Influence of Grain Size Evolution and Water Content on the Seismic Structure of the Oceanic Upper Mantle

By

James R. Elsenbeck, II

B.S., Georgia Institute of Technology, 2005

Submitted in partial fulfillment of the requirements of the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

JUNE 2007

Signature of Author

_____

Joint Program in Oceanography/Geology and Geophysics
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
May 25, 2007

Certified by

_____

Mark Behn
Thesis Supervisor

Accepted by

_____

Greg Hirth
Chair, Joint Committee for Geology and Geophysics
Massachusetts Institute of Technology/
Woods Hole Oceanographic Institution

Influence of Grain Size Evolution and Water Content on the Seismic Structure of the Oceanic Upper Mantle

by

James R. Elsenbeck, II

Submitted to the Department of Marine Geology and Geophysics on
May 23, 2007 in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Marine Geology and Geophysics

Abstract

Grain size is an important material property that has significant effects on the viscosity, dominant deformation mechanism, attenuation, and shear wave velocity of the oceanic upper mantle. Several studies have investigated the kinetics of grain size evolution, but have yet to incorporate these evolution equations into large-scale flow models of the oceanic upper mantle. We construct self-consistent 1.5-D steady-state Couette flow models for the oceanic upper mantle to constrain how grain size evolves with depth assuming a composite diffusion-dislocation creep rheology. We investigate the importance of water content by examining end-member models for a dry, wet, and dehydrated mantle (with dehydration above ~60-70 km depth). We find that grain size increases with depth, and varies with both plate age and water content. Specifically, the dehydration model predicts a grain size of ~11 mm at a depth of 150 km for 75 Myr-old oceanic mantle. This results in a viscosity of ~10^{19} Pa s, consistent with estimates from geoid and glacial rebound studies. We also find that deformation is dominated by dislocation creep beneath ~60-70 km depth, in agreement with observations of seismic anisotropy in the oceanic upper mantle. The calculated grain size profiles are input into a Burger's model system to calculate seismic quality factor (Q) and shear wave velocity (Vs). For ages older than 50 Myrs, we find that Q and Vs predicted by the dehydration case best match seismic reference models for Q and the low seismic shear wave velocity zone (LVZ) observed in the oceanic upper mantle.

Thesis Supervisors: Mark Behn; Dan Lizarralde; Greg Hirth
Titles: Assistant Scientist; Associate Scientist; Associate Scientist w/ Tenure

1. Introduction

The oceanic upper mantle is characterized by a zone of seismically observed low shear wave velocity (LVZ) that coincides in depth with the asthenosphere (Gaherty et. al, 1996; Gaherty et. al, 1999; Nishimura & Forsyth, 1989). It has been proposed that this region of low velocity arises because the temperature dependence of seismic velocity is greater than the pressure dependence at these depths (Anderson, 1962; Anderson et al., 1968; Birch, 1969). However, this explanation is insufficient in characterizing the magnitude of velocity changes in this region. For example, the Gutenberg (G)-discontinuity, observed in ScS reverberation data, shows a rapid increase in Vs (~0.2-0.4 km/s) between ~60-80 km depth (Gaherty et al., 1996; Revenaugh & Jordan, 1991), which cannot be explained by variations in temperature and pressure alone. Several models have been proposed to explain the origin of the LVZ including: 1) the presence of partial melt (Ringwood, 1969; Sato et. al 1989), 2) the presence of water (Karato & Jung, 1998), and 3) variations in the mineralogy (Stixrude & Lithgow-Bertelloni, 2004). Most recently, Faul & Jackson (2005) proposed that the LVZ is the result of variations in grain size with depth throughout the oceanic upper mantle. Using the sensitivity of shear wave velocity (Vs) and seismic quality factor (Q) with both temperature and grain-size (Faul & Jackson, 2005; Karato, 2003), Faul & Jackson (2005) demonstrate that the LVZ is consistent with a dry mantle and a grain size of 1 mm in the lithosphere (<165 km) increasing to 5 cm between 165-350 km depth. They

postulate this grain size distribution arises due to a change in the dominant deformation mechanism from dislocation to diffusion creep.

These two mechanisms are the primary components of a composite rheology thought to control deformation in the oceanic upper mantle (Hirth & Kohlstedt, 2003; Karato, 2003; Karato & Wu, 1993). Many studies argue that mantle deformation controls the formation of seismic anisotropy (e.g. Karato & Wu, 1993; Nicolas & Christensen, 1987; Ribe, 1992; Wenk & Christie, 1991), and we know the oceanic upper mantle is seismically anisotropic (e.g. Forsyth, 1975; Gaherty et. al, 1999), as manifest in both radial and azimuthal anisotropy. The dislocation and diffusion creep processes occur simultaneously, with dislocation creep dominating in regions of low stress, low temperature, and high strain rate. In contrast, diffusion creep dominates in regions of high stress, high temperature, and small grain size. Dislocation creep results in a lattice preferred orientation (LPO) that aligns the a-axis (fast direction) of olivine grains (the primary phase in the upper mantle) in the flow direction. This LPO is believed to cause the observed seismic anisotropy in the oceanic upper mantle. LPO does not form during diffusion creep, and the transition from dislocation creep to diffusion creep can possibly erase prior mineral fabrics if enough strain is accommodated. Thus, quantifying the transition in dominant deformation mechanism is essential to understanding the generation of LPO and by extension, seismic anisotropy in the oceanic upper mantle.

Podolefsky et al. (2004) investigated the transition from dislocation to diffusion creep. However, while using a composite rheology to calculate viscosity, they assumed constant water content and grain size. Diffusion creep viscosity is grain size dependent, and thus grain size variations with depth may affect both viscosity and dominant deformation mechanism. Unfortunately, grain size and water content (particularly grain size) are relatively poorly constrained in the oceanic upper mantle. A number of grain size evolution models have been proposed (e.g. Austin & Evans, 2007; Hall & Parmentier, 2003; Montési & Hirth, 2003) that combine competing processes of grain growth and reduction to evolve to a steady state grain size governed by environmental variables (i.e. temperature and stress) as well as material variables (i.e. shear modulus). This results in a relationship between grain size and mantle strain rate/viscosity.

The goal of this study is to use these new kinetic models for grain size evolution to investigate variations in grain size with depth in the oceanic upper mantle. We use a self-consistent numerical model of channel flow with a composite rheology to calculate steady-state grain size as a function of water content in the mantle. The calculated steady-state grain size profiles are then used to estimate seismic structure (Vp, Q, anisotropy) of the oceanic upper mantle. We show that the model that best fits the observed values of Q and Vs contains the presence of water and requires an increase in grain-size from ~250 μm at a depth of ~50 km to ~11 mm at a depth of ~150 km.

2. Methods

We simulate deformation in the oceanic upper mantle using a 1.5-D model for channel flow. The model space is 400 km wide and extends to a depth of 400 km (Fig. 1). Channel flow is implemented through our choice of boundary conditions: 1) a uniform spreading velocity, U_o, along the top of the model space, 2) no-slip along the bottom boundary, and 3) normal flow ($\mathbf{t}\cdot\mathbf{u} = 0$, where $\mathbf{t}$ is the vector tangent to the boundary surface) through the side boundaries. This results in a solution that varies vertically, but is constant in the horizontal direction.

We examine individual model runs for spreading rates of 1-5 cm/yr half-rate (Fig. 2a) and for increasing lithospheric age by imposing a half-space temperature profile on the runs corresponding to ages of 5, 12, 25, 50, and 75 Myr (Fig. 2b). We explore the influence of water content on our solutions by examining 3 end-member cases for a dry (50 H / 10^6 Si), wet (1000 H / 10^6 Si), and dehydrated mantle. The dehydration case assumes that water is removed from the upwelling mantle during melting under the ridge axis. An adiabatically upwelling mantle will cross the dry solidus at 60-70 km depth (Asimow, 2003), and thus we vary the mantle water content between the wet and the dry profiles over this depth range using an arctangent function (Fig. 2c). Using the COMSOL 3.2 finite element package, we solve the incompressible Navier-Stokes equation assuming a composite diffusion-dislocation creep rheology (Hirth & Kohlstedt, 2003)

$$\eta_{tot} = \left(\frac{1}{\eta_{diff}} + \frac{1}{\eta_{disl}} \right)^{-1} \quad (1)$$

where η_{tot} is the composite viscosity and η_{diff} and η_{disl} are the diffusion and dislocation creep viscosities defined as:

$$\eta_{diff} = A_{diff}^{-1/n_{diff}} d^{m/n_{diff}} C_{OH}^{-r/n_{diff}} \exp\left(\frac{Q_{diff} + PV_{diff}}{n_{diff} RT} \right), \text{ and} \quad (2)$$

$$\eta_{disl} = A_{disl}^{-1/n_{disl}} \dot{\varepsilon}^{1/n_{disl}-1} C_{OH}^{-r/n_{disl}} \exp\left(\frac{Q_{disl} + PV_{disl}}{n_{disl} RT} \right), \quad (3)$$

Here A is a prefactor, n is the stress exponent, d is the grain size, m is the grain size exponent, $\dot{\varepsilon}$ is the strain rate (quantified by its second invariant), C_{OH} is water content, r is the water content exponent, Q is the activation energy, P is pressure, V is the activation volume, R is the universal gas constant, and T is temperature. All rheological parameters are given in Table 1. We cap the maximum composite viscosity to be 10^{25} Pa s, and use the ratio of $\dot{\varepsilon}_{disl}$ to $\dot{\varepsilon}_{diff}$ to determine the dominant deformation mechanism with depth.

Grain size evolution is calculated using the synchronous grain growth and recrystallization model of Hall & Parmentier (2003):

$$d_{ss} = \left(\frac{G_0}{p \lambda A_{disl}} \right)^{1/p} \left(\frac{\mu}{\sigma} \right)^{n_{disl}/p} \exp\left(\frac{Q_{disl} - H}{pRT} \right), \quad (4)$$

where d_{ss} is the steady-state grain size, G_0 is a reference grain growth rate, p is the grain growth exponent, λ is a rate constant, μ is the shear modulus, σ is differential stress, H is an activation enthalpy, and all other variables are defined

above. Hall and Parmentier (2003) define the characteristic timescale required to reach the steady-state grain size as:

$$t_{ss} = \left(\frac{p\lambda\dot{\varepsilon}\eta_{tot}}{\eta_{disl}} \right)^{-1}, \quad (5)$$

To solve for the steady-state grain size in Eq. 4, we begin with a constant grain size of 10 mm at all depths and iterate until the solution converges to <0.1% between iterations. The minimum grain size is capped at 250 μm to reduce time consumption per iteration. We consider the steady-state grain-size to be a realistic estimate for the actual grain-size when the characteristic timescale, t_{SS}, is ≤10% of plate age.

We also consider an alternative grain size evolution model proposed by Austin & Evans (2007). This model uses a theoretical framework to explain recrystallization, assuming that the rate of mechanical work governs steady-state grain size:

$$d_{ss} = \left(\frac{K_g \exp\left(\frac{-H_{AE}}{RT} \right) c\gamma\eta_{disl}}{\lambda_{AE} p_{AE} \eta_{tot}^2 \dot{\varepsilon}^2} \right)^{\frac{1}{p_{AE}+1}}, \quad (6)$$

where λ is work fraction, c is a geometric constant, γ is the average specific grain boundary energy, and K_g is a prefactor. The model parameters for both grain-size models are given in Table 2.

2.1 Calculation of Upper Mantle Attenuation and S-wave Velocity

Seismic attenuation and S-wave velocity (Vs) are sensitive to both temperature and grain-size. Thus, we use the half-space temperature profile and the resultant steady-state grain size distribution from each model to calculate shear modulus (G) and seismic quality factor (Q – equivalent to the inverse of attenuation) with depth following the Burger's model outlined in Faul & Jackson (2005):

$$G = \left(J_1{}^2 + J_2{}^2\right)^{\frac{-1}{2}} \tag{7}$$

and

$$Q = \frac{J_2}{J_1}, \tag{8}$$

with

$$J_1 = J_u\left(1 + \left(\frac{\partial \ln J_u}{\partial T}\right)_R \left(\frac{d}{d_R}\right)^{-m_j}\left(T - T_R\right) + \frac{\alpha\Delta}{\tau_H^{\alpha} - \tau_L^{\alpha}} \int_{\tau_L}^{\tau_H} \frac{\tau^{\alpha-1}}{1+\omega^2\tau^2}\, d\tau\right) \tag{9}$$

and

$$J_2 = J_u\left(\frac{\omega\alpha\Delta}{\tau_H^{\alpha} - \tau_L^{\alpha}} \int_{\tau_L}^{\tau_H} \frac{\tau^{\alpha}}{1+\omega^2\tau^2}\, d\tau + \frac{1}{\omega\tau_M}\right), \tag{10}$$

where $\left(\frac{\partial \ln J_u}{\partial T}\right)_R$ is the anharmonic temperature derivative, d_R is a reference grain size, T_R is a reference temperature, α is a frequency dependence parameter, Δ is the anelastic relaxation strength, and ω is the circular

frequency. τ_H, τ_L, and τ_M are the upper, lower, and Maxwell bounds on relaxation time, respectively, and are calculated via:

$$\tau_i = \tau_{iR}\left(\frac{d}{d_R}\right)^m \exp\left(\frac{E}{R}\left(\frac{1}{T}-\frac{1}{T_R}\right)+\frac{PV_{disl}}{RT}\right)\left(\frac{C_{OH}^{dry}}{C_{OH}}\right)^r, \quad (11)$$

where τ_{iR} are the reference values of τ_i, E is the activation energy, m is m_A for τ_L and τ_H, and m_V for τ_M, and all other parameters defined above. Parameters for seismic calculations are given in Table 3. Q is then used to calculate shear wave velocity after Karato (2003):

$$V_s = V_\infty\left(1-\frac{1}{2}\cot\left(\frac{\pi\alpha}{2}\right)Q^{-1}\right), \quad (12)$$

where α is a constant and V_∞ is the velocity at infinite frequency. Assuming a pyrolitic mantle we calculate V_∞ following Stixrude & Lithgow-Bertelloni (2004):

$$V_\infty = 4.77 + 0.0380P - 0.000378(T-300) \quad (13)$$

3. Results

Using the grain size evolution model of Hall & Parmentier (2003), we first evaluate the influence of water content on deformation style and grain size in the oceanic upper mantle. Fig. 3 illustrates the variation in viscosity and grain-size with depth for a 75 Myr-old plate and a half spreading-rate of 5 cm/yr. There is very little difference between the wet and dehydration model results because the 75 Myr-old plate has cooled sufficiently to be effectively rigid at and above the dehydration depth of ~60-70 km. A low viscosity zone is predicted in the depth range of ~100-220 km, consistent with average depth extent of the oceanic

asthenosphere based on seismic observations (e.g. Gaherty et al., 1996; Nishimura & Forsyth, 1989). The size and position of the low viscosity zone are controlled by the interaction of pressure, temperature, and water content (Eqs. 2 & 3, Fig. 3a). Specifically, for a 75 Myr-old plate we find that the addition of water decreases mantle viscosity by up to an order of magnitude, with a minimum viscosity of ~10^{20} Pa s for a dry mantle, and ~10^{19} Pa s for a dehydrated mantle.

The low viscosity region also correlates with a region of large steady state grain size (Eq. 4), with a maximum grain size of ~3.5 mm for a dry mantle, and ~11 mm for a dehydrated mantle (Fig. 3b). We note that the assumption of channel flow in our models implies that differential stress is constant with depth, but only for a single plate age (thermal structure) and water profile, as these parameters affect viscosity. Thus, the depth variation in the grain size profiles are predominantly a function of the lithospheric thermal structure (Eq. 4), as all parameters except temperature in Equation 4 are constant.

Using the ratio of dislocation creep strain rate to diffusion creep strain rate we estimate the dominant deformation mechanism with depth (Fig. 3c). A ratio greater than 1 indicates the dominant deformation mechanism is dislocation creep, whereas a ratio less than 1 indicates deformation is dominated by diffusion creep. The dry and dehydration models both predict dislocation creep to dominate at depths greater than ~80 km, with a region between 50-80 km depth that is dominated by diffusion creep. The ratio for a dehydrated mantle is ~2-3 times larger than for a dry mantle below ~100 km.

Grain size does not evolve instantaneously to the steady state.grain value given in Eq. 4. To evaluate the validity of estimating grain-size from the steady state solution, we calculate the characteristic timescale defined by Hall & Parmentier (2003) (Eq. 5, Fig. 3d). We consider the steady-state solution to be valid when the characteristic timescale is $\leq 10\%$ of plate age. Fig. 3d shows that the dehydration model results in consistently larger characteristic timescales than does the dry model. However, both the dry and dehydration models predict a large region within the 10% threshold between depths of 100-400 km.

Water content in the upper mantle is constrained by geochemical estimates based on the water content of mid-ocean ridge basalts (MORB) (e.g. Michael, 1988), as well as estimates from solubility data for mantle materials (Hirth & Kohlstedt, 1996; Karato & Jung, 1998). As MORB source material rises beneath a mid-ocean ridge and partial melting occurs, water becomes depleted in the solid due to the high solubility of water in basaltic melt (Karato, 1986). Therefore, the dehydration model best captures the physics of the melting process in the oceanic upper mantle. Using the dehydration model and the Hall & Parmentier (2003) grain size evolution model, we next investigate the effect of lithospheric age on viscosity, grain size, characteristic timescale, and dominant deformation mechanism (Fig. 4). The low viscosity zone representing the center of the asthenophsere is found at all lithospheric ages. Both the average depth and minimum viscosity at the center of the asthenosphere vary with age (Fig. 4a) due to the cooling of the oceanic upper mantle. There is a ‘depression’ in the

minimum viscosity for the youngest lithospheric ages at 60-70 km. This is caused by the decrease in water content above the dry solidus. The result is to force deformation beneath the dehydration boundary (~60-70 km), thereby increasing the strain rate and thereby viscosity in this region.

Cooling of the oceanic lithosphere also influences the steady state grain size (Fig. 4b). As described previously, for channel flow stress is constant throughout the model space, and thus the depth-dependence of grain size is primarily a function of temperature. This is readily apparent when comparing the temperature profiles in Figure 2b to the grain size profiles in Figure 4b. We also examine the dependence of dominant deformation mechanism on lithospheric age (Fig. 4c). We find that dislocation creep dominates at depths greater than 100 km for all ages < 100 Myr. In all cases, a small region of diffusion dominated creep occurs at shallow depth (Fig. 4c).

Finally, we calculate the characteristic timescale for grain size evolution to evaluate the validity of the grain size profiles (Fig. 4d). The proportion of the mantle that has a characteristic timescale $\leq 10\%$ of the plate age increases with lithospheric age. Therefore, our steady-state grain size solution is most valid in older oceanic lithosphere, but may not be a good approximation for the grain size in lithosphere younger than 25 Myr. In addition, near the ridge axis our assumption of channel flow will break down as corner flow becomes more important.

3.1 Seismic Structure

To illustrate the influence of grain size on upper mantle seismic structure we calculate shear modulus and attenuation (Eqs. 7-11) versus inverse temperature for constant grain sizes of 0.1 mm, 1 mm, and 10 mm using the Burger's model at a fixed period of 8.2 s (Fig. 5). As inverse temperature increases for a constant grain size, shear modulus increases and attenuation decreases. Superimposed on the constant grain size relationships, we show the results for the steady state grain size solution for the dehydration model at a lithospheric age of 75 Myr (Fig. 5). The steady state solution for both shear modulus and attenuation falls between the 1 mm and 10 mm constant grain size values, moving from 1 mm to 10 mm as temperature increases. These variations correspond to the increase in grain size with depth (and thus temperature) shown in Fig. 4b). Plotting Q and Vs versus depth for an age of 75 Myr, we find a zone of low Q and low Vs (Fig. 6) coincident with the low viscosity zone discussed previously (Fig. 4). The minimum Q predicted is ~30, and the minimum Vs predicted is ~4.3 km/s.

4. Discussion

The goal of this study is to explore the structure of the oceanic upper mantle by combining temperature, rheology, grain size, and water content into a self-consistent steady state channel flow model. We allow grain size to vary as defined by three proposed models (Austin & Evans, 2007 - AE07; Faul & Jackson, 2005 - FJ05; Hall & Parmentier, 2003 - HP03), and test the effects of

water content on observable parameters of the oceanic upper mantle (e.g. viscosity, deformation mechanism, Q, & Vs). In order to directly compare the three grain size models, we focus on our results for a dehydrated 75 Myr-old plate.

Using the HP03 grain size evolution model we obtain grain size distributions, which strongly follow thermal structure. We also observe that the presence of water enhances grain size, as water content influences viscosity and thus stress.. For a 75 Myr-old plate and a dry water model, we find a maximum grain size of ~3.5 mm at ~150 km depth, whereas with a dehydration water model, we find a maximum grain size of ~11 mm at ~150 km depth. Both models contain a sharp gradient between ~50-150 km before reaching the minimum model-allowed grain size of 250 μm (Fig. 3b). This sharp gradient is associated with the rapid change in temperature at the base of the thermal boundary layer.

We also use two alternate upper mantle grain size models to better constrain grain size sensitivity in our models. The first is the grain size evolution model of Austin & Evans (2007) described in Eq. 6. Calculating the steady-state grain size in the same manner as for the HP03 model, we find a maximum grain size of ~8 mm at 400 km depth for the AE07 model. The AE07 and HP03 models have similar shapes, but the AE07 model has a more gradual slope (Fig. 3c). The second is the grain size profile proposed by Faul & Jackson (2005) for a dry mantle, with a maximum grain size of 50 mm at 400 km depth and a sharp

gradient between ~160-330 km before reaching a minimum grain size of 1 mm for the upper ~160 km.

