



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF MATERIALS SCIENCE AND ENGINEERING

ÚSTAV MATERIÁLOVÝCH VĚD A INŽENÝRSTVÍ

**COMPARISON OF DIFFERENT APPROACHES IN
AB INITIO MODELING OF MECHANICAL AND
MAGNETIC PROPERTIES OF MATERIALS**

POROVNÁNÍ RŮZNÝCH PŘÍSTUPŮ V AB INITIO MODELOVÁNÍ MECHANICKÝCH A
MAGNETICKÝCH VLASTNOSTÍ MATERIÁLŮ

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. Martin Čavojec

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Martin Zelený, Ph.D.

BRNO 2022

Assignment Master's Thesis

Institute: Institute of Materials Science and Engineering
Student: **Bc. Martin Čavojec**
Degree programme: Materials Engineering
Branch: no specialisation
Supervisor: **Ing. Martin Zelený, Ph.D.**
Academic year: 2021/22

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Master's Thesis:

Comparison of different approaches in ab initio modeling of mechanical and magnetic properties of materials

Brief Description:

Theoretical calculations of electronic structure based on ab initio methods belong today between well-established tools in material engineering. Because they arise only from fundamental postulates of quantum-mechanics, they do not need any input experimental data. Therefore, these methods can effectively predict properties of materials, which is difficult or expensive to prepare experimentally. There are many different approaches and approximations for description of interatomic interactions which can be used within these methods. Some of them can significantly decrease computational time but they also decrease precision of calculation. The most common approaches for description of atomic orbitals are plane waves or localized wave functions.

Master's Thesis goals:

The goal of this work is to estimate, how big is the influence of chosen approach for description of interatomic interaction on calculated mechanical and magnetic properties, e.g. tensor of elastic constants. A part of the work will be focused also on creation of own code for calculation of elastic constant with help of different ab initio program packages.

Recommended bibliography:

MARTIN, R. P. Electronic Structure: Basic theory and practical methods, Cambridge University Press, Cambridge, 2004.

TADMOR, E. B. and R. E. MILLER. Modeling materials: continuum, atomistic, and multiscale techniques, Cambridge University Press, New York, 2011.

Deadline for submission Master's Thesis is given by the Schedule of the Academic year 2021/22

In Brno,

L. S.

prof. Ing. Ivo Dlouhý, CSc.
Director of the Institute

doc. Ing. Jaroslav Katolický, Ph.D.
FME dean

Abstract

This thesis studies the influence of the quantum-mechanical description of interatomic interactions on the results of ab initio calculations of magnetic and mechanical properties of Fe, γ -TiAl and Ni₂MnGa. Two methods for the description of interactions were examined, the projector-augmented wave method and the linear combination of atomic orbitals, implemented in ab initio codes VASP and SIESTA, respectively. The results of the dependence of the magnetic moments on the volume or tetragonal deformation paths indicate relative comparability across these two methods. However, calculated elastic constants also point to the unsuitability of the SIESTA software for calculations of elastic constants using the stress-strain method. Additionally, as a part of the thesis, a script for the calculation of elastic constants was created in the programming language Python.

Keywords

ab initio, projector-augmented waves, linear combination of atomic orbitals, magnetic moments, elastic constants, Python, iron, TiAl, Ni₂MnGa

Abstrakt

Tato diplomová práce se zabývá zkoumáním vlivu kvantově-mechanického popisu mezi-atomárních interakcí na výsledky ab initio výpočtů magnetických a mechanických vlastností Fe, γ -TiAl a Ni₂MnGa. Zkoumány byly dvě metody pro popis interakcí, jmenovitě metoda projektovaných přidružených vln a metoda lineární kombinace atomových orbitalů, které jsou implementovány v ab initio softwarech VASP a SIESTA. Hodnoty magnetických momentů v závislosti na objemových a tetragonálních deformačních drahách poukazují na relativní srovnatelnost napříč těmito metodami. Avšak výsledky elastických konstant poukazují na nevhodnost ab initio softwaru SIESTA pro výpočet elastických konstant metodou napětí-deformace (stress-strain metoda). Součástí praktické části této diplomové práce je vytvoření skriptu pro výpočet elastických konstant v programovacím jazyku Python.

Klíčová slova

ab initio, metoda projektovaných přidružených vln, metoda lineární kombinace atomových orbitalů, magnetické momenty, elastické konstanty, Python, železo, Ni₂MnGa, TiAl

Rozšířený abstrakt

Metody ab initio jsou progresivní přístupy teoretického studia základních vlastností, termodynamické stability, nebo elektronové struktury materiálů. Jsou založeny na základních postulátech kvantové mechaniky a kvantové fyziky. Používají kvantově-mechanický model pro popis interakce elektronů. Ab initio výpočty zkoumají energetické stavy systémů, co může poskytnout informace o strukturní stabilitě zkoumaných materiálů, i bez jakýchkoli empirických dat.

Princip ab initio výpočtů je založen na řešení Schrödingerovi rovnice, která definuje interakce elektronů a jader atomů. Jelikož potenciál působící na elektrony je obyčejně jen funkce pozice a ne času, časově nezávislá verze Schrödingerovi rovnice je běžně použita. Schrödingerova rovnice je schopna popsat systémy s tisíci atomy a jejich elektrony, avšak analytické řešení Schrödingerovi rovnice je možné jen pro jeden atom vodíku, jelikož složitost výpočtů roste exponenciálně s počtem elektronů. Proto je potřeba použít aproximaci této rovnice. Jedna s nejdůležitějších aproximací je Born-Oppenheimerova (adiabatická) aproximace, která uvažuje že pohyb elektronů je mnohonásobně rychlejší než pohyb jader atomů, z důvodu rozdílu hmotností.

Metody založené na vlnových funkcích jsou považovány za velmi dobře definované teoretické přístupy v ab initio výpočtech. Jedním z nich je Hartree-Fockova metoda, která užívá variačních teorémů pro získání přibližného řešení. Tato metoda dosahuje uspokojivých řešení, nicméně má i několik nedostatků. Proto se používá komplementární přístup k Hartree-Fockově metodě, teorie funkcionálu hustoty (density functional theory, DFT). Tento přístup je založen na Hohenberg-Kohnových teorémech, které tvrdí že energie elektronů v základním stavu je určena elektronovou hustotou. První Hohenberg-Kohnův teorém, existenční, říká že totální energie systému s mnoha elektrony je jedineční funkcionál elektronové hustoty. Druhý Hohenberg-Kohnův teorém, variační princip, říká že je možné získat základní stav variacemi elektronové hustoty a externího potenciálu. Výhoda elektronové hustoty je její závislost jen na 3 proměnných namísto mnoha-elektronové vlnové funkce, která je závislá na $3N^{\text{el}}$ proměnných.

V rámci DFT se kinetická energie interagujících elektronů dělí na dvě části. První část je uvažuje jako neinteragující, co formuje hlavní příspěvek kinetické energie, a druhá část obsahuje opravu z důvodu zvážení jejích interakcí. Toto je součástí výměnné a korelační energie. Toto tvrzení umožňuje převod rovnice obsahující N elektronů na systém jedno-elektronových Kohn-Shamových rovnic, kde pohyb elektronů v mnoha elektronovém systému je popsán jako pohyb neinteragujících elektronů v efektivním potenciálu, generující stejnou hustotu jako systém interagujících částic. Systém Kohn-Shamových rovnic je řešen self-konzistentním přístupem, protože každý potenciál je závislý na elektronové hustotě, která je určena jedno-elektronovou vlnovou funkcí, což je řešení Kohn-Shamových rovnic.

Elektronová hustota je potřebná k správné aproximaci výměnných a korelačních potenciálů, které nemají explicitní formulaci jen implicitní. Jeden z nejjednodušších přístupů je lokální aproximace hustoty (Local Density Approximation, LDA). Příspěvky výměnné a

korelační energie v libovolném bodě systému jsou určeny jako odpovídající výměnná a korelační energie homogenního elektronového plynu o stejné elektronové hustotě. Metoda rozšiřující LDA je generalizovaná aproximace gradientu (Generalized Gradient Approximation, GGA), která využívá lokální elektronovou hustotu a její gradient v blízkosti pozorovaného bodu.

Bázové funkce hrají důležitou roli v DFT pro matematický popis jedno-elektronových vlnových funkcí. Bázové funkce mohou být složeny z atomových orbitalů, které jsou vhodnější pro molekuly, materiály s vysoce lokalizovanými elektrony, nebo rovinných vln, které jsou běžně používány k popisu krystalů, materiálů s delokalizovanými elektrony. Vysoce lokalizované bázové funkce trpí nedokonalým popisem intersticiálních prostor, zatímco u delokalizovaných bázových funkcí vzniká problém v blízkosti jádra atomu v důsledku oscilací, způsobených vysokou koncentrací elektronů v blízkosti jádra.

Metoda projektovaných přidružených vln (projector-augmented wave, PAW) je založena na principu superpozice k popisu skutečné vlnové funkce. Metoda rozděluje skutečnou vlnovou funkci na pseudo-vlnovou funkci pro valenční elektrony vzdálené od jádra atomu a parciální pravou vlnovou funkci pro valenční elektrony v blízkosti jádra atomu. Většina PAW metod využívá aproximaci zmraženého jádra (frozen core approximation), která ale není inherentní pro PAW metodu jako takovou.

Naproti tomu metoda lineární kombinace atomových orbitalů (Linear Combination of Atomic Orbitals, LCAO) používá rozvoj Kohn-Shamovy pseudo-vlnové funkce na množinu atomárních orbitalů. Několik rozdílných přístupů bylo vyvinutých k dosažení přiměřeně přesné interpretace pomocí malých bázových funkcí, konkrétně Gaussovské orbitály (GTO), Slaterovské orbitály (STO), číselné atomové orbitály (NAO) a pseudo-atomární orbitály (PAO).

Strukturní relaxace nebo strukturní optimalizace je proces, který má za úkol najít rovnovážný stav dané struktury. Dosáhnout rovnovážný stav znamená dosažení minima totální energie daného systému s žádnými silami působícími na atomy struktury a bez napětí působící na výpočetní buňku. To znamená, že proces strukturní relaxace hledá energeticky nejvýhodnější pozice atomů vzhledem k působícím silám a napětím. Působící síly a napětí musí být tedy známy pro proces relaxace, a proto se využívá Hellman-Feynmanův teorém, který umožňuje výpočet působících sil a napětí jako derivaci energie, anebo jako derivaci Hamiltoniánu v případě DFT. Existuje několik algoritmů pro výpočet minima energie dané struktury. Tyto algoritmy závisí na počátečním odhadu struktury, jelikož je zcela možné dosáhnout řady různých energetických minim. Jeden z takových přístupů je metoda nejstrmějšího sestupu (steepest descent, SD), která je relativně jednoduchá z pohledu implementace, ale může být méně přesná, nebo pomalá ke konvergování. Princip spočívá v přemísťování atomů na základě působících sil při každé iteraci ve směru síly. Pokročilejší přístup je metoda konjugovaným gradientem, která je založena na vývoji optimální metody pro minimalizaci obecné kvadratické energetické funkce.

Elastické konstanty jsou používány pro popis elasticity, základní vlastnosti krystalických materiálů, která popisuje makroskopickou odezvu krystalu na vnější napětí. Přímou souvisí

s tvrdostí a pevností materiálu. Elasticita dovoluje analýzu několik mikroskopických procesů, jako například interakce dislokací, nebo propagace mikrotrhlin. Elasticita také představuje důležité kritérium pro mechanickou stabilitu krystalu. Pro popis elastických konstant se používá Hookův zákon v tenzorové podobě. Jelikož takový zápis je zbytečně komplikovaný, používá se zjednodušená verze podle Voigtovi notace, která převede tenzor čtvrtého řádu na tenzor druhého řádu. Počet nezávislých elastických konstant závisí na symetrii studovaného krystalu, kde triklinická struktura má 21 nezávislých elastických konstant, tetragonální struktura má 6 a kubická struktura má jen 3 nezávislé elastické konstanty. Výpočty elastických konstant jsou obvykle založeny na energii zdeformovaného krystalu, což je výpočetně velmi náročný přístup. Naproti tomu relativně inovativní metoda napětí-deformace (stress-strain metoda), popsaná ve studii, publikované R. Yu et al., je výpočetně relativně nenáročná a rychlá. Metodě stress-strain je také věnována značná část této diplomové práce.

Železo existuje v konfiguracích BCC a FCC a má několik magnetických fází. Tato práce je zaměřena především na analýzu základního feromagnetického stavu železa BCC. Antiferomagnetický stav FCC železa proto není uvažován.

Slitiny na bázi aluminidu titanu (TiAl) patří do skupiny materiálů klasifikovaných jako intermetalika. Intermetalika využívají principu uspořádané krystalografické struktury, která obsahuje kombinaci různých kovů, jež vzniká omezenou rozpustností slitin. Takové slitiny mají obvykle odlišnou strukturu a vlastnosti na rozdíl od jejich výchozích materiálů. Intermetalika mají uspořádanou krystalografickou strukturu s dlouhým dosahem, což má za následek zajímavé mechanické vlastnosti i při vyšších teplotách. γ -TiAl krystalizuje v tetragonální prostorové grupě $P4/mmm$ s plošně centrovanou tetragonální strukturou typu CuAu, známou jako $L1_0$. Strukturu TiAl lze považovat za střídající se posloupnost vrstev titanu a hliníku podél osy c .

Ni_2MnGa je feromagnetická slitina s tvarovou pamětí, krystalizující v kubické prostorové grupě $Fm\bar{3}m$, s kubickou strukturou typu Cu_2MnAl , známou jako $L2_1$. Slitina patří do kategorie Heuslerových slitin. Heuslerovy slitiny jsou obecně ternární polovodičové nebo kovové materiály s chemickým složením X_2YZ . Heuslerovy slitiny vyvolávají značný zájem, především díky jejich pozoruhodné elektronové struktuře, která má za následek jedinečné magnetické vlastnosti. Vysokoteplotní austenitická fáze Ni_2MnGa je při pokojové teplotě kubická, a feromagnetická. Curieho teplota, a tedy teplota feromagnetického přechodu je přibližně $T_c \approx 376 K$. Stechiometrická Ni_2MnGa prochází martenzitickou transformací (transformací z vysokoteplotní kubické, austenitické, struktury na nízkoteplotní tetragonální, martenzitickou, strukturu) při teplotě $T_m \approx 210 K$. Bylo pozorováno, že tato strukturální transformace je reverzibilní, což umožňuje chování slitiny s tvarovou pamětí.

Výpočty závislosti magnetických momentů na objemových a tetragonálních deformačních drahách, a výpočty elastických konstant, provedeny pomocí stress-strain metody popsané ve studii, publikované R. Yu et al., byli provedeny pomocí metod PAW a LCAO, které jsou implementované v ab initio softwarech VASP a SIESTA

Výsledky těchto výpočtů porovnávají vliv rozdílného kvantově-mechanického popisu mezi-atomárních interakcí. Bylo zjištěno, že rozdíl mezi metodami na závislost magnetických momentů na objemové deformační dráze je relativně minimální. Patrné rozdíly jsou poznatelné jedině ve velmi nízkých objemech, kde výsledky ze softwaru SIESTA jsou lehce nadhodnocené v porovnání s výsledky ze softwaru VASP. Ve výpočtech tetragonálních deformačních drah železo ukázalo, že SIESTA preferuje tzv. high spin state, který se vyznačuje vyššími magnetickými momenty, zatímco VASP automaticky nachází tzv. low spin state, který je energeticky značně výhodnější. Výsledky tetragonální deformační dráhy pro Ni_2MnGa také ukázali, že SIESTA udává značně energeticky stabilnější NM martenzitickou fázi.

Výsledky elastických konstant spočtené metodou stress-strain ukazují, že elastické konstanty pro Ni_2MnGa a $\gamma\text{-TiAl}$ spočtené pomocí softwarů VASP a SIESTA jsou relativně srovnatelné s experimentálními, nebo vypočtenými hodnotami elastických konstant pro oba materiály, s poměrně minimálními odchylkami. Avšak hodnoty elastických konstant pro železo vypočtené softwarem SIESTA vykazují značné odchylky v porovnání s experimentálními hodnotami.

Z vypočtených tenzorů tuhosti byli dále spočteny prostorové rozložení Youngova modulu. Toto ukázalo značné nesrovnalosti napříč metodami, a to zejména u železa. Tyto výsledky poukazují na nevhodnost softwaru SIESTA pro výpočet elastických konstant pomocí metody stress-strain.

Bibliographic citation

ČAVOJEC, Martin. *Porovnání různých přístupů v ab initio modelování mechanických a magnetických vlastností materiálů* [online]. Brno, 2022 [cit. 2022-05-24]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140606>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav materiálových věd a inženýrství. Vedoucí práce Ing. Martin Zelený Ph.D.

Declaration

I declare that the master's thesis *Comparison of different approaches in ab initio modelling of mechanical and magnetic properties of materials* has been written on my own, according to the objectives set by the supervisor *Ing. Martin Zelený, Ph.D.*, using the literature source listed in references.

20.05.2022 in Brno

.....
Čavojec Martin

Acknowledgement

I would like to thank my supervisor *Ing. Martin Zelený, Ph.D.* for his advice, supervision and patience while writing this thesis. Furthermore, I would like to thank IT4Inovations facilities, supported through the project e-INFRA CZ (ID:90140), for providing computational resources.

Contents

1	Introduction	1
2	Thesis objectives	2
3	Ab Initio methods.....	3
3.1	Schrödinger equation.....	3
3.2	Density Functional Theory (DFT).....	4
3.3	Projector–Augmented Wave Method (PAW method).....	6
3.4	Linear Combination of Atomic Orbitals Method (LCAO Method).....	8
3.5	Structure relaxation, Hellmann-Feynman theorem.....	9
4	Elastic constants.....	11
4.1.1	Stress-strain method.....	13
5	Materials	15
5.1	Fe.....	15
5.2	TiAl	15
5.3	Ni ₂ MnGa.....	17
6	Simulation details.....	22
6.1	Atomic simulation environment	22
6.1.1	ASE computational details	22
6.2	Vienna Ab initio Simulation Package	23
6.2.1	VASP computational parameters.....	23
6.3	Spanish Initiative for Electronic Simulations with Thousands of Atoms.....	24
6.3.1	SIESTA computational parameters	24
6.4	Common calculation parameters.....	25
6.5	Python	26
6.5.1	Scripts for calculation of elastic constants.....	26
7	Results and discussion	27
7.1	Energy-Volume (E-V) curves and dependence of magnetic moments on cell volume.....	27
7.1.1	Fe.....	27
7.1.2	TiAl	28
7.1.3	Ni ₂ MnGa.....	29
7.2	Tetragonal deformation paths	31
7.2.1	Fe.....	32
7.2.2	TiAl	34
7.2.3	Ni ₂ MnGa.....	35
7.3	Elastic constants.....	36

7.3.1	Fe.....	36
7.3.2	γ -TiAl.....	40
7.3.3	Ni ₂ MnGa.....	41
8	Conclusion.....	44
	References.....	45
	Symbols and abbreviations.....	52
	Appendix.....	53
	Appendix 1: Script prep_nonDiag.py.....	53
	Appendix 2: Script results_Cij_stress-strain.py.....	57

1 Introduction

Ab initio methods are progressive tools for the theoretical study of materials. Principles of ab initio calculations are based on basic postulates of quantum physics and mechanics, namely a solution of the Schrödinger equation which defines interactions of electrons and atom nuclei. Ab initio calculations do not require any recorded empirical data. Therefore, these calculations can be performed for any kind of molecular species, which allows us to examine fundamental properties, structural stability, thermodynamic stability, or electron structure of materials, which could be otherwise incredibly difficult to produce and prepare. [1, 2]

Density functional theory (DFT) is a density-based theoretical framework within the ab initio calculations arising from the Hohenberg-Kohn theorems, which state that the ground-state electronic energy is determined by the electron density. Electron density is necessary to approximate exchange and correlation potentials, which can be accomplished through the local density approximation (LDA), or its expansion generalized gradient approximation (GGA). Wave-function of a single-electron system is an approximation by the linear combination of basis functions. Basis functions can be comprised of atomic orbitals, characterized by the highly localized electrons, or by plane waves, characterized by the delocalized electrons. Highly localized basis functions suffer from an imperfect description of interstitial space, while with delocalized basis functions, the problem arises in the vicinity of the atom nucleus due to oscillations, caused by the high concentration of electrons around the nucleus. This can be corrected by employing effective core potentials or pseudo-potentials, which replace the wave-function for electrons in the vicinity of the atom nucleus.

Elastic constants are a system for the description of elasticity, a fundamental property of crystalline materials, describing a macroscopic response of a crystal to external straining. It is possible to calculate elastic constants using ab initio methods or measure them experimentally. Elastic constants are usually described by the Hook's law in its tensor form, a fourth-order tensor of stiffness, which then can be further simplified by Voigt notation into a second-order tensor of stiffness. The number of independent elastic constants is entirely dependent on the symmetry of the examined structure, where triclinic structures have twenty-one, tetragonal structures have six, and cubic structures only have three independent elastic constants.

This thesis examines two different approaches for the description of wave function within the DFT, and theoretical approaches for calculations of elastic constants, which are further implemented in a python script. Results then examine the influence of these two approaches on elastic constants and magnetic moments as a function of volume and tetragonal deformation paths. These parameters are calculated for three chosen materials, which are briefly discussed in the theoretical part of this thesis.

2 Thesis objectives

The goal of this thesis is to examine the influence of two different ab initio approaches to describe interatomic interactions on calculations of mechanical and magnetic properties of materials. The interactions are described using localized and delocalized wavefunctions. The extent of the influence will be examined on three materials, *Fe*, γ -*TiAl* and *Ni₂MnGa*, for which energy-volume curves, tetragonal deformation paths and elastic constants will be calculated. Additionally, a practical part of the thesis will include the creation of the script for the calculation of elastic constants in the programming language Python.

3 Ab Initio methods

Ab initio methods are progressive techniques for the theoretical study of fundamental properties, thermodynamic stability, or electron structure of materials. Ab initio methods are based on laws of quantum physics and use a quantum mechanical description of electron interactions. These methods examine the energy states of systems, which can provide information about the structural stability of studied materials, even without any recorded empirical data. This thesis employed one of the most common ab initio applications known as density functional theory (DFT).

