MechELastic: A Python library for analysis of mechanical and elastic properties of bulk and 2D materials

Sobhit Singh\textsuperscript{a,b,*}, Logan Lang\textsuperscript{a}, Viviana Dovale-Farelo\textsuperscript{a}, Uthpala Heratha\textsuperscript{a}, Pedram Tavadze\textsuperscript{a}, François-Xavier Coudert\textsuperscript{c}, Aldo H. Romero\textsuperscript{a,}\textsuperscript{*}

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

\textbf{A R T I C L E I N F O}

Article history:
Received 27 November 2020
Received in revised form 26 April 2021
Accepted 6 June 2021
Available online 18 June 2021

\textbf{Keywords:}
Elastic properties
Mechanical stability
Elastic wave velocities
Elastic anisotropy
2D materials
High-throughput
DFT
Equation of state

\textbf{A B S T R A C T}

The MechELastic Python package evaluates the mechanical and elastic properties of bulk and 2D materials using the elastic coefficient matrix (C\textsubscript{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) \cite{29} and plot the spatial variation of several elastic properties such as Poison’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.

\textbf{Program summary}

\textbf{Program Title:} MechELastic
\textbf{CPC Library link to program files:} https://doi.org/10.17632/y9zc7zybrm.1
\textbf{Developer’s repository link:} https://github.com/romerogroup/MechElastic
\textbf{Licensing provisions:} GPLv3
\textbf{Programming language:} Python
\textbf{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.
\textbf{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.
\textbf{Additional comments:} MechELastic can be downloaded using the below link:
https://github.com/romerogroup/MechElastic
\textbf{Installation (via PyPI):} pip install mechelastic or pip3 install mechelastic

© 2021 Elsevier B.V. All rights reserved.

\textsuperscript{*} The review of this paper was arranged by Prof. N.S. Scott.
\textsuperscript{**} This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
\textsuperscript{*} Corresponding authors at: Department of Physics and Astronomy, West Virginia University, Morgantown, WV 26505-6315, USA.
\textit{E-mail addresses:} sobhit.singh@rutgers.edu (S. Singh), ahimero@mail.wvu.edu (A.H. Romero).
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-initio 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], AEALAS [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 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 [37,38,39]. 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 MechElastic 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 MechElastic 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 MechElastic 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], AEALAS [30], ElaStic [31], and ELAM [32].

This paper focuses on the presentation of the MechElastic 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 MechElastic package. Section 8 is dedicated to various implemented EOS models. In Section 9, we present an overview of the MechElastic library, and finally, in Section 10, we present some examples illustrating the capabilities of the MechElastic 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]

\[
\sigma_{ij} = C_{ijkl} \epsilon_{kl},
\]

(1)

where \(\sigma_{ij}\) and \(\epsilon_{kl}\) 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

\[
\sigma_{ij} = \sigma_{ji} \implies C_{ijkl} = C_{jikl}, \quad \text{and}
\]

\[
\epsilon_{ij} = \epsilon_{ji} \implies C_{ijkl} = C_{ijlk}.
\]

(2)

(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

\[
\sigma_{ij} = \frac{\partial U}{\partial \epsilon_{ij}}.
\]

(4)

Equation (1) suggests

\[
C_{ijkl} = \frac{\partial \sigma_{ij}}{\partial \epsilon_{kl}} \implies C_{ijkl} = \frac{\partial^2 U}{\partial \epsilon_{kl} \partial \epsilon_{ij}}.
\]

(5)
Here, we note that $C_{ijkl} = C_{ijlk}$ 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,

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

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

Similarly, $\varepsilon_{ij}$ can be simplified to: $(\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{yz}, \varepsilon_{xy}, \varepsilon_{zx}) \equiv (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6)$.

