MechElastic: A Python library for analysis of mechanical and elastic properties of bulk and 2D materials ☆,☆☆

Sobhit Singh [a,b,*], Logan Lang [a], Viviana Dovale-Farelo [a], Uthpala Herath [a], Pedram Tavadze [a], François-Xavier Coudert [c], Aldo H. Romero [a,*]

[a] *Department of Physics and Astronomy, West Virginia University, Morgantown, WV 26505-6315, USA*
[b] *Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854, USA*
[c] *Chimie ParisTech, PSL University, CNRS, Institut de Recherche de Chimie Paris, 75005, Paris, France*

ARTICLE INFO

ABSTRACT

The MechElastic Python package evaluates the mechanical and elastic properties of bulk and 2D materials using the elastic coefficient matrix (C_{ij}) obtained from any *ab-initio* density-functional theory (DFT) code. The current version of this package reads the output of VASP, ABINIT, and Quantum Espresso codes (but it can be easily generalized to any other DFT code) and performs the appropriate post-processing of elastic constants as per the requirement of the user. This program can also detect the input structure's crystal symmetry and test the mechanical stability of all crystal classes using the Born-Huang criteria. Various useful material-specific properties such as elastic moduli, longitudinal and transverse elastic wave velocities, Debye temperature, elastic anisotropy, 2D layer modulus, hardness, Pugh's ratio, Cauchy's pressure, Kleinman's parameter, and Lame's coefficients, can be estimated using this program. Another existing feature of this program is to employ the ELATE package (2016) [29] and plot the spatial variation of several elastic properties such as Poisson's ratio, linear compressibility, shear modulus, and Young's modulus in three dimensions. Further, the MechElastic package can plot the equation of state (EOS) curves for energy and pressure for a variety of EOS models such as Murnaghan, Birch, Birch-Murnaghan, and Vinet, by reading the inputted energy/pressure *versus* volume data obtained *via* numerical calculations or experiments. This package is particularly useful for the high-throughput analysis of elastic and mechanical properties of materials.

Program summary

Program Title: MechElastic
CPC Library link to program files: https://doi.org/10.17632/y9zc7zybrm.1
Developer's repository link: https://github.com/romerogroup/MechElastic
Licensing provisions: GPLv3
Programming language: Python
Nature of problem: To automatize and simplify the analysis of elastic and mechanical properties for bulk and 2D materials, especially for high-throughput DFT calculations.
Solution method: This Python program addresses the above problem by parsing the elastic coefficient matrix obtained from the first-principles DFT calculations, does the appropriate post-processing to evaluate the elastic properties, and performs the mechanical stability tests for all crystal classes using the generalized Born-Huang criteria. It can also be used to carry out the equation of state analysis to study the structural phase transitions.
Additional comments: MechElastic can be downloaded using the below link:
https://github.com/romerogroup/MechElastic
Installation (*via* PyPI): pip install mechelastic, or pip3 install mechelastic

---

1. Introduction

The elastic and mechanical properties of any material are among the key properties that must be thoroughly investigated to facilitate the proper integration of that material into the emerging technology. One way to analyze the elastic and mechanical response of materials is *via* the first-principles density-functional theory (DFT) calculations. The success of DFT [1–4] not only enabled us to predict the properties of materials from *ab-intio* calculations, but it also expedited the discovery and design of novel materials with a given set of properties for desired practical applications. Modern DFT codes, *e.g.*, ABINIT [5–8], VASP [9,10], SIESTA [11], Quantum Espresso [12,13], exciting [14], WIEN2k [15], CASTEP [16], ELK [17], and CRYSTAL [18,19], offer the capability to efficiently compute the elastic stiffness tensor of materials to a remarkable accuracy (*i.e.*, to the accuracy of the employed DFT method). In fact, such calculations have been performed in recent years and materials databases containing the DFT calculated elastic tensor of hundreds of bulk and two-dimensional materials have been developed [20–26]. Further, various software tools [27] such as ELATE [28,29], AELAS [30], ElaStic [31], and ELAM [32], to the best of our knowledge, have been developed to carry out the analysis of elastic properties from first-principles calculations.

With the ever-increasing power of computing resources and burgeoning interest in the high-throughput DFT calculations [33–36], there is a growing demand to develop software tools that can facilitate the automation of the elastic and mechanical analysis of bulk as well as of 2D materials, from the knowledge of the DFT computed elastic tensor. Moreover, access to the full elastic tensor enables one to determine numerous other important elastic, mechanical, and thermodynamic properties of materials vital for the screening of materials in the process of materials discovery and design [37,38,23,39]. Therefore, it is of crucial interest to have a software tool that can calculate and/or estimate the material-specific properties obtainable from the elastic tensor and be employed in a high-throughput manner.

Considering the above requirements, we develop the MᴇᴄʜEʟᴀsᴛɪᴄ package, a robust open-source Python library, that can facilitate the analysis of the elastic and mechanical properties of bulk and 2D materials in a manual way as well as in a high-throughput manner. The MᴇᴄʜEʟᴀsᴛɪᴄ package reads the DFT calculated elastic tensor data obtained from various DFT codes (current implementation supports the output generated from the VASP [9,10], ABINIT [5–8], and Quantum Espresso [12,13] codes, but it can be easily generalized to any other DFT code), stores the elastic tensor following the Voigt notation [37], and carries out the calculation of several important elastic moduli and mechanical parameters as per user's requirement. This package can also perform the Born-Huang mechanical stability tests [40] for all crystal classes. Furthermore, elastic wave velocities, hardness, Debye temperature, and melting temperatures can be estimated using this package. One can also use this package to plot the equation of state (EOS) curves and investigate the structural phase transitions in the spirit of the Murnaghan, Birch, Birch-Murnaghan, and Vinet EOS models [41–47].

Notably, we have interfaced the MᴇᴄʜEʟᴀsᴛɪᴄ package with the ELATE software [29]. This interface enables one to analyze the three-dimensional spatial distribution of linear compressibility, shear modulus, Young's modulus, and Poisson's ratio, and make the desired plots offline, *i.e.*, without requiring access to the website of ELATE [28]. This feature is particularly useful for characterization of the elastic anisotropy and auxetic features in materials [39,48–50,24,51–55]. It is worth mentioning that our package offers several capabilities, as described below, that are not yet offered by other existing software tools such as ELATE [28,29], AELAS [30], ElaStic [31], and ELAM [32].

This paper focuses on the presentation of the MᴇᴄʜEʟᴀsᴛɪᴄ package, details the underlying algorithms and numerical implementations, and provides essential instructions to users. The outline of this paper is as follows: In Section 2, we briefly explain some basic but necessary details of the elastic tensor and introduce the Voigt notation for crystalline systems. In Section 3, we present the Born-Huang mechanical stability test criteria for different crystal classes and discuss the role of crystal symmetries on the elastic tensor. Section 4 contains the details of the methods and mathematical expressions employed for evaluation of elastic, mechanical, and thermodynamic properties. Section 5 describes the methods used for the estimation of hardness and presents a guide to choose the best hardness model for a given crystal class and type (insulator/semiconductor/metal) of material. Section 6 focuses on the elastic properties and mechanical stability of 2D materials. Section 7 summarizes the integration details of the ELATE [28] software tool into the MᴇᴄʜEʟᴀsᴛɪᴄ package. Section 8 is dedicated to various implemented EOS models. In Section 9, we present an overview of the MᴇᴄʜEʟᴀsᴛɪᴄ library, and finally, in Section 10, we present some examples illustrating the capabilities of the MᴇᴄʜEʟᴀsᴛɪᴄ package, which is being followed by summary in Section 11.

2. Elastic tensor

Under the homogeneous deformation of crystal, the generalized form of stress-strain Hooke's law reads [37]

$$\sigma_{ij} = C_{ijkl}\, \epsilon_{kl}, \tag{1}$$

where σ_{ij} and ϵ_{kl} are the homogeneous two-rank stress and strain tensors, respectively. C_{ijkl} denotes the fourth-rank elastic stiffness tensor. Equation (1) represents a set of nine equations having a total of 81 C_{ijkl} coefficients in a compact form. The symmetry of the stress and strain tensors implies

$$\sigma_{ij} = \sigma_{ji} \implies C_{ijkl} = C_{jikl}, \text{ and} \tag{2}$$

$$\epsilon_{kl} = \epsilon_{lk} \implies C_{ijkl} = C_{ijlk}. \tag{3}$$

These relations reduce the total number of independent C_{ijkl} coefficients from 81 to 36. The stress–strain relation can be derived from a strain energy density-functional (U) through the work-conjugate relation [56], as given below

$$\sigma_{ij} = \frac{\partial U}{\partial \epsilon_{ij}}. \tag{4}$$

Equation (1) suggests

$$C_{ijkl} = \frac{\partial \sigma_{ij}}{\partial \epsilon_{kl}} \implies C_{ijkl} = \frac{\partial^2 U}{\partial \epsilon_{kl}\partial \epsilon_{ij}}. \tag{5}$$

Here, we note that $C_{ijkl} = C_{klij}$ due to the arbitrariness of the order of differentiation. These so-called major symmetries of the stiffness tensor further reduce the total number of independent C_{ijkl} coefficients from 36 to 21.

Using the matrix notation [37], we can abbreviate the four-suffixes stiffness tensor C_{ijkl} into a two-suffixes stiffness tensor, *i.e.*, the first two suffixes are abbreviated into a single suffix ranging from 1 to 6, and the last two suffixes are abbreviated in the same way [37]. To do this, we utilize the symmetry of the second-rank stress and strain tensors and write them as six-dimensional vectors in an orthonormal coordinate system. For instance,

$$\sigma_{ij} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{6}$$

is simplified to a six-dimensional vector in the Voigt notation as: $(\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}) \equiv (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$.

Similarly, ϵ_{ij} can be simplified to: $(\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, \epsilon_{yz}, \epsilon_{xz}, \epsilon_{xy}) \equiv (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6)$.

Thus, in the Voigt notation the stress-strain relationship can be expressed as follows

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}. \tag{7}$$

Similarly, the elastic compliance tensor $(s_{ij} = C_{ij}^{-1})$ can be written as

$$\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} \\ s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} \\ s_{51} & s_{52} & s_{53} & s_{54} & s_{55} & s_{56} \\ s_{61} & s_{62} & s_{63} & s_{64} & s_{65} & s_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix}. \tag{8}$$

The abbreviations in the Voigt notation are done according to the following scheme: $11 \rightarrow 1, 22 \rightarrow 2, 33 \rightarrow 3, 23, 32 \rightarrow 4, 31, 13 \rightarrow 5$, and $12, 21 \rightarrow 6$. Here $1, 2, 3, 4, 5$, and 6 numerals denote xx, yy, zz, yz, zx, and xy components, respectively, of the two-rank stress and strain tensors in an orthonormal coordinate system (note that $yz = zy$, $zx = xz$, and $xy = yx$). Henceforth, we describe the elastic stiffness tensor within the Voigt notation, *i.e.*, by a symmetric 6×6 matrix as defined in Eq. (7).

Considering the major symmetries of the stiffness tensor, as discussed above, and the crystal symmetries, the total number of independent coefficients can be reduced to 3, 5, 7, 7, 9, and 13 for cubic, hexagonal, rhombohedral, tetragonal, orthorhombic, and monoclinic crystal systems, respectively [37,57]. For triclinic systems, all 21 C_{ij} coefficients remain independent.

Expert users may choose to skip the next Section (Section 3) and move on to Section 4 for information regarding the evaluation of various parameters and elastic moduli obtainable from the C_{ij} tensor.

3. Mechanical stability criteria and role of crystal symmetries

The knowledge of C_{ij} matrix enables one to test the mechanical (or elastic) stability of an unstressed crystal using the well-known Born-Huang mechanical stability criteria [58,40,57,59]. A crystal can be considered mechanically stable if and only if C_{ij} matrix is positive definite. Although it is a necessary condition, it is not sufficient for generic class of crystal systems. Mouhat and Coudert [57] have elegantly described the necessary and sufficient conditions to test the mechanical stability for all crystal classes. We refer the reader to Ref. [57] for a detailed information of the total number of independent C_{ij} coefficients and mechanical stability conditions for all eleven Laue crystal classes.

In the following, we present the form of the symmetric C_{ij} matrix along with the mechanical stability conditions for various crystal classes that we employ in the MechElastic package. We use symbols $\bullet$ and $\cdot$ to represent the nonzero and zero components of the symmetric elastic tensor, respectively. Space group (SPG) numbers are also listed for each crystal class. The following information is collected from Refs. [37,40,57,59,60].

3.1. Cubic: SPG#195–230

Cubic crystal systems have only three independent C_{ij} coefficients, which are C_{11}, C_{12}, and C_{44}.

$$\begin{bmatrix} C_{11} & C_{12} & C_{12} & \cdot & \cdot & \cdot \\ \bullet & C_{11} & C_{12} & \cdot & \cdot & \cdot \\ \bullet & \bullet & C_{11} & \cdot & \cdot & \cdot \\ & & & C_{44} & \cdot & \cdot \\ & & & & C_{44} & \cdot \\ & & & & & C_{44} \end{bmatrix} \tag{9}$$

The mechanical stability conditions for cubic systems are:

$$C_{11} - C_{12} > 0, \quad C_{11} + 2C_{12} > 0, \quad \text{and } C_{44} > 0. \tag{10}$$

3.2. Hexagonal: SPG#168–194

Hexagonal crystal systems have five independent C_{ij} coefficients that are C_{11}, C_{12}, C_{13}, C_{33}, and C_{44}.