3.1 Schrödinger equation

Principles of ab initio methods are based on the solution of the Schrödinger equation, defining interactions of atomic nuclei and electrons. As the potential affecting the electrons is generally just a function of position and not time, the time-independent variation of the Schrödinger equation is used: [3]

$$\left(U(r) - \frac{\hbar}{2m_e} \nabla^2 \right) \psi_e(r) = \varepsilon_0 \psi_e(r), \quad (3.1)$$

where $U(r)$ is the field of potential energy affecting the electron movement (including electron-nuclei and electron-electron interactions), \hbar is reduced Planck constant, m_e is the mass of the electron, ∇^2 is an operator, Laplacian, $\psi_e(r)$ is the wave function of the whole electron system and ε_0 is the energy of wave function. \hat{H} , a Hamiltonian, an operator comprising all the energy contributions of the system from the electron and nuclei interactions is then defined by the formula:

$$\hat{H} = U(r) - \frac{\hbar^2}{2m_e} \nabla^2, \quad (3.2)$$

Systems described by the Schrödinger equation can contain from tens to thousands of atoms and their respective electrons. However, the analytical solution of the Schrödinger equation is possible only for one atom of Hydrogen, since the complexity of the calculation grows exponentially with the number of electrons. Therefore, an approximation of this equation needs to be employed. One of the most important approximations is the nonrelativistic Born-Oppenheimer approximation, which assumes that electrons move much faster than nuclei, due to major differences in mass.[4] This implies, that any changes in the potential energy of the atom nuclei initiated by the change of their position, causes an immediate reaction of the electrons. Thus, the motions of atom nuclei and the electrons are separated. As a result, it is then considered as a movement of electrons in a static potential generated by atom nuclei. [5]

3.2 Density Functional Theory (DFT)

Wave function-based approaches are considered a well-defined theoretical framework in ab initio methods. One such is Hartree – Fock method, which uses variational theorems to attain an approximate solution. It can provide satisfactory results for a number of properties, however, this method has also several drawbacks. One of them is the neglect of electron correlation on electronic structure calculations, which suggests that each electron accounts for the rest of them as a mean field. And the other is a high computational effort (calculations for a system containing N atoms scale as N^4) Therefore a complementary approach to Hartree-Fock is used. That being Density Functional Theory (DFT) based on Hohenberg-Kohn theorems, which state that the ground-state electronic energy is determined by the electron density.[5, 6]

The first Hohenberg-Kohn theorem, the proof of existence, states, that the total energy of a many-electron system is a unique functional of the electron density. The second Hohenberg-Kohn theorem, the variational principle, states, that it is possible to obtain a ground state by variations in electron density and external potential. [7]

Electron density function for general system is defined by the formula:

$$\rho(r) = N^{el} \int \psi^*(r, r_2, \dots, r_{N^{el}}) \psi(r, r_2, \dots, r_{N^{el}}) dr_2 dr_3 \dots dr_{N^{el}}, \quad (3.3)$$

Integral is not dependent on which electron is being integrated. Furthermore, in combination with anti-symmetry and Pauli's principle, it is possible to equate the density integral with the number of electrons N^{el} .

$$\int \rho(r) dr = N^{el}, \quad (3.4)$$

The main advantage of the electron density is its dependence on only 3 variables instead of multi-electron wavefunction, which is dependent on $3N^{el}$ variables.

Functional of total energy is dependent on external potential of static atom nuclei v_{ext} and electron density according to formula:

$$E[\rho(r)] = \int \rho(r) v_{ext}(r) dr + T_{en}[\rho(r)] + \iint \frac{\rho(r)\rho(r')}{|r - r'|} dr' dr + K[\rho(r)] \quad (3.5)$$

where second component, $T_{en}[\rho(r)]$, is kinetic energy of non-interacting electrons, the third component describes the electron repulsion, and $K[\rho(r)]$ is non-classical component of the equation describing main part of the exchange and correlation energy. [1, 6]

The kinetic energy of interacting electrons is divided into two parts. The first part considers them as non-interacting, which is the main contribution to the kinetic energy, and the second part, which encompasses a correction by considering their interactions. This is considered the part of the exchange and correlation energy. This proposition allows the conversion of equation containing N electrons to a system of one-electron Kohn-Sham equations, where the movement of electrons in the multi-electron system is described as a

movement of the non-interacting electrons in the effective potential, generating the same density as system of interacting particles. It is defined by the formula:

$$\left(-\frac{\hbar}{2m_e} \nabla^2 + V_{eff}(r) \right) \psi_i(r) = \varepsilon_i \psi_i(r), \quad (3.6)$$

where m_e is mass of the electron, $V_{eff}(r)$ is effective potential of non-interacting electrons, which includes electron interactions and external potential, $\psi_i(r)$ is wave function of one non-interacting electron and ε_i expresses the intrinsic value of the wave function. The effective potential is defined by three elements according to the formula:

$$v_{eff}(r) = v_{ext}(r) + e^2 \int \frac{\rho(r')}{|r - r'|} dr' + \frac{\delta K[\rho(r)]}{\delta \rho(r)}, \quad (3.7)$$

where first component, $v_{ext}(x)$, contains external potential, the second component contains Hartree interaction potential, and the third component contains exchange and correlation potentials. In case of Kohn-Sham equations the contribution to the non-classical electron interaction is an approximation through the exchange and correlation functional, which is dependent on the electron density $\rho(r)$. Electron density of non-interacting electrons in the N-particle fictitious system is defined according to the formula:

$$\rho(r) = \sum_i^N |\psi_i(r)|^2, \quad (3.8)$$

Kohn-Sham equations are a mathematical description based on two Hohenberg-Kohn theorems, which relate to any system consisting of electrons moving under the influence of an external potential. The system of Kohn-Sham equations has to employ a self-consistent approach, as each potential is dependent on electron density $\rho(r)$, which is determined by a single-electron wavefunction, the solution of the Kohn-Sham equations. To speed up the convergence of self-consistent calculations the procedure called density mixing is employed, which combines the input and output densities at each iteration cycle to estimate a new input density. [8]

The electron density is necessary to properly approximate the exchange and correlation potentials, which do not have explicit formulation only implicit formulation. One of the simplest approaches to exchange and correlation energy is Local Density Approximation (LDA). The contribution of exchange and correlation energy in any given point of the system is determined as the corresponding exchange and correlation energy of homogenous electron gas of the same electron density. Additionally, the method augmenting the LDA is Generalized Gradient Approximation (GGA), which is utilizing local electron density and its gradient in the proximity of the observed point. [9]

The value of total energy calculated using DFT represents only the internal energy of the system in its fundamental state, which includes all the contributions of the kinetic energy of the electrons, energy of electron interactions with the external potential of atom nuclei and mutual electron interactions. It carries no further information about the state of the system

in the form of temperature or pressure of the system. Total energy of the system could be defined by the formula:

$$E_0 = G - \sum_i F_i + pV, \quad (1.9)$$

where E_0 is total energy of given system, G is Gibbs energy, F_i are corresponding temperature dependent contributions to the overall energy of the system, p is pressure of the system and V is volume of the system. [10]

Wavefunction of single-electron system is an approximation by linear combination of basis functions:

$$\psi_i = \sum_{n=1}^N C_{In} \varphi_n, \quad \text{for } I = 1 \dots N, \quad (1.10)$$

Since basis functions play an important role in the calculations, selecting a correct methodology is necessary. Basis function can be comprised of atomic orbitals, which are more suitable for molecules, or plane waves, which are more commonly used to describe crystal structures. While atomic orbitals basis functions are more appropriate for materials with localized electrons, with plane-wave basis functions problems arise near the atom nucleus, where oscillations occur, due to the high concentration of electrons close to the nucleus. That is possible to correct by using effective core potentials, or pseudo-potentials, which replace the wave functions for electrons near the atom nucleus. Most of the DFT methods employ a frozen core approximation, in which the orbitals corresponding to the core electrons remain fixed throughout the calculation. [11] Likewise, the minimization of the Kohn-Sham DFT energy functional is carried out in the variational space of the valence electrons. When pseudo-potentials are used, it is possible to use plane waves to only describe valence electrons further from the atom nuclei, and to describe the valence electrons closer to the atom nuclei the approximative pseudo-wave functions are used. For a more accurate description of wave functions, the Projector-Augmented Waves method (PAW method) is used. This is one of the two methods used in this thesis to examine the mechanical and magnetic properties of three different materials.

3.3 Projector-Augmented Wave Method (PAW method)

Projector-Augmented Wave (PAW) method uses the principle of superposition to describe true wave function. The method divides true wave function into pseudo-wavefunction for valence electrons further from the atom nucleus and partial wavefunctions for valence electrons in the vicinity of the atom nucleus. True, one electron wavefunction ψ_{nk} is therefore derived as a linear transformation of pseudo-orbitals $\tilde{\psi}_{nk}$:

$$|\psi_{nk}\rangle = |\tilde{\psi}_{nk}\rangle + \sum_i (|\phi_i\rangle - |\tilde{\phi}_i\rangle) \langle \tilde{p}_i | \tilde{\psi}_{nk} \rangle, \quad (1.11)$$

where the first element, $|\tilde{\psi}_{nk}\rangle$ the pseudo-wave function correctly describes conditions further from the atom nucleus, the second and the third elements $\sum_i(|\phi_i\rangle - |\tilde{\phi}_i\rangle)$ are true partial wavefunction and partial pseudo-wavefunction respectively that describe conditions in the vicinity of the atom nucleus. Pseudo-orbitals $\tilde{\psi}_{nk}$, where nk are the band index and the k-point index respectively, are variational quantities, and they are expanded in plane waves. Pseudo-orbitals $\tilde{\psi}_{nk}$ are identical to true orbitals ψ_{nk} in the interstitial positions between PAW spheres. However, inside the PAW spheres the pseudo-orbitals are just computational tool and slightly inaccurate approximations to the true orbitals. [12, 13]

There is exactly one projector function $\langle\tilde{p}_i|$ for every pseudo-partial wave. These projector functions have to fulfil the condition $\sum_i|\tilde{\phi}_i\rangle\langle\tilde{p}_i|=1$ within volume Ω_R , so that the one-centre expansion $\sum_i|\tilde{\phi}_i\rangle\langle\tilde{p}_i|\tilde{\psi}\rangle$ of a pseudo-wave function is identical to the pseudo-wave function $|\tilde{\psi}_{nk}\rangle$ itself. Implying: [13, 14]

$$\langle\tilde{p}_i|\tilde{\phi}_j\rangle = \delta_{ij} \quad (3.12)$$

PAW method implemented in most of simulation packages can employ the frozen core approximation, which is not native to the PAW method. The pseudo-orbitals are expanded in the reciprocal space using plane waves according to formula: [14]

$$\langle r|\tilde{\psi}_{nk}\rangle = \frac{1}{\Omega^{1/2}} \sum_G C_{nkG} e^{i(G+k)\cdot r} \quad (3.13)$$

where Ω is volume of Wigner-Seitz cell. The all-electron partial waves ϕ_α are solutions to the radial Schrödinger equation for a non-spinpolarized reference atom at a specific energy ε_α and for a specific angular momentum l_α :

$$\langle r|\phi_\alpha\rangle = \frac{1}{|r - R_\alpha|} u_\alpha(|r - R_\alpha|) Y_\alpha(\widehat{r - R_\alpha}) = \frac{1}{|r - R_\alpha|} u_{l_\alpha\varepsilon_\alpha}(|r - R_\alpha|) Y_{l_\alpha m_\alpha}(\widehat{r - R_\alpha}) \quad (3.14)$$

where $\widehat{r - R_\alpha}$ describes that the spherical harmonics, Y , depends on the orientation, but not on the length of the vector $r - R_\alpha$. The radial component u_α is independent of m_α , an angular momentum quantum number, as the partial waves are determined from the spherical atom. The pseudo-waves $\tilde{\phi}_\alpha$ are equivalent to the all-electron partial waves outside a nuclei radius r_c and match continuously onto ϕ_α inside the nuclei radius:

$$\langle r|\tilde{\phi}_\alpha\rangle = \frac{1}{|r - R_\alpha|} \tilde{u}_\alpha(|r - R_\alpha|) Y_\alpha(\widehat{r - R_\alpha}) = \frac{1}{|r - R_\alpha|} \tilde{u}_{l_\alpha\varepsilon_\alpha}(|r - R_\alpha|) Y_{l_\alpha m_\alpha}(\widehat{r - R_\alpha}) \quad (3.15)$$

The nuclei radius r_c is usually approximately around half the nearest neighbour distance. [12, 13]

Figure 1 contains a graphical representation of the true wavefunction for the molecule of Cl_2 . The true wave function is black, the pseudo-wavefunction is blue, the true partial wavefunction is red and the partial pseudo-wavefunction is green. [15]

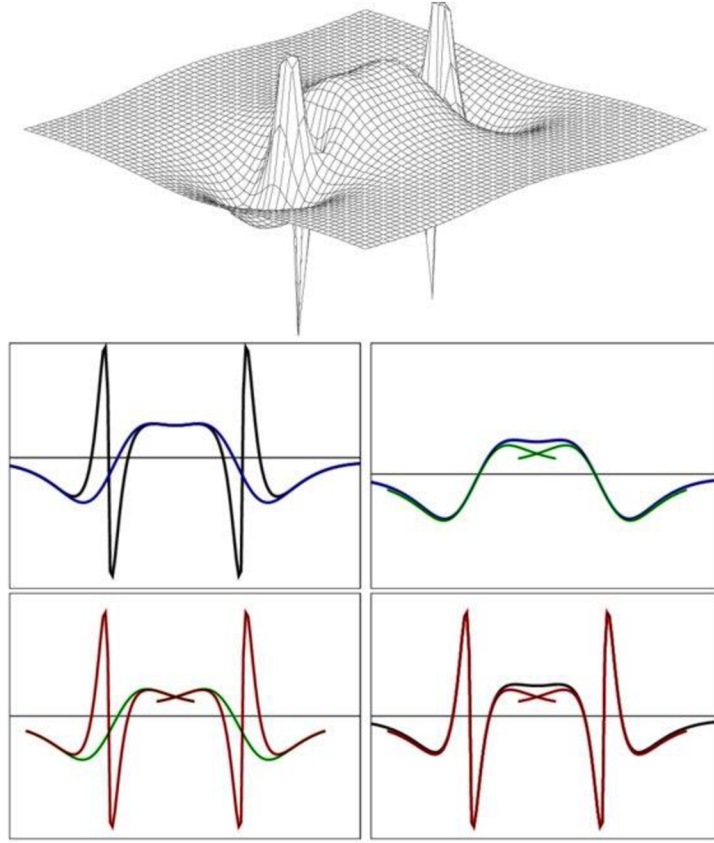


Figure 1- Diagram of projector-augmented wave function, where the true wavefunction is marked in black, pseudo-wavefunction is marked in blue, the true partial wavefunction is red and the partial pseudo-wavefunction is green [15]

3.4 Linear Combination of Atomic Orbitals Method (LCAO Method)

In contrast to the aforementioned plane-based methods, the DFT LCAO method uses the expansion of Kohn-Sham whole pseudo-wavefunctions $\tilde{\psi}_n(r)$ onto a set of atomic-like orbitals $\Phi_{nlm}(r)$: [16, 17]

$$\tilde{\psi}_n(r) = \sum_{\mu} c_{\mu n} \Phi_{\mu}(r) \quad (3.16)$$

where basis functions are constructed as products of numerical radial functions and spherical harmonics:

$$\Phi_{nlm}(r) = \Phi_{nlm}(r^a + R^a) = \varphi_{nl}(r^a) Y_{lm}(\hat{r}^a) \quad (3.17)$$

where R^a is the position of atom nucleus a , and $r^a = r - R^a$. This approximation uses the coefficients $c_{\mu n}$ as the variational parameters as opposed to the real space wavefunction.

By using small basis sets for atomic-like orbitals it is possible to reduce computational time or memory requirements at the possible cost of accuracy. A number of different approaches have been developed to attain reasonably precise interpretation using small basis sets: [18]

- Linear combinations of Gaussian-type orbitals (GTOs) are commonly used because the analytical evaluation of numerous expressions is thus possible.

- Slater-type orbitals (STOs) are another possibility, with a form $r^l e^{-\zeta r}$ resembling simple orbitals. Gaussian-based methods commonly approximate slater-type orbitals using a fixed linear combination of Gaussians in order to attain more orbital-like behaviour.
- Numerical atomic orbitals (NAOs) are represented on a radial grid numerically rather than analytically. Even though this increases the difficulty of the calculations and therefore makes calculations more time consuming, many radial integrations can be performed beforehand and tabulated. NAOs can be designed with any shape without any additional computational costs, they can be localized to avoid costly long-range interactions, therefore achieving better scaling.
- Another possible approach is basis localized pseudo-atomic orbital (PAOs), which are a variation of NAOs modified to use pseudo-potentials.

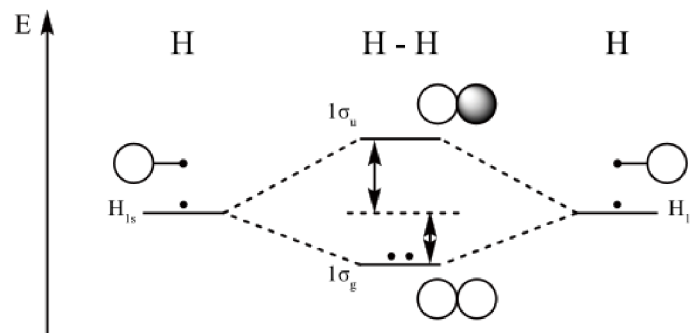


Figure 2 Diagram depicting linear combination of two atomic orbitals of hydrogen, which results in bonding orbitals (at the bottom) and anti-bonding orbitals (at the top) [19]

3.5 Structure relaxation, Hellmann-Feynman theorem

DFT calculations only provide energy at temperature $T = 0 K$ for a given atom distribution, therefore it is necessary to perform a structure relaxation to obtain the energy minimum of a given structure. Structure relaxation is an optimization process which aims to achieve the equilibrium state of a given structure. Reaching equilibrium means achieving minimal total energy of a given system with no forces acting on any of the atoms in the structure and no stresses acting on the computational cell. Thus, the process of structure relaxation is searching for the most energetically advantageous displacement of the atoms based on the acting forces and stresses.

Forces and stress tensors must be known for the structural relaxation process. Therefore, the Hellman-Feynman theorem is applied, which allows the calculation of forces and stress tensors acting on the atoms as a derivation of energy, and in the case of DFT as a derivation of Hamiltonian. [20]

The Hellman-Feynman theorem in DFT is applied through the derivation of wave functions for the given position of atoms r : [20, 21]

$$\frac{\delta E_\lambda}{\delta \lambda} = \left\langle \psi_\lambda \left| \frac{\delta \hat{H}_\lambda}{\delta \lambda} \right| \psi_\lambda \right\rangle \quad (3.18)$$

where E_λ is energy of the given state, \hat{H}_λ is operator Hamiltonian dependant on a parameter λ , and $|\psi_\lambda\rangle$ is an eigenfunction of the Hamiltonian dependant implicitly on a parameter λ .

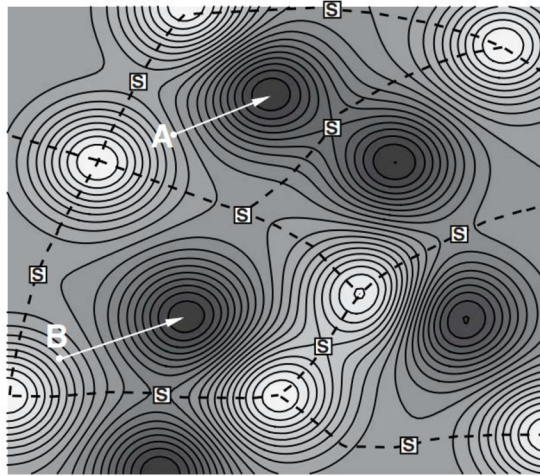


Figure 3 Schematic depiction of a 3N - dimensional energy landscape, where darker section are the energy minima, lighter sections are energy maxima and S indicate saddle points. A and B depict two different minima of a given structure [1]

There are a number of algorithms for calculations of the energy minimum of a given structure. These algorithms are dependent on the initial estimation of a structure, as it is entirely possible to reach a number of different energy minima from the initial estimate, as clearly shown in Figure 3. The steepest descent (SD) method is relatively easy to implement, however, it is possible that it might be less accurate, or slow to converge. The principle lies in the displacement of the atoms based on the acting forces at each iteration in the direction of the force. This creates a physical interpretation of the SD trajectory in the energy landscape corresponding to the steepest 'downhill' direction at any given point. A more advanced, conjugate gradient (CG), the method is considerably more efficient than the SD method. It is based on developing an optimal method for minimalizing a general quadratic energy function. [1]

4 Elastic constants

Elasticity, one of the fundamental properties of crystalline material, which describes the macroscopic response of crystal to external strain. It directly correlates to the hardness and strength of materials. The elasticity of materials allows the analysis of several microscopic processes, such as the interaction of dislocations or crack propagation. Additionally, elasticity provides an important criterion for the mechanical stability of a crystal.

One of the most used systems to describe the elasticity of materials during elastic deformation are elastic constants C_{ijkl} . These can be determined using Ab Initio methods, or experimentally. Elastic constants are a measurement of proportionality between strain and stress in a crystal, provided that the strain is not so large as to violate Hook's law. Elastic constants are also important constraints used to create interatomic potentials for large scale calculations. [22, 23]

To describe elastic constants of anisotropic materials the Hook's Law in its tensor form is used, which represents the relation between stress and strain tensors for small deformations:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \quad (4.1)$$

where σ_{ij} is stress tensor, ε_{kl} is strain tensor and C_{ijkl} is matrix of elastic constants. It states that each of the six components of the stress tensor is linear combination of six components of the strain tensor.

$$\{C_{ijkl}\} = \begin{pmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1123} & C_{1113} & C_{1112} \\ C_{2211} & C_{2222} & C_{2233} & C_{2223} & C_{2213} & C_{2212} \\ C_{3311} & C_{3322} & C_{3333} & C_{3323} & C_{3313} & C_{3312} \\ C_{2311} & C_{2322} & C_{2333} & C_{2323} & C_{2313} & C_{2312} \\ C_{1311} & C_{1322} & C_{1333} & C_{1323} & C_{1313} & C_{1312} \\ C_{1211} & C_{1222} & C_{1233} & C_{1223} & C_{1213} & C_{1212} \end{pmatrix} \quad (4.2)$$

C_{ijkl} , a fourth-order tensor of stiffness can be rewritten using Voigt notation to the form of C_{ij} , a second-order tensor. The principle lies in the conversion of i, j, k, l indices from the Latin-type indices i (1 ... 3) to the Greek-type indices α (1 ... 6) according to the Table 1: [24]

Latin indices	→	Greek indices
11	→	1
22	→	2
33	→	3
23 = 32	→	4
13 = 31	→	5
12 = 21	→	6

Table 1 Table of index conversions employed by the Voigt notation [24]

Elastic constants could be calculated by applying strain to a crystal, measuring the energy compared to the strain. They are then determined as a curvature of this function at a zero

strain. Specific strains used to determine elastic constants of a crystal are associated with a certain linear combination of elastic constants. The elasticity of single-crystal, characterized by a fourth-rank stiffness tensor, can be represented with a 6x6 matrix thanks to the Voigt notation: [24]

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} \quad (4.3)$$

where ε_i are components of the strain tensor, σ_i are components of stress tensor and C_{ij} are single-crystal elastic (stiffness) constants.

The number of independent elastic constants depends on the symmetry of the examined crystal structure, and the notation used to describe the elasticity matrix. In case the Voigt notation is not used, which means that C_{ijkl} is the fourth rank tensor and that the symmetry of the examined structure is non-existent it is possible to obtain 81 independent elastic constants. However, with increasing symmetry of structure and application of Voigt's notation, the number of elastic constants decreases. For example, in the case of a triclinic structure, the number of constants decreases to 21, and due to the plane symmetry of the monoclinic structure, it further decreases to just 13 constants. Additionally, while tetragonal structures have 6 constants, cubic systems only have 3 independent elastic constants, C_{11} , C_{12} and C_{44} . Therefore, matrix notation for cubic single-crystal systems looks like this:

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} \quad (4.4)$$

And matrix notation for tetragonal structure looks like this:

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} \quad (4.5)$$

To determine a full set of elastic constants experimentally, a single macroscopic-sized crystal is required. On the contrary, it is possible to obtain single-crystal elastic constants from ab initio calculations (as a non-experimental method). [25]

4.1.1 Stress-strain method

Most methods for the calculation of elastic constants are based on fitting total energies or stresses of properly deformed crystals. A stress-based method is much more efficient compared to the energy-based method, due to each deformation having six stress components, but only one energy component meant for the fitting. The downside of these methods is that crystals of different symmetries require specific strain patterns designed for a given crystal. Energies and stress components are then calculated for every strain pattern. Calculations are carried out gradually for every strain magnitude. The resulting stresses and energies are then fitted as a function of said strain magnitude, which gives us elastic constants. Such methods are computationally very challenging and tedious for anything but high symmetry crystals. Other methods based on linear response theory or interatomic force constants can be also used, however, such calculations are even more computationally challenging than stress and energy-based approaches.