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

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} .$$

(7)

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

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} \\ s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} \\ s_{51} & s_{52} & s_{53} & s_{54} & s_{55} & s_{56} \\ s_{61} & s_{62} & s_{63} & s_{64} & s_{65} & s_{66} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} .$$

(8)

The abbreviations in the Voigt notation are done according to the following scheme: $11 \to 1, 22 \to 2, 33 \to 3, 23 \to 4, 31 \to 5$, and $12, 21 \to 6$. 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 $yz = yz, zy = zy$, and $xy = yx$). Henceforth, we describe the elastic stiffness tensor within the Voigt notation, i.e., by a symmetric $6 \times 6$ 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 $3, 5, 7, 9, 13$, and 15 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 systems. 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 $C_{44}$.

$$\begin{pmatrix} C_{11} & C_{12} & C_{12} \\ \cdot & C_{11} & C_{12} \\ \cdot & \cdot & C_{44} \end{pmatrix}$$

(9)

The mechanical stability conditions for cubic systems are:

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

(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 $C_{44}$.

\[
\begin{bmatrix}
C_{11} & C_{12} & C_{13} & \cdot & \cdot \\
C_{11} & C_{13} & \cdot & \cdot \\
\cdot & C_{33} & \cdot & \cdot \\
C_{44} & \cdot & \cdot & \cdot \\
(11) \\
C_{44} & \cdot & \cdot & \cdot \\
(C_{11} - C_{12})/2
\end{bmatrix}
\]

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.

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

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

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

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.

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

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.

\[
\begin{bmatrix}
C_{11} & C_{12} & C_{13} & \cdot & \cdot & \cdot \\
\cdot & C_{22} & C_{23} & \cdot & \cdot & \cdot \\
\cdot & \cdot & C_{33} & \cdot & \cdot & \cdot \\
C_{44} & \cdot & \cdot & \cdot & \cdot & \cdot \\
C_{55} & \cdot & \cdot & \cdot & \cdot & \cdot \\
C_{66}
\end{bmatrix} \tag{17}
\]
The generic mechanical stability conditions for orthorhombic class are:

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

\[ (18) \]

3.6. Monoclinic class: SPG#3–15

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

\[
\begin{bmatrix}
C_{11} & C_{12} & C_{13} & C_{15} & \cdot & \cdot \\
C_{22} & C_{23} & C_{25} & \cdot & \cdot & \cdot \\
C_{33} & C_{34} & C_{35} & \cdot & \cdot & \cdot \\
C_{44} & C_{45} & C_{46} & \cdot & \cdot & \cdot \\
C_{55} & C_{56} & \cdot & \cdot & \cdot & \cdot \\
C_{66} & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix}
\]

(19)

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

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

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

(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

\[ \sigma_i(\mathbf{r}) = C_{ij}(\mathbf{r}) \varepsilon_j(\mathbf{r}). \]  

(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 \( \langle \varepsilon \rangle \) under the approximation of a “statistically uniform” sample, where

\[ \langle \sigma \rangle = \frac{1}{V} \int \sigma(\mathbf{r}) \, d\mathbf{r}, \quad \text{and} \quad \langle \varepsilon \rangle = \frac{1}{V} \int \varepsilon(\mathbf{r}) \, d\mathbf{r}. \]  

(23)

Here, \( V \) is the volume. In Equation (23) we assume that \( \sigma(\mathbf{r}) \) and \( \varepsilon(\mathbf{r}) \) 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., \( \varepsilon(\mathbf{r}) \) 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):**

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

Shear modulus,  
$$G_V = \frac{1}{15} \left[ (C_{11} + C_{22} + C_{33}) - (C_{12} + C_{23} + C_{31}) + 3(C_{44} + C_{55} + C_{66}) \right].$$

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

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

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

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

Bulk modulus,  
$$K_{VRH} = \frac{K_V + K_R}{2}.$$  

Shear modulus,  
$$G_{VRH} = \frac{G_V + G_R}{2}.$$  

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

Young’s modulus,  
$$E = \frac{9KG}{3K + G}.$$  

Isotropic Poisson’s ratio,  
$$\nu = \frac{3K - E}{6K} = \frac{3K - 2G}{2(3K + G)}.$$  

P-wave modulus,  
$$M = K + \frac{4G}{3}.$$  

Lame’s first parameter [38],  
$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}.$$  

Lame’s second parameter [38],  
$$\mu = \frac{E}{2(1 + \nu)}.$$  

Pugh’s ratio [65],  
$$\frac{K}{G}.$$  

Kleinman’s parameter [66,67],  
$$\zeta = \frac{C_{11} + 8C_{12}}{7C_{11} - 2C_{12}}.$$  

Cauchy’s pressure,  
$$P_C = C_{12} - C_{44}.$$  

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 $\lambda$ provides a measure of the compressibility, whereas the second parameter $\mu$ 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 ($\zeta$) describes the stability of a material under bending or stretching. A low/high value of $\zeta$ 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 Rues 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 (\( \rho \)) [74,75].

