From Benchmarking to Periodic Fock Exchange in the Auxiliary Density Matrix Method via k-Point Sampling with Gaussian Basis Sets

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde

(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Tiziano Müller

von

Zürich ZH

Promotionskommission

Prof. Dr. Jürg Hutter (Vorsitz & Leitung)

Dr. Joost VandeVondele

Prof. Dr. Peter Hamm

Zürich, 2024

Abstract

While Gaussian basis sets have been successfully used in various electronic structure codes for elucidation of structures and processes in computational chemistry, they are not nearly as popular in solid state physics. There are various reasons for this, one of them being that in comparison to the plane wave approach there is no single numerical parameter like the cutoff energy, which can be used to systematically increase the accuracy of the calculation. Furthermore do basis sets depend either on a specific method, or can only be used together with matching pseudopotentials in the case of valence basis sets. Finally, in contrast to a molecular setting are electrons in periodic systems often not localised and therefore many optimisations used with atom-centred Gaussian basis sets do not work as efficiently. The number of codes targeting the intersection of computational chemistry and solid efficiently state physics is therefore small, as is the availability of universally applicable basis set families.

In this thesis we are evaluating the recently revised MOLOPT basis set with pseudopotentials and all-electron calculations in both molecular and condensed matter settings, for three different functionals in Density Functional Theory, each on a different rung on "Jacob's Ladder". Once again, we demonstrate the outstanding performance of the MOLOPT basis sets for molecular systems and show that Gaussian-type Orbital codes perform reasonably well for solids, and that this can also be achieved with a universal basis set family.

We further investigate the required framework for running and analysing such large benchmark calculations to pave the way for fully automated benchmarking and iterative development of Gaussian basis sets and pseudopotentials. The implementation of the of the actual benchmarks with three different approaches leads to the development of Python-based parser and input generation libraries and utilities for CP2K, as well as contributions and extensions to the Automated Interactive Infrastructure and Database (AiiDA).

To extend the support for periodic systems further within CP2K, we then progress to implement **k**-point sampling in CP2K's Hartree-Fock Exchange, required to run calculations with Hybrid functionals. And to be able to make this usable on realistic systems, we extend the Auxiliary Density Matrix Method to **k**-point sampling as well. Together, this novel implementation can be used as a reference implementation and stepping stone for the development of more efficient algorithms for the Hartree-Fock approximation and Hybrid functionals in Density Functional Theory with CP2K.

Kurzzusammenfassung

Während Gaußsche Basissätze erfolgreich in verschiedenen Elektronenstrukturprogrammen zur Aufklärung von Strukturen und Prozessen in der rechnergestützten Chemie eingesetzt werden, sind sie in der Festkörperphysik weniger verbreitet. Dafür gibt es mehrere Gründe, einer davon ist, dass es im Vergleich zum ebenen Wellen Ansatz keine einzelne numerische Grösse wie die Grenzwert Energie gibt, welche dazu benutzt werden kann um systematisch die Genauigkeit der Rechnung zu erhöhen. Im Weiteren hängen die Basissätze entweder von einer spezifischen Methode ab, oder können im Fall von Valenz Basis Sätzen nur mit passenden Pseudopotentialen verwendet werden. Schließlich sind die Elektronen in periodischen Systemen im Gegensatz zu molekularen Systemen oft nicht lokalisiert, so dass viele Optimierungen, die mit atomzentrierten Gaußschen Basissätzen verwendet werden, nicht so effizient funktionieren. Die Anzahl der Softwareprogramme welche auf die Schnittstelle zwischen rechnergestützten Chemie und Festkörperphysik abzielen ist demzufolge klein, sowie die Verfügbarkeit von universell einsetzbaren Basissatzfamilien.

In dieser Arbeit evaluieren wir die kürzlich revidierte MOLOPT Basissatzfamilie mit Pseudopotentialen und Gesamtelektronenberechnungen für molekulare Systeme als auch für Systeme der kondensierten Matterie, für drei verschiedene Funktionale der Dichtefunktionaltheorie, jede auf einer anderen Sprosse der "Jacob's Ladder". Damit demonstrieren wir einmal mehr die hervorragende Leistung der MOLOPT-Basissätze für molekulare Systeme und zeigen, dass Codes welche Orbitale vom Gauß-Typ für Festkörper gut funktionieren, und dass dies auch mit einer universellen Basissatzfamilie erreicht werden kann.

Darüber hinaus untersuchen wir den erforderlichen Rahmen für die Durchführung und Analyse solch großer Benchmark-Berechnungen um den Weg für ein vollautomatisches Benchmarking und die iterative Entwicklung von Gaußschen Basissätzen und Pseudopotentialen zu ebnen. Die Implementierung der aktuellen Benchmarks mit drei verschiedenen Ansätzen führt zur Entwicklung von Python-basierten Parser- und Eingabegenerierungsbibliotheken und Dienstprogrammen für CP2K, sowie Beiträge und Erweiterungen zur Automated Interactive Infrastructure and Database (AiiDA).

Um die Unterstützung für periodische Systeme in CP2K weiter auszubauen, implementieren wir **k**-Punkt Abtastung in CP2Ks Hartree-Fock Austausch, die für Berechnungen mit Hybridfunktionalen erforderlich ist. Und um dies für realistische Systeme nutzbar zu machen, erweitern wir die Auxiliary Density Matrix Method (ADMM) ebenfalls auf **k**-Punkt Abtastung. Zusammengenommen kann diese neuartige Implementierung als Referenzimplementation und Sprungbrett für die Entwicklung von effizienteren Algorithmen für die Hartree-Fock-Näherung und Hybridfunktionale in der Dichtefunktionaltheorie mit CP2K verwendet werden.

Academic Record

- Thomas D. Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z. Khaliullin, Ole Schütt, Florian Schiffmann, Dorothea Golze, Jan Wilhelm, Sergey Chulkov, Mohammad Hossein Bani-Hashemian, Valéry Weber, Urban Borštnik, Mathieu Taillefumier, Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, Tiziano Müller, Robert Schade, Manuel Guidon, Samuel Andermatt, Nico Holmberg, Gregory K. Schenter, Anna Hehn, Augustin Bussy, Fabian Belleflamme, Gloria Tabacchi, Andreas Glöß, Michael Lass, Iain Bethune, Christopher J. Mundy, Christian Plessl, Matt Watkins, Joost VandeVondele, Matthias Krack, and Jürg Hutter. "CP2K: An Electronic Structure and Molecular Dynamics Software Package - Quickstep: Efficient and Accurate Electronic Structure Calculations." In: *J. Chem. Phys.* 152.19 (May 21, 2020)
  - Δ-test evaluation of original and *UCL*-extended MOLOPT basis set family (*XV.D*).
  - Migration from Subversion to Git, general tooling improvements (*XV.F*)
- Sebastiaan P. Huber, Spyros Zoupanos, Martin Uhrin, Leopold Talirz, Leonid Kahle, Rico Häuselmann, Dominik Gresch, Tiziano Müller, Aliaksandr V. Yakutovich, Casper W. Andersen, Francisco F. Ramirez, Carl S. Adorf, Fernando Gargiulo, Snehal Kumbhar, Elsa Passaro, Conrad Johnston, Andrius Merkys, Andrea Cepellotti, Nicolas Mounet, Nicola Marzari, Boris Kozinsky, and Giovanni Pizzi. "AiiDA 1.0, a Scalable Computational Infrastructure for Automated Reproducible Workflows and Data Provenance." In: *Sci Data* 7.1 (1 Sept. 8, 2020)
  - Full Python 3 port of AiiDA core.
  - Various contributions to both architecture and software development.
- Giovanni Pizzi, Emanuele Bosoni, Louis Beal, Marnik Bercx, Peter Blaha, Stefan Blügel, Jens Bröder, Martin Callsen, Stefaan Cottenier, Augustin Degomme, Vladimir Dikan, Kristjan Eimre, Espen Flage-Larsen, Marco Fornari, Alberto Garcia, Luigi Genovese, Matteo Giantomassi, Sebastiaan Huber, Henning Janssen, Georg Kastlunger, Matthias Krack, Georg Kresse, Thomas Kühne, Kurt Lejaeghere, Georg Madsen, Martijn Marsman, Nicola Marzari, Gregor Michalicek, Hossein Mirhosseini, Tiziano Müller, Guido Petretto, Chris Pickard, Samuel Poncé, Gian-Marco Rignanese, Oleg Rubel, Thomas Ruh, Michael Sluydts, Danny Vanpoucke, Sudarshan Vijay, Michael Wolloch, Daniel Wortmann, Aliaksandr Yakutovich, Jusong Yu, Austin Zadoks, and Bonan Zhu. "How to Verify the Precision of Density-Functional-Theory Implementations via Reproducible and Universal Workflows." In: *Nat Rev Phys* (2023). in printing
  - Early evaluation of concept for CP2K.
  - Contribution to CP2K workflow code, review.
- Tiziano Müller, Nina Mujkanovic, Juan J. Durillo, and Nicolay Hammer. "Survey of Adaptive Containerization Architectures for HPC." in: Supercomputing 23, CANOPIE-HPC. Denver, CO, Nov. 2023
  - Development of new Kubernetes-Slurm integration.
  - Review of HPC container solutions.
- Tiziano Müller and Nina Mujkanovic. "Survey and Practical Taxonomy of HPC Workflow Managers." In: *tbd* (2024). in preparation
  - Survey of domain-specific and general workflow managers, mainly for High-performance computing (HPC).
  - Development of practical taxonomy, classification.

Acknowledgment

First of all, I would like to thank Professor Jürg Hutter for giving me the opportunity to do my PhD in his group, for his continuous support and guidance throughout the projects.

For all the on- and off-topic discussions, early feedback, reviews and support I want to thank my colleagues in the group over the years: Augustin, Mauro, Alfio, Patrick, Fabian, Anna, Yasmine, Dorothea, Ralph, Michela, Marcella, Gabriele, Vladimir, Giacomo, Jinggang, Beliz, Maria, Fernanda, …

My gratitude also goes to the AiiDA team for the welcoming environment, in particular Giovanni, Leonid, Leopold, Martin, Sebastiaan & Aliaksandr. I have enjoyed the drive and intensity of the Hackathons very much.

My friends and family deserve an extra big *thank you*, for their support and for forgiving my communication silence in busy times while still being there for me when I needed them most.

Finally, to my partner, who took me on the greatest journey, life itself.

Funding bodies Part of the work on AiiDA as well as the evaluation of the original CP2K MOLOPT basis sets with the Δ-test were implemented as part of the NCCR MARVEL. Work on the DBCSR build system and related software engineering topics were partially sponsored by the Platform for Advanced Scientific Computing (PASC). The organisation of the Workshop on Modern Fortran as well as the two International Fortran Conference instalments were sponsored by the Graduate Campus of the University of Zurich. I also want to acknowledge the efforts of the Computational Science Zurich graduate school for the organisation of their Academia Industry Modelling (AIM) week, permitting insight into computational topics in the industry, as well as other courses and retreats.

Swiss National Science Foundation
The National Centres of Competence in Research (NCCR) are a research instrument of the Swiss National Science Foundation

COMPUTATIONAL SCIENCE ZURICH

Contents

Acronyms

ADMM Auxiliary Density Matrix Method. 2, 22, 24–26, 60, 61, 67–70, 72, 74–76, 78, 79, 81, 83
AE All-Electron. 1, 2, 16, 17, 27, 47, 61
API Application Programming Interface. 31, 45

BSSE basis set superposition error. 17
BvK Born-von Kármán. 18, 19
BZ Brillouin Zone. 18

CO Crystalline Orbital. 19

DFT Density Functional Theory. i, 1, 8, 13, 14, 16, 17, 20, 25–27, 67
DIIS Direct Inversion of the Iterative Subspace. 68, 75

ERI Electron Repulsion Integral. 2, 13, 14, 17, 20, 21, 25, 26, 67–69, 73, 75, 76, 79

FAM Fabric-attached memory. 76
FFT Fast-Fourier Transformation. 14
FP-LAPW Full-Potential Linearized Augmented Plane Wave method. 2, 15, 16, 51, 56, 77

GAPW Gaussian and Augmented Planewave. 1, 2, 16, 28, 49, 50, 53, 59, 60, 69, 77
GGA Generalized Gradient Approximation. 2, 10, 12, 25, 28
GPW Gaussian and Plane Waves Method. 1, 2, 14, 16, 19, 49, 50, 55, 60, 76, 85
GTO Gaussian-type Orbital. i, 1, 15, 16, 26, 27, 68, 77–79

h-BN Hexagonal boron nitride. 69–75
HF Hartree-Fock. i, 13, 19, 68
HFX Hartree-Fock Exchange. i, 2, 7, 8, 17, 20, 25, 26, 67–70, 72, 73, 75, 76, 78, 79
HFXk Hartree-Fock Exchange with **k**-point. 2, 69
HPC High-performance computing. v, 32, 80

ISDF interpolative separable density fitting. 68, 69

JSON JavaScript Object Notation. 30, 31, 35, 37

LCAO linear combination of atomic orbitals. 19, 22
LDA Local Density Approximation. 25
LiH Lithiumhydrid. 69–71, 75

MD molecular dynamics. 17, 20, 41
MO Molecular Orbital. 6, 13, 14, 16, 19, 20, 22, 23, 25, 68

occ-RI-K occupied orbital Resolution of Identity (RI)-K. 68

PAW Projector Augmented-Wave method. 16
PGF primitive Gaussian function. 21
PP pseudopotential. 1, 2, 14–16, 27, 78, 79, 81, 85, 105
PW plane wave. 1, 2, 14–16, 19, 77, 79

REST Representational state transfer. 30, 31
RI Resolution of Identity. xi, 68, 76, 79

SCF self-consistent field. 20, 26, 41, 74, 75, 79
SIE self-interaction error. 8, 12

SMDB Small Molecules Database. 28, 32, 48, 60, 78, 81–83

TC Truncated Coulomb. 75
TTS time to solution. 73

XC Exchange-Correlation. 14, 16

1. Introduction

Mankind's development has historically been driven by the discovery of elements and materials, starting from copper up to the current age of silicon. However, since the first predictions in the periodic table of elements by Mendeleev, this has started to change. Nowadays experimental discoveries in the materials sciences are not only elucidated by accompanying computational simulations, but guided by them. To do so, one has to solve the famous Schrödinger equation, describing the behaviour of quantum particles such as the electrons. There exist many possible ways to do so, each with their own strengths and weaknesses. While plane wave (PW) approaches cater more to the needs of the solid state community, approaches using Gaussian-type Orbital (GTO) orbitals permit direct insight into the chemistry involved, such as chemical bonds. As one of the few codes making use of a combined GTO and PW basis approach in what is called the Gaussian and Plane Waves Method (GPW), the CP2K software package can excellently target the intersection of both fields. While GTO based codes have been used successfully at least since the 60's in computational chemistry, they have generated a Zoo of different basis sets optimised for various levels of theory, methods and use cases. And although generating a basis set family – a series of increasingly larger number of basis functions – for a specific case is not difficult anymore (and has been a common task), developing one which is robust enough such that it can be used for a wide range of tasks is not trivial. Unlike with PW, there is also no rigorous way to proof that extending a GTO basis set will improve the result, other than by computation. But as the possibilities for computation have grown – quantum chemistry and classical molecular dynamics make up for 40% of all high-performance computation time –, so have the use cases. Meaning that a basis set family nowadays should ideally cover the full periodic table, as well as various functionals of Density Functional Theory (DFT). Finally, running a benchmark on such a large basis set family has its own challenges, requiring a large number calculations on different systems to cover all of the above.