Therefore, R. Yu et al. [26] presented in their study a much simpler stress-strain-based approach meant to significantly decrease computational time even for crystals with low symmetry. This method uses matrix notation, stated in (4.3), to describe single-crystal elastic constants. After calculating all the stresses for each strain, elastic constants are determined by linear least-squares fitting using singular value decomposition. The improvements of this method are based on the notions of high efficiency and universality of the used strains. Strains employed by this method are marked in Table 2. The idea is that each of the six stress components for a strain could be used as an independent ‘measurement’ of the elasticity of the crystal. [26]

	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_5	ϵ_6
U_1	1	2	3	4	5	6
U_2	-2	1	4	-3	6	-5
U_3	3	-5	-1	6	2	-4
U_4	-4	-6	5	1	-3	2
U_5	5	4	6	-2	-1	-3
U_6	-6	3	-2	5	-4	1

Table 2 Table of components of six typical linear-independent coupling strains in 10^{-3} [26]

Theoretically, the six stress and strain components can be described as vectors spanning a six-dimensional “stress and strain space” respectively. Subsequently, the elastic constants matrix C_{ij} is mapping between these two spaces. Strain vectors were named “universal linear-independent coupling-strains” (ULICS) and follow three principles:

- Strain vectors are linearly independent in the “strain space”.
- All elastic constants and stress vectors are coupled, as there are no zero elements in the strain vectors.
- Strain vectors can be applied to crystals with any symmetry. The only change comes with the symmetry of examined crystals, as more ULICSs are required for crystals with lower symmetry. This is noted in Table 3.

These three principles point to the efficiency of this method.

Crystal system	Number of ULICS required
Cubic	1
Tetragonal	2
Hexagonal	2
Rhombohedral	2
Orthorhombic	3
Monoclinic	5
Triclinic	6

Table 3 Number of ULICS required for seven crystal systems [26]

Stress and strain vectors need to be linearly independent, otherwise only partial sampling of all the subspaces would be realized. Therefore, angles between vectors are defined by the formula: [26]

$$\theta = \arccos\left(\frac{U_i \cdot U_j}{|U_i| \cdot |U_j|}\right) \quad (4.6)$$

Linear independence between ULICS is then marked in Table 4. A greater the angle means less linear dependence of the vectors.

	U_2	U_3	U_4	U_5	U_6
U_1	90.0°	90.0°	90.0°	90.0°	90.0°
U_2		90.6°	95.7°	68.7°	114.0°
U_3			86.9°	98.2°	98.2°
U_4				102.1°	80.5°
U_5					115.4°

Table 4 The angles between ULICS from Table 1 [26]

Table 3 indicates the high efficiency of the method as six or fewer strain vectors are required to obtain a full set of single-crystal elastic constants, which is contrary to the energy-volume fitting method which requires generally at least five volume strains. The efficiency is due to the linear independence of the strain vectors used, and strain vectors containing no zero elements. [21]

This method can be implemented and used by a number of different simulation packages, that provide stress tensor of a crystal, like VASP and SIESTA. In their study, R. Yu et al. calculated elastic constants of four different structures according to this stress-strain method using the projector-augmented wave method within the density functional theory implemented in VASP.

5 Materials

This thesis examines the mechanical and magnetic properties of three specific materials with two different approaches (PAW and LCAO). Examined materials are *Fe* (pure iron), in its body-centred cubic (BCC) and face-centred cubic (FCC) configurations, and more complex alloys, an intermetallic alloy γ – *TiAl* and Heusler alloy *Ni₂MnGa*.

5.1 Fe

Iron in its pure form is not a particularly remarkable material in terms of technical and industrial applications. Even though the production of iron accounts for 90% of global metal production, it is almost always combined with another alloying element, due to lacking mechanical properties.

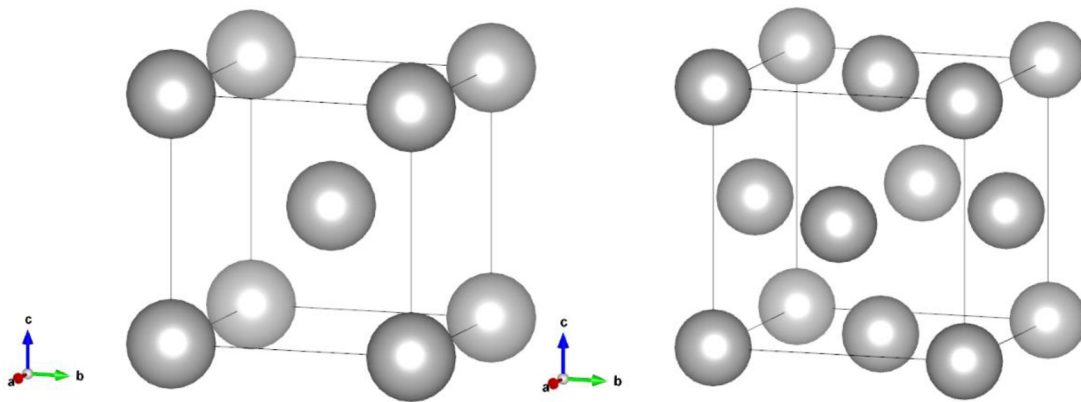


Figure 4 Fe lattice in its BCC configuration on the left, in its FCC configuration on the right – created in software VESTA

Iron exists in BCC and FCC configurations [Figure 4] and has several magnetic phases. This thesis is mainly focused on the analysis of the ferromagnetic state of BCC iron. Therefore, the antiferromagnetic state of FCC iron is not necessarily considered. Lattice parameters of examined bcc-Fe structure is $a \approx 2.83 \text{ \AA}$. [27, 28, 29, 30]

5.2 TiAl

Titanium aluminide (TiAl)-based alloys belong to the distinctive group of materials, classified as intermetallics [Figure 5]. Intermetallics utilize the principle of ordered crystallographic structure, which contains a combination of different metals, created by the limited solubility of the alloys. Such alloys usually have different structure and properties in contrast to their parental materials. Intermetallics have long-range order crystallographic structure, which results in interesting mechanical properties even at higher temperatures. TiAl-based alloys specifically generate significant interest due to their potential applications in high stress, high-temperature environments. [31, 32]

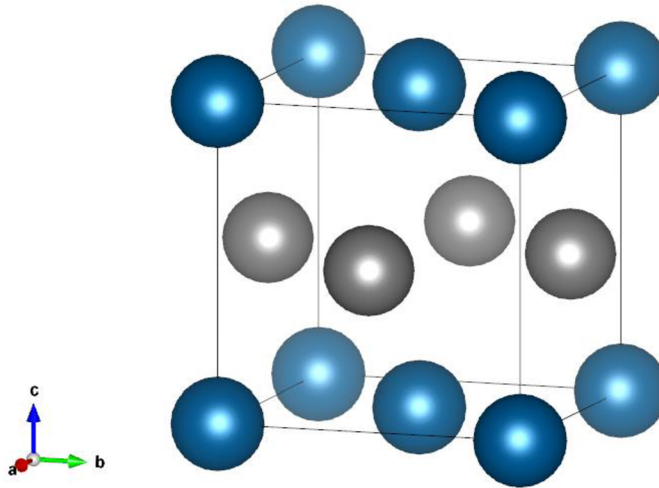


Figure 6 $L1_0$ lattice of γ -TiAl alloy (blue - titanium and grey - aluminium) – created in software VESTA

This thesis examined the $L1_0$ structure of γ -TiAl [Figure 6] using PAW and LCAO approaches. Lattice parameters of relaxed γ -TiAl structure are approximately $a \approx 3.98 \text{ \AA}$ and $c \approx 4.08 \text{ \AA}$.

5.3 Ni_2MnGa

The Ni_2MnGa is a ferromagnetic, shape memory alloy, crystallizing in cubic space group $Fm\bar{3}m$, with structure type according to the Cu_2MnAl ($L2_1$). The alloy belongs in the category of Heusler alloys. Heusler alloys are generally ternary semiconducting or metallic materials with chemical composition X_2YZ (for full-Heusler alloys), or XYZ (for half-Heusler alloys). Possible structures of Heusler alloys are depicted below, in Figure 7. [38]

Heusler alloys generate considerable interest, mainly due to their remarkable electron structure, which results in unique magnetic properties. Half-Heusler compounds exhibit only one magnetic sublattice formed by the atoms in octahedral positions, which can carry a magnetic moment. On the other hand, full-Heusler compounds exhibit two magnetic sublattices due to the X atoms occupying the tetrahedral positions, which allows magnetic interactions of X atoms and the formation of secondary, more delocalized magnetic sublattice, depicted below in Figure 7. As a result, full-Heusler compounds display a wide array of magnetic phenomena such as ferromagnetism, ferrimagnetism, or half-metallic ferromagnetism. [38]

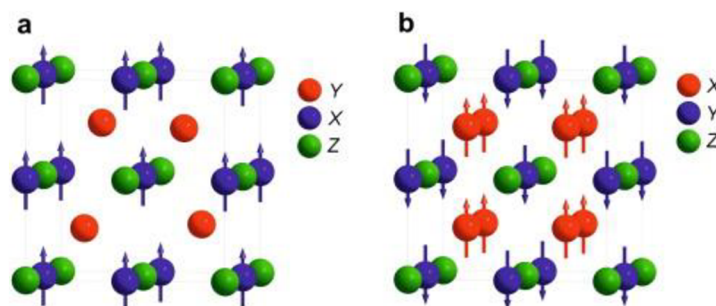


Figure 7 a) half-Heusler alloy can only exhibit one magnetic sublattice in octahedral positions, while b) full-Heusler alloys exhibit two magnetic sublattices, that can couple ferromagnetically or antiferromagnetically [38]

Ni₂MnGa in particular generated substantial interest in recent years due to its shape memory behaviour, and its potential application as a magnetically driven memory shape alloy. This means that macroscopic deformation can be controlled by external magnetic fields. [39]

Shape memory alloys (SMA) exhibit polymorphic behaviour dependent on the temperature. Phase transition between high-temperature, higher symmetry phase, austenite, and low-temperature, lower symmetry phase, martensite, occurs below the temperature of martensitic transformation. This transformation is athermal. A number of planar crystallographic defects, such as twin boundaries, are created during the martensitic transformation in order to decrease the energy of the interphase boundary, which is strained due to differences in lattice parameters of martensitic and austenitic lattices. The twins created during phase transformation have random orientations, which results in an internal stress of the martensitic structure. [40]

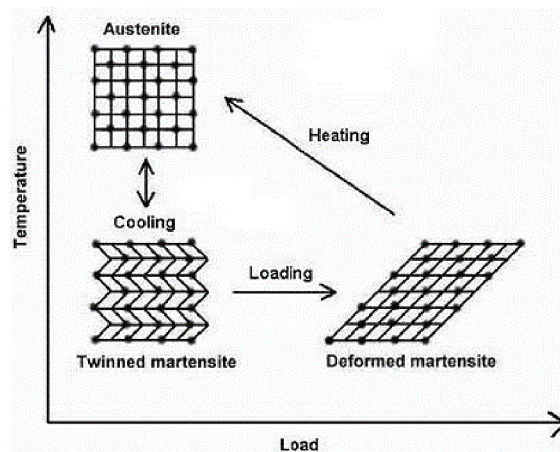


Figure 8 Principle of shape memory behaviour, caused by the reorientation of twins under loading [41]

The transformation of randomly oriented twins to a more favourable type of martensitic twins can be achieved by further loading of the transformed structure and causing additional strain. This is possible due to the phenomenon, pseudo-plasticity, which is based on the movement of twin boundaries due to the influence of shear stress being applied to the structure. The principle of shape memory behaviour is, that by heating up the deformed martensitic structure above the temperature of martensitic transformation, the austenitic transformation occurs, thus transforming the structure back to its original shape and macroscopic dimensions. The process is depicted above in Figure 8.

Furthermore, magnetic shape memory alloys (MSMA) are a special subcategory of shape memory alloys. The principle of MSMA lies in a large structure deformation induced by the external magnetic fields (magnetic field induced strain or MFIS). MFIS follows two principles. The first is characterized as a phase transformation between austenitic and martensitic phases at constant temperature induced by the applied external field. [42, 43]

The second principle of MFIS observed in the Ni-Mn-Ga MSMA is based on pseudo-plastic behaviour, which occurs at temperatures lower than the temperature of martensitic transformation. In this case, no new phase is created and only reorientation of already

existing twins occurs (magnetic induce reorientation – MIR). In the case of MIR, the magnetic field induced strain is not reversible by just removing a magnetic field. A driving force behind reverse reorientation is necessary, therefore the magnetic field of the same magnitude, but in the perpendicular direction is required. The highest magnetic field induced strain was measured in Ni-Mn-Ga alloy, with 4 at. % of Co, and 4at. % of Cu at the expense of Ni and Ga respectively. The relative deformation reached 12%. [41]

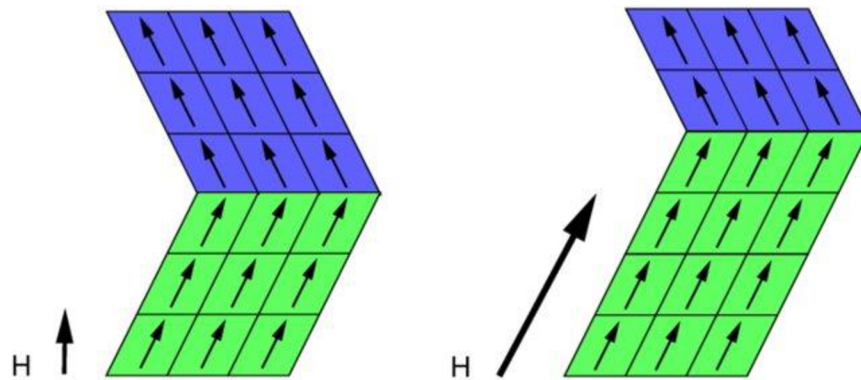


Figure 9 Schematic representation of magnetic field induced reorientation (MIR). Applied magnetic field favours martensitic twin with their easy axis parallel to given direction, causing MFIS [45]

The principle of MIR lies in the reorientation of pre-existing twins in the martensite with respect to the magnetic anisotropy of the lattice. Reorientation within the MSMA in the magnetic field occurs due to the high mobility of the twin boundaries. Therefore, it is easier for the material to react to the change in the magnetic field by macroscopic deformation than the reorientation of magnetic moments in the lattice. [43, 45, 46]

Low twinning stress (σ_{TS}) is not the only criterion for MIR, as this stress has to always be lower than σ_{mag} , a theoretical maximal stress induced by the magnetic field, defined by the formula:

$$\sigma_{mag} = K_U \left(\frac{c}{a}\right)^{-1}, \quad (5.1)$$

It is a function of c/a parameter and constant of magnetic anisotropy K_U . Materials are capable of MFIS if this condition is met, due to the high mobility of twin boundaries. The greater the difference between σ_{TS} and σ_{mag} the lower the intensity of the magnetic field is necessary for the movement of the twin boundaries. [47, 48]

The high-temperature austenitic phase of Ni_2MnGa is cubic ($L2_1$), with a lattice parameter approximately $a \approx 5.797 \text{ \AA}$, and it is ferromagnetic at room temperature. The Curie temperature, and therefore a temperature of ferromagnetic transition is at $T_c \approx 376 \text{ K}$. Stoichiometric Ni_2MnGa undergoes a martensitic transformation (transformation from high-temperature cubic, austenitic, structure to low-temperature tetragonal, martensitic, structure) at temperature $T_m \approx 210 \text{ K}$. This structural transformation has been observed to be reversible, which allows the shape memory behaviour of the alloy. The Curie

temperature and the temperature of martensitic transformation are heavily dependent on stoichiometry of the alloy, thus any change to its chemical composition can impact its magnetic behaviour. [43]

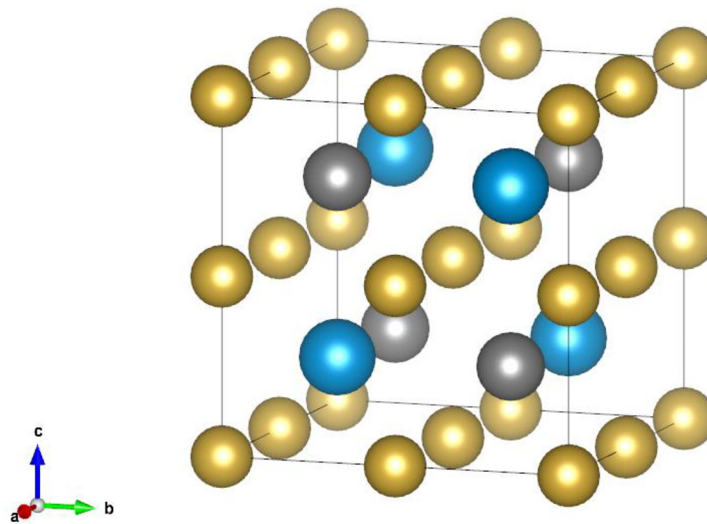


Figure 10 Cubic lattice of Ni_2MnGa alloy (yellow - nickel, grey - manganese and blue - gallium) – created in software VESTA

The martensitic transformation of Ni_2MnGa is caused by the electron instability of its austenitic phase. The simplest martensitic structure, dubbed non-modulated (NM) martensite [Figure 11], is based on an austenitic $L2_1$ space group structure with deformed geometry. The structure is elongated along the one axis c , and it is shortened along the two remaining axes a , and b in order to preserve the constant volume. The rate of tetragonal deformation is characterized by the c/a parameter, which depicts the ratio between the lattice vectors. [49]

Other more complex structures, called modulated, are also based on deformed $L2_1$ structure with a $c/a < 1$. Modulation of the structure means a noticeable decrease in symmetry of the structure due to the shuffling of atomic planes (110) in the direction $[1\bar{1}0]$. Structure modulation can be defined using two principles.

The first principle describes the displacement of atomic planes in the direction of the planes, or perpendicular to it, by means of harmonic function. This function can be proportional, or disproportional to the crystal lattice.

The second principle is based on the notion of adaptive martensite and uses the principle of nano-twinning of non-modulated martensite. Nano-twins are the result of a compromise between the energy of the phase interface between austenite and martensite, and the energy of the twin boundary. The presence of the nano-twins decreases the energy of the phase interface.

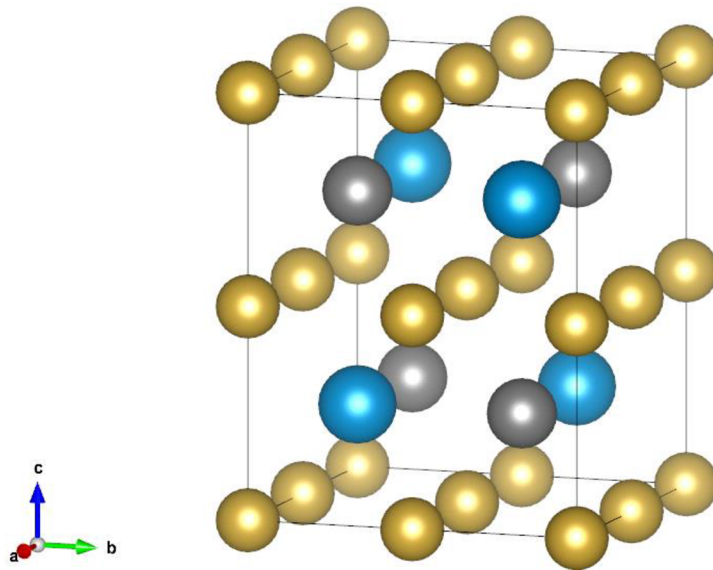


Figure 11 Tetragonal lattice of Ni_2MnGa alloy (yellow - nickel, grey - manganese and blue - gallium) – created in software VESTA

In the case of the relaxed non-modulated martensitic structure depicted in [Figure 11] the value of the c/a parameter lies in the range from 1.20 to 1.25, based on the stoichiometric composition of the alloy. [49]

6 Simulation details

This thesis compared two approaches to describe interatomic interactions, the projector-augmented wave method (PAW), implemented in VAPS (Vienna Ab initio Simulation Package), and the method of the linear combination of atomic orbitals (LCAO), implemented in SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms). The simulation runs were driven by the Atomic Simulation Environment (ASE) interface, based on Python programming language.

6.1 Atomic simulation environment

The atomic simulation environment (ASE) is a software package of tools and Python modules for setting up, running, analysing, and visualizing atomistic simulations. ASE was created using the programming language Python with the aim to be customizable, very flexible, and easy to use. ASE allows calculations of even very complex simulation tasks. ASE provides an interface through which different simulation packages, such as VASP, SIESTA, LAMMPS, etc., can be implemented. Besides this calculator interface, ASE can provide many modules for standard simulation tasks, such as structure optimization, molecular dynamics, etc. ASE version *ase-3.22.0* was used for these calculations. [50]

6.1.1 ASE computational details

This thesis used several ASE specific modules, from which a number of distinctive elements and tools were imported and applied:

Module	Description
ase	Main module of ASE, containing the most basic objects required for the simulations.
ase.io	Module allows reading and writing of input and output files from different simulation packages.
ase.calculators.vasp	Module contains VASP calculator.
ase.calculators.siesta	Module contains SIESTA calculator.
ase.units	Module defines physical units like Bohr, Hartree, Rydberg, kcal, etc.
ase.optimize	Module contains tools for local or global structural relaxation and energy minimization.
ase.constraints	Module allows changing the degrees of freedom in the system for structure relaxation.
ase.parallel	Module allows to run calculations in parallel mode.

Table 5 Table of ASE specific modules used for calculations [50]

Structural relaxation within the ASE was performed using Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, a quasi-Newtonian iterative method for solving nonlinear optimization problems. In principle, the method approximates the second-order derivate,

Hessian, to solve the optimization problem, where the solution of the second derivative is problematic. It is one of the most used algorithms for numerical optimization.

Structural relaxations within this thesis using the BFGS algorithm were performed for the positions and whole unit cells simultaneously, with maximum force condition equal to 0.005 eV/\AA^3 . [51, 52, 53, 54]

Object	Description
<code>set_initial_magnetic_moments</code>	Sets initial magnetic moments of the atoms.
<code>get_atomic_numbers</code>	Retrieves an integer array of atomic numbers in the system.
<code>get_potential_energy</code>	Calculates potential total energy of the system.
<code>get_magnetic_moment</code>	Calculates total magnetic moment of the system.
<code>get_magnetic_moments</code>	Calculates local magnetic moments of the atoms.
<code>get_cell_lengths_and_angles</code>	Retrieves a unit cell parameters (cell vector lengths and angles).
<code>UnitCellFilter</code>	Filter for optimizing positions and unit cell simultaneously.
<code>BFGS</code>	One of the minimizers in the ASE package. Determines movement of the atoms during structural relaxation.