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

\[
v_t = \sqrt{\frac{G}{\rho}}, \quad \text{and} \tag{43}
\]

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

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: \texttt{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-2354), Si (mp-149), Al (mp-134), Ag (mp-124), Ti (mp-46), Be (mp-87), ZnO (mp-2133), Y\( _2\)O\( _3 \) (mp-2652), Nb (mp-75), V (mp-146), AlN (mp-661), B (mp-160), Al\( _2\)O\( _3 \) (mp-1143), ZrO\( _2 \) (mp-1565), Mo (mp-129), NbC (mp-910), and W (mp-91).

4.3. Debye temperature

Debye temperature \( \Theta_D \) is an important thermodynamic parameter that can be estimated from the knowledge of the average elastic wave velocity \( \nu_{\text{m}} \) and the density of material \( \rho \). We note that at low-temperatures \( \Theta_D \) estimated from the elastic constants is approximately same as the \( \Theta_D \) 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 \( \Theta_D \) using the following expression [74]:

\[
\Theta_D = \frac{\hbar}{k_B} \left[ \frac{3q}{4\pi} \frac{N\rho}{M} \right]^{1/3} \nu_{\text{m}},
\]

where \( \hbar \) 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, \( \rho \) 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 \( (\nu) \). 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 \( \nu \). 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.

| Material | mp-ID | Crystal class | Bandgap (eV) | Bulk (GPa) | Shear (GPa) | Young (GPa) | Poisson | H<sub>v</sub> | H<sub>method</sub> | Ref. |
|----------|-------|---------------|-------------|------------|------------|-------------|---------|-------------|-----------------|------|
| Insulator | H<sub>2</sub> | Cubic | 4.34 | 435 (446) | 521 (535) | 1117 (1143) | 0.07 (0.07) | 89.2 (96.0) | H<sub>2</sub> | [87,88] |
| H<sub>3</sub> | Hexagonal | 0.85 | 83 (94) | 61 (63) | 148 (166) | 0.20 (12.0) | H<sub>3</sub> | [83,87] |
| H<sub>4</sub> | Orthorhombic | 0.00 | 236 (243) | 208 (204) | 477 (471) | 0.15 (0.12) | H<sub>4</sub> | [83] |
| Rhombohedral | 0.73 | 130 (148) | 41 (41) | 112 (110) | 0.36 (7.2) | H<sub>4</sub> | [83] |
| General | 0.00 | 299 (304) | 153 (180) | 392 (451) | 0.28 (0.25) | 170 (15.2) | H<sub>4</sub> | [89] |

Table 2
Bulk modulus (GPa), Young’s modulus (GPa), shear modulus (GPa), and Poisson’s ratio calculated with the MechELastic 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<sub>v</sub>) in GPa. The experimental values are given in parentheses.

