COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)

# BurnMan

a thermodynamics and thermoelasticity toolkit

User Manual
Version 0.9.0

Sanne Cottaar
Timo Heister
Robert Myhill
Ian Rose
Cayman Unterborn

http://geodynamics.org

# CONTENTS

The code documentation including class and function description can be found online at burnman.org/current-doc.

# CHAPTER ONE

# BURNMAN

BurnMan is an open source mineral physics toolbox written in Python which determines the velocities of seismic waves in mineral assemblages at high pressure and temperature. It was designed to calculate seismic velocities in the lower mantle, but is equally suited to any part of the solid Earth (or indeed any of the terrestrial planets). BurnMan calculates the isotropic thermoelastic moduli by solving the equations-of-state for a mixture of minerals defined by the user. The user may select from an extensive list of minerals obtained from published databases. Alternatively, they can easily define their own minerals.

Features:

- a range of thermoelastic models, choice between second or third order accuracy
- a range of thermodynamic models for mineral endmembers
- consistent, comprehensive treatment of minerals with solid solutions
- form composites of arbitrary combination of `materials`
- extensive `mineral_database`
- easy plotting and comparison of seismic profiles using matplotlib
- many examples highlighting different features of BurnMan
- different averaging schemes for seismic velocities in composite materials
- a catalogue of published geotherms
- extensible: all parts can be replaced by user-written modules if desired

Please cite:

- Cottaar S., Heister, T., Rose, I., and Unterborn, C., 2014, BurnMan: A lower mantle mineral physics toolkit, Geochemistry, Geophysics, and Geosystems, 15(4), 1164-1179 (link)

Acknowledgement and Support:

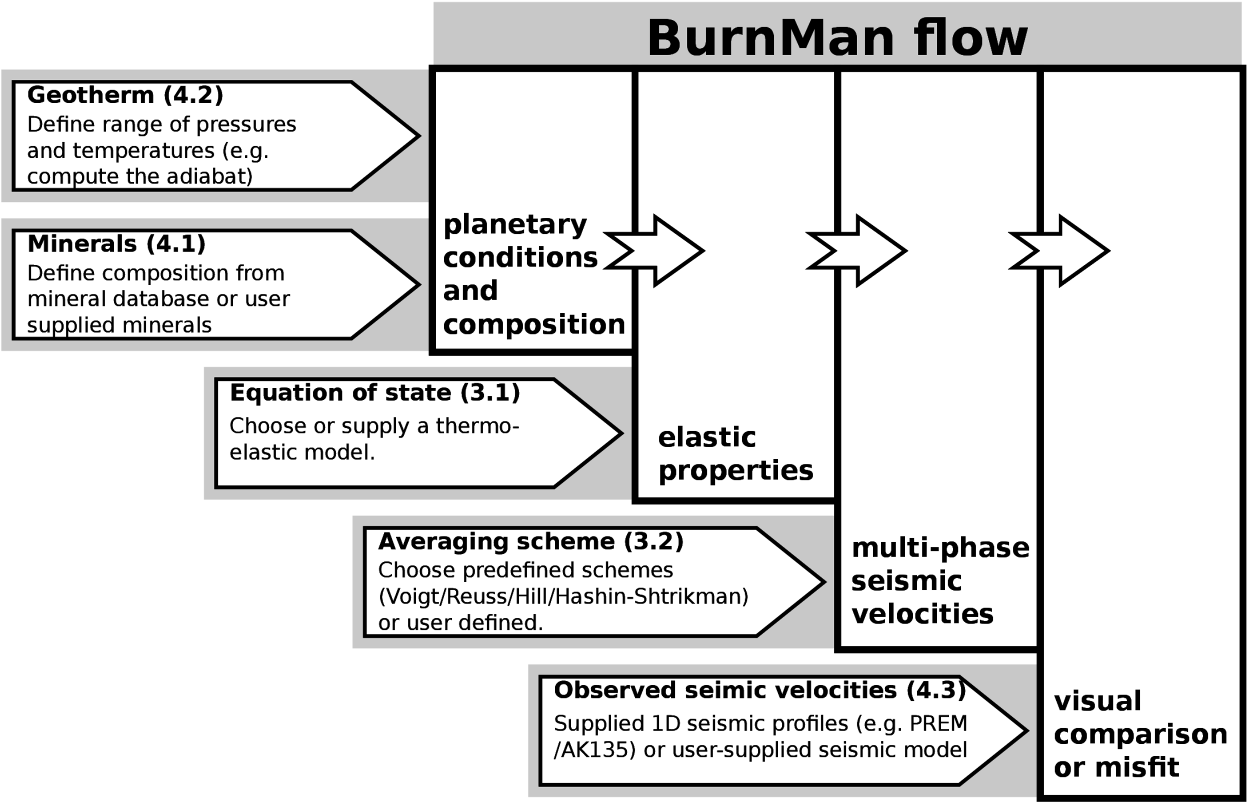
- This project was initiated at, and follow-up research support was received through, Cooperative Institute of Deep Earth Research, CIDER (NSF FESD grant 1135452) – see www.deep-earth.org
- We thank all the fellow members of the CIDER Mg/Si team for their input: Valentina Magni, Yu Huang, JiaChao Liu, Marc Hirschmann, and Barbara Romanowicz.
- We thank Lars Stixrude for providing benchmarking calculations.

- We thank CIG (www.geodynamics.org) for support and accepting our donation of BurnMan as an official project.
- We also welcomed helpful discussions with Zack Geballe, Motohiko Murakami, Bill McDonough, Quentin Williams, Wendy Panero, and Wolfgang Bangerth.

# CHAPTER TWO

# OVERVIEW

## 2.1 Overall Structure


BurnMan flow
Geotherm (4.2)
Define range of pressures and temperatures (e.g. compute the adiabat)
Minerals (4.1)
Define composition from mineral database or user supplied minerals
Equation of state (3.1)
Choose or supply a thermo-elastic model.
Averaging scheme (3.2)
Choose predefined schemes (Voigt/Reuss/Hill/Hashin-Shtrikman) or user defined.
Observed seimic velocities (4.3)
Supplied 1D seismic profiles (e.g. PREM /AK135) or user-supplied seismic model
planetary conditions and composition
elastic properties
multi-phase seismic velocities
visual comparison or misfit

The goal of BurnMan is to calculate seismic velocity profiles for a given mineral composition, geotherm, EoS, and averaging scheme. These calculated seismic velocity profiles can then be compared (either graphically or quantitatively) to profiles computed for other compositions or constrained by seismology. It is written in the Python language and is run from the command line. This allows the library to be incorporated into other projects. BurnMan makes extensive use of SciPy and NumPy, which are widely used Python libraries for scientific computation. Matplotlib is used to display results and produce publication quality figures. The computations are consistently done in SI units, although for this paper we convert units for plotting purposes. A large collection of annotated examples on the usage of BurnMan are provided. Scripts to reproduce the figures in this paper are included in the toolbox. We are happy to accept contributions in form of corrections, examples, or new features.

The figure above shows each of the various steps in BurnMan and the input required at each step. The user sets the composition and temperature and pressure profiles. At later steps, the user can pick from several existing methodologies or supply an alternative implementation. This makes BurnMan very extensible and allows for many combinations and configurations with which to run calculations. Eventually, one can choose or provide a seismic model for comparison.

This flow setup can be used to evaluate the isotropic seismic velocities in a geodynamic model or as the forward problem when inverting seismic profiles or seismic velocity variations for mineralogical compositions and temperature. The modular components of BurnMan can also be used separately or combined in different ways than shown in the figure above. For example, one can input experimental results at certain pressures and temperatures and fit reference elastic moduli for a specific EoS. Additionally, one can implement their own alternatives for each of the existing modules. BurnMan has the potential to expand to other planetary applications. While all the features are modular, everything is available in a single library.

CHAPTER

THREE

# MATHEMATICAL BACKGROUND

Here is a bit of background on the methods used to calculate thermoelastic and thermodynamic properties in BurnMan. More detail can be found in the cited papers.

## 3.1 Calculating Thermoelastic Properties

To calculate the bulk ($K$) modulus, shear modulus ($G$) and density ($\rho$) of a material at a given pressure ($P$) and temperature ($T$), optionally defined by a geotherm) and determine the seismic velocities ($V_S, V_P, V_\Phi$), one uses an Equation of State (EoS). Currently the following EoSs are supported in BurnMan:

- Birch-Murnaghan finite-strain EoS (excludes temperature effects, [Poirier1991]),
- Birch-Murnaghan finite-strain EoS with a Mie-Grüneisen-Debye thermal correction, as formulated by [Stixrude2005].
- Birch-Murnaghan finite-strain EoS with a Mie-Grüneisen-Debye thermal correction, as formulated by [Matas2007].
- Modified Tait EoS (excludes temperature effects, [HC1974]),
- Modified Tait EoS with a pseudo-Einstein model for thermal corrections, as formulated by [HP2011].
- Compensated-Redlich-Kwong for fluids, as formulated by [HP1991].

To calculate these thermoelastic parameters, the EoS requires the user to input the pressure, temperature, and the phases and their molar fractions. These inputs and outputs are further discussed in *User input*.

### 3.1.1 Birch-Murnaghan (isothermal)

The Birch-Murnaghan equation is an isothermal Eulerian finite-strain EoS relating pressure and volume. The negative finite-strain (or compression) is defined as

$$f = \frac{1}{2}\left[\left(\frac{V}{V_0}\right)^{-2/3} - 1\right], \tag{3.1}$$

where $V$ is the volume at a given pressure and $V_0$ is the volume at a reference state ($P = 10^5$ Pa , $T$ = 300 K). The pressure and elastic moduli are derived from a third-order Taylor expansion of Helmholtz free energy in $f$ and evaluating the appropriate volume and strain derivatives (e.g., [Poirier1991]). For an isotropic material one obtains for the pressure, isothermal bulk modulus, and shear modulus:

$$P = 3K_0 f \left(1+2f\right)^{5/2} \left[1 + \frac{3}{2}\left(K_0' - 4\right) f\right], \tag{3.2}$$

$$\begin{aligned} K_T = (1+2f)^{5/2} \bigg[ & K_0 + (3K_0K'_0 - 5K_0)f \\ & + \frac{27}{2}(K_0K'_0 - 4K_0)f^2 \bigg], \end{aligned} \tag{3.3}$$

$$\begin{aligned} G = (1+2f)^{5/2} \bigg[ & G_0 + (3K_0G'_0 - 5G_0)f \\ & + (6K_0G'_0 - 24K_0 - 14G_0 + \frac{9}{2}K_0K'_0)f^2 \bigg]. \end{aligned} \tag{3.4}$$

Here $K_0$ and $G_0$ are the reference bulk modulus and shear modulus and $K'_0$ and $G'_0$ are the derivative of the respective moduli with respect to pressure.

BurnMan has the option to use the second-order expansion for shear modulus by dropping the $f^2$ terms in these equations (as is sometimes done for experimental fits or EoS modeling).

### 3.1.2 Modified Tait (isothermal)

The Modified Tait equation of state was developed by [HC1974]. It has the considerable benefit of allowing volume to be expressed as a function of pressure. It performs very well to pressures and temperatures relevant to the deep Earth [HP2011].

$$\begin{aligned} \frac{V_{P,T}}{V_{1bar,298K}} &= 1 - a(1-(1+bP)^{-c}), \\ a &= \frac{1+K'_0}{1+K'_0+K_0K''_0}, \\ b &= \frac{K'_0}{K_0} - \frac{K''_0}{1+K'_0}, \\ c &= \frac{1+K'_0+K_0K''_0}{K'^2_0+K'_0-K_0K''_0} \end{aligned} \tag{3.5}$$

### 3.1.3 Mie-Grüneisen-Debye (thermal correction to Birch-Murnaghan)

The Debye model for the Helmholtz free energy can be written as follows [Matas2007]

$$\begin{aligned} \mathcal{F} &= \frac{9nRT}{V}\frac{1}{x^3}\int_0^x \xi^2 \ln(1-e^{-\xi})d\xi, \\ x &= \theta/T, \\ \theta &= \theta_0 \exp\left(\frac{\gamma_0-\gamma}{q_0}\right), \\ \gamma &= \gamma_0 \left(\frac{V}{V_0}\right)^{q_0} \end{aligned}$$

where $\theta$ is the Debye temperature and $\gamma$ is the Grüneisen parameter.

Using thermodynamic relations we can derive equations for the thermal pressure and bulk modulus

$$\begin{aligned}
P_{th}(V,T) &= -\frac{\partial \mathcal{F}(\mathcal{V},\mathcal{T})}{\partial V}, \\
&= \frac{3n\gamma RT}{V} D(x), \\
K_{th}(V,T) &= -V\frac{\partial P(V,T)}{\partial V}, \\
&= \frac{3n\gamma RT}{V}\gamma\left[(1-q_0-3\gamma)D(x)+3\gamma\frac{x}{e^x-1}\right], \\
D(x) &= \frac{3}{x^3}\int_0^x \frac{\xi^3}{e^\xi-1}d\xi
\end{aligned}$$

The thermal shear correction used in BurnMan was developed by [HS1998]

$$G_{th}(V,T) = \frac{3}{5}\left[K_{th}(V,T) - 2\frac{3nRT}{V}\gamma D(x)\right]$$

The total pressure, bulk and shear moduli can be calculated from the following sums

$$\begin{aligned}
P(V,T) &= P_{\text{ref}}(V,T_0) + P_{th}(V,T) - P_{th}(V,T_0), \\
K(V,T) &= K_{\text{ref}}(V,T_0) + K_{th}(V,T) - K_{th}(V,T_0), \\
G(V,T) &= G_{\text{ref}}(V,T_0) + G_{th}(V,T) - G_{th}(V,T_0)
\end{aligned}$$

This equation of state is substantially the same as that in SLB2005 (see below). The primary differences are in the thermal correction to the shear modulus and in the volume dependences of the Debye temperature and the Gruneisen parameter.