Table 6 Table of ASE specific methods of an object describing atomic arrangements used for the calculations [50]

6.2 Vienna Ab initio Simulation Package

The Vienna Ab initio Simulation Package (VASP) is a computational software package for atomic-scale materials modelling, electronic structure calculations and quantum-mechanical molecular dynamics, from the first principle. VASP calculates an approximate solution to the many-body Schrödinger equation within DFT, solving the system of Kohn-Sham equations. For the description of electron and ion interaction, VASP uses the PAW method. VASP version *vasp.5.4.4* was used for these calculations. [55, 56]

6.2.1 VASP computational parameters

Standard VASP calculations use four input files: [55, 56]

- INCAR, which contains the input parameters that control the calculation.
- POSCAR, which contains information about the structure (lattice parameters, Bravais matrix, names and number of atoms, specification for the coordinate system and atom positions)
- KPOINTS determines the sampling of the first Brillouin zone
- POTCAR contains the information about the pseudopotentials that are necessary to run the calculations.

However, due to the use of the atomic simulation environment all the parameters of the simulations were specified in the ASE script using values in Table 7 below, and only POSCAR, as a definition of analysed structure, was used.

VASP simulations employed in this thesis used parameters and their values from Table 7. Parameters specifically not mentioned kept their default settings.

Calculator parameter	Description	Used values
ENCUT	ENCUT specifies the cut-off energy for the plane-wave-basis set in eV.	600
SIGMA	SIGMA specifies the width of the smearing in eV.	0.02
EDIFF	EDIFF specifies the global break condition for the electronic self-consistent loop. It is specified in units of eV.	1e-5
SYMPREC	SYMPREC determines to which accuracy the position in the POSCAR file must be specified.	1e-8
GGA	GGA specifies an LDA or GGA exchange-correlation functional.	PE
PREC	PREC specifies the precision-mode	Accurate
IALGO	IALGO selects the algorithm used to optimize the orbitals	48
ISMEAR	ISMEAR determines how the partial occupancies are set for each orbital.	1
ISPIN	ISPIN specifies spin polarization.	2
ISTART	ISTART determines whether or not to read the WAVECAR file.	0
LORBIT	LORBIT, together with an appropriate RWIGS, determines whether PROCAR or PROOUT files are written.	11
LREAL	LREAL determines whether the projection operators are evaluated in real-space or in reciprocal space.	False

Table 7 Table of the VASP parameters used to configure the calculations [55, 56]

6.3 Spanish Initiative for Electronic Simulations with Thousands of Atoms

The Spanish Initiative for Electronic Simulations with Thousands of Atoms (SIESTA) is a computational software package and method for efficient electronic structure calculations and ab initio molecular dynamics simulations of molecules and crystals. SIESTA's efficiency originates from the use of basis sets of strictly-localized atomic orbitals. In particular, the double zeta + polarization (DZP) basis functions were used in the thesis. The capabilities of SIESTA's code can cover a wide range of possible calculations, from quick and wide exploratory simulations to highly accurate simulations, matching the quality of other approaches such as plane-wave-based methods. SIESTA version *MaX-1.2.0* was used for these calculations.

6.3.1 SIESTA computational parameters

Standard SIESTA calculations use an FDF-format file. It contains all the input data necessary for the calculations similar to the files from VASP like information about the initial structure, pseudopotentials used for the simulations, sampling of the first Brillouin zone and parameters controlling the simulations.

However, similarly to the VASP calculations, ASE was also used to set up and control the SIESTA calculations. Therefore SIESTA's FDF-type input file was generated using parameters from Table 8 with the help of the ASE script, and only a POSCAR-like file was used to define the analysed structure. [57]

Calculator Parameter	Description	Used values
MeshCutoff	Defines the plane-wave cutoff for the grid.	5442.277 eV
PAO.EnergyShift	Modifies the localization of basis function.	0.136 eV
MaxSCFiteration	Maximum number of self-consistent field (SCF) iterations.	100
SCF.DM.Tolerance	The self-consistency is achieved, when the maximum difference between the output and input on each element of the density matrix (DM) in a SCF cycle is below its value.	0.0001
SCF.EDM.Tolerance	The self-consistency is achieved, when the maximum absolute change in the energy density matrix elements is below its value.	1e-2 eV
SCF.H.Tolerance	The self-consistency is achieved, when the maximum absolute change in Hamiltonian matrix elements is below its value.	1e-3 eV
SCF.Mixer.History	Determines the number of previous SCF steps used in estimating the subsequent inputs.	16
SCF.Mixer.Method	Chooses the mixing algorithm between different methods.	Pulay
SCF.Mix.Spin	Controls how the mixing is performed, when carrying out spin-polarized calculations.	spinor
SCF.Mixer.Weight	Determines the mixing weight used to mix the quantity.	0.4
LongOutput	By default, not all data sets are written in standard outputs. This allows SIESTA to write more information into the output.	True
WriteMullikenPop	Determines the level of Mulliken population analysis. (1 – to calculate total and local magnetic moments)	1

Table 8 Table of the SIESTA parameters used to configure the calculations [57]

6.4 Common calculation parameters

Some parameters were consistent across these two methods:

- For the exchange and correlation energies generalized gradient approximation (GGA) according to the Pedrew-Burke-Ernzerhof (PBE) was used.

- The number of k-points, the sampling of the first Brillouin zone, was defined by the k-point matrix:

$$\begin{bmatrix} 14 & 0 & 0 \\ 0 & 14 & 0 \\ 0 & 0 & 14 \end{bmatrix}$$

Higher k-point matrices were also considered, however, influence on the results has proven to be extremely minimal.

- Pseudo-potentials used for the VASP calculations were obtained from the VASP software itself, while PSML pseudo-potentials used for SIESTA calculations were obtained from the *Pseudo-Dojo* database.[59]

Element	VASP	SIESTA
Fe	PAW_PBE Fe_sv 23Jul2007	Fe_NC_FR_PBE_PSML
Ti	PAW_PBE Ti_pv 07Sep2000	Ti_NC_FR_PBE_PSML
Al	PAW_PBE Al 04Jan2001	Al_NC_FR_PBE_PSML
Ni	PAW_PBE Ni_pv 06Sep2000	Ni_NC_FR_PBE_PSML
Mn	PAW_PBE Mn_pv 02Aug2007	Mn_NC_FR_PBE_PSML
Ga	PAW_PBE Ga_d 06Jul2010	Ga_NC_FR_PBE_PSML

Table 9 Table of used pseudo-potentials [13, 58, 59]

6.5 Python

Python is a high-level, general-purpose programming language, which emphasizes significant code readability. It supports multiple programming paradigms, such as object-oriented, structured, or functional programming. As previously mentioned, ASE was fully scripted using the python programming language, therefore implementation of simulations into the python script was incredibly straightforward. [60]

6.5.1 Scripts for calculation of elastic constants

Scripts for the calculation of elastic constants were written using the programming language Python, based on the previously mentioned study by R. Yu et al. [26] Two scripts were written, and both are included in the appendix. The purpose of the first script is to create strain tensors using Voigt notation, create a deformation matrix, adjust input files, and create twelve deformed structures. For these twelve deformed structures, the relaxation of ionic positions is then launched, during which lattice vectors stay unchanged. The main output from these calculations is stress tensors for all twelve structures. These stress tensors, together with deformation tensors, then serve as an input for the second script, which calculates the elastic constants of the examined structure. The output of the second script is a 6 x 6 elastic constants matrix. Both of the scripts in the Appendix are modified for VASP simulations, however, with minor alterations to the code it is possible to use them with other methods as well. Scripts were created with Python3.

7 Results and discussion

Calculations for our chosen materials, *Fe*, γ -*TiAl* and *Ni₂MnGa*, were performed and the results of these simulations are presented and discussed in this chapter.

7.1 Energy-Volume (E-V) curves and dependence of magnetic moments on cell volume

Calculation of the energy-volume curve allows us to determine the equilibrium energy, the equilibrium cell volume, the bulk modulus and its derivatives. These properties can be further used to possibly establish an initial prediction of thermodynamic properties. The main parameter examined by these simulations was the dependence of the total energy of the system, and magnetic moments per atom on the cell volume. These parameters were calculated using VASP and SIESTA simulation packages and are shown in Figure 12 through Figure 17.

7.1.1 Fe

The dependence of relative total energy on the cell volume of Fe is represented in Figure 12. The energy-volume curve for Fe was calculated for 17 different atomic volumes. Total energies were then recalculated with respect to the local minimum. The figure depicts that the equilibrium atomic volume for VASP is approximately 11.33 Å³/atom, and the equilibrium atomic volume for SIESTA is approximately 11.45 Å³/atom. Furthermore, it shows that values of relative total energies for Fe from both VASP and SIESTA are noticeably comparable.

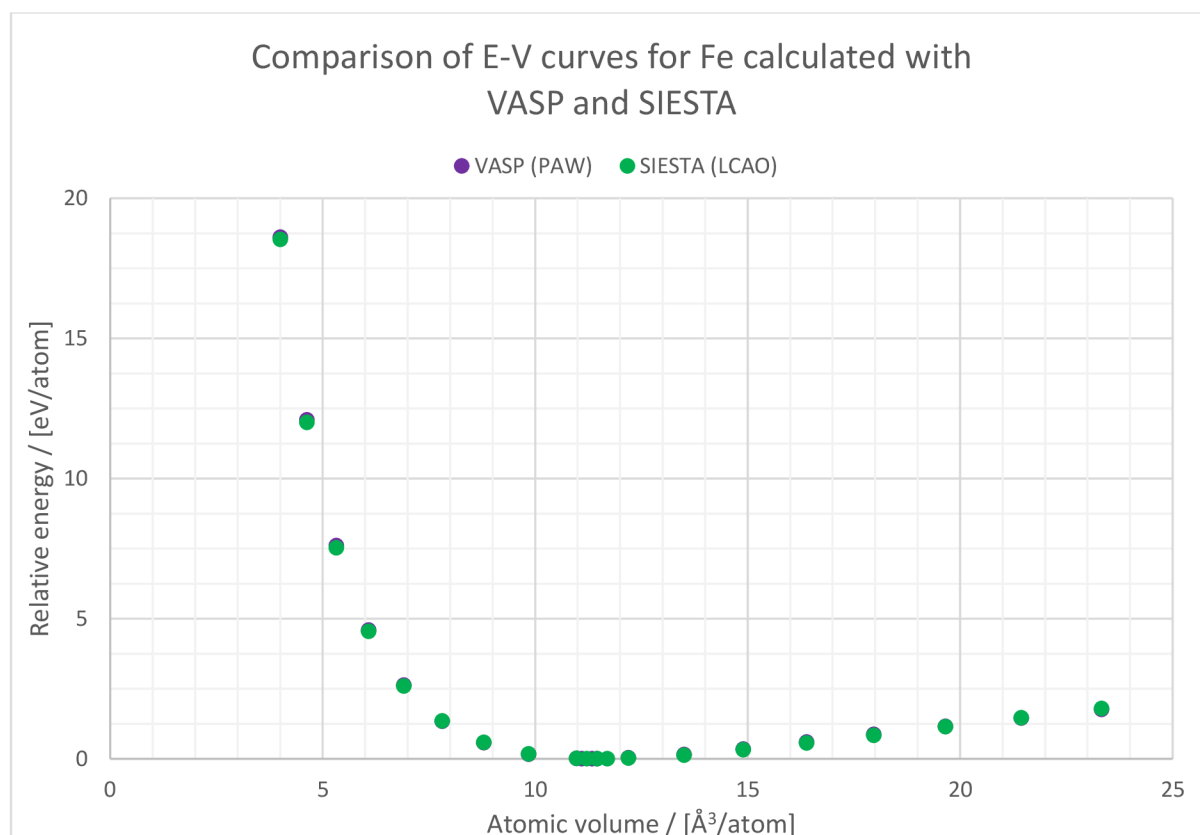


Figure 12 Comparison of energy-volume curves for Fe

The reason behind calculations being performed for such small atomic volumes is the examination of average magnetic moments. Average magnetic moments were calculated and marked in Figure 13 as a function of atomic volume.

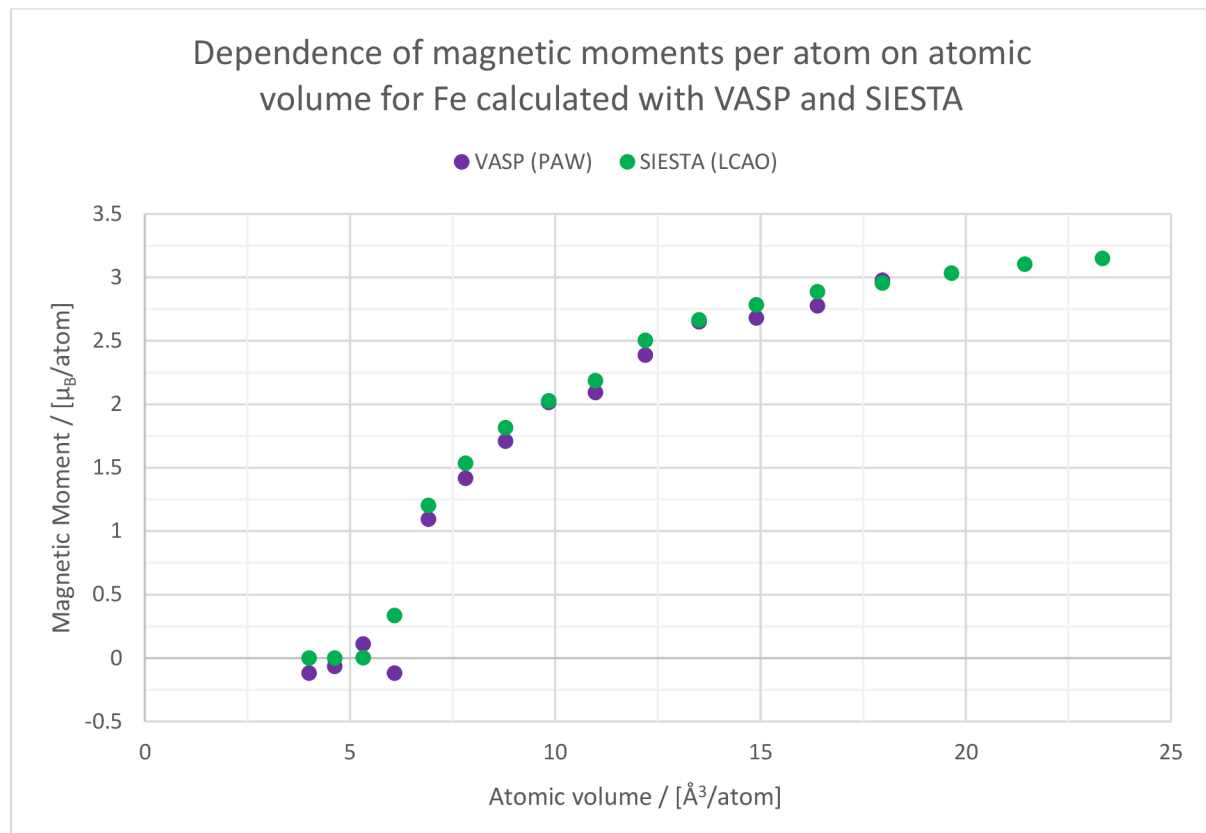


Figure 13 Comparison of the magnetic moments per atom as a function of cell volume for Fe

The dependence of magnetic moments per atom on atomic volume for Fe calculated by VASP and SIESTA is represented in Figure 13. It shows comparable results across these two methods. As previously stated, calculations included very small atomic volumes, where a sharp drop in the magnetic moment per atom can be observed at an atomic volume of 6.91 Å³/atom, which is approximately 60% of equilibrium atomic volume. Any significant differences below this volume can be attributed to the calculation error at a given point.

7.1.2 TiAl

The dependence of relative total energy on the cell volume of γ -TiAl is represented in Figure 14. The energy-volume curve for γ -TiAl was calculated for 13 different atomic volumes. Total energy was recalculated with respect to the local minimum. The figure shows that the equilibrium atomic volume of γ -TiAl for VASP is approximately 16.32 Å³/atom, and for SIESTA is 16.18 Å³/atom. It also shows a noticeable disparity in relative total energy with changing atomic volume, which can result in different elastic properties calculated by these two methods.

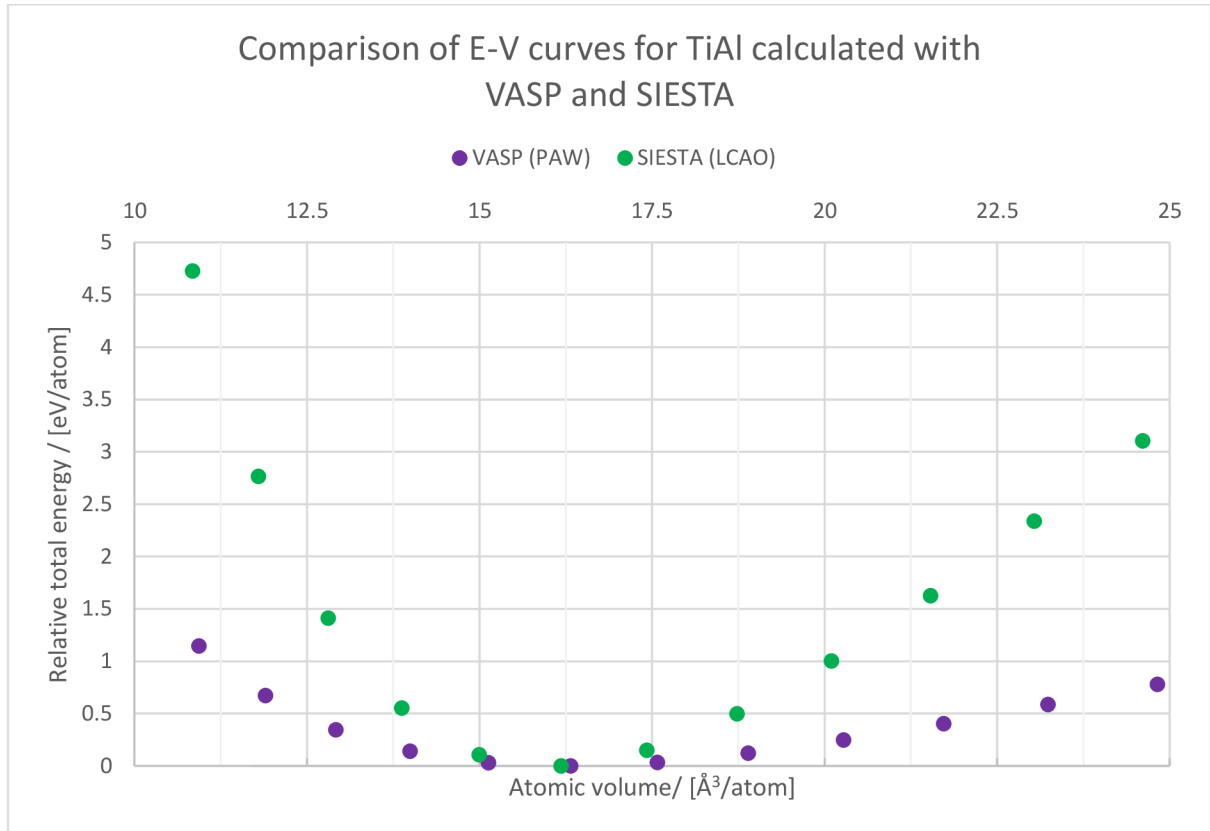


Figure 14 Comparison of energy-volume curves for γ -TiAl

Due to the nonmagnetic nature of γ -TiAl, depicting the dependence of magnetic moments on cell volume was deemed unnecessary.

7.1.3 Ni₂MnGa

The dependence of relative total energy on the cell volume of Ni₂MnGa austenite is represented in Figure 15. The energy-volume curve for Ni₂MnGa was calculated for 24 different atomic volumes. Total energy was recalculated with respect to the local minimum. The figure shows very small differences in relative total energies in the proximity of equilibrium atomic volume, while the only noticeable differences are present in small atomic volumes. Due to the inclusion of small atomic volumes in calculations, for the purpose of the examination of magnetic moments, the equilibrium energy is difficult to differentiate, therefore the second figure depicting a detail of the energy-volume curves in the proximity of the minimums was calculated.

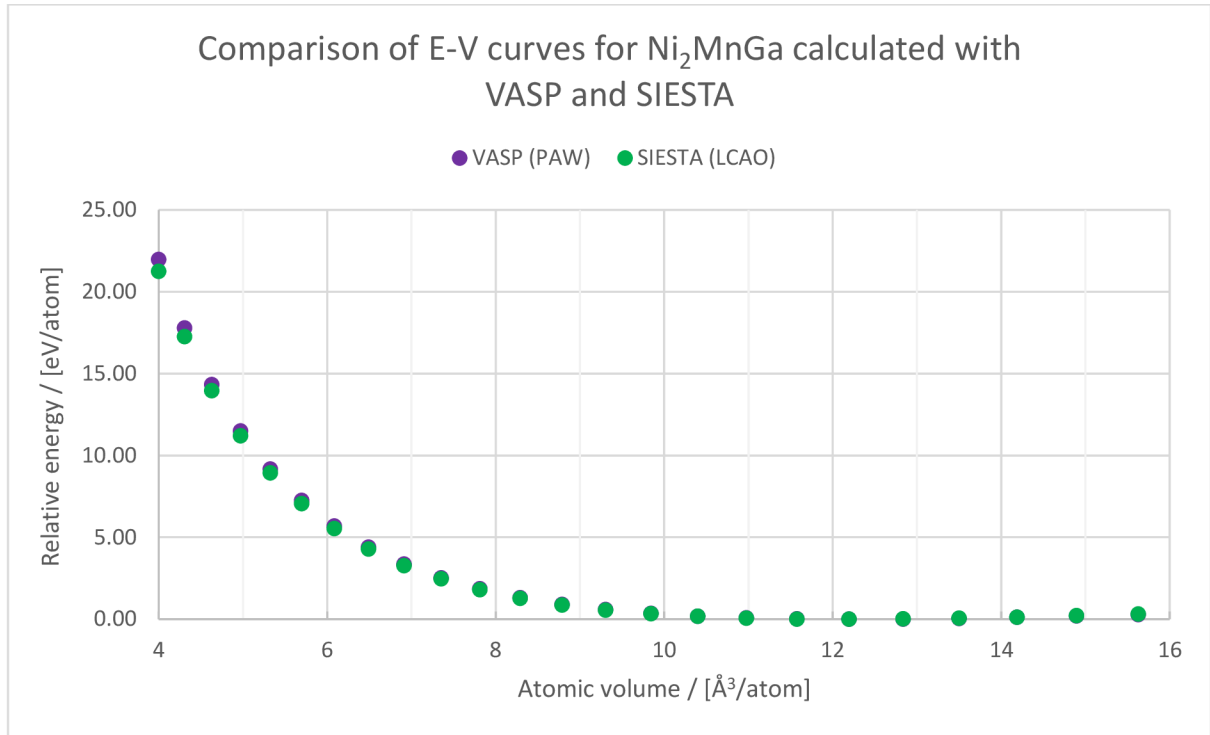


Figure 15 Comparison of energy-volume curves for Ni₂MnGa

The detail of energy-volume curves in the proximity of the minimums, from Figure 15 is represented in Figure 16. It shows that the equilibrium atomic volume for VASP is approximately 12.26 Å³/atom, while the equilibrium atomic volume for SIESTA is approximately 12.01 Å³/atom. Consequently, it shows that the energy-volume curve for the SIESTA is shifted slightly towards the smaller atomic volumes.