| Material | mp-ID | Crystal class | Bandgap (eV) | Young | Poisson | H<sub>v</sub> | H<sub>method</sub> | Ref. |
|----------|-------|---------------|-------------|-------|---------|-------------|-----------------|------|
| Diamond | mp-66 | Cubic | 4.34 | 435 (446) | 521 (535) | 1117 (1143) | 0.07 (0.07) | 89.2 (96.0) | H<sub>2</sub> | [87,88] |
| AlN | mp-661 | Hexagonal | 4.05 | 195 (202) | 122 (115) | 302 (290) | 0.24 | 18.3 (18.0) | H<sub>1</sub> | [83] |
| MgC<sub>2</sub>B<sub>2</sub> | mp-568803 | Orthorhombic | 2.29 | 229 (228) | 214 (217) | 489 (494) | 0.14 | 35.0 (29.9) | H<sub>2</sub> | [85] |
| Cr<sub>2</sub>B<sub>4</sub> | mp-889 | Orthorhombic | 0.00 | 299 (299) | 210 (215) | 511 (520) | 0.21 | 23.4 (21.9) | H<sub>4</sub> | [86] |
| Si | mp-149 | Cubic | 0.85 | 83 (94) | 61 (63) | 148 (166) | 0.20 | 12.5 (12.0) | H<sub>5</sub> | [83,87] |
| α-B <sup>3</sup>O<sub>3</sub> | mp-960 | Hexagonal | 0.00 | 299 (304) | 153 (180) | 392 (451) | 0.28 | 170 (15.2) | H<sub>4</sub> | [89] |
| MgC<sub>2</sub>B<sub>2</sub> | mp-568803 | Orthorhombic | 2.29 | 229 (228) | 214 (217) | 489 (494) | 0.14 | 35.0 (29.9) | H<sub>2</sub> | [85] |
| Cr<sub>2</sub>B<sub>4</sub> | mp-889 | Orthorhombic | 0.00 | 299 (299) | 210 (215) | 511 (520) | 0.21 | 23.4 (21.9) | H<sub>4</sub> | [86] |
| Si | mp-149 | Cubic | 0.85 | 83 (94) | 61 (63) | 148 (166) | 0.20 | 12.5 (12.0) | H<sub>5</sub> | [83,87] |

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

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

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 MechELastic 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 MechELastic 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 $C_{ij}^{2D}$ (≡ $c_{ij}$) 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. Moreover, $c_{11}$ and $c_{22}$ are symmetric, while $c_{12}$ is antisymmetric. $c_{66}$ is present as a result of the lattice strain occurring in the $c$-direction.

\[ c_{ij} = \begin{pmatrix} c_{11} & c_{12} & c_{16} \\ c_{22} & c_{26} & c_{66} \\ \end{pmatrix}. \]  

(57)

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

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

(54)

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

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

(55)

2D shear modulus is defined as [90]:

\[ G = c_{66}. \]  

(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 Mażdziarz 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], Marżdziarz 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

\[ \begin{pmatrix} c_{11} & c_{12} & c_{16} \\ c_{22} & c_{26} & c_{66} \\ \end{pmatrix}. \]  

(57)

6.1.1. Square lattice conditions

\begin{align*}
C_{11} - C_{12} &> 0 \\
C_{11} + C_{12} &> 0 \\
C_{66} &> 0
\end{align*}  

(58)

6.1.2. Rectangular class lattice conditions

\begin{align*}
\frac{1}{2}(C_{11} + C_{22} + \sqrt{4C_{12}^2 - (C_{11} - C_{22})^2}) &> 0 \\
\frac{1}{2}(C_{11} + C_{22} - \sqrt{4C_{12}^2 - (C_{11} - C_{22})^2}) &> 0 \\
C_{66} &> 0
\end{align*}  

(59)

6.1.3. Hexagonal lattice conditions

\begin{align*}
C_{11} - C_{12} &> 0 \\
C_{11} + C_{12} &> 0 \\
C_{66} &> 0
\end{align*}  

(60)

6.1.4. Oblique lattice conditions

\begin{align*}
C_{11} &> 0 \\
C_{11}C_{22} &> C_{12}^2 \\
\det(C_{ij}) &> 0
\end{align*}  

(61)
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]:

\[ E(\eta) = E_0 + \frac{2B_0V_0}{(B_0 - 1)^2} \times (2 - (5 + 3B'_0(\eta - 1) - 3\eta)e^{3(B'_0-1)(\eta-1)/2}) \] (62)

where, \( \eta = (V/V_0)^{1/3} \), \( V_0, E_0, B_0, \) 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]:

\[ E(V) = E_0 + \frac{9}{8}B_0V_0\left[(V_0/V)^{2/3} - 1\right]^2 + \frac{9}{16}B_0V_0(B'_0 - 4) \times \left[(V_0/V)^{2/3} - 1\right]^3 + \sum_{n=4}^{N} \gamma_n \left[(V_0/V)^{2/3} - 1\right]^n \] (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 Td-MoTe2 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]:

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

(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.