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \cdot & \cdot & \cdot \\ \bullet & C_{11} & C_{13} & \cdot & \cdot & \cdot \\ \bullet & \bullet & C_{33} & \cdot & \cdot & \cdot \\ & & & C_{44} & \cdot & \cdot \\ & & & & C_{44} & \cdot \\ & & & & & (C_{11}-C_{12})/2 \end{bmatrix} \tag{11}$$

The mechanical stability conditions for this system are:

$$C_{11} - C_{12} > 0, \;\; 2C_{13}^2 < C_{33}(C_{11} + C_{12}), \;\; \text{and } C_{44} > 0. \tag{12}$$

3.3. Rhombohedral class: SPG#143–167

Rhombohedral crystals can be divided into two classes: I (Laue class $\bar{3}m$), and II (Laue class $\bar{3}$) [57]. Rhombohedral class II has seven independent C_{ij} coefficients (C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{33}, and C_{44}), as shown below. One coefficient, C_{15}, vanishes in the rhombohedral class I, thus leaving only six nonzero independent coefficients in the class I.

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & \cdot \\ \bullet & C_{11} & C_{13} & -C_{14} & -C_{15} & \cdot \\ \bullet & \bullet & C_{33} & \cdot & \cdot & \cdot \\ \bullet & \bullet & & C_{44} & \cdot & -C_{15} \\ \bullet & \bullet & & & C_{44} & C_{14} \\ & & & \bullet & \bullet & (C_{11}-C_{12})/2 \end{bmatrix} \tag{13}$$

The generic mechanical stability conditions for rhombohedral class I and II are:

$$C_{11} - C_{12} > 0, \;\; C_{44} > 0,$$
$$C_{13}^2 < \frac{1}{2}C_{33}(C_{11}+C_{12}), \;\; \text{and} \tag{14}$$
$$C_{14}^2 + C_{15}^2 < \frac{1}{2}C_{44}(C_{11}-C_{12}) \equiv C_{44}C_{66}.$$

3.4. Tetragonal: SPG#75–142

Similar to the rhombohedral crystal systems, tetragonal crystal systems can be divided into two classes: I (Laue class $4/mmm$) and II (Laue class $4/m$) [57]. Class I has six independent C_{ij} coefficients (C_{11}, C_{12}, C_{13}, C_{33}, C_{44}, and C_{66}), whereas class II has seven independent C_{ij} coefficients with $C_{16} \neq 0$, as shown below.

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \cdot & \cdot & C_{16} \\ \bullet & C_{11} & C_{13} & \cdot & \cdot & -C_{16} \\ \bullet & \bullet & C_{33} & \cdot & \cdot & \cdot \\ & & & C_{44} & \cdot & \cdot \\ & & & & C_{44} & \cdot \\ \bullet & \bullet & & & & C_{66} \end{bmatrix} \tag{15}$$

The generic mechanical stability conditions for both tetragonal classes are:

$$C_{11} - C_{12} > 0, \;\; 2C_{13}^2 < C_{33}(C_{11}+C_{12}),$$
$$C_{44} > 0, \;\; \text{and } 2C_{16}^2 < C_{66}(C_{11}-C_{12}). \tag{16}$$

3.5. Orthorhombic class: SPG#16–74

Orthorhombic crystal systems have nine independent C_{ij} coefficients as shown below.

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \cdot & \cdot & \cdot \\ \bullet & C_{22} & C_{23} & \cdot & \cdot & \cdot \\ \bullet & \bullet & C_{33} & \cdot & \cdot & \cdot \\ & & & C_{44} & \cdot & \cdot \\ & & & & C_{55} & \cdot \\ & & & & & C_{66} \end{bmatrix} \tag{17}$$

The generic mechanical stability conditions for orthorhombic class are:

$$C_{11} > 0, \quad C_{11}C_{22} > C_{12}^2,$$
$$C_{44} > 0, \quad C_{55} > 0, \quad C_{66} > 0, \tag{18}$$
$$C_{11}C_{22}C_{33} + 2C_{12}C_{13}C_{23} - C_{11}C_{23}^2 - C_{22}C_{13}^2 - C_{33}C_{12}^2 > 0.$$

3.6. Monoclinic class: SPG#3–15

Monoclinic crystal systems have thirteen independent C_{ij} coefficients as shown below.

$$
\begin{bmatrix}
C_{11} & C_{12} & C_{13} & \cdot & C_{15} & \cdot \\
\bullet & C_{22} & C_{23} & \cdot & C_{25} & \cdot \\
\bullet & \bullet & C_{33} & \cdot & C_{35} & \cdot \\
 & & & C_{44} & \cdot & C_{46} \\
\bullet & \bullet & \bullet & & C_{55} & \cdot \\
 & & & \bullet & & C_{66}
\end{bmatrix}
\tag{19}
$$

The generic mechanical stability conditions for the standard monoclinic class are [59]:

$$C_{ii} > 0, \quad \forall\, i \in \{1,6\},$$
$$C_{11} + C_{22} + C_{33} + 2(C_{12} + C_{13} + C_{23}) > 0,$$
$$C_{33}C_{55} - C_{35}^2 > 0,$$
$$C_{44}C_{66} - C_{46}^2 > 0,$$
$$C_{22} + C_{33} - 2C_{23} > 0, \tag{20}$$
$$C_{22}(C_{33}C_{55} - C_{35}^2) + 2C_{23}C_{25}C_{35} - C_{23}^2C_{55} - C_{25}^2C_{33} > 0, \text{ and}$$
$$2[C_{15}C_{25}(C_{33}C_{12} - C_{13}C_{23}) + C_{15}C_{35}(C_{22}C_{13} - C_{12}C_{23}) + C_{25}C_{35}(C_{11}C_{23} -$$
$$C_{12}C_{13})] - [C_{15}^2(C_{22}C_{33} - C_{23}^2) + C_{25}^2(C_{11}C_{33} - C_{13}^2) + C_{35}^2(C_{11}C_{22} - C_{12}^2)] + C_{55}g > 0,$$
$$\text{where, } g = C_{11}C_{22}C_{33} - C_{11}C_{23}^2 - C_{22}C_{13}^2 - C_{33}C_{12}^2 + 2C_{12}C_{13}C_{23}.$$

3.7. Triclinic class: SPG#1–2

Triclinic crystal systems can have all 21 nonzero independent C_{ij} coefficients.

$$
\begin{bmatrix}
C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\
 & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\
 & & C_{33} & C_{34} & C_{35} & C_{36} \\
 & & & C_{44} & C_{45} & C_{46} \\
 & & & & C_{55} & C_{56} \\
 & & & & & C_{66}
\end{bmatrix}
\tag{21}
$$

The generic expressions for the mechanical stability conditions for triclinic class are quite complex [57]. One can check the positive definiteness of the C_{ij} matrix to ensure the mechanical stability of triclinic systems [57].

4. Elastic and mechanical properties of materials

In this section, we present the formulae used for the evaluation of various parameters and moduli related to the elastic and mechanical properties of materials.

Ideally, the stress and strain tensors (or fields) vary in position space $\mathbf{r}$ within a polycrystalline material. Hence, the generic expression for Hooke's law in a homogeneous linear elastic polycrystal (in a simplified Voigt notation) is

$$\sigma_i(\mathbf{r}) = C_{ij}(\mathbf{r})\,\epsilon_j(\mathbf{r}). \tag{22}$$

In a polycrystalline material, the evaluation of effective elastic constants and mechanical properties requires an averaging scheme (in $\mathbf{r}$ space) to determine the average stress $\langle \sigma \rangle$ and strain $\langle \epsilon \rangle$ under the approximation of a "statistically uniform" sample, where

$$\langle \sigma \rangle = \frac{1}{V} \int \sigma(\mathbf{r})d\mathbf{r}, \quad \text{and} \quad \langle \epsilon \rangle = \frac{1}{V} \int \epsilon(\mathbf{r})d\mathbf{r}. \tag{23}$$

Here, V is the volume. In Equation (23) we assume that $\sigma(\mathbf{r})$ and $\epsilon(\mathbf{r})$ vary slowly and continuously in the position space of a polycrystal.

Voigt (1928) [61] and Reuss (1929) [62] proposed two different averaging schemes to obtain the effective elastic constants. Voigt's scheme, also known as Voigt bound, relies on the assumption that the strain field is uniform throughout the sample, *i.e.*, $\epsilon(\mathbf{r})$ is independent of $\mathbf{r}$. Whereas, Reuss's scheme, also known as Reuss bound, relies on the assumption that the stress field is uniform throughout

the sample, *i.e.*, $\sigma(\mathbf{r})$ is independent of $\mathbf{r}$. The Voigt and Reuss bounds provide an upper and lower estimate of the actual elastic moduli, respectively.

In 1952, Hill [63] noted that an arithmetic mean of the Voigt and Reuss bounds, also known as the Voigt-Reuss-Hill (VRH) average, provides a good estimate of the elastic moduli, which is often close to the experimental data [38]. Here it is worth mentioning that, in addition to the aforementioned three bounds, numerous other bounds have been reported in literature, which are nicely summarized by Kube and de Jong in Ref. [64].

In the MECHELASTIC package, we employ the Voigt [61] (subscript V), Reuss [62] (subscript R), and VRH [63] (subscript VRH) averaging schemes to obtain an estimate of the effective elastic moduli. The expressions of various elastic moduli for all three averaging schemes are given below [38].

Voigt averaging scheme (upper bound):

$$K_V = \frac{1}{9}\left[(C_{11} + C_{22} + C_{33}) + 2(C_{12} + C_{23} + C_{31})\right]. \tag{24}$$

Bulk modulus,

$$G_V = \frac{1}{15}[(C_{11} + C_{22} + C_{33}) - (C_{12} + C_{23} + C_{31}) \tag{25}$$
$$+ 3(C_{44} + C_{55} + C_{66})].$$

Shear modulus,

Reuss averaging scheme (lower bound): In the Reuss averaging scheme [62] we use the elastic compliance tensor s_{ij} to compute the elastic moduli.

$$\frac{1}{K_R} = (s_{11} + s_{22} + s_{33}) + 2(s_{12} + s_{23} + s_{31}). \tag{26}$$

Bulk modulus,

$$\frac{15}{G_R} = 4(s_{11} + s_{22} + s_{33}) - 4(s_{12} + s_{23} + s_{31}) \tag{27}$$
$$+ 3(s_{44} + s_{55} + s_{66}).$$

Shear modulus,

Voigt-Reuss-Hill (VRH) average: In this scheme [63], we take an arithmetic average of the Voigt and Reuss bounds.

$$K_{VRH} = \frac{K_V + K_R}{2} \tag{28}$$

Bulk modulus,

$$G_{VRH} = \frac{G_V + G_R}{2} \tag{29}$$

Shear modulus,

From the knowledge of C_{ij}, bulk (K) and shear (G) moduli, one can estimate various other material-specific parameters, as listed below

$$E = \frac{9KG}{3K + G}. \tag{30}$$

Young's modulus,

$$\nu = \frac{3K - E}{6K} \equiv \frac{3K - 2G}{2(3K + G)}. \tag{31}$$

Isotropic Poisson's ratio,

$$M = K + \frac{4G}{3}. \tag{32}$$

P-wave modulus,

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}. \tag{33}$$

Lame's first parameter [38],

$$\mu = \frac{E}{2(1 + \nu)}. \tag{34}$$

Lame's second parameter [38],

$$[65] = \frac{K}{G}. \tag{35}$$

Pugh's ratio [65] =

$$\zeta = \frac{C_{11} + 8C_{12}}{7C_{11} - 2C_{12}}. \tag{36}$$

Kleinman's parameter [66,67],

$$P_C = C_{12} - C_{44}. \tag{37}$$

Cauchy's pressure,

In the following, we present a brief introduction to the moduli defined in Eqs. (32)–(37). The pressure wave or P-wave modulus, also known as the longitudinal modulus, is associated with the homogeneous isotropic linear elastic materials and it describes the ratio of axial stress to the axial strain in a uniaxial strain state [38].

Lame's first and second parameters help us to parameterize the Hooke's law in 3D for homogeneous and isotropic materials using the stress and strain tensors. The first parameter λ provides a measure of the compressibility, whereas the second parameter μ is associated with the shear stiffness of a given material [38].

Pugh's ratio defines the ductility or brittleness of a given material. The critical value of Pugh's ratio is found to be 1.75. Materials with $K/G > 1.75$ are ductile, whereas those with $K/G < 1.75$ are brittle in nature [53,54,65,68].

Kleinman's parameter (ζ) describes the stability of a material under bending or stretching. A low/high value of ζ indicates that system is less/more resistant to bond bending. $\zeta = 0$ implies that bond bending would be dominated, while $\zeta = 1$ implies that bond stretching would be dominated [66,67].

Cauchy's pressure (P_C) is associated with the angular characteristic of atomic bonding in a given material. A positive value of P_C dictates the dominance of ionic bonding, while a negative value of P_C dictates the dominance of covalent bonding.

4.1. Elastic anisotropy

Elastic anisotropy is an important property to characterize for a comprehensive understanding of the physical and mechanical properties of crystals [48,69–73,64]. Even the highly-symmetric cubic crystals show substantial directional dependence of the elastic properties [69]. Various different methods [69–72], as listed below, have been reported in literature to quantify the elastic anisotropy from the knowledge of elastic moduli and C_{ij} tensor.