### 3.1.4 HP2011 (thermal correction to Modified Tait)

The thermal pressure can be incorporated into the Modified Tait equation of state, replacing $P$ with $P - (P_{\text{th}} - P_{\text{th0}})$ in Equation (3.5) [HP2011]. Thermal pressure is calculated using a Mie-Grüneisen equation of state and an Einstein model for heat capacity, even though the Einstein model is not actually used for the heat capacity when calculating the enthalpy and entropy (see following section).

$$\begin{aligned}
P_{\text{th}} &= \frac{\alpha_0 K_0 E_{\text{th}}}{C_{V0}}, \\
E_{\text{th}} &= 3nR\Theta\left(0.5 + \frac{1}{\exp(\frac{\Theta}{T}) - 1}\right), \\
C_V &= 3nR\frac{(\frac{\Theta}{T})^2\exp(\frac{\Theta}{T})}{(\exp(\frac{\Theta}{T})-1)^2}
\end{aligned}$$

$\Theta$ is the Einstein temperature of the crystal in Kelvin, approximated for a substance $i$ with $n_i$ atoms in the unit formula and a molar entropy $S_i$ using the empirical formula

$$\Theta_i = \frac{10636}{S_i/n_i + 6.44}$$

### 3.1.5 SLB2005 (for solids, thermal)

Thermal corrections for pressure, and isothermal bulk modulus and shear modulus are derived from the Mie-Grüneisen-Debye EoS with the quasi-harmonic approximation. Here we adopt the formalism of [Stixrude2005] where these corrections are added to equations (3.2)–(3.4):

$$\begin{aligned}P_{th}(V,T) &= \frac{\gamma \Delta \mathcal{U}}{V}, \\ K_{th}(V,T) &= (\gamma + 1 - q)\frac{\gamma \Delta \mathcal{U}}{V} - \gamma^2 \frac{\Delta(C_V T)}{V}, \\ G_{th}(V,T) &= -\frac{\eta_S \Delta \mathcal{U}}{V}.\end{aligned} \tag{3.6}$$

The $\Delta$ refers to the difference in the relevant quantity from the reference temperature (300 K). $\gamma$ is the Grüneisen parameter, $q$ is the logarithmic volume derivative of the Grüneisen parameter, $\eta_S$ is the shear strain derivative of the Grüneisen parameter, $C_V$ is the heat capacity at constant volume, and $\mathcal{U}$ is the internal energy at temperature $T$. $C_V$ and $\mathcal{U}$ are calculated using the Debye model for vibrational energy of a lattice. These quantities are calculated as follows:

$$\begin{aligned}C_V &= 9nR\left(\frac{T}{\theta}\right)^3 \int_0^{\frac{\theta}{T}} \frac{e^\tau \tau^4}{(e^\tau - 1)^2} d\tau, \\ \mathcal{U} &= 9nRT\left(\frac{T}{\theta}\right)^3 \int_0^{\frac{\theta}{T}} \frac{\tau^3}{(e^\tau - 1)} d\tau, \\ \gamma &= \frac{1}{6}\frac{\nu_0^2}{\nu^2}(2f+1)\left[a_{ii}^{(1)} + a_{iikk}^{(2)} f\right], \\ q &= \frac{1}{9\gamma}\left[18\gamma^2 - 6\gamma - \frac{1}{2}\frac{\nu_0^2}{\nu^2}(2f+1)^2 a_{iikk}^{(2)}\right], \\ \eta_S &= -\gamma - \frac{1}{2}\frac{\nu_0^2}{\nu^2}(2f+1)^2 a_S^{(2)}, \\ \frac{\nu^2}{\nu_0^2} &= 1 + a_{ii}^{(1)} f + \frac{1}{2} a_{iikk}^{(2)} f^2, \\ a_{ii}^{(1)} &= 6\gamma_0, \\ a_{iikk}^{(2)} &= -12\gamma_0 + 36\gamma_0^2 - 18 q_0 \gamma_0, \\ a_S^{(2)} &= -2\gamma_0 - 2\eta_{S0},\end{aligned}$$

where $\theta$ is the Debye temperature of the mineral, $\nu$ is the frequency of vibrational modes for the mineral, $n$ is the number of atoms per formula unit (e.g. 2 for periclase, 5 for perovskite), and $R$ is the gas constant. Under the approximation that the vibrational frequencies behave the same under strain, we may identify $\nu/\nu_0 = \theta/\theta_0$. The quantities $\gamma_0$, $\eta_{S0}$ $q_0$, and $\theta_0$ are the experimentally determined values for those parameters at the reference state.

Due to the fact that a planetary mantle is rarely isothermal along a geotherm, it is more appropriate to use the adiabatic bulk modulus $K_S$ instead of $K_T$, which is calculated using

$$K_S = K_T(1 + \gamma \alpha T), \tag{3.7}$$

where $\alpha$ is the coefficient of thermal expansion:

$$\alpha = \frac{\gamma C_V V}{K_T}. \tag{3.8}$$

There is no difference between the isothermal and adiabatic shear moduli for an isotropic solid. All together this makes an eleven parameter EoS model, which is summarized in the Table below. For more details on the EoS, we refer readers to [Stixrude2005].

| User Input | Symbol | Definition | Units |
|---|---|---|---|
| V_0 | $V_0$ | **Volume at P =** $10^5$ Pa , T = 300 K | m $^3$ mol $^{-1}$ |
| K_0 | $K_0$ | Isothermal bulk modulus at *P=10^5* Pa, T = 300 K | Pa |
| Kprime_0 | $K'_0$ | Pressure derivative of $K_0$ | |
| G_0 | $G_0$ | Shear modulus at P = $10^5$ Pa, T = 300 K | Pa |
| Gprime_0 | $G'_0$ | Pressure derivative of $G_0$ | |
| molar_mass | $\mu$ | mass per mole formula unit | kg mol$^{-1}$ |
| n | n | number of atoms per formula unit | |
| Debye_0 | $\theta_0$ | Debye Temperature | K |
| grueneisen_0 | $\gamma_0$ | Grüneisen parameter at P = $10^5$ Pa, T = 300 K | |
| q0 | $q_0$ | Logarithmic volume derivative of the Grüneisen parameter | |
| eta_s_0 | $\eta_{S0}$ | Shear strain derivative of the Grüneisen parameter | |

This equation of state is substantially the same as that of the Mie-Gruneisen-Debye (see above). The primary differences are in the thermal correction to the shear modulus and in the volume dependences of the Debye temperature and the Gruneisen parameter.

### 3.1.6 Compensated-Redlich-Kwong (for fluids, thermal)

The CORK equation of state [HP1991] is a simple virial-type extension to the modified Redlich-Kwong (MRK) equation of state. It was designed to compensate for the tendency of the MRK equation of state to overestimate volumes at high pressures and accommodate the volume behaviour of coexisting gas and liquid phases along the saturation curve.

$$\begin{aligned}
V &= \frac{RT}{P} + c_1 - \frac{c_0 R T^{0.5}}{(RT + c_1 P)(RT + 2c_1 P)} + c_2 P^{0.5} + c_3 P, \\
c_0 &= c_{0,0} T_c^{2.5}/P_c + c_{0,1} T_c^{1.5}/P_c T, \\
c_1 &= c_{1,0} T_c/P_c, \\
c_2 &= c_{2,0} T_c/P_c^{1.5} + c_{2,1}/P_c^{1.5} T, \\
c_3 &= c_{3,0} T_c/P_c^2 + c_{3,1}/P_c^2 T
\end{aligned}$$

## 3.2 Calculating Thermodynamic Properties

At any given pressure and temperature, the equilibrium assemblage of minerals is that which minimises the Gibbs Free Energy of the system. The Gibbs free energy is the maximum amount of non-expansion work that can be extracted from a closed system. For each phase, the Gibbs free energy is equal to

$$\mathcal{G} = \mathcal{U} - T\mathcal{S} + PV = \mathcal{H} - TS = \mathcal{F} + PV \tag{3.9}$$

where $P$ is the pressure, $T$ is the temperature and $\mathcal{U}$, $\mathcal{F}$, $\mathcal{H}$, $\mathcal{S}$ and $V$ are the molar internal energy, Helmholtz Free Energy, enthalpy, entropy and volume respectively. The total energy of the system is then

$$\mathcal{G}_{\textrm{system}} = \sum_i n_i \mathcal{G}_i \tag{3.10}$$

### 3.2.1 HP2011

$$\begin{aligned}
\mathcal{G}(P,T) &= \mathcal{H}_{\textrm{1 bar, T}} - T\mathcal{S}_{\textrm{1 bar, T}} + \int_{\textrm{1 bar}}^{P} V(P,T)\, dP, \\
\mathcal{H}_{\textrm{1 bar, T}} &= \Delta_f \mathcal{H}_{\textrm{1 bar, 298 K}} + \int_{298}^{T} C_P \, dT, \\
\mathcal{S}_{\textrm{1 bar, T}} &= \mathcal{S}_{\textrm{1 bar, 298 K}} + \int_{298}^{T} \frac{C_P}{T} \, dT, \\
\int_{\textrm{1 bar}}^{P} V(P,T)\, dP &= PV_0 \left( 1 - a + \left( a \frac{(1 - bP_{th})^{1-c} - (1 + b(P - P_{th}))^{1-c}}{b(c-1)P} \right) \right)
\end{aligned} \tag{3.11}$$

The heat capacity at one bar is given by an empirical polynomial fit to experimental data

$$C_p = a + bT + cT^{-2} + dT^{-0.5}$$

The entropy at high pressure and temperature can be calculated by differentiating the expression for $\mathcal{G}$ with respect to temperature

$$\begin{aligned}
\mathcal{S}(P,T) &= S_{\textrm{1 bar, T}} + \frac{\partial \int V dP}{\partial T}, \\
\frac{\partial \int V dP}{\partial T} &= V_0 \alpha_0 K_0 a \frac{C_{V0}(T)}{C_{V0}(T_{\textrm{ref}})} ((1 + b(P - P_{th}))^{-c} - (1 - bP_{th})^{-c})
\end{aligned}$$

Finally, the enthalpy at high pressure and temperature can be calculated

$$\mathcal{H}(P,T) = \mathcal{G}(P,T) + T\mathcal{S}(P,T)$$

### 3.2.2 SLB2005

The Debye model yields the Helmholtz free energy and entropy due to lattice vibrations

$$\begin{aligned}
\mathcal{G} &= \mathcal{F} + PV, \\
\mathcal{F} &= nRT\left(3\ln(1-e^{-\frac{\theta}{T}}) - \int_0^{\frac{\theta}{T}} \frac{\tau^3}{(e^{\tau}-1)}d\tau\right), \\
\mathcal{S} &= nR\left(4\int_0^{\frac{\theta}{T}} \frac{\tau^3}{(e^{\tau}-1)}d\tau - 3\ln(1-e^{-\frac{\theta}{T}})\right), \\
\mathcal{H} &= \mathcal{G} + T\mathcal{S}
\end{aligned}$$

## 3.3 Calculating Solid Solution Properties

Many minerals can exist over a range of compositions. The compositional domains of minerals with a common crystal structure are called solid solutions. Different elements substitute for one another within distinct crystallographic sites in the structure. For example, low pressure silicate garnets have two distinct sites on which mixing takes place; a dodecahedral site (3 per unit cell) and octahedral site (2 per unit cell). The chemical formula of many low pressure garnets exist within the solid solution:

$$\text{[Mg,Fe,Mn,Ca]}_3\text{[Al,Fe,Cr]}_2\text{Si}_3\text{O}_{12}$$

A solid solution is not simply a mechanical mixture of its constituent endmembers. Most fundamentally, the mixing of different elements on sites results in an excess configurational entropy

$$\mathcal{S}_{\text{conf}} = R\ln\prod_s (X_c^s)^{\nu}$$

where $s$ is a site in the lattice $M$, $c$ are the cations mixing on site $s$ and $\nu$ is the number of $s$ sites in the formula unit. Solid solutions where this configurational entropy is the only deviation from a mechanical mixture are termed *ideal*.

Many solid solutions exhibit further deviations from ideality, which arise as a result of interactions between ions with different physical and chemical characteristics. Regular solid solution models are designed to account for this, by allowing the addition of excess enthalpies, entropies and volumes to the solution model. These excess terms have the matrix form [DPWH2007]

$$\alpha^T p(\phi^T W \phi)$$

where $p$ is a vector of molar fractions of each of the $n$ endmembers, $\alpha$ is a vector of “van Laar parameters” governing asymmetry in the excess properties, and

$$\begin{aligned}
\phi_i &= \frac{\alpha_i p_i}{\sum_{k=1}^{n} \alpha_k p_k}, \\
W_{ij} &= \frac{2w_{ij}}{\alpha_i + \alpha_j}\text{for i<j}
\end{aligned}$$

The $w_{ij}$ terms are a set of interaction terms between endmembers $i$ and $j$. If all the $\alpha$ terms are equal to unity, a non-zero $w$ yields an excess with a quadratic form and a maximum of $w/4$ half-way between the two endmembers.