We initially tested the 3 grain size models by comparing the oceanic upper mantle viscosity structure predicted by each model (Fig. 3d) with estimates from geoid and glacial rebound studies (e.g. Hager & Richards, 1989; Mitrovica, 1996; Pollitz & Sacks, 1996). The HP03 grain size evolution model results in a low viscosity zone at ~100-220 km depths, with a minimum of ~10^{19} Pa s. The AE07 grain size evolution model generates a low viscosity zone that is shallower than both HP03 and FJ05. Throughout this region it retains a relatively constant minimum of ~10^{19} Pa s. The FJ05 grain size model for a dry mantle results in a similar low viscosity zone, except with a minimum of ~10^{18} Pa s. However, as described earlier it is likely that the oceanic upper mantle contains ~1000 H / 10^6 Si at depths >70 km. Evaluating the FJ05 grain size distribution assuming our dehydration water model, we find a minimum viscosity of ~10^{16} Pa s. Aside from incorporating water in the FJ05 grain size model, all three of these profiles are quite similar and in agreement with results from geoid and glacial rebound studies, which estimate asthenospheric viscosity to be on the order of ~10^{18}-10^{19} Pa s.

We further test the three grain size models by comparing each model's output for Q and Vs with the observations (Fig. 7a,b, & e). The HP03 grain size evolution model produces Q profiles which fit the observations (Dziewonski & Anderson, 1981; Selby & Woodhouse, 2002) (Fig. 6a), and below ~80 km the Vs

profiles generated also fit the observations (Nishimura & Forsyth, 1989; Gaherty et al., 1996) (Fig. 6b). The AE07 grain size evolution model produces profiles for Q and Vs that are slightly lower than those of HP03, while the FJ05 grain size profile for a dry mantle predicts much larger values for Q, and only slightly larger values for Vs (Figs. 7a-b).

Q is function of grain size, with small grain sizes resulting in stronger attenuation. From this relationship, one might expect the FJ05 profile to have the lowest Q at ~120-250 km depth because in this depth range it predicts the smallest grain size (Fig. 7a-c). However, the FJ05 grain size profile assumes a dry mantle, and the presence of water causes Q to decrease (by decreasing cutoff times in Eq. 11 by factor of ~36). Calculating FJ05 for a dehydrated mantle, we find it indeed predicts the lowest Q of the three grain size hypotheses in this depth range.

We also evaluate the dominant deformation mechanism with depth generated by each grain size model (Fig. 7e). We find that for HP03, deformation is dominated by dislocation creep from ~60-400 km depth, peaking at ~150 km and gradually weakening with depth, which is consistent with observations of seismic anisotropy for the oceanic upper mantle (Gaherty et. al, 1999; Montagner, 2002) (Fig. 4c). AE07 has similar results, with dislocation creep dominating deformation, but in the range of ~120-400 km depth. There is a small increase in AE07's deformation mechanism ratio occurring at ~55 km depth, due to a feedback mechanism between the effect of grain size on diffusion

creep viscosity and the AE07 grain size evolution model (Eq. 6). As diffusion creep becomes more dominant, the ratio of dislocation creep viscosity to total viscosity in Eq. 6 increases (recall that total viscosity follows the deformation mechanism with the smallest individual viscosity – Eq. 1). This enhances grain size, increasing diffusion creep viscosity and reducing the strength of diffusion creep (from Eq. 2 – evident as the small increase in the dislocation creep/diffusion creep ratio in Fig. 7e). This process reduces the ratio of viscosities in Eq. 6, causing grain size to decrease. Grain size reduction lowers diffusion creep viscosity and correspondingly strengthens diffusion creep (seen as the decrease in dislocation creep/diffusion creep ratio beneath the small increase at ~55 km depth). As depth increases beyond the dehydration zone at ~65 km depth, strain rate rapidly increases. This strengthens the effect of dislocation creep (from Eq. 3), and as dislocation creep viscosity becomes dominant, the ratio of viscosities in Eq. 6 becomes constant. Eventually this feedback mechanism turns off, as diffusion creep viscosity becomes quite large (due to grain growth with depth).

The FJ05 grain size profile for a dry mantle predicts deformation to be dominated by diffusion creep in the range of ~50-250 km depth (Fig. 7e). This is counter to evidence of seismic anisotropy in the oceanic upper mantle. In the presence of water, the FJ05 grain size profile is even more strongly dominated by diffusion creep, as water content enhances the effect of diffusion creep relative to dislocation creep because of a difference in stress exponent (Eq. 2-3,

Table 2). Accordingly, if we calculate HP03 for a dry mantle, we find that dislocation creep strength is enhanced. Of the three grain size models tested we believe that both the Hall & Parmentier (2003) and Austin & Evans (2007) models for a dehydrated 75 Myr-old plate best fit the constrainable parameters, and thus represent a sufficient estimate of upper mantle grain size.

One of the most difficult seismic observations to reconcile has been the presence of the G-discontinuity - a sharp decrease observed in shear wave velocity at ~60-80 km depth (Revenaugh & Jordan, 1991), which is prominent in the PA5 velocity model of Gaherty et al. (1996) (Fig. 6b). Our shear wave velocity results based on the HP03 grain size model do not predict an abrupt change in Vs consistent with the G-discontinuity . However, we find at young plate ages (and thus high temperatures), the dehydration boundary does generate a relatively abrupt velocity change. This is due in part to elevated temperatures positioning the low viscosity zone within the dehydration region (~60-70 km, Fig. 4a). The removal of water forces viscosity to rapidly increase, with a viscosity minimum directly beneath the onset of dehydration at 70 km. This rapid increase in viscosity causes a similarly rapid increase in shear wave velocity, rising ~0.1-0.2 km/s over ~10-20 km. Though, we note that the channel flow assumption for young plate ages may not be accurate due to significant vertical flow.

Gung et al. (2003) propose the G-discontinuity may indicate the boundary for seismic anisotropy. Using the HP03 grain size evolution model, we find that

dislocation creep does initiate near the base of the G-discontinuity (Fig. 7e). However, in the AE07 grain size evolution model, dislocation creep does not become dominant until beneath ~120 km depth. At the same time, the AE07 model generates lower values for Q, and more importantly, for Vs (Fig. 7a-b). While this decrease is relatively modest, it suggests grain size may play a role in the presence of the G-discontinuity.

Despite our shear wave velocity results not capturing the sharp contrasts seen in PA5, we do find that our velocity profiles are strikingly similar to those of Nishimura & Forsyth (1989, Fig. 8a-c). Both the depth range for the low velocity zone (LVZ) and magnitude for each plate age of our model line up almost exactly with their observations. This demonstrates that our model is performing reasonably as a function of age. The fit improves with plate age, and we believe this is due in part to a lack of corner flow in our models. We additionally show that the effect of spreading rate on shear wave velocity is very minimal compared to the effect of plate age. However, as spreading rate decreases, shear wave velocity does show a slight increase for all depths. Faster spreading rates produce a larger strain rate throughout the model space, but also produce a lower viscosity. The decrease in viscosity is stronger than the strain rate increase, which results in a smaller differential stress, causing a slight decrease in grain size (Eq. 4). Attenuation increases (Q decreases) with decreasing grain size, and thus shear wave velocity slightly decreases (Eq. 12) with increasing spreading rate.

Extending this discussion to anisotropy, we show that the depth of maximum dislocation creep strength (and thus maximum strength of anisotropy) increases with plate age (Fig. 8d-f). This effect is also visible in the radial anisotropy observations of Nishimura & Forsyth (1989), which we uniformly scale to fit within the figure axes. Our models at 25-Myr and 75-Myr do not show strong dislocation creep at shallow depths, something which is visible in the observations. This difference arises due to our model design. An oceanic plate is progressively deformed as it moves off-axis. This deformation remains recorded within the plate structure, and thus these observations record a time-integration of all deformation that has occurred over the course of the particular plate's history. Our models do not consist of a time-integration of deformation with age; they show only a snapshot of deformation at a particular age. If we average our deformation mechanism ratio over all three ages (5, 25, and 75-Myr) we obtain a rough time-integration in which dislocation creep dominates at all depths, similar to the observed radial anisotropy for a 52-110 Myr-old plate in Nishimura & Forsyth (1989, Fig. 8g). In order to capture a more robust time-integration of plate deformation, we would need to both track deformation as a plate moves off-axis, as well as incorporate corner flow. This type of model design extends beyond the time constraints of a Master's Thesis.

5. Conclusions

We present a self-consistent numerical model of channel flow, combining temperature, rheology, grain size, and water content to explore the structure of

the oceanic upper mantle. Using a synchronous model of grain size evolution proposed by Hall & Parmentier (2003), we find that grain size strongly follows thermal structure, and the presence of water enhances grain size via a decrease in viscosity. We test two alternate grain size hypotheses (Austin & Evans, 2007; Faul & Jackson, 2005), and find that the theoretical approach to dynamic recrystallization proposed in Austin & Evans (2007) results in a grain size profile very similar to that of Hall & Parmentier (2003). While our model results for shear wave velocity do not capture the G-discontinuity seen in PA5 (Gaherty et. al, 1996), our results for both shear wave velocity and anisotropic structure do closely match the observations of Nishimura & Forsyth (1989). We believe that the Hall & Parmentier (2003) and Austin & Evans (2007) models for a dehydrated 75 Myr-old plate best fit observations of Q, Vs, and seismic anisotropy, and thus provide a reasonable estimate of grain size for the oceanic upper mantle.

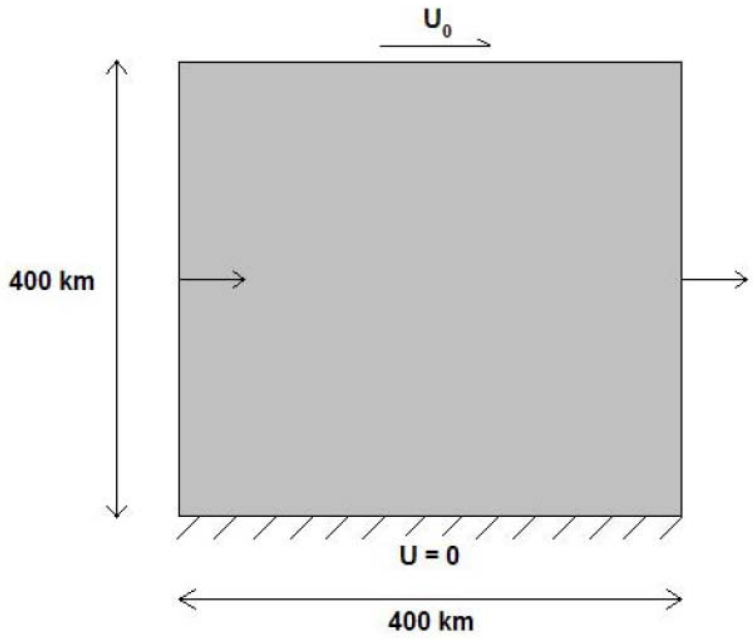

Fig. 1. Model design schematic. The height and width of the model space are each 400 km. The boundary conditions are as follows: the top boundary has a fixed velocity U_0, the bottom boundary is fixed with U = 0, and the side boundaries allow for flow normal to their surface (as indicated by arrows).

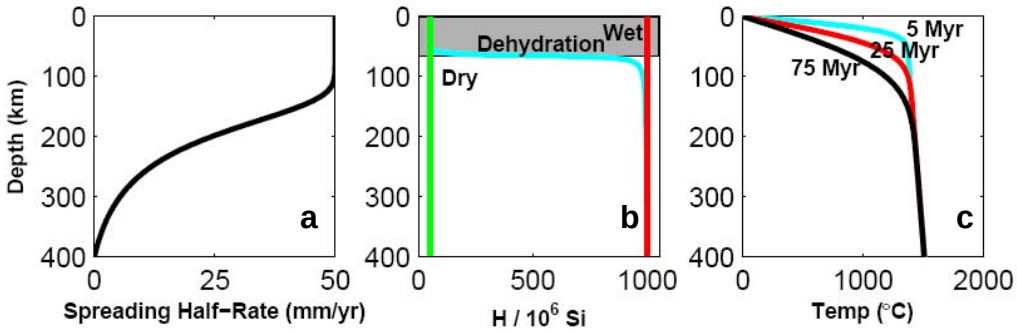

Fig. 2. (a) Velocity structure for an imposed half-rate spreading of 50 mm/yr. (b) The three water profiles used in this study. (c) Thermal profiles for 5, 25, and 75 Myr-old plates.

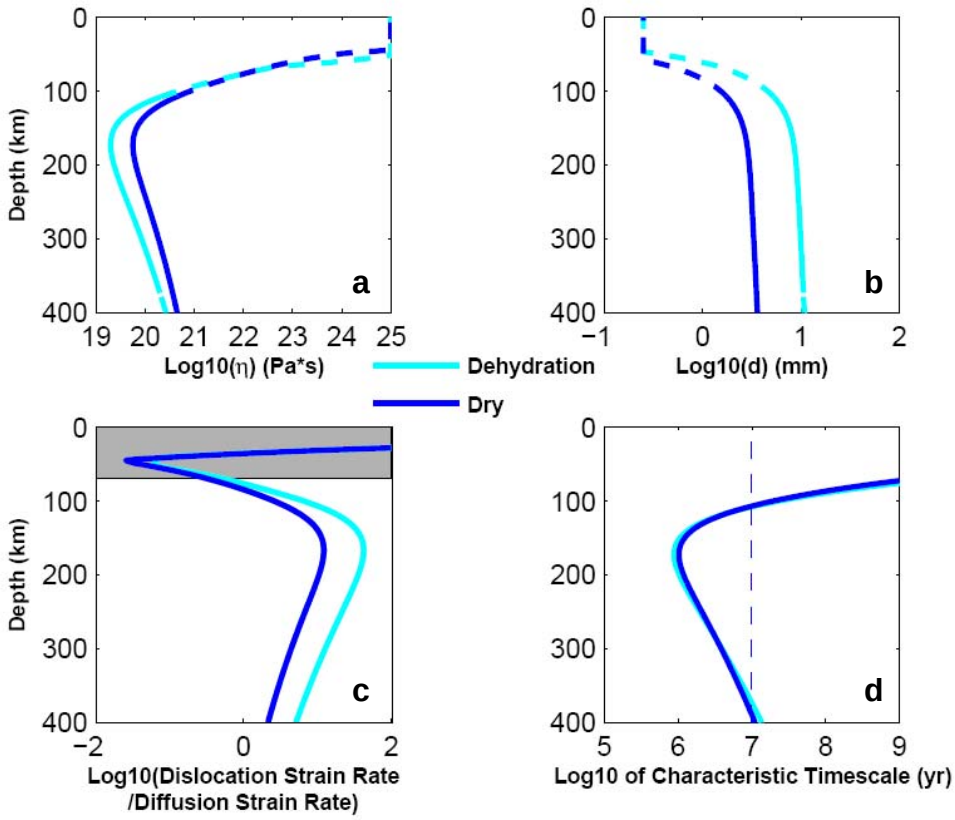

Fig. 3. Initial model results for dry & dehydration water profiles for a 75 Myr-old plate with a 50 mm/yr spreading half-rate. The dashed portions of (a) and (b) indicate where characteristic timescale is greater than ~10% of plate age. (a) Log10 of viscosity. (b) Log10 of grain size. (c) Log10 of the ratio of dislocation creep to diffusion creep. The grey box represents the dehydration depth. The black dashed line indicates where the ratio is equal to 1. At values greater than the black dashed line dislocation creep is dominant, and at values less than the black dashed line diffusion creep is dominant. (d) Log10 of characteristic timescale. The dashed line indicates ~10% of plate age. We believe the steady state results for grain size and viscosity are obtainable for depths at which the characteristic timescale is less than the dashed line.

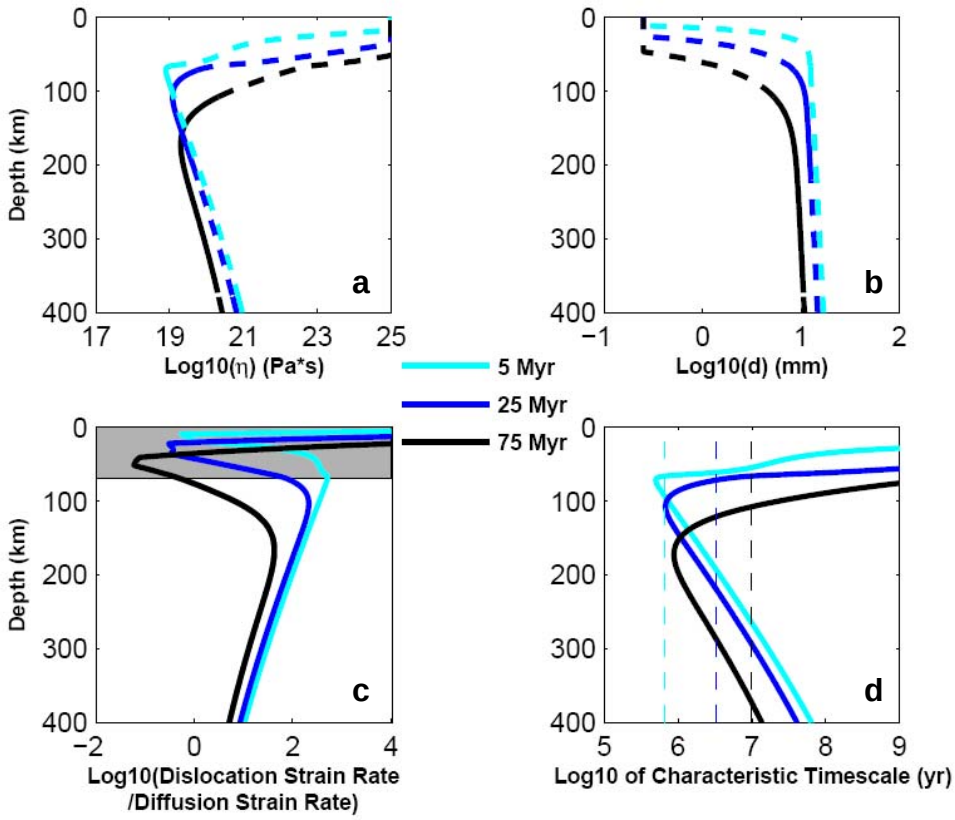

Fig. 4. Initial model results for dehydrated 5, 25, and 75 Myr-old plates with a 50 mm/yr spreading half-rate. The dashed portions of (a) and (b) are the same as described in Fig. 2. (a) Log10 of viscosity. As age increases (and temperature decreases) both the depth and magnitude of minimum viscosity increase. (b) Log10 of grain size. As age increases the maximum grain size decreases, and the depth of transition to minimum grain size increases. (c) Log10 of the ratio of dislocation creep to diffusion creep. The grey box represents the dehydration depth. The dashed line is as indicated in Fig. 3. At younger ages dislocation creep initiates more shallowly, and the relative strength is also much larger. (d) Log10 of characteristic timescale. The dashed lines are as indicated in Fig. 3. The proportion of model space within the ~10% of plate age criterion increases with plate age.

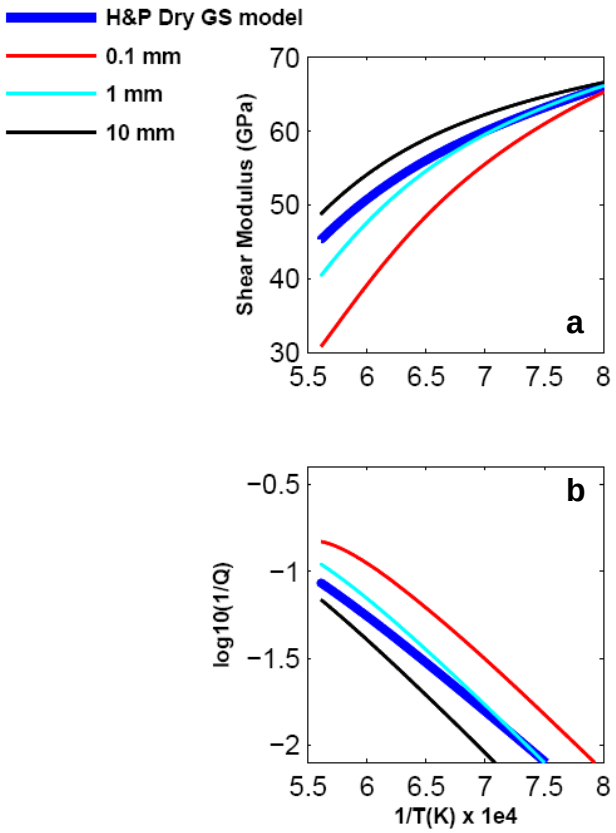

Fig. 5. Comparison of shear modulus (a) and attenuation (b) to reduced temperature as a function of grain size (Faul & Jackson, 2005). The steady-state grain size curve using the Hall & Parmentier (2003) grain size evolution model for a dry 75 Myr-old plate is shown in the thick blue line.

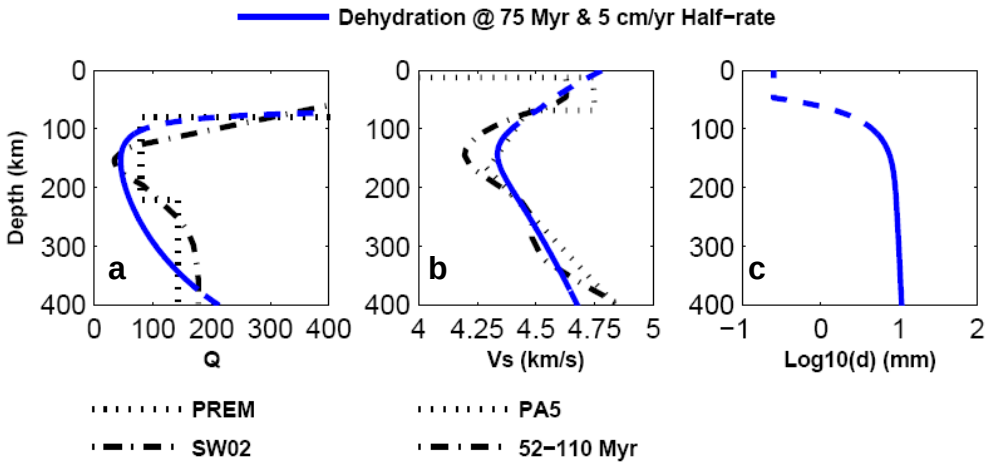

Fig. 6. Comparison of seismic parameters with observations using the Hall & Parmentier (2003) grain size evolution model for a dehydrated 75 Myr-old plate. The dashed portion of the blue line is where characteristic timescale is greater than ~10% of plate age. (a) Q plotted against PREM (Dziewonski & Anderson, 1981) and SW02 (Selby & Woodhouse, 2002). (b) Vs plotted against the 52-110 Myr velocity curve of Nishimura & Forsyth (1989) and PA5 (Gaherty et al., 1996). (c) Log10 of grain size.