Zener's anisotropy (for cubic crystals only) [69]

$$A_Z = \frac{2C_{44}}{C_{11} - C_{12}}. \tag{38}$$

Chung-Buessem empirical anisotropy index (for cubic crystals only) [70]

$$A_{CB} = \frac{G_V - G_R}{G_V + G_R}. \tag{39}$$

Ranganathan and Ostoja-Starzewski universal anisotropy index (for any locally isotropic crystal) [71]

$$A_U = \frac{K_V}{K_R} + 5\frac{G_V}{G_R} - 6. \tag{40}$$

Kube's log-Euclidean anisotropy index (for any crystal class) [72]

$$A^L = \sqrt{\left[ln\left(\frac{K_V}{K_R}\right)\right]^2 + 5\left[ln\left(\frac{G_V}{G_R}\right)\right]^2}. \tag{41}$$

Zener's anisotropy [69] was the first attempt at quantifying anisotropy in cubic crystals. Zener proposed a comparison of the ratio of shear resistances of the on-axis and the off-axis terms, this produced the term shown in Eq. (38). Whenever the resistances are equal (*i.e.*, isotropic case) equation Eq. (38) produces unity, and any deviation from unity can be regarded as a measure of the anisotropic behavior of the crystal under investigation. The shortcomings of this equation are: (i) only the cubic elastic tensor was considered, and (ii) it is a non-unique definition, *i.e.*, the equation could have easily been defined as its inverse instead. In literature, this was a starting point for an anisotropy measure, but more recently more generalized definitions of elastic anisotropy are used since they can be used on other crystal classes and are uniquely defined [70–72].

Chung-Buessem anisotropy [70] was defined to obtain "single-valuedness" for cubic crystals, which allows for relative comparison of cubic crystals. In Eq. (39), $A_{CB} = 0$ for isotropic crystals and any positive deviation from this limiting value would indicate an anisotropic behavior. With this definition one can say for certain whether a given cubic crystal is more anisotropic than the other.

The evolution of the definition of elastic anisotropy eventually led to the need of its generalization for any crystal system. The universal anisotropy index (Eq. (40)) was proposed for this purpose [71]. This generalization takes into account the tensorial nature of the elastic tensor and as well as all the elastic moduli. Similar to the Chung-Buessem anisotropy, the universal anisotropy index [71] defines the limiting value of isotropic crystals to be zero, and any positive deviations from zero would indicate the elastic anisotropy. One must take care in this definition as it only provides relative comparison between two crystals; it does not allow for definitive comparison such as: one crystal is twice as anisotropic as another.

The Kube's log-Euclidean anisotropy [72] is the most general definition of anisotropy at present, as it was devised to make definitive comparisons between any two crystals. This is done by defining a distance measure between the Ruess and Voigt estimation of elastic constants. Log-Euclidean anisotropy is very similar in behavior to the previous two anisotropies. Here, isotropy is determined by $A^L = 0$ and any positive value denote a measure of the elastic anisotropy. Now, thanks to this definition, one can definitively say that one crystal is twice as anisotropic as another one.

4.2. Elastic wave velocities

The longitudinal (v_l), transverse (v_t), and average (v_m) elastic wave velocities can be estimated from the knowledge of the bulk and shear moduli, and the density of material (ρ) [74,75].

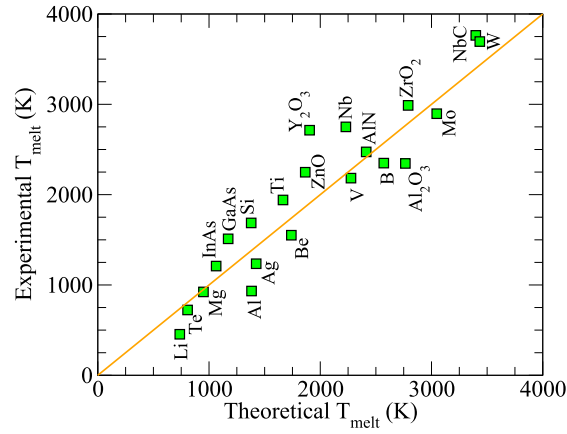$$v_l = \sqrt{\frac{3K + 4G}{3\rho}}, \tag{42}$$

$$v_t = \sqrt{\frac{G}{\rho}}, \text{ and} \tag{43}$$

$$v_m = \left[\frac{1}{3}\left(\frac{2}{v_t^3} + \frac{1}{v_l^3}\right)\right]^{-1/3}. \tag{44}$$

The above equations also imply that one can determine the elastic moduli and elastic stiffness constants by measuring the elastic wave velocities using ultrasonic waves [76–79]. The MechElastic package uses the VRH average of K and G values for calculation of v_l and v_t.

Fig. 1. A comparison of the theoretically estimated (using Eq. (46)) and experimentally reported melting temperature (T_{melt}) data for some elemental and binary compounds. All the structures and their elastic-tensor data were obtained from the Materials Project database [22], and the experimental data were obtained from the website: https://www.americanelements.com/. The mp ID of the structures is as follows: Li (mp-135), Te (mp-19), Mg (mp-153), InAs (mp-20305), GaAs (mp-2534), Si (mp-149), Al (mp-134), Ag (mp-124), Ti (mp-46), Be (mp-87), ZnO (mp-2133), Y_2O_3 (mp-2652), Nb (mp-75), V (mp-146), AlN (mp-661), B (mp-160), Al_2O_3 (mp-1143), ZrO_2 (mp-1565), Mo (mp-129), NbC (mp-910), and W (mp-91).

4.3. Debye temperature

Debye temperature (Θ_D) is an important thermodynamic parameter that can be estimated from the knowledge of the average elastic wave velocity v_m and the density of material ρ. We note that at low-temperatures Θ_D estimated from the elastic constants is approximately same as the Θ_D obtained from the specific heat measurements. This is due to the fact that at low-temperatures the acoustic phonons are the only vibrational excitations contributing to the specific heat of material. We calculate Θ_D using the following expression [74]:

$$\Theta_D = \frac{h}{k_B}\left[\frac{3q}{4\pi}\frac{N\rho}{M}\right]^{1/3} v_m, \tag{45}$$

where h is the Planck's constant, k_B is the Boltzmann's constant, q is the total number of atoms in cell, N is the Avogadro's number, ρ is the density, and M is the molecular weight.

4.4. Melting temperature

The melting temperature (T_{melt}) is estimated using the below empirical relation [80,81], which is a very crude approximation and its validity must be checked carefully for the material under investigation.

$$T_{melt} = 607 + 9.3\,K_{VRH} \pm 555. \tag{46}$$

In order to test the validity of Eq. (46), we estimated T_{melt} for several materials using the elastic-tensor data obtained from the Materials Project database [22], and compared it with the experimental values, as shown in Fig. 1. As one can notice, there is a definite correlation between the theoretical and experimental values, however, the uncertainty in the theoretical estimates of T_{melt} is substantial. Therefore, one should be careful in using Eq. (46) for theoretical estimation of T_{melt}. Also, note that the units of K_{VRH} and T_{melt} in the above empirical relation are in GPa and K, respectively.

5. Hardness

DFT has become a powerful tool for calculating the elastic tensor of materials, allowing the successful estimation of mechanical properties such as bulk (K), shear (G), Young's moduli (E), and Poisson's ratio (ν). Nevertheless, first-principles calculations are unable to predict hardness directly. Since hardness is a fundamental property that is essential to fully describe the mechanical behavior of a solid, various semi-empirical relations have been proposed to estimate hardness using the commonly known elastic moduli. The MECHELASTIC package has implementation of six different semi-empirical relations, as given below with their respective sources [83–87], to calculate Vickers hardness (GPa units) from the macroscopic elastic parameters K, G, E, and ν. These semi-empirical relations are nicely summarized by Ivanovskii in Ref. [82].

$$H_{1a} = 0.1475\,G \quad \text{Ref. [83]} \tag{47}$$

$$H_{1b} = 0.0607\,E \quad \text{Ref. [83]} \tag{48}$$

$$H_2 = 0.1769\,G - 2.899 \quad \text{Ref. [84]} \tag{49}$$

$$H_3 = 0.0635\,E \quad \text{Ref. [85]} \tag{50}$$

$$H_4 = \frac{(1 - 2\nu)\,K}{6\,(1 + \nu)} \quad \text{Ref. [86]} \tag{51}$$
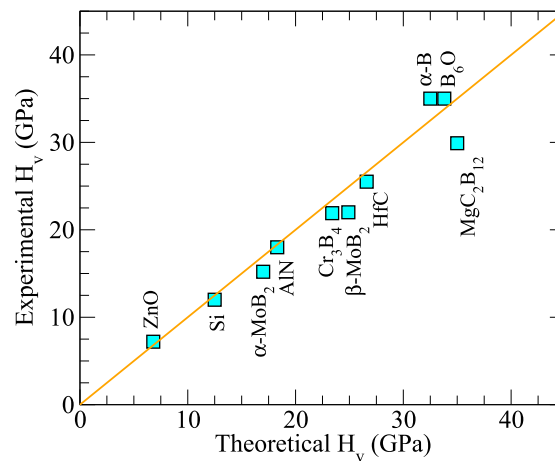
Table 1

A guide to select the best hardness calculation method as a function of the crystal class and bandgap.

	Cubic	Hexagonal	Orthorhombic	Rhombohedral	General
Insulator	H_2	H_{1b}	H_2	H_2	H_2
Semiconductor	H_5	H_{1b}, H_3		H_2	H_5
Metal	H_{1a}	H_4	H_4	H_4	H_4

Table 2

Bulk modulus (GPa), Young's modulus (GPa), shear modulus (GPa), and Poisson's ratio calculated with the MᴇᴄʜEʟᴀsᴛɪᴄ package using the elastic tensor provided in the Materials Project database for several materials. The bandgap (eV) extracted from the Materials Project database was used to select the recommendation model to compute the Vickers hardness (H_v in GPa). The experimental values are given in parenthesis.

Material	mp-ID	Crystal class	Bandgap	Bulk	Shear	Young	Poisson	H_v	H_{method}	Ref.
Diamond	mp-66	Cubic	4.34	435 (446)	521 (535)	1117 (1143)	0.07 (0.07)	89.2 (96.0)	H_2	[87,88]
Si	mp-149	Cubic	0.85	83 (94)	61 (63)	148 (166)	0.20	12.5 (12.0)	H_5	[83,87]
HfC	mp-21075	Cubic	0.00	236 (243)	180 (180)	431 (433)	0.20	26.6 (25.5)	H_{1a}	[83]
AlN	mp-661	Hexagonal	4.05	195 (202)	122 (115)	302 (290)	0.24	18.3 (18.0)	H_{1b}	[83]
ZnO	mp-2133	Hexagonal	0.73	130 (148)	41 (41)	112 (110)	0.36	6.8 (7.2)	H_{1b}	[83]
α-MoB$_2$	mp-960	Hexagonal	0.00	299 (304)	153 (180)	392 (451)	0.28 (0.25)	17.0 (15.2)	H_4	[89]
MgC$_2$B$_{12}$	mp-568803	Orthorhombic	2.29	229 (228)	214 (217)	489 (494)	0.14 (0.14)	35.0 (29.9)	H_2	[85]
Cr$_3$B$_4$	mp-889	Orthorhombic	0.00	299 (299)	210 (215)	511 (520)	0.21 (0.21)	23.4 (21.9)	H_4	[86]
B$_6$O	mp-1346	Rhombohedral	2.21	227 (228)	208 (204)	477 (471)	0.15	33.8 (35.0)	H_2	[83]
α-Boron	mp-160	Rhombohedral	1.52	211 (224)	200 (205)	456 (470)	0.14 (0.12)	32.5 (35.0)	H_2	[83,85]
β-MoB$_2$	mp-2331	Rhombohedral	0.00	295 (298)	225 (215)	537 (520)	0.20 (0.21)	24.9 (22.0)	H_4	[89]

Fig. 2. A comparison of the available experimental Vickers hardness (H_v) data with the theoretical values obtained using the recommended model in Section 5.1 for several materials listed in Table 2.

$$H_5 = 2(k^2 G)^{0.585} - 3 \quad \text{Ref. [87]} \tag{52}$$

The k parameter in H_5 corresponds to the inverse of the Pugh's ratio (G/K).

5.1. Hardness recommendation model

In order to figure out the aptness of the above-mentioned methods in the prediction of hardness for different type of materials, we calculated the Vickers hardness for more than fifty different materials, and compared the obtained theoretical values with the available experimental data. The elastic tensor for each material was extracted from the Materials Project's database [22], and the mechanical properties (K, G, E, and ν) were estimated using the MᴇᴄʜEʟᴀsᴛɪᴄ package. The purpose was to create a guide for users to select the most appropriate method for the hardness calculation based on the specific characteristics of each material. Information of crystal class, energy bandgap (E_g), and density and the presence of specific atoms in the crystal (like C, B, N, O and transition metals) was analyzed, and correlated to minimize the absolute error in the hardness calculation (a more comprehensive study would be published elsewhere). The best model was found to be the one that correlates the crystal class and the energy bandgap. For this study the materials were classified as insulators ($E_g > 2$ eV), semiconductors ($0 < E_g < 2$ eV) and metals ($E_g = 0$ eV). Table 1 presents a guide to select the best method to compute hardness for different type of materials.

Table 2 presents the mechanical properties of several materials calculated using the MᴇᴄʜEʟᴀsᴛɪᴄ package. In this investigation we used the elastic tensor provided in the Materials Project database [22] for each material and compared the theoretically estimated hardness values with the experimental data reported in the literature. The hardness values were calculated using the above-mentioned recommendation model (see Table 1). Fig. 2 illustrates a comparison between the theoretical and experimental values of the Vickers hardness for materials listed in Table 2. Note that the data for diamond was removed in Fig. 2 for clarity.

