EOS*fit*

Bruno A. Calfa

October 24, 2012

Contents

1 Introduction

The EOS*fit* package, written in Python, contains a number of Equations of State (EOS) models to fit the first-principles calculated static energy, E, (without zero-point vibrational energy) *versus* volume, V, points [Shang et al., 2010]. In addition to estimating the parameters of the EOS models, it computes the *confidence interval* of each parameter using the Student's t-distribution.

The EOS models supported are as follows:

- ***5-parameter Birch-Murnaghan (BM5)***

$$E(V) = a + bV^{-2/3} + cV^{-4/3} + dV^{-2} + eV^{-8/3}$$

where a, b, c, d, and e are fitting parameters.

- ***Modified 5-parameter Birch-Murnaghan (mBM5)***

$$E(V) = a + bV^{-1/3} + cV^{-2/3} + dV^{-1} + eV^{-4/3}$$

where a, b, c, d, and e are fitting parameters.

- ***4-parameter Birch-Murnaghan (BM4)***

$$E(V) = a + bV^{-2/3} + cV^{-4/3} + dV^{-2}$$

where a, b, c, and d are fitting parameters.

- ***Modified 4-parameter Birch-Murnaghan (mBM4)***

$$E(V) = a + bV^{-1/3} + cV^{-2/3} + dV^{-1}$$

where a, b, c, and d are fitting parameters.

- ***5-parameter Logarithmic (LOG5)***

$$E(V) = a + b\ln V + c(\ln V)^2 + d(\ln V)^3 + e(\ln V)^4$$

where a, b, c, d, and e are fitting parameters.

- ***4-parameter Logarithmic (LOG4)***

$$E(V) = a + b\ln V + c(\ln V)^2 + d(\ln V)^3$$

where a, b, c, and d are fitting parameters.

- ***4-parameter Murnaghan (MU4)***

$$E(V) = a + \frac{B_0 V}{B_0'}\left(1 + \frac{(V_0/V)^{B_0'}}{B_0' - 1}\right)$$

where $a = E_0 - \frac{B_0 V_0}{B_0' - 1}$. The fitting parameters and their meaning are: E_0 (equilibrium energy), B_0 (equilibrium bulk modulus), B_0' (first derivative of B_0 with respect to pressure), and V_0 (equilibrium volume).

- ***4-parameter Vinet (VI4)***

$$E(V) = a - \frac{4B_0V_0}{(B_0'-1)^2}\left\{1 - \frac{3}{2}(B_0'-1)\left[1 - \left(\frac{V_0}{V}\right)^{1/3}\right]\right\} \times \exp\left\{\frac{3}{2}(B_0'-1)\left[1 - \left(\frac{V_0}{V}\right)^{1/3}\right]\right\}$$

where $a = E_0 + \frac{4B_0V_0}{(B_0'-1)^2}$. The fitting parameters and their meaning are same as in MU4.

- ***4-parameter Morse (MO4)***

$$E(V) = a + b\exp(dV^{1/3}) + c\exp(2dV^{1/3})$$

where a, b, c, and d are fitting parameters.

The models BM5, mBM5, BM4, mBM4, LOG5, and LOG5 are linear (in the fitting parameters), whereas the models MU4, VI4, and MO4 are nonlinear.

2 Methodology

In order to estimate the parameters of the linear models, the Moore-Penrose Pseudoinverse is calculated and the solution is obtained in a least-squares sense. The procedure is as follows. Each linear EOS model can be written in the form $y = Xp$, where y is the vector of energy values, X is the regressor matrix whose columns contain each term in V, and p is the vector of parameters to be estimated. Therefore, the parameters are calculated by:

$$\begin{aligned} Xp &= y \\ X^T X p &= X^T y \\ (X^T X)^{-1} X^T X p &= (X^T X)^{-1} X^T y \\ p &= X^+ y \end{aligned}$$

where $X^+ = (X^T X)^{-1} X^T$ is the pseudoinverse of X. Remark: in the actual implementation, no matrix is directly inverted. The computationally efficient way to calculate the optimal p is to solve the linear system $X^T X p = X^T y$ for p using `numpy.linalg.solve`.