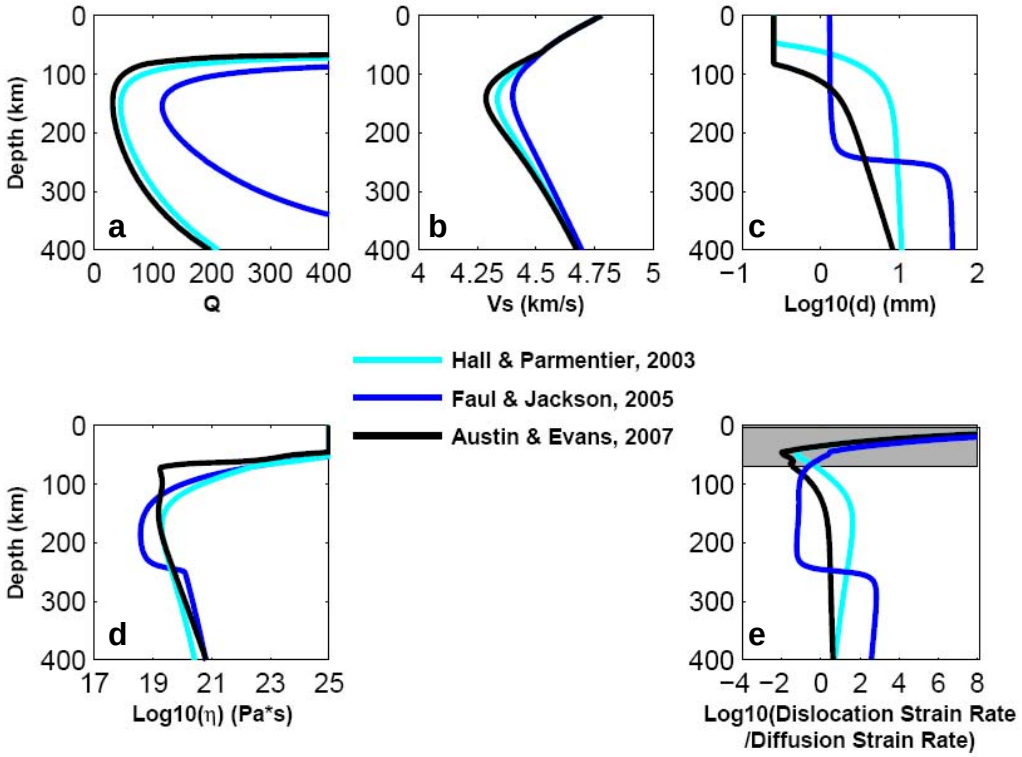

Fig. 7. Comparison of model results for the three grain size models tested. The Hall & Parmentier (2003) and Austin & Evans (2007) grain size evolution models are calculated for a dehydrated 75 Myr-old plate, while the Faul & Jackson (2005) grain size model is for a dry mantle. (a) Faul & Jackson (2005) results in a higher Q than both grain size evolution models. (b) Hall & Parmentier (2003) appears to bisect the Austin & Evans (2007) and Faul & Jackson (2005) Vs profiles. (c) The grain size evolution models result in similar grain size profiles. (d) Hall & Parmentier (2003) and Faul & Jackson (2005) produce similar viscosity profiles, but the minimum viscosity for Faul & Jackson (2005) is an order of magnitude smaller. Austin & Evans (2007) generates a wider LVZ that remains relatively constant. (e) The grain size evolution models both display dislocation creep as the dominant deformation mechanism in the range of ~100-400 km depth. Faul & Jackson (2005) results in deformation being dominated by diffusion creep for ~50-250 km depth.

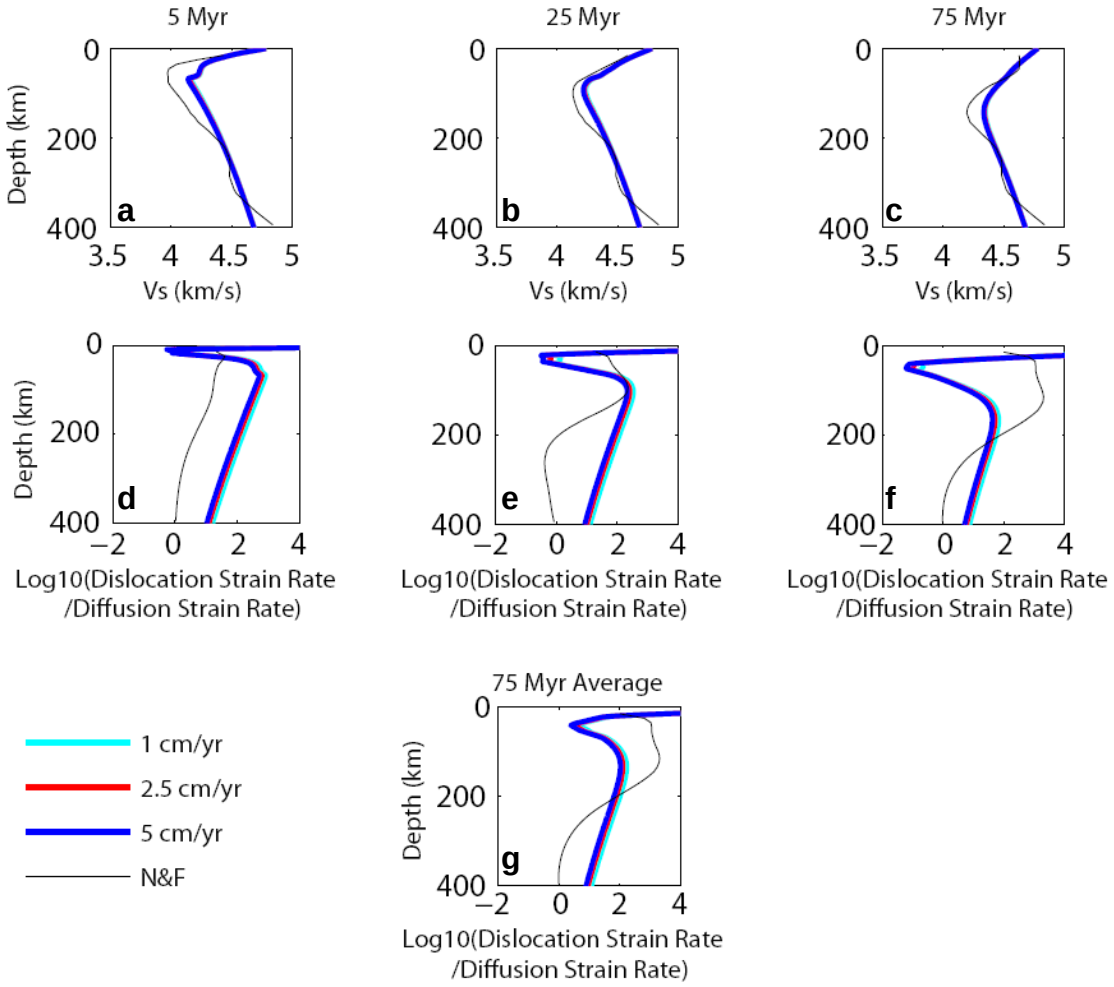

Fig. 8. Shear wave velocity and attenuation as a function of plate age and spreading rate. The black contours are velocity and radial anisotropy observations from Nishimura & Forsyth (1989). (a) Shear wave velocity for a 5 Myr-old plate against the NF89 0-4 Myr velocity model. (b) Shear wave velocity for a 25 Myr-old plate against the NF89 4-20 Myr velocity model. (c) Shear wave velocity for a 75 Myr-old plate against the NF89 52-110 Myr velocity model. (d) Ratio of deformation mechanism for a 5 Myr-old plate against scaled NF89 0-4 Myr radial anisotropy. (e) Ratio of deformation mechanism for a 25 Myr-old plate against scaled NF89 4-20 Myr radial anisotropy. (f) Ratio of deformation mechanism for a 75 Myr-old plate against scaled NF89 52-110 Myr radial anisotropy. (g) Ratio of deformation mechanism for an average of the 5, 25, & 75 Myr-old plate against scaled NF89 52-110 Myr radial anisotropy.

Table 1

Parameters of Rheological Model

Parameter	Unit	Value
T_Θ	°C	1350
R	J mol^{-1} K^{-1}	8.314
r		1.2
A_{diff}	s^{-1} MPa^{-n}	3.33E+05
A_{disl}	s^{-1} MPa^{-n}	30
n_{diff}		1
n_{disl}		3.5
m_{diff}		3
m_{disl}		0
Q_{diff}	J mol^{-1}	3.35E+05
Q_{disl}	J mol^{-1}	4.80E+05
V_{diff}	m^3 mol^{-1}	4.00E-06
V_{disl}	m^3 mol^{-1}	1.60E-05

Table 2

Parameters of Grain Size Models

Parameter	Unit	Value
G_0	m^3 s^{-1}	0.43
R	J mol^{-1} K^{-1}	8.314
p		3
λ		1
μ	Pa	5.00E+10
A_{disl}	s^{-1} MPa^{-n}	30
n_{disl}		3.5
Q_{disl}	J mol^{-1}	4.80E+05
H	J mol^{-1}	6.00E+05
K_g	μm^p s^{-1}	7.00E+16
p_{AE}		2
λ_{AE}		0.1
c		1
γ	J μm^{-2}	1
H_{AE}	J mol^{-1}	5.20E+05

Table 3

Parameters of Seismic Calculations

Parameter	Unit	Value
R	J mol^{-1} K^{-1}	8.314
$(\partial \ln J_U / \partial T)_R$	K^{-1}	9.10E-04
d_R	m	1.00E-05
m_j		0.16
m_A		1.09
m_V		2.1
E	J mol^{-1}	5.05E+05
T_R	K	1223
α		0.27
Δ		1.4
τ_{HR}		5.26E+06
τ_{LR}		3.98E-03
τ_{MR}		4.31E+06
ω	rad s^{-1}	0.766
J_U	GPa^{-1}	1.49E-02
C_{OH} (dry)	ppm H / Si	50
r		1.2
V_{disl}	m^3 mol^{-1}	1.60E-05

References

Anderson, D.L., 1962. The plastic layer of the Earth's mantle. Scientific American 207, 52?59.

Anderson, O.L., Schreiber, E., Liebermann, R.C., 1968. Some elastic constant data on minerals relevant to geophysics. Rev. Geophys. 6, 491?524.

Asimow, P.D., 2003. A slice of history. Nature 423, 491?493.

Austin, N., Evans, B., 2007. Paleowattmeters: A scaling relation for dynamically recrystallized grain size. Geology 35, 343?346.

Birch, F., 1969. Density and composition of the upper mantle: First approximation as an olivine layer. Geophys. Mono. 13, 18?36.

Dziewonski, A.M., Anderson, D.L., 1981. Preliminary reference Earth model. Phys. Earth Planet. Interiors 25, 297?356.

Faul, U.H., Jackson, I., 2005. The seismological signature of temperature and grain size variations in the upper mantle. Earth. Planet. Sci. Lett. 234, 119?134.

Forsyth, D.W., 1975. The early structural evolution and anisotropy of the oceanic upper mantle. Geophys. J. Royal Astro. Soc. 43, 103?162.

Gaherty, J.B., Jordan, T.H., Gee, L.S., 1996. Seismic structure of the upper mantle in a Central Pacific corridor. J. Geophys. Res. 101, 22291?22309.

Gaherty, J.B., Kato, M., Jordan, T.H., 1999. Seismological structure of the upper mantle; a regional comparison of seismic layering. Phys. Earth Planet. Int. 110, 21?41.

Gung, Y., Panning, M., Romanowicz, B., 2003. Global anisotropy and the thickness of continents. Nature 422, 707?711.

Hager, B.H., Richards, M.A., 1989. Long-wavelength variations in Earth's geoid: physical models and dynamical implications. Philo. Trans. Royal Soc. London, Series A: Math. Phys. Sci. 328, 309?327.

Hall, C.E., Parmentier, E.M., 2003. Influence of grain size evolution on convective instability. Geochem., Geophys., Geosys. 4.

Hirth, G., Kohlstedt, D., 2003. Rheology of the upper mantle and mantle wedge; a view from the experimentalists. Geophys. Mono. 138, 83?105.

Karato, S., 2003. Mapping water content in the upper mantle. Geophys. Mono. 138, 135?152.

Karato, S., Jung, H., 1998. Water, partial melting and the origin of the seismic low velocity and high attenuation zone in the upper mantle. Earth. Planet. Sci. Lett. 157, 193?207.

Karato, S., Wu, P., 1993. Rheology of the upper mantle: a synthesis. Science 260, 771?778.

Michael, P.J., 1988. The concentration, behavior and storage of H_2O in the suboceanic upper mantle: Implications for mantle metasomatism. Geochim. Cosmochim. Acta 52, 555?566.

Mitrovica, J.X., 1996. Haskell [1935] Revisited. J. Geophys. Res. 101, 555?569.

Montagner, J.P., 2002. Upper mantle low anisotropy channels below the Pacific Plate. Earth Planet. Sci. Lett. 202, 263?274.

Montési, L.G.J., Hirth, G., 2003. Grain size evolution and the rheology of ductile shear zones; from laboratory experiments to postseismic creep. Earth Planet. Sci. Lett. 211, 97?110.

Nicolas, A., Christensen, N.I., 1987. Formation of anisotropy in upper mantle peridotites; a review. Geodyn. Series 16, 111?123.

Nishimura, C.E., Forsyth, D.W., 1989. The anisotropic structure of the upper mantle in the Pacific. Geophys. J. Royal Astro. Soc. 96, 203?229.

Podolefsky, N.S., Zhong, S., McNamara, A.K., 2004. The anisotropic and rheological structure of the oceanic upper mantle from a simple model of plate shear. Geophys. J. Int. 158, 287?296.

Pollitz, F.F., Sacks, I.S., 1996. Viscosity structure beneath northeast Iceland. J. Geophys. Res. 101, 17771?17793.

Revanaugh, J., Jordan, T.H., 1991. Mantle layering from ScS reverberations; 3, The upper mantle. J. Geophys. Res. 96, 19781?19810.

Ribe, N.M., 1992. On the relation between seismic anisotropy and finite strain. J. Geophys. Res. 97, 8737?8747.

Ringwood, A.E., 1969. Composition and evolution of the upper mantle. Geophys. Mono. 13, 1?17.

Sato, H., Sacks, I.S., Murase, T., Muncill, G.E., Fukuyama, H., 1989. Qp-melting temperature relation in peridotite at high pressure and temperature; attenuation mechanism and implications for the mechanical properties of the upper mantle. J. Geophys. Res. 94, 10647?10661.

Selby, N.D., Woodhouse, J.H., 2002. The Q structure of the upper mantle; constraints from Rayleigh wave amplitudes. J. Geophys. Res. 107, 12 pp.

Stixrude, L., Lithgow-Bertelloni, C., 2005. Mineralogy and elasticity of the oceanic upper mantle; origin of the low velocity zone. J. Geophys. Res. 110, 16 pp.

Wenk, H.R., Christie, J.M., 1991. Comments on the interpretation of deformation textures in rocks. J. Struct. Geol. 13, 1091?1110.

Acknowledgements

I would like to thank Mark Behn, Greg Hirth, Dan Lizarralde, and Stephané Rondenay for both their insight and their comments on this manuscript. I would also like to thank both Academic Programs and NSF Grant # OCE-0625178 for financial support.

Appendix: Code Documentation

This work uses two software packages: Comsol Multiphysics 3.2 and Matlab 7.2. Comsol Multiphysics 3.2 is a finite element modeling package we use to solve the incompressible Navier-Stokes equation. We solve each model (variable spreading rate and plate age) first for a constant grain size of 10 mm, which we then load into the grain size evolution code *ubergsize.m* which uses functions *dFEMconpoly.m*, *dFEMconpolyf.m*, *dFEMconpolyn.m*, and *dFEMconpolyNA.m*. This code outputs a suite of textfiles, the most important file containing the steady-state grain size solution. This grain size solution is then input into the original Comsol Multiphysics model as a function, and these models are re-solved to store all variables for easy access. We last use a compiling code *uberdeform.m* to load data from the steady-state grain size solution models and visualize this data. This code uses the function *deformread.m* to load an individual model and calculate attenuation and shear wave velocity from parameters stored within the model. All code was written by James Elsenbeck in MATLAB 7.2.

1. Comsol Model Naming Convention

The naming convention for the Comsol Multiphysics *.mph files is as follows:

```
1Ddeform050+CH2O-1e25tri-hidconst-1.0n.mph
1Ddeform250+CH2O-1e25tri-hidconst-2.5n.mph
1Ddeform750+CH2O-1e25tri-hidconst-5.0n.mph
```

Where 050, 250, and 750 represent plate ages of 5, 25, and 75 Myr. The +CH2O means it uses a dehydration water model, 1e25 is the maximum viscosity used, tri is triangular elements, and hidconst means it is solved for a constant grain size of 10 mm. 1.0, 2.5, and 5.0 are spreading half-rates of 1, 2.5, and 5.0 cm/yr, and n means it incorporates the Hall & Parmentier (2003) grain size evolution model. For our dry and wet water model we change the filename to

```
1Ddeform050-CH2O-1e25tri-hidconst-1.0n.mph
```

and

```
1Ddeform050+fH2O-1e25tri-hidconst-1.0n.mph
```

respectively. We also use the Austin & Evans (2007) grain size evolution model and change the filename to

```
1Ddeform050+CH2O-1e25tri-hidconst-1.0N3.mph
```

After these files are run through the grain size evolution codes listed previously, they are re-solved using the calculated grain size profiles for both the Hall & Parmentier (2003) and Austin & Evans (2007) grain size evolution models, as well as for the grain size profile from Faul & Jackson (2005). We change the filename as follows:

```
1Ddeform050+CH2O-1e25tri-hidconv-1.0n.mph
1Ddeform250+CH2O-1e25tri-hidconv-2.5n.mph
1Ddeform750+CH2O-1e25tri-hidconv-5.0n.mph

1Ddeform050-CH2O-1e25tri-hidconv-1.0n.mph

1Ddeform050+fH2O-1e25tri-hidconv-1.0n.mph

1Ddeform050+CH2O-1e25tri-hidconv-1.0N3.mph

1Ddeform050-CH2O-1e25tri-F&J4con-1.0n.mph
```

with the last representing the Faul & Jackson profile.

2. Grain Size Evolution Codes

ubergsize.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% ubergsize - Calls function dFEMconpoly for Comsol models of varying         %
%             plate age/spreading rate/water content/grain size               %
%             evolution model.                                                %
%             5/20/07 - Jimmy Elsenbeck                                       %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                                                                      %
%                                                                             %
%        data - lists input constant grain size solution models               %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

data=['1Ddeform050+CH2O-1e25tri-hidconst-1.0n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconst-1.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconst-1.0n.mph'; ...
      '1Ddeform050+CH2O-1e25tri-hidconst-2.5n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconst-2.5n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconst-2.5n.mph'; ...
      '1Ddeform050+CH2O-1e25tri-hidconst-5.0n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconst-5.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconst-5.0n.mph'; ...
      '1Ddeform750-CH2O-1e25tri-hidconst-5.0n.mph'; ...
      '1Ddeform750+fH2O-1e25tri-hidconst-5.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconst5.0N3.mph'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output:                                                                     %
%                                                                             %
%   deqfiles - final steady-state grain size evolution solutions              %
%     ofiles - differential stress solution                                   %
%    PIfiles - steady-state grain size solutions at each iteration            %
%   errfiles - error values of grain size variation at each iteration         %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

deqfiles=['deq050-1.0n.txt';'deq250-1.0n.txt'; ...
          'deq750-1.0n.txt';'deq050-2.5n.txt'; ...
          'deq250-2.5n.txt';'deq750-2.5n.txt'; ...
          'deq050-5.0n.txt';'deq250-5.0n.txt'; ...
          'deq750-5.0n.txt';'deq750-5.0C.txt'; ...
          'deq750-5.0f.txt';'deq7505.0N3.txt'];
```

```
ofiles=['o050-1.0n.txt';'o250-1.0n.txt'; ...
        'o750-1.0n.txt';'o050-2.5n.txt'; ...
        'o250-2.5n.txt';'o750-2.5n.txt'; ...
        'o050-5.0n.txt';'o250-5.0n.txt'; ...
        'o750-5.0n.txt';'o750-5.0C.txt'; ...
        'o750-5.0f.txt';'o7505.0N3.txt'];