6. Elastic properties of 2D materials

In order to evaluate the elastic and mechanical properties of 2D materials, one requires to convert the 3D C_{ij}^{3D} tensor obtained for a bulk unit cell (*i.e.*, a 2D layer plus vacuum added to avoid the periodic interactions) from a DFT calculation into a 2D C_{ij}^{2D} tensor. This can be done by multiplying the C_{ij}^{3D} with the vacuum thickness t, *i.e.*, $C_{ij}^{2D} = tC_{ij}^{3D}$. t can be provided by the user, else it would be automatically estimated by the MechElastic package when '-d = 2D' option is specified for the 2D case. By default, MechElastic assumes that vacuum is added along the c-axis. Moreover, MechElastic automatically converts the bulk units of C_{ij}^{3D} from GPa (or kBar) to N/m for 2D case.

Once C_{ij}^{2D} ($\equiv c_{ij}$) is determined, one can estimate various 2D elastic moduli using the expressions given below [30,90–95]. Here, we use the lowercase letter c_{ij} to denote the 2D elastic constants. Note that only four c_{ij} (*i.e.*, c_{11}, c_{22}, c_{12}, and c_{66}) are relevant for 2D square, rectangular, or hexagonal lattices, where symmetry of the square and hexagonal lattices enforces $c_{11} = c_{22}$, and $c_{66} = \frac{1}{2}(c_{11} - c_{12})$. In 2D oblique lattices having no symmetry, two additional coefficient c_{16} and c_{26} are allowed to remain nonzero [96].

2D layer modulus is defined as [90]:

$$\gamma = \frac{1}{4}(c_{11} + c_{22} + 2c_{12}). \tag{53}$$

2D Young's moduli (*i.e.*, in-plane stiffness) for strains in the Cartesian [10] and [01] directions are [90]:

$$Y_{[10]} = \frac{c_{11}c_{22} - c_{12}^2}{c_{22}}, \text{ and } Y_{[01]} = \frac{c_{11}c_{22} - c_{12}^2}{c_{11}}. \tag{54}$$

2D Poisson's ratio in the Cartesian [10] and [01] directions [90]:

$$\nu_{[10]} = \frac{c_{12}}{c_{22}}, \text{ and } \nu_{[01]} = \frac{c_{12}}{c_{11}}. \tag{55}$$

2D shear modulus is defined as [90]:

$$G = c_{66}. \tag{56}$$

Since the above-mentioned properties are well defined only for the 2D materials, they require a re-scaling for quasi-2D systems having a considerable finite thickness along the out-of-plane direction. One can convert the 2D units of N/m into the bulk unit of N/m^2 by dividing the elastic moduli by the thickness of material.

6.1. 2D mechanical stability criteria and role of crystal symmetries

With the c_{ij} matrix we are able to test the mechanical stability of an unstressed 2D lattice using the conditions presented by Marcin Maździarz in Ref. [96]. Similar to the 3D case, mechanical stability in 2D is not just determined by a check of the positive definiteness of the c_{ij} matrix.

In Ref. [96], Maździarz presented the mechanical stability criteria for five Bravais lattices, which are implemented in the MechElastic package. The general form of c_{ij} matrix for any 2D lattice can be written as

$$\begin{bmatrix} C_{11} & C_{12} & C_{16} \\ \bullet & C_{22} & C_{26} \\ \bullet & \bullet & C_{66} \end{bmatrix}. \tag{57}$$

6.1.1. Square lattice conditions

$$C_{11} - C_{12} > 0$$
$$C_{11} + C_{12} > 0 \tag{58}$$
$$C_{66} > 0$$

6.1.2. Rectangular class lattice conditions

$$\frac{1}{2}\left(C_{11} + C_{22} + \sqrt{4C_{12}^2 - (C_{11} - C_{22})^2}\right) > 0$$
$$\frac{1}{2}\left(C_{11} + C_{22} - \sqrt{4C_{12}^2 - (C_{11} - C_{22})^2}\right) > 0 \tag{59}$$
$$C_{66} > 0$$

6.1.3. Hexagonal lattice conditions

$$C_{11} - C_{12} > 0$$
$$C_{11} + C_{12} > 0 \tag{60}$$

6.1.4. Oblique lattice conditions

$$C_{11} > 0$$
$$C_{11}C_{22} > C_{12}^2 \tag{61}$$
$$det(C_{ij}) > 0$$

7. ELATE implementation

ELATE is a popular online elastic tensor analysis tool developed by Gaillac and Coudert [28,29], which takes an elastic tensor as input, and subsequently calculates the spatial variation of the Young's and shear moduli, the linear compressibility, and the Poisson's ratio. With the addition of the mass density, it can also calculate the variations in the compressional and shear sound speed, the ratio of the speeds, and also an estimate of the Debye speed. Along with the variations of the properties, the package calculates the spatial average of all the aforementioned properties. In ELATE, the mechanical stability is tested by finding the eigenvalues of the elastic tensor and checking if they all are positive. A concise summary of the spatial variations of the Young's and shear moduli, linear compressibility, and Poisson's ratio is given below. We refer the reader to Ref. [29] for a more detailed information. In this summary, the minimum and the maximum values, and a measurement of the anisotropy for each property are given along with the direction for which they occur. The spatial variations of the properties are then visualized with 2D contour plots and 3D plots in the elastic property space. The color scheme of these plots are green, red, and blue, which corresponds to the minimal positive value, the minimal negative value, and the maximum possible positive values these properties can take. The 2D contours are taken along the primary axes, whereas the 3D plots belong to the full surface.

In MECHELASTIC, ELATE is converted to an accessible offline tool in Python. An object called ELATE is constructed, which can be used to access the elastic data as the properties of the object. With this object the method `print_properties()` can be used to produce the same concise analysis as the original results obtained by MECHELASTIC. The data for the 2D and 3D plots are generated in the same way as done in the original ELATE software, but are returned as tuples for direct user access. This gives the user freedom to manipulate the data to create their own graphs and figures. These plots are produced by functions `plot_2D()` and `plot_3D()` of the object, and are plotted using the packages `matplotlib` [97] and `pyvista` [98], respectively. The color scheme of the plots have the same meaning as in the original ELATE software [28,29].

8. Equation of state analysis

An Equation of State (EOS) is a thermodynamic relation between the state variables such as temperature (T), volume (V), pressure (P), internal energy, and/or specific heat, etc. It helps us to describe the state of a system under given physical conditions. For instance, it could be used to study a certain property of a material such as its density, under varying pressure or temperature or any other physical variable. The EOS's are used in a wide range of fields ranging from Geo-Physics to Materials Science [99]. Other than predicting thermodynamical properties, EOS can be used to gain insight into the nature of solid-state and molecular theories [100].

The EOS models considered here are for the isothermal processes, where T is kept constant while P and V could be varying. Experimentally, these values can be obtained through x-ray diffraction (XRD), Diamond Anvil Cells (DAC), or shock experiments. However, the pressure range obtainable by such experimental techniques is limited and therefore, it calls for theoretical methods to extend the range for environments in extreme conditions such as planetary interiors.

MECHELASTIC's EOS class contains methods to plot 'Energy vs. Volume' and 'Pressure vs. Volume' curves from energy, pressure and volume data obtained from either calculations or experiments using several fitting models including Vinet, Birch, Murnaghan, and Birch-Murnaghan [41–45,101,100,46,47,102]. As calculations are sensitive to the type of the EOS model, outcomes from multiple models should be carefully examined [103]. MECHELASTIC employs a central-difference scheme to calculate the pressure from the 'Energy vs. Volume' data, and similarly an integration scheme to calculate the energy from 'Pressure vs. Volume' data. This is especially helpful for studying the phase diagrams of materials to investigate the existence of phase boundaries.

To the best of our knowledge, currently there is no universal EOS model that is applicable to all types of solids and accurate over the whole range of pressure, especially when a solid undergoes several structural phase transitions within the region of interest [104]. Therefore, we have included several EOS models in the MECHELASTIC package whose accuracy can be evaluate by the mean-squared error (MSE) of the fitting. These models are briefly described below.

8.1. Vinet EOS

The Vinet EOS model is based on the empirical interatomic potentials and is formulated as follows [100,46]:

$$E(\eta) = E_o + \frac{2B_o V_o}{\left(B'_o - 1\right)^2} \times \left(2 - \left(5 + 3B'_o(\eta - 1) - 3\eta\right) e^{-3\left(B'_o - 1\right)(\eta - 1)/2} \right) \tag{62}$$

where, $\eta = (V/V_o)^{1/3}$. V_0, E_0, B_0, and B'_0 denote the equilibrium volume, total energy, bulk modulus and its pressure derivative at zero pressure, respectively.

Vinet model performs better at high pressure and high compressibility (up to 40%) since it includes non-linear pressure contributions as opposed to the Birch and Murnaghan EOS models [100]. However, it is not suitable for solids with significant structural flexibility, such as bond-bending in materials such as feldspars.

8.2. Birch EOS

The Birch EOS model is an empirical model as written below [106]:

$$E(V) = E_0 + \frac{9}{8} B_0 V_0 \left[(V_0/V)^{2/3} - 1 \right]^2 + \frac{9}{16} B_0 V_0 \left(B'_0 - 4\right) \times$$
$$\left[(V_0/V)^{2/3} - 1 \right]^3 + \sum_{n=4}^{N} \gamma_n \left[(V_0/V)^{2/3} - 1 \right]^n \tag{63}$$
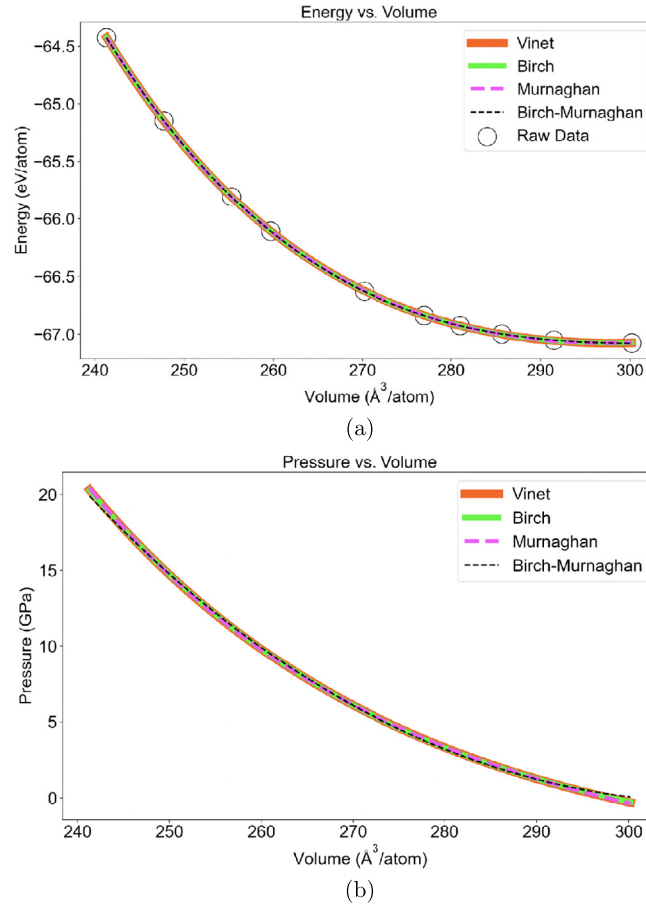
Fig. 3. The fitted equation of state (EOS) plots for (a) Energy vs. Volume and (b) Pressure vs. Volume data obtained by differentiating the energy with a central difference scheme using different EOS models. Here, the material under investigation is T_d-MoTe$_2$ for which 'Energy vs. Volume' data was computed using the VASP code (numerical details are given in Ref. [105]).

where, E_0, V_0, B_0 and B_0' are the equilibrium energy, volume, bulk modulus and pressure derivative of the bulk modulus, respectively. n denotes the order of the fit.

For experimental high pressure data, Birch model fares better than the Murnaghan model, but underperforms the Vinet model [107, 108].

8.3. Murnaghan EOS

The Murnaghan EOS model is given as below [109]:

$$E_T(V) = E_T(V_0) + \frac{B_0 V}{B_0'} \left[\frac{(V_0/V)^{B_0'}}{B_0' - 1} + 1 \right] - \frac{V_0 B_0}{B_0' - 1} \tag{64}$$

where, V is the volume, and B_0 and B_0' denote the bulk modulus and its pressure derivative at the equilibrium volume V_0.

While results from the Murnaghan model agree well with the data obtained at low pressures and low compressions (up to 10%), they deviate from the high pressure ones [100]. The Murnaghan EOS is based on the empirical data and it does not include non-linear pressure contributions. When deriving the Murnaghan EOS, it is assumed that the bulk modulus is a linear function of pressure, *i.e.*, $K = K_0 + P K_0'$ [110].