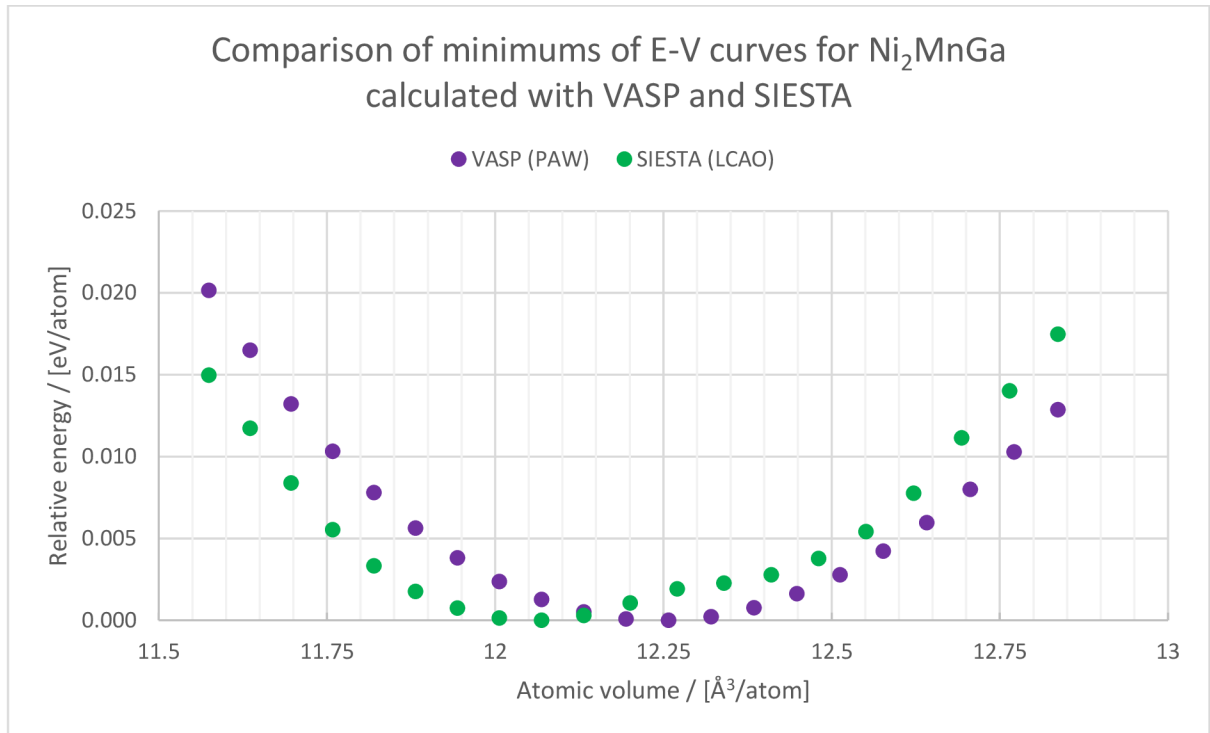


Figure 16 Comparison of minimums of energy-volume curves for Ni₂MnGa

The dependence of magnetic moments per atom on atomic volume for Ni₂MnGa calculated by VASP and SIESTA is depicted in Figure 17. It once again shows comparable results across these two methods. Similarly to the Fe calculations, very small atomic volumes were included, where once again a sharp drop in the magnetic moment per atom can be observed at small atomic volumes, namely at 5.69 Å³/atom for VASP and 4.96 Å³/atom for SIESTA. Which is approximately 46.41% of equilibrium atomic volume for VASP, and 41.30% of equilibrium atomic volume for SIESTA. Any significant differences below these volumes could be once again attributed to the calculation error in these points.

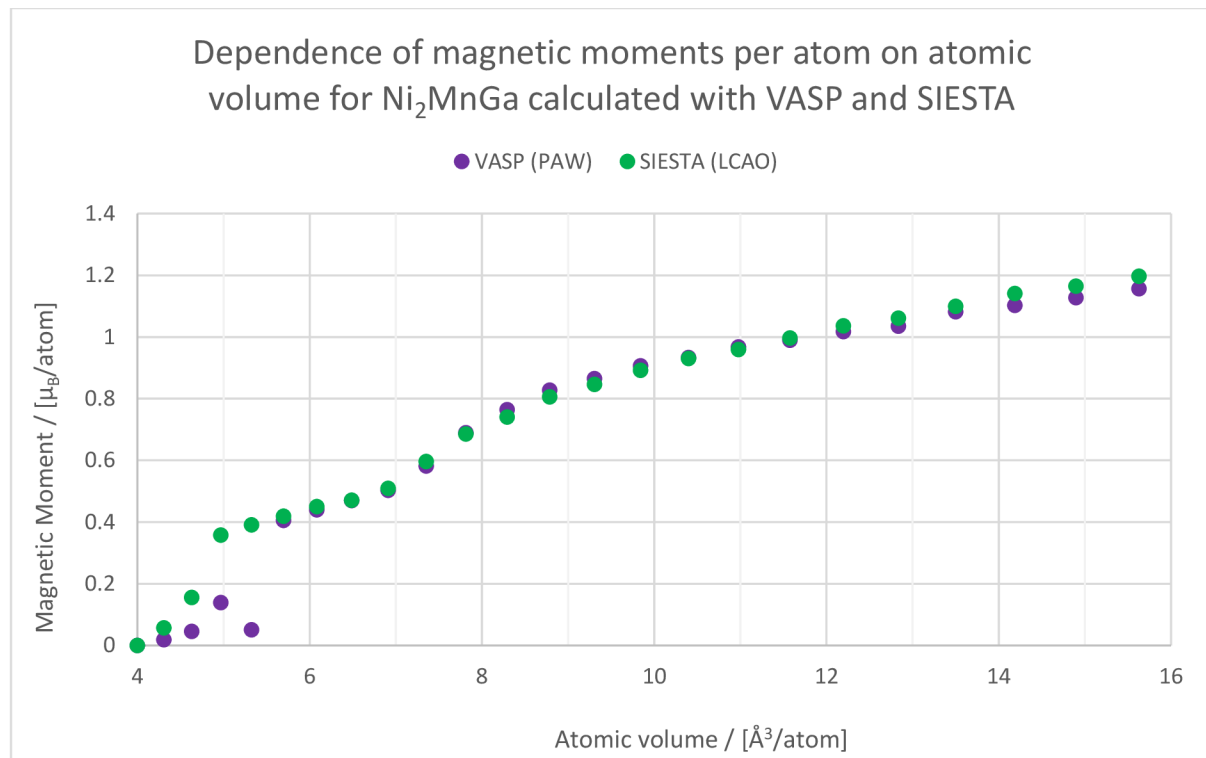


Figure 17 Comparison of the average magnetic moment as a function of cell volume for Ni₂MnGa

7.2 Tetragonal deformation paths

The tetragonal deformation path or Bain's transformation path describes the dependence of total energy on a tetragonal distortion of the structure. The degree of tetragonal distortion, a c/a parameter, is characterized as a ratio between c and a lattice parameters. Figure 18 depicts an example of a tetragonal deformation path for stable, metastable, or unstable material, while Figure 19 depicts a tetragonal distortion of a cubic cell.

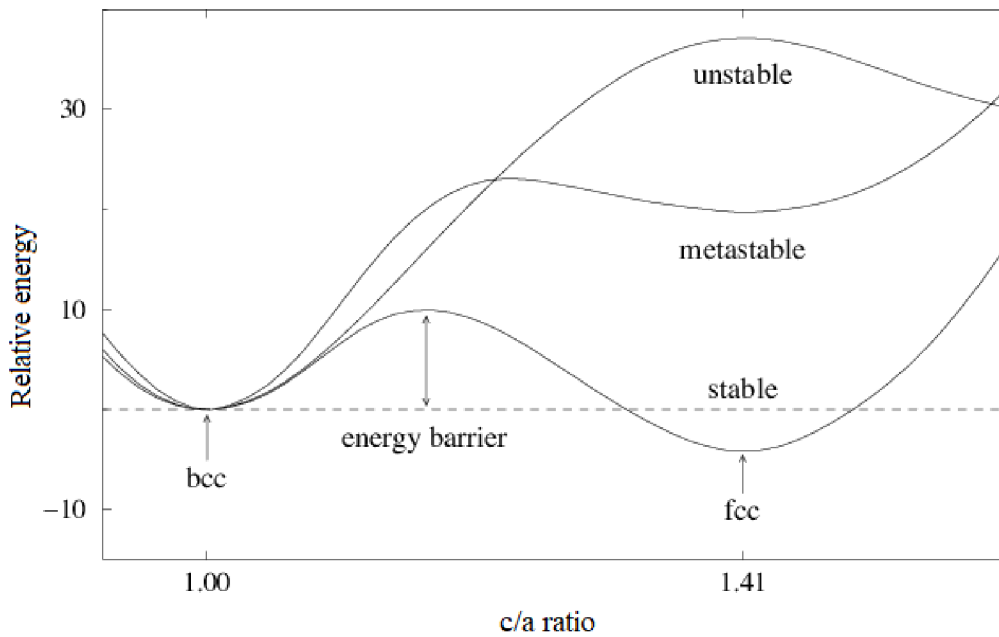


Figure 18 Example of tetragonal deformation path, depicting three possible states of the material, stable, metastable, and unstable [61 - edited]

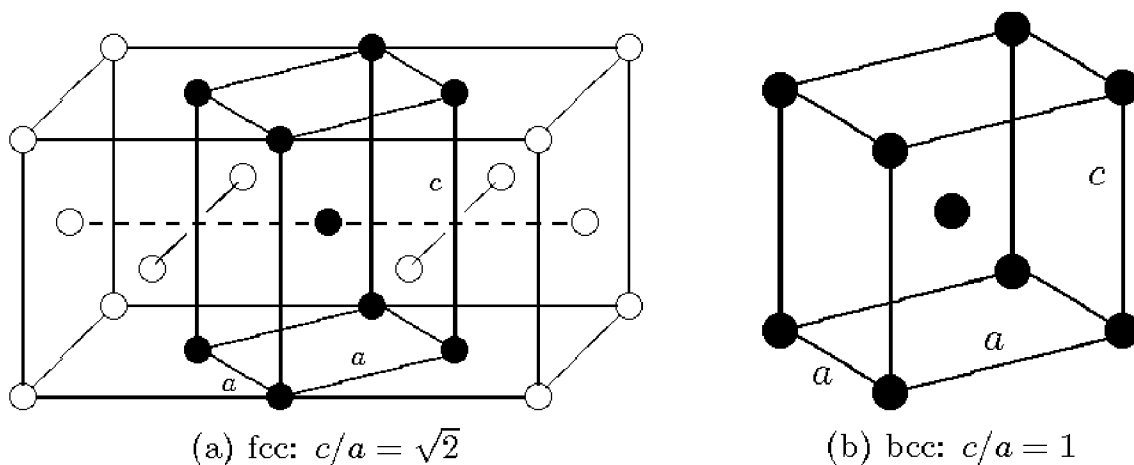


Figure 19 Tetragonal deformation of cubic cell [62]

7.2.1 Fe

The tetragonal deformation path of Fe is depicted in Figure 20. Total energy was calculated for 29 different c/a parameters, from $c/a = 0.9$ to $c/a = 1.6$. Total energies were then recalculated with respect to the energy of the cubic structure (BCC, $c/a = 1.0$). Results from VASP calculations depict a tetragonal deformation path for Fe with low spin state electrons, a metastable state, while results from SIESTA calculations depict a tetragonal deformation path for Fe with high spin state electrons, an unstable state. The figure depicts FCC Fe at deformation $c/a = \sqrt{2}$.

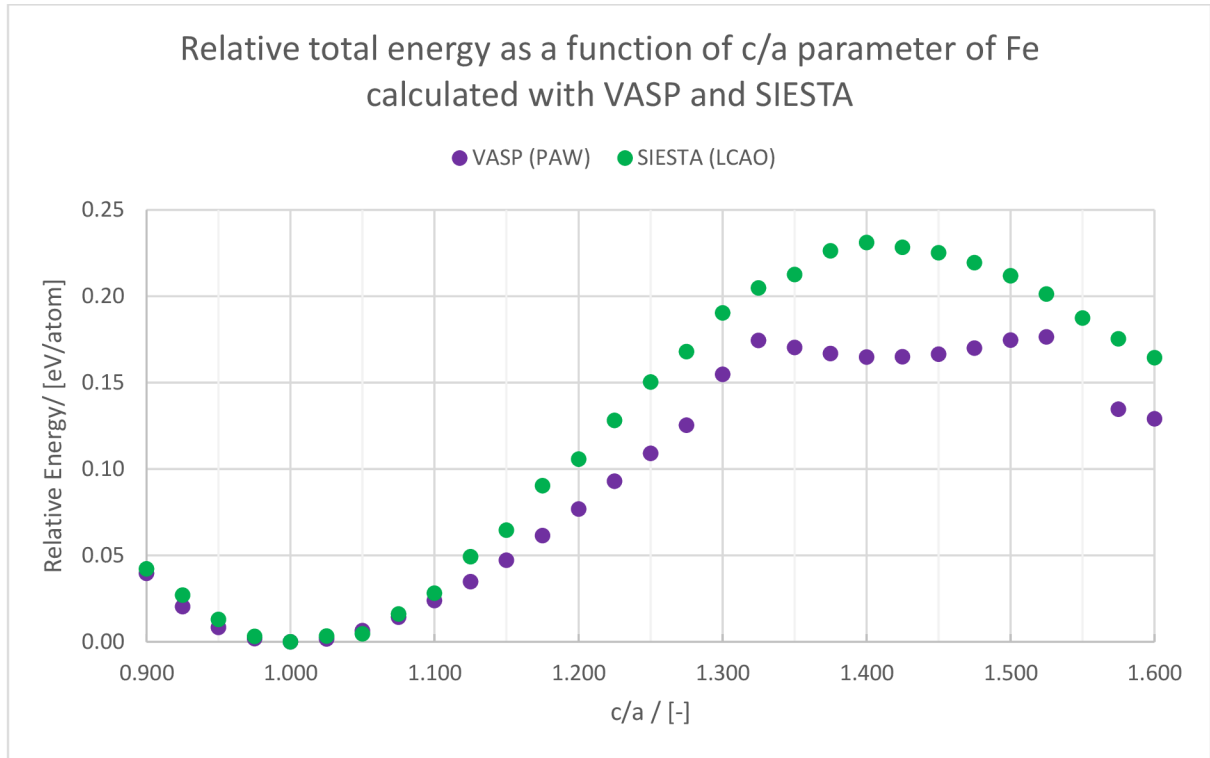


Figure 20 Tetragonal deformation path for Fe

BCC iron has only one possible electron state, while FCC iron has two. A high spin state, which results in a higher magnetic moment, while a low spin state results in lower magnetic moments in comparison. This is possible to be observed in Figure 21. It is apparent that SIESTA calculations hold this high spin state throughout the whole deformation path, while VASP automatically finds the low spin state, which is energetically more convenient. Whilst accounting for the different spin states of the electron across the two methods, the results could be considered acceptable.

The dependence of magnetic moments per atom on the tetragonal deformation of Fe calculated with VASP and SIESTA is depicted in Figure 21. It describes the difference between the high spin electron state of Fe in SIESTA, and low spin state of Fe in VASP. It is apparent that magnetic moments from SIESTA calculations change very little with the tetragonal deformation, while magnetic moments from VASP calculations decrease at $c/a = 1.275$, which corresponds with the occurrence of metastable Fe (FCC) at previously mentioned $c/a = \sqrt{2}$.

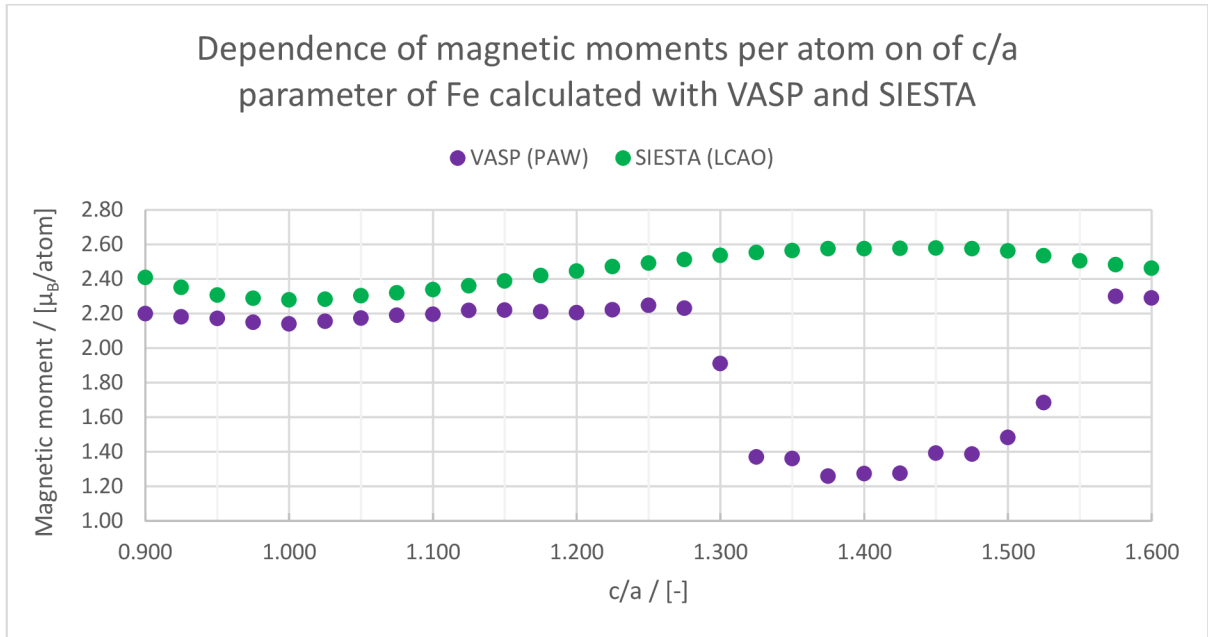


Figure 21 Magnetic moments per atom as a function of tetragonal deformation for Fe

7.2.2 TiAl

The tetragonal deformation path of γ -TiAl is represented in Figure 22. Total energy was calculated for 25 different c/a parameters, from $c/a = 0.9$ to $c/a = 1.2$. Total energies were then recalculated with respect to the energy of the cubic structure ($c/a = 1$). It indicates that VASP reaches a local minimum at approximately $c/a = 1.02$, while SIESTA reaches a local minimum at approximately $c/a = 1.015$. This notion, $c/a > 1$, is consistent with experimental observations. [63] It appears that the increase of relative energy with volume in SIESTA is slightly steeper compared to the VASP, which once again points to the possible difference in the elastic constants.

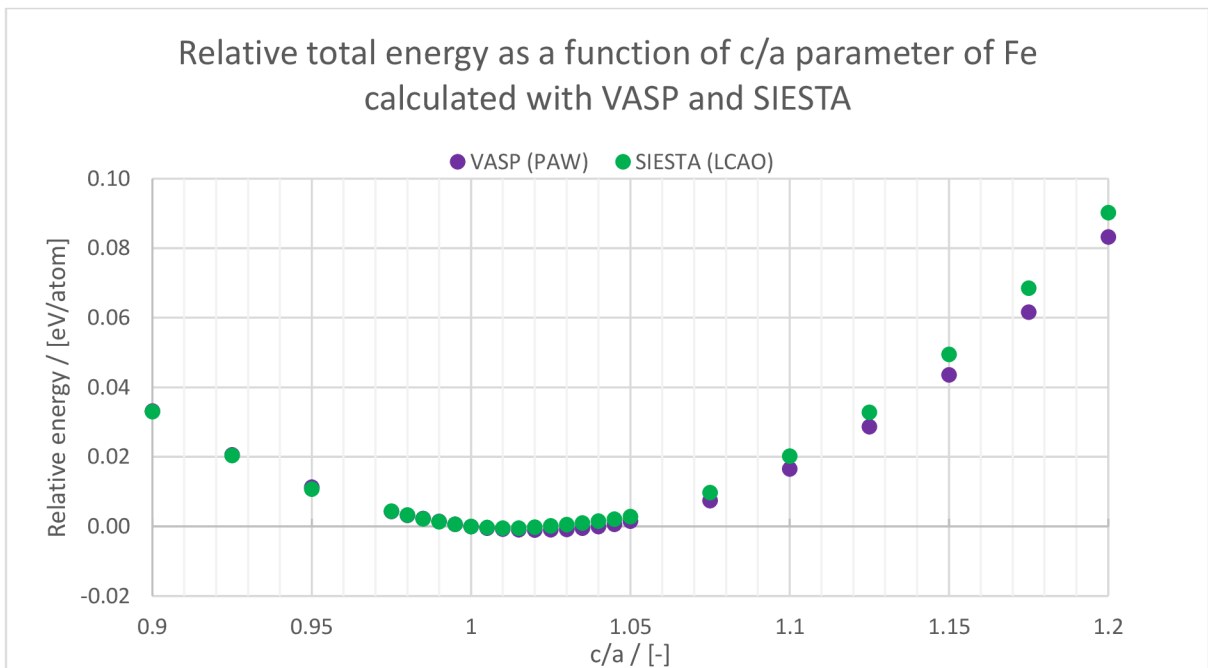


Figure 22 Tetragonal deformation path for γ -TiAl

Similarly to the calculations of energy-volume curves, the paramagnetic nature of γ -TiAl makes portraying the dependence of magnetic moment on tetragonal deformation unnecessary.

7.2.3 Ni₂MnGa

The tetragonal deformation path of Ni₂MnGa is represented in Figure 23. Total energy was calculated for 19 different c/a parameters, from $c/a = 0.9$ to $c/a = 1.35$. Total energies were then recalculated with respect to the energy of the austenitic structure (cubic L2₁ structure, $c/a = 1$). It depicts a stable martensitic phase at tetragonal deformation $c/a \approx 1.275$ for both VASP and SIESTA calculations. It is apparent that SIESTA reports a much more stable NM martensitic structure compared to the VASP. The austenitic structure reported by SIESTA exhibits considerably unusual energetic behaviour. The reasons behind such behaviour are currently unknown, therefore additional analysis should be considered.