This dissertation thus encompasses three distinct areas of computation.

In the first section, we concentrate on the fundamental components needed to run a large amount of calculations for evaluating basis sets and pseudopotentials for various systems and functionals, and how they can be executed effectively and dependably. This involves the development of a minimal domain-specific workflow manager, along with required foundation libraries for automated input configuration generation, output data parsing, as well as data transport and load distribution. By using a proper component-based software architecture we can then integrate the tools in the much more general AiiDA workflow management system to improve the integration of CP2K.

We then progress to applying the developed tools to run two different benchmarks to evaluate the recently revised MOLOPT valence basis sets alongside their also revised Gödecker-Teter-Hutter pseudopotential (PP) with the GPW method, and the newly added All-Electron (AE) MOLOPT basis set using the Gaussian and Augmented Planewave (GAPW) method.

First, a database of small molecules will be used to compare the performance of CP2K in a molecular setting, with reference calculations obtained using the G16 software package. This benchmark is performed for three different functionals on different rungs of *Jacob's Ladder of Density Functional Approximations*: Generalized Gradient Approximation (GGA), meta-GGA and Hybrid functionals, using PBE, TPSS and PBE0. For the Hybrid functional PBE0 also the effect of using Auxiliary Density Matrix Method (ADMM) is investigated. This allows us to draw a complete picture of the performance of the complete MOLOPT basis set in a molecular setting, as well as investigate the transferability of basis set and pseudopotential within a rung. Additionally, we identify important requirements for running such benchmarks in an automated fashion.

The second benchmark is then geared towards the solid state use case by using the Δ-test metric and comparing against published high-precision results. Given the larger number of calculations, it makes full use of our own workflow manager, as well as AiiDA together with custom extensions. Since this benchmark is aimed at determining accuracy between different codes or methods and consists only of elemental crystals, it draws its significance even more from single-point comparisons with reference values than the previous benchmark. To extend this benchmarks towards covering different functionals, we therefore devise and implement a strategy to generate all data starting from the high-accuracy Full-Potential Linearized Augmented Plane Wave method (FP-LAPW) method (available via the integration of SIRIUS) to AE GAPW calculations to PP calculations with PW (again via SIRIUS) and GPW. This reference is obtained with the PBE functional and we validate the approach with the closely related PBEsol.

The third and final part of this thesis covers the enabling of the previously mentioned solid state benchmarks to be run with CP2K using Hybrid functionals such as PBE0. While Hartree-Fock Exchange (HFX) has been implemented in CP2K since quiet some time, it only covers the periodic case without **k**-point sampling. A supercell approach could be employed to sample the Brillouin zone instead, but this supercell must become very large to reach the same **k**-point density as with direct **k**-point sampling, rendering such calculations prohibitively expensive. The rate limiting step in this calculation is the number of Electron Repulsion Integral (ERI) which have to be calculated for the HFX, which depends highly on both size and type of the basis set. Within a Hartree-Fock Exchange with **k**-point (HFX**k**) calculation this becomes even more pronounced as some of the techniques employed previously – namely the caching of summed-up four-centre ERI – can not be applied anymore due to the sheer amount of such ERIs and their associated storage requirement. And since the goal is to calculate solid state – including metallic – systems, screening on the density matrix to reduce the number of quartets to calculate becomes ineffective. Which is why we have not only added **k**-point-support to HFX, but also extended the ADMM to support **k**-point calculations. By employing a smaller (and less diffuse) auxiliary basis set, it is possible to significantly lower the number of primitive basis functions, making such calculations feasible.

2. Theory

In this chapter, we are reviewing the required theory for the Density Functional Theory (DFT) and the Hartree-Fock approximation when solving the Schrödinger equation of electronic structure theory. We will furthermore look into the Auxiliary Density Matrix Method to reduce the computational effort and – more importantly – its extension to periodic **k**-point Hartree-Fock theory, permitting the use of ADMM for condensed matter systems. Additionally, a short recapitulation of Gaussian-type orbitals in electronic structure theory in the form of the Gaussian and plane-waves (GPW) method will be presented, together with the mentioning of pseudopotentials and basis sets since these have been under investigation as part of this thesis.

A more stringent introduction into the matter can be found in the usual textbooks [6–11]

2.1. Hartree-Fock Theory

To calculate the electronic structure of molecules and solids we are trying to solve the time-independent Schrödinger equation in the Born-Oppenheimer (BO) approximation of fixed atomic nuclei, which is

$$\mathscr{H}\psi = E\psi \tag{2.1}$$

and where the (non-relativistic) electronic Hamilton operator in atomic units [7] is written as

$$\mathscr{H} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}}. \tag{2.2}$$

The terms of the Hamiltonian are in the order of appearance: the kinetic energy of the electrons, the Coulomb terms of the attraction between electrons and nuclei and the electron-electron interaction, with r_{ij} and r_{iA} the distance between the i-th and j-th electron, respectively the i-th electron and the A-th nucleus, and N and M the number of electrons and nuclei.

As part of the BO approximation, we can neglect the contribution of the kinetic energy of the nuclei as the electrons can be assumed to adjust themselves immediately compared to the time frame in which the nuclei change position. Given that the proton-electron mass ratio is $\mu = \frac{m_p}{m_e} \approx 1836$ the inverse effect can also be neglected. Since the Coulomb term of the nucleus-nucleus interaction can also be assumed to be constant, it has been omitted as well as it would only add a constant shift to the energy.

The general form of ψ is a many-body $4N$-dimensional wavefunction $\psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ with each $\mathbf{x}_i$ consisting of three real space components and one spin component $\mathbf{x}_i = \{\mathbf{r}_i, \omega_i\}$. Since this is not solvable in useful

time for non-trivial systems we need a different ansatz. Quantum mechanics demands that for Fermionic particles (electrons in our case) the wavefunction must be totally antisymmetric

$$\psi(\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_j, \dots, \mathbf{x}_N) = -\psi(\mathbf{x}_1, \dots, \mathbf{x}_j, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N), \tag{2.3}$$

which means that the simple Hartree product ansatz of

$$\psi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \chi_1(\mathbf{x}_1)\chi_2(\mathbf{x}_2)\cdots\chi_N(\mathbf{x}_N), \tag{2.4}$$

where $\chi_i(\mathbf{x}_j)$ are single-electron wavefunctions, is insufficient as it would violate that requirement. From Linear Algebra we know that a determinant is totally antisymmetric and thus arrive at the single-determinant Hartree-Fock ansatz for the wavefunction in form of a Slater determinant

$$\psi_{\mathrm{HF}}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \dots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \dots & \chi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \dots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \dots & \chi_N(\mathbf{x}_N) \end{vmatrix}. \tag{2.5}$$

There are of course multiple possible determinants such that a complete solution should encompass all possible ones, which is the point of the Full Configuration Interaction approach, but which we are not going to pursue further.

To obtain the energy from the Schrödinger equation we take a look at the expectation value of the Hamiltonian

$$\langle\psi|\mathscr{H}|\psi\rangle = \langle\psi|E|\psi\rangle \Rightarrow E[\psi] = \frac{\langle\psi|\mathscr{H}|\psi\rangle}{\langle\psi|\psi\rangle}, \tag{2.6}$$

with the usual definition of the Dirac notation of the expectation value

$$\langle\psi|\mathscr{A}|\psi\rangle = \int \psi^*(\mathbf{x})\mathscr{A}(\mathbf{x})\psi(\mathbf{x})d\mathbf{x}. \tag{2.7}$$

The variational principle dictates the existence of a uniquely defined energy value E_0 – the ground state energy – along with its ground state wave function ψ_0 for which for any other trial wave function ψ the inequality $E[\psi] \geq E[\psi_0] = E_0$ holds. Based on this we define the Hartree-Fock energy as the variational minimum

$$E_{\mathrm{HF}} \coloneqq \min_{\chi_i} \langle\psi_{\mathrm{HF}} \,|\, \mathscr{H} \,|\, \psi_{\mathrm{HF}}\rangle \tag{2.8}$$

Additionally, we mandate that the χ_i in this minimization procedure form an orthonormal set (which will allow us to apply the Slater-Condon rules for the one- and two-body integrals), e.g.

$$\int \chi_i^*(\mathbf{x})\chi_j(\mathbf{x})d\mathbf{x} = \delta_{ij}. \tag{2.9}$$

Inserting the electronic Hamiltonian of Equation 2.2 in Equation 2.8 then yields

$$\begin{aligned} E_{\mathrm{HF}} &= \langle \psi_{\mathrm{HF}} | \mathscr{H} | \psi_{\mathrm{HF}} \rangle \\ &= \sum_{i}^{N} \underbrace{\int \chi_i^*(\mathbf{x}) \left[-\frac{1}{2}\nabla^2 + v(\mathbf{x}) \right] \chi_i(\mathbf{x}) d\mathbf{x}}_{=:H_i} \\ &+ \frac{1}{2} \sum_{i,j,i\neq j}^{N} \underbrace{\iint \chi_i(\mathbf{x}_1)\chi_i^*(\mathbf{x}_1) \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \chi_j^*(\mathbf{x}_2)\chi_j(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2}_{=:J_{ij}}, \\ &- \frac{1}{2} \sum_{i,j,i\neq j}^{N} \underbrace{\iint \chi_i^*(\mathbf{x}_1)\chi_j(\mathbf{x}_1) \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \chi_i(\mathbf{x}_2)\chi_j^*(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2}_{=:K_{ij}} \end{aligned} \tag{2.10}$$

where $v(\mathbf{x})$ as the external potential captures the electron-nuclei interaction.

By restricting ourselves to the closed-shell approximation where each orbital is double occupied (hence we always have an even number of electrons) and a product ansatz in which we express the spin orbitals as a product of spatial orbitals and spin functions $\chi_i(\mathbf{x}) = \chi_i(\mathbf{x})\alpha_i(\omega_i)$, Equation 2.10 can be simplified to

$$E_{\mathrm{HF}} = 2 \sum_{i=1}^{N/2} H_i(\mathbf{r}) + \sum_{i,j}^{N/2} \left(2J_{ij}(\mathbf{r}) - K_{ij}(\mathbf{r}) \right), \tag{2.11}$$

where the Hartree- and Exchange-integrals J, K but also the one-particle kinetic energy term H depend only on the spatial coordinates. Based on those integrals J and K we can now define new operators j and k with the following actions on a function $f(\mathbf{r})$

$$\begin{aligned} j(\mathbf{r}_1)f(\mathbf{r}_1) &= 2 \sum_{j}^{N/2} \int f(\mathbf{r}_1) \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \chi_j(\mathbf{r}_2)\chi_j^*(\mathbf{r}_2) d\mathbf{r}_2 \\ k(\mathbf{r}_1)f(\mathbf{r}_1) &= \sum_{j}^{N/2} \int \chi_j^*(\mathbf{r}_1) \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \chi_j(\mathbf{r}_2) f(\mathbf{r}_1) d\mathbf{r}_2 \end{aligned}, \tag{2.12}$$

which in turn lets us define the Fock operator

$$F(\mathbf{r}) = -\frac{1}{2}\nabla^2 + v(\mathbf{r}) + j(\mathbf{r}) - k(\mathbf{r}), \tag{2.13}$$

and leads us to the Hartree-Fock eigenvalue equations

$$F(\mathbf{r})\chi_i(\mathbf{r}) = \sum_{j} \epsilon_{ij}\chi_j(\mathbf{r}). \tag{2.14}$$

Since j and k depend on a sum over an integral over spatial orbitals, the Hartree-Fock theory is inherently orbital-dependant, yet $j - k$ can be interpreted in the same way as v, as an average potential controlling the behaviour of each orbital. This orbital dependency, despite the mean-field characteristics of the operators, can thus computationally only be resolved by an iterative procedure, starting from an initial guess for the orbitals (for example for isolated atoms) and repeated until consistency is reached, recalculating $j - k$ again from the newly obtained orbitals χ_i. This approach is known as a self-consistent field (SCF) method.

To be able to solve the Hartree-Fock Equation 2.14 practically, we have to choose a basis in which we are going to express our orbitals, allowing us to solve the SCF problem by optimizing the coefficients. With $\{\phi_\mu(\mathbf{r})\}$ as our basis we write

$$\chi_i(\mathbf{r}) = \sum_\mu C_{\mu i}\phi_\mu(\mathbf{r}) \tag{2.15}$$

and call the $C_{\mu j}$ the Molecular Orbital (MO) coefficients. The variational minimisation therefore becomes a linear variational problem, solveable by diagonalisation of the corresponding operator matrix.