8.4. Birch-Murnaghan EOS

This model is based on the pressure expansion of the bulk modulus and finite strain theory, and consequently, it is valid only at moderate compression [103]. This model assumes Eulerian strains under hydrostatic compression. This EOS is continuously differentiable and higher-order terms of the Taylor expansion are negligible. The Birch-Murnaghan EOS model is the most used EOS model in Earth physics. The model for the second order is shown below.

$$E(\eta) = E_o + \frac{9 B_o V_o}{16} \left(\eta^2 - 1 \right)^2 \left(6 + B_o' \left(\eta^2 - 1 \right) - 4\eta^2 \right) \tag{65}$$

where, $\eta = (V/V_o)^{1/3}$ and V_0, B_0, B_0' and E_0 the total energy at zero pressure are the fitting parameters. B_0, B_0' are the bulk modulus at zero pressure and its pressure derivative, respectively.

```
Fitting coefficients [E0 (eV), B0 (GPa), Bp (GPa), V0 (Anstroms^3)]
along with Mean Squared Error (MSE)

Vinet : [-67.074  0.259  8.232  297.941]
Vinet (MSE) : 8.377e-06

Birch : [-67.075  0.251  8.939  298.112]
Birch (MSE) : 6.999e-06

Murnaghan : [-67.074  0.278  6.698  297.574]
Murnaghan (MSE) : 1.385e-05

Birch-Murnaghan : [-67.076  0.128 -28.930  301.014]
Birch-Murnaghan (MSE) : 4.219e-07

Based on MSE Birch-Murnaghan is the best EOS model for this dataset
```

Fig. 4. The output of an EOS analysis with MᴇᴄʜEʟᴀsᴛɪᴄ. MᴇᴄʜEʟᴀsᴛɪᴄ also prints the best EOS model to use for the provided dataset. The EOS fitting was performed for the T_d-MoTe$_2$ dataset, as shown in Fig. 3.

An improvement of the Birch-Murnaghan model from other models is the inclusion of the third-order strain components [111].

Initially, a text file with two columns containing volume and energy or volume and pressure, respectively, should be passed into MᴇᴄʜEʟᴀsᴛɪᴄ. With this, an initial second order parabolic `numpy` [112] polyfit is performed to obtain the initial fitting parameters, E_0 or P_0, B_0, B_p and V_0. Afterwards, a more accurate fitting for the energy or pressure is performed using the least square method in `scipy` [113] against each EOS model. A central-difference scheme is used to obtain the pressure from this fitted energy. If pressure is to be obtained from energy, an integrating scheme is used instead. Finally, a plot of 'Energy vs. Volume' and 'Pressure vs. Volume' is obtained, as shown in Fig. 3, comparing the values obtained from the different EOS models with the originally provided raw data. With these plots, an analysis of the phase boundaries could be performed. A range of initial and final values for the volume can be set with the `vlim` parameter and if not given, the minimum and maximum would be calculated from the provided dataset. A desired model can be provided with the `model` parameter and if not specified, the calculation would be performed against all the available models.

In addition to the fitting, MᴇᴄʜEʟᴀsᴛɪᴄ also performs a regression analysis for fitting using the Mean-Squared Error (MSE) of the residuals. This allows users to decide the best EOS model to use for their dataset. A sample output of the EOS analysis is shown in Fig. 4, which displays the fitting coefficients, E_0, B_0, B_p, and V_0 along with the Mean-Squared Error (MSE) of the residuals for each EOS model.

Usage:

```
from mechelastic import EOS
eos_object = EOS()
eos_object.plot_eos("energyvsvolume.dat", eostype="energy", natoms=1, au=False)
```

`energyvsvolume.dat` is a text file with volume as the first column and energy as the second. `eostype` is set to `energy` for this case. If pressure and volume data is provided, `eostype` is set to `pressure` instead. `natoms` is the number of atoms in the structure and is used to output the quantities per atom. Setting `au=True` sets the units in Ha while it is in eV when False.
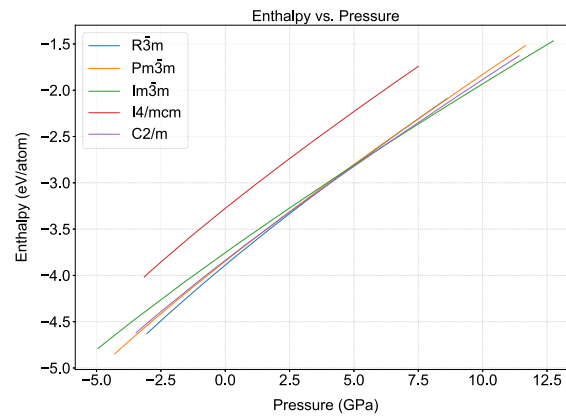
8.5. Enthalpy curves

Enthalpy (H) is defined as the sum of a system's internal energy (U) and the product of its pressure (P) and volume (V), *i.e.*,

$$H = U + PV. \tag{66}$$

It is a measure of a system's capacity to do non-mechanical work and the ability to release heat. MᴇᴄʜEʟᴀsᴛɪᴄ's EOS analyzing tool is equipped with a function to calculate the 'Enthalpy vs. Pressure' curves provided the energy. Pressure is calculated from the 'Energy vs. Volume' data using a central-difference scheme, similar to the one mentioned in the previous section. MᴇᴄʜEʟᴀsᴛɪᴄ first automatically detects the suitable EOS model for the fitting for each phase, and then it calculates the pressure for corresponding energy values. The `plot_enthalpy_curves` method implemented within MᴇᴄʜEʟᴀsᴛɪᴄ plots the 'Enthalpy vs. Pressure' curves for multiple datasets of different phases and determines the phase-transition pressure by identifying the intersection points among the curves. The volume ranges are to be set individually for each phase, using the `vlim_list` option. This is a list that contains elements of the minimum and maximum value of volume for each phase. Care must be given while selecting the volume ranges, as different models are sensitive to different volume ranges for each phase, and it may produce erroneous results if `vlim_list` exceeds the valid volume range. If not provided MᴇᴄʜEʟᴀsᴛɪᴄ will determine the best volume ranges based on the input data.

The 'Enthalpy vs. Pressure' curves for five different phases of bismuth are plotted in Fig. 5. The data used in these calculations were obtained using the VASP code [9,10] with the PBE exchange-correlation functional [114]. An energy cutoff of 350 eV was employed together with k-grids of size $6 \times 6 \times 12$, $5 \times 5 \times 10$, $12 \times 12 \times 12$, $14 \times 14 \times 14$, and $10 \times 10 \times 4$ for phases $C2/m$, $I4/mcm$, $Im\bar{3}m$, $Pm\bar{3}m$, and

Fig. 5. Enthalpy variation with pressure for five different phases of Bi as calculated using the `plot_enthalpy_curves` function.

Fig. 6. Possible phase transitions of Bi as calculated using the `plot_enthalpy_curves` function. The top most transition has the lowest enthalpy, hence it is energetically the most favorable phase transition.

$R\bar{3}m$, respectively. Additionally, the phase-transition pressures and their corresponding enthalpies were calculated for all the intersections among the 'Enthalpy vs. Pressure' curves for different phases. This is particularly helpful in identifying phase boundaries and the potential phase transitions. The output of this calculation is shown in Fig. 6. The transition paths are ordered with ascending enthalpy. This means that the first transition is energetically the most favorable one and the last transition is the least favorable. This is further elucidated in Fig. 7, where a network plot is presented with a color bar depicting the relative energetics of different phase transition paths. Furthermore, the enthalpy differences with respect to a certain phase can be plot with `deltaH_index=<phase_index>`, where `phase_index` is the index of the file corresponding to the baseline phase.

Usage:

```
from mechelastic import EOS
eos_object = EOS()
infiles = ['R-3m', 'Pm-3m', 'Im-3m', 'I4_mcm', 'C2_m']
natoms = [6, 1, 2, 9, 4]
eos_object.plot_enthalpy_curves(infiles, natoms, au=False)
# set au=True to convert units from eV to Ha.
```
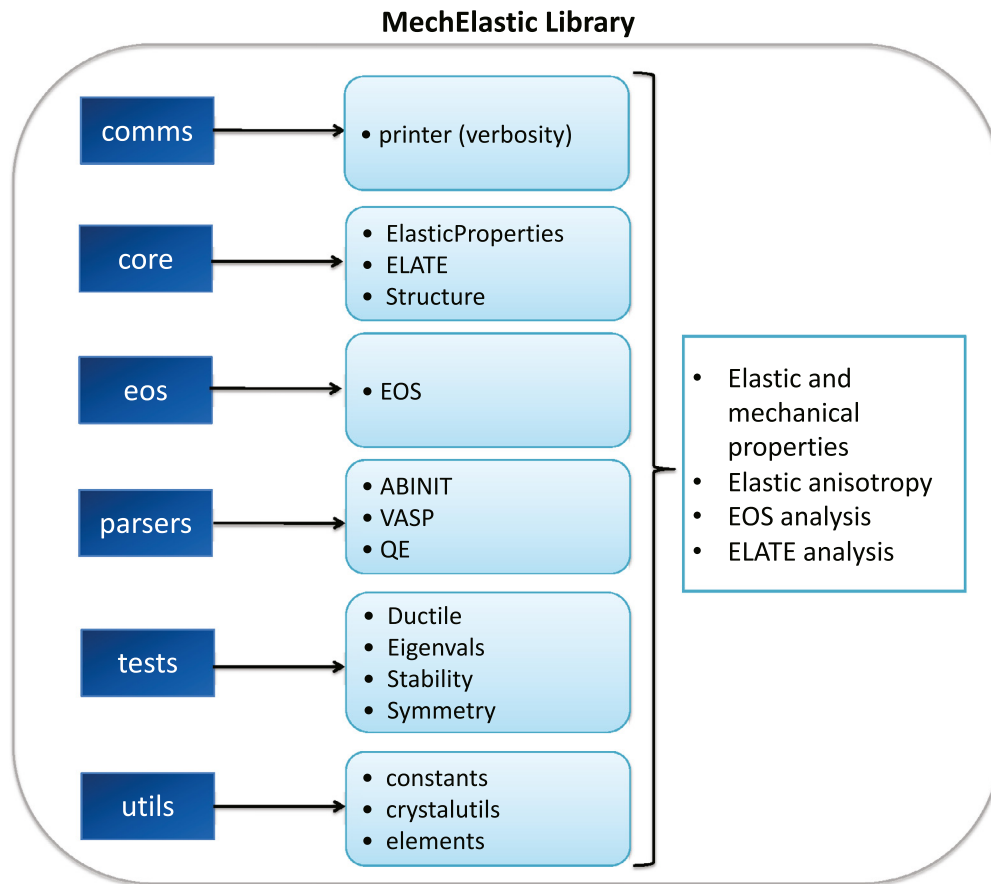
9. Library overview

MECHELASTIC package makes the best out of the Python's object-oriented nature, and utilizes classes and methods for the optimal efficiency. MECHELASTIC consists of six main components namely, `comms`, `core`, `eos`, `parsers`, `tests`, and `utils` to perform a multitude of tasks ranging from parsing the data generated by DFT calculations to testing the structure stability, EOS fittings, and providing estimation of various useful mechanical and elastic moduli. A brief overview of MECHELASTIC library is shown in Fig. 8. The six main components are briefly explained below.

1. `comms`: This contains methods to print messages (verbosity) to the screen such as the MECHELASTIC welcome screen, warning messages, and formatting of the matrices with proper precision.
2. `core`: This is the heart of the MECHELASTIC library. It contains the main code to calculate the elastic properties of 2D and 3D materials. The ELATE library is also housed here. Other than that, the `core` contains the `Structure` class of an assortment of methods to calculate structure-related properties ranging from the cell volume to the space group symmetry of the structure.
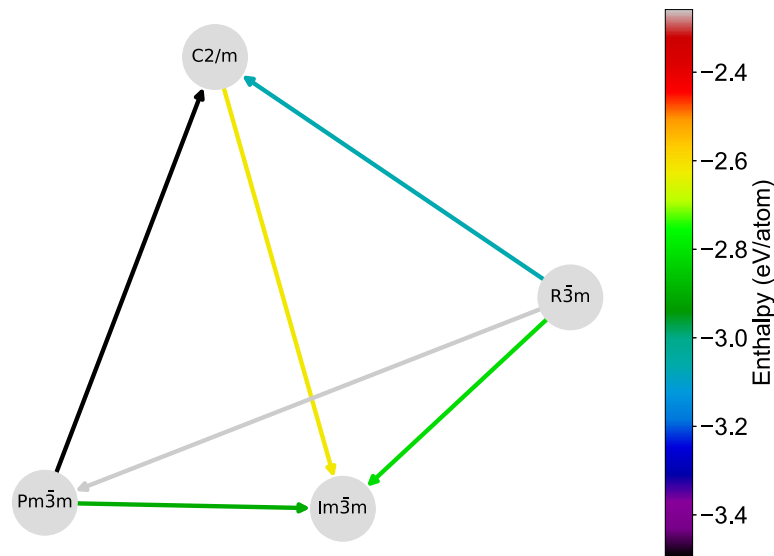
Fig. 7. A network plot of the phase transition paths for different phases of Bi. The color of the arrows represents the relative enthalpy for different phase transitions. This plot was generated using the NetworkX package [115]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Fig. 8. A structural overview of the MECHELASTIC library.