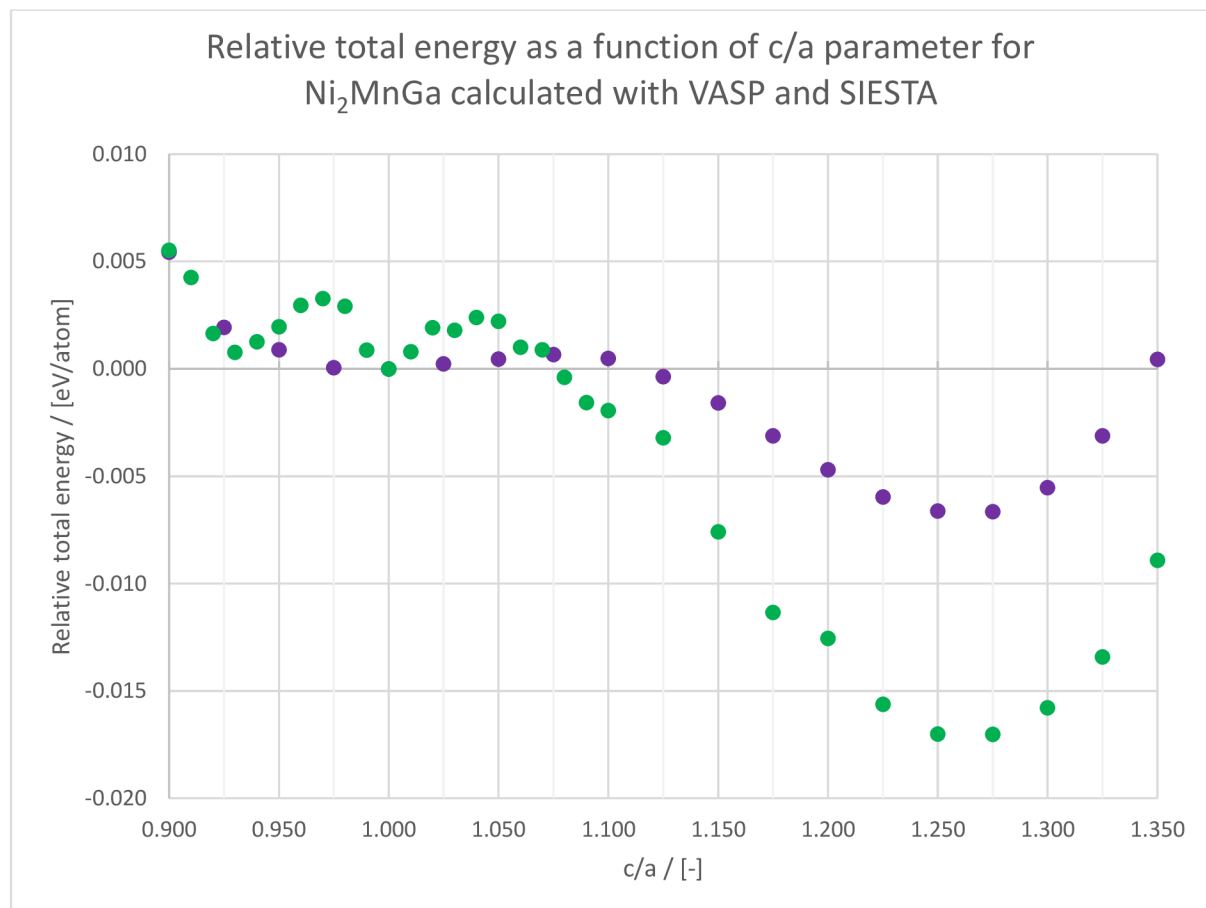


Figure 23 Tetragonal deformation path for Ni₂MnGa

The dependence of magnetic moments per atom on tetragonal deformation of Ni₂MnGa calculated with VASP and SIESTA is depicted in Figure 24. It shows that SIESTA calculations report higher magnetic moments compared to the VASP calculations while copying roughly the same path.

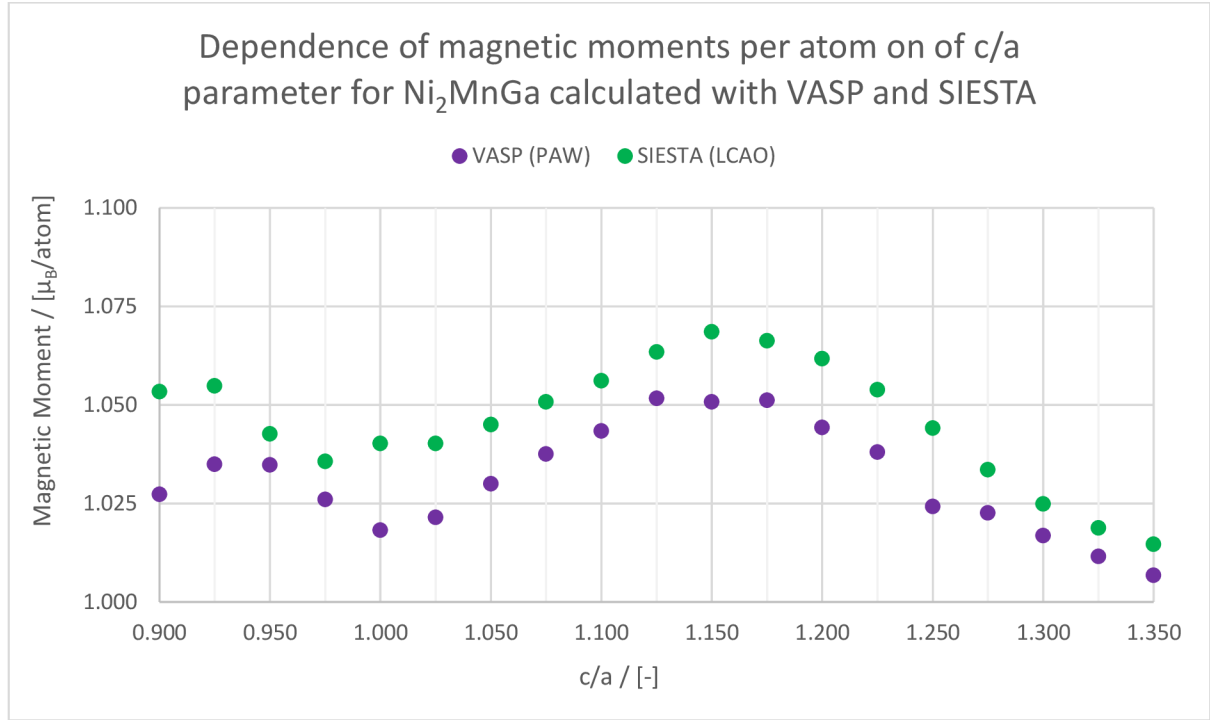


Figure 24 Magnetic moments per atom as a function of tetragonal deformation for Ni₂MnGa

7.3 Elastic constants

Elastic constants are parameters for the description of elasticity, a fundamental property of crystalline materials, during elastic deformation. Elastic constants are described by a matrix, using Hook's law in its tensor form, or its simplified version, due to Voigt notation. Calculations of elastic constants for this thesis were performed using principles presented in a study by R. Yu et al. Principle of the calculation lies in the creation of twelve different deformed structures under predefined strain, for which stress and strain tensors are calculated. From these stress and strain tensors, elastic constants are then determined.

Bulk modulus is a measure of resistance to compressive straining. It is one of the properties possible to obtain from the elastic constants. The formula for bulk modulus of cubic materials is:

$$B = \frac{C_{11} + 2C_{12}}{3}, \quad (7.1)$$

while formula for bulk modulus of tetragonal materials is: [64]

$$B = \frac{1}{9}(2C_{11} + 2C_{12} + 4C_{13} + C_{33}), \quad (7.2)$$

7.3.1 Fe

Elastic constants for Fe were calculated using scripts in the Appendix. Due to the cubic nature of Fe, only three independent elastic constants were expected, while all the other elements of the elasticity matrix should remain zero. However results from both VASP and SIESTA calculations show that these elements do not remain zero, which is caused by the inaccuracies in the calculations of stress tensors. Nevertheless, this is not an issue if the

variations from zero are relatively minor, which is not the case with the results from SIESTA, as the variations of the zero elements are far greater than in the results from the VASP. These variations and the accuracy of the stress tensor calculations can be influenced by the strain used in the calculations. Therefore, the dependence of elasticity matrices and subsequently elastic constants on the strain was investigated. Calculations of elastic constants in VASP and SIESTA were performed for five different strains. The variations on zero elements were then compiled into Figures 25 and 26. Examples of elasticity matrices in dependence on the strain, namely relative strain 0.01 and 0.03, are listed in Table 10.

Strain = 0.01					
VAPS (PAW) method					
239.39	153.96	154.21	0.32	-0.96	0.59
153.96	238.83	155.27	-0.31	1.20	1.50
154.21	155.27	240.16	-0.20	0.91	1.63
0.32	-0.31	-0.2	82.97	0.78	-0.70
-0.96	1.20	0.91	0.78	83.91	0.33
0.59	1.50	1.63	-0.70	0.33	85.46
SIESTA (LCAO) method					
272.156	158.028	141.979	29.810	4.727	37.019
158.028	314.520	162.046	1.348	20.041	49.038
141.979	162.046	298.496	13.549	23.743	56.307
29.810	1.348	13.549	112.121	-4.324	-26.439
4.727	20.041	23.743	-4.324	134.383	-25.565
37.019	49.038	56.307	-26.439	-25.565	106.154
Strain = 0.03					
VAPS (PAW) method					
267.65	150.21	152.94	0.30	-1.33	0.17
150.21	267.95	156.81	-1.46	1.25	1.02
152.94	156.81	271.05	-0.74	1.60	2.38
0.30	-1.46	-0.74	95.87	2.14	0.34
-1.33	1.25	1.60	2.14	96.19	0.90
0.17	1.02	2.38	0.34	0.90	97.9
SIESTA (LCAO) method					
257.867	111.66	100.789	2.648	-0.435	3.325
111.660	265.665	100.328	14.300	0.103	12.862
100.789	100.328	246.197	5.361	-4.533	-12.176
2.648	14.300	5.361	96.532	-2.362	1.671
-0.435	0.103	-4.533	-2.362	74.572	6.183
3.325	12.862	-12.176	1.671	6.183	93.483

Table 10 Example of elasticity matrices obtained for different strain calculated with VASP and SIESTA

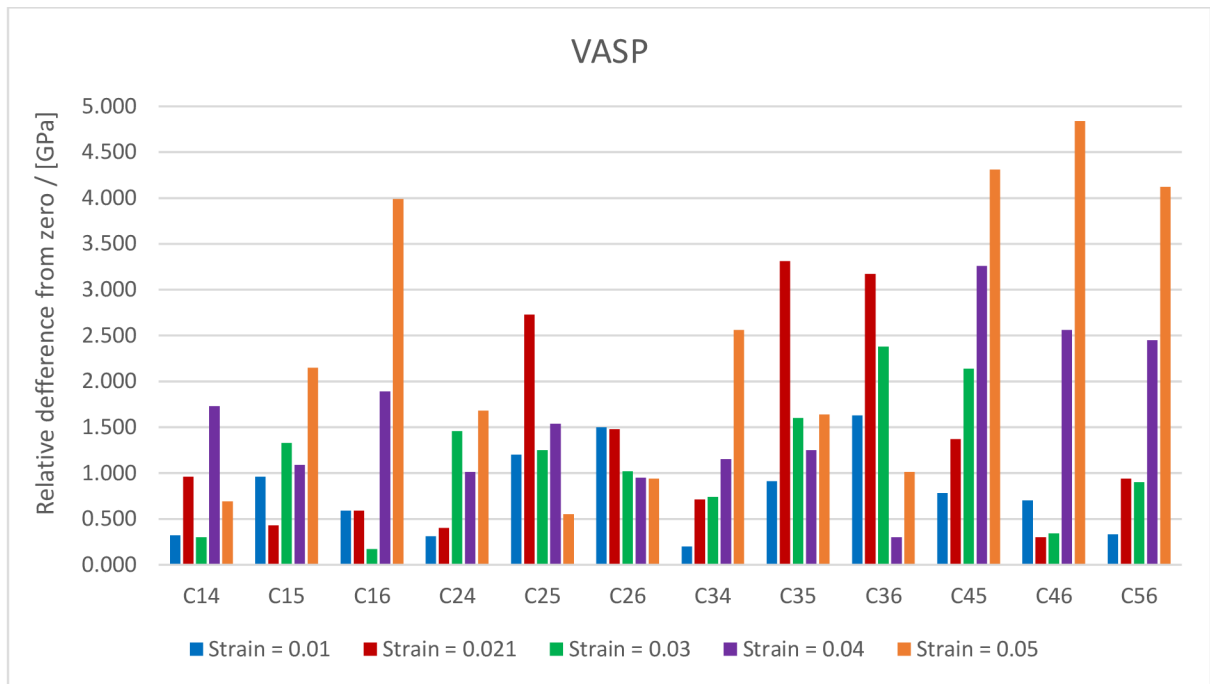


Figure 25 Comparison of relative differences of zero members for different strains obtained from VASP calculations

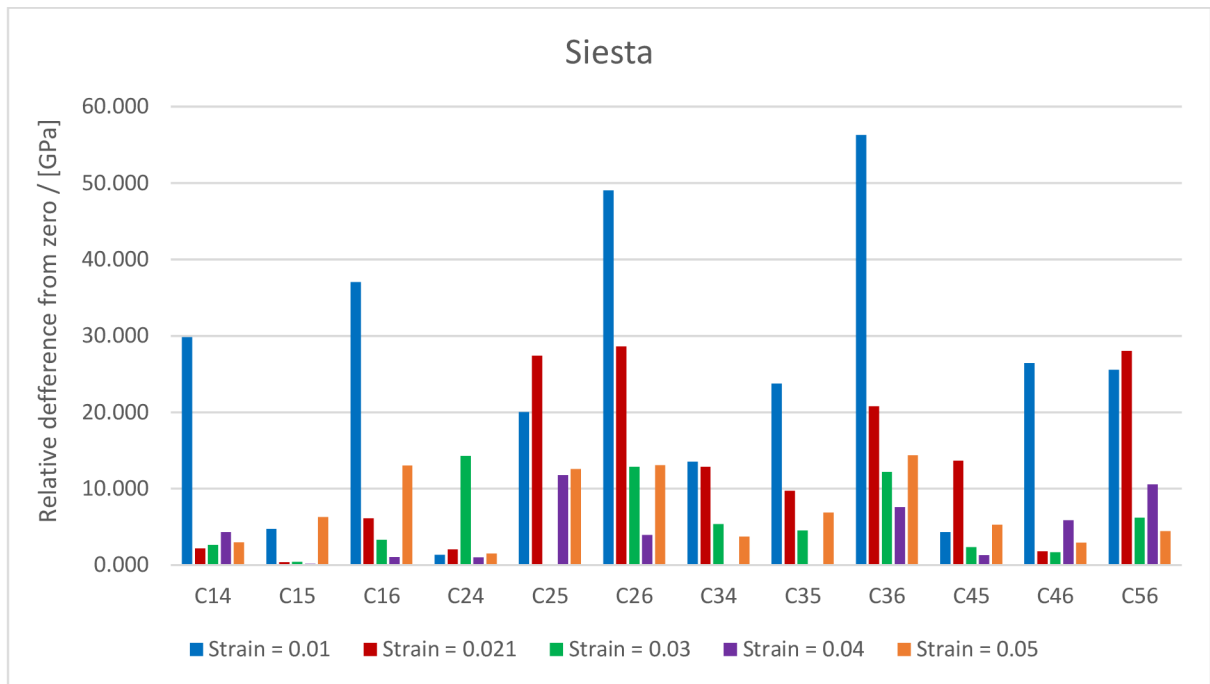


Figure 26 Comparison of relative differences of zero members for different strains obtained from SIESTA calculation

From Figure 25 it is apparent that calculation of elastic constants using VASP benefits from lower strains, while in contrast calculations of elastic constants using SIESTA can benefit from higher strains. Elastic constants calculated using VASP and SIESTA for all different strains, together with experimental elastic constants are listed in Table 11. Experimental elastic constants of iron were measured using resonant ultrasound spectroscopy over the temperature range 3–500 K by J. J. Adams et al. [65]

	C₁₁ [GPa]	C₁₂ [GPa]	C₄₄ [GPa]
Experimental [60]	239.55	135.75	120.75
Strain = 0.01			
VASP (PAW)	239.46	154.48	84.11
SIESTA (LCAO)	295.06	154.02	117.55
Strain = 0.021			
VASP (PAW)	261.49	155.67	92.57
SIESTA (LCAO)	278.34	102.97	64.24
Strain = 0.03			
VASP (PAW)	268.88	153.32	96.65
SIESTA (LCAO)	256.58	104.26	88.20
Strain = 0.04			
VASP (PAW)	265.62	150.94	100.82
SIESTA (LCAO)	261.47	106.86	79.50
Strain = 0.05			
VASP (PAW)	263.71	150.04	104.92
SIESTA (LCAO)	260.86	115.66	84.04

Table 11 Table of experimental elastic constants and elastic constants for Fe calculated with VASP and SIESTA for all strains

A comparison of calculated elastic constants to experimental elastic constants shows that both methods overestimate C_{11} , with the only exception of VASP at strain = 0.01. Furthermore, it shows that element C_{12} is overestimated by VASP, while being underestimated by SIESTA, with the only exception at strain = 0.01 where it is once again overestimated. Additionally, the last elastic constant C_{44} appears to be underestimated by both VASP and SIESTA.

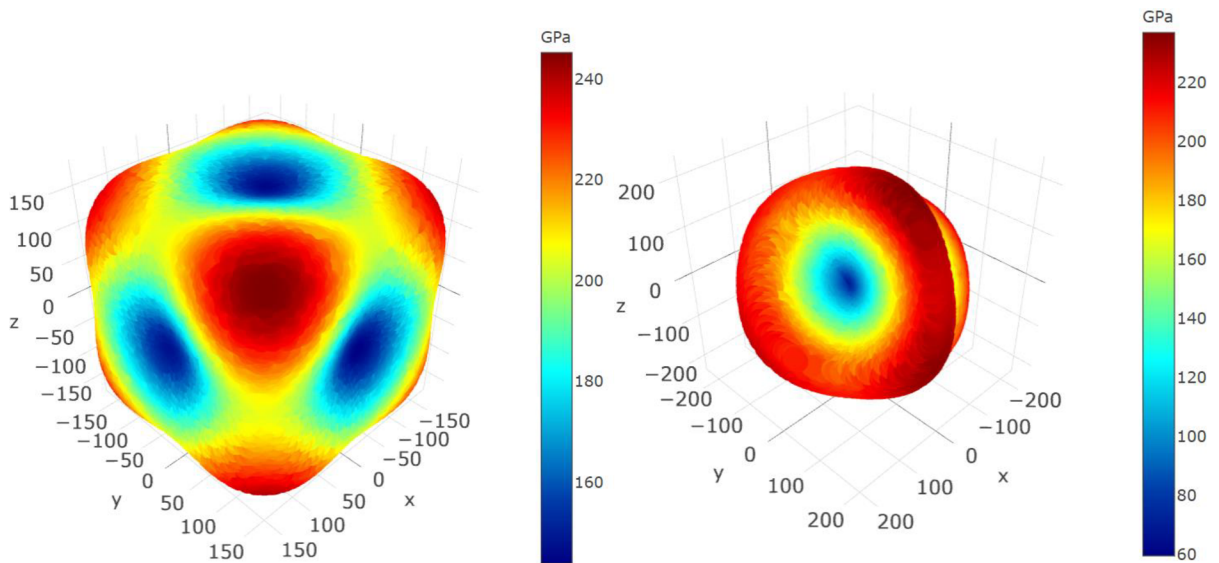


Figure 27 Comparison of spatial dependence of Young's modulus for elastic constants of Fe calculated using VASP (left) and SIESTA (right) for strain = 0.021 [70]

From the elastic constants in Table 11, a spatial dependence of Young's modulus of Fe was calculated using The MELASA software tool. [70] Results are depicted and compared in Figure 27. The spatial representation of Young's modulus obtained from VASP closely resembles the spatial representation of Young's modulus reported by J. J. Adams et al. [65]. However, that can not be said about the spatial representation of Young's modulus from SIESTA, where the shape of Young's modulus seems to be drastically different.

	Bulk Modulus [GPa]				
	Strain = 0.01	Strain = 0.021	Strain = 0.03	Strain = 0.04	Strain = 0.05
VASP (PAW)	182.81	190.94	191.84	189.17	187.93
SIESTA (LCAO)	201.03	161.43	155.03	158.40	164.06

Table 12 Table of calculated bulk moduli from elastic constants for Fe obtained from VASP and SIESTA

From the elastic constants listed in Table 11, bulk moduli for both methods and all the strains were calculated. The resulting bulk moduli are listed in Table 12. Bulk modulus obtained experimentally, reported J. J. Adams et al. [65], is $B = 170.35$ GPa. Results show that all bulk moduli from VASP are moderately overestimated, while bulk moduli from SIESTA are slightly underestimated, for all but strain = 0.01. [65, 66]

7.3.2 γ -TiAl

Calculations of elastic constants for γ -TiAl using VASP, and SIESTA were performed, and the results are listed in Table 13. Experimental elastic constants were measured using the RPR method in the temperature range 4–298 K by K. Tanaka et al. [67] From Table 13 it is noticeable that elastic constants obtained from SIESTA calculations match the experimental elastic constants better than the constants obtained from the VASP calculations. However, both sets of elastic constants can be considered viable.

	C₁₁ [GPa]	C₁₂ [GPa]	C₁₃ [GPa]	C₃₃ [GPa]	C₄₄ [GPa]	C₆₆ [GPa]
Experimental[]	186.00	72.00	74.00	176.00	101.00	77.00
VASP (PAW)	168.77	89.80	85.47	170.54	112.06	71.06
SIESTA (LCAO)	177.36	90.08	87.35	165.04	113.99	79.60

Table 13 Table of elastic constants for γ -TiAl calculated with VASP and SIESTA

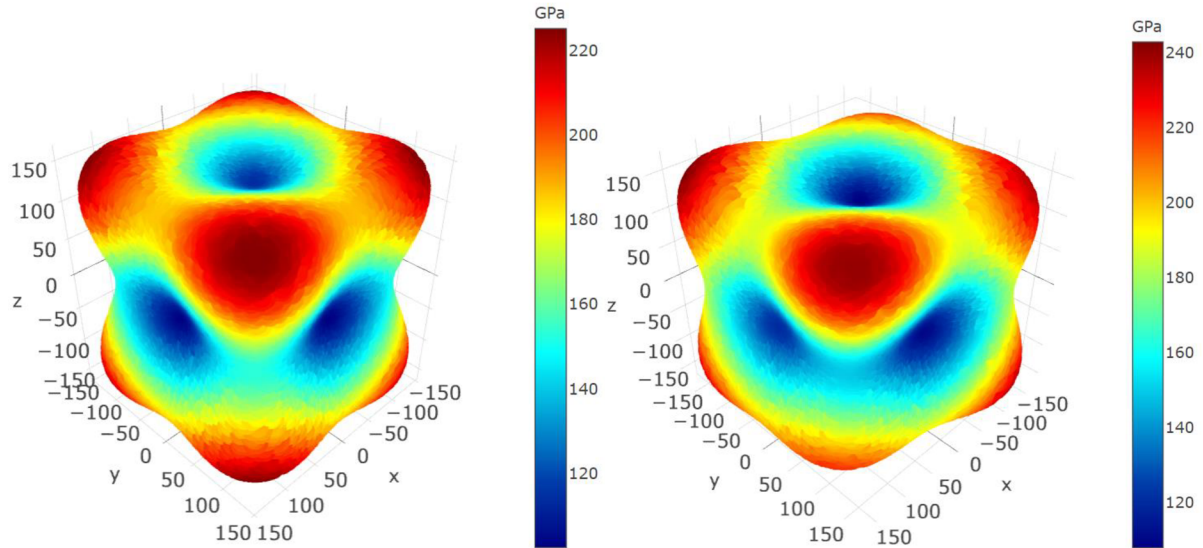


Figure 28 Comparison of spatial dependence of Young's modulus for elastic constants of γ -TiAl calculated using VASP (left) and SIESTA (right) [70]

From the elastic constants in Table 13, a spatial dependence of Young's modulus of γ -TiAl was calculated using the MELASA software tool [70]. Results are depicted and compared in Figure 28. It can be stated that spatial representations of Young's modulus calculated with VASP, and SIESTA are fairly similar as opposed to the drastic difference of Fe previously depicted in Figure 27.