From the preceeding equations, we can define the thermodynamic potentials of solid solutions:

$$\begin{aligned}
\mathcal{H}_{\text{SS}} &= \sum_i n_i \mathcal{H}_i + \mathcal{H}_{\text{excess}} + PV_{\text{excess}} \\
\mathcal{S}_{\text{SS}} &= \sum_i n_i \mathcal{S}_i + \mathcal{S}_{\text{conf}} + \mathcal{S}_{\text{excess}} \\
\mathcal{G}_{\text{SS}} &= \mathcal{H}_{\text{SS}} - T\mathcal{S}_{\text{SS}} \\
V_{\text{SS}} &= \sum_i n_i V_i + V_{\text{excess}}
\end{aligned}$$

We can also define the derivatives of volume with respect to pressure and temperature

$$\begin{aligned}
\alpha_{P,\text{SS}} &= \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_P = \left(\frac{1}{V_{\text{SS}}}\right)\left(\sum_i (n_i\,\alpha_i\,V_i)\right) \\
K_{T,\text{SS}} &= V\left(\frac{\partial P}{\partial V}\right)_T = V_{\text{SS}}\left(\frac{1}{\sum_i\left(n_i\frac{V_i}{K_{Ti}}\right)} + \frac{\partial P}{\partial V_{\text{excess}}}\right)
\end{aligned}$$

Making the approximation that the excess entropy has no temperature dependence

$$\begin{aligned}
C_{P,\text{SS}} &= \sum_i n_i C_{Pi} \\
C_{V,\text{SS}} &= C_{P,\text{SS}} - V_{\text{SS}}\,T\,\alpha_{\text{SS}}^2\,K_{T,\text{SS}} \\
K_{S,\text{SS}} &= K_{T,\text{SS}}\,\frac{C_{P,\text{SS}}}{C_{V,\text{SS}}} \\
\gamma_{\text{SS}} &= \frac{\alpha_{\text{SS}}\,K_{T,\text{SS}}\,V_{\text{SS}}}{C_{V,\text{SS}}}
\end{aligned}$$

### 3.3.1 Including order-disorder

Order-disorder can be treated trivially with solid solutions. The only difference between mixing between ordered and disordered endmembers is that disordered endmembers have a non-zero configurational entropy, which must be accounted for when calculating the excess entropy within a solid solution.

### 3.3.2 Including spin transitions

The regular solid solution formalism should provide an elegant way to model spin transitions in phases such as periclase and bridgmanite. High and low spin iron can be treated as different elements, providing distinct endmembers and an excess configurational entropy. Further excess terms can be added as necessary.

## 3.4 Calculating Multi-phase Aggregate Properties

### 3.4.1 Averaging schemes

After the thermoelastic parameters ($K_S$, $G$, $\rho$) of each phase are determined at each pressure and/or temperature step, these values must be combined to determine the seismic velocity of a multiphase assemblage.

We define the volume fraction of the individual minerals in an assemblage:

$$\nu_i = n_i \frac{V_i}{V},$$

where $V_i$ and $n_i$ are the molar volume and the molar fractions of the $i$ th individual phase, and $V$ is the total molar volume of the assemblage:

$$V = \sum_i n_i V_i. \tag{3.12}$$

The density of the multiphase assemblage is then

$$\rho = \sum_i \nu_i \rho_i = \frac{1}{V} \sum_i n_i \mu_i, \tag{3.13}$$

where $\rho_i$ is the density and $\mu_i$ is the molar mass of the $i$ th phase.

Unlike density and volume, there is no straightforward way to average the bulk and shear moduli of a multiphase rock, as it depends on the specific distribution and orientation of the constituent minerals. BurnMan allows several schemes for averaging the elastic moduli: the Voigt and Reuss bounds, the Hashin-Shtrikman bounds, the Voigt-Reuss-Hill average, and the Hashin-Shtrikman average [Watt1976].

The Voigt average, assuming constant strain across all phases, is defined as

$$X_V = \sum_i \nu_i X_i, \tag{3.14}$$

where $X_i$ is the bulk or shear modulus for the $i$ th phase. The Reuss average, assuming constant stress across all phases, is defined as

$$X_R = \left( \sum_i \frac{\nu_i}{X_i} \right)^{-1}. \tag{3.15}$$

The Voigt-Reuss-Hill average is the arithmetic mean of Voigt and Reuss bounds:

$$X_{VRH} = \frac{1}{2} \left( X_V + X_R \right). \tag{3.16}$$

The Hashin-Shtrikman bounds make an additional assumption that the distribution of the phases is statistically isotropic and are usually much narrower than the Voigt and Reuss bounds [Watt1976]. This may be a poor assumption in regions of Earth with high anisotropy, such as the lowermost mantle, however these bounds are more physically motivated than the commonly-used Voigt-Reuss-Hill average. In most instances, the Voigt-Reuss-Hill average and the arithmetic mean of the Hashin-Shtrikman bounds are quite similar with the pure arithmetic mean (linear averaging) being well outside of both.

It is worth noting that each of the above bounding methods are derived from mechanical models of a linear elastic composite. It is thus only appropriate to apply them to elastic moduli, and not to other thermoelastic properties, such as wave speeds or density.

### 3.4.2 Computing seismic velocities

Once the moduli for the multiphase assemblage are computed, the compressional ($P$), shear ($S$) and bulk sound ($\Phi$) velocities are then result from the equations:

$$V_P = \sqrt{\frac{K_S + \frac{4}{3}G}{\rho}}, \qquad V_S = \sqrt{\frac{G}{\rho}}, \qquad V_\Phi = \sqrt{\frac{K_S}{\rho}}. \tag{3.17}$$

To correctly compare to observed seismic velocities one needs to correct for the frequency sensitivity of attenuation. Moduli parameters are obtained from experiments that are done at high frequencies (MHz-GHz) compared to seismic frequencies (mHz-Hz). The frequency sensitivity of attenuation causes slightly lower velocities for seismic waves than they would be for high frequency waves. In BurnMan one can correct the calculated acoustic velocity values to those for long period seismic tomography [Minster1981]:

$$V_{S/P} = V_{S/P}^{\text{uncorr.}} \left(1 - \frac{1}{2}\cot(\frac{\beta\pi}{2})\frac{1}{Q_{S/P}}(\omega)\right).$$

Similar to [Matas2007], we use a $\beta$ value of 0.3, which falls in the range of values of $0.2$ to $0.4$ proposed for the lower mantle (e.g. [Karato1990]). The correction is implemented for $Q$ values of PREM for the lower mantle. As $Q_S$ is smaller than $Q_P$, the correction is more significant for S waves. In both cases, though, the correction is minor compared to, for example, uncertainties in the temperature (corrections) and mineral physical parameters. More involved models of relaxation mechanisms can be implemented, but lead to the inclusion of more poorly constrained parameters, [Matas2007a]. While attenuation can be ignored in many applications [Trampert2001], it might play a significant role in explaining strong variations in seismic velocities in the lowermost mantle [Davies2012].

## 3.5 User input

### 3.5.1 Mineralogical composition

A number of pre-defined minerals are included in the mineral library and users can create their own. The library includes wrapper functions to include a transition from the high-spin mineral to the low-spin mineral [Lin2013] or to combine minerals for a given iron number.

*Standard minerals* – The 'standard' mineral format includes a list of parameters given in the above table. Each mineral includes a suggested EoS with which the mineral parameters are derived. For some minerals the parameters for the thermal corrections are not yet measured or calculated, and therefore the corrections can not be applied. An occasional mineral will not have a measured or calculated shear moduli, and therefore can only be used to compute densities and bulk sound velocities. The mineral library is subdivided by citation. BurnMan includes the option to produce a LaTeX; table of the mineral parameters used. BurnMan can be easily setup to incorporate uncertainties for these parameters.

*Minerals with a spin transition* – A standard mineral for the high spin and low spin must be defined separately. These minerals are "wrapped," so as to switch from the high spin to high spin mineral at a give pressure. While not realistic, for the sake of simplicity, the spin transitions are considered to be sharp at a given pressure.

*Minerals depending on Fe partitioning* – The wrapper function can partition iron, for example between ferropericlase, fp, and perovskite, pv. It requires the input of the iron mol fraction with regards to Mg, $X_{\text{fp}}$

and $X_{\mathrm{pv}}$, which then defines the chemistry of an Mg-Fe solid solution according to $(\mathrm{Mg}_{1-X_{\mathrm{Fe}}^{\mathrm{fp}}}, \mathrm{Fe}_{X_{\mathrm{Fe}}^{\mathrm{fp}}})\mathrm{O}$ or $(\mathrm{Mg}_{1-X_{\mathrm{Fe}}^{\mathrm{pv}}}, \mathrm{Fe}_{X_{\mathrm{Fe}}^{\mathrm{pv}}})\mathrm{SiO}_3$. The iron mol fractions can be set to be constant or varying with P and T as needed. Alternatively one can calculate the iron mol fraction from the distribution coefficient $K_D$ defined as

$$K_D = \frac{X_{\mathrm{Fe}}^{\mathrm{pv}}/X_{\mathrm{Mg}}^{\mathrm{pv}}}{X_{\mathrm{Fe}}^{\mathrm{fp}}/X_{\mathrm{Mg}}^{\mathrm{fp}}}. \tag{3.18}$$

We adopt the formalism of [Nakajima2012] choosing a reference distribution coefficient $K_{D0}$ and standard state volume change ($\Delta v^0$) for the Fe-Mg exchange between perovskite and ferropericlase

$$K_D = K_{D0} \exp\left(\frac{(P_0 - P)\Delta v^0}{RT}\right), \tag{3.19}$$

where $R$ is the gas constant and $P_0$ the reference pressure. As a default, we adopt the average $\Delta v^0$ of [Nakajima2012] of $2 \cdot 10^{-7}$ $m^3 mol^{-1}$ and suggest using their $K_{D0}$ value of $0.5$.

The multiphase mixture of these minerals can be built by the user in three ways:

1. Molar fractions of an arbitrary number of pre-defined minerals, for example mixing standard minerals mg_perovskite ($\mathrm{MgSiO_3}$), fe_perovskite ($\mathrm{FeSiO_3}$), periclase ($\mathrm{MgO}$) and wüstite ($\mathrm{FeO}$).

2. A two-phase mixture with constant or $(P, T)$ varying Fe partitioning using the minerals that include Fe-dependency, for example mixing $(\mathrm{Mg}, \mathrm{Fe})\mathrm{SiO_3}$ and $(\mathrm{Mg}, \mathrm{Fe})\mathrm{O}$ with a pre-defined distribution coefficient.

3. Weight percents (wt%) of (Mg, Si, Fe) and distribution coefficient (includes (P,T)-dependent Fe partitioning). This calculation assumes that each element is completely oxidized into its corresponding oxide mineral (MgO, FeO, $\mathrm{SiO_2}$) and then combined to form iron-bearing perovskite and ferropericlase taking into account some Fe partition coefficient.

### 3.5.2 Geotherm

Unlike the pressure, the temperature of the lower mantle is relatively unconstrained. As elsewhere, BurnMan provides a number of built-in geotherms, as well as the ability to use user-defined temperature-depth relationships. A geotherm in BurnMan is an object that returns temperature as a function of pressure. Alternatively, the user could ignore the geothermal and compute elastic velocities for a range of temperatures at any give pressure.

Currently, we include geotherms published by [Brown1981] and [anderson1982earth]. Alternatively one can use an adiabatic gradient defined by the thermoelastic properties of a given mineralogical model. For a homogeneous material, the adiabatic temperature profile is given by integrating the ordinary differential equation (ODE)

$$\left(\frac{\mathrm{d}T}{\mathrm{d}P}\right)_S = \frac{\gamma T}{K_S}. \tag{3.20}$$

This equation can be extended to multiphase composite using the first law of thermodynamics to arrive at

$$\left(\frac{\mathrm{d}T}{\mathrm{d}P}\right)_S = \frac{T \displaystyle\sum_i \frac{n_i C_{Pi} \gamma_i}{K_{Si}}}{\displaystyle\sum_i n_i C_{Pi}}, \tag{3.21}$$

where the subscripts correspond to the $i$ th phase, $C_P$ is the heat capacity at constant pressure of a phase, and the other symbols are as defined above. Integrating this ODE requires a choice in anchor temperature ($T_0$) at the top of the lower mantle (or including this as a parameter in an inversion). As the adiabatic geotherm is dependent on the thermoelastic parameters at high pressures and temperatures, it is dependent on the equation of state used.

### 3.5.3 Seismic Models

BurnMan allows for direct visual and quantitative comparison with seismic velocity models. Various ways of plotting can be found in the examples. Quantitative misfits between two profiles include an L2-norm and a chi-squared misfit, but user defined norms can be implemented. A seismic model in BurnMan is an object that provides pressure, density, and seismic velocities ($V_P, V_\Phi, V_S$) as a function of depth.

To compare to seismically constrained profiles, BurnMan provides the 1D seismic velocity model PREM [dziewonski1981]. One can choose to evaluate $V_P, V_\Phi, V_S, \rho, K_S$ and/or $G$. The user can input their own seismic profile, an example of which is included using AK135 [kennett1995].

Besides standardized 1D radial profiles, one can also compare to regionalized average profiles for the lower mantle. This option accommodates the observation that the lowermost mantle can be clustered into two regions, a ‘slow’ region, which represents the so-called Large Low Shear Velocity Provinces, and ‘fast’ region, the continuous surrounding region where slabs might subduct [Lekic2012]. This clustering as well as the averaging of the 1D model occurs over five tomographic S wave velocity models (SAW24B16: [megnin2000]; HMSL-S: [houser2008]; S362ANI: [kustowski2008]; GyPSuM: [Simmons2010]; S40RTS: [Ritsema2011]). The strongest deviations from PREM occur in the lowermost 1000 km. Using the ‘fast’ and ‘slow’ S wave velocity profiles is therefore most important when interpreting the lowermost mantle. Suggestion of compositional variation between these regions comes from seismology [to2005][He2012] as well as geochemistry [Deschamps2012][jackson2010]. Based on thermo-chemical convection models, [Styles2011] also show that averaging profiles in thermal boundary layers may cause problems for seismic interpretation.