3. `eos`: It contains scripts associated with the EOS fittings (see Section 8).
4. `parsers`: This segment parses the output data from DFT calculations, mainly the elastic tensor and other quantities such as structural properties and pressure. It currently supports parsing output from the VASP, ABINIT, and Quantum Espresso codes. In the case of VASP, whenever there is a non-zero hydrostatic pressure, the elastic tensor needs to be corrected for the residual pressure on cell. This is explained in more detail in Appendix A.
5. `tests`: To correctly study the elastic properties of materials, it is necessary to perform certain tests. This class contains tests for ductility, positive eigenvalues of the elastic tensor, and the Born-Huang mechanical stability tests for different crystal classes, as described in Section 3. Another test is the symmetry of the elastic tensor.
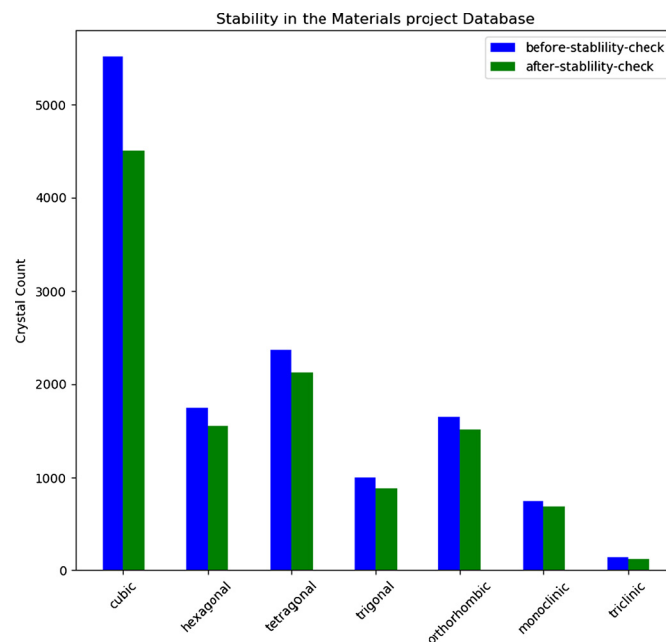
Fig. 9. Use of MECHELASTIC in the stability screening of structures obtained from the materials project (MP) database. The total number of materials with available elastic tensor in the MP database is 13,167 (at the time of writing this paper). After the stability check, 11,390 structure are determined to be mechanically stable.

Fig. 10. MECHELASTIC's `print_properties()` for stability check.

6. `utils`: This class contains several handy utilities including a dictionary of commonly used constants, the atomic masses and the atomic number of elements, and a method to retrieve the crystal symmetry calculated from the crystal structure using the `spglib` library [116].

9.1. Installation

The latest stable version of the MECHELASTIC, version 1.2.21 at the time of writing this paper, can be installed using the Python Packaging Index (`pip`) using the following command:

```
pip install mechelastic
```

or

```
pip3 install mechelastic
```

The project's GitHub repository is located at https://github.com/romerogroup/MechElastic.

An easy to follow documentation with examples can be found at https://romerogroup.github.io/mechelastic/.

```
--------------------------------------------------------------
Elastic Moduli
--------------------------------------------------------------

                        Voigt    Reuss    Average
--------------------------------------------------------------
Bulk modulus   (GPa)    102.889  102.439  102.664
Shear modulus  (GPa)     58.333   55.261   56.797
Young's modulus (GPa)   147.184  140.515  143.850
Poisson's ratio           0.262    0.271    0.266
P-wave modulus (GPa)    180.667  176.120  178.393
Bulk/Shear ratio          1.764    1.854    1.808 (ductile)
--------------------------------------------------------------

Elastic parameters
--------------------------------------------------------------

Lame's first and second parameter; Ref.[3]
Lambda (GPa) =       64.808
Mu (GPa)     =       56.791

--------------------------------------------------------------
Bonding information
--------------------------------------------------------------

Kleinman's parameter; Ref.[4,5]
NOTE: K = 0 (1) bending (stretching) would dominate
K =      0.55000

Cauchy's Pressure calculated from the relation : CP = C_12 - C_44
      CP > 0 (+ve) indicates that ionic bonding dominates
      CP < 0 (-ve) indicates that covalent bonding dominates
CP (GPa) =        8.000
Bonding is mainly ionic

--------------------------------------------------------------
Elastic Anisotropy
--------------------------------------------------------------

Zener's anisotropy (true for cubic crystals only); Az =  0.784; Ref.[6]
Chung-Buessem's anisotropy (true for cubic crystals only); Acb =  0.027; Ref.[7]
Universal anisotropy index; Au =  0.282; Ref.[8]
Log-Euclidean's anisotropy; AL =  0.121; Ref.[9]
--------------------------------------------------------------
Elastic Wave Velocities and Debye Temperature
--------------------------------------------------------------

Longitudinal wave velocity (vl) :   6358.119 m/s; Ref.[10]
Transverse wave velocity (vt) :   3587.588 m/s; Ref.[10]
Average wave velocity (vm) :   3990.686 m/s; Ref.[10]
Debye temperature  :    532.192 K; Ref.[10]

WARNING: The  Debye model for the atomic displacement is based on a monoatomic crystal approximation.
 Here we consider an averaged mass, in case your crystal has several species.

--------------------------------------------------------------
Melting Temperature
--------------------------------------------------------------

Melting temperature calculated from the empirical relation: Tm = 607 + 9.3*Kvrh \pm 555 (in K); Ref.[11]
Tm =    1561.773 K (plus-minus 555 K)
```

Fig. 11. MECHELASTIC's `print_properties()` for elastic properties.

The MECHELASTIC package is supported by Python 3.x.

10. Features and implementations of the MECHELASTIC package

10.1. Mechanically stability test

The materials project [22,21] database has numerous compounds reported with elastic, electronic, and thermal properties. Of those compounds, 13,167 structures have been listed with an elastic tensor. This readily attainable elastic and structure information is useful for determining trends in properties with relation to their structures. To begin this analysis, MECHELASTIC is used as a screening tool to filter out the mechanically unstable structures by checking for positive eigenvalues and the Born-Huang mechanical stability criteria. The results have been filtered by crystal system and are shown in Fig. 9. Of the total 13,167 structures collected from the materials project database, 11,390 structures are found to be mechanically stable at the DFT level.

10.2. LiNbO$_3$ MECHELASTIC IMPLEMENTATION

In this section we demonstrate the implementation of using MECHELASTIC on the example structure LiNbO$_3$. This system is chosen since it is known for having unique elastic properties, possessing piezoelectricity, and it is readily accessible on the materials project database. Its `mp-ID` is mp-3731, and this specific example can be readily accessed with the `mechelastic_w_mpDatabase.py` script in the `examples` folder of the MECHELASTIC package. This script demonstrates how to properly use MECHELASTIC to access properties in an object-oriented fashion.

```
-----------------------------------------------------------------
Hardness Analysis
-----------------------------------------------------------------


Hardness (H1a) = 8.38 GPa;    Ref.[12]
Hardness (H1b) = 8.73 GPa;    Ref.[12]
Hardness (H2)  = 7.15 GPa;    Ref.[13]
Hardness (H3)  = 9.13 GPa;    Ref.[14]
Hardness (H4)  = 6.31 GPa;    Ref.[15]
Hardness (H5)  = 7.63 GPa;    Ref.[16]


Hardness recommendation model:
            ..............................................................
                    Cubic  Hexagonal  Orthorhombic  Rhombohedral  General
            ..............................................................
            Insulator       H2       H1b         H2           H2          H2
            Semiconductor   H5      H1b, H3                    H2          H5
            Metal           H1a       H4          H4           H4          H4
            ..............................................................
            Insulator     : bandgap > 2 eV
            Semiconductor : bandgap < 2 eV
            Metal         : bandgap = 0
```

Fig. 12. MechElastic's `print_properties()` for hardness analysis.

```
Average properties


                        Voigt     Reuss     Hill
-------------------------------------------------------
Bulk modulus   (GPa)    26.889    26.601    26.745
Shear modulus  (GPa)    87.803    81.528    84.712
Young's modulus (GPa)    45.933    41.209    43.571
Poisson's ratio         -0.044    -0.011    -0.028
-------------------------------------------------------


Eigenvalues of compliance matrix


   lamda_1   lamda_2   lamda_3   lamda_4   lamda_5   lamda_6
-------------------------------------------------------------
   31.508    41.272    64.492    68.883    86.117    99.728
-------------------------------------------------------------


Variations of the elastic moduli


                              Min       Max        Anisotropy
-------------------------------------------------------------
Young's Modulus    (GPa)      66.745   107.354       1.608
        Min Axis:   (-0.809, 0.473, 0.35)
        Max Axis:   (0.0, 0.786, 0.619)
-------------------------------------------------------------
Linear Compression (TPa^-1)   10.620    13.486       1.270
        Min Axis:   (0.0, 0.0, 1.0)
        Max Axis:   (0.707, 0.707, 0.0)
-------------------------------------------------------------
Shear Modulus   (GPa)         31.508    64.636       2.051
        Min Axis:   (-1.0, -0.0, 0.0)
        Max Axis:   (0.536, 0.311, 0.785)
        Second Min Axis:   (-0.0, -0.788, -0.615)
        Second Max Axis:   (-0.0, -0.788, -0.615)
-------------------------------------------------------------
Poisson's Ratio              -0.224     0.166    &infin;
        Min Axis:   (0.698, 0.399, -0.595)
        Max Axis:   (-0.452, 0.261, 0.853)
        Second Min Axis:   (-0.495, 0.869, 0.002)
        Second Max Axis:   (-0.504, -0.863, -0.003)
```

Fig. 13. ELATE's output inside MechElastic.

The main summary of the MechElastic package's output screen is displayed in Figs. 10–12. It gives a concise overview of the estimated properties. Fig. 10 shows the summary's mechanical stability check through the Born-Huang criteria and the positive eigenvalues check of the compliance tensor. After checking the mechanical stability, the user can move on to analyze the elastic properties and other estimates

(a)

(b)

(c)

(d)
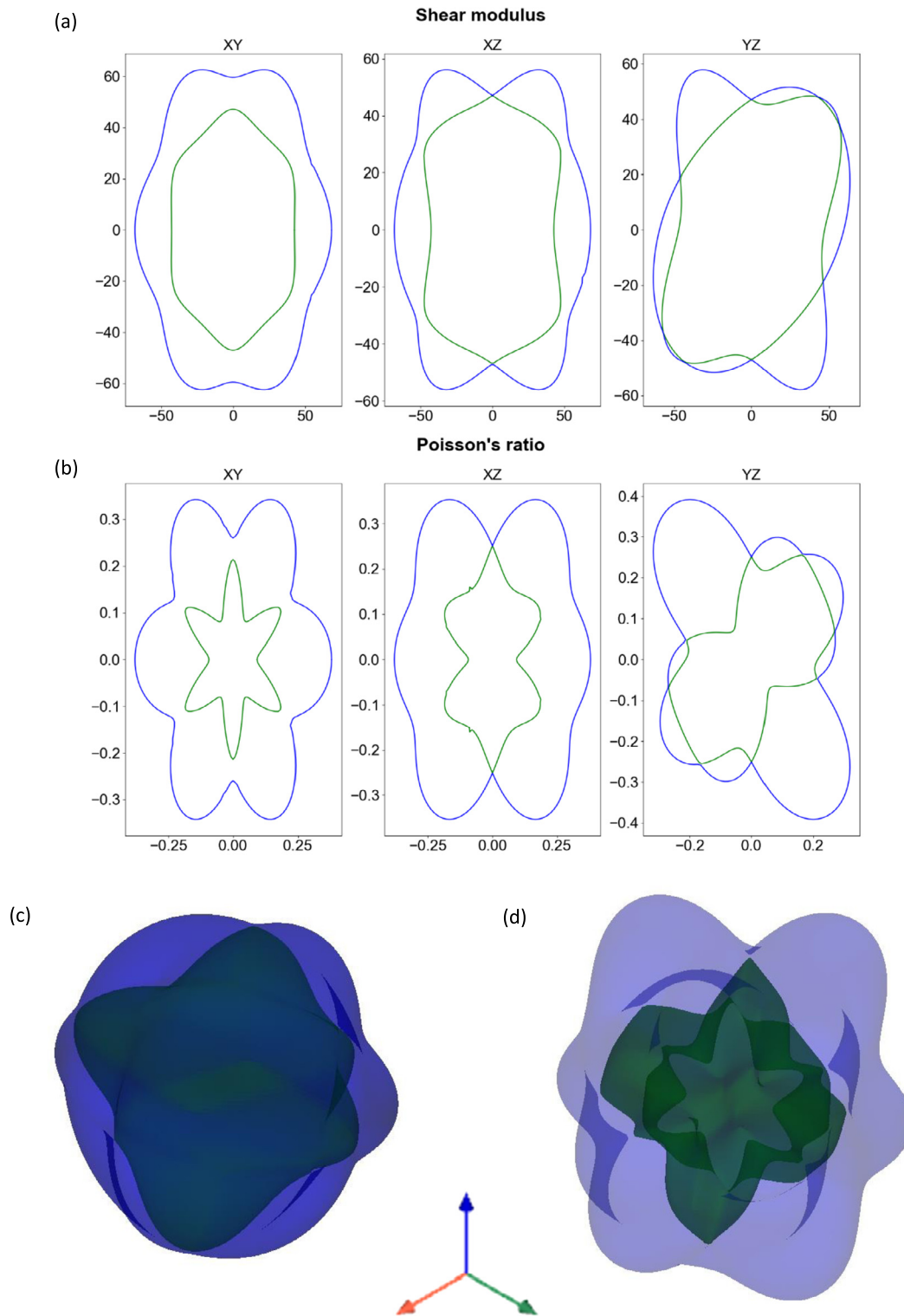
Fig. 14. 2D and 3D examples of the spatial dependence of the shear modulus (a, c) and Poisson's ratio (b, d) obtained by the ELATE-MᴇᴄʜEʟᴀsᴛɪᴄ interface. The green and blue surfaces are the minimal and maximal positive value the elastic property takes, respectively.

that MᴇᴄʜEʟᴀsᴛɪᴄ provides, as shown in Fig. 11. The last part of the MᴇᴄʜEʟᴀsᴛɪᴄ output prints the hardness analysis and the best hardness recommendation model. All these sections are printed subsequently, followed by a list of references so that users can easily find the paper which contains the relevant information on certain elastic estimates.