$$E(\eta) = E_0 + \frac{9 B_0 V_0}{16} \left( \eta^2 - 1 \right)^2 \left( 6 + B_0' \left( \eta^2 - 1 \right) - 4 \eta^2 \right)$$

(65)

where, $\eta = (V/V_0)^{1/3}$ 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 \([E_0 \text{ (eV)}, B_0 \text{ (GPa)}, B_p \text{ (GPa)}, V_0 \text{ (Angstroms}^3)]\) along with Mean Squared Error (MSE)

\[
\begin{align*}
\text{Vinet} & \colon [-67.074 \ 0.259 \ 8.232 \ 297.941] \\
\text{Vinet (MSE)} & \colon 8.377e-06 \\
\text{Birch} & \colon [-67.075 \ 0.251 \ 8.939 \ 298.112] \\
\text{Birch (MSE)} & \colon 6.999e-06 \\
\text{Murnaghan} & \colon [-67.074 \ 0.278 \ 6.698 \ 297.574] \\
\text{Murnaghan (MSE)} & \colon 1.385e-05 \\
\text{Birch-Murnaghan} & \colon [-67.076 \ 0.128 \ -28.930 \ 301.014] \\
\text{Birch-Murnaghan (MSE)} & \colon 4.219e-07
\end{align*}
\]

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

Fig. 4. The output of an EOS analysis with MechElastic. MechElastic 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 MechElastic. With this, an initial second order parabolic \(\text{numpy} [112]\) 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 \text{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 \(v\text{lim}\) 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, MechElastic 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:

```python
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.,

\[
H = U + PV.
\]

(66)

It is a measure of a system's capacity to do non-mechanical work and the ability to release heat. MechElastic’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. MechElastic 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 MechElastic 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 \(v\text{lim\_list}\) 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 \(v\text{lim\_list}\) exceeds the valid volume range. If not provided MechElastic 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 \(C2/m, I4/mcm, Im\_3m, Pm\_3m, \) and
Fig. 5. Enthalpy variation with pressure for five different phases of Bi as calculated using the plot_enthalpy_curves function.

Determining best model for each phase...
R\textsuperscript{3m} : Murnaghan
Pn\textsuperscript{3m} : Murnaghan
Im\textsuperscript{3m} : Murnaghan
I\textsubscript{4}m\textsubscript{cm} : Birch-Murnaghan
C\textsubscript{2\textit{\i}}m : Murnaghan

Possible transition paths for the provided volume range:
NOTE: Ordered with ascending enthalpy, i.e. topmost transition is the most probable.

- Pn\textsuperscript{3m} \rightarrow C\textsubscript{2\textit{\i}}m (at 1.639 GPa) \rightarrow Im\textsuperscript{3m} (at 6.067 GPa)
- R\textsuperscript{3m} \rightarrow C\textsubscript{2\textit{\i}}m (at 3.761 GPa) \rightarrow Im\textsuperscript{3m} (at 6.067 GPa)
- Pn\textsuperscript{3m} \rightarrow Im\textsuperscript{3m} (at 4.488 GPa)
- R\textsuperscript{3m} \rightarrow Im\textsuperscript{3m} (at 4.999 GPa)
- C\textsubscript{2\textit{\i}}m \rightarrow Im\textsuperscript{3m} (at 6.067 GPa)
- N\textsuperscript{3m} \rightarrow Pn\textsuperscript{3m} (at 7.753 GPa)

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\textsuperscript{3m}, 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:

```python
from mechelastic import EOS
eos = EOS()
infiles = ['R-3m', 'Pn-3m', 'Im-3m', 'I4_mcm', 'C2_m']
natoms = [6, 1, 2, 9, 4]
eos.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 MECHESlastic 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 MECHESlastic welcome screen, warning messages, and formatting of the matrices with proper precision.
2. core: This is the heart of the MECHESlastic 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.)