We additionally apply cluster analysis to and provide models for P wave velocity based on two tomographic models (MIT-P08: [Li2008]; GyPSuM: [Simmons2012]). The clustering results correlate well with the fast and slow regions for S wave velocities; this could well be due to the fact that the initial model for the P wave velocity models is scaled from S wave tomographic velocity models. Additionally, the variations in P wave velocities are a lot smaller than for S waves. For this reason using these adapted models is most important when comparing the S wave velocities.

While interpreting lateral variations of seismic velocity in terms of composition and temperature is a major goal [Trampert2004][Mosca2012], to determine the bulk composition the current challenge appears to be concurrently fitting absolute P and S wave velocities and incorporate the significant uncertainties in mineral physical parameters).

CHAPTER

FOUR

# EXAMPLES

BurnMan comes with a small tutorial in the tutorial/ folder, and large collection of example programs under examples/. Below you can find a summary of the different examples. They are grouped into *Tutorial*, *Simple Examples*, and *More Advanced Examples*. We suggest starting with the tutorial before moving on to the simpler examples, especially if you are new to using BurnMan.

Finally, we also include the scripts that were used for all computations and figures in the 2014 BurnMan paper in the misc/ folder, see *Reproducing Cottaar, Heister, Rose and Unterborn (2014)*.

## 4.1 Tutorial

**The tutorial for BurnMan currently consists of three separate units:**

- `step 1`,
- `step 2`, and
- `step 3`.

### 4.1.1 CIDER 2014 BurnMan Tutorial — step 1

In this first part of the tutorial we will acquaint ourselves with a basic script for calculating the elastic properties of a mantle mineralogical model.

In general, there are three portions of this script:

1) Define a set of pressures and temperatures at which we want to calculate elastic properties

2) Setup a composite of minerals (or "rock") and calculate its elastic properties at those pressures and temperatures.

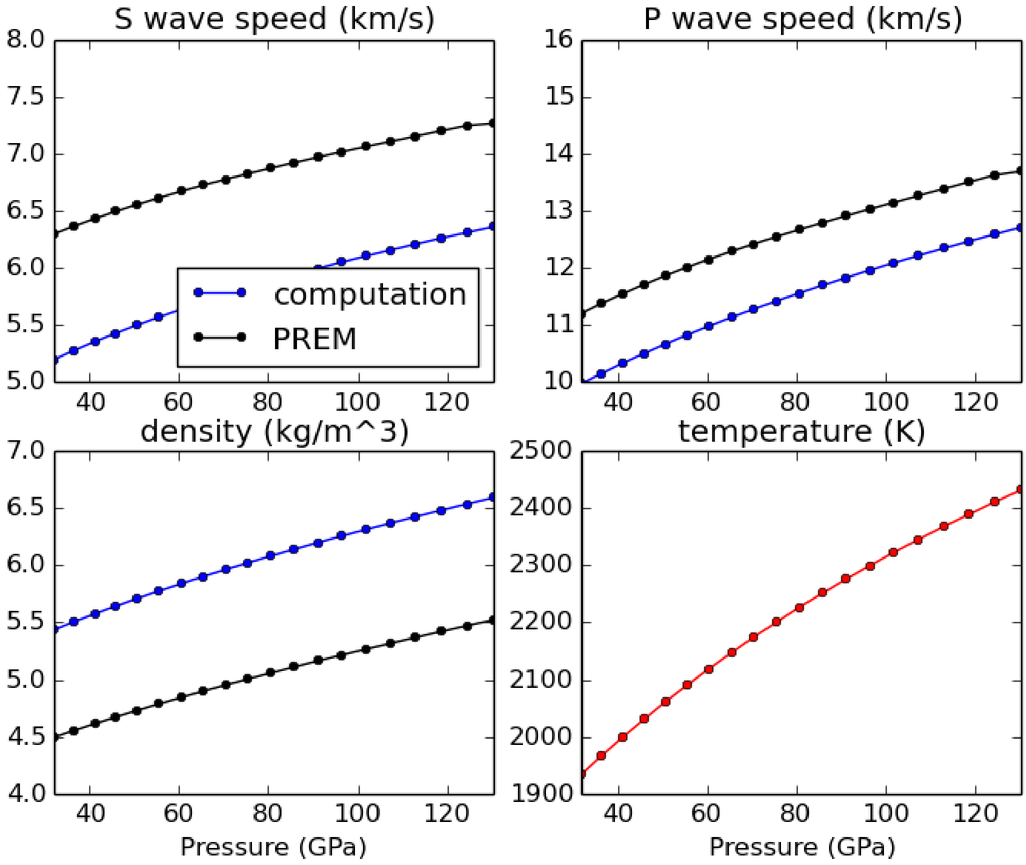
3) Plot those elastic properties, and compare them to a seismic model, in this case PREM

The script is basically already written, and should run as is by typing:

> python step_1.py

on the command line. However, the mineral model for the rock is not very realistic, and you will want to change it to one that is more in accordance with what we think the bulk composition of Earth's lower mantle is.

When run (without putting in a more realistic composition), the program produces the following image:



Your goal in this tutorial is to improve this awful fit...

### 4.1.2 CIDER 2014 BurnMan Tutorial — step 2

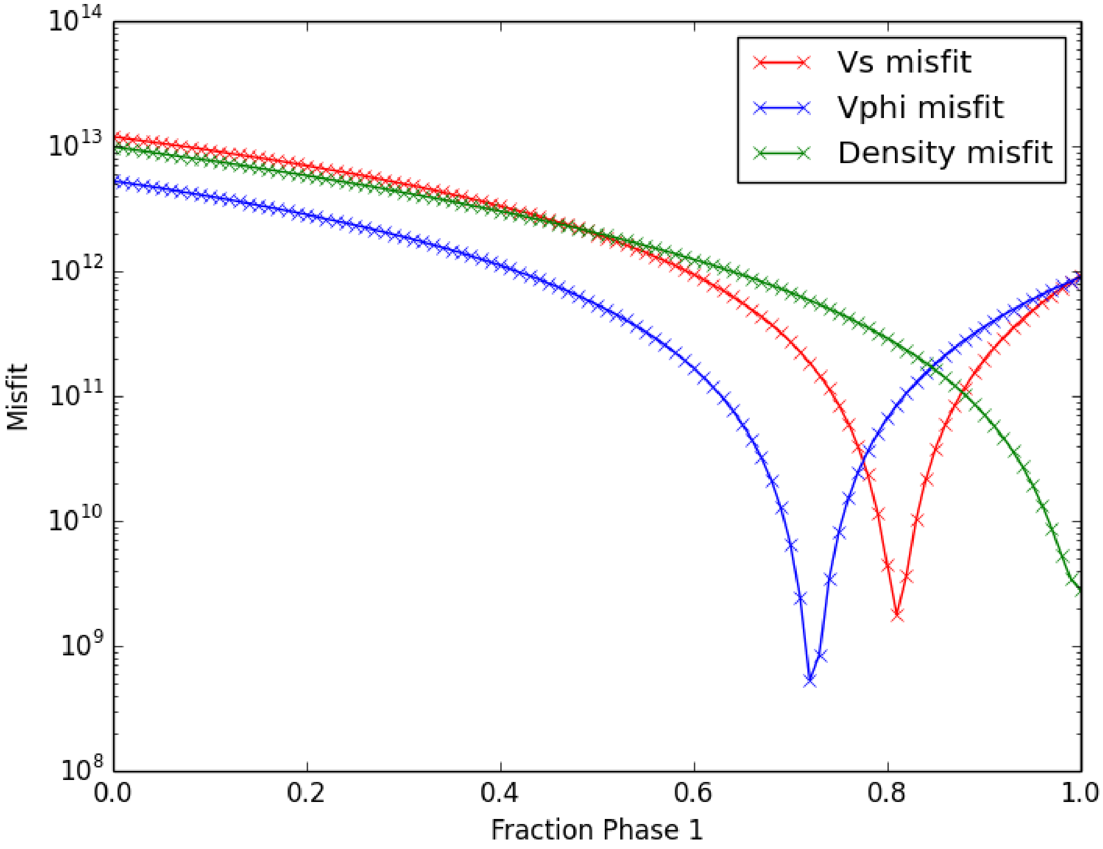
In this second part of the tutorial we try to get a closer fit to our 1D seismic reference model. In the simple Mg, Si, and O model that we used in step 1 there was one free parameter, namely phase_1_fraction, which goes between zero and one.

In this script we want to explore how good of a fit to PREM we can get by varying this fraction. We create a simple function that calculates a misfit between PREM and our mineral model as a function of phase_1_fraction, and then plot this misfit function to try to find a best model.

This script may be run by typing

> python step_2.py

Whithout changing any input, the program should produce the following image showing the misfit as a function of perovskite content:



### 4.1.3 CIDER 2014 BurnMan Tutorial — step 3

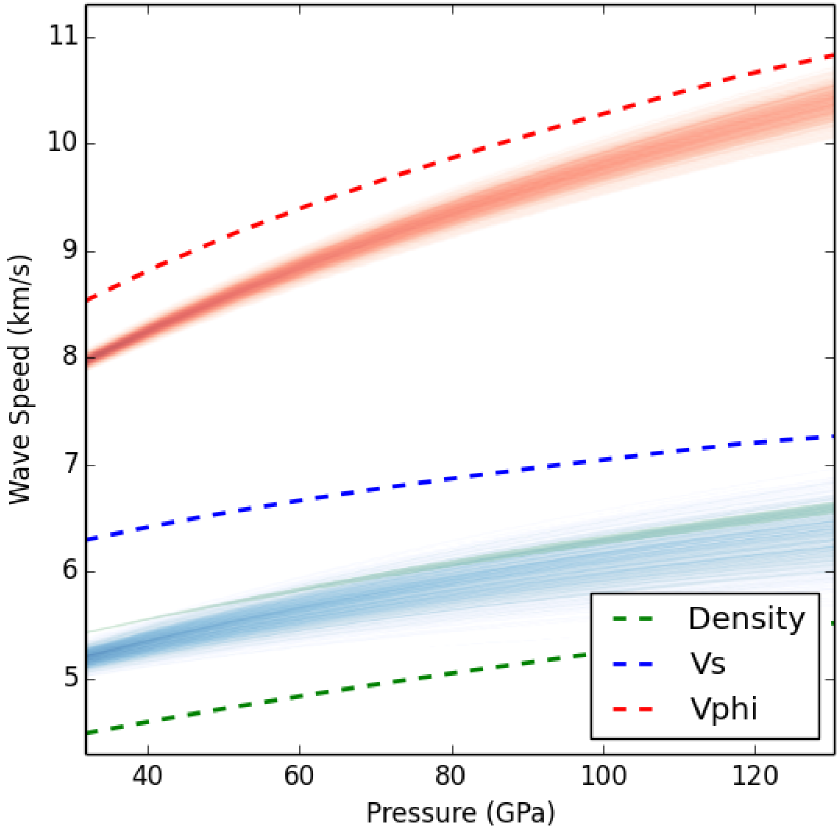
In the previous two steps of the tutorial we tried to find a very simple mineralogical model that best fit the 1D seismic model PREM. But we know that there is consideral uncertainty in many of the mineral physical parameters that control how the elastic properties of minerals change with pressure and temperature. In this step we explore how uncertainties in these parameters might affect the conclusions you draw.

The strategy here is to make many different "realizations" of the rock that you determined was the closest fit to PREM, where each realization has its mineral physical parameters perturbed by a small amount, hopefully related to the uncertainty in that parameter. In particular, we will look at how perturbations to $K_0'$ and $G_0'$ (the pressure derivatives of the bulk and shear modulus, respectively) change the calculated 1D seismic profiles.

This script may be run by typing

    python step_3.py

After changing the standard deviations for $K_0'$ and $G_0'$ to 0.2, the following figure of velocities for 1000 realizations is produced:



## 4.2 Simple Examples

**The following is a list of simple examples:**

- example_beginner,
- example_solid_solution,
- example_geotherms,
- example_seismic,
- example_composition,
- example_averaging, and
- example_chemical_potentials.

### 4.2.1 example_beginner

This example script is intended for absolute beginners to BurnMan. We cover importing BurnMan modules, creating a composite material, and calculating its seismic properties at lower mantle pressures and temperatures. Afterwards, we plot it against a 1D seismic model for visual comparison.

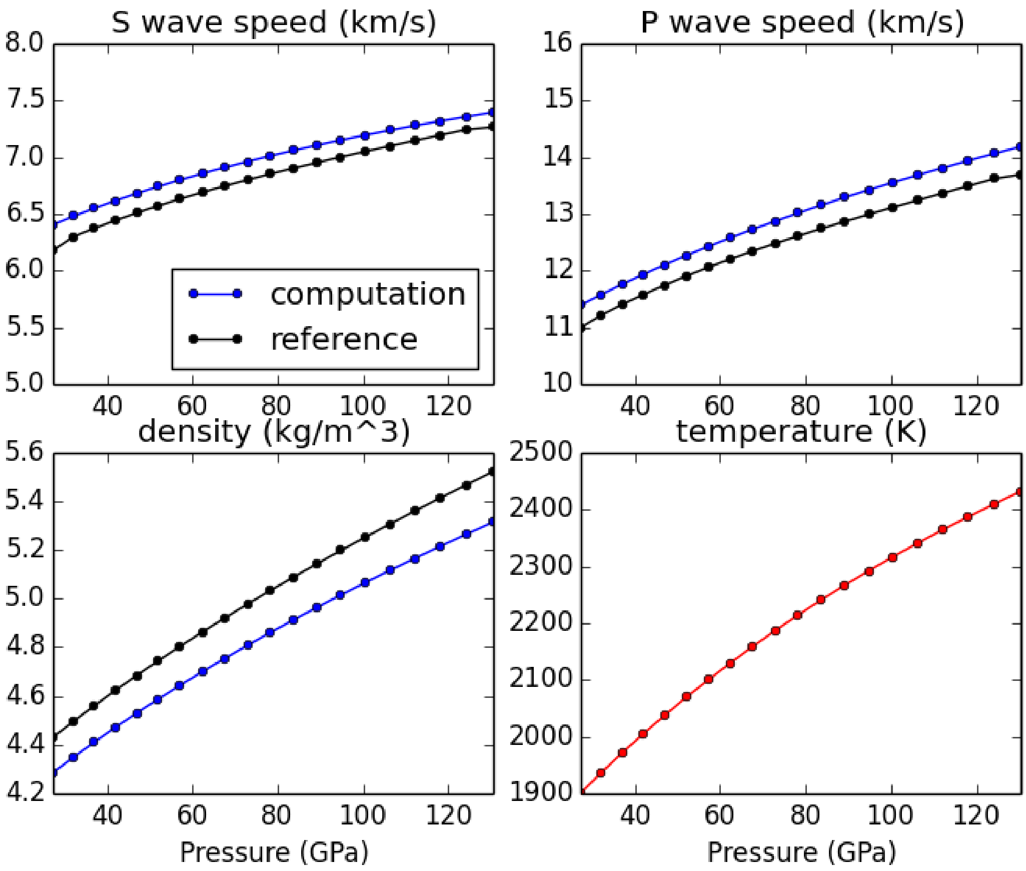
*Uses:*

- `mineral_database`
- `burnman.composite.Composite`
- `burnman.seismic.PREM`
- `burnman.geotherm.brown_shankland()`
- `burnman.material.Material.evaluate()`

*Demonstrates:*

- creating basic composites
- calculating thermoelastic properties
- seismic comparison

*Resulting figure:*



### 4.2.2 example_solid_solution

This example shows how to create different solid solution models and output thermodynamic and thermoelastic quantities.