As for the previously defined Hartree-Fock equations: Given that the Fock operator is a Hermitian operator, we can always find a unitary transformation $\mathbf{U}$, which diagonalizes ϵ_{ij}, such that the Hartree-Fock equations can be written in the canonical form (e.g. without a sum) of

$$F(\mathbf{r})\chi_i(\mathbf{r}) = \epsilon_i\chi_i(\mathbf{r}). \tag{2.16}$$

Inserting the basis expansion then leads from

$$F(\mathbf{r})\sum_\mu C_{\mu i}\phi_\mu(\mathbf{r}) = \epsilon_i\sum_\mu C_{\mu i}\phi_\mu(\mathbf{r}). \tag{2.17}$$

by integrating after multiplying with ϕ_ν^* to

$$\begin{gathered} C_{\mu i}\underbrace{\int \phi_\nu^*(\mathbf{r})F(\mathbf{r})\phi_\mu(\mathbf{r})d\mathbf{r}}_{=:F_{\nu\mu}} = \epsilon_i\sum_\mu C_{\mu i}\underbrace{\int \phi_\nu^*(\mathbf{r})\phi_\mu(\mathbf{r})d\mathbf{r}}_{=:S_{\nu\mu}} \\ \Leftrightarrow \\ \sum_\mu F_{\nu\mu}C_{\mu i} = \epsilon_i\sum_\mu S_{\nu\mu}C_{\mu i}. \end{gathered} \tag{2.18}$$

The final equations are nothing else but a matrix equation written out, in proper matrix notation they read as follows and are called the *Roothan equations*

$$FC = \epsilon SC, \tag{2.19}$$

where ϵ is a diagonal matrix with the ϵ_i on its diagonal. When using a finite-sized basis set, this is therefore a solvable matrix eigenvalue problem. F and S are consequently called the Fock and Overlap matrices.

As we will see later, we will need the isolated Hartree-Fock Exchange (HFX) energy term K_{ij} from Equation 2.10 in this basis set representation (for the closed-shell case). The other components we will also need are the electron density $\rho(\mathbf{r})$ and the corresponding density matrix P. They let us write the HFX energy in a more concise way.

As we know from general quantum mechanics, the square modulus of the wave function can be interpreted as a probability density (in our case) to find an electron at position $\mathbf{r}$. Again for the closed-shell case and expressed in our basis $\{\phi_i\}$ this means

$$\begin{aligned}\rho(\mathbf{r}) &= 2\sum_{i}^{N/2} |\chi_i(\mathbf{r})|^2 \\ &= 2\sum_{i}^{N/2} \chi_i^*(\mathbf{r})\chi_i(\mathbf{r}) \\ &= \sum_{\mu\nu} \underbrace{2\sum_{i}^{N/2} C_{\mu i}C_{\nu i}}_{P_{\mu\nu}}\, \phi_\mu(\mathbf{r})\phi_\nu^*(\mathbf{r}) \\ &= \sum_{\mu\nu} P_{\mu\nu}\, \phi_\mu(\mathbf{r})\phi_\nu^*(\mathbf{r}).\end{aligned} \tag{2.20}$$

The exchange term elements of the Fock matrix then becomes

$$\begin{aligned}F_{\mu\sigma}^{\mathrm{x}} &= -\frac{1}{2}\sum_{\nu\lambda} P_{\nu\lambda} \iint \phi_\mu^*(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\frac{1}{\|\mathbf{r}_2-\mathbf{r}_1\|}\phi_\lambda^*(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \\ &= -\frac{1}{2}\sum_{\nu\lambda} P_{\nu\lambda}(\mu\nu\,|\,\lambda\sigma),\end{aligned} \tag{2.21}$$

where we have used the standard expression of two-electron integrals (in Mulliken notation with round brackets to emphasise that the integral is over spatial rather than spin orbitals, e.g. the spin has already been integrated out)

$$(\mu\nu\,|\,\lambda\sigma) = \iint \phi_\mu^*(\mathbf{r_1})\phi_\nu(\mathbf{r_1})\frac{1}{|\mathbf{r_2}-\mathbf{r_1}|}\phi_\lambda^*(\mathbf{r_2})\phi_\sigma(\mathbf{r_2})d\mathbf{r_1}d\mathbf{r_2}. \tag{2.22}$$

And thus follows the Hartree-Fock energy as a contraction of density matrix elements and four-center integrals

$$E_{\mathrm{x}}^{\mathrm{HF}} = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma} P_{\mu\sigma}P_{\nu\lambda}(\mu\nu\,|\,\lambda\sigma). \tag{2.23}$$

The restriction to take only a single determinant is an artificial one, as already mentioned, to keep computational cost at bay. Together with the ansatz of spin function separation for the spin-orbital, this leads to an error compared to a truly exact exchange energy, which is the Hartree-Fock correlation energy [6]

$$E_C^{\mathrm{HF}} = E_{\mathrm{exact}} - E_{\mathrm{HF}}. \tag{2.24}$$

In the context of Density Functional Theory (DFT), Hartree-Fock Exchange sometimes also referred to as *exact exchange* because of the absence of any self-interaction error (SIE) [12].

This correlation energy arises because of the missing excitations given by additional determinants due the interaction of single electrons with the mean field of electrons, while the same-spin exchange interaction (Pauli principle) is still included. This correlation energy can be corrected by either expanding the wavefunction in linear combinations of Slater determinants, or a number of other post-Hartree-Fock methods, which will then increase the computational complexity again.

2.2. Density Functional Theory

Density Functional Theory is based on using the electronic density as a primary quantity rather than directly the orbitals which generate this density. Respectively, while in Hartree-Fock theory the Hamiltonian is solely defined by the potential v and the number of electrons N, in DFT it is the density which determines the external potential and hence the Hamiltonian, according to the first Hohenberg-Kohn theorem [13].

The density in this case is defined as an integral over the $3N$-dimensional wave function $\psi(\mathbf{r})$ as

$$\rho(\mathbf{r}_1) = N \int \cdots \int |\psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)|^2 d\mathbf{r}_2 \cdots d\mathbf{r}_N, \tag{2.25}$$

with the integral over this density yielding the total number of electrons in the system

$$\int \rho(\mathbf{r}) = N. \tag{2.26}$$

This allows us to write the energy expressed as a functional of this density in a similarly separated way as the Hartree-Fock energy in Equation 2.10, namely

$$\begin{aligned} E[\rho] &= \underbrace{T[\rho] + V_{\mathrm{ee}}[\rho]}\ + V_{\mathrm{ne}} \\ &= \quad F_{\mathrm{HK}}[\rho] \quad + \int \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r}, \end{aligned} \tag{2.27}$$

with the kinetic term T, and the electron-electron and nuclei-electron interaction terms V_{ee} and V_{ne}. The terms independent of the nuclei are collected in the common Hohenberg-Kohn functional F_{HK}, while the electron-electron interaction can be expressed as a sum of the classical Hartree repulsion term J and a non-classical part (which turns out to be part of the exchange-correlation energy term):

$$V_{\mathrm{ee}} = J[\rho] + \text{non-classical term}. \tag{2.28}$$

While this formulation is convenient, it all hinges on the availability of the HK functional, which unfortunately can only be determined for very

simple systems [14]. Luckily, the second Hohenberg-Kohn theorem introduces an analogue to the variational approach of the wave function theory in Equation 2.8. Namely that there is a ground state energy E_0 to the true ground state density ρ_0 and that for each trial density $\tilde{\rho}$ the inequality

$$E_0 \leq E[\tilde{\rho}] \tag{2.29}$$

holds. Together with a reformulation of the kinetic energy term T which still contains interacting and non-interacting parts, let us derive a solution. Kohn-Sham [15] introduced this by rewriting T as a sum of the kinetic energies of non-interacting electrons T_S (the non-interacting reference system) and a still unknown term V, together with the reintroduction of orbitals as defined before, allowing us to write

$$\begin{gathered}(T_s + V)\psi_s = E_s\psi_s \\ \text{and} \\ T_s[\rho] = \sum_i^N \left\langle \psi_i \middle| -\frac{1}{2}\nabla^2 \middle| \psi_i \right\rangle,\end{gathered} \tag{2.30}$$

with the density now again defined as in Equation 2.20.

This allows us to rewrite Equation 2.27 and Equation 2.28 as

$$\begin{aligned}E[\rho] &= T[\rho] + J[\rho] + T_s[\rho] - T_s[\rho] + \text{non-classical term} \\ &= T_s[\rho] + J[\rho] + E_{\text{xc}}[\rho],\end{aligned} \tag{2.31}$$

where the non-classical term and the missing correlation in the non-interacting kinetic energy are captured in the new E_{xc} energy. Together with the re-introduction of orbitals, the minimization procedure has to be adjusted in the sense that the orthonormality constraint from Equation 2.9 takes the place of the integration yielding the total number of electrons N. It must also be noted that this extension does not invalidate the variational principle introduced by the second Hohenberg-Kohn theorem. Hence the density-dependent energy term we want to minimize reads by putting Equation 2.30 and as Equation 2.31 together:

$$E[\rho] = \sum_i^N \left\langle \psi_i \middle| -\frac{1}{2}\nabla^2 \middle| \psi_i \right\rangle + J[\rho] + E_{\text{xc}}[\rho] + \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}. \tag{2.32}$$

This allows us now to define the Kohn-Sham operator which now contains functional derivatives of the density

$$\begin{aligned}K(\mathbf{r}) &= -\frac{1}{2}\nabla^2 + v(\mathbf{r}) + \frac{\delta J[\rho]}{\delta\rho(\mathbf{r})} + \frac{\delta E_{\text{xc}}[\rho]}{\delta\rho(\mathbf{r})} \\ &= v(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}d\mathbf{r}' + v_{\text{xc}}(\mathbf{r}),\end{aligned} \tag{2.33}$$

with the exchange-correlation potential v_{xc}. Again, as K is hermitian, we can find a unitary transformation which diagonalizes our energy eigenvalues such that we can without loss of generality write the Kohn-Sham equations in their canonical form

$$K(\mathbf{r})\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}), \tag{2.34}$$

which can again be turned into a general matrix eigenvalue problem by introducing a (finite) basis, to be solved numerically.

2.2.1. Exchange-Correlation Functionals of DFT

As we have been investigating the performance of basis sets (and pseudopotentials) with different exchange-correlation functionals, we will quickly discuss them here. While the Kohn-Sham equations are easier to solve, due to their density dependency, they now depend on the quality of the exchange-correlation potential v_{xc}. This leads to the exchange-correlation energy via $E_{\mathrm{xc}}[\rho] = \int \rho(\mathbf{r})v_{\mathrm{xc}}(\rho(\mathbf{r}))$. Even though this has been intensively studied in the past and new functionals are still being proposed, there is no formally systematic way of improving those functionals. Yet, there is a scheme proposed by Perdew [16] called "Jacob's Ladder", allowing for categorisation of them. The start at the lowest rung make functionals which depend solely on the density ρ. As they depend only on the functional at that given point in space or in other words on a constant electron density, they are called *Local (Spin) Density Approximation* (LDA). One step higher, the XC potentials depend additionally on the gradient of the density $\nabla\rho(\mathbf{r})$, giving them the name Generalized Gradient Approximation (GGA) and making them semi-local in the process. Going a step further, the *meta-GGA* are adding either a second derivative or orbital kinetic energy density τ, rendering them more accurate but making property calculation more sensitive to numerics. The next rungs towards chemical accuracy are then the *Hybrid Functionals*, which add exact exchange (as given above) to either GGA or meta-GGA functionals, rendering those functionals fully non-local. The final rung (for completeness) then includes the *Double-Hybrids*, which in turn include *Random Phase Approximation* (RPA) contributions or possibly others.

In the following, we are going to look at the explicit forms of the functionals used in this work: PBE, TPSS, PBEsol and PBE0. While all the functionals discussed here are based on the separation into an exchange and a correlation part $E_{\mathrm{x}} + E_{\mathrm{c}}$, this is not a general requirement.

PBE

The *Perdew, Burke, Ernzerhof* functional is arguably one of the most widely used functionals, especially in materials science [17]. As a GGA functional it depends on both the density and its gradient

$$E_{\mathrm{xc}}^{\mathrm{GGA}}[\rho] = \int \rho(\mathbf{r})v_{\mathrm{xc}}\left(\rho(\mathbf{r}), \nabla\rho\left(\mathbf{r}\right)\right). \tag{2.35}$$

The exchange part relies on the *Local Density Approximation* and augments it by an enhancement factor F_x, while introducing a reduced density gradient s

$$E_\mathrm{X} = \int \rho v_\mathrm{x}^\mathrm{LDA}(\rho) F_\mathrm{x}\left(s(\mathbf{r})\right) d\mathbf{r} \tag{2.36}$$

$$F(s) = 1 + \alpha - \frac{\alpha}{1 + \beta(s^2)} \tag{2.37}$$

$$v_\mathrm{x}^\mathrm{LDA}(\mathbf{r}) = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \rho(\mathbf{r})^{4/3} \tag{2.38}$$

$$s(\mathbf{r}) = \frac{|\nabla\rho(\mathbf{r})|}{\rho^{4/3}(\mathbf{r})}, \tag{2.39}$$

where the exchange energy density $v_\mathrm{x}^\mathrm{LDA}(\mathbf{r})$ has been obtained from the generalisation of the homogeneous electron gas to an inhomogeneous system under the assumption of a locally constant electron density [18].

The correlation part also derives directly from the LDA functional by

$$v_\mathrm{c}^\mathrm{PBE} = v_\mathrm{c}^\mathrm{LDA} + H(t), \tag{2.40}$$

where t is a dimensionless density gradient (e.g. of the form $|\nabla\rho|/\rho$, similar to s in Equation 2.39).

Even though highly parameterised (α, β but also the parameters contained in H), it must be stressed that those parameters are not derived from fitting the functional but rather from required asymptotic behaviour of the functional in different settings and additional constraints the functional should satisfy. The exact form and discussion can be found in Perdew, Burke, and Ernzerhof [19].