PIfiles=['PI050-1.0n.txt';'PI250-1.0n.txt'; ...
         'PI750-1.0n.txt';'PI050-2.5n.txt'; ...
         'PI250-2.5n.txt';'PI750-2.5n.txt'; ...
         'PI050-5.0n.txt';'PI250-5.0n.txt'; ...
         'PI750-5.0n.txt';'PI750-5.0C.txt'; ...
         'PI750-5.0f.txt';'PI7505.0N3.txt'];

errfiles=['err050-1.0n.txt';'err250-1.0n.txt'; ...
          'err750-1.0n.txt';'err050-2.5n.txt'; ...
          'err250-2.5n.txt';'err750-2.5n.txt'; ...
          'err050-5.0n.txt';'err250-5.0n.txt'; ...
          'err750-5.0n.txt';'err750-5.0C.txt'; ...
          'err750-5.0f.txt';'err7505.0N3.txt'];

count=size(data);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Here we check the file type and load the appropriate function for            %
% the given grain size evolution model/water content                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:count(1)
   if data(i,13)=='f'
       DEQ(i,:)=dFEMconpolyf(deqfiles(i,:),ofiles(i,:), ...
                             PIfiles(i,:),errfiles(i,:),data(i,:));
   elseif data(i,38)=='3'
       DEQ(i,:)=dFEMconpolyNA(deqfiles(i,:),ofiles(i,:), ...
                              PIfiles(i,:),errfiles(i,:),data(i,:));
   elseif data(i,12)=='-'
       DEQ(i,:)=dFEMconpolyn(deqfiles(i,:),ofiles(i,:), ...
                             PIfiles(i,:),errfiles(i,:),data(i,:));
   else
       DEQ(i,:)=dFEMconpoly(deqfiles(i,:),ofiles(i,:), ...
                            PIfiles(i,:),errfiles(i,:),data(i,:));
   end
end
```

dFEMconpoly.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% dFEMconpoly - Grain size evolution model for a dehydration water             %
%               model and Hall & Parmentier grain size evolution               %
%               model.                                                         %
%             - Jimmy Elsenbeck 5/20/07                                        %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                                                                       %
%                                                                              %
%        f1 – deqfiles filename from ubergsize.m                               %
%        f2 – ofiles filename from ubergsize.m                                 %
%        f3 – PIfiles filename from ubergsize.m                                %
%        f4 – errfiles filename from ubergsize.m                               %
%        d1 – Comsol model filename from ubergsize.m                           %
%                                                                              %
% Output:                                                                      %
%                                                                              %
%        PIdeq – steady-state grain size solution                              %
%                                                                              %
%        Additionally outputs a suite of text files named by f1, f2,           %
%        f3, and f4.                                                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [PIdeq]=dFEMconpoly(f1,f2,f3,f4,d1);

path(path,'/home/jelsenbeck/');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Here we load the file specified in input d1 and read the grain size          %
% and differential stress, averaging over the horizontal span of the           %
% model space.                                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xfem1=flload(d1);

path(path,'/home/jelsenbeck/textfiles/');

xmax=4e5;
zmax=4e5;
xsize=1e3;
zsize=1e3;
dx=xmax/xsize;
dz=zmax/zsize;
x_fem=0:dx:xmax;
```

```
z_fem=-zmax:dz:0;
[X,Z] = meshgrid(x_fem,z_fem);
XMAX=size(X);
xx(1,:)=X(:);
xx(2,:)=Z(:);

[deq01 sigma01]=postinterp(xfem1,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeq=zeros(1,zsize+1);
PIz=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);
PIsigmatot=0;

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIz(1,cc)=Z(cc,1);
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(1,cc))==1 || PIdeq(cc)<250
        PIdeq(1,cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;

PIdeqtest=zeros(1,zsize+1);

xstr=X(1,:);
zstr=Z(:,1);

str=deq01_fem;
str2=sigma01_fem;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing output text files for both f1 and f2.                          %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grid='% Grid';
data='% Data (deq)';
blank=' ';

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/textfiles/',f1);
xfem1.functions = fcns;

fem0=xfem1;

n=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing grain size evolution loop for a maximum variation of          %
% 1 percent for grain size from one iteration to the next at each            %
% depth.                                                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while (max(abs((PIdeq-PIdeqtest)./PIdeq))>.01 || n<=2)

err(n)=max(abs((PIdeq-PIdeqtest)./PIdeq))

if n>100

   error(strcat('Solution has not converged. Reached max # of
iterations: ',num2str(n-1)));

end

fem=fem0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing and solving the grain size evolution model for                 %
% for iteration i.                                                            %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Global expressions
fem.expr = {'PR','-g*y*p0', ...
  'srII','.5*sqrt((ux-vy)^2+4*(uy+vx)^2)', ...
  'CH2O','abs(500*2/pi*(atan((y+65000)/2000)-pi/2))', ...
  'etawetdiff','(1/(A1*10^(-6*n1)))^(1/n1)*F1(y)^(m1/n1)*CH2O^(-
r1/n1)*exp((Q1+V1*PR)/(n1*R*(273+initT)))', ...
  'etawetdis','(1/(A2*10^(-6*n2)))^(1/n2)*F1(y)^(m2/n2)*srII^(1/n2-
1)*CH2O^(-r2/n2)*exp((Q2+V2*PR)/(n2*R*(273+initT)))', ...
  'etatot','min(etamax,(1/etawetdiff+1/etawetdis)^(-1))', ...
  'o','etatot*srII', ...
  'o2',num2str(PIsigmatot), ...
  'initT','abs(Tm*erf(y/(2*sqrt(k*(x0/u0)))))', ...
  'deq','(G0/(m1*A2))^(1/m1)*(mu/o2)^(n2/m1)*exp((Q2-
H)/(m1*R*initT))'};

% Extend mesh
fem.xmesh=meshextend(fem0);

fem.sol=femnlin(fem, ...
                'init',fem0.sol, ...
                'conjugate','on', ...
                'solcomp',{'u','p','v'}, ...
                'outcomp',{'u','p','v'}, ...
                'maxiter',40);
```

```
[deq01 sigma01]=postinterp(fem,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeqtest=PIdeq;

PIdeq=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(cc))==1 || PIdeq(cc)<250
        PIdeq(cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;
PIdeq=PIdeq-(PIdeq-PIdeqtest).*0.9;

str=deq01_fem;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Writing/adding iteration solution to output text files.                     %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
```

```
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f3,'at');
fprintf(fid,'%s\n',strcat('PIdeq #',num2str(n)));
fprintf(fid,'%24.20G\t',PIdeq);
fprintf(fid,'%s\n',blank);

save(f4,'err','-ascii','-tabs');

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/textfiles/',f1);
xfem1.functions = fcns;

n=n+1;
fem0=fem;

plot(PIdeq,PIz,'k',PIdeqtest,PIz,'kx');
hold on

end
```

dFEMconpolyn.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% dFEMconpolyn - Grain size evolution model for a dry water model              %
%                and Hall & Parmentier grain size evolution model              %
%              - Jimmy Elsenbeck 5/20/07                                       %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                                                                       %
%                                                                              %
%        f1 – deqfiles filename from ubergsize.m                               %
%        f2 – ofiles filename from ubergsize.m                                 %
%        f3 – PIfiles filename from ubergsize.m                                %
%        f4 – errfiles filename from ubergsize.m                               %
%        d1 – Comsol model filename from ubergsize.m                           %
%                                                                              %
% Output:                                                                      %
%                                                                              %
%        PIdeq – steady-state grain size solution                              %
%                                                                              %
%        Additionally outputs a suite of text files named by f1, f2,           %
%        f3, and f4.                                                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [PIdeq]=dFEMconpolyn(f1,f2,f3,f4,d1);

path(path,'/home/jelsenbeck/');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Here we load the file specified in input d1 and read the grain size          %
% and differential stress, averaging over the horizontal span of the           %
% model space.                                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xfem1=flload(d1);

xmax=4e5;
zmax=4e5;
xsize=1e3;
zsize=1e3;
dx=xmax/xsize;
dz=zmax/zsize;
x_fem=0:dx:xmax;
z_fem=-zmax:dz:0;
[X,Z] = meshgrid(x_fem,z_fem);
XMAX=size(X);
```

```
xx(1,:)=X(:);
xx(2,:)=Z(:);

[deq01 sigma01]=postinterp(xfem1,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeq=zeros(1,zsize+1);
PIz=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);
PIsigmatot=0;

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIz(1,cc)=Z(cc,1);
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(1,cc))==1 || PIdeq(cc)<250
        PIdeq(1,cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;

PIdeqtest=zeros(1,zsize+1);

xstr=X(1,:);
zstr=Z(:,1);

str=deq01_fem;
str2=sigma01_fem;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing output text files for both f1 and f2.                         %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grid='% Grid';
data='% Data (deq)';
blank=' ';

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
xfem1.functions = fcns;

fem0=xfem1;

n=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing grain size evolution loop for a maximum variation of          %
% 1 percent for grain size from one iteration to the next at each            %
% depth.                                                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while (max(abs((PIdeq-PIdeqtest)./PIdeq))>.01 || n<=2)

err(n)=max(abs((PIdeq-PIdeqtest)./PIdeq))

if n>100

   error(strcat('Solution has not converged.  Reached max # of
iterations: ',num2str(n-1)));

end

fem=fem0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing and solving the grain size evolution model for                 %
% a dry water model at iteration i.                                           %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Global expressions
fem.expr = {'PR','-g*y*p0', ...
  'srII','.5*sqrt((ux-vy)^2+4*(uy+vx)^2)', ...
  'CH2O','abs(500*2/pi*(atan((y+65000)/2000)-pi/2))', ...
  'etawetdiff','(1/(A1*10^(-6*n1)))^(1/n1)*F1(y)^(m1/n1)*nH2O^(-
r1/n1)*exp((Q1+V1*PR)/(n1*R*(273+initT)))', ...
  'etawetdis','(1/(A2*10^(-6*n2)))^(1/n2)*F1(y)^(m2/n2)*srII^(1/n2-
1)*nH2O^(-r2/n2)*exp((Q2+V2*PR)/(n2*R*(273+initT)))', ...
  'etatot','min(etamax,(1/etawetdiff+1/etawetdis)^(-1))', ...
  'o','etatot*srII', ...
  'o2',num2str(PIsigmatot), ...
  'initT','abs(Tm*erf(y/(2*sqrt(k*(x0/u0)))))', ...
  'deq','(G0/(m1*A2))^(1/m1)*(mu/o2)^(n2/m1)*exp((Q2-
H)/(m1*R*initT))'};

% Extend mesh
fem.xmesh=meshextend(fem0);

fem.sol=femnlin(fem, ...
                'init',fem0.sol, ...
                'conjugate','on', ...
                'solcomp',{'u','p','v'}, ...
                'outcomp',{'u','p','v'}, ...
                'maxiter',40);
```

```
[deq01 sigma01]=postinterp(fem,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeqtest=PIdeq;

PIdeq=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(cc))==1 || PIdeq(cc)<250
        PIdeq(cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;
PIdeq=PIdeq-(PIdeq-PIdeqtest).*0.9;

str=deq01_fem;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Writing/adding iteration solution to output text files.                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
```

```
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f3,'at');
fprintf(fid,'%s\n',strcat('PIdeq #',num2str(n)));
fprintf(fid,'%24.20G\t',PIdeq);
fprintf(fid,'%s\n',blank);

save(f4,'err','-ascii','-tabs');

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
% fcns{1}.filename='C:\Program Files\COMSOL32\deq1.txt';
xfem1.functions = fcns;

n=n+1;
fem0=fem;

plot(PIdeq,PIz,'k',PIdeqtest,PIz,'kx');
hold on

end
```

dFEMconpolyf.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% dFEMconpolyf - Grain size evolution model for a wet water model              %
%                and Hall & Parmentier grain size evolution model              %
%              - Jimmy Elsenbeck 5/20/07                                       %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                                                                       %
%                                                                              %
%        f1 – deqfiles filename from ubergsize.m                               %
%        f2 – ofiles filename from ubergsize.m                                 %
%        f3 – PIfiles filename from ubergsize.m                                %
%        f4 – errfiles filename from ubergsize.m                               %
%        d1 – Comsol model filename from ubergsize.m                           %
%                                                                              %
% Output:                                                                      %
%                                                                              %
%        PIdeq – steady-state grain size solution                              %
%                                                                              %
%        Additionally outputs a suite of text files named by f1, f2,           %
%        f3, and f4.                                                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [PIdeq]=dFEMconpolyf(f1,f2,f3,f4,d1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Here we load the file specified in input d1 and read the grain size          %
% and differential stress, averaging over the horizontal span of the           %
% model space.                                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

path(path,'/home/jelsenbeck/');

xfem1=flload(d1);

xmax=4e5;
zmax=4e5;
xsize=1e3;
zsize=1e3;
dx=xmax/xsize;
dz=zmax/zsize;
x_fem=0:dx:xmax;
z_fem=-zmax:dz:0;
[X,Z] = meshgrid(x_fem,z_fem);
XMAX=size(X);
```

```
xx(1,:)=X(:);
xx(2,:)=Z(:);

[deq01 sigma01]=postinterp(xfem1,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeq=zeros(1,zsize+1);
PIz=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);
PIsigmatot=0;

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIz(1,cc)=Z(cc,1);
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(1,cc))==1 || PIdeq(cc)<250
        PIdeq(1,cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;

PIdeqtest=zeros(1,zsize+1);

xstr=X(1,:);
zstr=Z(:,1);

str=deq01_fem;
str2=sigma01_fem;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing output text files for both f1 and f2.                          %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grid='% Grid';
data='% Data (deq)';
blank=' ';

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
xfem1.functions = fcns;

fem0=xfem1;

n=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing grain size evolution loop for a maximum variation of          %
% 1 percent for grain size from one iteration to the next at each            %
% depth.                                                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while (max(abs((PIdeq-PIdeqtest)./PIdeq))>.01 || n<=2)

err(n)=max(abs((PIdeq-PIdeqtest)./PIdeq))

if n>100

   error(strcat('Solution has not converged.  Reached max # of
iterations: ',num2str(n-1)));

end

fem=fem0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing and solving the grain size evolution model for                 %
% a wet water model at iteration i.                                           %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Global expressions
fem.expr = {'PR','-g*y*p0', ...
  'srII','.5*sqrt((ux-vy)^2+4*(uy+vx)^2)', ...
  'CH2O','abs(500*2/pi*(atan((y+65000)/2000)-pi/2))', ...
  'etawetdiff','(1/(A1*10^(-6*n1)))^(1/n1)*F1(y)^(m1/n1)*fH2O^(-
r1/n1)*exp((Q1+V1*PR)/(n1*R*(273+initT)))', ...
  'etawetdis','(1/(A2*10^(-6*n2)))^(1/n2)*F1(y)^(m2/n2)*srII^(1/n2-
1)*fH2O^(-r2/n2)*exp((Q2+V2*PR)/(n2*R*(273+initT)))', ...
  'etatot','min(etamax,(1/etawetdiff+1/etawetdis)^(-1))', ...
  'o','etatot*srII', ...
  'o2',num2str(PIsigmatot), ...
  'initT','abs(Tm*erf(y/(2*sqrt(k*(x0/u0)))))', ...
  'deq','(G0/(m1*A2))^(1/m1)*(mu/o2)^(n2/m1)*exp((Q2-
H)/(m1*R*initT))'};

% Extend mesh
fem.xmesh=meshextend(fem0);

fem.sol=femnlin(fem, ...
                'init',fem0.sol, ...
                'conjugate','on', ...
                'solcomp',{'u','p','v'}, ...
                'outcomp',{'u','p','v'}, ...
                'maxiter',40);
```

```
[deq01 sigma01]=postinterp(fem,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeqtest=PIdeq;

PIdeq=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(cc))==1 || PIdeq(cc)<250
        PIdeq(cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;
PIdeq=PIdeq-(PIdeq-PIdeqtest).*0.9;

str=deq01_fem;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Writing/adding iteration solution to output text files.                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
```

```
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f3,'at');
fprintf(fid,'%s\n',strcat('PIdeq #',num2str(n)));
fprintf(fid,'%24.20G\t',PIdeq);
fprintf(fid,'%s\n',blank);

save(f4,'err','-ascii','-tabs');

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
xfem1.functions = fcns;

n=n+1;
fem0=fem;

plot(PIdeq,PIz,'k',PIdeqtest,PIz,'kx');
hold on

end
```

dFEMconpolyNA.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% dFEMconpolyNA - Grain size evolution model for a dehydration water  %
%                 model and Austin & Evans grain size evolution model %
%               - Jimmy Elsenbeck 5/20/07                                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                                                                       %
%                                                                              %
%        f1 – deqfiles filename from ubergsize.m                               %
%        f2 – ofiles filename from ubergsize.m                                 %
%        f3 – PIfiles filename from ubergsize.m                                %
%        f4 – errfiles filename from ubergsize.m                               %
%        d1 – Comsol model filename from ubergsize.m                           %
%                                                                              %
% Output:                                                                      %
%                                                                              %
%        PIdeq – steady-state grain size solution                              %
%                                                                              %
%        Additionally outputs a suite of text files named by f1, f2,  %
%        f3, and f4.                                                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [PIdeq]=dFEMconpoly(f1,f2,f3,f4,d1);

path(path,'/home/jelsenbeck/');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Here we load the file specified in input d1 and read the grain size %
% and differential stress, averaging over the horizontal span of the  %
% model space.                                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xfem1=flload(d1);

xmax=4e5;
zmax=4e5;
xsize=1e3;
zsize=1e3;
dx=xmax/xsize;
dz=zmax/zsize;
x_fem=0:dx:xmax;
z_fem=-zmax:dz:0;
[X,Z] = meshgrid(x_fem,z_fem);
XMAX=size(X);
```

```
xx(1,:)=X(:);
xx(2,:)=Z(:);

[deq01 sigma01]=postinterp(xfem1,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeq=zeros(1,zsize+1);
PIz=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);
PIsigmatot=0;

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIz(1,cc)=Z(cc,1);
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(1,cc))==1 || PIdeq(cc)<250
        PIdeq(1,cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;

PIdeqtest=zeros(1,zsize+1);

xstr=X(1,:);
zstr=Z(:,1);

str=deq01_fem;
str2=sigma01_fem;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Initializing output text files for both f1 and f2.                           %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grid='% Grid';
data='% Data (deq)';
blank=' ';

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma);
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
fcns{2}.type='interp';
fcns{2}.name='F2';
fcns{2}.method='cubic';
fcns{2}.extmethod='extrap';
fcns{2}.filename='/home/jelsenbeck/deq750-5.0n.txt';
xfem1.functions = fcns;

fem0=xfem1;

n=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing grain size evolution loop for a maximum variation of          %
% 1 percent for grain size from one iteration to the next at each            %
% depth.                                                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while (max(abs((PIdeq-PIdeqtest)./PIdeq))>.01 || n<=2)

err(n)=max(abs((PIdeq-PIdeqtest)./PIdeq))

if n>100

   error(strcat('Solution has not converged.  Reached max # of
iterations: ',num2str(n-1)));

end

fem=fem0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Initializing and solving the grain size evolution model for                 %
% a dehydration water model at iteration i.                                   %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Global expressions
fem.expr = {'PR','-g*y*p0', ...
  'srII','.5*sqrt((ux-vy)^2+4*(uy+vx)^2)', ...
  'CH2O','abs(500*2/pi*(atan((y+65000)/2000)-pi/2))', ...
  'etawetdiff','(1/(A1*10^(-6*n1)))^(1/n1)*F1(y)^(m1/n1)*CH2O^(-
r1/n1)*exp((Q1+V1*PR)/(n1*R*(273+initT)))', ...
  'etawetdis','(1/(A2*10^(-6*n2)))^(1/n2)*F1(y)^(m2/n2)*srII^(1/n2-
1)*CH2O^(-r2/n2)*exp((Q2+V2*PR)/(n2*R*(273+initT)))', ...
  'etatot','min(etamax,(1/etawetdiff+1/etawetdis)^(-1))', ...
  'o','etatot*srII', ...
  'initT','abs(Tm*erf(y/(2*sqrt(k*(x0/u0)))))', ...
  'o2',num2str(PIsigmatot), ...

'deq','(etawetdis*c*gamma*Kg/(p1*lamb*o2^2*exp((H+PR*V1)/(R*(initT+273)
))))^(1/(p1+1))'};

% Extend mesh
fem.xmesh=meshextend(fem0);

fem.sol=femnlin(fem, ...
                'init',fem0.sol, ...
                'conjugate','on', ...
                'solcomp',{'u','p','v'}, ...
                'outcomp',{'u','p','v'}, ...
```

```
                'maxiter',40);

[deq01 sigma01]=postinterp(fem,'deq','o',xx);

deq01_fem=reshape(deq01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));

PIdeqtest=PIdeq;

PIdeq=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);

for cc=1:zsize+1

    for bb=1:xsize+1

    PIsigma(1,cc)=PIsigma(1,cc)+sigma01_fem(cc,bb);
    PIdeq(1,cc)=PIdeq(1,cc)+deq01_fem(cc,bb);

    if isnan(deq01_fem(cc,bb))==1
        deq01_fem(cc,bb)=0;
    end

    end

    PIdeq(1,cc)=PIdeq(1,cc)/bb;
    PIsigma(1,cc)=PIsigma(1,cc)/bb;

    PIsigmatot=PIsigmatot+PIsigma(1,cc);

    if isnan(PIdeq(cc))==1 || PIdeq(cc)<250
        PIdeq(cc)=250;
    end

end

PIsigmatot=PIsigmatot/cc;
PIdeq=PIdeq-(PIdeq-PIdeqtest).*0.9;

str=deq01_fem;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Writing/adding iteration solution to output text files.                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen(f1, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
```

```
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIdeq); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid

fid = fopen(f2, 'wt');
fprintf(fid, '%s\n',grid);
fprintf(fid, '%24.20G\t',zstr);
fprintf(fid, '%s\n',blank);
fprintf(fid, '%s\n',data);
fprintf(fid, '%24.20G\t', PIsigma); % str(i,:));
fprintf(fid, '%s\n',blank);
fclose(fid);

clear fid;

fid = fopen(f3,'at');
fprintf(fid,'%s\n',strcat('PIdeq #',num2str(n)));
fprintf(fid,'%24.20G\t',PIdeq);
fprintf(fid,'%s\n',blank);

save(f4,'err','-ascii','-tabs');