There are four main types of solid solution currently implemented in BurnMan:

1. Ideal solid solutions
2. Symmmetric solid solutions
3. Asymmetric solid solutions
4. Subregular solid solutions

These solid solutions can potentially deal with:

- Disordered endmembers (more than one element on a crystallographic site)
- Site vacancies
- More than one valence/spin state of the same element on a site

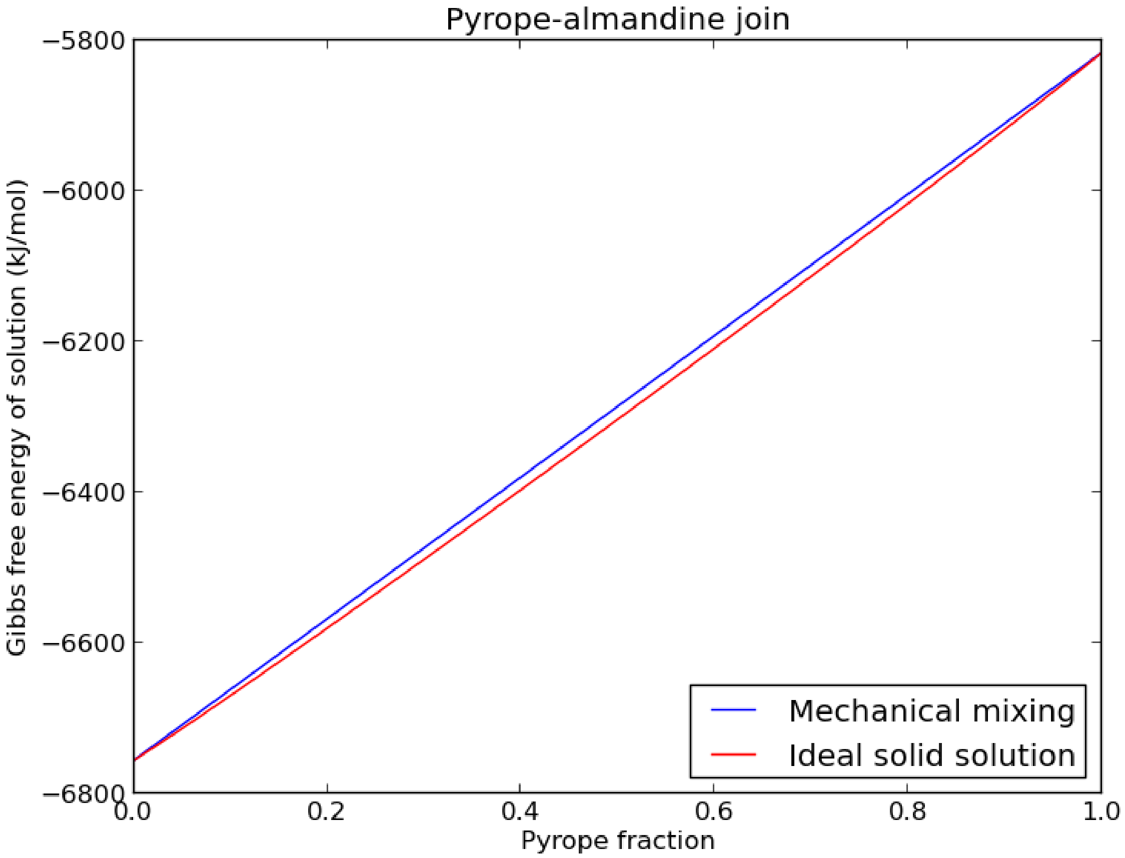
*Uses:*

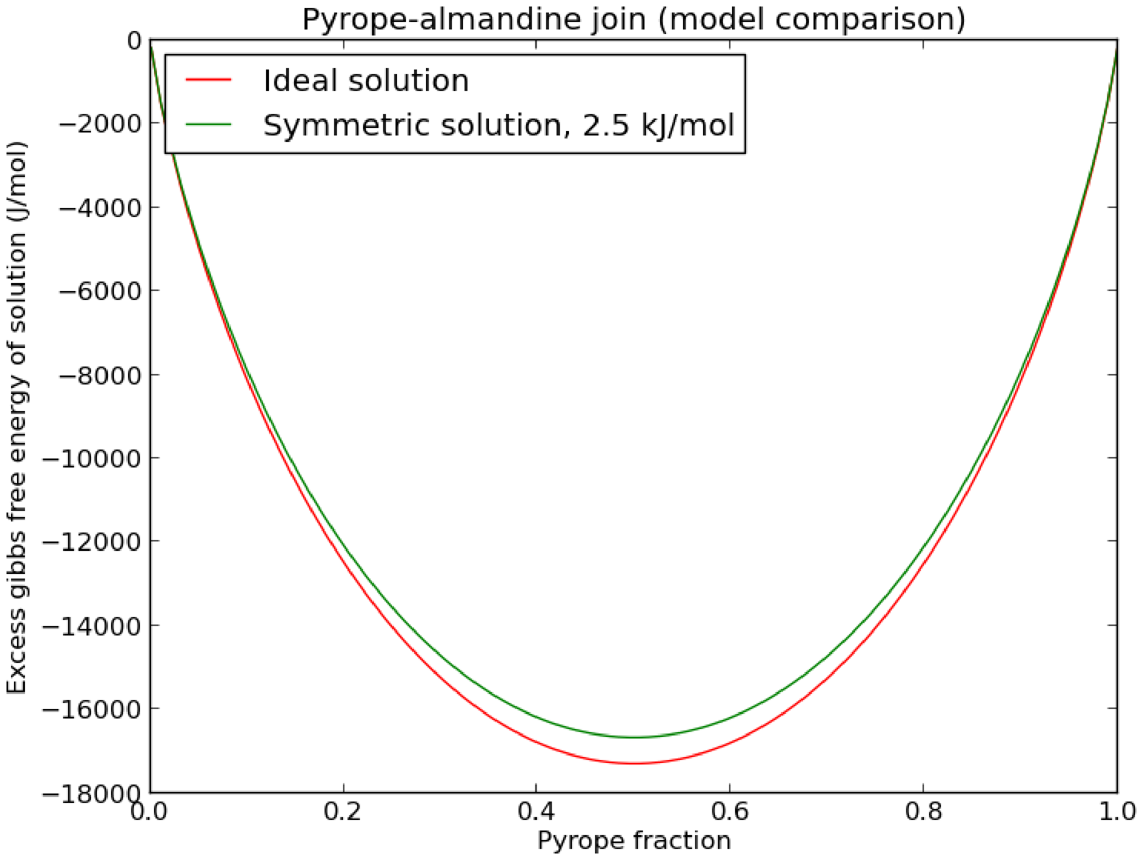
- `mineral_database`
- `burnman.solidsolution.SolidSolution`
- `burnman.solutionmodel.SolutionModel`

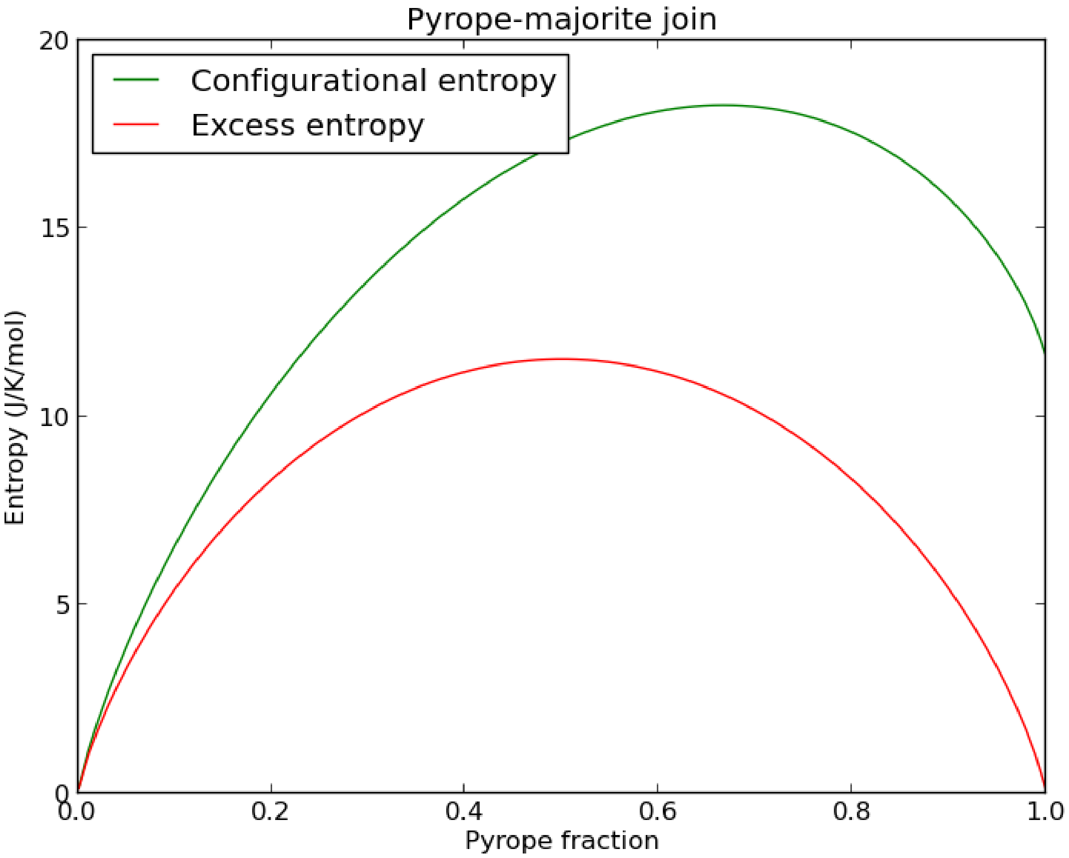
*Demonstrates:*

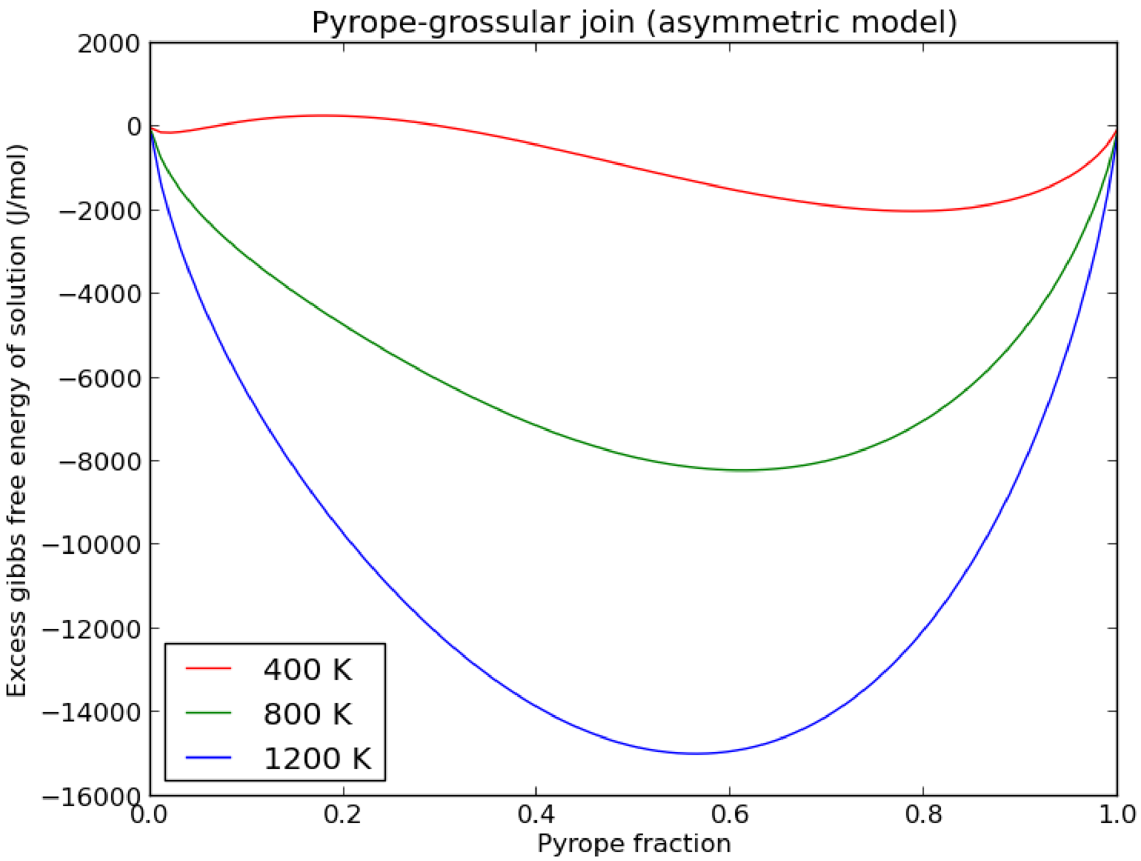
- Different ways to define a solid solution
- How to set composition and state
- How to output thermodynamic and thermoelastic properties