TPSS

The meta-GGA *Tao-Perdew-Staroverov-Scuseria* (TPSS) functional adds a dependency on τ, the Kohn-Sham orbital kinetic energy

$$E_\mathrm{xc}^\mathrm{GGA}[\rho] = \int \rho(\mathbf{r}) v_\mathrm{xc}(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}), \tau(\mathbf{r})) \tag{2.41}$$

$$\tau(\mathbf{r}) = \frac{1}{2}\sum_i^\mathrm{occ} |\nabla\phi_i(\mathbf{r})|^2. \tag{2.42}$$

The exchange part again depends on the previously shown $v_\mathrm{x}^\mathrm{LDA}$ with the enhancement factor now depending on τ, but similar to Equation 2.36 on auxiliary parameters:

$$F_{\mathrm{x}}(p,z) = 1 + \kappa - \frac{\kappa}{1 + x(p,z)/\kappa} \tag{2.43}$$

$$p = s^2 \tag{2.44}$$

$$z = \frac{\tau^W}{\tau}\,, \text{with}\, \tau = \tau_\uparrow + \tau_\downarrow\,, \text{and}\, \tau^W = \frac{1}{8}\frac{|\nabla\rho|^2}{\rho} \tag{2.45}$$

The form of F_{x} is again chosen to obey the Lieb-Oxford bound [20] with a fixed κ.

The correlation part derives in a similar manner from the PBE correlation energy shown above, adding a τ dependency in the process.

For this and the full form of $x(p,z)$ we are referring again to the original source of Tao et al. [21]. What makes this functional interesting to be used as a prototype for meta-GGAs in the following work, is its property of correcting for the one-electron SIE, by ensuring the correlation energy vanishes exactly in the valence region.

PBEsol

Even though the parametrization of PBE relies on general physical requirements towards the functional, there is still a significant degree of freedom when it comes to which and how well those constraints are respected. While PBE respects and reproduces atomization energies well in the process, it is far less accurate when it comes to weakly-varying densities in densely packed solids, something the LDA gets right. To remedy this issue, two parameters in the PBE functional were adjusted (part of the *beta* and H functions mentioned above), by fitting to the behaviour of the previously mentioned TPSS functional for surface densities in a homogeneous electron gas setup. As the GGA form itself is limited, this worsens for example atomization energies (but not worse than LDA) and makes it less suitable for molecular calculations, but improves upon bulk properties like lattice constants and surface energies [22].

PBE0

The PBE0 functional is a hybrid functional, directly mixing PBE and the exact exchange energy from Hartree-Fock in a straightforward way, while maintaining the complete correlation energy from PBE:

$$E_{\mathrm{xc}}^{\mathrm{PBE0}} = \frac{1}{4}E_{\mathrm{x}}^{\mathrm{HF}} + \frac{3}{4}E_{\mathrm{x}}^{\mathrm{PBE}} + E_{\mathrm{c}}^{\mathrm{PBE}} \tag{2.46}$$

The motivation for this comes from the re-introduction of orbitals in Kohn-Sham DFT, which allows us to define an exact Kohn-Sham exchange energy and one of the major differences between Hartree-Fock and DFT when it comes to Self-Interaction Correction (SIC): While in Hartree-Fock self-interaction is cancelled because the respective contribution is present in both Hartree and Exchange terms, no equivalent mechanism is available in DFT due to the density “intermediary” which gets integrated over to obtain the energies. With the mixing one is able to

(partially) recover some of the properties most influenced by this, namely atomisation energies, bond lengths and vibration frequencies [23].

Even though this way of mixing DFT and Hartree-Fock (HF) exchange energy may look arbitrary, with mixing coefficients in hybrid functionals either determined empirically or by required asymptotic behaviour, their use is rigorously justified from DFT itself via the *adiabatic connection* of the Exchange-Correlation energy [24]

$$E_{\mathrm{XC}} = \int_0^1 U_{\mathrm{XC}}^{\lambda} d\lambda. \tag{2.47}$$

Which connects a non-interacting Kohn-Sham system with its fully-interacting real counterpart. In the case of PBE0, the $\frac{1}{4}E_{\mathrm{x}}^{\mathrm{HF}}$ can be exactly recovered by an analysis of the behaviour of electron pair spin states in DFT [25].

An important part worth mentioning is that physical meaning is only obtained for the whole of the Exchange-Correlation energy, hence a segmentation into E_{X} and E_{C} and then replacing either does not work, as realized by Becke [26], which subsequently led to the creation of the B88 functional [27, 28].

2.3. Basis Sets and Pseudopotentials

As mentioned in the previous section, to actually be able to solve the Schrödinger equation numerically, we have to introduce a basis in which we express the MOs. For the largest part of this work, we relied on the CP2K software package, which utilises atom-centred Gaussian-type orbitals (GTO).

While a natural choice for a basis would be Slater functions of the type $e^{-\alpha r}$ as they form the analytical solution to the single particle Schrödinger equation, we rely in several parts – like in the Fock matrix – on Electron Repulsion Integrals (ERIs), which do not have an analytical solution for such functions. For that purpose Gaussian functions $e^{-\alpha r^2}$ are much more efficient to calculate as the product of two Gaussian functions around different centres A and B can be rewritten as a single Gaussian function around a common centre C. The disadvantage of functions of the form $e^{-\alpha r^2}$ is that they do not reproduce the cusp forming at the nucleus (e.g. $\frac{\partial \phi(r)}{\partial r} \neq 0$ at $r = 0$) and that they decay too quickly in the long-range (due to the r^2 in the exponent).

For both types of functions, α determines the *diffuseness* of the basis, a larger value leading to a narrower and a lower coefficient to a broader distribution.

Using single (or primitive) Gaussian functions in place of a Slater-type orbital would require a lot of basis functions to recover the same characteristics, which is why contracted Cartesian Gaussian basis functions (centred around a nucleus A) are being used for a single Slater function with the contraction coefficients c_i either fitted to Slater orbitals or to data. Such a Gaussian function is then written as

$$\psi_\nu(\mathbf{r}-\mathbf{A}) = \sum_{i=1}^{L} c_i (x-A_x)^l (y-A_y)^m (z-A_z)^n e^{-\alpha_i(\mathbf{r}-\mathbf{A})^2}, \tag{2.48}$$

with L determining the *length of the contraction*, and $l+m+n$ corresponds to the Cartesian angular momenta. With that, the MOs of Equation 2.15 and the Electron Repulsion Integrals of Equation 2.22 become

$$\phi_i(\mathbf{r}-\mathbf{A}) = \sum_{\nu} C_{\nu i} \sum_{i=1}^{L} c_i (x-A_x)^l (y-A_y)^m (z-A_z)^n e^{-\alpha_i(\mathbf{r}-\mathbf{A})^2} \tag{2.49}$$

and

$$\begin{aligned}(\mu\nu \mid \lambda\sigma)_g = \sum_{a=1}^{L_A}\sum_{b=1}^{L_B}\sum_{c=1}^{L_C}\sum_{d=1}^{L_D} \iint \frac{1}{|\mathbf{r}_2-\mathbf{r}_1|} \\ \cdot c_a (x-A_x)^{l_A} \left(y-A_y\right)^{m_A} (z-A_z)^{n_A} \\ \cdot c_b (x-B_x)^{l_B} \left(y-B_y\right)^{m_B} (z-B_z)^{n_B} \\ \cdot c_c (x-C_x)^{l_C} \left(y-C_y\right)^{m_C} (z-C_z)^{n_C} \\ \cdot c_d (x-D_x)^{l_D} \left(y-D_y\right)^{m_D} (z-D_z)^{n_D} \\ \cdot e^{-\alpha_a(\mathbf{r}_1-\mathbf{A})^2} e^{-\alpha_b(\mathbf{r}_1-\mathbf{B})^2} e^{-\alpha_c(\mathbf{r}_2-\mathbf{C})^2} e^{-\alpha_d(\mathbf{r}_2-\mathbf{D})^2} \\ d\mathbf{r}_1 d\mathbf{r}_2,\end{aligned} \tag{2.50}$$

each a function of length of the contraction L, the angular momenta l, m, n and atomic centres **A**, **B**, **C** and **D**. From the above it is immediately clear then that the number of ERIs to calculate depends heavily on the size of the basis set, favouring small and heavily contracted basis sets (low number of contractions).

Expanding atomic orbitals in terms of atom-centred Gaussians comes naturally and is standard for quantum chemistry codes as it permits direct insight into chemical properties like bonds. Expressing the electron density required in KS-DFT in plane wave (PW) instead, which are independent of the atomic positions, instead has several benefits as well. In particular can the Hartree and Exchange-Correlation (XC) energies be calculated more easily and the complexity of computing the Kohn-Sham matrix follows $\mathcal{O}(N\log(N))$ (the underlying discrete Fast-Fourier Transformation (FFT)) if combined with the Gaussian expansion approach in a dual Gaussian and Plane Waves Method (GPW) approach [29]. The weakness of PWs lies in their approximation of the density near the atomic centres: to represent the density accurately enough, a large number of PWs must be employed (i.e. a high cutoff must be chosen), which comes at a large cost in terms of memory and computation as the same representation must be maintained for regions where the accuracy is not needed. This can be remedied by using pseudopotentials (PPs) which collect the potential generated by the nucleus together with the non-valence electrons into one effective core potential, often based on a frozen-core assumption. The role of the PP is then to ensure that valence electrons are

kept out of the core and within the valence space, and must thus contain a repulsive short-range and an attractive long-range part. [30]

For both PW and Gaussian-type Orbital (GTO) approaches do PPs reduce the number of required basis functions within a calculation since the (localised) core behaviour does not have to be captured anymore. For GTOs this comes also into play during basis set construction by permitting more compact basis functions. It should be noted that PP rely on a valence and core electron partition which must be obtained prior to the generation of PP by some other method. This is accurate enough for many calculations but physically not fully correct. For species where this becomes a problem (for example heavier alkali) this can be remedied by treating more electrons as valence electrons (then called *semi-core* electrons), resulting in fewer core electrons covered by the PP.

For heavier atoms also relativistic effects can be directly taken into account via the PP, simplifying the calculations.

While there are many different forms of PP as illustrated by Dolg and Cao [31], we are only going to mention the ones relevant for this work, the norm-conserving [32] Separable Dual-Space Pseudopotentials of Goedecker, Teter, and Hutter [33] (GTH), respectively Hartwigsen, Goedecker, and Hutter [34] (HGH).

This means that they take the general form

$$V(\mathbf{r},\mathbf{r}') = V_{\text{loc}}(r)\delta(|\mathbf{r}-\mathbf{r}'|) + \sum_l V_{\text{l}}(\mathbf{r},\mathbf{r}') + \Delta V_l^{\text{SO}}(\mathbf{r},\mathbf{r}')\mathbf{L}\cdot\mathbf{S}, \tag{2.51}$$

with the local part given by

$$\begin{aligned} V_{\text{loc}}(r) =& \frac{-Z_{\text{ion}}}{r}\,\text{erf}\left(\frac{r}{\sqrt{2}r_{\text{loc}}}\right) + \exp\left[-\frac{1}{2}\left(\frac{r}{r_{\text{loc}}}\right)^2\right] \\ &\cdot\left[C_1 + C_2\left(\frac{r}{r_{\text{loc}}}\right)^2 + C_3\left(\frac{r}{r_{\text{loc}}}\right)^4 + C_4\left(\frac{r}{r_{\text{loc}}}\right)^6\right], \end{aligned} \tag{2.52}$$

where erf is the error function, Z_{ion} the ionic charge of the core (i.e. without the charge of the valence elctrons).

The non-local part V_{l} is given as

$$V_{\text{l}}(\mathbf{r},\mathbf{r}') = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{m=-l}^{+l} Y_{l,m}(\hat{\mathbf{r}})p_i^l(r)h_{ij}^l p_j^l(r')Y_{l,m}^*(\hat{\mathbf{r}}'), \tag{2.53}$$

with the projectors p_i^l being products of a polynomial and a Gaussian function in both real and reciprocal space (*dual-space*) and $Y_{l,m}$ the spherical harmonics.

A different solution than the use of PP to the issue of modelling the true wave functions around the core is the use of separate grids. In a PW code this can be resolved by replacing the plane-wave functions by a range-separated basis, which uses a sum of products of radial functions again with spherical harmonics inside a certain *muffin-tin* radius, and plane-waves outside (the *interstitial region*). The radii are then chosen as big as possible without overlap. This gives rise to the Full-Potential Linearized

Augmented Plane Wave method (FP-LAPW)[35], as implemented for example by SIRIUS, Exciting and Wien2k.

In a GPW code such as CP2K, one can use a conceptually similar approach of treating regions around atoms differently, keeping in mind that only the density is expanded into PW, the Gaussian and Augmented Planewave (GAPW) method [36–38]. In GAPW, the sharp variations of the density close to the nuclei is captured on (radial) atomic grids, while the smoothly varying density in the interstitial regions can be described by a low amount of PW. The requirement for this is the ability of separating the Hartree and XC functionals into independent global and local contributions as proven possible in the Projector Augmented-Wave method (PAW) approach by Blöchl [39]. For the local atom centred representation one chooses again a projector basis based on Gaussians. Despite that, by ensuring non-overlapping regions (and assuming contributions by other centres are negligible), the cost of calculating the local densities is system-size independent. Finally, even though there is a partitioning of space for consistency of the theory, no explicit boundaries where functions have to match, as in the FP-LAPW approach, appear. And, the GAPW method is not limited to All-Electron (AE) calculations but can also employed together with PP for improved accuracy.

From the above, it becomes clear that any non-AE basis set depends on a corresponding PP in terms of number of valence electrons. And since the GTH-PPs depend on the functional used when obtaining their parameters, so do the matching basis sets. Because basis sets are not transferable between different computational methods (e.g. DFT and correlated wave function methods) [40, 41], in this work we will investigate their transferability within the same rung of DFT.