% Functions
clear fcns
fcns{1}.type='interp';
fcns{1}.name='F1';
fcns{1}.method='cubic';
fcns{1}.extmethod='extrap';
fcns{1}.filename=strcat('/home/jelsenbeck/',f1);
fcns{2}.type='interp';
fcns{2}.name='F2';
fcns{2}.method='cubic';
fcns{2}.extmethod='extrap';
fcns{2}.filename='/home/jelsenbeck/deq750-5.0n.txt';
xfem1.functions = fcns;

n=n+1;
fem0=fem;

plot(PIdeq,PIz,'k',PIdeqtest,PIz,'kx');
hold on

end
```

3. Attenuation/Shear wave velocity calculations and data visualization

uberdeform.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% uberdeform - Calls function deformread for Comsol models of varying         %
%              plate age/spreading rate/water content/grain size              %
%              evolution model. Additionally plots the loaded data.           %
%              5/20/07 - Jimmy Elsenbeck                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Input:                                                                      %
%                                                                             %
%        data - lists input steady-state grain size solution models           %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

data=['1Ddeform050+CH2O-1e25tri-hidconv-1.0n.mph'; ...
      '1Ddeform120+CH2O-1e25tri-hidconv-1.0n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconv-1.0n.mph'; ...
      '1Ddeform500+CH2O-1e25tri-hidconv-1.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconv-1.0n.mph'; ...
      '1Ddeform050+CH2O-1e25tri-hidconv-2.5n.mph'; ...
      '1Ddeform120+CH2O-1e25tri-hidconv-2.5n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconv-2.5n.mph'; ...
      '1Ddeform500+CH2O-1e25tri-hidconv-2.5n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconv-2.5n.mph'; ...
      '1Ddeform050+CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform120+CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform250+CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform500+CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform050-CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform120-CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform050+fH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform120+fH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform750-CH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform750+fH2O-1e25tri-hidconv-5.0n.mph'; ...
      '1Ddeform750-CH2O-1e25tri-F&J4con-5.0n.mph'; ...
      '1Ddeform750+CH2O-1e25tri-hidconv5.0N3.mph'];

count=size(data);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Loading of data from each model.                                             %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=[1:count(1)]

    [U(i,:),T(i,:),Z(i,:),eta(i,:),disl(i,:),diff(i,:), ...
     etarat(i,:),teq(i,:),sigma(i,:),srII(i,:),deq(i,:), ...
     deq6(i,:),Q(i,:),Q1(i,:),Q2(i,:),Q3(i,:),Q4(i,:), ...
     Q5(i,:),Q6(i,:),Qn(i,:),Q1n(i,:),Q2n(i,:),Q3n(i,:), ...
     Q4n(i,:),Q5n(i,:),Q6n(i,:),V(i,:),V1(i,:),V2(i,:), ...
     V3(i,:),V4(i,:),V5(i,:),V6(i,:),CH2O(i,:),G(i,:), ...
     G1(i,:),G2(i,:),G3(i,:),G4(i,:),G5(i,:),Ga(i,:), ...
     G1a(i,:),G2a(i,:),G3a(i,:),G4a(i,:),G5a(i,:),Gn(i,:), ...
     G1n(i,:),G2n(i,:),G3n(i,:),G4n(i,:),G5n(i,:),G6n(i,:), ...
     Gna(i,:),G1na(i,:),G2na(i,:),G3na(i,:),G4na(i,:), ...
     G5na(i,:),G6na(i,:),fH2O(i,:),Th3(i,:),Th4(i,:), ...
     J31(i,:),J41(i,:),dJ3(i,:),dJ4(i,:),Tl3(i,:),Tl4(i,:), ...
     int3(i,:),int4(i,:),Gu(i,:),Qsb(i,:),Qsb1(i,:), ...
     Qsb2(i,:),Qsb3(i,:),Qsb4(i,:),Qsb5(i,:),Qsb6(i,:), ...
     Vsb(i,:),Vsb1(i,:),Vsb2(i,:),Vsb3(i,:),Vsb4(i,:), ...
     Vsb5(i,:),Vsb6(i,:)]=deformread(data(i,:));

    [Vx(i) Zx(i)]=min(V(i,:)./1e3);
    [Vx1(i) Zx1(i)]=min(V1(i,:)./1e3);
    [Vx2(i) Zx2(i)]=min(V2(i,:)./1e3);
    [Vx3(i) Zx3(i)]=min(V3(i,:)./1e3);
    [Vx4(i) Zx4(i)]=min(V4(i,:)./1e3);
    [Vx5(i) Zx5(i)]=min(V5(i,:)./1e3);

    [Qx(i) Zy(i)]=min(Q(i,:));
    [Qx1(i) Zy1(i)]=min(Q1(i,:));
    [Qx2(i) Zy2(i)]=min(Q2(i,:));
    [Qx3(i) Zy3(i)]=min(Q3(i,:));
    [Qx4(i) Zy4(i)]=min(Q4(i,:));
    [Qx5(i) Zy5(i)]=min(Q5(i,:));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Digitized PA5 velocity model from Gaherty et al., 1996 & 1999.               %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PA5a=[0 .92 .92 3.68 3.68 4.65 4.67 4.37 4.26 4.29 4.84];
PA5b=[0 .92 .92 3.68 3.68 4.84 4.83 4.56 4.34 4.29 4.84];
PA5=(PA5a+PA5b)./2;
PAZ=[0 5 5.2 5.2 12 12 68 68 166 166 400];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Digitized velocity (V) & attenuation (A) models from Nishimura &             %
% Forsyth, 1989. #1 is for 0-4 Myr, #2 is 4-20 Myr, #3 is 20-52 Myr,           %
% and #4 is 52-110 Myr.                                                        %
%                                                                              %
% A is the ratio of Vsh to Vsv.                                                %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

VZm=[15 25 35 45 55 65 75 85 95 105 115 125 135 145 155 165 ...
     175 185 195 205 215 225 235 245 255 265 275 285 295 305 ...
     315 325 335 345 355 365 375 385 395];
V1m=[4.4316 4.2212 4.0463 3.9817 3.9707 3.9782 3.9744 3.9981 ...
     4.0274 4.0580 4.0869 4.1138 4.1382 4.1624 4.1962 4.2234 ...
     4.2683 4.3101 4.3479 4.3862 4.4240 4.4494 4.4647 4.4789 ...
     4.4873 4.4838 4.4814 4.4802 4.4923 4.5045 4.5178 4.5392 ...
     4.5786 4.6181 4.6565 4.6989 4.7442 4.7897 4.8364];
V2m=[4.5835 4.5231 4.4419 4.3728 4.3005 4.2349 4.1681 4.1416 ...
     4.1318 4.1331 4.1411 4.1527 4.1660 4.1824 4.2106 4.2334 ...
     4.2755 4.3145 4.3509 4.3893 4.4267 4.4524 4.4675 4.4829 ...
     4.4917 4.4886 4.4868 4.4851 4.4975 4.5090 4.5225 4.5440 ...
     4.5825 4.6221 4.6596 4.7021 4.7465 4.7921 4.8380];
V3m=[4.6146 4.5908 4.5485 4.5084 4.4468 4.3778 4.2970 4.2486 ...
     4.2145 4.1927 4.1813 4.1774 4.1791 4.1881 4.2126 4.2330 ...
     4.2758 4.3155 4.3537 4.3938 4.4322 4.4593 4.4751 4.4908 ...
     4.4996 4.4974 4.4953 4.4933 4.5054 4.5165 4.5287 4.5500 ...
     4.5883 4.6267 4.6641 4.7057 4.7502 4.7947 4.8399];
V4m=[4.6294 4.6326 4.6315 4.6293 4.5909 4.5303 4.4447 4.3804 ...
     4.3242 4.2785 4.2438 4.2185 4.2020 4.1969 4.2106 4.2237 ...
     4.2610 4.2977 4.3352 4.3744 4.4137 4.4417 4.4586 4.4759 ...
     4.4863 4.4851 4.4843 4.4838 4.4961 4.5087 4.5216 4.5441 ...
     4.5827 4.6215 4.6604 4.7022 4.7472 4.7923 4.8377];

At1=[1.0268 1.0323 1.0303 1.0280 1.0267 1.0259 1.0253 1.0249 ...
     1.0246 1.0242 1.0236 1.0228 1.0218 1.0206 1.0193 1.0180 ...
     1.0166 1.0152 1.0138 1.0125 1.0112 1.0100 1.0088 1.0078 ...
     1.0068 1.0059 1.0051 1.0044 1.0038 1.0032 1.0028 1.0023 ...
     1.0020 1.0017 1.0014 1.0012 1.0010 1.0008 1.0006];
At2=[1.0260 1.0338 1.0354 1.0369 1.0394 1.0422 1.0448 1.0466 ...
     1.0472 1.0465 1.0442 1.0406 1.0360 1.0305 1.0247 1.0189 ...
     1.0133 1.0082 1.0038 1.0002 0.9973 0.9951 0.9937 0.9928 ...
     0.9923 0.9922 0.9924 0.9927 0.9931 0.9936 0.9942 0.9947 ...
     0.9953 0.9958 0.9963 0.9967 0.9971 0.9976 0.9984];
At3=[1.0410 1.0547 1.0568 1.0562 1.0560 1.0563 1.0569 1.0577 ...
     1.0582 1.0582 1.0574 1.0556 1.0529 1.0495 1.0455 1.0411 ...
     1.0366 1.0322 1.0280 1.0242 1.0207 1.0176 1.0150 1.0127 ...
     1.0107 1.0090 1.0076 1.0064 1.0054 1.0046 1.0039 1.0033 ...
     1.0029 1.0025 1.0021 1.0019 1.0016 1.0013 1.0009];
```

```
At4=[1.0401 1.0563 1.0607 1.0610 1.0611 1.0616 1.0624 1.0635 ...
     1.0648 1.0658 1.0663 1.0659 1.0646 1.0624 1.0591 1.0551 ...
     1.0506 1.0457 1.0406 1.0356 1.0308 1.0263 1.0222 1.0185 ...
     1.0153 1.0124 1.0100 1.0079 1.0061 1.0046 1.0034 1.0024 ...
     1.0016 1.0009 1.0004 1.0001 0.9998 0.9996 0.9997];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Scaling of radial anisotropy to fit in Figure 16 below (Fig. 8 in           %
% the thesis).                                                                %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

At1=(At1-1).*50;
At2=(At2-1).*50;
At3=(At3-1).*50;
At4=(At4-1).*50;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Digitized SW02 attenuation model from Selby & Woodhouse, 2002.              %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SW0=4.207;
SW1=5.495;
SW02a=[5.480 4.415 4.368 4.321 4.316 4.329 4.345 4.376 4.475 ...
       4.675 4.722 4.751 4.769 4.779 4.779];
SW02=(SW02a-SW0)./(SW1-SW0).*400;
SWZa=[3.507 3.785 3.809 3.866 3.884 3.928 3.952 3.983 4.053 ...
      4.288 4.374 4.460 4.590 4.704 4.873];
SWZ0=3.465;
SWZ1=4.873;
SWZ=(SWZ0-SWZa)./(SWZ1-SWZ0).*350-50;
SWZ=SWZ.*-1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Digitized PREM attenuation model from Dziewonski & Anderson, 1981.          %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

QPREM=[600 600 600 80 80 143 143];
QPREMZ=[25 25 80 80 220 220 400];

Z=Z./1e3;
l=length(Z(1,:));
V=V./1e3;
X=[5e6 12e6 25e6 50e6 75e6];
k=1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Calculation of Faul & Jackson grain size profile, viscosity, and             %
% dominant deformation mechanism ratio for both a dry and dehydrated           %
%
% mantle.                                                                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A1=3.33e5;
r1=1;
n1=1;
m1=3;
Q1=335e3;
V1=4e-6;
A2=30;
r2=1.2;
n2=3.5;
m2=0;
Q2=480e3;
V2=1.1e-5;
R=8.314;

zstr=Z(15,:)'.*1e3;

deq6(1,:)=10.^(0.8.*2./pi.*atan((zstr.*-1-240500)/5000-pi./2)+3.9);

FJetadiff=(1./(1.5e9.*10.^(-
6*n1))).^(1/n1).*(deq6(1,:)).^(m1/n1).*CH2O(20,:).^(-
r1/n1).*exp((375e3+V1.*Z(20,:).*1e3.*9.81.*3300)./(n1.*R.*(273+T(20,:))
));
FJetadisl=(1./(1.1e5.*10.^(-
6*n2))).^(1/n2).*(deq6(1,:)).^(m2/n2).*srII(20,:).^(1/n2-
1).*exp((530e3+V2.*Z(20,:).*1e3.*9.81.*3300)./(n2.*R.*(273+T(20,:))));
FJetadry=min(1e25,(1./FJetadiff+1./FJetadisl).^(-1));

FJ1etadiff=(1./(A1.*10.^(-
6*n1))).^(1/n1).*(deq6(1,:)).^(m1/n1).*CH2O(15,:).^(-
r1/n1).*exp((Q1+V1.*Z(15,:).*1e3.*9.81.*3300)./(n1.*R.*(273+T(15,:))));
FJ1etadisl=(1./(A2.*10.^(-
6*n2))).^(1/n2).*(deq6(1,:)).^(m2/n2).*srII(15,:).^(1/n2-
1).*CH2O(15,:).^(-
r2/n2).*exp((Q2+V2.*Z(15,:).*1e3.*9.81.*3300)./(n2.*R.*(273+T(15,:))));
FJetawet=min(1e25,(1./FJ1etadiff+1./FJ1etadisl).^(-1));

FJetarat=FJetadiff./FJetadisl;
FJ1etarat=FJ1etadiff./FJ1etadisl;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Setting up Characteristic Timescale in order to generate dashed              %
% portions in figures for depths that predict a timescale greater              %
% than ~10% of plate age.                                                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:count(1)

    if i==16 || i==18
        k=1;
    elseif i==17 || i==19
        k=2;
    elseif i>19
        k=5;
    end

    teqmark(i,1:l)=.1333333*X(k);
    tct(i,1)=l;
    tct(i,2)=l;

    for j=1:l-1

        if teq(i,j)>teqmark(i,j) && teq(i,j+1)<teqmark(i,j)
            tct(i,1)=j;
        elseif teq(i,j)<teqmark(i,j) && teq(i,j+1)>teqmark(i,j)
            tct(i,2)=j;
        end

    end

    if k==5 && i<16
        k=1;
    elseif i<16
        k=k+1;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Preparing particular variables for log plotting.                             %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

deq=deq./1e4;
deq6=deq6./1e4;

Z=Z.*-1;
Vsb=Vsb./1e3;
```

```
FJetarat=log10(FJetarat);
etarat=log10(etarat);
eta=log10(eta);
FJetadry=log10(FJetadry);
FJetawet=log10(FJetawet);
teq=log10(teq);
deq=log10(deq.*10);
deq6=log10(deq6.*10);
teqmark=log10(teqmark);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Figures 1 - 17                                                               %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(1);
subplot(1,3,1)
plot(U(15,:).*3600.*24.*364.25.*1000,Z(15,:),'k','LineWidth',2);
xlabel('Spreading Half-Rate (mm/yr)','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([0 50 0 400]);
set(gca,'YDir','reverse','XTick',[0 25 50], ...
    'plotboxaspectratio',[2 2 2]);

subplot(1,3,2)
patch([0 0 1050 1050],[0 65 65 0],[.7 .7 .7],'FaceAlpha',.7, ...
      'EdgeAlpha',0);
hold on;
plot(CH2O(15,:),Z(15,:),'c',CH2O(16,:),Z(16,:),'g', ...
     CH2O(18,:),Z(18,:),'r','LineWidth',2),
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
text(100,100,'Dry','FontSize',8,'FontWeight','bold');
text(255,45,'Dehydration','FontSize',8,'FontWeight','bold');
text(805,25,'Wet','FontSize',8,'FontWeight','bold');
xlabel('H / 10^6 Si','FontSize',8,'FontWeight','bold');
axis([0 1050 0 400]);
hold off;

subplot(1,3,3)
plot(T(11,:),Z(11,:),'c',T(13,:),Z(13,:),'r', ...
     T(15,:),Z(15,:),'k','LineWidth',2);
xlabel('Temp (\circC)','FontSize',8,'FontWeight','bold');
text(1355,20,'5 Myr','FontSize',8,'FontWeight','bold');
text(1055,55,'25 Myr','FontSize',8,'FontWeight','bold');
text(400,85,'75 Myr','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);

figure(2)
subplot(2,3,1)
plot(etarat(1,:),Z(1,:),'c',etarat(3,:),Z(3,:),'r', ...
     etarat(5,:),Z(5,:),'k','LineWidth',2);
```

```
hold on;
plot(ones(1,l),Z(1,:),'k--');
hold off;
xlabel('Log10 of Dislocation Strain Rate/Diffusion Strain Rate', ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-4 4 0 400]);
set(gca,'YDir','reverse');
patch([-4 -4 4 4],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);

subplot(2,3,2)
plot(eta(1,:),Z(1,:),'c',eta(3,:),Z(3,:),'r', ...
     eta(5,:),Z(5,:),'k','LineWidth',2);
xlabel('Log10 of Viscosity (Pa*s)','FontSize',8,'FontWeight','bold');
axis([18 25 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,3)
plot(Q(1,1:tct(1,1)),Z(1,1:tct(1,1)),'c--',Q(1,tct(1,1):tct(1,2)), ...
     Z(1,tct(1,1):tct(1,2)),'c',Q(1,tct(1,2):l),Z(1,tct(1,2):l), ...
     'c--',Q(3,1:tct(3,1)),Z(3,1:tct(3,1)),'r--', ...
     Q(3,tct(3,1):tct(3,2)),Z(3,tct(3,1):tct(3,2)),'r', ...
     Q(3,tct(3,2):l),Z(3,tct(3,2):l),'r--',Q(5,1:tct(5,1)), ...
     Z(5,1:tct(5,1)),'k--',Q(5,tct(5,1):tct(5,2)), ...
     Z(5,tct(5,1):tct(5,2)),'k',Q(5,tct(5,2):l),Z(5,tct(5,2):l), ...
     'k--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,4)
plot(teq(1,:),Z(1,:),'c',teq(3,:),Z(3,:),'r', ...
teq(5,:),Z(5,:),'k','LineWidth',2);
hold on;
plot(teqmark(1,:),Z(1,:),'c--',teqmark(3,:),Z(3,:),'r--', ...
     teqmark(5,:),Z(5,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
       8,'FontWeight','bold')
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,5)
plot(deq(1,:),Z(1,:),'c',deq(3,:),Z(3,:),'r', ...
     deq(5,:),Z(5,:),'k','LineWidth',2);
xlabel('Log10 of Eq. Grain Size (mm)','FontSize', ...
       8,'FontWeight','bold');
set(gca,'YDir','reverse');axis([-2 2 0 400]);
legend('Location','sw','5 Mya @ 1 cm/yr Half-rate','25 Mya','75 Mya');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
```

```
subplot(2,3,6)
plot(Vsb(1,1:tct(1,1)),Z(1,1:tct(1,1)),'c--', ...
     Vsb(1,tct(1,1):tct(1,2)),Z(1,tct(1,1):tct(1,2)), ...
     'c',Vsb(1,tct(1,2):l),Z(1,tct(1,2):l),'c--', ...
     Vsb(3,1:tct(3,1)),Z(3,1:tct(3,1)),'r--', ...
     Vsb(3,tct(3,1):tct(3,2)),Z(3,tct(3,1):tct(3,2)),'r', ...
     Vsb(3,tct(3,2):l),Z(3,tct(3,2):l),'r--', ...
     Vsb(5,1:tct(5,1)),Z(5,1:tct(5,1)),'k--', ...
     Vsb(5,tct(5,1):tct(5,2)), Z(5,tct(5,1):tct(5,2)),'k', ...
     Vsb(5,tct(5,2):l),Z(5,tct(5,2):l),'k--','LineWidth',2);
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
axis([3.9 5 0 400]);
set(gca,'YDir','reverse');

figure(3)
subplot(2,3,1)
plot(etarat(6,:),Z(6,:),'c',etarat(8,:),Z(8,:),'r', ...
     etarat(10,:),Z(10,:),'k','LineWidth',2);
hold on;
plot(ones(1,l),Z(1,:),'k--');
hold off;
xlabel('Log10 of Dislocation Strain Rate/Diffusion Strain Rate', ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-4 4 0 400]);
set(gca,'YDir','reverse');
patch([-4 -4 4 4],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);

subplot(2,3,2)
plot(eta(6,:),Z(6,:),'c',eta(8,:),Z(8,:),'r', ...
     eta(10,:),Z(10,:),'k','LineWidth',2);
xlabel('Log10 of Viscosity (Pa*s)','FontSize',8,'FontWeight','bold');
axis([18 25 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,3)
plot(Q(6,1:tct(6,1)),Z(6,1:tct(6,1)),'c--',Q(6,tct(6,1):tct(6,2)), ...
     Z(6,tct(6,1):tct(6,2)),'c',Q(6,tct(6,2):l),Z(6,tct(6,2):l), ...
     'c--',Q(8,1:tct(8,1)),Z(8,1:tct(8,1)),'r--', ...
     Q(8,tct(8,1):tct(8,2)),Z(8,tct(8,1):tct(8,2)),'r', ...
     Q(8,tct(8,2):l),Z(8,tct(8,2):l),'r--', ...
     Q(10,1:tct(10,1)),Z(10,1:tct(10,1)),'k--', ...
     Q(10,tct(10,1):tct(10,2)), Z(10,tct(10,1):tct(10,2)),'k', ...
     Q(10,tct(10,2):l),Z(10,tct(10,2):l),'k--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,4)
plot(teq(6,:),Z(6,:),'c',teq(8,:),Z(8,:),'r', ...
     teq(10,:),Z(10,:),'k','LineWidth',2);
hold on;
plot(teqmark(6,:),Z(6,:),'c--', teqmark(8,:),Z(8,:),'r--', ...
```

```
     teqmark(10,:),Z(10,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
       8,'FontWeight','bold')
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,5)
plot(deq(6,:),Z(6,:),'c',deq(8,:),Z(8,:),'r', ...
     deq(10,:),Z(10,:),'k','LineWidth',2);
xlabel('Log10 of Eq. Grain Size (mm)','FontSize', ...
       8,'FontWeight','bold');
set(gca,'YDir','reverse');axis([-2 2 0 400]);
legend('Location','sw','5 Mya @ 2.5 cm/yr Half-rate','25 Mya', ...
       '75 Mya');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(2,3,6)
plot(Vsb(6,1:tct(6,1)),Z(6,1:tct(6,1)),'c--', ...
     Vsb(6,tct(6,1):tct(6,2)),Z(6,tct(6,1):tct(6,2)), ...
     'c',Vsb(6,tct(6,2):l),Z(6,tct(6,2):l),'c--', ...
     Vsb(8,1:tct(8,1)),Z(8,1:tct(8,1)),'r--', ...
     Vsb(8,tct(8,1):tct(8,2)),Z(8,tct(8,1):tct(8,2)),'r', ...
     Vsb(8,tct(8,2):l),Z(8,tct(8,2):l),'r--', ...
     Vsb(10,1:tct(10,1)),Z(10,1:tct(10,1)),'k--', ...
     Vsb(10,tct(10,1):tct(10,2)), Z(10,tct(10,1):tct(10,2)),'k', ...
     Vsb(10,tct(10,2):l),Z(10,tct(10,2):l),'k--','LineWidth',2);
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
axis([3.9 5 0 400]);
set(gca,'YDir','reverse');

figure(4)
subplot(3,3,1)
patch([-4 -4 4 4],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);
hold on;
plot(etarat(11,:),Z(11,:),'c',etarat(13,:),Z(13,:),'r', ...
     etarat(15,:),Z(15,:),'k','LineWidth',2);
hold on;
plot(zeros(1,l),Z(1,:),'k--');
hold off;
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-2 4 0 400]);
set(gca,'YDir','reverse');
hold off;

subplot(3,3,2)
plot(eta(11,:),Z(11,:),'c',eta(13,:),Z(13,:),'r', ...
     eta(15,:),Z(15,:),'k','LineWidth',2);
xlabel('Log10(\eta) (Pa*s)','FontSize',8,'FontWeight','bold');
```

```
axis([18 25 0 400]);
set(gca,'YDir','reverse','XTick',[19 21 23 25]);

subplot(3,3,3)
plot(Q(11,1:tct(11,1)),Z(11,1:tct(11,1)),'c--', ...
     Q(11,tct(11,1):tct(11,2)), Z(11,tct(11,1):tct(11,2)),'c', ...
     Q(11,tct(11,2):l),Z(11,tct(11,2):l),'c--', ...
     Q(13,1:tct(13,1)),Z(13,1:tct(13,1)),'r--', ...
     Q(13,tct(13,1):tct(13,2)), Z(13,tct(13,1):tct(13,2)),'r', ...
     Q(13,tct(13,2):l),Z(13,tct(13,2):l),'r--', ...
     Q(15,1:tct(15,1)),Z(15,1:tct(15,1)),'k--', ...
     Q(15,tct(15,1):tct(15,2)), Z(15,tct(15,1):tct(15,2)),'k', ...
     Q(15,tct(15,2):l),Z(15,tct(15,2):l),'k--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');

subplot(3,3,4)
plot(teq(11,:),Z(11,:),'c',teq(13,:),Z(13,:),'r', ...
     teq(15,:),Z(15,:),'k','LineWidth',2);
hold on;
plot(teqmark(11,:),Z(11,:),'c--', ...
     teqmark(13,:),Z(13,:),'r--',teqmark(15,:),Z(15,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
       8,'FontWeight','bold'),
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse','XTick',[5 6 7 8 9]);

subplot(3,3,5)
plot(deq(11,:),Z(11,:),'c',deq(13,:),Z(13,:),'r', ...
     deq(15,:),Z(15,:),'k','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');axis([-1 2 0 400]);
legend('Location','sw','5 Mya @ 5 cm/yr Half-rate','25 Mya','75 Mya');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(3,3,6)
plot(Vsb(11,1:tct(11,1)),Z(11,1:tct(11,1)),'c--', ...
     Vsb(11,tct(11,1):tct(11,2)),Z(11,tct(11,1):tct(11,2)), ...
     'c',Vsb(11,tct(11,2):l),Z(11,tct(11,2):l),'c--', ...
     Vsb(13,1:tct(13,1)),Z(13,1:tct(13,1)),'r--', ...
     Vsb(13,tct(13,1):tct(13,2)), Z(13,tct(13,1):tct(13,2)),'r', ...
     Vsb(13,tct(13,2):l),Z(13,tct(13,2):l),'r--', ...
     Vsb(15,1:tct(15,1)),Z(15,1:tct(15,1)),'k--', ...
     Vsb(15,tct(15,1):tct(15,2)), Z(15,tct(15,1):tct(15,2)),'k', ...
     Vsb(15,tct(15,2):l),Z(15,tct(15,2):l),'k--','LineWidth',2);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold'),
axis([4 5 0 400]);
set(gca,'YDir','reverse');
```

```
figure(5)
subplot(2,1,1)
plot(1e4./(T(20,:)+273),Gna(20,:),'b','LineWidth',3);
hold on;
plot(1e4./(T(20,:)+273),G2na(20,:),'r',1e4./(T(20,:)+273), ...
     G3na(20,:),'c',1e4./(T(20,:)+273),G4na(20,:),'k','LineWidth',1);
ylabel('Shear Modulus (GPa)','FontSize',8,'FontWeight','bold');
axis([5.5 8 30 70]);
set(gca,'plotboxaspectratio',[1 1 1],'XTick',[5.5 6 6.5 7 7.5 8]);

subplot(2,1,2)
plot(1e4./(T(20,:)+273),log10(1./Qn(20,:)),'b','LineWidth',3);
hold on;
plot(1e4./(T(20,:)+273),log10(1./Q2n(20,:)),'r',1e4./(T(20,:)+273), ...
     log10(1./Q3n(20,:)),'c', 1e4./(T(20,:)+273), ...
     log10(1./Q4n(20,:)),'k','LineWidth',1)
xlabel('1/T(K) x 1e4','FontSize',8,'FontWeight','bold');
ylabel('log10(1/Q)','FontSize',8,'FontWeight','bold');
legend('H&P Dry GS model','0.1 mm','1 mm','10 mm');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
axis([5.5 8 -2.1 -.4]);
set(gca,'plotboxaspectratio',[1 1 1],'XTick',[5.5 6 6.5 7 7.5 8]);

figure(6)
subplot(2,2,1)
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q3(15,:),Z(15,:),'r', ...
     Q4(15,:),Z(15,:),'c',Q5(15,:),Z(15,:),'k','LineWidth',1);
hold on;
plot(Q(15,:),Z(15,:),'b','LineWidth',3);
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
legend('Location','nw','PREM','SW02','1 mm','1 cm','5 cm', ...
       'Eq. Grain Size @ 75 Mya & 5 cm/yr Half-rate');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');
set(gca,'plotboxaspectratio',[1 1 1]);

subplot(2,2,2)
plot(PA5,PAZ,'k:',V4m,VZm,'k-.',Vsb3(15,:)./1e3,Z(15,:),'r', ...
     Vsb4(15,:)./1e3,Z(15,:),'c',Vsb5(15,:)./1e3,Z(15,:), ...
     'k','LineWidth',1);
hold on;
plot(Vsb(15,:),Z(15,:),'b','LineWidth',3);
xlabel('Shear-wave Velocity (km/s)','FontSize', ...
       8,'FontWeight','bold');
axis([3.7 5 0 400]);