MechElastic Library

- comms
  - printer (verbosity)
- core
  - ElasticProperties
  - ELATE
  - Structure
- eos
  - EOS
- parsers
  - ABINIT
  - VASP
  - QE
- tests
  - Ductile
  - Eigenvals
  - Stability
  - Symmetry
- util
  - constants
  - crystalutils
  - elements

- Elastic and mechanical properties
- Elastic anisotropy
- EOS analysis
- ELATE analysis

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/.
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. LiNbO3 MechElastic implementation

In this section we demonstrate the implementation of using MechElastic on the example structure LiNbO3. 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           |         |

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| 26.608| 26.745|
| Shear modulus (GPa) | 87.803| 81.528| 84.712|
| Young’s modulus (GPa)| 45.933| 41.209| 43.571|
| Poisson’s ratio     | -0.046| -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  | 107.334 | 1.608      |
| Min Axis:            | (-0.009, 0.473, 0.35) | (0.0, 0.786, 0.619) |
| Max Axis:            | (0.0, 0.786, 0.619) | (-0.009, 0.473, 0.35) |
| Linear Compression (TPa^-1) | 10.620 | 13.486 | 1.270 |
| Min Axis:            | (0.0, 0.0, 1.0) | (0.787, 0.787, 0.0) |
| Max Axis:            | (0.787, 0.787, 0.0) | (0.0, 0.0, 1.0) |
| Shear Modulus (GPa)  | 1.014   | 2.051   | 1.014      |
| Min Axis:            | (-1.0, -0.0, 0.0) | (0.536, 0.311, 0.785) |
| Max Axis:            | (0.536, 0.311, 0.785) | (-1.0, -0.0, 0.0) |
| Poisson’s Ratio      | -0.224  | 0.166   | Binfin;    |
| Min Axis:            | (0.608, 0.399, -0.959) | (-0.452, 0.261, 0.023) |
| Max Axis:            | (-0.452, 0.261, 0.023) | (0.608, 0.399, -0.959) |
| Second Min Axis:     | (-0.495, 0.860, 0.002) | (-0.584, -0.863, -0.003) |
| Second Max Axis:     | (-0.584, -0.863, -0.003) | (-0.495, 0.860, 0.002) |

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.
that MechElastic provides, as shown in Fig. 11. The last part of the MechElastic 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 MechElastic-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 MechElastic-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 plotting 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 MechElastic, 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, MechElastic 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 MechElastic parses the elastic tensor data generated from the VASP, ABINIT, and Quantum Espresso packages (in future we plan to support others DFT packages). Notably, MechElastic 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. MechElastic is specifically designed to be easily integrated in high throughput calculations. From the inputted elastic tensor data, the current version of MechElastic 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

We acknowledge the support from the XSEDE facilities which are supported by the National Science Foundation (NSF) under grant number ACI-1053575. The authors also acknowledge the support from the Texas Advanced Computer Center (Stampede2) funded by the NSF through award ACI-1540931 and the Pittsburgh supercomputer center (Bridges), and the support from the Super Computing System (Thorny Flat) at WVU that is funded in part by the NSF Major Research Instrumentation Program (MRI) Award #1726534. The research presented here was primarily supported by the U.S. Department of Energy (DOE), Office of Science, Basic Energy Sciences (BES), under Award DOE DE-SC0021375 (code extension dynamical mean field theory calculations and other DFT packages), and by the NSF under Awards # DMREF-NSF 1434897, and NSF OAC-1740111 (development of the basic package and the graphical interface).

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.

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

(A.1)

By default the pressure correction is enabled if MechElastic 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–B877, 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, C. Ceder, Calphad 29 (3) (2005) 163–211, https://doi.org/10.1016/j.calphad.2005.01.002.
[5] K. Gonze, J.-M. Beuken, R. Caracas, F. Detaux, M. Fuchs, G.-M. Bignanese, 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)00032-7.
[6] K. Gonze, B. Amadon, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Brunetah, D. Caliste, R. Caracas, M. Côrte, T. Deutsch, L. Genovese, P. Ghosez, M. Giontoma, S. Goeckeler, D. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. Oliveira, G. Osida, Y. Pouillon, T. Rangel, G.-M. Bignanese, 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] K. Gonze, F. Jollet, F.A. Araujo, D. Adams, B. Amadon, T. Appelssoncourt, C. Audouze, J.-M. Beuken, J. Biedert, A. Bokhanchuk, et al., Comput. Phys. Commun. 205 (2016) 106–131, https://doi.org/10.1016/j.cpc.2016.04.003.