Focusing on the basis functions primarily used in this work, we are quickly looking at how the MOLOPT basis sets are constructed. They are generally (respectively fully) contracted[42] GTO basis sets with the general form of

$$\begin{aligned} \varphi_j &= \sum_i c_{i,j}\chi_i \\ \chi_i(\alpha, n, m, l; r, \theta, \phi) &= NY_{l,m}(\theta, \phi) r^{2n-2-l} \mathrm{e}^{-\alpha r^2}, \end{aligned} \tag{2.54}$$

meaning that the same exponents, respectively primitive Gaussians, are shared across all contracted Gaussian functions forming the basis set, with non-zero contraction coefficients. This is in contrast to minimal and segmented basis sets, where contractions are carefully built based on the anticipated representation of MOs, and diffuse Gaussians (small exponents) are therefore often retained as single basis functions. Consequently a double- or triple-ζ MOLOPT basis set again uses the same exponents for the second and third contracted functions, but with different contraction coefficients. Originally drafted as basis sets to be used with PP, the MOLOPT basis sets contain cover only *valence* electrons and must be used with matching PP. Starting at double-ζ they also contain additional polarisation functions in the same way as the ** variants of the popular Pople 6-31G basis sets [43] (e.g. the polarisation functions are not doubled/tripled), but again based on the full contraction instead, and denoted with a P suffix. The general series of contractions in the MOLOPT basis sets is therefore: SZV (1s1p/1s), DZVP

(2s2p1d /2s1p), TZVP (3s3p1d/3s1p), TZV2P (3s3p2d/s2p), and TZV2PX (3s3p2d1f/3s2p1d) , whereas the newly optimised set evaluated in this thesis only covers DZVP, TZVP and TZV2P.

The rational behind those decisions is based on the observations that for DFT the computationally intensive task is the update of the density matrix, which depends on the number of basis functions [44, 45], rather than the primitives. More relevant are small basis set superposition errors (BSSEs) and a small condition number[1] of the overlap matrix to achieve stable optimisation. The latter is important since the usual techniques of treating this problem is by removing eigenvectors belonging to small eigenvalues, which may lead to discontinuities in the energy in either geometry optimisation or molecular dynamics (MD) use cases as the system evolves. The former can be alleviated by the addition of diffuse functions, which also a requirement for the proper description of weak interactions such as hydrogen bonding. It has been shown, that these highly contracted MOLOPT basis sets satisfy all those criteria [46–49].

1: ratio of largest energy to the smallest eigenvalue, directly affects the sparsity and hence the performance of the overlap and matrices derived from it

With the same primitives reused across basis functions, a final design decision was then that larger basis sets extend their respective smaller ones. E.g. by neglecting the third basis function of a TZVP basis for each angular momentum, one ends up at the DZVP basis of the same species. This makes it much easier to argue that larger basis sets extend the quality of their respective smaller ones.

In contrast to the original MOLOPT procedure, the revised set follows the *even-tempered* approach [50, 51] for selecting the exponents of the primitive Gaussians. What this means is that rather than using N independent exponents, they are instead related via

$$\alpha_k = \alpha_1 \varepsilon^{k-1}. \tag{2.55}$$

This greatly simplifies the non-linear optimisation procedure, but there is also another aspect. The overlap of Gaussian primitives for the same angular momentum only depends on the ratio of their exponents. If the exponents are defined as in Equation 2.55 this means that the ratio between two adjacent primitives is exactly ε. Not only that, but it can be shown that "such a constant overlap will lead to even coverage of the Hilbert space" [52].

The new AE variants of the MOLOPT basis sets follow the *def2-SVP*, *-TZVPP*, *-QZVPP* contraction scheme [53, 54] instead.

The smaller number of basis functions is useful to reduce the size of the density matrix important for DFT calculations. But the large number of primitives becomes problematic for hybrid calculations in which the HFX ERI must be calculated. This is due to the dependency on the primitive Cartesian Gaussian functions as shown in Equation 2.50, whose number is at least equal or even larger than the spherical ones [55]. Another problem for HFX are the diffuse functions as they limit the application of Schwarz-screening discussed in Section 2.4.1 and lead to significantly larger number of quartets contributing.

2.4. Periodic Hartree-Fock Exchange

One of the results of condensed matter theory is the realisation that in a periodic system (e.g. a crystal) the solutions to the Schrödinger equation – the wave functions – must satisfy the Bloch theorem, stating that they can always be written as Bloch functions

$$\psi_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}}u(\mathbf{r}), \tag{2.56}$$

with $u(\mathbf{r}) = u(\mathbf{r}+\mathbf{R})$ being of lattice periodicity and $\mathbf{R}$ being a periodic cell displacement vector. As a further consequence, the uniqueness of $\psi_{\mathbf{k}}$ is only guaranteed up to a displacement of a reciprocal lattice vector $\mathbf{G}$

$$\psi_{\mathbf{k}+\mathbf{G}}(\mathbf{r}) = \psi_{\mathbf{k}}(\mathbf{r}). \tag{2.57}$$

allowing us to restrict $\mathbf{k}$ to lie within the first Brillouin Zone (BZ). If we are to introduce primitive vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ such that

$$\mathbf{R} = \sum_{i=1}^{3} n_i \mathbf{a}_i, \tag{2.58}$$

with n_i being integer numbers, spawns our Bravais lattice, we can also define a set of reciprocal lattice vectors by

$$\mathbf{b}_1 = 2\pi\frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{b}_2 = 2\pi\frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{b}_3 = 2\pi\frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \tag{2.59}$$

in which to express $\mathbf{k}$ as

$$\mathbf{k} = \sum_{i=1}^{3} x_i \mathbf{b}_i, \tag{2.60}$$

with x_i being real valued numbers, and the relationship $\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi\delta_{ij}$ between the two basis.

Applying the Born-von Kármán (BvK) boundary conditions by defining our crystal to consist of a finite number $N = N_1 N_2 N_3$ of primitive cells such that $\psi_{\mathbf{k}}(\mathbf{r} + N_i\mathbf{a}_i) = \psi_{\mathbf{k}}(\mathbf{r})$ then leads to

$$e^{i\mathbf{k}\cdot(\mathbf{r}+N_i\mathbf{a}_i)} = e^{i\mathbf{k}\cdot\mathbf{r}} \Rightarrow e^{i\mathbf{k}\cdot N_i\mathbf{a}_i} = 1 \ \forall i \in \{1,2,3\}, \tag{2.61}$$

and thus

$$\mathbf{k} = \sum_{i=1}^{3} \frac{m_i}{N_i}\mathbf{b}_i, \tag{2.62}$$

with m_i being integer numbers again. The reciprocal space can therefore be sampled at a finite number of $\mathbf{k}$-points given by the N_i.

For our linear combination of atomic orbitals (LCAO) basis, this means that the MOs are now dependent on $\mathbf{k}$

$$\psi_i(\mathbf{r};\mathbf{k}) = \sum_\mu C^{\mathbf{k}}_{\mu i}\phi_\mu(\mathbf{r};\mathbf{k}) \tag{2.63}$$

and are now in fact Crystalline Orbitals (COs).

Plugging those back into the HF Equation 2.18 then directly leads to

$$\sum_\mu F^{\mathbf{k}}_{\nu\mu}C^{\mathbf{k}}_{\mu i} = \epsilon^{\mathbf{k}}_i \sum_\mu S^{\mathbf{k}}_{\nu\mu}C^{\mathbf{k}}_{\mu i}, \tag{2.64}$$

with the important distinction that the integration for the overlap matrix $S_{\nu\mu}$ and the Fock matrix $F_{\nu\mu}$ is now over the space defined by the BvK boundary condition.

While this is very convenient to evaluate in PW codes, for a GPW code such as CP2K we need a real space formulation. The key to this is to start from the atom centred functions φ_μ in a reference unit cell and expand them into a Bloch sum [56, 57]

$$\phi_\mu(\mathbf{r};\mathbf{k}) = \frac{1}{\sqrt{N}}\sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}}\varphi_\mu(\mathbf{r}-\mathbf{R}), \tag{2.65}$$

which is essentially a discrete Fourier transform when taking Equation 2.62 into account. This satisfies the Bloch theorem in Equation 2.56, considering that the summation is in principle over infinite R.

Extending this to the overlap matrix then let's us define real space (neighbour cell dependent) variants as

$$\begin{aligned} S^{\mathbf{k}}_{\nu\mu} &= \sum_{\mathbf{R},\mathbf{R}'}\frac{1}{N}\int e^{i\mathbf{k}(\mathbf{R}'-\mathbf{R})}\phi^\star_\nu(\mathbf{r}-\mathbf{R})\phi_\mu(\mathbf{r}-\mathbf{R}')d\mathbf{r} \\ &= \sum_{\mathbf{Q}=\mathbf{R}'-\mathbf{R}}\frac{1}{N}\int e^{i\mathbf{k}\mathbf{Q}}\phi^\star_\nu(\mathbf{r})\phi_\mu(\mathbf{r}-\mathbf{Q})d\mathbf{r} \\ &= \frac{1}{N}\sum_{\mathbf{Q}} e^{i\mathbf{k}\mathbf{Q}}\int \phi^\star_\nu(\mathbf{r})\phi_\mu(\mathbf{r}-\mathbf{Q})d\mathbf{r} \\ &= \frac{1}{N}\sum_{\mathbf{Q}} e^{i\mathbf{k}\mathbf{Q}}S^{\mathbf{Q}}_{\nu\mu}, \end{aligned} \tag{2.66}$$

and likewise for the density matrix, but with the integration taking place over the Brillouin zone.

As a four-centre quantity, the same transformation becomes slightly more involved when looking at the exchange matrix from Equation 2.21, which now reads as follows and where the expansion leads to additional summations over $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4$ in the same way as the summation over $\mathbf{R}, \mathbf{R}'$ before:

$$
\begin{aligned}
F^{\mathbf{k}}_{\mu\sigma} &= -\frac{1}{2}\sum_{\nu\lambda\mathbf{k}'} P^{\mathbf{k}'}_{\nu\lambda} \iint \phi^*_\mu(\mathbf{r}_1;\mathbf{k})\phi_\nu(\mathbf{r}_1;\mathbf{k}')\frac{1}{\|\mathbf{r}_2-\mathbf{r}_1\|}\phi^*_\lambda(\mathbf{r}_2;\mathbf{k}')\phi_\sigma(\mathbf{r}_2;\mathbf{k})d\mathbf{r}_1 d\mathbf{r}_2 \\
&= -\frac{1}{2N^2}\sum_{\nu\lambda\mathbf{k}'} P^{\mathbf{k}'}_{\nu\lambda} \sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4} e^{-i\mathbf{k}(\mathbf{R}_1-\mathbf{R}_4)}e^{-i\mathbf{k}(\mathbf{R}_2-\mathbf{R}_3)} \\
&\quad \iint \phi^*_\mu(\mathbf{r}_1-\mathbf{R}_1)\phi_\nu(\mathbf{r}_1-\mathbf{R}_2)\frac{1}{\|\mathbf{r}_2-\mathbf{r}_1\|}\phi^*_\lambda(\mathbf{r}_2-\mathbf{R}_3)\phi_\sigma(\mathbf{r}_2-\mathbf{R}_4)d\mathbf{r}_1 d\mathbf{r}_2.
\end{aligned}
\tag{2.67}
$$

By translation symmetry of the crystal we can shift the entirety again by $\mathbf{R}_1$ and introduce new displacement vectors $\mathbf{Q}, \mathbf{S}, \mathbf{T}$ instead, together with the definition of the real space density matrix $P^{\mathbf{Q}}_{\nu\mu} = \frac{1}{N}\sum_{\mathbf{k}} e^{i\mathbf{k}\mathbf{Q}} P^{\mathbf{k}}_{\nu\mu}$, which then leads to the general expression for the exchange part of the periodic Fock matrix of

$$
K^{\mathbf{T}}_{\mathrm{x}} = -\frac{1}{2}\sum_{\mathbf{Q}}\sum_{\nu\lambda} P^{\mathbf{Q}}_{\nu\lambda}\sum_{\mathbf{S}}(\mu^0\nu^{\mathbf{S}} \mid \lambda^{\mathbf{T}}\sigma^{\mathbf{S}+\mathbf{Q}}). \tag{2.68}
$$

Contracting with the density matrix $P_{\mu\nu}$ then leads to the periodic Hartree-Fock exchange energy

$$
E_{\mathrm{x}} = -\frac{1}{2}\sum_{Q,T}\sum_{\nu\lambda} P^{T}_{\mu\sigma}P^{Q}_{\nu\lambda}\sum_{S}(\mu^0\nu^{S} \mid \lambda^{T}\sigma^{S+Q}). \tag{2.69}
$$

When only taking the Γ point into consideration, the density matrix in real space becomes the same for each periodic image, letting us simplify the sum to

$$
E_{\mathrm{x}} = -\frac{1}{2}\sum_{\lambda\mu\nu\sigma} P_{\mu\sigma}P_{\nu\lambda}\sum_{S,Q,T}(\mu^0\nu^{S} \mid \lambda^{T}\sigma^{S+Q}), \tag{2.70}
$$

which is what prior to this work has been implemented in CP2K by Guidon et al. [58]. Since the 4c-ERIs $(\mu^0\nu^S|\lambda^T\sigma^{S+Q})$ do not depend on the MOs which are being optimised in the self-consistent field (SCF) loop, but only on the atom centre, they only have to be updated when the geometry changes, e.g. in a MD or geometry optimisation run. Together with a precision truncation scheme, usage of symmetries and screening at different levels of the calculation, this allowed implementation of an in-memory and an on-disk cache between SCF cycles, speeding up the calculation significantly.

Unfortunately, this caching implementation turns unfeasible for the general $\mathbf{k}$ point case as the number of coefficients to store grows quadratically with the number of neighbour cells to consider, given that only the sum over S can be executed before contraction with the density matrix.

2.4.1. Screening

Reducing the number of ERIs to calculate is of utmost importance to keep the calculations for larger systems tractable. In particular if HFX is to be combined with DFT, which can scale quasi-linearly, an $\mathcal{O}(N^4)$ scaling

with the system size as derived in Equation 2.50 is undesirable. The ability to determine a-priori which integrals are significant is therefore key to reducing the scaling. Since the Fock matrix F^{x} is a contraction of the density matrix $P_{\lambda\nu}$ with ERIs $F_{\nu\lambda} = \sum_{\mu\sigma}(\mu\nu|\lambda\sigma)$, the selection of which integrals to calculate based on their contribution – the *screening* – can be based on either quantity.