*Resulting figures:*


Pyrope-almandine join
Gibbs free energy of solution (kJ/mol)
−5800
−6000
−6200
−6400
−6600
−6800
0.0
0.2
0.4
0.6
0.8
1.0
Pyrope fraction
Mechanical mixing
Ideal solid solution


Pyrope-almandine join (model comparison)
Ideal solution
Symmetric solution, 2.5 kJ/mol
Excess gibbs free energy of solution (J/mol)
0
−2000
−4000
−6000
−8000
−10000
−12000
−14000
−16000
−18000
0.0
0.2
0.4
0.6
0.8
1.0
Pyrope fraction


Pyrope-majorite join
Configurational entropy
Excess entropy
Entropy (J/K/mol)
20
15
10
5
0
0.0
0.2
0.4
0.6
0.8
1.0
Pyrope fraction



### 4.2.3 example_geotherms

This example shows each of the geotherms currently possible with BurnMan. These are:

1. Brown and Shankland, 1981 [Brown1981]
2. Anderson, 1982 [anderson1982earth]
3. Watson and Baxter, 2007 [Watson2007]
4. linear extrapolation
5. Read in from file from user
6. Adiabatic from potential temperature and choice of mineral

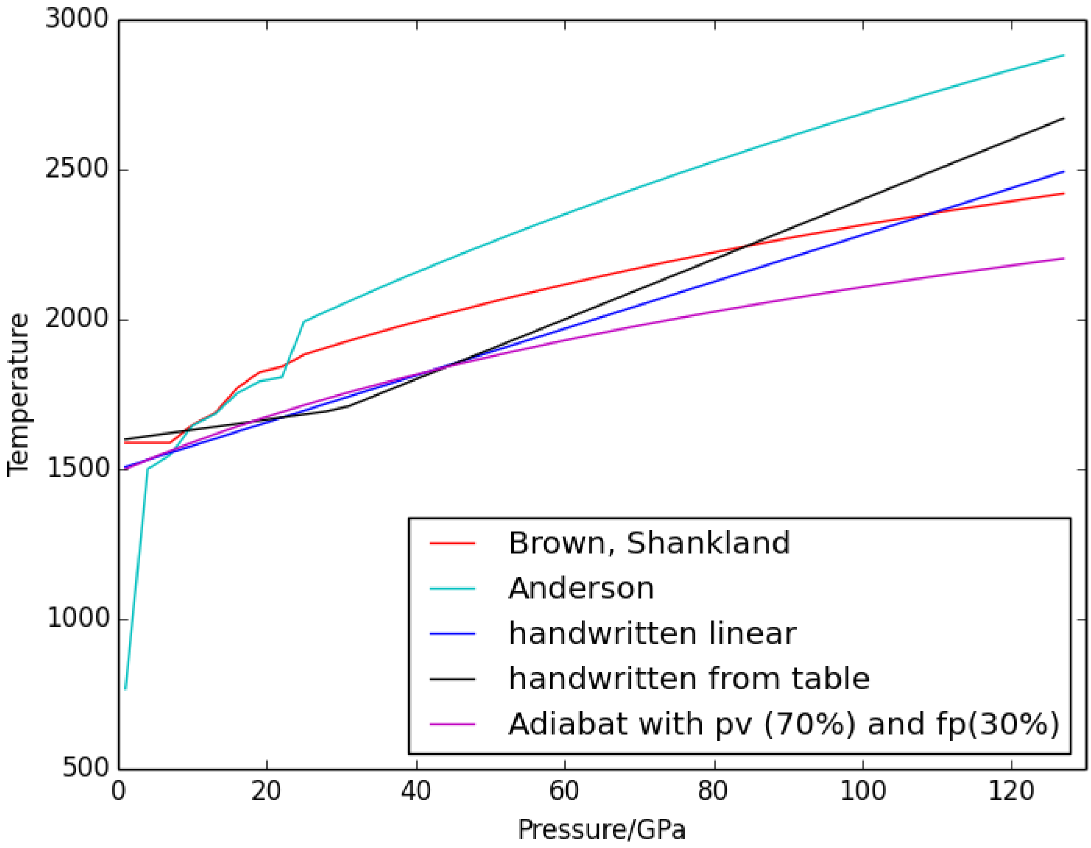
*Uses:*

- `burnman.geotherm.brown_shankland()`
- `burnman.geotherm.anderson()`
- input geotherm file *input_geotherm/example_geotherm.txt* (optional)
- `burnman.composite.Composite` for adiabat

*Demonstrates:*

- the available geotherms

*Resulting figure:*



### 4.2.4 example_seismic

Shows the various ways to input seismic models ($V_s, V_p, V_\phi, \rho$) as a function of depth (or pressure) as well as different velocity model libraries available within Burnman:

1. PREM [dziewonski1981]
2. STW105 [kustowski2008]
3. AK135 [kennett1995]
4. IASP91 [kennett1991]

This example will first calculate or read in a seismic model and plot the model along the defined pressure range. The example also illustrates how to import a seismic model of your choice, here shown by importing AK135 [kennett1995].

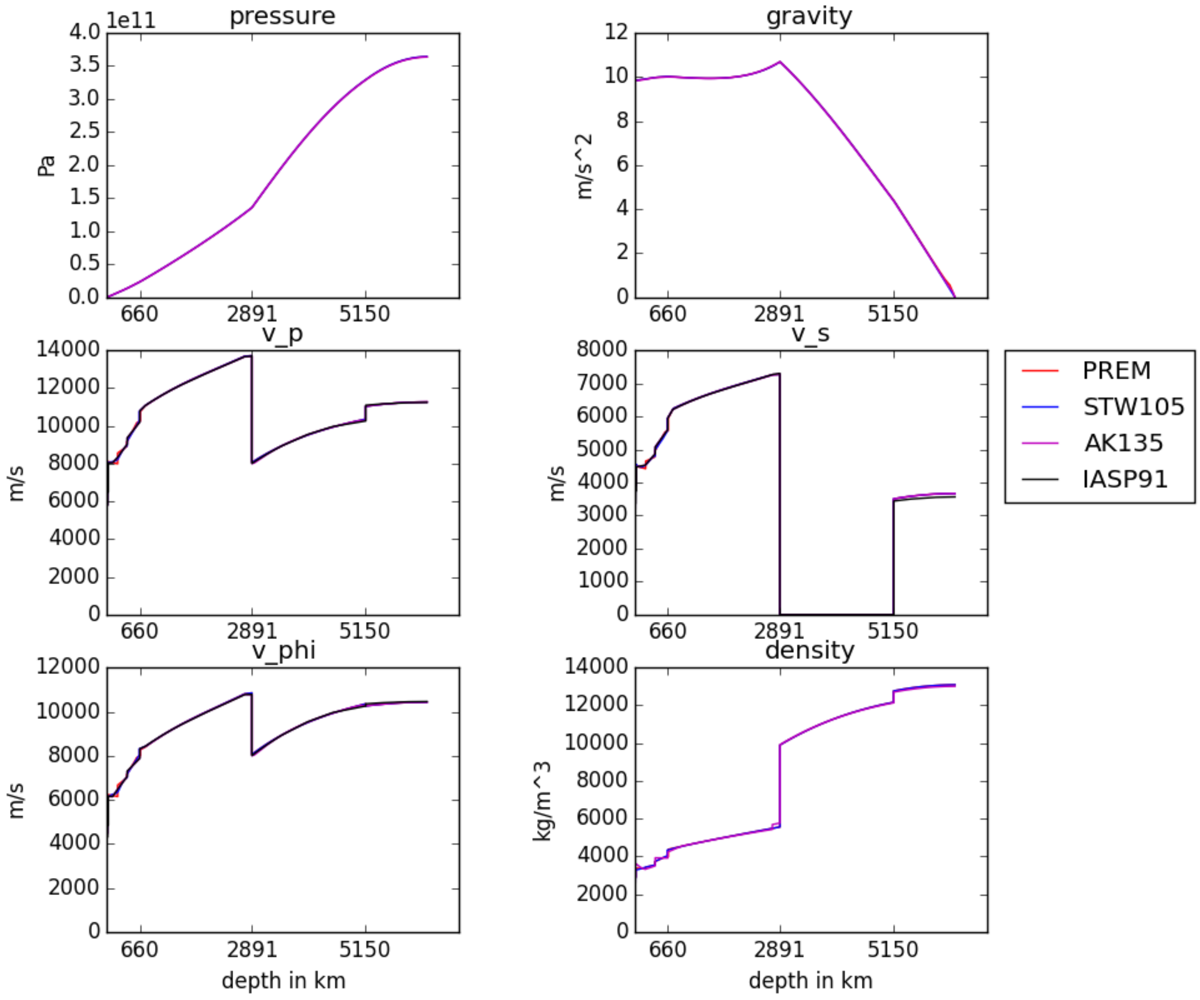
*Uses:*

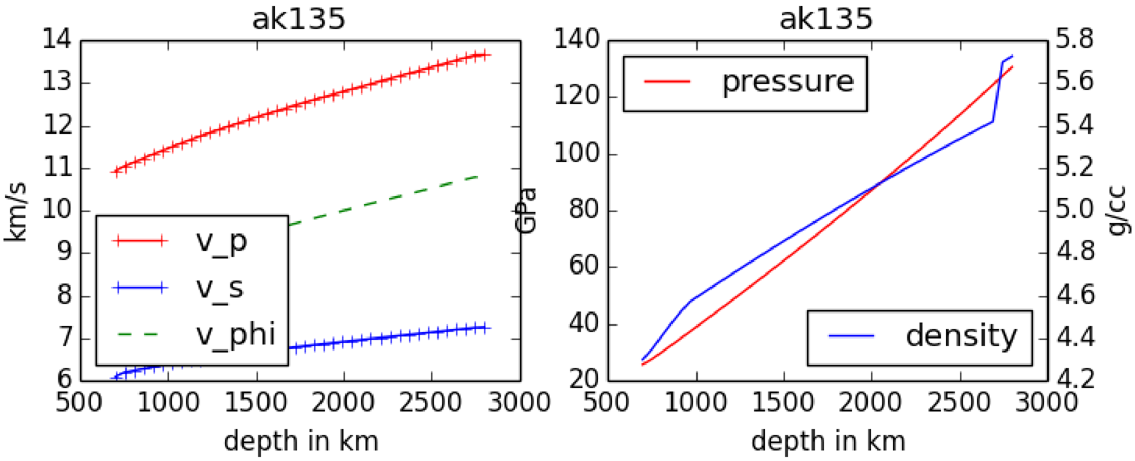
- `seismic`

*Demonstrates:*

- Utilization of library seismic models within BurnMan
- Input of user-defined seismic models

*Resulting figures:*





### 4.2.5 example_composition

This example shows how to create different minerals, how to compute seismic velocities, and how to compare them to a seismic reference model.

There are many different ways in BurnMan to combine minerals into a composition. Here we present a couple of examples:

1. Two minerals mixed in simple mole fractions. Can be chosen from the BurnMan libraries or from user defined minerals (see example_user_input_material)
2. Example with three minerals
3. Using preset solid solutions
4. Defining your own solid solution

To turn a method of mineral creation "on" the first if statement above the method must be set to True, with all others set to False.

Note: These minerals can include a spin transition in (Mg,Fe)O, see example_spintransition.py for explanation of how to implement this

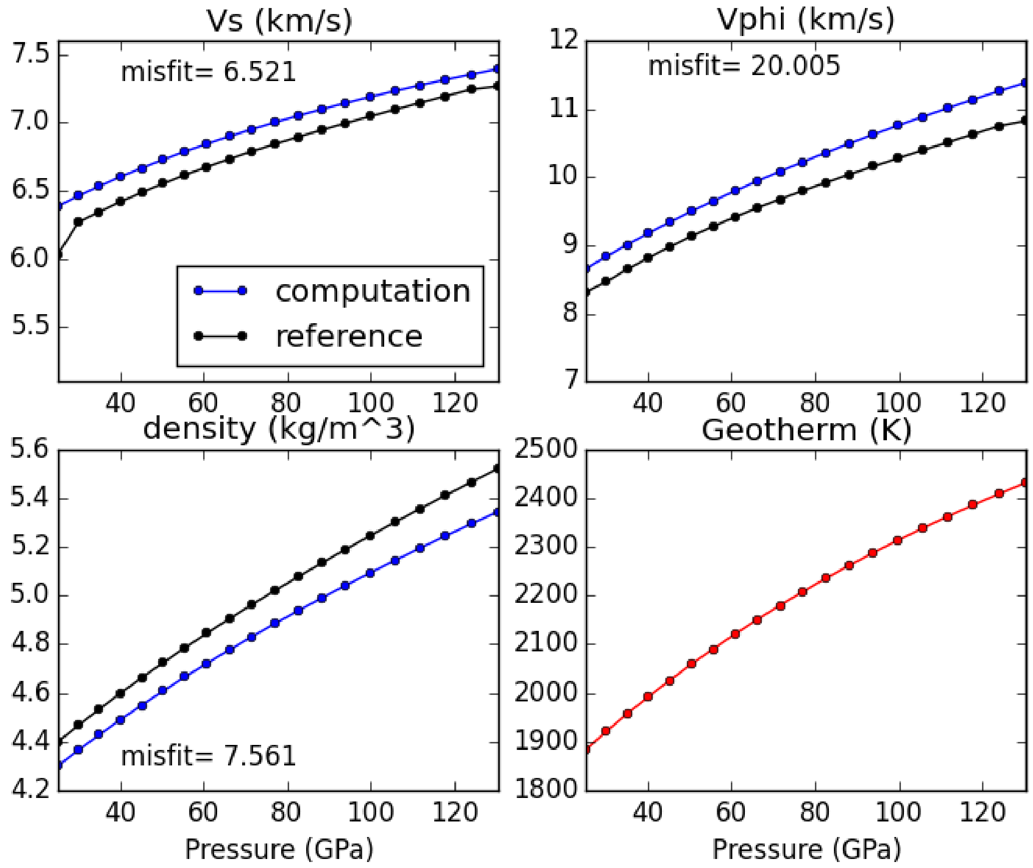
*Uses:*

- `mineral_database`
- `burnman.composite.Composite`
- `burnman.mineral.Mineral`
- `burnman.solidsolution.SolidSolution`

*Demonstrates:*

- Different ways to define a composite
- Using minerals and solid solutions
- Compare computations to seismic models

*Resulting figure:*



### 4.2.6 example_averaging

This example shows the effect of different averaging schemes. Currently four averaging schemes are available:

1. Voight-Reuss-Hill
2. Voight averaging

3. Reuss averaging
4. Hashin-Shtrikman averaging

See [Watt1976] Journal of Geophysics and Space Physics for explanations of each averaging scheme.