set(gca,'YDir','reverse');
legend('Location','sw','PA5','52-110 Myr'); legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'plotboxaspectratio',[1 1 1]);
```

```
figure(7)
subplot(2,3,1)
plot(etarat(11,:),Z(11,:),'c',etarat(16,:),Z(16,:),'g', ...
     etarat(18,:),Z(18,:),'r','LineWidth',2);
hold on;
plot(ones(1,l),Z(1,:),'k--');
hold off;
xlabel('Log10 of Dislocation Strain Rate/Diffusion Strain Rate', ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
legend('Dehydration @ 5 Mya & 5 cm/yr Half-rate','Dry','Wet');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
axis([-3 3 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,2)
plot(eta(11,:),Z(11,:),'c',eta(16,:),Z(16,:),'g',eta(18,:), ...
     Z(18,:),'r','LineWidth',2);
xlabel('Log10 of Viscosity (Pa*s)','FontSize',8,'FontWeight','bold');
axis([18 25 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,3)
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q(11,1:tct(11,1)), ...
     Z(11,1:tct(11,1)),'c--',Q(11,tct(11,1):tct(11,2)), ...
     Z(11,tct(11,1):tct(11,2)),'c',Q(11,tct(11,2):l), ...
     Z(11,tct(11,2):l),'c--',Q(16,1:tct(16,1)), ...
     Z(16,1:tct(16,1)),'g--',Q(16,tct(16,1):tct(16,2)), ...
     Z(16,tct(16,1):tct(16,2)),'g',Q(16,tct(16,2):l), ...
     Z(16,tct(16,2):l),'g--',Q(18,1:tct(18,1)), ...
     Z(18,1:tct(18,1)),'r--',Q(18,tct(18,1):tct(18,2)), ...
     Z(18,tct(18,1):tct(18,2)),'r',Q(18,tct(18,2):l), ...
     Z(18,tct(18,2):l),'r--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');
legend('Location','se','PREM','SW02');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(2,3,4)
plot(teq(11,:),Z(11,:),'c',teq(16,:),Z(16,:),'g',teq(18,:), ...
     Z(18,:),'r','LineWidth',2);
hold on;
plot(teqmark(11,:),Z(11,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)', ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse');
```

```
subplot(2,3,5),
plot(deq(11,:),Z(11,:),'c',deq(16,:),Z(16,:),'g',deq(18,:), ...
     Z(18,:),'r','LineWidth',2);
xlabel('Log10 of Eq. Grain Size (mm)','FontSize',8, ...
       'FontWeight','bold');
set(gca,'YDir','reverse');

subplot(2,3,6)
plot(PA5,PAZ,'k:',V1m,VZm,'k-.',Vsb(11,1:tct(11,1)), ...
     Z(11,1:tct(11,1)),'c--',Vsb(11,tct(11,1):tct(11,2)), ...
     Z(11,tct(11,1):tct(11,2)),'c',Vsb(11,tct(11,2):l), ...
     Z(11,tct(11,2):l),'c--',Vsb(16,1:tct(16,1)), ...
     Z(16,1:tct(16,1)),'g--',Vsb(16,tct(16,1):tct(16,2)), ...
     Z(16,tct(16,1):tct(16,2)),'g',Vsb(16,tct(16,2):l), ...
     Z(16,tct(16,2):l),'g--',Vsb(18,1:tct(18,1)), ...
     Z(18,1:tct(18,1)),'r--',Vsb(18,tct(18,1):tct(18,2)), ...
     Z(18,tct(18,1):tct(18,2)),'r',Vsb(18,tct(18,2):l), ...
     Z(18,tct(18,2):l),'r--','LineWidth',2);
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
axis([3.9 5 0 400]);
set(gca,'YDir','reverse');
legend('Location','sw','PA5','0-4 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

figure(8)
subplot(2,3,1)
plot(etarat(12,:),Z(12,:),'c',etarat(17,:),Z(17,:),'g', ...
     etarat(19,:),Z(19,:),'r','LineWidth',2);
hold on;
plot(ones(1,l),Z(1,:),'k--');
hold off;
xlabel('Log10 of Dislocation Strain Rate/Diffusion Strain Rate', ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
legend('Dehydration @ 12 Mya & 5 cm/yr Half-rate','Dry','Wet');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
axis([-3 3 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,2),
plot(eta(12,:),Z(12,:),'c',eta(17,:),Z(17,:),'g', ...
     eta(19,:),Z(19,:),'r','LineWidth',2);
xlabel('Log10 of Viscosity (Pa*s)','FontSize',8,'FontWeight','bold');
axis([18 25 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,3)
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q(12,1:tct(12,1)), ...
     Z(12,1:tct(12,1)),'c--',Q(12,tct(12,1):tct(12,2)), ...
     Z(12,tct(12,1):tct(12,2)),'c',Q(12,tct(12,2):l), ...
```

```
     Z(12,tct(12,2):l),'c--',Q(17,1:tct(17,1)), ...
     Z(17,1:tct(17,1)),'g--',Q(17,tct(17,1):tct(17,2)), ...
     Z(17,tct(17,1):tct(17,2)),'g',Q(17,tct(17,2):l), ...
     Z(17,tct(17,2):l),'g--',Q(19,1:tct(19,1)), ...
     Z(19,1:tct(19,1)),'r--',Q(19,tct(19,1):tct(19,2)), ...
     Z(19,tct(19,1):tct(19,2)),'r',Q(19,tct(19,2):l), ...
     Z(19,tct(19,2):l),'r--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');
legend('Location','se','PREM','SW02'); legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(2,3,4)
plot(teq(12,:),Z(12,:),'c',teq(17,:),Z(17,:),'g',teq(19,:), ...
     Z(19,:),'r','LineWidth',2);
hold on;
plot(teqmark(12,:),Z(12,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
       8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse');

subplot(2,3,5),
plot(deq(12,:),Z(12,:),'c',deq(17,:),Z(17,:),'g',deq(19,:), ...
     Z(19,:),'r','LineWidth',2);
xlabel('Log10 of Eq. Grain Size (mm)','FontSize',8,'FontWeight', ...
       'bold');
set(gca,'YDir','reverse');

subplot(2,3,6)
plot(PA5,PAZ,'k:',V2m,VZm,'k-.',Vsb(12,1:tct(12,1)), ...
     Z(12,1:tct(12,1)),'c--',Vsb(12,tct(12,1):tct(12,2)), ...
     Z(12,tct(12,1):tct(12,2)),'c',Vsb(12,tct(12,2):l), ...
     Z(12,tct(12,2):l),'c--',Vsb(17,1:tct(17,1)), ...
     Z(17,1:tct(17,1)),'g--',Vsb(17,tct(17,1):tct(17,2)), ...
     Z(17,tct(17,1):tct(17,2)),'g',Vsb(17,tct(17,2):l), ...
     Z(17,tct(17,2):l),'g--',Vsb(19,1:tct(19,1)), ...
     Z(19,1:tct(19,1)),'r--',Vsb(19,tct(19,1):tct(19,2)), ...
     Z(19,tct(19,1):tct(19,2)),'r',Vsb(19,tct(19,2):l), ...
     Z(19,tct(19,2):l),'r--','LineWidth',2);
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
axis([4 5 0 400]);
set(gca,'YDir','reverse');
legend('Location','sw','PA5','4-20 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

figure(9)
subplot(1,2,2)
plot(Vx(1,1:5),Z(1,Zx(1:5)),'-cd',Vx(1,6:10),Z(1,Zx(6:10)), ...
```

```
     '-gd',Vx(1,11:15),Z(1,Zx(11:15)),'-rd','LineWidth',2);
hold on;
plot(Vx1(1,1:5),Z(1,Zx1(1:5)),'-c*',Vx2(1,1:5),Z(1,Zx2(1:5)),'-cp', ...
     Vx3(1,1:5),Z(1,Zx3(1:5)),'-co',Vx4(1,1:5),Z(1,Zx4(1:5)),'-cs', ...
     Vx5(1,1:5),Z(1,Zx5(1:5)),'-ch',Vx1(1,6:10),Z(1,Zx1(6:10)), ...
     '-g*',Vx2(1,6:10),Z(1,Zx2(6:10)),'-gp', Vx3(1,6:10), ...
     Z(1,Zx3(6:10)),'-go',Vx4(1,6:10),Z(1,Zx4(6:10)),'-gs', ...
     Vx5(1,6:10),Z(1,Zx5(6:10)),'-gh',Vx1(1,11:15),Z(1,Zx1(11:15)), ...
     '-r*',Vx2(1,11:15),Z(1,Zx2(11:15)),'-rp', Vx3(1,11:15), ...
     Z(1,Zx3(11:15)),'-ro',Vx4(1,11:15),Z(1,Zx4(11:15)),'-rs', ...
     Vx5(1,11:15),Z(1,Zx5(11:15)),'-rh');
hold off;
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');

subplot(1,2,1)
plot(Qx(1,1:5),Z(1,Zy(1:5)),'-cd',Qx(1,6:10),Z(1,Zy(6:10)),'-gd', ...
     Qx(1,11:15),Z(1,Zy(11:15)),'-rd','LineWidth',2);
hold on;
plot(Qx1(1,1:5),Z(1,Zy1(1:5)),'-c*',Qx2(1,1:5),Z(1,Zy2(1:5)),'-cp', ...
     Qx3(1,1:5),Z(1,Zy3(1:5)),'-co',Qx4(1,1:5),Z(1,Zy4(1:5)),'-cs', ...
     Qx5(1,1:5),Z(1,Zy5(1:5)),'-ch',Qx1(1,6:10),Z(1,Zy1(6:10)), ...
     '-g*',Qx2(1,6:10),Z(1,Zy2(6:10)),'-gp',Qx3(1,6:10), ...
     Z(1,Zy3(6:10)),'-go',Qx4(1,6:10),Z(1,Zy4(6:10)),'-gs', ...
     Qx5(1,6:10),Z(1,Zy5(6:10)),'-gh',Qx1(1,11:15),Z(1,Zy1(11:15)), ...
     '-r*',Qx2(1,11:15),Z(1,Zy2(11:15)),'-rp', Qx3(1,11:15), ...
     Z(1,Zy3(11:15)),'-ro',Qx4(1,11:15),Z(1,Zy4(11:15)),'-rs', ...
     Qx5(1,11:15),Z(1,Zy5(11:15)),'-rh');
hold off;
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
legend('1 cm/yr Eq.','2.5 cm/yr Eq.','5 cm/yr Eq.', ...
       '1 cm/yr 10um','1 cm/yr 100um','1 cm/yr 1mm', ...
       '1 cm/yr 1cm','1 cm/yr 5cm','2.5 cm/yr 10um', ...
       '2.5 cm/yr 100um','2.5 cm/yr 1mm','2.5 cm/yr 1cm', ...
       '2.5 cm/yr 5cm','5 cm/yr 10um','5 cm/yr 100um', ...
       '5 cm/yr 1mm','5 cm/yr 1cm','5 cm/yr 5cm');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');

figure(10)
subplot(2,3,1)
axis([0 400 0 400]);
patch([0 0 400 400],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7, ...
      'EdgeAlpha',0);
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q(22,:),Z(20,:),'b', ...
     Q(15,1:tct(15,1)),Z(15,1:tct(15,1)),'c--', ...
     Q(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'c', ...
     Q(15,tct(15,2):l),Z(15,tct(15,2):l),'c--',Q(20,1:tct(20,1)), ...
     Z(20,1:tct(20,1)),'g--',Q(20,tct(20,1):tct(20,2)), ...
     Z(20,tct(20,1):tct(20,2)),'g',Q(20,tct(20,2):l), ...
```

```
     Z(20,tct(20,2):l),'g--',Q(23,1:tct(23,1)),Z(23,1:tct(23,1)), ...
     'r--', Q(23,tct(23,1):tct(23,2)),Z(23,tct(23,1):tct(23,2)), ...
     'r',Q(23,tct(23,2):l),Z(23,tct(23,2):l),'r--','LineWidth',2),
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse');
legend('Location','se','PREM','SW02');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(2,3,2)
plot(PA5,PAZ,'k:',V4m,VZm,'k-.',Vsb(22,:),Z(20,:),'b', ...
     Vsb(15,1:tct(15,1)),Z(15,1:tct(15,1)),'c--', ...
     Vsb(15,tct(15,1):tct(15,2)), Z(15,tct(15,1):tct(15,2)),'c', ...
     Vsb(15,tct(15,2):l),Z(15,tct(15,2):l),'c--', ...
     Vsb(20,1:tct(20,1)),Z(20,1:tct(20,1)),'g--', ...
     Vsb(20,tct(20,1):tct(20,2)),Z(20,tct(20,1):tct(20,2)),'g', ...
     Vsb(20,tct(20,2):l),Z(20,tct(20,2):l),'g--', ...
     Vsb(23,1:tct(23,1)),Z(23,1:tct(23,1)),'r--', ...
     Vsb(23,tct(23,1):tct(23,2)),Z(23,tct(23,1):tct(23,2)),'r', ...
     Vsb(23,tct(23,2):l),Z(23,tct(23,2):l),'r--','LineWidth',2);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
axis([4 5 0 400]);
set(gca,'YDir','reverse');
legend('Location','sw','PA5','52-110 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(2,3,3)
plot(deq(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'c', ...
     deq(20,tct(20,1):tct(20,2)), Z(20,tct(20,1):tct(20,2)),'g', ...
     deq6(1,:),Z(15,:),'b', deq(23,tct(23,1):tct(23,2)), ...
     Z(23,tct(23,1):tct(23,2)),'r', deq(15,1:tct(15,1)), ...
     Z(15,1:tct(15,1)),'c--',deq(15,tct(15,2):l),Z(15,tct(15,2):l), ...
     'c--',deq(20,1:tct(20,1)),Z(20,1:tct(20,1)),'g--', ...
     deq(20,tct(20,2):l),Z(20,tct(20,2):l),'g--', ...
     deq(23,1:tct(23,1)),Z(23,1:tct(23,1)),'r--', ...
     deq(23,tct(23,2):l),Z(23,tct(23,2):l),'r--','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
legend('Dehydration @ 75 Mya & 5 cm/yr Half-rate','Dry', ...
       'F&J Grain Size','A&E Grain Size');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');
axis([-1 3.5 0 400]);

subplot(2,3,4)
plot(FJetadry,Z(15,:),'b',eta(15,tct(15,1):tct(15,2)), ...
     Z(15,tct(15,1):tct(15,2)),'c',eta(20,tct(20,1):tct(20,2)), ...
     Z(20,tct(20,1):tct(20,2)),'g',eta(23,tct(23,1):tct(23,2)), ...
     Z(23,tct(23,1):tct(23,2)),'r', eta(15,1:tct(15,1)), ...
     Z(15,1:tct(15,1)),'c--',eta(15,tct(15,2):l),Z(15,tct(15,2):l), ...
```

```
     'c--',eta(20,1:tct(20,1)),Z(20,1:tct(20,1)),'g--', ...
     eta(20,tct(20,2):l),Z(20,tct(20,2):l),'g--', ...
     eta(23,1:tct(23,1)),Z(23,1:tct(23,1)),'r--', ...
     eta(23,tct(23,2):l),Z(23,tct(23,2):l),'r--','LineWidth',2);
xlabel('Log10(\eta) (Pa*s)','FontSize',8,'FontWeight','bold');
axis([17 25 0 400]);
set(gca,'YDir','reverse','XTick',[17 19 21 23 25]);

subplot(2,3,6)
patch([-4 -4 8 8],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);
hold on;
plot(etarat(15,:),Z(15,:),'c',etarat(20,:),Z(20,:),'g', ...
     etarat(23,:),Z(23,:),'r',FJetarat,Z(15,:),'b','LineWidth',2);
hold on;
plot(zeros(1,l),Z(1,:),'k--');
hold off;
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-4 8 0 400]);
set(gca,'YDir','reverse');
hold off;

figure(11)
subplot(1,2,1)
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q3(20,:),Z(20,:),'b', ...
     Q4(20,:),Z(20,:),'r',Q3(21,:),Z(21,:),'g',Q4(21,:),Z(21,:), ...
     'c',Qsb3(20,:),Z(20,:),'b--',Qsb4(20,:),Z(20,:),'r--', ...
     Qsb3(21,:),Z(21,:),'g--',Qsb4(21,:),Z(21,:),'c--','LineWidth',2);
axis([0 400 0 400]);
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
legend('PREM','SW02','Dry 1mm F&J','Dry 10mm F&J','Wet 1mm F&J', ...
       'Wet 10mm F&J','Dry 1mm K','Dry 10mm K','Wet 1mm K', ...
       'Wet 10mm K');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[1 1 1]);

subplot(1,2,2)
plot(PA5,PAZ,'k:',V4m,VZm,'k-.',V3(20,:)./1e3,Z(20,:),'b', ...
     V4(20,:)./1e3,Z(20,:),'r',V3(21,:)./1e3,Z(21,:),'g', ...
     V4(21,:)./1e3,Z(21,:),'c',Vsb3(20,:)./1e3,Z(20,:),'b--', ...
     Vsb4(20,:)./1e3,Z(20,:),'r--',Vsb3(21,:)./1e3,Z(21,:),'g--', ...
     Vsb4(21,:)./1e3,Z(21,:),'c--','LineWidth',2);
axis([3 5 0 400]);
xlabel('Shear-wave Velocity (km/s)','FontSize',8,'FontWeight','bold');
legend('PA5','52-110 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[1 1 1]);

figure(12)
```

```
subplot(2,2,1)
plot(eta(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'c', ...
     eta(20,tct(20,1):tct(20,2)),Z(20,tct(20,1):tct(20,2)),'b', ...
     eta(15,1:tct(15,1)),Z(15,1:tct(15,1)),'c--', ...
     eta(15,tct(15,2):l),Z(15,tct(15,2):l),'c--', ...
     eta(20,1:tct(20,1)),Z(20,1:tct(20,1)),'b--', ...
     eta(20,tct(20,2):l),Z(20,tct(20,2):l),'b--','LineWidth',2);
xlabel('Log10(\eta) (Pa*s)','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([19 25 0 400]);
set(gca,'YDir','reverse','XTick',[19 20 21 22 23 24 25], ...
    'plotboxaspectratio',[2 2 2]);

subplot(2,2,2)
plot(deq(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'c', ...
     deq(20,tct(20,1):tct(20,2)),Z(20,tct(20,1):tct(20,2)),'b', ...
     deq(15,1:tct(15,1)),Z(15,1:tct(15,1)),'c--', ...
     deq(15,tct(15,2):l),Z(15,tct(15,2):l),'c--', ...
     deq(20,1:tct(20,1)),Z(20,1:tct(20,1)),'b--', ...
     deq(20,tct(20,2):l),Z(20,tct(20,2):l),'b--','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
legend('Dehydration','Dry');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
axis([-1 2 0 400]);

subplot(2,2,3)
patch([-2 -2 2 2],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);
hold on;
plot(etarat(15,:),Z(15,:),'c',etarat(20,:),Z(20,:),'b', 'LineWidth',2);
hold on;
plot(zeros(1,l),Z(1,:),'k--');
hold off;
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-2 2 0 400]);
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
hold off;

subplot(2,2,4)
plot(teq(15,:),Z(15,:),'c',teq(20,:),Z(20,:),'b','LineWidth',2);
hold on;
plot(teqmark(15,:),Z(15,:),'c--',teqmark(20,:),Z(20,:),'b--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
      8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse','XTick',[5 6 7 8 9],'plotboxaspectratio', ...
    [2 2 2]);

figure(13)
```

```
subplot(2,2,1)
plot(eta(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'k', ...
     eta(11,tct(11,1):tct(11,2)), Z(11,tct(11,1):tct(11,2)),'c', ...
     eta(13,tct(13,1):tct(13,2)),Z(13,tct(13,1):tct(13,2)),'b', ...
     eta(11,1:tct(11,1)),Z(11,1:tct(11,1)),'c--', ...
     eta(11,tct(11,2):l),Z(11,tct(11,2):l),'c--', ...
     eta(13,1:tct(13,1)),Z(13,1:tct(13,1)),'b--', ...
     eta(13,tct(13,2):l),Z(13,tct(13,2):l),'b--', ...
     eta(15,1:tct(15,1)),Z(15,1:tct(15,1)),'k--', ...
     eta(15,tct(15,2):l),Z(15,tct(15,2):l),'k--','LineWidth',2);
xlabel('Log10(\eta) (Pa*s)','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([17 25 0 400]);
set(gca,'YDir','reverse','XTick',[17 19 21 23 25], ...
    'plotboxaspectratio',[2 2 2]);

subplot(2,2,2)
plot(deq(11,tct(11,1):tct(11,2)),Z(11,tct(11,1):tct(11,2)),'c', ...
     deq(13,tct(13,1):tct(13,2)),Z(13,tct(13,1):tct(13,2)),'b', ...
     deq(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'k', ...
     deq(11,1:tct(11,1)),Z(11,1:tct(11,1)),'c--', ...
     deq(11,tct(11,2):l),Z(11,tct(11,2):l),'c--', ...
     deq(13,1:tct(13,1)),Z(13,1:tct(13,1)),'b--', ...
     deq(13,tct(13,2):l),Z(13,tct(13,2):l),'b--', ...
     deq(15,1:tct(15,1)),Z(15,1:tct(15,1)),'k--', ...
     deq(15,tct(15,2):l),Z(15,tct(15,2):l),'k--','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
legend('5 Myr','25 Myr','75 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
axis([-1 2 0 400]);

subplot(2,2,3)
patch([-2 -2 4 4],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);
hold on;
plot(etarat(11,:),Z(11,:),'c',etarat(13,:),Z(13,:),'b', ...
     etarat(15,:),Z(15,:),'k','LineWidth',2);
hold on;
plot(zeros(1,l),Z(1,:),'k--');
hold off;
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([-2 4 0 400]);
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
hold off;

subplot(2,2,4)
plot(teq(11,:),Z(11,:),'c',teq(13,:),Z(13,:),'b',teq(15,:), ...
     Z(15,:),'k','LineWidth',2);
hold on;
plot(teqmark(11,:),Z(11,:),'c--',teqmark(13,:),Z(13,:),'b--', ...
```

```
     teqmark(15,:),Z(15,:),'k--');
hold off;
xlabel('Log10 of Characteristic Timescale (yr)','FontSize', ...
       8,'FontWeight','bold');
axis([5 9 0 400]);
set(gca,'YDir','reverse','XTick',[5 6 7 8 9],'plotboxaspectratio', ...
    [2 2 2]);

figure(14)
subplot(1,3,1)
axis([0 400 0 400]);
patch([0 0 400 400],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7, ...
      'EdgeAlpha',0);
plot(QPREM,QPREMZ,'k:',SW02,SWZ,'k-.',Q(15,1:tct(15,1)), ...
     Z(15,1:tct(15,1)),'b--',Q(15,tct(15,1):tct(15,2)), ...
     Z(15,tct(15,1):tct(15,2)),'b', ...
     Q(15,tct(15,2):l),Z(15,tct(15,2):l),'b--','LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse','XTick',[0 100 200 300 400], ...
    'plotboxaspectratio',[2 2 2]);
legend('Location','se','PREM','SW02');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(1,3,2)
plot(PA5,PAZ,'k:',V4m,VZm,'k-.',Vsb(15,1:tct(15,1)), ...
     Z(15,1:tct(15,1)),'b--',Vsb(15,tct(15,1):tct(15,2)), ...
     Z(15,tct(15,1):tct(15,2)),'b',Vsb(15,tct(15,2):l), ...
     Z(15,tct(15,2):l),'b--','LineWidth',2);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
axis([4 5 0 400]);
set(gca,'YDir','reverse','XTick',[4 4.25 4.5 4.75 5], ...
    'plotboxaspectratio',[2 2 2]);
legend('Location','sw','PA5','52-110 Myr');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');

subplot(1,3,3)
plot(deq(15,tct(15,1):tct(15,2)),Z(15,tct(15,1):tct(15,2)),'b', ...
     deq(15,1:tct(15,1)),Z(15,1:tct(15,1)),'b--', ...
     deq(15,tct(15,2):l),Z(15,tct(15,2):l),'b--','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
legend('Dehydration @ 75 Myr & 5 cm/yr Half-rate');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
axis([-1 2 0 400]);

figure(15)
subplot(2,3,1)
axis([0 400 0 400]);
```

```
patch([0 0 400 400],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7, ...
      'EdgeAlpha',0);
plot(Q(15,:),Z(15,:),'c',Q(22,:),Z(22,:),'b',Q(23,:),Z(23,:),'k', ...
     'LineWidth',2);
xlabel('Q','FontSize',8,'FontWeight','bold');
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
axis([0 400 0 400]);
set(gca,'YDir','reverse','XTick',[0 100 200 300 400], ...
    'plotboxaspectratio',[2 2 2]);

subplot(2,3,2)
plot(Vsb(15,:),Z(15,:),'c', Vsb(22,:),Z(22,:),'b',Vsb(23,:), ...
     Z(23,:),'k','LineWidth',2);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
axis([4 5 0 400]);
set(gca,'YDir','reverse','XTick',[4 4.25 4.5 4.75 5], ...
    'plotboxaspectratio',[2 2 2]);

subplot(2,3,3)
plot(deq(15,:),Z(15,:),'c', deq6(1,:),Z(15,:),'b',deq(23,:), ...
     Z(23,:),'k','LineWidth',2);
xlabel('Log10(d) (mm)','FontSize',8,'FontWeight','bold');
legend('Hall & Parmentier, 2003','Faul & Jackson, 2005', ...
       'Austin & Evans, 2007');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
axis([-1 2 0 400]);

subplot(2,3,4),
plot(FJetadry(1,:),Z(15,:),'b',eta(15,:),Z(15,:),'c',eta(23,:), ...
     Z(23,:),'k','LineWidth',2);
xlabel('Log10(\eta) (Pa*s)','FontSize',8,'FontWeight','bold');
axis([17 25 0 400]);
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse','XTick',[17 19 21 23 25], ...
    'plotboxaspectratio',[2 2 2]);

subplot(2,3,6)
patch([-4 -4 8 8],[0 70 70 0],[.7 .7 .7],'FaceAlpha',.7);
hold on;
plot(etarat(15,:),Z(15,:),'c', etarat(23,:),Z(23,:),'k', ...
     FJetarat,Z(15,:),'b','LineWidth',2);
hold on;
plot(zeros(1,l),Z(1,:),'k--');
hold off;
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold'),
axis([-4 8 0 400]);
set(gca,'YDir','reverse','plotboxaspectratio',[2 2 2]);
hold off;

figure(16)
```

```
subplot(3,3,1)
plot(Vsb(1,:),Z(1,:),'c',Vsb(6,:),Z(6,:),'r',Vsb(11,:),Z(11,:), ...
     'b','LineWidth',2);
hold on;
plot(V1m,VZm,'k');
axis([3.5 5 0 400]);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');
hold off;
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');
title('5 Myr','FontSize',8,'FontWeight','bold');

subplot(3,3,2),
plot(Vsb(3,:),Z(3,:),'c',Vsb(8,:),Z(8,:),'r',Vsb(13,:),Z(13,:), ...
     'b','LineWidth',2);
hold on;
plot(V2m,VZm,'k');
axis([3.5 5 0 400]);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');
hold off;
title('25 Myr','FontSize',8,'FontWeight','bold');

subplot(3,3,3)
plot(Vsb(5,:),Z(5,:),'c',Vsb(10,:),Z(10,:),'r',Vsb(15,:),Z(15,:), ...
     'b','LineWidth',2);
hold on;
plot(V4m,VZm,'k');
axis([3.5 5 0 400]);
xlabel('Vs (km/s)','FontSize',8,'FontWeight','bold');
set(gca,'YDir','reverse');
hold off;
legend('1 cm/yr','2.5 cm/yr','5 cm/yr','N&F');
legend('boxoff');
set(legend,'FontSize',8,'FontWeight','bold');
title('75 Myr','FontSize',8,'FontWeight','bold');



subplot(3,3,4)
plot(etarat(1,:),Z(1,:),'c',etarat(6,:),Z(6,:),'r',etarat(11,:), ...
     Z(11,:),'b','LineWidth',2);
hold on;
plot(At1,VZm,'k');
axis([-2 4 0 400]);
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold'),
set(gca,'YDir','reverse');
hold off;
ylabel('Depth (km)','FontSize',8,'FontWeight','bold');

subplot(3,3,5),
plot(etarat(3,:),Z(3,:),'c',etarat(8,:),Z(8,:),'r',etarat(13,:), ...
```

```
     Z(13,:),'b','LineWidth',2);
hold on;
plot(At2,VZm,'k');
axis([-2 4 0 400]);
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold'),
set(gca,'YDir','reverse');
hold off;

subplot(3,3,6)
plot(etarat(5,:),Z(5,:),'c',etarat(10,:),Z(10,:),'r',etarat(15,:), ...
     Z(15,:),'b','LineWidth',2);
hold on;
plot(At4,VZm,'k');
axis([-2 4 0 400]);
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold'),
set(gca,'YDir','reverse');
hold off;

etaratavg1=(etarat(1,:)+etarat(3,:)+etarat(5,:))./3;
etaratavg2=(etarat(6,:)+etarat(8,:)+etarat(10,:))./3;
etaratavg3=(etarat(11,:)+etarat(13,:)+etarat(15,:))./3;

subplot(3,3,8)
plot(etaratavg1,Z(5,:),'c',etaratavg2,Z(10,:),'r',etaratavg3, ...
     Z(15,:),'b','LineWidth',2);
hold on;
plot(At4,VZm,'k');
axis([-2 4 0 400]);
xlabel({'Log10(Dislocation Strain Rate';'/Diffusion Strain Rate)'}, ...
       'FontSize',8,'FontWeight','bold'),
set(gca,'YDir','reverse');
hold off;
title('75 Myr Average','FontSize',8,'FontWeight','bold');
```