For the integral screening on the ERI, the main idea is the (Cauchy-)Schwarz-inequality

$$|(\mu\nu \,|\, \lambda\sigma)| \leq |(\mu\nu \,|\, \mu\nu)|^{1/2} \cdot |(\lambda\sigma \,|\, \lambda\sigma)|^{1/2}, \tag{2.71}$$

which holds for for any inner product. While the requirements for an inner product like conjugate symmetry and linearity are fulfilled by the integral definition, the positive definiteness has been shown by Roothaan [59] for the Coulomb metric, while [60] have shown this to be rigorous upper bound.

Together with the maximum of the respective density matrix element $\bar{P} = \max\left\{\left|P_{\mu\lambda}\right|, |P_{\nu\lambda}|, \left|P_{\mu\sigma}\right|, |P_{\nu\sigma}|\right\}$ we can then define an estimate, which when smaller than a given threshold (named `EPS_SCHWARZ` in CP2K) leads to a neglect of the integral in Equation 2.71:

$$\mathrm{EST} = |(\mu\nu \,|\, \mu\nu)|^{1/2} \cdot |(\lambda\sigma \,|\, \lambda\sigma)|^{1/2}. \tag{2.72}$$

This means by evaluating a subset of integrals, and continued application of the inner product, one can obtain a hierarchy of screenings for:

1. pairs of basis functions belonging to the same atom (in fact their atomic kinds) by evaluating their largest estimate,
2. sets of of Gaussian basis functions (a grouping based on equal angular momentum) where again their maximum contribution is estimated,
3. and finally uncontracted primitive Gaussian functions (PGFs).

The buildup of large screening matrices can further be avoided by instead calculating fitting parameters for a second-order polynomial for the product of the co-densities at different distances

$$(\mu\nu \,|\, \mu\nu)(r_{\mu\nu}) = c_2 \cdot r_{\mu\nu}^2 + c_0. \tag{2.73}$$

2.5. The Auxiliary Density Matrix Method

In Kohn-Sham Density Functional Theory (DFT) the total energy of the system

$$E[\rho] = T_s[\rho] + J[\rho] + E_{\mathrm{xc}}[\rho] + \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \tag{2.74}$$

is defined as a functional dependency of the density

$$\rho(\mathbf{r}) = \sum_{i=1}^{N_e} |\psi_i(\mathbf{r})|^2, \tag{2.75}$$

with N_e being the number of electrons in the system, and $\psi_i(\mathbf{r})$ the (real-valued) single particle wave functions. The terms in the total energy denote (in that order) the kinetic, Hartree and exchange-correlation energies followed by the part from the external potential.

For a Hybrid DFT calculation, a fraction α of the exchange energy E_x of the exchange-correlation potential E_{xc} is replaced by the Hartree-Fock exact exchange

$$\begin{aligned} E_{\text{xc}}^{\text{hybrid}}[\rho] &= E_{\text{xc}}^{\text{DFT}}[\rho] + \alpha \left(E_{\text{x}}^{\text{HFX}}[\{\psi_i\}] - E_{\text{x}}^{\text{DFT}}[\rho] \right) \\ &= E_{\text{c}}^{\text{DFT}}[\rho] + \alpha E_{\text{x}}^{\text{HFX}}[\{\psi_i\}] + (1-\alpha) E_{\text{x}}^{\text{DFT}}[\rho]. \end{aligned} \tag{2.76}$$

When using a LCAO as a basis set, the single-particle wave functions can be written as

$$\psi_i(\mathbf{r}) = \sum_{\mu} C_{\mu i} \phi_\mu(\mathbf{r}) \tag{2.77}$$

with $C^{\mu i}$ the MO coefficients. From this, the Hartree-Fock exchange energy can be expressed as a functional of the density matrix

$$P^{\mu\nu} = \sum_i C_{\mu i} C_{\nu i} \tag{2.78}$$

and two-electron four-centre electron repulsion integrals (4c-ERIs)

$$E_x^{\text{HFX}}[P] = -\frac{1}{2} \sum_{\lambda\sigma\mu\nu} P^{\mu\sigma} P^{\nu\lambda} (\mu\nu|\lambda\sigma), \tag{2.79}$$

With the matrix-matrix multiplication scaling as $\mathcal{O}(N^3)$ of the number of basis functions, the calculation of the HFX energy scales per Equation 2.79 as $\mathcal{O}(N^4)$. We will shortly see that, while in periodic calculations at the Γ-point sums of ERIs can be cached, effectively trading computation time with storage, the same trick can not be applied in the more general K-point formulation. Reducing the basis set size or choosing a basis with a larger minimal exponent to achieve a more rapidly decaying, respectively sparser density matrix is, therefore, desirable to accelerate the evaluation of the HFX energy. The Auxiliary Density Matrix Method (ADMM) [61] solves this by introducing such an auxiliary density matrix $\hat{P} \approx P$. Together with the assumption that the difference in the exchange energies calculated for the primary and the auxiliary density matrix (respectively the corresponding density) between the DFT and the Hartree-Fock exchange is small enough, we can rewrite the HFX energy for the primary density matrix as

$$\begin{aligned} E_x^{\mathrm{HFX}}[P] &= E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{HFX}}[P] - E_x^{\mathrm{HFX}}[\hat{P}]\right) \\ &\approx E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{DFT}}[\rho] - E_x^{\mathrm{DFT}}[\hat{\rho}]\right). \end{aligned} \tag{2.80}$$

While there are multiple ways to construct such an auxiliary density matrix $\hat{P}$, we are restricting ourselves to the ones constructed by introducing an auxiliary basis set $\hat{\phi}_\mu(\mathbf{r})$ to express the wave function

$$\hat{\psi}_i(\mathbf{r}) = \sum_\mu \hat{C}_{\mu i}\hat{\phi}_\mu(\mathbf{r}), \tag{2.81}$$

which directly leads us to an auxiliary density matrix

$$\hat{P}^{\mu\nu} = \sum_i \hat{C}_{\mu i}\hat{C}_{\nu i}. \tag{2.82}$$

The MO coefficients $\hat{C}^{\mu i}$ can be determined by minimising the square difference for the occupied wave functions expressed in the auxiliary and primary basis set:

$$\min_{\hat{C}} \sum_j \int \left(\psi_j(\mathbf{r}) - \hat{\psi}_j(\mathbf{r})\right)^2 dr. \tag{2.83}$$

With the overlap matrices for the auxiliary basis and the one between the auxiliary and primary basis

$$\hat{S}_{nn'} \coloneqq \int \hat{\phi}_n(\mathbf{r})\hat{\phi}_{n'}(\mathbf{r})dr \quad \text{and} \quad Q_{nm} \coloneqq \int \hat{\phi}_n(\mathbf{r})\phi_m(\mathbf{r})dr \tag{2.84}$$

we can write the solution for the auxiliary MO coefficients subject to the minimising Equation 2.83 as

$$\hat{C} = \underbrace{\hat{S}^{-1}Q}_{=:A} C \tag{2.85}$$

Additional constraints – like the required orthonormality – can be integrated in Equation 2.83 using Lagrange multipliers:

$$\min_{\tilde{C}} \left[\sum_j \int \left(\psi_j(\mathbf{r}) - \hat{\psi}_j(\mathbf{r})\right)^2 d\mathbf{r} + \sum_{k,l} \Lambda_{k,l} \left(\int \hat{\psi}_k(\mathbf{r})\hat{\psi}_l(\mathbf{r})d\mathbf{r} \right) - \delta_{kl} \right]. \tag{2.86}$$

Which yields the solution

$$\tilde{C} = \hat{C}\Lambda^{-1/2} = \hat{C}(\hat{C}^T\hat{S}\hat{C})^{-1/2}. \tag{2.87}$$

The density matrices required for the HFX energy are therefore analogue to Equation 2.78 given by

$$\begin{aligned}\hat{P} &= \hat{C}\hat{C}^T = APA^T \\ \tilde{P} &= \tilde{C}\tilde{C}^T = \hat{C}\Lambda^{-1}\hat{C}^T\end{aligned} \tag{2.88}$$

The total energy of the system can be written as a sum of the energy depending on the primary design matrix $E[P]$ and one depending on the auxiliary matrix $\tilde{E}[\tilde{P}]$:

$$E_{\text{total}} = E[P] + \tilde{E}[\tilde{P}]. \tag{2.89}$$

For the final Kohn-Sham matrix, this then means that it too can be expressed by a simple sum of the original and projected auxiliary Kohn-Sham matrices

$$K_{\text{total}} = K[P] + A^T\tilde{K}A. \tag{2.90}$$

The ADMM implementation in which the auxiliary density matrix is obtained by solving the optimisation in Equation 2.83 is referred to as *ADMM2* or *non-purified wavefunction fitting*, the one based on Equation 2.86 yielding orthonormal orbitals *ADMM1*, or *purified* wavefunction fitting).

A third variant *ADMM3* can be derived for systems where the density matrix is blocked due to the physical setup, i.e. where the system can be divided into subsystems and most of the significant exchange between those subsystems is captured by the *GGA* functional. In this case one can express the auxiliary density matrix as the Hadamard product $(A \circ B)_{ij} = (A)_{ij}(B)_{ij}$ of the original density matrix and a blocking matrix

$$\mathring{P} \coloneqq P \circ B, \tag{2.91}$$

with $B_{ij} \in \{1, 0\}$, and thus setting coefficients corresponding to inter-subsystem interaction to 0.

The term *purification* refers to the fact that a regular or *pure* density matrix is symmetric, idem-potent and conserves the number of particles:

$$\begin{aligned}P &= P^T, \\ PSPS &= PS, \\ \text{tr}(PS) &= N_e,\end{aligned} \tag{2.92}$$

which is naturally only the case for *ADMM1*. For *ADMM2* however one can recover the *ADMM1 pure* density matrix with the purification algorithm by McWeeny [62], here given in its original form

$$\begin{aligned}\bar{P}_{n+1} &= 3\bar{P}_n S\bar{P}_n - 2\bar{P}_n S\bar{P}_n S\bar{P}_n, \\ \tilde{P} &\coloneqq \lim_{n\to\infty} \bar{P}_n,\end{aligned} \tag{2.93}$$

with $\bar{P}_0 = \hat{P}$ as initial guess. It also exists in a Cauchy integral form instead, which permits to formulate it directly as a function of the non-purified density matrix [61, 63].

Since the density matrix $\tilde{P}$ of *ADMM1* can not be calculated directly from the original P, the *ADMMQ* and its variations *ADMMS* and *ADMMP* have been developed [64]. The main idea being to directly enforce the charge conservation by modifying the Lagrangian from Equation 2.86 to forgoe the orthogonality in lieu of an overall charge constraint N_e. This results in an ADMM where the resulting auxiliary density matrix satisfies all but the idem-potency and where the density matrix is directly related to the one from *ADMM2* via a scaling coefficient ξ

$$\begin{aligned} P_Q &= \xi \hat{P} \\ \xi &= \frac{\mathrm{tr}(PS)}{\mathrm{tr}(\hat{P}\hat{S})}. \end{aligned} \tag{2.94}$$

The unmodified *ADMMQ* has the issue of exhibiting a different effective scaling for the exact and DFT exchange energy in Equation 2.80, resulting in a larger deviation of the total energy compared to *ADMM2*, depending on the functional. The remedy in *ADMMS* is to directly counter this difference by rescaling DFT energy contribution but otherwise maintaining the scaled projected density from *ADMMQ*. *ADMMP* on the other hand assumes the functional being used to be Local Density Approximation (LDA) (or LDA being a good enough first-order approximation of the employed functional), permitting to factor out ξ^2 from both exact and DFT exchange energy. As Merlot et al. [64] already mentioned, applying the *McWeeny* transformation on top of *ADMMQ* (and by extension *ADMMS*, *ADMMP*) is indeed possible, but brings back the non-trivial dependence of the auxiliary density matrix on the full one.

It should also be noted that in any case the auxiliary orbitals obtained are not directly connected to the primary ones in a physical sense, i.e. the resulting energy eigenvalues may be different. This is particularly the case for the *purified* variant which should subsequently not be used for calculation of band gaps where correct orbital energies are relevant.

The concrete choice of the functional has been intentionally left out of the initial definition in Equation 2.80, other than the requirement having to be of the GGA type to capture the necessary interactions. A natural choice – and the default in CP2K – is be the *PBE* Exchange functional, adapted to the interaction potential used for the ERI in the respective HFX calculation itself. Other popular choices are either the full *PBE*, the *OPTX* [65, 66] or the *B88* [28] Exchange functionals.

2.6. k-point-enabled Hartree-Fock Exchange for ADMM

Finally we are going to extend the ADMM approximation to a **k**-point dependent density matrix. As the density matrix $P_{\mu\nu}$ depends on the MO coefficients $C_{\mu i}$ the extension of $\hat{P}$ to the periodic case at the same **k**-points as the density matrix for the full basis becomes

$$\hat{P}^{\mathbf{k}}_{\mu\nu} = \sum_i \hat{C}^{\mathbf{k}}_{\mu i} \hat{C}^{\mathbf{k}}_{\nu i}. \tag{2.95}$$

Where $\hat{C}^{\mathbf{k}}_{\mu i}$ is given by

$$\hat{C}_{\mathbf{k}} = \underbrace{\hat{S}^{-1}_{\mathbf{k}} Q_{\mathbf{k}}}_{=:A_{\mathbf{k}}} C_{\mathbf{k}} \tag{2.96}$$

with $\hat{S}_{\mathbf{k}}$ and $Q_{\mathbf{k}}$ being the reciprocal space pure auxiliary basis and mixed basis overlaps from in Equation 2.84 written in a similar way as the overlap matrix S of the full basis from Equation 2.66,

$$\begin{aligned} \hat{S}^{\mathbf{k}}_{nn'} &= \int \hat{\phi}_n(\mathbf{r};\mathbf{k}) \hat{\phi}_{n'}(\mathbf{r};\mathbf{k}) dr, \\ Q^{\mathbf{k}}_{nm} &= \int \hat{\phi}_n(\mathbf{r};\mathbf{k}) \phi_m(\mathbf{r};\mathbf{k}) dr, \end{aligned} \tag{2.97}$$

and then expanding $\phi_k(\mathbf{r};\mathbf{k})$ into a Bloch sum with consistent cell displacements.