	Bulk Modulus [GPa]
Experimental [61]	112.00
VASP (PAW)	114.40
SIESTA (LCAO)	116.59

Table 14 Table of calculated bulk moduli from elastic constants for γ -TiAl obtained from VASP and SIESTA

From the elastic constants listed in Table 13, bulk moduli for both methods were calculated. The resulting bulk modulus is listed in Table 14. It is apparent that the bulk modulus from both methods is slightly overestimated compared to the experimentally obtained bulk modulus by K. Tanaka et al. [67]

7.3.3 Ni_2MnGa

Calculations of elastic constants for austenitic Ni_2MnGa using VASP, and SIESTA were performed, and the results are listed in Table 15. Experimental elastic constants were measured using the ultrasonic pulse echo technique by J. Worgull et al. [68]

	C₁₁ [GPa]	C₁₂ [GPa]	C₄₄ [GPa]
Experimental [62]	152.00	143.00	103.00
VASP (PAW)	169.47	159.69	113.97
SIESTA (LCAO)	168.89	169.73	107.40

Table 15 Table of elastic constants for austenitic Ni₂MnGa calculated with VASP and SIESTA

Both sets of elastic constants can be considered slightly overestimated. However, results from SIESTA point to the instability of the crystals due to $C_{11} < C_{12}$. Therefore, owing to the apparent mechanical instability of a crystal, a set of elastic constants for martensitic Ni₂MnGa was also calculated. Results for NM martensitic Ni₂MnGa are listed in Table 17.

	Bulk Modulus [GPa]
Experimental [62]	146.00
VASP (PAW)	163.13
SIESTA (LCAO)	169.45

Table 16 Table of calculated bulk moduli from elastic constants for austenitic Ni₂MnGa obtained from VASP and SIESTA

From the elastic constants listed in Table 15, bulk moduli for both methods were calculated. The resulting bulk modulus is listed in Table 16. It is apparent that bulk moduli from both methods are overestimated compared to the experimentally obtained bulk modulus by J. Worgull et al. [68]

The set of elastic constants for NM martensitic Ni₂MnGa is listed in Table 18. Elastic constants for comparison were calculated by VASP with the help of the energy-strain method by S. Ozdemir Kart et al. [69] show that elastic constants obtained by VASP match the calculated constants better than the elastic constants obtained by SIESTA, which could be once again considered slightly overestimated.

	C11 [GPa]	C12 [GPa]	C13 [GPa]	C33 [GPa]	C44 [GPa]	C66 [GPa]
Calculated [63]	252.00	74.00	144.00	194.00	100.00	55.00
VASP (PAW)	246.24	76.34	148.74	191.58	99.37	53.40
SIESTA (LCAO)	285.46	67.01	142.87	223.95	98.97	43.88

Table 17 Table of elastic constants for martensitic Ni₂MnGa calculated with VASP and SIESTA

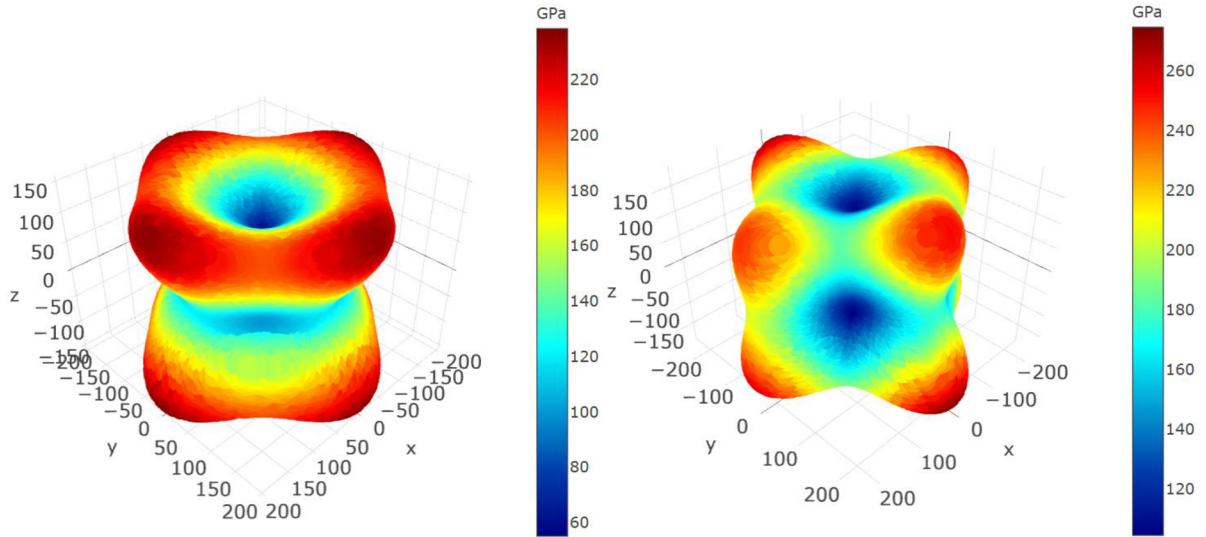


Figure 29 Comparison of spatial dependence of Young's modulus for elastic constants of Ni_2MnGa calculated using VASP (left) and SIESTA (right) [70]

From the elastic constants in Table 17, a spatial dependence of Young's modulus of Ni_2MnGa was calculated using the MELASA software tool. [70] Results are depicted and compared in Figure 29. It is apparent that there is a significant disparity between the spatial representations of Young's moduli from VAPS and SIESTA.

Bulk Modulus [GPa]	
Calculated	158.00
VASP (PAW)	159.08
SIESTA (LCAO)	166.71

Table 18 Table of calculated bulk moduli from elastic constants for martensitic Ni_2MnGa obtained from VASP and SIESTA

From the elastic constants listed in Table 17, bulk moduli for both methods were calculated. The resulting bulk modulus is listed in Table 18. Results from VASP are closely comparable to the bulk modulus calculated by S. Ozdemir Kart et al. [69], while results from SIESTA are slightly overestimated.

8 Conclusion

The influence of different quantum-mechanical descriptions of the interatomic interactions in DFT-based ab initio calculations was investigated in this work. In particular, the highly localized wave functions described by the linear combination of atomic orbitals (LCAO) implemented within the *Spanish Initiative for Electronic Simulations with Thousands of Atoms* (SIESTA) simulation package were compared with the delocalized wave functions described by the projector-augmented wave method implemented within *Vienna ab initio simulation package* (VASP). The influence of these two approaches was explored on three different materials: Fe, γ -TiAl, and Ni₂MnGa, for which the dependence of magnetic moments on the deformation paths, namely the tetragonal deformation path and the volume deformation, were calculated, as well as the elastic properties described by stiffness tensor.

The dependence of magnetic moments on the volume deformation path showed relatively minor divergence across these two calculation approaches, as both methods reported highly comparable results. A slight change occurred in the calculations of the dependence of magnetic moments on tetragonal deformation paths, where SIESTA reported slightly overestimated magnetic moments compared to the VASP calculations. In the calculation of the tetragonal deformation path for iron, it also demonstrated that SIESTA prefers the high spin state, exhibiting a higher magnetic moment, which is kept during the whole deformation path, while VASP automatically finds a low spin state, with a low magnetic moment, as energetically more favourable. The calculation of the tetragonal deformation path for Ni₂MnGa showed that SIESTA reports significantly more stable tetragonal NM martensite compared to the results from the VASP calculations.

Elastic constants were calculated using the stress-strain method presented by R. Yu et al. for all three materials. The results show us that the elastic constants for Ni₂MnGa and γ -TiAl calculated by VASP, and SIESTA are relatively comparable with the experimental or calculated elastic constants for both materials with fairly minimal variations. However, elastic constants for Fe calculated by SIESTA exhibit greater discrepancies in comparison to the experimental elastic constants or even elastic constants calculated by VASP. This is caused by the nature of atomic bonds within the materials. Both Ni₂MnGa and γ -TiAl exhibit higher proportionality of covalent bonds, which is quite suitable for the LCAO method. While highly delocalized electrons of iron necessitate the use of plane-wave approaches.

Furthermore, calculations of spatial representation of Young's moduli from calculated stiffness tensors showed a significant disparity between the methods, especially in the case of Fe. This points to the unsuitability of the SIESTA for the calculations of elastic constants using the stress-strain approach. More conventional energy-strain approaches could yield more appropriate results, however, that is the subject of further study.

The results from SIESTA could become clearer with the further tuning of the simulation settings. This was however beyond the scope of this thesis. Therefore, VASP proved itself to be much safer in this regard, as finding the correct simulation settings is far simpler. As such the use of VASP is in these cases more preferable, despite being more computationally challenging.

References

- [1] TADMOR, E., and R. MILLER. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. 1. New York: Cambridge University Press, 2011. ISBN 978-0-521-85698-0.
- [2] MARTIN, R. (2004). *Electronic Structure: Basic Theory and Practical Methods*. Cambridge: Cambridge University Press. ISBN-10: 0521782856 doi: <https://doi.org/10.1017/CBO9780511805769>
- [3] SCHRÖDINGER, E. An Undulatory Theory of the Mechanics of Atoms and Molecules. *Phys. Rev.* 1926, **28**(6), 1049-1070. doi: <https://doi.org/10.1103/PhysRev.28.1049>
- [4] BORN, M. and R. OPPENHEIMER. Zur Quantentheorie der Molekeln. *Annalen der Physik.* 1927, **389**(20), 457-484. doi: <https://doi.org/10.1002/andp.19273892002>
- [5] BLINDER, S. M. Chapter 14 - Density functional theory. *Introduction to Quantum Mechanics (Second edition)*. Academic Press, 2021, s. 235-244. ISBN 978-0-12-822310-9. doi: <https://doi.org/10.1016/B978-0-12-822310-9.00022-7>
- [6] KURTH, S., M. A. L. MARQUES, and E. K. U. GROSS. Density-Functional Theory. *Encyclopedia of Condensed Matter Physics*. 2005, 395-402. doi: <https://doi.org/10.1016/B0-12-369401-9/00445-9>
- [7] HOHENBERG, P. and W. KOHN. Inhomogeneous Electron Gas. *Phys. Rev.* 1964, **136**(3B), B864-B871. doi: <https://doi.org/10.1103/PhysRev.136.B864>
- [8] KOHN, W., and L. J. SHAM. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* 1965, **140**(4A), A1133-A1138. doi: <https://doi.org/10.1103/PhysRev.140.A1133>
- [9] PERDEW, J. P., K. BURKE, and M. ERNZERHOF. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* 1996, **77**(18), 3865-3868. doi: <https://doi.org/10.1103/PhysRevLett.77.3865>
- [10] KITTEL, Ch. *Úvod do fyziky pevných látek : Introduction to solid state physics (Orig.)*. 1. Vydání. Praha: Academia, 1985.
- [11] YU, V. W., J. MOUSSA, and V. BLUM. Accurate frozen core approximation for all-electron density-functional theory. *J. Chem. Phys.* 2021, **154**(22), 224107. doi: <https://doi.org/10.1063/5.0050296>

- [12] BLÖCHL, P. E. Projector augmented-wave method. *Phys. Rev. B*. American Physical Society, 1994, **50**(24), 17953-17979. doi: <https://doi.org/10.1103/PhysRevB.50.17953>
- [13] KRESSE, G. a D. JOUBERT. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B*. 1999, **59**(3), 1758-1775. doi: <https://doi.org/10.1103/PhysRevB.59.1758>
- [14] HOBBS, D., G. KRESSE, and J. HAFNER. Fully unconstrained noncollinear magnetism within the projector augmented-wave method. *Phys. Rev. B*. 2000, **62**(17), 11556-11570. doi: <https://doi.org/10.1103/PhysRevB.62.11556>
- [15] BLÖCHL, P. E., J. KÄSTNER, and C. J. FÖRST. Electronic Structure Methods: Augmented Waves, Pseudopotentials and The Projector Augmented Wave Method. YIP, S. *Handbook of Materials Modeling*. Springer, Dordrecht, 2005, s. 93-119. ISBN 978-1-4020-3286-8. doi: https://doi.org/10.1007/978-1-4020-3286-8_7
- [16] LARSEN, A. H., M. VANIN, and J. J. MORTENSEN. Localized atomic basis set in the projector augmented wave method. *Phys. Rev. B*. 2009, **80**(19), 195112-195122. doi: <https://doi.org/10.1103/PhysRevB.80.195112>
- [17] SOLER, J. M., E. ARTACHO, and J. D. GALE. The SIESTA method for ab initio order-N materials simulation. *Journal of Physics: Condensed Matter*. 2002, **14**(11), 2745-2779. doi: <http://doi.org/10.1088/0953-8984/14/11/302>
- [18] LARSEN, A. H. *Localized Atomic Orbital Basis Sets in the Projector Augmented Wave Method*. Center for Atomic-scale Materials Design Department of Physics Technical University of Denmark, 2008. Master's thesis. DTU, Lyngby. Supervisors: K. S. Thygesen and J. J. Mortensen.
- [19] CCoil. *Dihydrogen MO Diagram* [online]. In: . 02.03. 2008 [cited 2022-05-24]. Available from: <https://commons.wikimedia.org/wiki/File:Dihydrogen-MO-Diagram.svg>
- [20] FEYNMAN, R. P. Forces in Molecules. *Phys. Rev.* 1939, **56**(340), 340-343. doi: <https://doi.org/10.1103/PhysRev.56.340>
- [21] NOVÁK, M., J. VACKÁŘ, and R. CIMRMAN. Evaluating Hellmann–Feynman forces within non-local pseudopotentials. *Computer Physics Communications*. 2020, **250**, 107034-107044. doi: <https://doi.org/10.1016/j.cpc.2019.107034>
- [22] COTTRELL, A. Elastic Behavior of Materials: Physical Basis. *Encyclopedia of Materials: Science and Technology (Second Edition)*. 2001, 2404-2408. doi: <https://doi.org/10.1016/B0-08-043152-6/00426-5>

- [23] BOWMAN, K. Mechanical Properties: Elastic Behavior. *Encyclopedia of Condensed Matter Physics*. 2005, 286-291. doi: <https://doi.org/10.1016/B0-12-369401-9/00568-4>
- [24] VOIGT, W. *Lehrbuch der Kristallphysik (mit Ausschluss der Kristalloptik)*. 1. Springer Fachmedien Wiesbaden, 1966. ISBN 978-3-663-15884-4. doi: <https://doi.org/10.1007/978-3-663-15884-4>
- [25] NYE, J. F. *Physical Properties Of Crystals: Their Representation by Tensors and Matrices*. Illustrated, reprint. Oxford University Press, U.S.A.; Reprint edition (20 Jun. 2002), 2002. ISBN 978-0198511656.
- [26] YU, R., J. ZHU, and H. Q. YE. Calculations of single-crystal elastic constants made simple. *Computer Physics Communications*. 2010, **181**(3), 671-675. doi: <https://doi.org/10.1016/j.cpc.2009.11.017>
- [27] FRIÁK, M., M. ŠOB, and V. VITEK. Ab initio calculation of phase boundaries in iron along the bcc-fcc transformation path and magnetism of iron overlayers. *Phys. Rev. B*. 2000, **63**(5), 052405. doi: <https://doi.org/10.1103/PhysRevB.63.052405>
- [28] ZHANG, L., M. ŠOB, and Z. WU. Characterization of iron ferromagnetism by the local atomic volume: from three-dimensional structures to isolated atoms. *Journal of Physics: Condensed Matter*. 2014, **26**, 086002. doi: <https://doi.org/10.1088/0953-8984/26/8/086002>
- [29] SOULAIROL, R., Ch. Ch. FU, and C. BARRETEAU. Structure and magnetism of bulk Fe and Cr: from plane waves to LCAO methods. *Journal of Physics: Condensed Matter*. 2010, **22**, 295502. doi: <https://doi.org/10.1088/0953-8984/22/29/295502>
- [30] HALBAUER, R. and U. GONSER. Antiferromagnetism of fcc iron films. *Journal of Magnetism and Magnetic Materials*. 1983, **35**(1-3), 55-56. doi: [https://doi.org/10.1016/0304-8853\(83\)90454-7](https://doi.org/10.1016/0304-8853(83)90454-7)
- [31] LIU, B., and Y. LIU. 27 - Powder metallurgy titanium aluminide alloys. QIAN, M. a F. H. FROES. *Titanium Powder Metallurgy*. Butterworth-Heinemann, 2015, s. 515-531. ISBN 978-0-12-800054-0. doi: <https://doi.org/10.1016/B978-0-12-800054-0.00027-7>
- [32] MPHAHLELE, M. R., E. A. OLEVSKY, and P. A. OLUBAMBI. Chapter 12 - Spark plasma sintering of near net shape titanium aluminide: A review. CAO, G., C. ESTOURNÈS, J. GARAY, and R. ORRÙ. *Spark Plasma Sintering*. Elsevier, 2019, s. 281-299. ISBN 978-0-12-817744-0. doi: <https://doi.org/10.1016/B978-0-12-817744-0.00012-X>

- [33] COBBINAH, P., and W. MATIZAMHUKA. Solid-State Processing Route, Mechanical Behaviour, and Oxidation Resistance of TiAl Alloys. *Advances in Materials Science and Engineering*. 2019, **2019**(10), 4251953. doi: <https://doi.org/10.1155/2019/4251953>
- [34] DUAN, B., Y. YANG, S. HE, Q. FENG, et al. History and development of γ -TiAl alloys and the effect of alloying elements on their phase transformations. *Journal of Alloys and Compounds*. 2022, **909**, 164811. doi: <https://doi.org/10.1016/j.jallcom.2022.164811>
- [35] RUICHENG, F., M. WANG, H. LI, Q. YONGNIAN, Q. WANG, and Z. RUI. Micromechanism of Cold Deformation of Two-Phase Polycrystalline Ti–Al Alloy with Void. *Materials*. 2019, **12**(184), 30621116. doi: <https://doi.org/10.3390/ma12010184>
- [36] HUSSAIN, A., S. S. HAYAT and M.A. CHOUDHRY. Ab-initio calculations of electronic structure and optical properties of TiAl alloy. *Physica B: Condensed Matter*. 2011, **406**(10), 1961-1965. doi: <https://doi.org/10.1016/j.physb.2011.02.065>
- [37] MATAR, S. F., and J. ETOURNEAU. Investigation of the electronic structure of carbon-containing TiAl. *Journal of Alloys and Compounds*. 1996, **233**(1-2), 112-120. doi: [https://doi.org/10.1016/0925-8388\(96\)80042-9](https://doi.org/10.1016/0925-8388(96)80042-9)
- [38] GRAF, T., C. FELSER, and S. S. P. PARKIN. Simple rules for the understanding of Heusler compounds. : *Progress in Solid State Chemistry*. 1, 2011, (39), 1-50. doi: <https://doi.org/10.1016/j.progsolidstchem.2011.02.001>
- [39] WIRTH, S., A. LEITHE-JASPER, A. N. VASILEV and J. M. D. COEY. Structural and magnetic properties of Ni₂MnGa. *Journal of Magnetism and Magnetic Materials*. 1997, **167**(1-2), L7-L11. doi: [https://doi.org/10.1016/S0304-8853\(96\)00745-7](https://doi.org/10.1016/S0304-8853(96)00745-7)
- [40] HELM, D. and P. HAUPT. Shape memory behaviour: modelling within continuum thermomechanics. *International Journal of Solids and Structures*. 2003, **40**(4), 827-849. doi: [https://doi.org/10.1016/S0020-7683\(02\)00621-2](https://doi.org/10.1016/S0020-7683(02)00621-2)
- [41] COLLARDO, M., N. E. N. RODRIGUEZ, and R. CABÁS. CHARACTERIZATION RESULTS OF A NOVEL SHAPE MEMORY ALLOY AND PIN PULLER MECHANISMS BASED ON THIS TECHNOLOGY. *ESMATS - European Space Mechanisms And Tribology Symposium*. 2011, **14**, 289. doi: <https://doi.org/10.13140/2.1.4353.0246>
- [42] HECZKO, O. Magnetic shape memory effect and highly mobile twin boundaries. *Materials Science and Technology*. 2013, **30**(13), 1559-1578. doi: <https://doi.org/10.1179/1743284714Y.0000000599>

- [43] HECZKO, O., N. SCHEERBAUM, and O. GUTFLEISCH. Magnetic Shape Memory Phenomena. GUTFLEISCH, O., E. FULLERTON, O. GUTFLEISCH, and D. SELLMYER. *Nanoscale Magnetic Materials and Applications*. Springer, Boston, MA, 2009, s. 399-439. ISBN 978-0-387-85600-1. doi: https://doi.org/10.1007/978-0-387-85600-1_14
- [44] SOZINOV, A., N. LANSKA, A. SOROKA, and W. ZOU. 12% magnetic field-induced strain in Ni-Mn-Ga-based non-modulated martensite. *Appl. Phys. Lett.* 2013, **102**(2), 021902. doi: <https://doi.org/10.1063/1.4775677>
- [45] GRUNER, M. E. and P. ENTEL. TOPICAL REVIEW: Simulating functional magnetic materials on supercomputers. *Journal of Physics: Condensed Matter*. 2009, **21**(29), 293201. doi: <https://doi.org/10.1088/0953-8984/21/29/293201>
- [46] ULLAKKO, K., J. K. HUANG, and V. V. KOKORIN. Magnetically controlled shape memory effect in Ni₂MnGa intermetallics. *Scripta Materialia*. 1997, **36**(10), 1133-1138. doi: [https://doi.org/10.1016/S1359-6462\(96\)00483-6](https://doi.org/10.1016/S1359-6462(96)00483-6)
- [47] BARMAN, S. R., S. BANIK and A. CHAKRABARTI. Structural and electronic properties of Ni₂MnGa. *Phys. Rev. B*. 2005, **72**(18), 184410. doi: <https://doi.org/10.1103/PhysRevB.72.184410>
- [48] BRUNO, N. M., S. WANG, I. KARAMAN a Y. I. CHUMLYAKOV. Reversible Martensitic Transformation under Low Magnetic Fields in Magnetic Shape Memory Alloys. *Scientific Reports*. 2017, **7**, 40434. doi: <https://doi.org/10.1038/srep40434>
- [49] ZELENÝ, M., L. STRAKA, A. SOZINOV and O. HECZKO. Transformation Paths from Cubic to Low-Symmetry Structures in Heusler Ni₂MnGa Compound. *Scientific Reports volume*. 2018, **8**, 7275. doi: <https://doi.org/10.1038/s41598-018-25598-z>
- [50] LARSEN, A. H., J. J. MORTENSEN, J. BLOMQVIST, I. E CASTELLI, et al. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter*. 2017, **29**(7), 273002. doi: <https://doi.org/10.1088/1361-648X/aa680e>
- [51] BROYDEN, C. G. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*. 1970, **6**(1), 76–90. doi: <https://doi.org/10.1093/imamat/6.1.76>
- [52] FLETCHER, R. A new approach to variable metric algorithms. *The Computer Journal*. 1970, **13**(3), 317–322. doi: <https://doi.org/10.1093/comjnl/13.3.317>