deformread.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% deformread – Function called by uberdeform for a single Comsol               %
%              model for a steady-state grain size solution. Loads             %
%              data stored in the model, and uses this data to                 %
%              calculate attenuation and shear wave velocity.                  %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Input:                                                                       %
%                                                                              %
%     xfem – Comsol model with a steady-state grain size solution.             %
%                                                                              %
% Output:                                                                      %
%                                                                              %
%     U – Velocity structure in m/s.                                           %
%     PIT – Temperature profile in Celsius.                                    %
%     PIz – Grid spacing for depth in m.                                       %
%     PIeta – Viscosity profile in Pa s.                                       %
%     PIetadisl – Dislocation creep viscosity profile in Pa s.                 %
%     PIetadiff – Diffusion creep viscosity profile in Pa s.                   %
%     PIetarat – Ratio of Diffusion creep viscosity to Dislocation             %
%                creep viscosity.                                              %
%     PIteq – Characteristic timescale in s.                                   %
%     PIsigma – Differential stress profile in Pa.                             %
%     PIsrII – Second invariant of strain rate profile in 1/s.                 %
%     PIdeq – Grain size profile in um.                                        %
%     PIdeq6 – F&J grain size profile in um.                                   %
%     PIQ – Attenuation profile for the steady-state grain size                %
%           solution.                                                          %
%     PI1Q – Attenuation profile for a constant grain size of 10 um.           %
%     PI2Q – Attenuation profile for a constant grain size of 100 um.          %
%     PI3Q – Attenuation profile for a constant grain size of 1 mm.            %
%     PI4Q – Attenuation profile for a constant grain size of 10 mm.           %
%     PI5Q – Attenuation profile for a constant grain size of 50 mm.           %
%     PI6Q – Attenuation profile for the Faul & Jackson (2005) grain           %
%            size profile.                                                     %
%     PIQn  - \                                                                %
%     PI1Qn -  \                                                               %
%     PI2Qn -   \    Same as above except w/o pressure & temperature           %
%     PI3Qn -    >   scaling the F&J Burger's model cutoff & Maxwell           %
%     PI4Qn -   /    times (from pg. 121 of F&J 2005).                         %
%     PI5Qn -  /                                                               %
%     PI6Qn - /                                                                %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Output (cont.):                                                            %
%                                                                             %
%     PIV – Shear wave velocity profile calculated using formulation  %
%           in Jackson (1993) for the steady-state grain size         %
%           solution.                                                 %
%     PIV1 – Shear wave velocity profile for a constant grain size of %
%            10 um using Jackson (1993).                              %
%     PIV2 – Shear wave velocity profile for a constant grain size of %
%            100 um using Jackson (1993).                             %
%     PIV3 – Shear wave velocity profile for a constant grain size of %
%            1 mm using Jackson (1993).                               %
%     PIV4 – Shear wave velocity profile for a constant grain size of %
%            10 mm using Jackson (1993).                              %
%     PIV5 – Shear wave velocity profile for a constant grain size of %
%            50 mm using Jackson (1993).                              %
%     PIV6 – Shear wave velocity profile for the F&J grain size       %
%            profile using Jackson (1993).                            %
%     CH2O – Water content profile in ppm H/Si.                       %
%     Gw – Shear modulus profile calculated using formulation in Faul %
%          & Jackson (2005) for the steady-state grain size solution. %
%     G1 – Shear modulus profile calculated for a constant grain size %
%          of 10 um.                                                  %
%     G2 – Shear modulus profile calculated for a constant grain size %
%          of 100 um.                                                 %
%     G3 – Shear modulus profile calculated for a constant grain size %
%          of 1 mm.                                                   %
%     G4 – Shear modulus profile calculated for a constant grain size %
%          of 10 mm.                                                  %
%     G5 – Shear modulus profile calculated for a constant grain size %
%          of 50 mm.                                                  %
%     Gwa - \                                                         %
%     G1a -  \                                                        %
%     G2a -   \ Same as above except w/o a pressure and temperature   %
%     G3a -   / dependence incorporated into the unrelaxed compliance %
%     G4a -  /  term Ju (from pg. 120 F&J 2005).                      %
%     G5a - /                                                         %
%     Gwn - \                                                         %
%     G1n -  \                                                        %
%     G2n -   \ Same as above except w/o a pressure and temperature   %
%     G3n -   / dependence incorporated into the cutoff & Maxwell     %
%     G4n -  /  times.                                                %
%     G5n - /                                                         %
%     Gwna - \                                                        %
%     G1na -  \                                                       %
%     G2na -   \ Same as above except w/o a pressure and temperature  %
%     G3na -   / dependence incorporated into the cutoff & Maxwell    %
%     G4na -  /  times or the unrelaxed compliance term Ju.           %
%     G5na - /                                                        %
%     fH2O – Water scaling term for cutoff & Maxwell times.  The      %
%            actual water content divided by the dry water content.   %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Output (cont.):                                                              %
%                                                                              %
%     Th3 – Upper cutoff time for a constant grain size of 1 mm.               %
%     Th4 – Upper cutoff time for a constant grain size of 10 mm.              %
%     J31 – Real portion of complex compliance J* (pg. 122 F&J 2005)           %
%           for a constant grain size of 1 mm.                                 %
%     J41 – Real portion of complex compliance J* (pg. 122 F&J 2005)           %
%           for a constant grain size of 10 mm.                                %
%     dlnJu3 – Fractional adjustment to Ju for the viscoelastic                %
%              regime (grain size & T dependent – pg. 122 F&J 2005)            %
%              for a constant grain size of 1 mm.                              %
%     dlnJu4 – Fractional adjustment to Ju for the viscoelastic                %
%              regime (grain size & T dependent – pg. 122 F&J 2005)            %
%              for a constant grain size of 10 mm.                             %
%     Tl3 – Lower cutoff time for a constant grain size of 1 mm.               %
%     Tl4 – Lower cutoff time for a constant grain size of 10 mm.              %
%     int3 – Integral term from calculation of J31 & J32.                      %
%     int4 – Integral term from calculation of J41 & J42.                      %
%     G – Temperature & pressure dependent shear modulus used to               %
%         calculate the unrelaxed compliance Ju (1/Ju=G).                      %
%     Qsb – Attenuation profile for steady-state grain size using              %
%           formulation in Karato (2003).                                      %
%     Qsb1 – Attenuation profile for a constant grain size of 10 um            %
%            using formuation in Karato (2003).                                %
%     Qsb2 – Attenuation profile for a constant grain size of 100 um           %
%            using formulation in Karato (2003).                               %
%     Qsb3 – Attenuation profile for a constant grain size of 1 mm             %
%            using formulation in Karato (2003).                               %
%     Qsb4 – Attenuation profile for a constant grain size of 10 mm            %
%            using formulation in Karato (2003).                               %
%     Qsb5 – Attenuation profile for a constant grain size of 50 mm            %
%            using formulation in Karato (2003).                               %
%     Qsb6 – Attenuation profile for the Faul & Jackson (2005) grain           %
%            size profile using formulation in Karato (2003).                  %
%     Vsb – Shear wave velocity for steady-state grain size using              %
%           Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from             %
%           the Faul & Jackson (2005) Burger's model, and formulation          %
%           in Karato (2003).                                                  %
%     Vsb1 – Shear wave velocity for a constant grain size of 10 um,           %
%            Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from            %
%            the Faul & Jackson (2005) Burger's model, and                     %
%            formulation in Karato (2003).                                     %
%     Vsb2 – Shear wave velocity for a constant grain size of 100 um,          %
%            Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from            %
%            the Faul & Jackson (2005) Burger's model, and                     %
%            formulation in Karato (2003).                                     %
%     Vsb3 – Shear wave velocity for a constant grain size of 1 mm,            %
%            Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from            %
%            the Faul & Jackson (2005) Burger's model, and                     %
%            formulation in Karato (2003).                                     %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Output (cont.):                                                            %
%                                                                             %
%     Vsb4 - Shear wave velocity for a constant grain size of 10 mm,  %
%            Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from   %
%            the Faul & Jackson (2005) Burger's model, and            %
%            formulation in Karato (2003).                            %
%     Vsb5 – Shear wave velocity for a constant grain size of 50 mm,  %
%            Vinf from Stixrude & Lithgow-Bertelloni (2005), Q from   %
%            the Faul & Jackson (2005) Burger's model, and            %
%            formulation in Karato (2003).                            %
%     Vsb6 – Shear wave velocity for the Faul & Jackson (2005) grain  %
%            size profile using Vinf from Stixrude &                  %
%            Lithgow-Bertelloni (2005), Q from the Faul & Jackson     %
%            (2005) Burger's model, and formulation in Karato (2003). %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [U,PIT,PIz,PIeta,PIetadisl,PIetadiff,PIetarat,PIteq, ...
          PIsigma,PIsrII,PIdeq,PIdeq6,PIQ,PI1Q,PI2Q,PI3Q, ...
          PI4Q,PI5Q,PI6Q,PIQn,PI1Qn,PI2Qn,PI3Qn,PI4Qn,PI5Qn,PI6Qn, ...
          PIV,PIV1,PIV2,PIV3,PIV4,PIV5,PIV6,CH2O,Gw,G1,G2,G3,G4, ...
          G5,Gwa,G1a,G2a,G3a,G4a,G5a,Gwn,G1n,G2n,G3n,G4n,G5n, ...
          G6n,Gwna,G1na,G2na,G3na,G4na,G5na,G6na,fH2O,Th3,Th4,J31, ...
          J41,dlnJu3,dlnJu4,Tl3,Tl4,int3,int4,G,Qsb,Qsb1,Qsb2,Qsb3, ...
          Qsb4,Qsb5,Qsb6,Vsb,Vsb1,Vsb2,Vsb3,Vsb4,Vsb5,Vsb6] ...
         =deformread(xfem);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Here we load the file specified in input xfem and organize the size %
% of the data vectors we will be reading.                                    %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xfem1=flload(xfem);

xmax=4e5;
zmax=4e5;
xsize=1e3;
zsize=1e3;
dx=xmax/xsize;
dz=zmax/zsize;
x_fem=0:dx:xmax;
z_fem=-zmax:dz:0;
[X,Z] = meshgrid(x_fem,z_fem);
XMAX=size(X);
xx(1,:)=X(:);
xx(2,:)=Z(:);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Constants - p,lambda for teq; Tr,dr,ma,E,Ju,Th,Tl,Tm,aq,omega,delta          %
%             for calculation of G,Q, & Vs.                                    %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p=3;
lambda=1;