Given the linearity of this transformation for each of required quantities, this means that the full ADMM fitting procedure for *ADMM2* can be done straight forward in real space (e.g. for each of the neighbours **R**), after Fourier transformed real-space density matrices have been obtained. By extension this also applies to the *ADMMQ* variations *ADMMS* and *ADMMP* mentioned in the preceeding section.

ADMM3 on the other hand may not be properly defined, since it relies on the notion of being able to define subsystems based on some physical property. While this makes sense for the primary unit cell, it becomes difficult to reason the same holds true for a density matrix $P^{\mathbf{R}}_{\nu\mu}$ which depends on an additional cell displacement vector. Whether or not the original blocking matrix should be reused for displacements beyond the Γ-point, is therefore unclear. Furthermore *ADMM3* relies on the sparsity of the density matrix, something which comes not as easily for the solid state case where **k**-points are being used in particular.

While the orthonormality-constrained *ADMM1* would be properly defined, it is much more complex and would require a minimisation procedure to determine the Lagrangian coefficients in each SCF step and for each **k**-point. Hence it stands to reason whether an implementation is computationally reasonable. In particular since with *ADMMS* a similar or better performing variant of ADMM is available, which is computationally less expensive [64].

It must be noted that care must be taken when it comes to building the list of neighbour cells to consider in the calculation: Due to the different reach of the primary and the auxiliary basis sets, the larger of both lists must be applied consistently for both sets.

Finally, we would like to point out that while density fitting approaches for periodic **k**-point-sampled HFX with atom-centred GTO have been implemented for example in Patterson [67], ADMM differs from this approach by employing the auxiliary basis not only for the expansion of the ERI, but also for the calculation of the DFT correction term in Equation 2.80.

3. Implementation

3.1. The revised MOLOPT protocol for Basis Set and Pseudopotential generation

Unless otherwise noted, we have used newly generated GTH PP for all non-all-electron calculations, and a new family of GTO basis sets for all CP2K calculations. As they have not been formally published before, we are giving some overview of the generation procedure and their general properties, here where not already covered by Section 2.3.

The MOLOPT PP are norm-conserving separable dual-space Pseudopotentials as shown in Equation 2.51 and Equation 2.53, but with neglected spin-orbit part, i.e.

$$V_{\mathrm{l}}(\mathbf{r},\mathbf{r}') = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{m=-l}^{+l} Y_{l,m}(\mathbf{r})p_i^l(r)h_{ij}^l p_j^l(r')Y_{l,m}^*(\mathbf{r}'). \tag{3.1}$$

This means they contain all scalar parts of the relativistic PP, but not more.

Their generation followed the same procedure as the previously published GTH potentials by Krack [68], which involves fitting of the local and non-local coefficients r_{loc}, C_i and $h_i j^l$ (see Equation 2.52) against atomic scalar-relativistic AE calculations. The pseudopotentials were generated up to Radon and are included in the file `POTENTIAL_UZH` of the CP2K software package. As mentioned before, the dual-space PP have a dependency on the DFT functional being used during optimisation. Therefore three sets of PP were generated, for the functionals PBE, SCAN and PBE0 as representatives for GGA, Meta-GGA and Hybrid functionals. Following the discussion about the the split into valence and core electrons, for several elements one more semi-core PP have been generated. This includes all the alkali, as well as all elements starting from Scandium onward. The reference wave functions were generated using the CP2K integrated `ATOM` code within KS-DFT, and a third-order Douglas-Kroll Hamiltonian [69, 70]. For the optimisation, Powell's method [71] was employed with a target accuracy of 0.003 eV for valence, semi-core and virtual state eigenvalues.

The procedure for generating the accompanying basis set family has been slightly altered compared to the previously published MOLOPT basis sets [46] as explained in Section 2.3. While the basis set family covers all chemical elements up to Rn as well, not all semi-core configurations have been included. The new valence basis set family (DZVP, TZVP, TZV2P) is therefore released as a revision and extension of the existing MOLOPT basis set family. As with the PP, the basis sets have been optimised with the three functionals PBE, SCAN and PBE0 as representatives for GGA, Meta-GGA and Hybrid functionals, as can be seen from their aliases. They are included in the CP2K package in the file `BASIS_MOLOPT_UZH`.

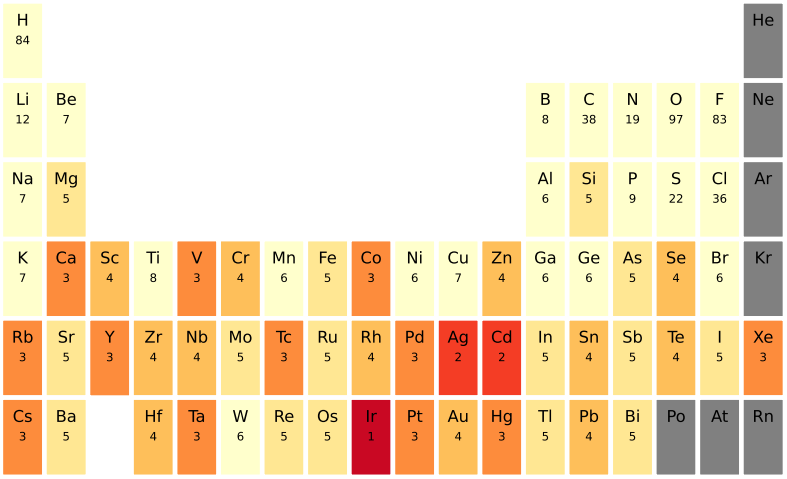


Figure 3.1.: Number of molecules from the SMDB covering an individual element.

The contraction schemes for the all-electron variants (SZV, TZVPP, QZVPP) follow the *Ahlrichs def2* [53] basis sets and are available for use with GGA functionals (obtained with PBE) and with the GAPW method. Their optimisation scheme followed the same protocol outlined below.

First, as a reference set of molecules, the database published by Weigend and Ahlrichs [53] – in the following referred to as the Small Molecules Database (SMDB) – was used. This database covers most of the periodic table Figure 3.1, except for the Lanthanides, and ensures optimising the basis set for use within a molecular environment rather than just an isolated atomistic one. For elements not (sufficiently) covered by the SMDB, in particular the noble gases, the set of structures was augmented by single atomic systems.

The reference calculations were done with sufficiently large and fully uncontracted basis sets obtained from atomistic calculations with the `ATOM` code contained in CP2K.

Secondly, for the optimisation procedure, the new (to be optimised) basis was treated as an auxiliary basis set for the uncontracted one in the same way as an ADMM auxiliary basis for the primary basis set. The optimisation procedure then involved simultaneously maximising the overlap and minimising the condition number, as well as the difference in the generated electronic densities.

While the optimisation relied on the SMDB for the selection of molecular systems to obtain reference calculations, it must be stressed that this part of the optimisation procedure did not aim for reproduction of any particular quantity other than the generated density itself.

3.2. The FLexible Automated Testing MANager

Orchestrating thousands of calculations while maintaining reproducibility and load balancing across different systems is a complex problem, as the vast number of available workflow managers illustrates [72], of which many target data science or bioinformatics in particular. Adapating such a workflow manager is possible, but as the integration with

AiiDA later in this chapter shows, requires much adaption to generate input for or parse output from electronic structure codes while making structures and parameter sets, like basis sets or pseudopotentials, traceable. In the realm of materials science, available codes are Python ASE – which gained workflow capabilities rather recently –, AiiDA and pymatgen, with an additional number of more specialised solutions. Some larger computational materials science consortia concerned with high-throughput calculations and computational studies have setup online data sharing platforms, like Materials Cloud[1], Materials Project[2], NOMAD[3] and AFLOW[4]. While some of them existed when this project started, many of them were still heavily in flux, had or still have a different focus, and were, as such, not directly usable for our purpose or required extra infrastructure for the deployment.

1: https://www.materialscloud.org/
2: https://materialsproject.org
3: https://nomad-lab.eu/
4: https://aflow.org/

In particular, we needed the following functionality covered:

- Explicit management of parameters like pseudopotentials and basis sets, including different versions of a basis set or pseudopotential in the same family as they are being tested and had to be occasionally reoptimised.
- Integration of unmanaged computes nodes (e.g. without a queueing manager like SLURM or PBS) is crucial to incorporate local resources in high-throughput calculations.
- Load balancing over multiple single nodes to employ multiple local disconnected compute nodes.
- Integration of computing centres with queueing systems to accelerate calculations for structures with a high number of basis functions.
- Asset tracking (in- and output) across all calculations for reproducibility across compute nodes and code versions.
- Templating of input configurations for CP2K.

Some of the required features, like the asset tracking (*Data Provenance*), would have been already covered by domain-specific workflow managers like AiiDA [73] when the project started. Others, like the load balancing across multiple disconnected nodes, are still not natively supported in their latest version [2] but would be achievable with additional plugins, some of which we have implemented as part of this work.

This has led us to develop the lean custom workflow management system "FLexible Automated Testing MANager" (FLATMAN), specifically tailored to testing computational chemistry parameters like basis sets and pseudopotentials. Given that many computational chemistry or materials science tools are written in Python, including the prominent Python ASE [74], the AiiDA workflow tool, and the pymatgen library [75], we followed the same direction and chose Python as the primary language. By an ongoing collaboration with the AiiDA community we ensured that components developed for FLATMAN could be reused in AiiDA or as standalone tools independently of FLATMAN to foster a growing software ecosystem around the CP2K software package. An excellent example of a similar endeavour for the Abinit code is the AbiPy [5] package, which has been instrumental for the setup of many high-throughput analyses, such as a large-scale validation study for *GW* [76].

5: https://github.com/abinit/abipy

3.2.1. Architecture

The basic architecture of FLATMAN is based on the client-server model, with a communication protocol based on Representational state transfer (REST) [77] over encrypted HTTP with the JavaScript Object Notation (JSON) file format as the data interchange format. As a direct consequence must the web-, respectively, the application-server running the FLATMAN Application Server be accessible by all clients. This is illustrated in Figure 3.3.

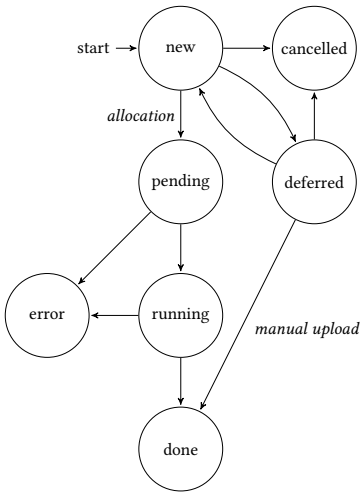


Figure 3.2.: Lifecycle of (calculation) tasks within FLATMAN. Deferring a task can prevent it from being run momentarily, but it is also used for importing already-run calculations for further processing. The transition from *new* to *pending* is used for nodes to allocate tasks to be run. The *running* state is synchronized with that of a workload manager (if used).

For task distribution over single compute nodes (as available in many lab setups), the easiest principle we found was a single task queue of jobs with each node running a worker (in FLATMAN called *runners*) picking tasks from that queue in a First In First Out (FIFO) order. Sites with workload managers like SLURM can be easily integrated by running a separate runner targeting that queuing manager and using a specific account for submission. And although not implemented, this same principle could be used to push jobs to sites from which the FLATMAN Application Server is not accessible, lifting the restriction of the FLATMAN Application Server having to be publicly available. To avoid a runner fetching jobs it can not run (e.g. due to missing code setup), jobs can be restricted to certain hosts and users, with the runners having to submit their hostname and the credentials of the FLATMAN user when requesting a new job from the task queue.

As typical in a RESTful architecture, the clients – including the FLATMAN runners – are mostly stateless with the task state tracked on the server as illustrated in Figure 3.2. By tracking the activity of runners by hostname and monitoring the task queue, interrupted runners (for example, due to node failures) can be easily determined and their work rescheduled. While this does not allow more complex automated error recovery as can be achieved with a more general solution like the work chains in AiiDA, this design has proven quite resilient to intermittent node failure while achieving a good task distribution over heterogeneous compute nodes and sites.

The stateless architecture of the clients also requires that most of the output data processing occurs on the server side. Since this data processing, which involves output parsing as well as fitting procedures, can be time-consuming in terms of HTTP requests, this work is controlled by using a separate task queue managed by Celery[6] as illustrated in Figure 3.4 and can be scaled as needed by adding more workers. The drawback of such a solution is that FLATMAN Workers (subprocesses started by Celery) must contain explicit support for each test type. In contrast to an architecture like AiiDA, where a code plugin contains the code to parse its code's output and parsing is done when fetching the output from the remote side (and that data node becoming immutable), FLATMAN allows regenerating results from the original as needed. The architecture makes use of the possibility to specify dependencies and event listeners in the Celery task queue such that a forced refresh of the parsed data automatically leads to the regeneration of dependent values, such as a Birch-Murnaghan equation of state fitting procedure to obtain an equilibrium volume and bulk modulus. This allowed us to add additional code output parsers (like the one to parse condition numbers from the CP2K output) later in the project or improve our existing parsers without

6: https://celeryproject.org

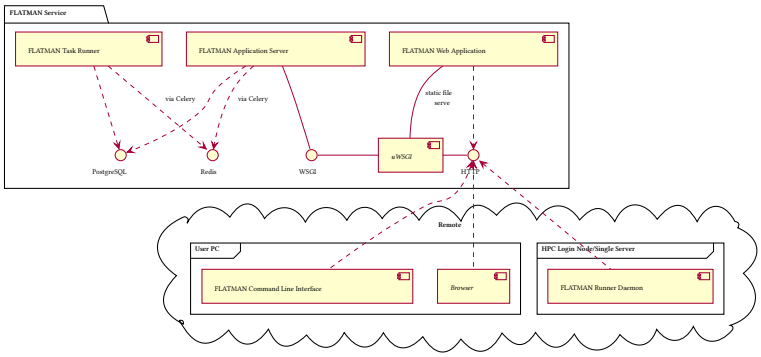


Figure 3.3.: FLATMAN software components and their dependencies. Items set in italic are off-the-shelf components. The FLATMAN web application can be served by the same server, which in turn uses the Javascript `XMLHttpRequest` method to make a request to the *FLATMAN REST API* served by the *FLATMAN Application*. While some code is shared between the *Worker* and the *Application Server*, the communication between them is strictly asynchronous via *Celery*, which in turn uses *Redis* as its backend.

having to rerun calculations or having to resort to more complex caching mechanisms.