The parameters of the nonlinear models are estimated using the function `curve_fit` available in the SciPy package. The function `curve_fit` not only computes the optimal parameters, given a ***good*** (this cannot be emphasized more!) initial guess, but also the estimated covariance matrix evaluated at the solution. The covariance matrix can be used to estimate the confidence intervals for each parameter as described in [Bates and Watts, 1988].

The basic procedure to obtain the confidence intervals for the linear models is given below:

1. Compute optimal parameters: $p^* = X^+ y$
2. Calculate residuals: $r = y - Xp^*$
3. Calculate degrees of freedom: $dof = \text{length}(y) - \text{length}(p^*)$
4. Obtain estimated residual variance: $s^2 = \frac{||r||_2^2}{dof}$, where $|| \cdot ||_2$ is the 2-norm
5. Estimate covariance matrix (Jacobian approximation): $cov = s^2 (X^T X)$
6. Obtain confidence intervals on parameters: $ci = p^* \pm tinv(1-\alpha, dof)\sqrt{\text{diag}(cov)}$, where α is the percentage of confidence desired, $tinv(\cdot, \cdot)$ is the Student's t inverse survival function, and $\text{diag}(\cdot)$ retrieves the elements in the main diagonal of its argument.

For nonlinear models, the covariance matrix is already estimated by the function `curve_fit` and only the last step above has to be performed to get the confidence intervals.

3 Using the EOS*fit* Package

The EOS*fit* package is composed of a Python module called `eosfit`, which contains the class `EOS`. The user instantiates an object from the class `EOS` by passing the volumetric and energy data arrays. An optional argument is the EOS model (default: MU4). The selection of the model is done through class `EOSmodel`, which serves as an "`enum`" type (commonly found in programming languages such as C and others).

The following listing shows an example of fitting the same data to BM5 and MU4 models. The package contains a `test.py` file with the code below.

```
from eosfit import EOS, EOSmodel
import numpy as np
import matplotlib.pyplot as plt

V = np.array([8., 8.5, 9., 9.6, 10.2, 10.9, 11.6, 12.2, 13., 13.8,
    14.5])
E = np.array([-4.65, -5.05, -5.3, -5.48, -5.57, -5.59, -5.575, -5.5,
    -5.4, -5.3, -5.18])

# MU4 model
eos = EOS(V, E)
p0 = [-6., 2., 5, 10.]
pMU4 = eos.fit(p0) # Initial guess required (nonlinear model)
ciMU4 = eos.get_ci()
EMU4 = eos.MU4(V, pMU4[0], pMU4[1], pMU4[2], pMU4[3])
print pMU4, ciMU4, EMU4

# BM5 model
eos.set_model(EOSmodel.BM5)
pBM5 = eos.fit() # No initial guess required (linear model)
ciBM5 = eos.get_ci()
EBM5 = eos.BM5(V, pBM5[0], pBM5[1], pBM5[2], pBM5[3], pBM5[4])
print pBM5, ciBM5, EBM5

# Plot results
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.plot(V,E,marker='o',markersize=10,color='r',linestyle='')
ax.plot(V,EMU4,color='b')
ax.plot(V,EBM5,color='k')
plt.xlabel('Volume [$\r{A}^3$]')
plt.ylabel('Energy [eV/atom]')
plt.legend(['Data', 'MU4', 'BM5'], loc='best')
plt.savefig('eosfit_example.png')
plt.title('EOSfit Example')
plt.show()
```

The resulting graph is shown in Figure 1. The parameters estimated for both models fit them well to the data; however, their confidence intervals are significantly different (not shown).