- [53] GOLDFARB, D. A family of variable-metric methods derived by variational means. *Math. Comp.* 1970, **24**, 23-26. doi: <https://doi.org/10.1090/S0025-5718-1970-0258249-6>
- [54] SHANNO, D. F. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.* 1970, **24**, 647-656. doi: <https://doi.org/10.1090/S0025-5718-1970-0274029-X>
- [55] KRESSE, G. and J. FURTHMÜLLER. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B.* 1996, **54**(16), 11169-11186. doi: <https://doi.org/10.1103/PhysRevB.54.11169>
- [56] KRESSE, G. and J. FURTHMÜLLER. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science.* 1, 1996, (6), 15-50. doi: [https://doi.org/10.1016/0927-0256\(96\)00008-0](https://doi.org/10.1016/0927-0256(96)00008-0)
- [57] GARCÍA, A., N. PAPIOR, A. AKHTAR, E. ARTACHO, V. BLUM, et al. Siesta: Recent developments and applications. *J. Chem. Phys.* 2020, **152**(20), 204108. doi: <https://doi.org/10.1063/5.0005077>
- [58] KRESSE, G., and J. HAFNER. Norm-conserving and ultrasoft pseudopotentials for first-row and transition elements. *Journal of Physics: Condensed Matter.* 1994, **6**(40), 8245. doi: <https://doi.org/10.1088/0953-8984/6/40/015>
- [59] The PSML format and library for norm-conserving pseudopotential data curation and interoperability, A. García, M. Verstraete, Y. Pouillon, J. Junquera, *Computer Physics Communications* 227, 51-71 (2018) doi: <https://doi.org/10.1016/j.cpc.2018.02.011>
- [60] ROSSUM, Guido Van a Drake L. FRED. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 978-1-4414-1269-0. doi: <https://dl.acm.org/doi/book/10.5555/1593511>
- [61] GRIMVALL, G., B. MAGYARI-KOPE, V. OZOLINS and K. ASLAUG PERSSON. Lattice instabilities in metallic elements. *Review of Modern Physics.* 2012, **84**(2), 945-986. doi: <https://doi.org/10.1103/RevModPhys.84.945>
- [62] SCHÖNECKER, S. *Calculation of the Epitaxial Bain Path of Palladium*. Institute of theoretical physics, Department of Physics, Faculty of Mathematics and Natural Science, 2007. Master's thesis. Technical University of Dresden. Supervisor: H. Eschrig. doi: https://www.ifw-dresden.de/uploads/groups/itf_folder/Theses/stephan_schoenecker_diplom.pdf

- [63] ABDOSHAHI, N., P. SPOERK-ERDELY, M. FRIÁK, S. MAYER, M. ŠOB a D. HOLEC. Ab initio study of chemical disorder as an effective stabilizing mechanism of bcc-based TiAl (+ Mo). *Phys. Rev. Materials*. 2020, **4**(10), 103604. doi: <https://doi.org/10.1103/PhysRevMaterials.4.103604>
- [64] SHANG, S. L., A. SAENGDEEJING, Z. G. MEI, D. E. KIM, et al. First-principles calculations of pure elements: Equations of state and elastic stiffness constants. *Computational Materials Science*. 2010, **48**(4), 813-826. doi: <https://doi.org/10.1016/j.commatsci.2010.03.041>
- [65] ADAMS, J. J., D. S. AGOSTA, and R. G. LEISURE. Elastic constants of monocrystal iron from 3 to 500 K. *Journal of Applied Physics*. 2006, **100**(113530), 113530. doi: <https://doi.org/10.1063/1.2365714>
- [66] RAYNE, J. A., and B. S. CHANDRASEKHAR. Elastic Constants of Iron from 4.2 to 300°K. *Phys. Rev.* 1961, **122**(6), 1714. doi: <https://doi.org/10.1103/PhysRev.122.1714>
- [67] TANAKA, K. A. Single-crystal elastic constants of γ -TiAl. *Philosophical Magazine Letters*. 1996, **73**(2), 71-78. Dostupné z: doi: <https://doi.org/10.1080/095008396181019>
- [68] WORGULL, J., E. PETTI, and J. TRIVISONNO. Behavior of the elastic properties near an intermediate phase transition in Ni₂MnGa. *Phys. Rev. B*. 1996, **54**(22), 15695--15699. doi: <https://doi.org/10.1103/PhysRevB.54.15695>
- [69] OZDEMIR KART, S. a T. CAGIN. Elastic properties of Ni₂MnGa from first-principles calculations. *Journal of Alloys and Compounds*. 2010, **508**(1), 177-183. doi: <https://doi.org/10.1016/j.jallcom.2010.08.039>
- [70] FRIÁK, M., D. LAGO, N. KOUTNÁ, T. REBOK a M. ŠOB. Multi-phase ELAStic Aggregates (MELASA) software tool for modeling anisotropic elastic properties of lamellar composites. *Computer Physics Communications*. 2020, **247**(106863), 106863. doi: <https://doi.org/10.1016/j.cpc.2019.106863>
- [71] MOMMA, K. a F. IZUMI. VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data. *Journal of Applied Crystallography*. 2011, **44**, 1272-1276. doi: <https://doi.org/10.1107/S0021889811038970>
- [72] ZHOU, L., D. HOLEC and P. H. MAYRHOFER. First-principles study of elastic properties of cubic Cr_{1-x}Al_xN alloys. *Journal of Applied Physics*. 2013, **113**(4), 043511. doi: <https://doi.org/10.1063/1.4789378>

Symbols and abbreviations

DFT	Density functional theory
LDA	Local Density Approximation
GGA	Generalized Gradient Approximation
PBE	Pedrew-Burke-Ernzerhof
BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithm
PAW	Projector-Augmented Wave
VASP	Vienna Ab initio Simulation Package
LCAO	Linear Combination of Atomic Orbitals
SIESTA	Spanish Initiative for Electronic Calculations with Thousands of Atoms
ASE	Atomic Simulation Environment
GTO	Gaussian-Type Orbitals
STO	Slater-Type Orbitals
NAO	Numerical-Type Orbitals
PAO	Pseudo-Atomic Orbitals
SD	Steepest Decent
CG	Conjugate Gradient
ULICS	Universal Linear Independent Coupling Strains
BBC	Body-Centred Cubic
FCC	Face-Centred Cubit
BCT	Body-Centred Tetragonal
NM	non-modulated
SMA	Shape Memory Alloys
MSMA	Magnetic Shape Memory Alloys
MFIS	Magnetic Field Induced Strain
MIR	Magnetic Induced Reorientation
SCF	Self-Consistent Field
DM	density matrix
EDM	energy density matrix

Table 19 Table of abbreviations

Appendix

Appendix 1: Script prep_nonDiag.py

```
#####  
####This script was written as a practical part of master's thesis by Bc. Martin Čavojec. Purpose of this script is to create  
####twelve different deformed structures and calculate stress and strain tensors, which later serve as an input for  
####further calculations.  
####This code is based on works:  
####YU, R., J. ZHU a H. Q. YE. Calculations of single-crystal elastic constants made simple. Computer Physics  
####Communications. 2010, 181(3), 671-675. doi: https://doi.org/10.1016/j.cpc.2009.11.017  
####ZHOU, L., D. HOLEC and P. H. MAYRHOFER. First-principles study of elastic properties of cubic Cr1-xAlxN  
####alloys. Journal of Applied Physics. 2013, 113(4), 043511. doi: https://doi.org/10.1063/1.4789378  
#####  
  
import os  
import shutil  
import sys  
from distutils.dir_util import copy_tree  
from subprocess import call  
  
eps = 0.021      #set the degree of relative deformation  
maindir = os.getcwd()      #maindir - main directory where 'eq' folder should be  
os.chdir('eq')  
eqdir = os.getcwd()      #eqdir - 'eq' directory  
if os.path.exists('strain.dat'):  
    os.remove('strain.dat')  
  
#create the case folder  
os.chdir(maindir)  
casename=('eps=%s' % (eps))  
print('Preparing case %s in %s' % (casename, maindir))  
os.mkdir(casename)  
os.chdir(casename)  
casedir= os.getcwd()      #casedir - the 'eps=0.021' directory  
  
#creation of strain vector, using Voigt's notation  
for n in range(1, 7):  
    if n == 1:  
        eps1 = 1; eps2 = -2; eps3 = 3  
        eps4 = -4; eps5 = 5; eps6 = -6  
    elif n == 2:  
        eps1 = 2; eps2 = 1; eps3 = -5  
        eps4 = -6; eps5 = 4; eps6 = 3  
    elif n == 3:  
        eps1 = 3; eps2 = 4; eps3 = -1  
        eps4 = 5; eps5 = 6; eps6 = -2  
    elif n == 4:  
        eps1 = 4; eps2 = -3; eps3 = 6  
        eps4 = 1; eps5 = -2; eps6 = 5  
    elif n == 5:  
        eps1 = 5; eps2 = 6; eps3 = 2  
        eps4 = -3; eps5 = -1; eps6 = -4  
    elif n == 6:  
        eps1 = 6; eps2 = -5; eps3 = -4  
        eps4 = 2; eps5 = -3; eps6 = 1  
  
os.chdir(casedir)  
  
eps1="{:.8f}".format((eps*eps1)/6); eps2="{:.8f}".format((eps*eps2)/6); eps3="{:.8f}".format((eps*eps3)/6)  
eps4="{:.8f}".format((eps*eps4)/6); eps5="{:.8f}".format((eps*eps5)/6); eps6="{:.8f}".format((eps*eps6)/6)  
print(' -> strain vector eps = (%s, %s, %s, %s, %s, %s)' % (eps1, eps2, eps3, eps4, eps5, eps6))  
with open('strain.dat', 'a') as f:
```

```

f.writelines('%s %s %s %s %s %s\n' % (eps1, eps2, eps3, eps4, eps5, eps6))    #Write eps1-6 to strain.dat

eps1=float(eps1); eps2=float(eps2); eps3=float(eps3)
eps4=float(eps4); eps5=float(eps5); eps6=float(eps6)
#definition of the deformation matrix
eps4= float((eps4)/2.0); eps5= float((eps5)/2.0); eps6= float((eps6)/2.0)

d11=eps1; d12=eps6; d13=eps5
d21=eps6; d22=eps2; d23=eps4
d31=eps5; d32=eps4; d33=eps3

os.chdir(casedir)
caseeps=('eps%s' % (n))
os.mkdir(caseeps)
os.chdir(caseeps)
epsdir= os.getcwd()
files = os.listdir(eqdir)    #copying to $caseeps directory
copy_tree(eqdir, epsdir)
os.chdir(epsdir)
with open("INCAR", 'r') as f:    #editing of the INCAR
    get_all=f.readlines()
with open("INCAR", 'w') as f:    #removal of any ISIF, NSW and IBRION
    for i,line in enumerate(get_all,1):
        if "ISIF" in line:
            f.writelines("")
        elif "NSW" in line:
            f.writelines("")
        elif "IBRION" in line:
            f.writelines("")
        else:
            f.writelines(line)
with open('INCAR', 'a') as f:    #append ISIF, NSW and IBRION to INCAR
    f.writelines(' ISIF = 2\n')
    f.writelines(' NSW = 60\n')
    f.writelines(' IBRION = 1\n')

with open("run.sh", 'r') as f:
    get_all=f.readlines()
with open("run.sh", 'w') as f:
    for i,line in enumerate(get_all,1):
        if i == 3:
            f.writelines('#PBS -N Xx-eps%s\n' % (n))    #Name of the job
        else:
            f.writelines(line)

#construction of POSCAR -----definitions-----
#creating POSCAR - repeating for possitive eps

#getting transformation matrix
eps11= float("{:.10f}".format(1+d11)); eps12= float("{:.10f}".format(d12)); eps13= float("{:.10f}".format(d13))
eps21= float("{:.10f}".format(d21)); eps22= float("{:.10f}".format(1+d22)); eps23= float("{:.10f}".format(d23))
eps31= float("{:.10f}".format(d31)); eps32= float("{:.10f}".format(d32)); eps33= float("{:.10f}".format(1+d33))

#getting equilibrium unicell shape
with open("POSCAR", 'r') as f:
    get_all=f.readlines()
with open("POSCAR", 'w') as f:
    for i,line in enumerate(get_all,1):
        if i == 3:
            split_line = line.split()
            a11 = float(split_line[0]); a12 = float(split_line[1]); a13 = float(split_line[2])
            f.writelines(line)
        elif i == 4:

```

```

split_line = line.split()
a21 = float(split_line[0]); a22 = float(split_line[1]); a23 = float(split_line[2])
f.writelines(line)
elif i == 5:
split_line = line.split()
a31 = float(split_line[0]); a32 = float(split_line[1]); a33 = float(split_line[2])
f.writelines(line)
else:
f.writelines(line)

# definition of the new unitcell
b11= "{:.10f}".format(eps11*a11+eps12*a12+eps13*a13);
b12= "{:.10f}".format(eps21*a11+eps22*a12+eps23*a13)
b13= "{:.10f}".format(eps31*a11+eps32*a12+eps33*a13)
b21= "{:.10f}".format(eps11*a21+eps12*a22+eps13*a23)
b22= "{:.10f}".format(eps21*a21+eps22*a22+eps23*a23)
b23= "{:.10f}".format(eps31*a21+eps32*a22+eps33*a23)
b31= "{:.10f}".format(eps11*a31+eps12*a32+eps13*a33)
b32= "{:.10f}".format(eps21*a31+eps22*a32+eps23*a33)
b33= "{:.10f}".format(eps31*a31+eps32*a32+eps33*a33)

#construction of the new POSCAR
with open("POSCAR",'r') as f:
get_all=f.readlines()
with open("POSCAR",'w') as f:
for i,line in enumerate(get_all,1):
if i == 3:
f.writelines('%s %s %s\n' %(b11, b12, b13))
elif i == 4:
f.writelines('%s %s %s\n' %(b21, b22, b23))
elif i == 5:
f.writelines('%s %s %s\n' %(b31, b32, b33))
else:
f.writelines(line)
#----end of definitions----

qsub_call = "qsub %s"
call(qsub_call % "run.sh", shell=True) #launch the job into queue

os.chdir(casedir)
#negative eps

eps1="{:.8f}".format((-1.0)*eps1); eps2="{:.8f}".format((-1.0)*eps2); eps3="{:.8f}".format((-1.0)*eps3)
eps4="{:.8f}".format((-2.0)*eps4); eps5="{:.8f}".format((-2.0)*eps5); eps6="{:.8f}".format((-2.0)*eps6)
print(' -> strain vector eps = (%s, %s, %s, %s, %s, %s)' % (eps1, eps2, eps3, eps4, eps5, eps6))
with open('strain.dat', 'a') as f:
f.writelines('%s %s %s %s %s %s\n' % (eps1, eps2, eps3, eps4, eps5, eps6)) #Write eps1-6 to strain.dat

eps1=float(eps1); eps2=float(eps2); eps3=float(eps3)
eps4=float(eps4); eps5=float(eps5); eps6=float(eps6)

#definition of the deformation matrix
eps4= (eps4)/2.0; eps5= (eps5)/2.0; eps6= (eps6)/2.0

d11=float(eps1); d12=float(eps6); d13=float(eps5)
d21=float(eps6); d22=float(eps2); d23=float(eps4)
d31=float(eps5); d32=float(eps4); d33=float(eps3)

os.chdir(casedir)
caseeps=('eps-%s' % (n))
os.mkdir(caseeps)
os.chdir(caseeps)
epsdir= os.getcwd()

```

```

files = os.listdir(eqdir) #copying to $cseeps directory
copy_tree(eqdir, epsdir)
os.chdir(epsdir)
with open("INCAR", 'r') as f:
    get_all=f.readlines()
with open("INCAR", 'w') as f:
    for i,line in enumerate(get_all,1):
        if "ISIF" in line:
            f.writelines("")
        elif "NSW" in line:
            f.writelines("")
        elif "IBRION" in line:
            f.writelines("")
        else:
            f.writelines(line)
with open('INCAR', 'a') as f:
    f.writelines(' ISIF = 2\n')
    f.writelines(' NSW = 60\n')
    f.writelines(' IBRION = 1\n')

with open("run.sh", 'r') as f:
    get_all=f.readlines()
with open("run.sh", 'w') as f:
    for i,line in enumerate(get_all,1):
        if i == 3:
            f.writelines('#PBS -N 32Fe-eps-%s\n' %(n))
        else:
            f.writelines(line)

#construction of POSCAR ----definitions----
#creating POSCAR - repeating for negative eps

#getting transformation matrix
eps11= float("{:.10f}".format(1+d11)); eps12= float("{:.10f}".format(d12)); eps13= float("{:.10f}".format(d13))
eps21= float("{:.10f}".format(d21)); eps22= float("{:.10f}".format(1+d22)); eps23= float("{:.10f}".format(d23))
eps31= float("{:.10f}".format(d31)); eps32= float("{:.10f}".format(d32)); eps33= float("{:.10f}".format(1+d33))

#getting equilibrium unicell shape
with open("POSCAR", 'r') as f:
    get_all=f.readlines()
with open("POSCAR", 'w') as f:
    for i,line in enumerate(get_all,1):
        if i == 3:
            split_line = line.split()
            a11 = float(split_line[0]); a12 = float(split_line[1]); a13 = float(split_line[2])
            f.writelines(line)
        elif i == 4:
            split_line = line.split()
            a21 = float(split_line[0]); a22 = float(split_line[1]); a23 = float(split_line[2])
            f.writelines(line)
        elif i == 5:
            split_line = line.split()
            a31 = float(split_line[0]); a32 = float(split_line[1]); a33 = float(split_line[2])
            f.writelines(line)
        else:
            f.writelines(line)

#definition of the new unitcell
b11= "{:.10f}".format(eps11*a11+eps12*a12+eps13*a13)
b12= "{:.10f}".format(eps21*a11+eps22*a12+eps23*a13)
b13= "{:.10f}".format(eps31*a11+eps32*a12+eps33*a13)
b21= "{:.10f}".format(eps11*a21+eps12*a22+eps13*a23)
b22= "{:.10f}".format(eps21*a21+eps22*a22+eps23*a23)

```

```

b23= "{:.10f}".format(eps31*a21+eps32*a22+eps33*a23)
b31= "{:.10f}".format(eps11*a31+eps12*a32+eps13*a33)
b32= "{:.10f}".format(eps21*a31+eps22*a32+eps23*a33)
b33= "{:.10f}".format(eps31*a31+eps32*a32+eps33*a33)

#construction of the new POSCAR
with open("POSCAR",'r') as f:
    get_all=f.readlines()
with open("POSCAR",'w') as f:
    for i,line in enumerate(get_all,1):
        if i == 3:
            f.writelines('%s %s %s\n' %(b11, b12, b13))
        elif i == 4:
            f.writelines('%s %s %s\n' %(b21, b22, b23))
        elif i == 5:
            f.writelines('%s %s %s\n' %(b31, b32, b33))
        else:
            f.writelines(line)

qsub_call = "qsub %s"
call(qsub_call % "run.sh", shell=True) #launch the job into queue
#----end of definitions----

```

Appendix 2: Script results_Cij_stress-strain.py

```

#####
####This script was written as a practical part of master's thesis by Bc. Martin Čavojec. Purpose of this script is to
####calculate stiffness tensor from stress and strain tensors calculated by previous script.
####This code is based on works:
####YU, R., J. ZHU a H. Q. YE. Calculations of single-crystal elastic constants made simple. Computer Physics
####Communications. 2010, 181(3), 671-675. doi: https://doi.org/10.1016/j.cpc.2009.11.017
####ZHOU, L., D. HOLEC and P. H. MAYRHOFER. First-principles study of elastic properties of cubic Cr1-xAlxN
####alloys. Journal of Applied Physics. 2013, 113(4), 043511. doi: https://doi.org/10.1063/1.4789378
#####

import os
from pathlib import Path
import numpy as np

#Setting up directory pathing
eps = 0.021 #define the strain to be examined
maindir = os.getcwd(); maindir = Path(maindir)
print("Main directory is: {0}".format(os.getcwd())) #Main directory

os.chdir('eq'); eqdir = os.getcwd(); eqdir = Path(eqdir)
print("Directory of equilibrium structure is: {0}".format(os.getcwd())) #'eq' directory

os.chdir('../eps=%s' %(eps)); casedir = os.getcwd(); casedir = Path(casedir)
print("Directory of deformed structures is: {0}".format(os.getcwd())) #'eps=0.021' directory

os.chdir(eqdir); print('Searching for the equilibrium structure in : %s' %(eqdir))

#Getting equilibrium structure
if os.path.exists('CONTCAR'):
    eqposcar = eqdir/CONTCAR; print('Equilibrium structure is read from the: %s' %(eqposcar)) #The equilibrium structure
read from CONTCAR
elif os.path.exists('POSCAR'):
    eqposcar = eqdir/POSCAR; print('Equilibrium structure is read from the: %s' %(eqposcar)) #The equilibrium structure
read from POSCAR
else: print('Cannot find equilibrium structure, terminating') #The equilibrium structure is missing

os.chdir(maindir)
file = open(eqposcar); content = file.readlines()

```

```

with open('A.mat', 'w') as f:
    f.writelines(content[2]); f.writelines(content[3]); f.writelines(content[4])

#Remove existing strain.dat and stress.dat
if os.path.exists(casedir/'strain.dat'):
    os.remove(casedir/'strain.dat')

if os.path.exists(casedir/'stress.dat'):
    os.remove(casedir/'stress.dat')

#Collecting data from all eps subfolders
cnt=0
epsarray = np.array([1, -1, 2, -2, 3, -3, 4, -4, 5, -5, 6, -6])
for casen in epsarray:
    cnt=(cnt+1)
    print(""); epsdir = ('eps%s' %(casen)); print(epsdir)
    #Calculation of the strain tensor, and adding the strain vector to strain.dat (Voigt's notation)
    os.chdir(casedir/epsdir)
    epsdirp = Path(os.getcwd())
    file = open(epsdirp/'OUTCAR'); content = file.readlines()
    os.chdir(maindir)
    with open('B.mat', 'w') as f: #Creating matrix for deformed structure
        f.writelines(content[2]); f.writelines(content[3]); f.writelines(content[4])

A = np.loadtxt("A.mat"); B = np.loadtxt("B.mat") #Loading eq and deformed structures as matrices
I = np.identity(3) #Identity matrix 3x3
C = np.linalg.lstsq(A, B, rcond=None) [0] -I #C=(A\B)-I

C1="{:.8f}".format(C[0,0]); C2="{:.8f}".format(C[1,1]); C3="{:.8f}".format(C[2,2])
C4="{:.8f}".format(2*C[1,2]); C5="{:.8f}".format(2*C[0,2]); C6="{:.8f}".format(2*C[0,1])

with open(casedir/'strain.dat', 'a') as f: #writing stress tensor
    f.writelines('%s %s %s %s %s %s\n' % (C1, C2, C3, C4, C5, C6))
print('Strain vector is: %s %s %s %s %s %s' % (C1, C2, C3, C4, C5, C6))

#read the stresses and add them to stress.dat
with open(epsdirp/'OUTCAR', 'r') as f: #reading from OUTCAR
    get_all=f.readlines()
    for i,line in enumerate(get_all, 1):
        if "in kB" in line:
            vec = line.replace('in kB', '') #finding the stress tensor
            split_line = vec.split()
            sv1 = float(split_line[0])/10; sv2 = float(split_line[1])/10; sv3 = float(split_line[2])/10
            sv4 = float(split_line[4])/10; sv5 = float(split_line[5])/10; sv6 = float(split_line[3])/10

print('Stress tensor is: %s %s %s %s %s %s' % (sv1, sv2, sv3, sv4, sv5, sv6))
with open(casedir/'stress.dat', 'a') as f:
    f.writelines('%s %s %s %s %s %s\n' % (sv1, sv2, sv3, sv4, sv5, sv6))

print(""); print('Total number of deformations: %s' %(cnt))
#calculate the elastic constants
print(""); print('Calculating the elastic constants'); print('...')
stress = np.loadtxt(casedir/'stress.dat'); strain = np.loadtxt(casedir/'strain.dat')
Cij=np.linalg.lstsq(-strain, stress, rcond=None) [0]; CijT= np.transpose(Cij); Cij=(Cij+CijT)/2 #Cij=(-strain\stress)
#Cij=(Cij+Cij)/2
Cij=(np.array_str(Cij, precision=3, suppress_small=True).replace(' [', '').replace('[', '').replace(']', '')) #Formating the tensor
print('Elasticity matrix: '); print("")
with open(maindir/'Cij.dat', 'w') as f: #creation of stiffness tensor
    f.writelines('%s' % (Cij))
print(Cij); os.remove('A.mat'); os.remove('B.mat') #removal of temporary files

```