The print summary of the MᴇᴄʜEʟᴀsᴛɪᴄ-ELATE interface is shown in Fig. 13. We have kept the the same print format as done by the online ELATE interface, so that users can have the same familiarity. The MᴇᴄʜEʟᴀsᴛɪᴄ-ELATE interface outputs a table showing the spatial average of elastic moduli computed using different averaging schemes, eigenvalues of the compliance matrix to show the stability of the structure, and information about the spatial variation of various elastic moduli. We have also kept the ploting format as true to the original as possible. A demonstration of the plots can be seen in Fig. 14, where we use the Poisson's ratio and shear modulus as

an example. Figs. 14(a) and 14(b) show the XY, XZ, and YZ 2D cross-section plots of the shear modulus and Poisson's ratio, respectively. Figs. 14(c) and 14(d) show a snapshot of the 3D distribution of the shear modulus and Poisson's ratio, respectively. In MᴇᴄʜEʟᴀsᴛɪᴄ, we use `pyvista`'s GUI to make the plots. This gives users the ability to rotate the structures and gain a perception of the elastic anisotropy in the studied material.

11. Summary

In summary, MᴇᴄʜEʟᴀsᴛɪᴄ is a user friendly open-source Python library that offers various tools to carry out analysis of the elastic and mechanical properties of bulk as well as of 2D materials. It also provides an offline interface to the ELATE software [28,29] facilitating the study of three-dimensional spatial dependence of Young's modulus, linear compressibility, shear modulus, and Poisson's ratio. The current version of MᴇᴄʜEʟᴀsᴛɪᴄ parses the elastic tensor data generated from the VASP, ABINIT, and Quantum Espresso packages (in future we plan to support others DFT packages). Notably, MᴇᴄʜEʟᴀsᴛɪᴄ assists the users (especially, the apprentices) to correctly study the elastic properties of materials by doing following: (i) By making sure that the elastic tensor under investigation represents a material system that is mechanically stable, *i.e.*, Born-Huang conditions are satisfied, (ii) It ensures that the elastic tensor is properly corrected in case of any residual pressure on the unit cell (for VASP users), and (iii) It provides various warning messages and comments along with proper references to help the user in better understanding the underlying physics of the system. MᴇᴄʜEʟᴀsᴛɪᴄ is specifically designed to be easily integrated in high throughput calculations. From the inputted elastic tensor data, the current version of MᴇᴄʜEʟᴀsᴛɪᴄ can compute or estimate (using empirical relations) the following quantities: bulk, shear, and Young's elastic moduli, Poisson's ratio, Pugh's ratio, P-wave modulus, longitudinal and transverse elastic wave velocities, Debye temperature, elastic anisotropy, 2D layer modulus, hardness, Cauchy's pressure, Kleinman's parameter, and Lame's coefficients. Furthermore, one can analyze several equation of state models such as Vinet, Murnaghan, Birch, and Birch-Murnaghan for a given two column 'Energy vs. Volume' or 'Pressure vs. Volume' data, and further calculate the transition pressures if datasets for multiple phases are present.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

Appendix A. Pressure dependence of elastic constants in VASP

Unlike in several other DFT codes such as ABINIT, in VASP the elastic tensor components need to be adjusted for any residual pressure on the cell while performing elastic constants analysis for the case of non-zero hydrostatic pressure [57,117].

To overcome this issue, hydrostatic pressure (P) is subtracted from the diagonal components (C_{ii}) and added to C_{12}, C_{13}, C_{23}, C_{21}, C_{31} and C_{32}. This is elaborated in the following matrix representation.

$$\begin{bmatrix} C_{11} - P & C_{12} + P & C_{13} + P & C_{14} & C_{15} & C_{16} \\ C_{21} + P & C_{22} - P & C_{23} + P & C_{24} & C_{25} & C_{26} \\ C_{31} + P & C_{32} + P & C_{33} - P & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} - P & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} - P & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} - P \end{bmatrix} \tag{A.1}$$

By default the pressure correction is enabled if MᴇᴄʜEʟᴀsᴛɪᴄ detects a non-zero pressure in VASP's output. However, it could be disabled with the flag `adjust_pressure=False`.

References

[1] P. Hohenberg, W. Kohn, Phys. Rev. 136 (1964) B864–B871, https://doi.org/10.1103/PhysRev.136.B864.
[2] W. Kohn, L.J. Sham, Phys. Rev. 140 (1965) A1133–A1138, https://doi.org/10.1103/PhysRev.140.A1133.
[3] J. Hafner, C. Wolverton, G. Ceder, Mater. Res. Soc. Bull. 31 (9) (2006) 659–668, https://doi.org/10.1557/mrs2006.174.
[4] S. Curtarolo, D. Morgan, G. Ceder, Calphad 29 (3) (2005) 163–211, https://doi.org/10.1016/j.calphad.2005.01.002.
[5] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J.-Y. Raty, D. Allan, Comput. Mater. Sci. 25 (3) (2002) 478–492, https://doi.org/10.1016/S0927-0256(02)00325-7.
[6] X. Gonze, B. Amadon, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Côté, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G.-M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M. Verstraete, G. Zerah, J. Zwanziger, Comput. Phys. Commun. 180 (12) (2009) 2582–2615, https://doi.org/10.1016/j.cpc.2009.07.007.
[7] X. Gonze, F. Jollet, F.A. Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, et al., Comput. Phys. Commun. 205 (2016) 106–131, https://doi.org/10.1016/j.cpc.2016.04.003.

[8] A.H. Romero, D.C. Allan, B. Amadon, G. Antonius, T. Applencourt, L. Baguet, J. Bieder, F. Bottin, J. Bouchet, E. Bousquet, F. Bruneval, G. Brunin, D. Caliste, M. Côté, J. Denier, C. Dreyer, P. Ghosez, M. Giantomassi, Y. Gillet, O. Gingras, D.R. Hamann, G. Hautier, F. Jollet, G. Jomard, A. Martin, H.P.C. Miranda, F. Naccarato, G. Petretto, N.A. Pike, V. Planes, S. Prokhorenko, T. Rangel, F. Ricci, G.-M. Rignanese, M. Royo, M. Stengel, M. Torrent, M.J. van Setten, B. Van Troeye, M.J. Verstraete, J. Wiktor, J.W. Zwanziger, X. Gonze, J. Chem. Phys. 152 (12) (2020) 124102, https://doi.org/10.1063/1.5144261.
[9] G. Kresse, J. Furthmüller, Phys. Rev. B 54 (1996) 11169–11186, https://doi.org/10.1103/PhysRevB.54.11169.
[10] G. Kresse, D. Joubert, Phys. Rev. B 59 (1999) 1758–1775, https://doi.org/10.1103/PhysRevB.59.1758.
[11] J.M. Soler, E. Artacho, J.D. Gale, A. García, J. Junquera, P. Ordejón, D. Sánchez-Portal, J. Phys. Condens. Matter 14 (11) (2002) 2745–2779, https://doi.org/10.1088/0953-8984/14/11/302.
[12] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M.B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A.D. Corso, S. de Gironcoli, P. Delugas, R.A.D. Jr, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N.L. Nguyen, H.-V. Nguyen, A.O. de-la Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A.P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, S. Baroni, J. Phys. Condens. Matter 29 (46) (2017) 465901, https://doi.org/10.1088/1361-648X/aa8f79.
[13] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G.L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A.P. Seitsonen, A. Smogunov, P. Umari, R.M. Wentzcovitch, J. Phys. Condens. Matter 21 (39) (2009) 395502, https://doi.org/10.1088/0953-8984/21/39/395502.
[14] A. Gulans, S. Kontur, C. Meisenbichler, D. Nabok, P. Pavone, S. Rigamonti, S. Sagmeister, U. Werner, C. Draxl, J. Phys. Condens. Matter 26 (36) (2014) 363202, https://doi.org/10.1088/0953-8984/26/36/363202.
[15] P. Blaha, K. Schwarz, F. Tran, R. Laskowski, G.K.H. Madsen, L.D. Marks, J. Chem. Phys. 152 (7) (2020) 074101, https://doi.org/10.1063/1.5143061.
[16] S.J. Clark, M.D. Segall, C.J. Pickard, P.J. Hasnip, M.I. Probert, K. Refson, M.C. Payne, Z. Kristallogr. 220 (5–6) (2005) 567–570, https://doi.org/10.1524/zkri.220.5.567.65075.
[17] K. Dewhurst, S. Sharma, L. Nordstrom, F. Cricchio, F. Bultmark, H. Gross, C. Ambrosch-Draxl, C. Persson, C. Brouder, R. Armiento, et al., http://elk.sourceforge.net, 2016.
[18] W. Perger, J. Criswell, B. Civalleri, R. Dovesi, Comput. Phys. Commun. 180 (10) (2009) 1753–1759, https://doi.org/10.1016/j.cpc.2009.04.022.
[19] R. Dovesi, A. Erba, R. Orlando, C.M. Zicovich-Wilson, B. Civalleri, L. Maschio, M. Rérat, S. Casassa, J. Baima, S. Salustro, B. Kirtman, WIREs Comput. Mol. Sci. 8 (4) (2018) e1360, https://doi.org/10.1002/wcms.1360, https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1360.
[20] W. Setyawan, R.M. Gaume, S. Lam, R.S. Feigelson, S. Curtarolo, ACS Comb. Sci. 13 (4) (2011) 382–390, https://doi.org/10.1021/co200012w.
[21] A. Jain, G. Hautier, C.J. Moore, S. Ping Ong, C.C. Fischer, T. Mueller, K.A. Persson, G. Ceder, Comput. Mater. Sci. 50 (8) (2011) 2295–2310, https://doi.org/10.1016/j.commatsci.2011.02.023.
[22] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, APL Mater. 1 (1) (2013) 011002, https://doi.org/10.1063/1.4812323.
[23] M. de Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. Krishna Ande, S. van der Zwaag, J.J. Plata, C. Toher, S. Curtarolo, G. Ceder, K.A. Persson, M. Asta, Sci. Data 2 (1) (2015) 150009, https://doi.org/10.1038/sdata.2015.9.
[24] F.-X. Coudert, A.H. Fuchs, Coord. Chem. Rev. 307 (2016) 211–236, https://doi.org/10.1016/j.ccr.2015.08.001.
[25] K. Choudhary, I. Kalish, R. Beams, F. Tavazza, Sci. Rep. 7 (1) (2017) 5179, https://doi.org/10.1038/s41598-017-05402-0.
[26] K. Choudhary, G. Cheon, E. Reed, F. Tavazza, Phys. Rev. B 98 (2018) 014107, https://doi.org/10.1103/PhysRevB.98.014107.
[27] R. Yu, J. Zhu, H. Ye, Comput. Phys. Commun. 181 (3) (2010) 671–675, https://doi.org/10.1016/j.cpc.2009.11.017.
[28] http://progs.coudert.name/elate.
[29] R. Gaillac, P. Pullumbi, F.-X. Coudert, J. Phys. Condens. Matter 28 (27) (2016) 275201.
[30] S. Zhang, R. Zhang, Comput. Phys. Commun. 220 (2017) 403–416, https://doi.org/10.1016/j.cpc.2017.07.020.
[31] R. Golesorkhtabar, P. Pavone, J. Spitaler, P. Puschnig, C. Draxl, Comput. Phys. Commun. 184 (8) (2013) 1861–1873, https://doi.org/10.1016/j.cpc.2013.03.010.
[32] A. Marmier, Z.A. Lethbridge, R.I. Walton, C.W. Smith, S.C. Parker, K.E. Evans, Comput. Phys. Commun. 181 (12) (2010) 2102–2115, https://doi.org/10.1016/j.cpc.2010.08.033.
[33] D. Morgan, G. Ceder, S. Curtarolo, Meas. Sci. Technol. 16 (1) (2004) 296–301, https://doi.org/10.1088/0957-0233/16/1/039.
[34] T.R. Munter, D.D. Landis, F. Abild-Pedersen, G. Jones, S. Wang, T. Bligaard, Comput. Sci. Discov. 2 (1) (2009) 015006, https://doi.org/10.1088/1749-4699/2/1/015006.
[35] W. Setyawan, S. Curtarolo, Comput. Mater. Sci. 49 (2) (2010) 299–312, https://doi.org/10.1016/j.commatsci.2010.05.010.
[36] S. Curtarolo, G.L.W. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, Nat. Mater. 12 (3) (2013) 191–201, https://doi.org/10.1038/nmat3568.
[37] J.F. Nye, et al., Physical Properties of Crystals: Their Representation by Tensors and Matrices, Oxford University Press, 1985.
[38] G. Mavko, T. Mukerji, J. Dvorkin, The Rock Physics Handbook, Cambridge University Press, 2020.
[39] S. Chibani, F.-X. Coudert, Chem. Sci. 10 (2019) 8589–8599, https://doi.org/10.1039/C9SC01682A.
[40] M. Born, K. Huang, Dynamical Theory of Crystal Lattices, Clarendon Press, 1954.
[41] F. Birch, J. Appl. Phys. 9 (4) (1938) 279–288, https://doi.org/10.1063/1.1710417.
[42] F. Birch, Phys. Rev. 71 (1947) 809–824, https://doi.org/10.1103/PhysRev.71.809.
[43] F.D. Murnaghan, Finite Deformation of an Elastic Solid, Wiley, 1951.
[44] F. Birch, J. Geophys. Res. 57 (2) (1952) 227–286, https://doi.org/10.1029/JZ057i002p00227.
[45] F. Birch, J. Phys. Chem. Solids 38 (2) (1977) 175–177, https://doi.org/10.1016/0022-3697(77)90162-7.
[46] P. Vinet, J.H. Rose, J. Ferrante, J.R. Smith, J. Phys. Condens. Matter 1 (11) (1989) 1941, https://doi.org/10.1088/0953-8984/1/11/002.
[47] F.D. Stacey, Phys. Earth Planet. Inter. 128 (1–4) (2001) 179–193, https://doi.org/10.1016/S0031-9201(01)00285-0.
[48] R.F.S. Hearmon, Rev. Mod. Phys. 18 (1946) 409–440, https://doi.org/10.1103/RevModPhys.18.409.
[49] A.U. Ortiz, A. Boutin, A.H. Fuchs, F.-X. Coudert, Phys. Rev. Lett. 109 (2012) 195502, https://doi.org/10.1103/PhysRevLett.109.195502.
[50] F.-X. Coudert, Phys. Chem. Chem. Phys. 15 (2013) 16012–16018, https://doi.org/10.1039/C3CP51817E.
[51] R.S. Lakes, Annu. Rev. Mater. Res. 47 (1) (2017) 63–81, https://doi.org/10.1146/annurev-matsci-070616-124118.
[52] J. Dagdelen, J. Montoya, M. de Jong, K. Persson, Nat. Commun. 8 (1) (2017) 323, https://doi.org/10.1038/s41467-017-00399-6.
[53] O. Pavlic, W. Ibarra-Hernandez, I. Valencia-Jaime, S. Singh, G. Avendaño-Franco, D. Raabe, A.H. Romero, J. Alloys Compd. 691 (2017) 15–25, https://doi.org/10.1016/j.jallcom.2016.08.217.
[54] S. Singh, I. Valencia-Jaime, O. Pavlic, A.H. Romero, Phys. Rev. B 97 (2018) 054108, https://doi.org/10.1103/PhysRevB.97.054108.
[55] S.K. Singh, Structural Prediction and Theoretical Characterization of Bi-Sb Binaries: Spin-Orbit Coupling Effects, West Virginia University, 2018, https://doi.org/10.33915/etd.7255.
[56] L. Wang, J.D. Lee, C.-D. Kan, Int. J. Smart and Nano Mater. 7 (3) (2016) 144–178, https://doi.org/10.1080/19475411.2016.1233141.
[57] F. Mouhat, F.-X. Coudert, Phys. Rev. B 90 (2014) 224104, https://doi.org/10.1103/PhysRevB.90.224104.
[58] M. Born, in: Proc. Cambridge Philos. Soc, vol. 36, 1940, pp. 160–172.
[59] Z.-j. Wu, E.-j. Zhao, H.-p. Xiang, X.-f. Hao, X.-j. Liu, J. Meng, Phys. Rev. B 76 (2007) 054115, https://doi.org/10.1103/PhysRevB.76.054115.
[60] S. Singh, Q. Wu, C. Yue, A.H. Romero, A.A. Soluyanov, Phys. Rev. Mater. 2 (2018) 114204, https://doi.org/10.1103/PhysRevMaterials.2.114204.
[61] W. Voigt, et al., Lehrbuch der Kristallphysik, vol. 962, Teubner, Leipzig, 1928.
[62] A. Reuss, Z. Angew. Math. Mech. 9 (1) (1929) 49–58, https://doi.org/10.1002/zamm.19290090104, https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.19290090104.
[63] R. Hill, Proc. Phys. Soc. A 65 (5) (1952) 349–354, https://doi.org/10.1088/0370-1298/65/5/307.
[64] C.M. Kube, M. de Jong, J. Appl. Phys. 120 (16) (2016) 165105, https://doi.org/10.1063/1.4965867.
[65] S.F. Pugh, Philos. Mag. 45 (367) (1954) 823–843, https://doi.org/10.1080/14786440808520496.
[66] L. Kleinman, Phys. Rev. 128 (1962) 2614–2621, https://doi.org/10.1103/PhysRev.128.2614.
[67] W.A. Harrison, Electronic Structure and the Properties of Solids: The Physics of the Chemical Bond, Courier Corporation, 2012.
[68] S. Singh, A.C. Garcia-Castro, I. Valencia-Jaime, F. Muñoz, A.H. Romero, Phys. Rev. B 94 (2016) 161116, https://doi.org/10.1103/PhysRevB.94.161116.