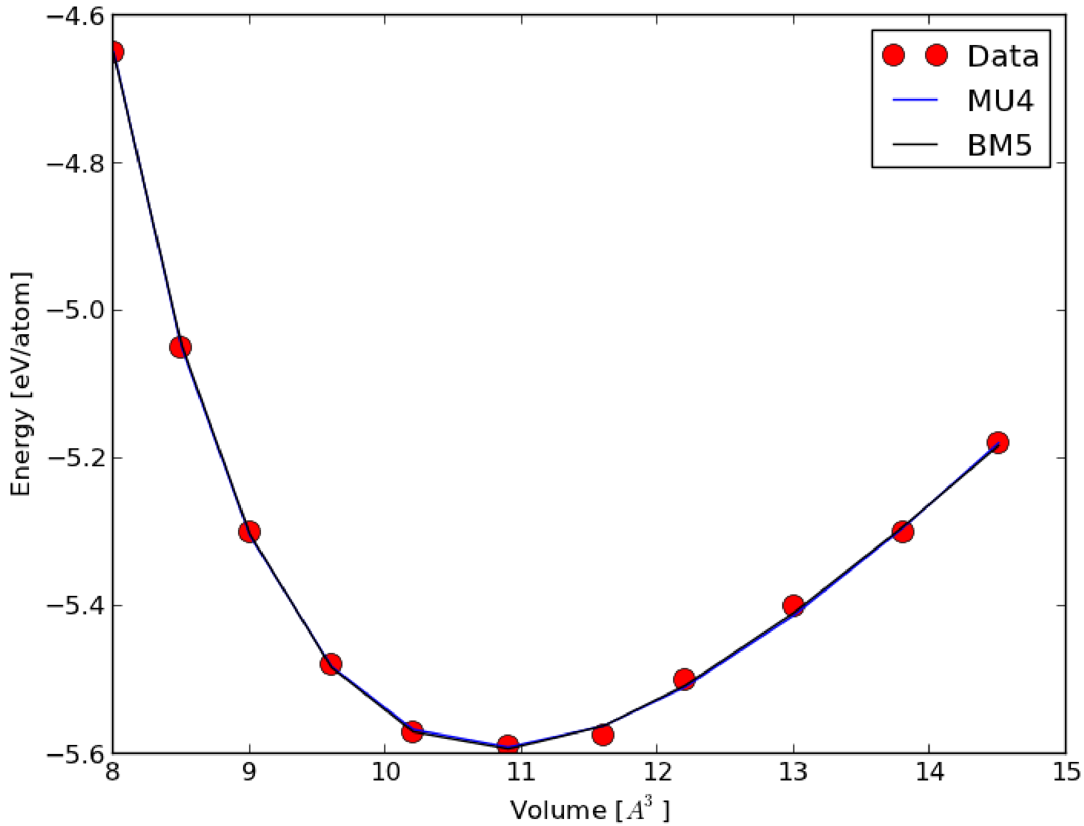

Figure 1: Fitting parameters of MU4 and BM5 models to experimental data.

To illustrate the relevance of computing confidence intervals, the models BM4 and VI4 were fit to the same data as above and the results are given in Table 1. Note that the confidence intervals for VI4 are much *tighter* than for BM4.

Table 1: Optimal parameters and their confidence intervals for models BM4 and VI4 (95% confidence).

	BM4		VI4	
Parameter	p^*	ci	p^*	ci
p_0	-6.0280	$[-9.0572,\ -2.9987]$	-5.5959	$[-5.6031,\ -5.5888]$
p_1	83.2840	$[39.0626,\ 127.5053]$	1.3132	$[1.2808,\ 1.3456]$
p_2	-782.4178	$[-995.8613,\ -568.9743]$	6.1059	$[5.7164,\ 6.4953]$
p_3	1885.1266	$[1544.446,\ 2225.8071]$	10.7855	$[10.7400,\ 10.8310]$

where $p = (a,\ b,\ c,\ d)$ for BM4 and $p = (E_0,\ B_0,\ B_0',\ V_0)$ for VI4.

References

D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons, 1988.

S-L. Shang, Y. Wang, D. Kim, and Z-K Liu. First-principles Thermodynamics from Phonon and Debye Model: Application to Ni and Ni_3Al. *Computational Materials Science.*, 47(4): 1040–1048, 2010.