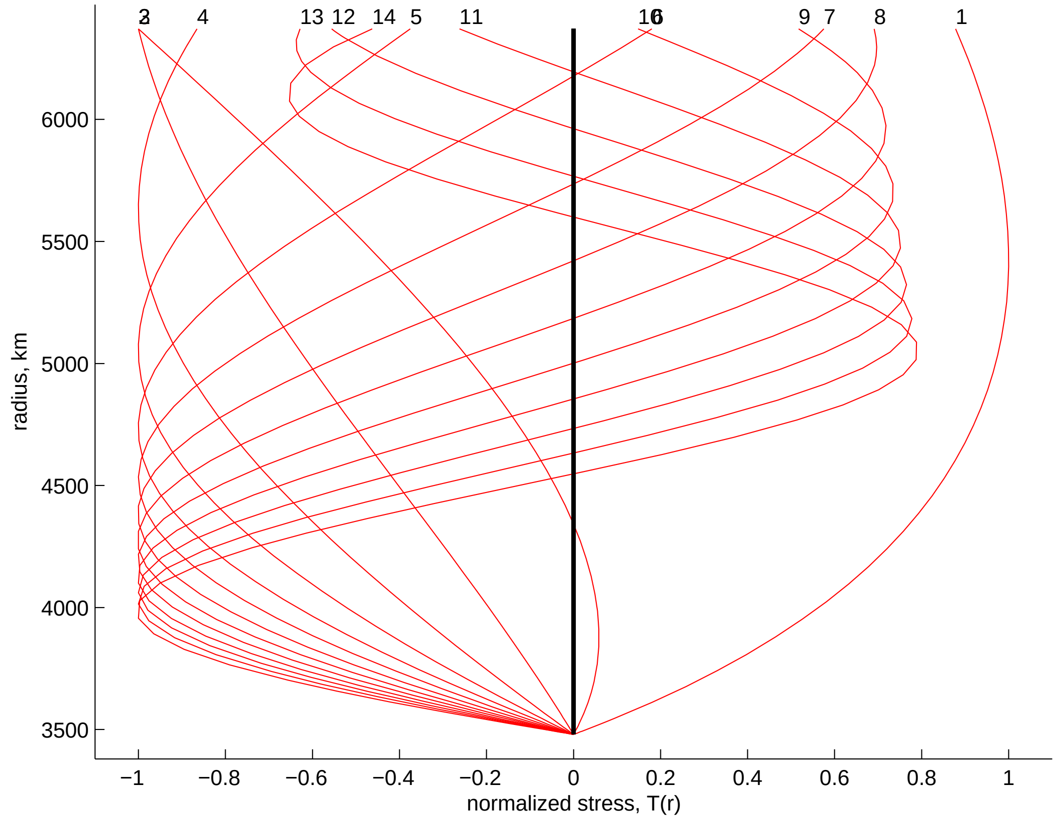

Figure 4: Stress profiles, $T(r)$, for the first 14 values in the frequency vector. The index into the frequency vector, i, is labeled at $T(a)$ for each curve. See Problem 1-2.

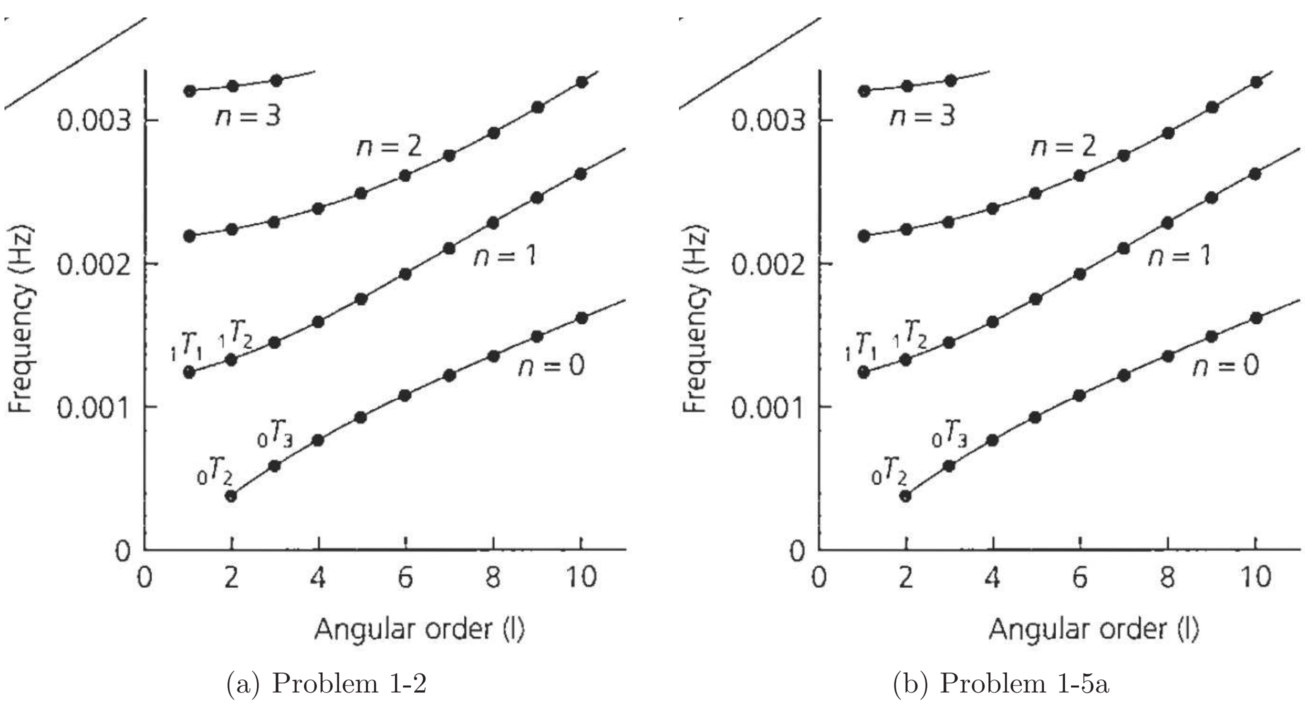

Figure 5: Inset figure from *Stein and Wysession* (2003), Figure 2.9-10.