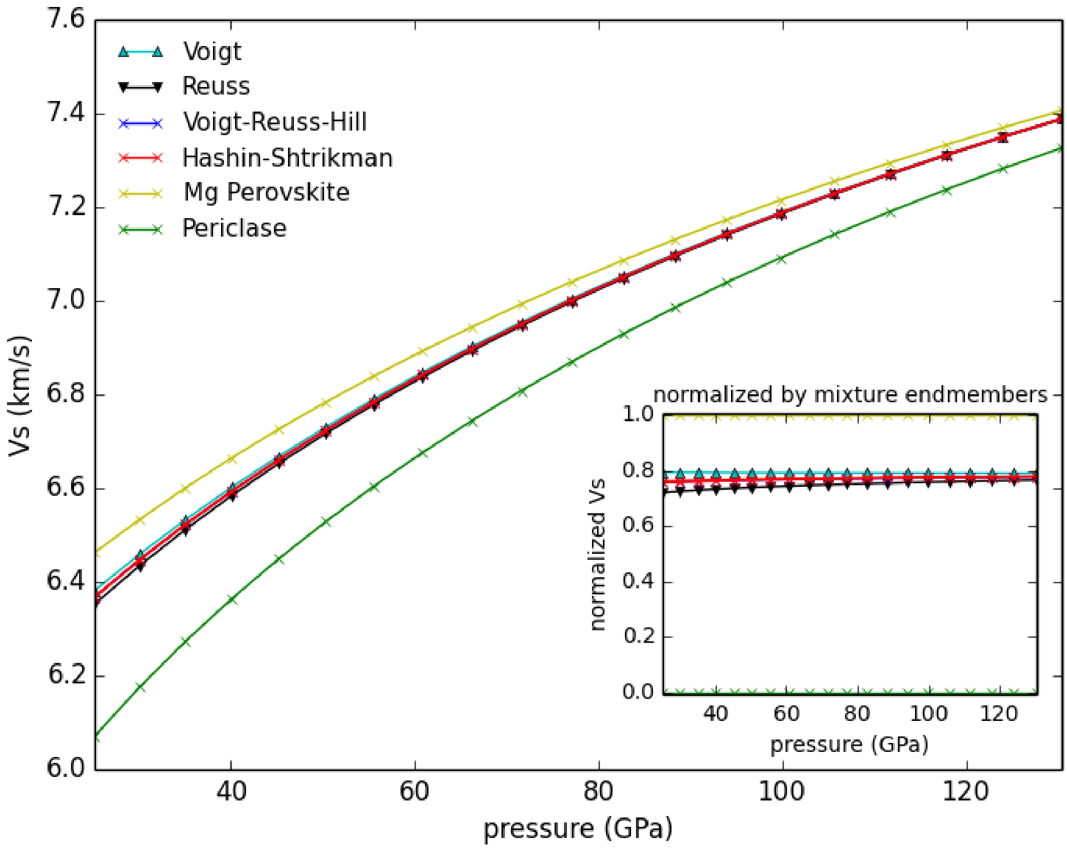
*Specifically uses:*

- `burnman.averaging_schemes.VoigtReussHill`
- `burnman.averaging_schemes.Voigt`
- `burnman.averaging_schemes.Reuss`
- `burnman.averaging_schemes.HashinShtrikmanUpper`
- `burnman.averaging_schemes.HashinShtrikmanLower`

*Demonstrates:*

- implemented averaging schemes

*Resulting figure:*



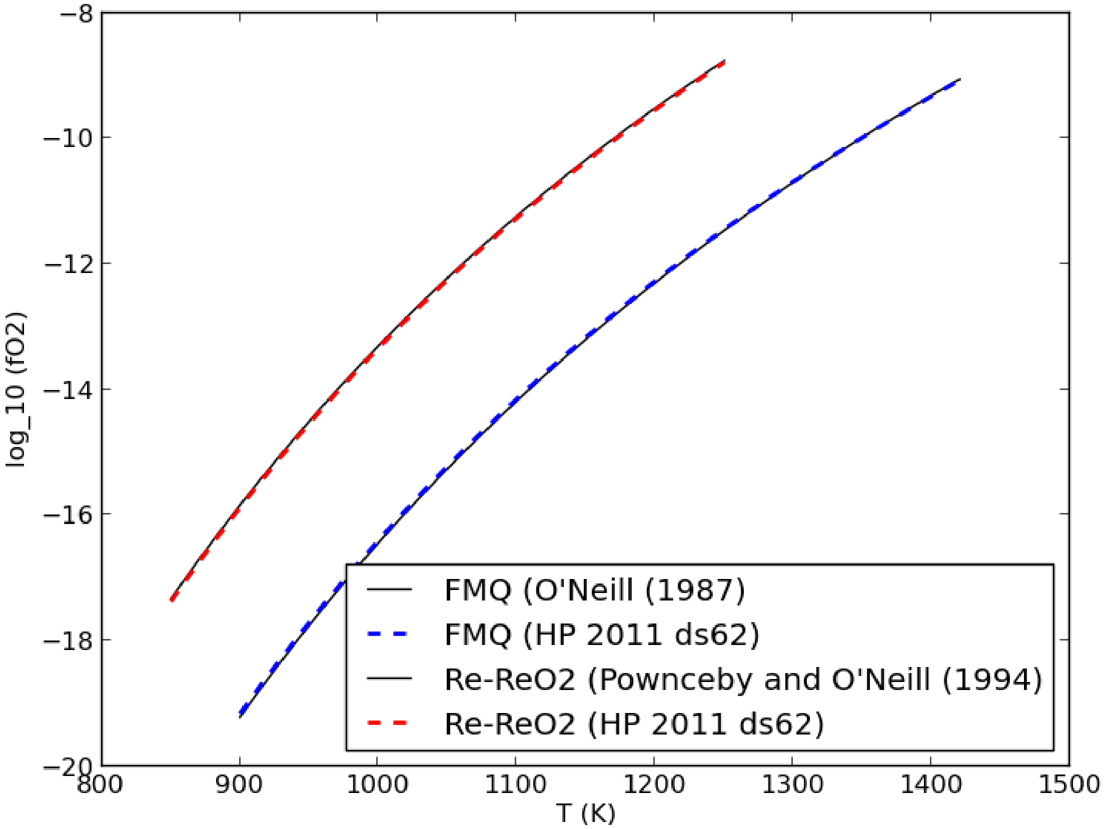
### 4.2.7 example_chemical_potentials

This example shows how to use the chemical potentials library of functions.

*Demonstrates:*

- How to calculate chemical potentials
- How to compute fugacities and relative fugacities

*Resulting figure:*



## 4.3 More Advanced Examples

**Advanced examples:**

- example_spintransition,
- example_user_input_material,
- example_optimize_pv, and
- example_compare_all_methods.

### 4.3.1 example_spintransition

This example shows the different minerals that are implemented with a spin transition. Minerals with spin transition are implemented by defining two separate minerals (one for the low and one for the high spin state). Then a third dynamic mineral is created that switches between the two previously defined minerals by comparing the current pressure to the transition pressure.

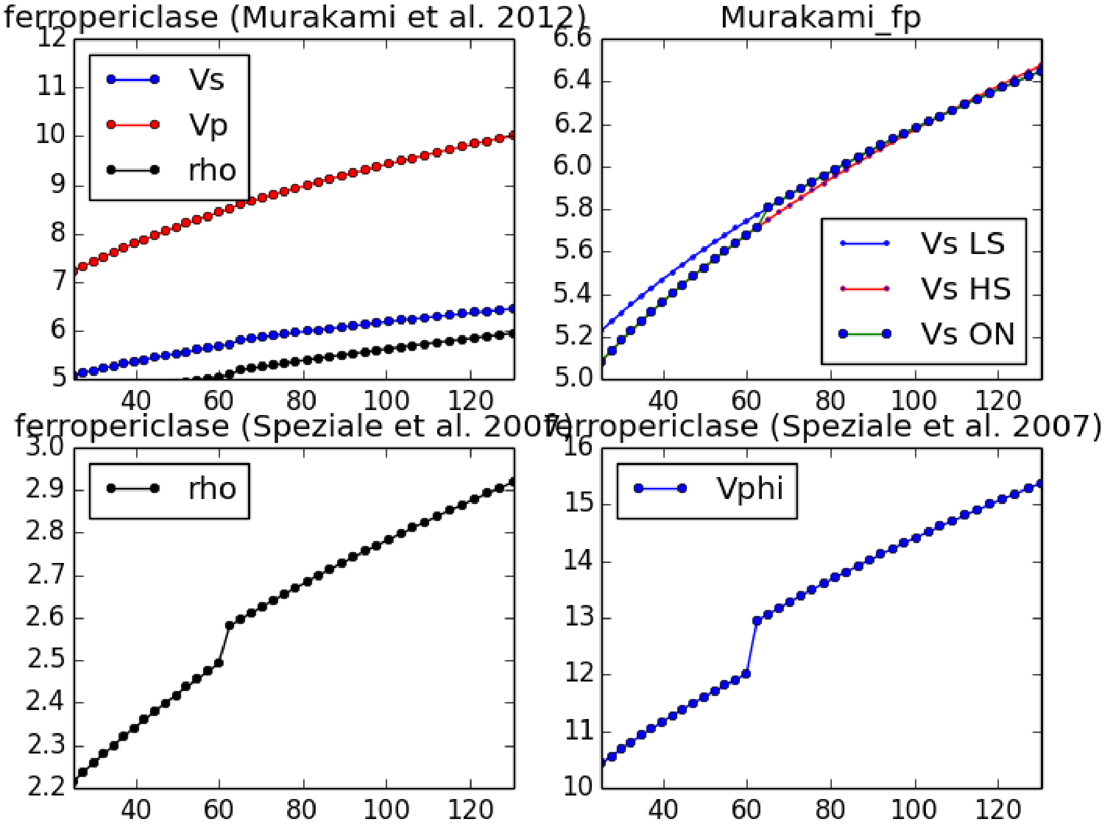
*Specifically uses:*

- `burnman.mineral_helpers.HelperSpinTransition()`
- `burnman.minerals.Murakami_etal_2012.fe_periclase()`
- `burnman.minerals.Murakami_etal_2012.fe_periclase_HS()`
- `burnman.minerals.Murakami_etal_2012.fe_periclase_LS()`

*Demonstrates:*

- implementation of spin transition in (Mg,Fe)O at user defined pressure

*Resulting figure:*



### 4.3.2 example_user_input_material

Shows user how to input a mineral of his/her choice without usint the library and which physical values need to be input for BurnMan to calculate $V_P, V_\Phi, V_S$ and density at depth.