Tr=1223;
dr=1e-5;
ma=1.09;
mj=0.16;
mv=2.1;
E=5.05e5;
R=8.314;
Jur=.0149;
Th=5.26e6;
Tl=3.981e-3;
Tm=4.31e6;
aq=.27;
omega=1/8.2;
delta=1.4;
A1=2.6036e3;
dlnJu=9.1e-4;
rho=3300;
V1=1.6e-5;
g=9.81;
r=1.2;
CH2Od=50;
CH2Ow=1000;
dGdPR=1.8;
dGdT=-.0136;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% Determines if the loaded model is wet, dehydrated, or dry, and               %
% interpolates the appropriate variables.                                      %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if xfem(12:13)=='+C'
    [u01,T01,eta01,etadiff01,etadisl01,sigma01,srII01,deq01,CH2O01] ...
    =postinterp(xfem1,'u','initT','etatot','etawetdiff', ...
                'etawetdis','o','srII','deq','CH2O',xx);
    CH2O1=reshape(CH2O01,length(z_fem),length(x_fem));
    CH2O(1,:)=CH2O1(:,1);
elseif xfem(12:13)=='-C'
    [u01,T01,eta01,etadiff01,etadisl01,sigma01,srII01,deq01]= ...
    postinterp(xfem1,'u','initT','etatot','etawetdiff','etawetdis', ...
               'o','srII','deq',xx);
    CH2O(1,1:zsize+1)=CH2Od;
```

```
elseif xfem(12:13)=='+f'
    [u01,T01,eta01,etadiff01,etadisl01,sigma01,srII01,deq01]= ...
    postinterp(xfem1,'u','initT','etatot','etawetdiff','etawetdis', ...
               'o','srII','deq',xx);
    CH2O(1,1:zsize+1)=CH2Ow;
end

u01_fem=reshape(u01,length(z_fem),length(x_fem));
T01_fem=reshape(T01,length(z_fem),length(x_fem));
eta01_fem=reshape(eta01,length(z_fem),length(x_fem));
etadiff01_fem=reshape(etadiff01,length(z_fem),length(x_fem));
etadisl01_fem=reshape(etadisl01,length(z_fem),length(x_fem));
sigma01_fem=reshape(sigma01,length(z_fem),length(x_fem));
srII01_fem=reshape(srII01,length(z_fem),length(x_fem));
deq01_fem=reshape(deq01,length(z_fem),length(x_fem));

PIz=zeros(1,zsize+1);
PIT=zeros(1,zsize+1);
PIetadiff=zeros(1,zsize+1);
PIetadisl=zeros(1,zsize+1);
PIeta=zeros(1,zsize+1);
PIsigma=zeros(1,zsize+1);
PIsrII=zeros(1,zsize+1);
PIdeq=zeros(1,zsize+1);
PIetarat=zeros(1,zsize+1);
PIteq=zeros(1,zsize+1,1);
PIQ=zeros(1,zsize+1,1);
PITm=zeros(1,zsize+1);
PITl=zeros(1,zsize+1);
PITh=zeros(1,zsize+1);
PIdlnJu=zeros(1,zsize+1);
J1=zeros(1,zsize+1);
J2=zeros(1,zsize+1);
PIV=zeros(1,zsize+1);
U=zeros(1,zsize+1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Averaging of interpolated variables over each interval in x                 %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for cc=1:zsize+1

    for bb=1:xsize+1

        PIz(cc)=Z(cc,1);
        PIT(cc)=T01_fem(cc,1);
        PIetadiff(cc)=PIetadiff(cc)+etadiff01_fem(cc,bb);
        PIetadisl(cc)=PIetadisl(cc)+etadisl01_fem(cc,bb);
        PIeta(cc)=PIeta(cc)+eta01_fem(cc,bb);
        PIsigma(cc)=PIsigma(cc)+sigma01_fem(cc,bb);
        PIsrII(cc)=PIsrII(cc)+srII01_fem(cc,bb);
```

```
        PIdeq(cc)=PIdeq(cc)+deq01_fem(cc,bb);
        U(cc)=U(cc)+u01_fem(cc,bb);

    end

    PIetadiff(cc)=PIetadiff(cc)/bb;
    PIetadisl(cc)=PIetadisl(cc)/bb;
    PIeta(cc)=PIeta(cc)/bb;
    PIsigma(cc)=PIsigma(cc)/bb;
    PIsrII(cc)=PIsrII(cc)/bb;
    PIdeq(cc)=PIdeq(cc)/bb;
    U(cc)=U(cc)/bb;

    if PIdeq(cc)<250
        PIdeq(cc)=250;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of scaling P & T terms for the Faul & Jackson (2005)    %
% Burger's model system.                                                      %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fH2O=((CH2Od./CH2O).^r);

PR=g.*-PIz.*rho;

T=PIT+273;

G=1./Jur+dGdT.*(T-Tr)+1.8.*PR./1e9;

Ju=1./G;

term=exp(PR.*V1./(R.*T));

PIetarat=PIetadiff./PIetadisl;

PIteq=1./(p.*lambda.*PIsigma./PIetadisl)./(24*3600*364.25);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Setting constant grain sizes in um to solve for G, Q, & Vs.         %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

deqconst1=1e1;
deqconst2=1e2;
deqconst3=1e3;
```

```
deqconst4=1e4;
deqconst5=5e4;

deq=PIdeq./1e6;

deq1=deqconst1./1e6;
deq2=deqconst2./1e6;
deq3=deqconst3./1e6;
deq4=deqconst4./1e6;
deq5=deqconst5./1e6;

PIdeq6=zeros(1,zsize+1);

PIdeq6(1,:)=10.^(0.8.*2./pi.*atan((zstr.*-1-240500)/5000-pi./2)+3.9);

deq6=PIdeq6./1e6;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of prefactor B from Karato (2003).                              %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B1=(A1.*deq1.^-p.*CH2O.^r).^aq;
B2=(A1.*deq2.^-p.*CH2O.^r).^aq;
B3=(A1.*deq3.^-p.*CH2O.^r).^aq;
B4=(A1.*deq4.^-p.*CH2O.^r).^aq;
B5=(A1.*deq5.^-p.*CH2O.^r).^aq;
B6=(A1.*deq6.^-p.*CH2O.^r).^aq;
B=(A1.*deq.^-p.*CH2O.^r).^aq;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of Maxwell times from Faul & Jackson (2005).                   %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Tm1=Tm.*(deq1./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm1n=Tm.*(deq1./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

Tm2=Tm.*(deq2./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm2n=Tm.*(deq2./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

Tm3=Tm.*(deq3./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm3n=Tm.*(deq3./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

Tm4=Tm.*(deq4./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm4n=Tm.*(deq4./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

Tm5=Tm.*(deq5./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm5n=Tm.*(deq5./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;
```

```
Tm6=Tm.*(deq6./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tm6n=Tm.*(deq6./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

PITm=Tm.*(deq./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O.^aq;
Tmn=Tm.*(deq./dr).^mv.*exp((E/R).*(1./T-1/Tr)).*fH2O.^aq;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of lower cutoff times from Faul & Jackson (2005).               %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Tl1=Tl.*(deq1./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl1n=Tl.*(deq1./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Tl2=Tl.*(deq2./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl2n=Tl.*(deq2./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Tl3=Tl.*(deq3./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl3n=Tl.*(deq3./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Tl4=Tl.*(deq4./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl4n=Tl.*(deq4./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Tl5=Tl.*(deq5./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl5n=Tl.*(deq5./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Tl6=Tl.*(deq6./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tl6n=Tl.*(deq6./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

PITl=Tl.*(deq./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Tln=Tl.*(deq./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of upper cutoff times from Faul & Jackson (2005).               %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Th1=Th.*(deq1./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th1n=Th.*(deq1./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Th2=Th.*(deq2./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th2n=Th.*(deq2./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Th3=Th.*(deq3./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th3n=Th.*(deq3./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Th4=Th.*(deq4./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th4n=Th.*(deq4./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;
```

```
Th5=Th.*(deq5./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th5n=Th.*(deq5./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

Th6=Th.*(deq6./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Th6n=Th.*(deq6./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

PITh=Th.*(deq./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*term.*fH2O;
Thn=Th.*(deq./dr).^ma.*exp((E/R).*(1./T-1/Tr)).*fH2O;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Setup of integrals for calculating the real (J1) & imaginary (J2)   %
% portions of the complex compliance J.                                       %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

J1int=@(x)x.^(aq-1)./(1+4*pi^2*omega^2.*x.^2);
J2int=@(x)x.^aq./(1+4*pi^2*omega^2.*x.^2);

for i=1:zsize+1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of dlnJu for all grain size profiles (constant or       %
% otherwise).  Above Tr (950 C, 1223 K), dlnJu varies with grain size %
% and temperature.                                                            %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if T(i)>=Tr
        dlnJu1(i)=dlnJu*(deq1/dr)^-mj*(T(i)-Tr);
        dlnJu2(i)=dlnJu*(deq2/dr)^-mj*(T(i)-Tr);
        dlnJu3(i)=dlnJu*(deq3/dr)^-mj*(T(i)-Tr);
        dlnJu4(i)=dlnJu*(deq4/dr)^-mj*(T(i)-Tr);
        dlnJu5(i)=dlnJu*(deq5/dr)^-mj*(T(i)-Tr);
        dlnJu6(i)=dlnJu*(deq6(i)/dr)^-mj*(T(i)-Tr);
        PIdlnJu(i)=dlnJu*(deq(i)/dr)^-mj*(T(i)-Tr);
    else
        dlnJu1(i)=dlnJu;
        dlnJu2(i)=dlnJu;
        dlnJu3(i)=dlnJu;
        dlnJu4(i)=dlnJu;
        dlnJu5(i)=dlnJu;
        dlnJu6(i)=dlnJu;
        PIdlnJu(i)=dlnJu;
    end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of integrals for the real (J1) & imaginary (J2)                 %
% portions of the complex compliance J.                                       %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    J11(i)=Ju(i)*(1+dlnJu1(i)+aq*delta/(Th1(i)^aq-Tl1(i)^aq) ...
           *quadl(J1int,Tl1(i),Th1(i)));
    J11n(i)=Ju(i)*(1+dlnJu1(i)+aq*delta/(Th1n(i)^aq-Tl1n(i)^aq) ...
            *quadl(J1int,Tl1n(i),Th1n(i)));

    J12(i)=Ju(i)*(2*pi*omega*aq*delta/(Th1(i)^aq-Tl1(i)^aq) ...
           *quadl(J2int,Tl1(i),Th1(i))+1/(2*pi*omega*Tm1(i)));
    J12n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th1n(i)^aq-Tl1n(i)^aq) ...
            *quadl(J2int,Tl1n(i),Th1n(i))+1/(2*pi*omega*Tm1n(i)));

    J21(i)=Ju(i)*(1+dlnJu2(i)+aq*delta/(Th2(i)^aq-Tl2(i)^aq) ...
           *quadl(J1int,Tl2(i),Th2(i)));
    J21n(i)=Ju(i)*(1+dlnJu2(i)+aq*delta/(Th2n(i)^aq-Tl2n(i)^aq) ...
            *quadl(J1int,Tl2n(i),Th2n(i)));

    J22(i)=Ju(i)*(2*pi*omega*aq*delta/(Th2(i)^aq-Tl2(i)^aq) ...
           *quadl(J2int,Tl2(i),Th2(i))+1/(2*pi*omega*Tm2(i)));
    J22n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th2n(i)^aq-Tl2n(i)^aq) ...
            *quadl(J2int,Tl2n(i),Th2n(i))+1/(2*pi*omega*Tm2n(i)));

    int3(i)=quadl(J1int,Tl3(i),Th3(i));
    J31(i)=Ju(i)*(1+dlnJu3(i)+aq*delta/(Th3(i)^aq-Tl3(i)^aq) ...
           *quadl(J1int,Tl3(i),Th3(i)));
    J31n(i)=Ju(i)*(1+dlnJu3(i)+aq*delta/(Th3n(i)^aq-Tl3n(i)^aq) ...
            *quadl(J1int,Tl3n(i),Th3n(i)));

    J32(i)=Ju(i)*(2*pi*omega*aq*delta/(Th3(i)^aq-Tl3(i)^aq) ...
           *quadl(J2int,Tl3(i),Th3(i))+1/(2*pi*omega*Tm3(i)));
    J32n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th3n(i)^aq-Tl3n(i)^aq) ...
            *quadl(J2int,Tl3n(i),Th3n(i))+1/(2*pi*omega*Tm3n(i)));

    int4(i)=quadl(J1int,Tl4(i),Th4(i));
    J41(i)=Ju(i)*(1+dlnJu4(i)+aq*delta/(Th4(i)^aq-Tl4(i)^aq) ...
           *quadl(J1int,Tl4(i),Th4(i)));
    J41n(i)=Ju(i)*(1+dlnJu4(i)+aq*delta/(Th4n(i)^aq-Tl4n(i)^aq) ...
            *quadl(J1int,Tl4n(i),Th4n(i)));

    J42(i)=Ju(i)*(2*pi*omega*aq*delta/(Th4(i)^aq-Tl4(i)^aq) ...
           *quadl(J2int,Tl4(i),Th4(i))+1/(2*pi*omega*Tm4(i)));
    J42n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th4n(i)^aq-Tl4n(i)^aq) ...
            *quadl(J2int,Tl4n(i),Th4n(i))+1/(2*pi*omega*Tm4n(i)));

    J51(i)=Ju(i)*(1+dlnJu5(i)+aq*delta/(Th5(i)^aq-Tl5(i)^aq) ...
           *quadl(J1int,Tl5(i),Th5(i)));
    J51n(i)=Ju(i)*(1+dlnJu5(i)+aq*delta/(Th5n(i)^aq-Tl5n(i)^aq) ...
```

```
            *quadl(J1int,Tl5n(i),Th5n(i)));

    J52(i)=Ju(i)*(2*pi*omega*aq*delta/(Th5(i)^aq-Tl5(i)^aq) ...
           *quadl(J2int,Tl5(i),Th5(i))+1/(2*pi*omega*Tm5(i)));
    J52n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th5n(i)^aq-Tl5n(i)^aq) ...
            *quadl(J2int,Tl5n(i),Th5n(i))+1/(2*pi*omega*Tm5n(i)));

    J61(i)=Ju(i)*(1+dlnJu6(i)+aq*delta/(Th6(i)^aq-Tl6(i)^aq) ...
           *quadl(J1int,Tl6(i),Th6(i)));
    J61n(i)=Ju(i)*(1+dlnJu6(i)+aq*delta/(Th6n(i)^aq-Tl6n(i)^aq) ...
            *quadl(J1int,Tl6n(i),Th6n(i)));

    J62(i)=Ju(i)*(2*pi*omega*aq*delta/(Th6(i)^aq-Tl6(i)^aq) ...
           *quadl(J2int,Tl6(i),Th6(i))+1/(2*pi*omega*Tm6(i)));
    J62n(i)=Ju(i)*(2*pi*omega*aq*delta/(Th6n(i)^aq-Tl6n(i)^aq) ...
            *quadl(J2int,Tl6n(i),Th6n(i))+1/(2*pi*omega*Tm6n(i)));

    J1(i)=Ju(i)*(1+PIdlnJu(i)+aq*delta/(PITh(i)^aq-PITl(i)^aq) ...
          *quadl(J1int,PITl(i),PITh(i)));
    J1n(i)=Ju(i)*(1+PIdlnJu(i)+aq*delta/(Thn(i)^aq-Tln(i)^aq) ...
           *quadl(J1int,Tln(i),Thn(i)));

    J2(i)=Ju(i)*(2*pi*omega*aq*delta/(PITh(i)^aq-PITl(i)^aq) ...
          *quadl(J2int,PITl(i),PITh(i))+1/(2*pi*omega*PITm(i)));
    J2n(i)=Ju(i)*(2*pi*omega*aq*delta/(Thn(i)^aq-Tln(i)^aq) ...
           *quadl(J2int,Tln(i),Thn(i))+1/(2*pi*omega*Tmn(i)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Here we ensure that the arctan function for the dehydration water          %
% model does not go below the dry water model value.                         %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if CH2O(i)<CH2Od
        CH2O(i)=CH2Od;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of shear modulus (G) from formulation in Faul & Jackson (2005). %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

G1=(J11.^2+J12.^2).^(-1/2);
G1n=(J11n.^2+J12n.^2).^(-1/2);
G1a=((J11./Ju.*Jur).^2+(J12./Ju.*Jur).^2).^(-1/2);
G1na=((J11n./Ju.*Jur).^2+(J12n./Ju.*Jur).^2).^(-1/2);
```

```
G2=(J21.^2+J22.^2).^(-1/2);
G2n=(J21n.^2+J22n.^2).^(-1/2);
G2a=((J21./Ju.*Jur).^2+(J22./Ju.*Jur).^2).^(-1/2);
G2na=((J21n./Ju.*Jur).^2+(J22n./Ju.*Jur).^2).^(-1/2);

G3=(J31.^2+J32.^2).^(-1/2);
G3n=(J31n.^2+J32n.^2).^(-1/2);
G3a=((J31./Ju.*Jur).^2+(J32./Ju.*Jur).^2).^(-1/2);
G3na=((J31n./Ju.*Jur).^2+(J32n./Ju.*Jur).^2).^(-1/2);

G4=(J41.^2+J42.^2).^(-1/2);
G4n=(J41n.^2+J42n.^2).^(-1/2);
G4a=((J41./Ju.*Jur).^2+(J42./Ju.*Jur).^2).^(-1/2);
G4na=((J41n./Ju.*Jur).^2+(J42n./Ju.*Jur).^2).^(-1/2);

G5=(J51.^2+J52.^2).^(-1/2);
G5n=(J51n.^2+J52n.^2).^(-1/2);
G5a=((J51./Ju.*Jur).^2+(J52./Ju.*Jur).^2).^(-1/2);
G5na=((J51n./Ju.*Jur).^2+(J52n./Ju.*Jur).^2).^(-1/2);

G6=(J61.^2+J62.^2).^(-1/2);
G6n=(J61n.^2+J62n.^2).^(-1/2);
G6a=((J61./Ju.*Jur).^2+(J62./Ju.*Jur).^2).^(-1/2);
G6na=((J61n./Ju.*Jur).^2+(J62n./Ju.*Jur).^2).^(-1/2);

Gw=(J1.^2+J2.^2).^(-1/2);
Gwn=(J1n.^2+J2n.^2).^(-1/2);
Gwa=((J1./Ju.*Jur).^2+(J2./Ju.*Jur).^2).^(-1/2);
Gwna=((J1n./Ju.*Jur).^2+(J2n./Ju.*Jur).^2).^(-1/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of attenuation using both formulation from Faul &               %
% Jackson (2005 - PIQ) and Karato (2003 - Qsb).                               %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PI1Q=J11./J12;
PI1Qn=J11n./J12n;
Qsb1=1./(B1.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

PI2Q=J21./J22;
PI2Qn=J21n./J22n;
Qsb2=1./(B2.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

PI3Q=J31./J32;
PI3Qn=J31n./J32n;
Qsb3=1./(B3.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

PI4Q=J41./J42;
PI4Qn=J41n./J42n;
Qsb4=1./(B4.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));
```

```
PI5Q=J51./J52;
PI5Qn=J51n./J52n;
Qsb5=1./(B5.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

PI6Q=J61./J62;
PI6Qn=J61n./J62n;
Qsb6=1./(B6.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

PIQ=J1./J2;
PIQn=J1n./J2n;
Qsb=1./(B.*(2.*pi.*omega).^-aq.*exp(-aq.*(E+PR.*V1)./(R.*T)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Setup of null hypothesis (Vinf) from Stixrude & Lithgow-Bertelloni  %
% (2005) for use in calculating Vs.                                           %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Vinf=4.77+(.0380.*3300.*9.81.*-PIz./1e9)-(.000378.*(T-300));
Vinf=Vinf.*1e3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Calculation of Vs using both Jackson (1993 – PIV) and a combination %
% of Vinf from Stixrude & Lithgow-Bertelloni (2005) and Karato (2003) %
% using Q from Faul & Jackson (2005).                                         %
%                                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PIV=(rho.*J1./1e9).^(-1/2);
Vsb=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PIQ));

PIV1=(rho.*J11./1e9).^(-1/2);
Vsb1=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI1Q));

PIV2=(rho.*J21./1e9).^(-1/2);
Vsb2=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI2Q));

PIV3=(rho.*J31./1e9).^(-1/2);
Vsb3=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI3Q));

PIV4=(rho.*J41./1e9).^(-1/2);
Vsb4=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI4Q));

PIV5=(rho.*J51./1e9).^(-1/2);
Vsb5=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI5Q));

PIV6=(rho.*J61./1e9).^(-1/2);
Vsb6=Vinf.*(1-.5.*cot(pi.*.27./2).*(1./PI6Q));
```