[69] C. Zener, Elasticity and Anelasticity of Metals, University of Chicago Press, 1948.
[70] D.H. Chung, W.R. Buessem, J. Appl. Phys. 38 (5) (1967) 2010–2012, https://doi.org/10.1063/1.1709819.
[71] S.I. Ranganathan, M. Ostoja-Starzewski, Phys. Rev. Lett. 101 (2008) 055504, https://doi.org/10.1103/PhysRevLett.101.055504.
[72] C.M. Kube, AIP Adv. 6 (9) (2016) 095209, https://doi.org/10.1063/1.4962996.
[73] H. Ledbetter, A. Migliori, J. Appl. Phys. 100 (6) (2006) 063516, https://doi.org/10.1063/1.2338835.
[74] O.L. Anderson, J. Phys. Chem. Solids 24 (7) (1963) 909–917, https://doi.org/10.1016/0022-3697(63)90067-2.
[75] E. Screiber, O. Anderson, N. Soga, McGrawHill, New York, 1973.
[76] K.K. Gopinathan, A.R.K.L. Padmini, J. Phys. D, Appl. Phys. 7 (1) (1974) 32–40, https://doi.org/10.1088/0022-3727/7/1/311.
[77] A.J. Lichnowski, G.A. Saunders, J. Phys. C, Solid State Phys. 9 (6) (1976) 927–928, https://doi.org/10.1088/0022-3719/9/6/012.
[78] P.A. Varkey, A.R.K.L. Padmini, Pramana 11 (6) (1978) 717–724, https://doi.org/10.1007/BF02878871.
[79] M. Wiesner, Phase Transit. 82 (10) (2009) 699–754, https://doi.org/10.1080/01411590903372622.
[80] M. Fine, L. Brown, H. Marcus, Scr. Metall. 18 (9) (1984) 951–956, https://doi.org/10.1016/0036-9748(84)90267-9.
[81] I. Johnston, G. Keeler, R. Rollins, S. Spicklemire, Wiley, New York, 1996.
[82] A. Ivanovskii, in: Special Section: Recent Advances of Functionally Graded Hard Materials, Int. J. Refract. Met. Hard Mater. 36 (2013) 179–182, https://doi.org/10.1016/j.ijrmhm.2012.08.013.
[83] X. Jiang, J. Zhao, X. Jiang, Comput. Mater. Sci. 50 (7) (2011) 2287–2290, https://doi.org/10.1016/j.commatsci.2011.01.043.
[84] D.M. Teter, Mater. Res. Soc. Bull. 23 (1) (1998) 22–27, https://doi.org/10.1557/S0883769400031420.
[85] X. Jiang, J. Zhao, A. Wu, Y. Bai, X. Jiang, J. Phys. Condens. Matter 22 (31) (2010) 315503, https://doi.org/10.1088/0953-8984/22/31/315503.
[86] N. Miao, B. Sa, J. Zhou, Z. Sun, Comput. Mater. Sci. 50 (4) (2011) 1559–1566, https://doi.org/10.1016/j.commatsci.2010.12.015.
[87] X.-Q. Chen, H. Niu, D. Li, Y. Li, Intermetallics 19 (9) (2011) 1275–1281, https://doi.org/10.1016/j.intermet.2011.03.026.
[88] C.A. Klein, G.F. Cardinale, Diam. Relat. Mater. 2 (5–7) (1993) 918–923, https://doi.org/10.1016/0925-9635(93)90250-6.
[89] Q. Tao, X. Zhao, Y. Chen, J. Li, Q. Li, Y. Ma, J. Li, T. Cui, P. Zhu, X. Wang, RSC Adv. 3 (40) (2013) 18317–18322, https://doi.org/10.1039/C3RA42741B.
[90] R.C. Andrew, R.E. Mapasha, A.M. Ukpong, N. Chetty, Phys. Rev. B 85 (2012) 125428, https://doi.org/10.1103/PhysRevB.85.125428.
[91] Q. Peng, A.R. Zamiri, W. Ji, S. De, Acta Mech. 223 (12) (2012) 2591–2596, https://doi.org/10.1007/s00707-012-0714-0.
[92] Q. Peng, C. Liang, W. Ji, S. De, Comput. Mater. Sci. 68 (2013) 320–324, https://doi.org/10.1016/j.commatsci.2012.10.019.
[93] S. Singh, A.H. Romero, Phys. Rev. B 95 (2017) 165444, https://doi.org/10.1103/PhysRevB.95.165444.
[94] D. Akinwande, C.J. Brennan, J.S. Bunch, P. Egberts, J.R. Felts, H. Gao, R. Huang, J.-S. Kim, T. Li, Y. Li, K.M. Liechti, N. Lu, H.S. Park, E.J. Reed, P. Wang, B.I. Yakobson, T. Zhang, Y.-W. Zhang, Y. Zhou, Y. Zhu, Extrem. Mech. Lett. 13 (2017) 42–77, https://doi.org/10.1016/j.eml.2017.01.008.
[95] S. Singh, C. Espejo, A.H. Romero, Phys. Rev. B 98 (2018) 155309, https://doi.org/10.1103/PhysRevB.98.155309.
[96] M. Maździarz, 2D Mater. 6 (4) (2019) 048001, https://doi.org/10.1088/2053-1583/ab2ef3.
[97] J.D. Hunter, Comput. Sci. Eng. 9 (3) (2007) 90–95, https://doi.org/10.1109/MCSE.2007.55.
[98] C.B. Sullivan, A.A. Kaszynski, J. Open Sour. Softw. 4 (37) (2019) 1450, https://doi.org/10.21105/joss.01450.
[99] D.L. Anderson, Equations of State of Solids for Geophysics and Ceramic Science, 1996.
[100] P. Vinet, J. Ferrante, J.H. Rose, J.R. Smith, J. Geophys. Res., Solid Earth 92 (B9) (1987) 9319–9325, https://doi.org/10.1029/JB092iB09p09319.
[101] F. Birch, J. Geophys. Res., Solid Earth 83 (B3) (1978) 1257–1268, https://doi.org/10.1029/JB083iB03p01257.
[102] C. Loose, J. Kortus, High Press. Res. 33 (3) (2013) 622–632, https://doi.org/10.1080/08957959.2013.806657.
[103] M. Hebbache, M. Zemzemi, Phys. Rev. B (2004) 6.
[104] E.E. Gordon, J. Köhler, M.-H. Whangbo, Sci. Rep. 6 (1) (2016) 39212, https://doi.org/10.1038/srep39212.
[105] S. Singh, J. Kim, K.M. Rabe, D. Vanderbilt, Phys. Rev. Lett. 125 (2020) 046402, https://doi.org/10.1103/PhysRevLett.125.046402.
[106] Michael J. Mehl, Barry M. Klein, Dimitris A. Papaconstantopoulos, First-Principles Calculation of Elastic Properties, Intermetallic Compounds, vol. 1, John Wiley & Sons Ltd, 1994.
[107] P. Vinet, J. Ferrante, J.H. Rose, J.R. Smith, J. Geophys. Res., Solid Earth 92 (B9) (1987) 9319–9325, https://doi.org/10.1029/JB092iB09p09319, _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JB092iB09p09319.
[108] F. Birch, J. Geophys. Res. (1896-1977) 57 (2) (1952) 227–286, https://doi.org/10.1029/JZ057i002p00227, _eprint, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JZ057i002p00227.
[109] C.L. Fu, K.M. Ho, Phys. Rev. B 28 (10) (1983) 5480–5486, https://doi.org/10.1103/PhysRevB.28.5480.
[110] F.D. Murnaghan, Proc. Natl. Acad. Sci. USA 30 (9) (1944) 244–247, https://doi.org/10.1073/pnas.30.9.244.
[111] F. Birch, Phys. Rev. 71 (11) (1947) 809–824, https://doi.org/10.1103/PhysRev.71.809.
[112] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.
[113] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A.P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C.N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D.A. Nicholson, D.R. Hagen, D.V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G.A. Price, G.-L. Ingold, G.E. Allen, G.R. Lee, H. Audren, I. Probst, J.P. Dietrich, J. Silterra, J.T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J.L. Schönberger, J. de Miranda Cardoso, J. Reimer, J. Harrington, J.L.C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N.J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P.A. Brodtkorb, P. Lee, R.T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T.J. Pingel, T.P. Robitaille, T. Spura, T.R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y.O. Halchenko, Y. Vázquez-Baeza, S. Contributors, Nat. Methods 17 (3) (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.
[114] J.P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (18) (1996) 3865–3868, https://doi.org/10.1103/PhysRevLett.77.3865.
[115] A.A. Hagberg, D.A. Schult, P.J. Swart, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), Proceedings of the 7th Python in Science Conference (SciPy 2008), 2008, pp. 11–15.
[116] A. Togo, I. Tanaka, preprint, arXiv:1808.01590, 2018, https://arxiv.org/abs/1808.01590.
[117] I. Mosyagin, A. Lugovskoy, O. Krasilnikov, Y. Vekilov, S. Simak, I. Abrikosov, Comput. Phys. Commun. 220 (2017) 20–30, https://doi.org/10.1016/j.cpc.2017.06.008.