*Specifically uses:*

- `burnman.mineral.Mineral`

*Demonstrates:*

- how to create your own minerals

### 4.3.3 example_optimize_pv

Vary the amount perovskite vs. ferropericlase and compute the error in the seismic data against PREM. For more extensive comments on this setup, see tutorial/step_2.py

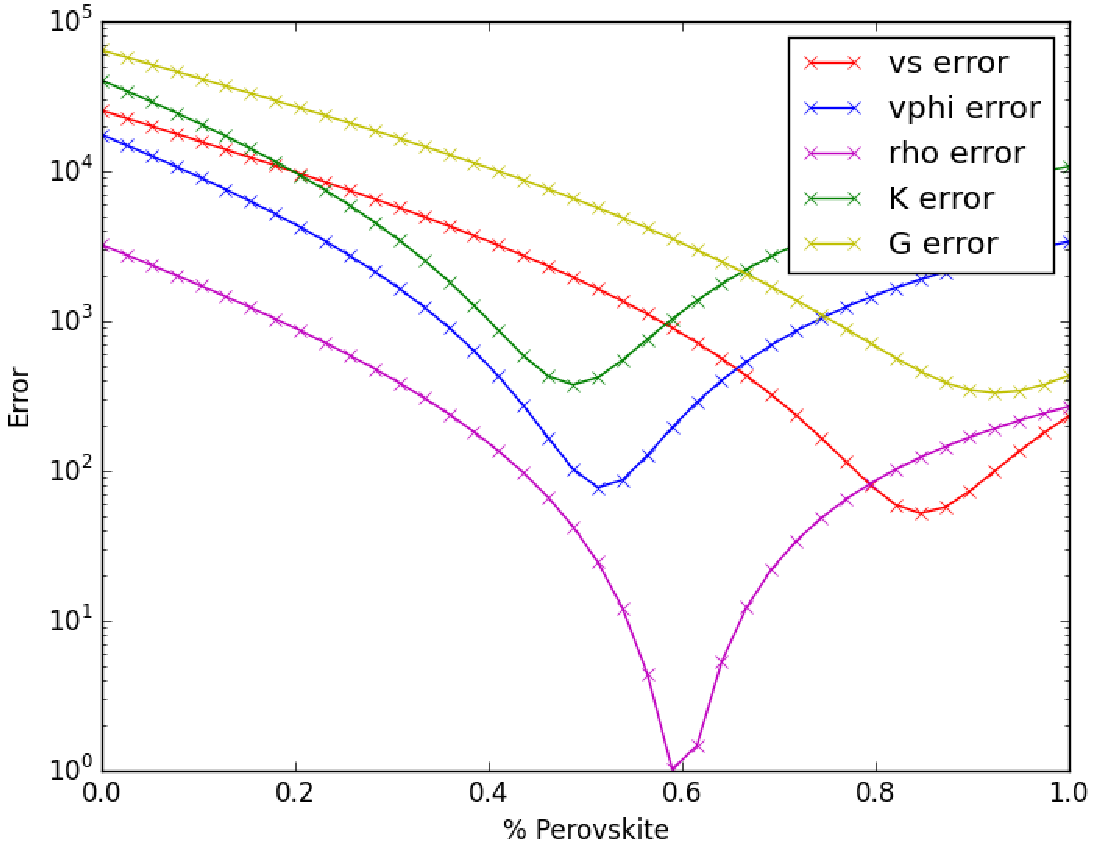
*Uses:*

- `mineral_database`
- `burnman.composite.Composite`
- `burnman.seismic.PREM`
- `burnman.geotherm.brown_shankland()`
- `burnman.material.Material.evaluate()`
- `burnman.main.compare_l2()`

*Demonstrates:*

- compare errors between models
- loops over models

*Resulting figure:*



### 4.3.4 example_build_planet

For Earth we have well-constrained one-dimensional density models. This allows us to calculate pressure as a funcion of depth. Furthermore, petrologic data and assumptions regarding the convective state of the planet allow us to estimate the temperature.

For planets other than Earth we have much less information, and in particular we know almost nothing about the pressure and temperature in the interior. Instead, we tend to have measurements of things like mass, radius, and moment-of-inertia. We would like to be able to make a model of the planet's interior that is consistent with those measurements.

However, there is a difficulty with this. In order to know the density of the planetary material, we need to know the pressure and temperature. In order to know the pressure, we need to know the gravity profile. And in order to the the gravity profile, we need to know the density. This is a nonlinear problem which requires us to iterate to find a self-consistent solution.

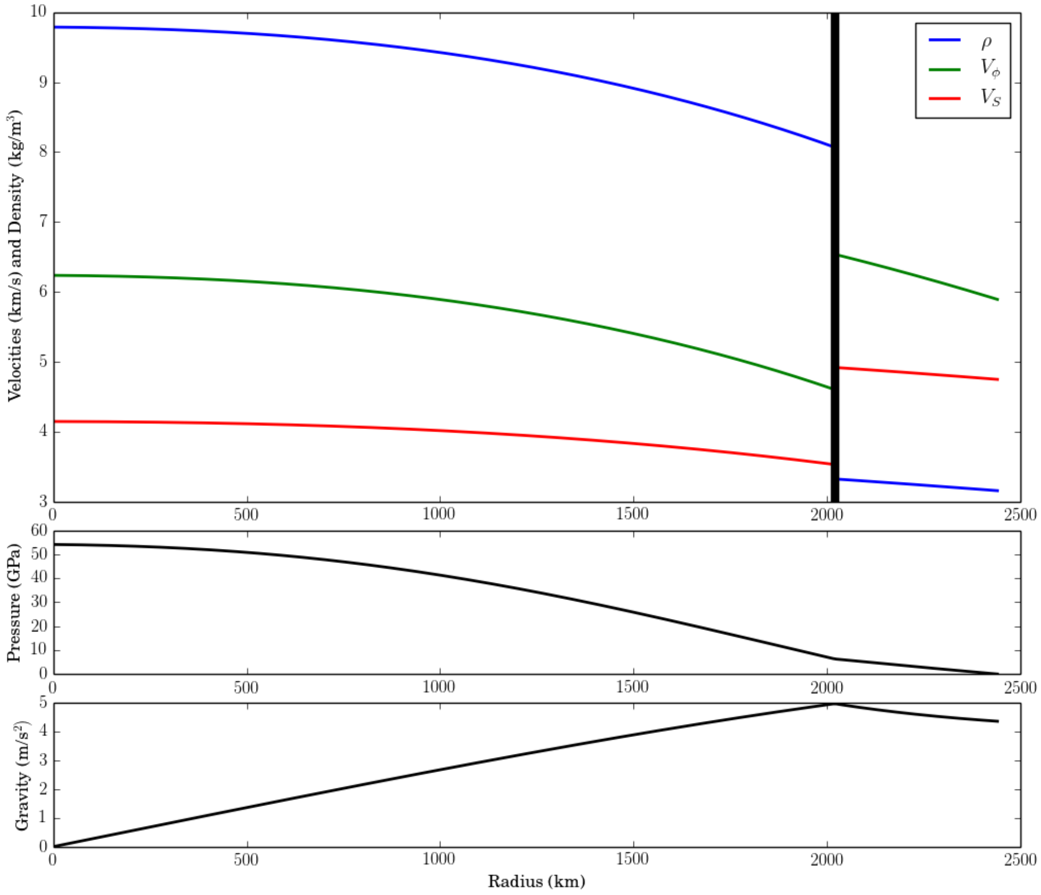
Here we show an example that does this, using the planet Mercury as motivation.

*Uses:*

- `mineral_database`
- `burnman.composite.Composite`

- `burnman.material.Material.evaluate()`

*Resulting figure:*



### 4.3.5 example_compare_all_methods

This example demonstrates how to call each of the individual calculation methodologies that exist within BurnMan. See below for current options. This example calculates seismic velocity profiles for the same set of minerals and a plot of $V_s$, $V_\phi$ and $\rho$ is produce for the user to compare each of the different methods.

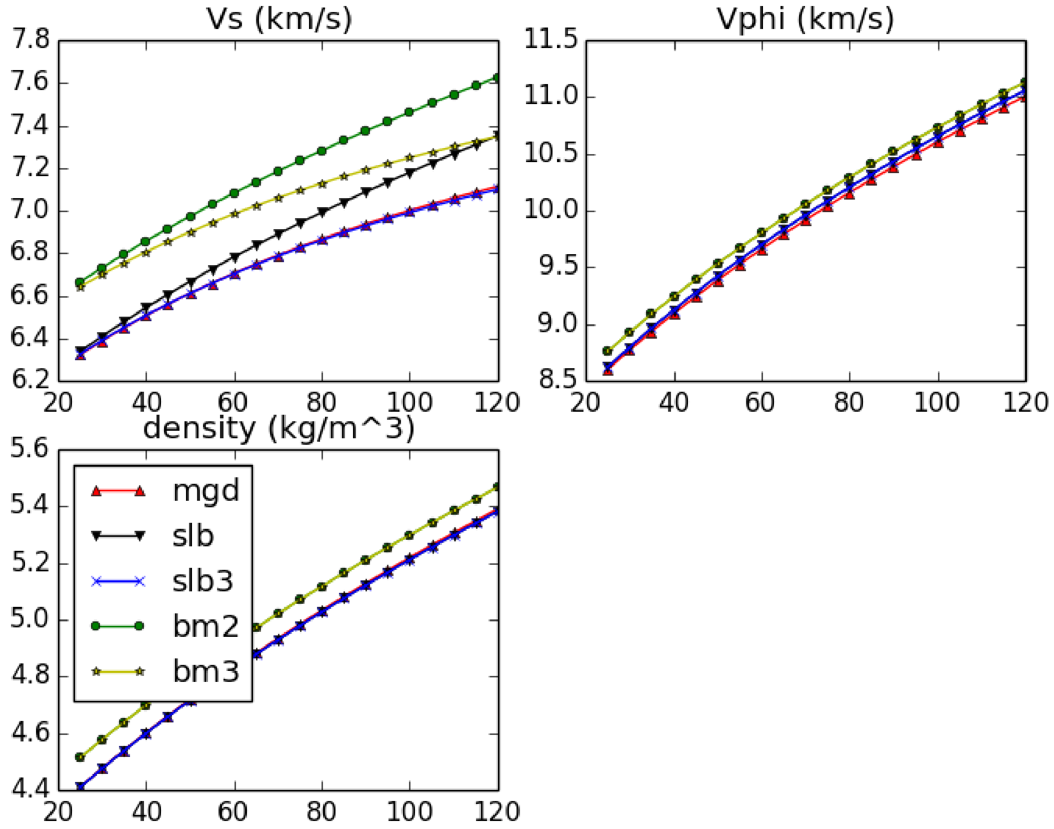
*Specifically uses:*

- `eos`

*Demonstrates:*

- Each method for calculating velocity profiles currently included within BurnMan

*Resulting figure:*



## 4.4 Reproducing Cottaar, Heister, Rose and Unterborn (2014)

In this section we include the scripts that were used for all computations and figures in the 2014 BurnMan paper: Cottaar, Heister, Rose & Unterborn (2014) [Cottaar2014]

### 4.4.1 paper_averaging

This script reproduces [Cottaar2014], Figure 2.

This example shows the effect of different averaging schemes. Currently four averaging schemes are available: 1. Voight-Reuss-Hill 2. Voight averaging 3. Reuss averaging 4. Hashin-Shtrikman averaging

See [Watt1976] for explanations of each averaging scheme.

requires: - geotherms - compute seismic velocities

teaches: - averaging

### 4.4.2 paper_benchmark

This script reproduces the benchmark in [Cottaar2014], Figure 3.

### 4.4.3 paper_fit_data

This script reproduces [Cottaar2014] Figure 4.

This example demonstrates BurnMan's functionality to fit thermoelastic data to both 2nd and 3rd orders using the EoS of the user's choice at 300 K. User's must create a file with $P, T$ and $V_s$. See input_minphys/ for example input files.

requires: - compute seismic velocities

teaches: - averaging

`contrib.CHRU2014.paper_fit_data.`**`calc_shear_velocities`**(*G_0*, *Gprime_0*, *mineral*, *pressures*)

`contrib.CHRU2014.paper_fit_data.`**`error`**(*guess*, *test_mineral*, *pressures*, *obs_vs*)

### 4.4.4 paper_incorrect_averaging

This script reproduces [Cottaar2014], Figure 5. Attempt to reproduce Figure 6.12 from [Murakami2013]

### 4.4.5 paper_opt_pv

This script reproduces [Cottaar2014], Figure 6. Vary the amount perovskite vs. ferropericlase and compute the error in the seismic data against PREM.

requires: - creating minerals - compute seismic velocities - geotherms - seismic models - seismic comparison

teaches: - compare errors between models - loops over models

### 4.4.6 paper_onefit

This script reproduces [Cottaar2014], Figure 7. It shows an example for a best fit for a pyrolitic model within mineralogical error bars.

### 4.4.7 paper_uncertain

This script reproduces [Cottaar2014], Figure 8. It shows the sensitivity of the velocities to various mineralogical parameters.

## 4.5 Misc or work in progress

### 4.5.1 example_compare_enstpyro

This example shows you how to create two materials from wt% determines the optimum mixing between the two to match the seismic model of your choice. Currently it compares two end member meteorite groups among the chondrites: carbonaceous and enstatite. Velocities are calculated for each set of minerals and plotted for comparison.

requires: - geotherms - seismic models - compute seismic velocities - creating minerals

teaches: - weight percent materials

### 4.5.2 example_parition_coef

This example shows how to vary the distribution coefficient of the perovskite/ferropericlase system. The user sets $K_{d0}$ and BurnMan scales $K_d$ as a function of $P$ and $T$ adopting the formalism of [Nakajima2012]. Specifically we adopt equation 5 of [Nakajima2012] with $\Delta V_0$ = 0.2 cc/mol, and calculating the partition coefficient of Fe in each phase from stoichiometry.

This example will calculate mineral input parameters from Mg and Fe endmembers from Stixrude and Lithgow-bertelloni, 2005 with weighting determined by the calculated partition coefficients. Finally, output plots of $X_{Fe}$ in pv and $X_{Fe}$ in fp our output as well as the user's choice of geotherm

requires: - geotherms -input distribution coefficient $K_{d0}$

teaches: - creating varying proportions of Fe and its effect on seismic velocities

### 4.5.3 example_fit_data

This example demonstrates BurnMan's functionality to fit thermoelastic data to both 2nd and 3rd orders using the EoS of the user's choice at 300 K. User's must create a file with $P, T$ and $V_s$. See input_minphys/ for example input files.

requires: - compute seismic velocities

teaches: - averaging

### 4.5.4 example_grid

This example shows how to evaluate seismic quantities on a $P, T$ grid.

### 4.5.5 example_woutput

This example explains how to perform the basic i/o of BurnMan. A method of calculation is chosen, a composite mineral/material (see example_composition.py for explanation of this process) is created in the class "rock," finally a geotherm is created and seismic velocities calculated.

Post-calculation, the results are written to a simple text file to plot/manipulate at the user's whim.

requires: - creating minerals - compute seismic velocities - geotherms

teaches: - output computed seismic data to file

# INDEX