To verify the format of the JSON data in the communication and parse additional data (query parameters, extra HTTP headers) in the requests, a serialization/deserialization schema using the *webargs* and *marshmallow*[7] packages were used, with the venerable Flask[8] web development kit to implement a WSGI-compatible application served by uWSGI[9]. Strict use of database transactions have been implemented to ensure data integrity. When a runner allocates a task (e.g. a task is set to *pending*), files need to get generated and corresponding database records stored. Only when all succeed is the whole transaction committed. This transactional functionality is provided by the underlying PostgreSQL SQL Database Management System (DBMS) and made readily accessible within Python by the Object Relational Mapper (ORM) of SQLAlchemy[10]. While intermittent failures mean that files are temporarily stored in the data directory which lack a reference in the database These can easily be cleaned out by a periodic task pruning orphaned files. On the other hand it can never occur that a database entry is created with missing files, which we consider much more important. This improves robustness significantly, while allowing us to keep the database and data tree compact.

The web frontend (FLATMAN Web Application) is written as a Single Page Application (SPA) in TypeScript with the Angular[11] web framework. It uses the same REST Application Programming Interface (API) as the FLATMAN runners and the command line client (CLI) to get the data from the application server. Such an architecture avoids a divergence in functionality between the different interfaces and forces proper asynchronous task management on the server side to avoid blocking requests. However it requires synchronisation of the front- and backend-code to be able to give the user early feedback on input validation. With the web-interface implemented mainly for browsing, this could be kept minimal.

7: `https://marshmallow.readthedocs.io/`
8: `https://flask.palletsprojects.com/`
9: `https://uwsgi-docs.readthedocs.io`
10: `https://www.sqlalchemy.org/`
11: `https://angular.io/`

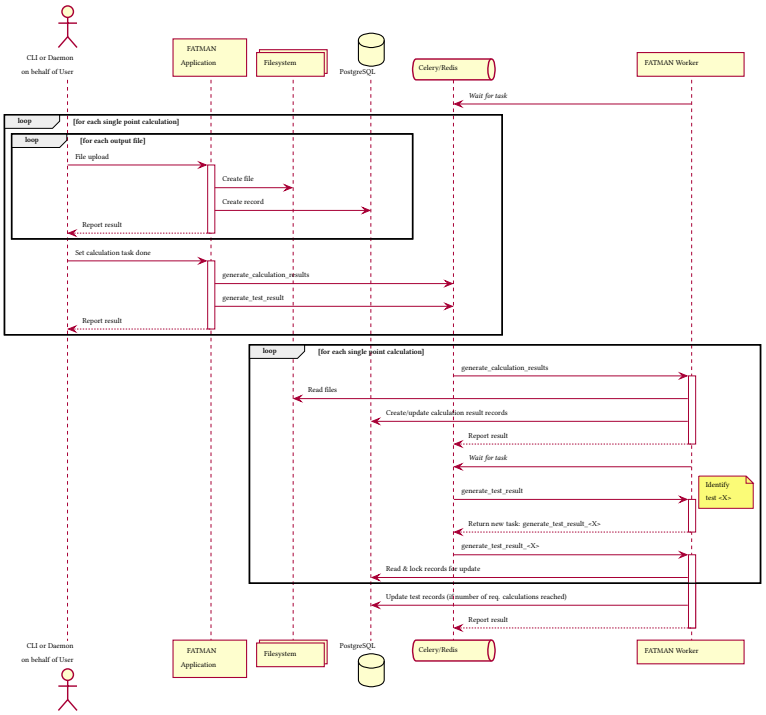


Figure 3.4.: Sequence diagram of the FLATMAN calculation output upload on the user's behalf by either the command line interface (CLI) or the *fdaemon* service to generate a test result (Δ-test, GW100). The calls to *Celery/Redis* from the application are done asynchronously and are linked: Only if the function execution of `generate_calculation_results` in the context of the *FLATMAN Worker* succeeds, will *Celery* schedule the execution of `generate_test_result`. Only the initial call and the link are specified in the *FLATMAN Application* context. Also, the execution of the functions `generate_calculation_results`, `generate_test_result` and `generate_test_result_<X>` do not have to be run by the same worker but can be executed by any of the workers in the worker pool. By locking the database for an update, the execution of the actual test result generation (consisting of a fitting procedure and database record generation) is serialized.

3.3. Comparison of automation approaches

In this work, we have implemented three different methods of calculation automatisation: With the bespoke workflow management solution FLATMAN, the more general but materials science specific AiiDA and a manual approach using CP2K preprocessor directives. The Δ-test values for the previously published MOLOPT basis sets [1, 46] as well as preliminary results for the SMDB have been obtained with FLATMAN. At the same time, AiiDA has been used for running the Δ-test with the revised MOLOPT basis sets, and the CP2K preprocessor directives approach was used for the molecular database calculations.

With hundreds of available workflow projects, as we uncovered in a recent survey [72], the development of any new general-purpose workflow management system must be met with scrutiny. As already observed by da Silva et al. [78], workflow management tools are often spun out of scripting frameworks already in use in a workgroup, rather than developed from scratch. Gradually adding additional features which may cover features otherwise found in full-fledged workflow management tools is from that perspective then only part of the progress to satisfy immediate needs. In particular, since often thought-after features such as running commands or copying data are easily implemented in any scripting language, if only a very limited number of methods must be supported (e.g. only running calculations directly via `mpirun`). In contrast, many general purpose workflow managers for High-performance computing (HPC) have a rather steep learning curve due to the higher abstraction, or non-negligible setup cost, and are often coupled with nomenclature likely foreign to a domain scientist [79]. General-purpose workflow managers often lack support for tools and applications used in a scientific domain, increasing the initial cost of adoption. After that initial hurdle though, they quickly start to outpace bespoke frameworks

since they often provide rich ecosystems of plugins (such as the AiiDA registry [12]) and support for various transport mechanisms and workload managers (SLURM, PBS, etc.). But if bespoke frameworks are developed in a way such that their components are reusable components, as in the design of FLATMAN, which now shares code with the AiiDA CP2K plugin and other tools, transitioning between the two paradigms as-needed becomes easier.

12: https://aiidateam.github.io/aiida-registry/

While this work shares many characteristics with the high-throughput efforts mentioned above, its immediate goals are different. While different model systems are calculated, we are mainly interested in the evaluation of the codes (CP2K, SIRIUS) and their respectively calculation parameters (basis sets, pseudopotentials, functional). More explicitly stated, we are not varying the structures under investigation for the same parameter set, but we are varying the parameters for the same structures. This directly implies that the database (the molecular or solid-state description) remains largely unchanged over the course of the project. Instead, it must be expected that manual adjustments to parameters are required more frequently in the initial stage of such a project. This covers both parameters applied to all calculations in a test, like the calculation of initial magnetisation parameters or reference cells, as well as system-specific corrections. In a suitable workflow system, the possibility of quickly iterating between modifications of the templating engine (taking the abstract input parameters, a specification of the physical system, and a description of the target machine on which to run the workload) is therefore crucial. The templating engine itself must be flexible enough to allow the specification of custom rules to make per-system parameters possible in the first place. An a-posteriori connection between the abstract input, the generated input for the code, and the output of both the electronic structure code and the fitting procedure must be establishable for publication. Another important feature is the efficient usage of available resources, which can vary over the course of such a project from local compute nodes to running on clusters and mixed environments. Finally, it must be assumed that codes have many failing modes, requiring proper output parsing to detect them properly. Any workflow manager relying on caches to determine whether to rerun calculations must be able to determine whether a calculation has finished correctly.

CP2K preprocessor directives Relying purely on the CP2K preprocessor and a small number of scripts (shell scripts, GNU Makefiles) can avoid any friction introduced by intermediate layers like templating layers. It allows for quick iteration between running codes and adjusting runtime parameters. It is suitable for calculations running on mostly one machine and where calculations can be assumed to be robust enough to succeed. The drawback of this approach is that one must manually track reruns in case of changed parameters (including basis sets) and overall progress itself. Distribution of the work on multiple compute nodes must be done manually unless a workflow manager specialised only in task distribution is introduced at this point. This incurs the least upfront cost while shifting most of the required time to manual a-posteriori evaluation of the output.

Bespoke workflow manager FLATMAN This approach had the advantage of combining much of the simplicity of the simple scripting approach above with a high degree of automation, enabled by the design of FLATMAN. It allows for both manual single runs making it possible to inspect automatically generated code input and trial runs, and fully automated load-balanced task distribution across single node instances and workload manager (SLURM) driven supercomputing centres. The task queue model permits automated parameter fitting as soon as enough data is generated and live status overview to detect sudden systematic errors early. It still has limited retry capabilities in case of intermittent failures, yet the task pull architecture limits the effect of such failures. Automated reruns due to changed parameters must still be done manually and thus require careful tracking of changes. The drawback of this approach was clearly the time required for development of basic facilities and extensions, even if a large part of it could be salvaged by the component based architecture.

AiiDA with CP2K and custom workflow plugins The AiiDA workflow manager provides facilities for automated task submission to various types of computing resources. It features strong provenance tracking allowing for graph-assisted tracking of input parameters to output data and automated export and publishing capabilities to the Materials Cloud project. While this resolves various tedious tasks around running calculations, additional plugins are needed to handle other aspects. An extended version of the aiida-cp2k plugin and a separate data plugin were developed to manage GTO basis sets and pseudopotentials. Relying solely on the built-in file data plugin was not possible as this would track basis sets and pseudopotentials only via implicit code parameters and file content, requiring either manual tracking of calculations to rerun in the case of updated basis sets or having to avoid rerunning unaffected calculations due to the changed file contents. This is due to how the caching of calculations within AiiDA works: based on changes in the input AiiDA determines whether a calculation step needs to be rerun (and which generates new output nodes) or whether the existing output nodes can be reused. If all calculations were to use the same data file, an update or extension to this file would trigger AiiDA to rerun all calculations using this data file if the workflow were to be called again. Another restriction imposed by the architecture of AiiDA is using specific resources for a single workflow rather than distributing tasks onto available resources at runtime. E.g. a code node is always directly tied to a computing resource. Hence the computing resource being used is determined for all sub-workflows and steps at the beginning of a workflow, together with the code. Load-balancing over different computing resources, therefore, must be done externally to the workflow. The immutability of the code or compute nodes requires a manual restart of whole workflows, in case of changes to the compute or code configuration. This invalidates any already run single-point energy calculation and requires them to be rerun in such an event. Therefore, the implemented workflow in the forked aiida-cp2k plugin still requires explicit specification of the chemical element to run the Δ-test workflow for, with the iteration over elements having to be scripted separately. While this separate scripting was also used as an opportunity for load-balancing on the level of model systems, it partially defies the point of workflow automation. A workflow management sys-

tem should be efficiently usable beyond final benchmark runs. A final point in which the caching failed was the parsing of code output. The experimental nature of our protocol again influenced this and is unlikely to occur in a setting where the focus is on the simulated systems instead. In our case, we encountered many cases where the code (CP2K and SIRIUS) could not converge but also did not return proper error codes due to intermittent failures (e.g. compute node errors, scientific libraries). Such cases AiiDA (together with the present version of the aiida-cp2k plugin) considered successful runs, requiring time-consuming post-processing and analysis, followed by manually invalidating (e.g. deleting) affected calculations.

This comparison hopefully also illustrates how quickly the complexity of running seemingly simple tasks can increase. If strong data provenance is required or complex workflows must be orchestrated, a fully-fledged workflow manager like AiiDA has a clear advantage. Its current limitations considering task distribution make it difficult to be used in mixed computing environments and projects where the object under investigation is the code rather than the simulated system. Automation via code-provided tools (like the CP2K preprocessor definitions to include shared definitions) has the advantage of being easily accessible to any domain specialist, which is the intended audience in a project like ours. Combining this with a workflow manager to automate the process of task distribution (including up- and download to computing centres) will likely solve most issues. And in our case, the modular development of the FLATMAN components has proven invaluable.

3.4. CP2K Input Tools Python Package

While running CP2K calculations with the two workflow managers FLATMAN introduced in Section 3.2 and AiiDA [73], a common challenge emerged: validating, converting and importing CP2K input configuration. When running high-throughput calculations, a scientist will eventually use the workflow tools and language – in the example of AiiDA, the Python language and its dictionary syntax or YAML in the protocols, for FLATMAN, a JSON representation – to write the CP2K input as part of their larger workflow. But more often than not, scientists will already have some CP2K configuration they would like to run via a workflow manager like AiiDA or FLATMAN or use as a template for a new project with a workflow management tool. Furthermore, CP2K writes its restart files in the same input format, making it possible to parse output data from it, given a CP2K input format parser. And last, tools which integrate CP2K as part of their workflow (e.g. as a force calculator) must be able to parse a CP2K input file to obtain required data like the cell information and then generate modified CP2K input files. An example of this class of applications is Phonopy [80], where the user specifies an input file of their code, and Phonopy will generate derived inputs for single site displacements to obtain the forces. For codes where the geometries are easily parseable because they are either in a specific place of the input configuration with a limited syntax variability (Quantum-Espresso) or even kept in a separate file (PODCAR) this is an easy enough task, but CP2K offers a much richer syntax when it comes to its input configuration which is

readily exploited by its users and thus requires greater care when parsing. A pure-Python package that can parse the CP2K input configuration and different data files like the basis set and pseudopotential formats is, therefore, crucial for the continued support of CP2K in the expanding quantum chemistry and material science software ecosystem.

The overall design of the input parsing framework is shown in Figure 3.5. A layered approach has been taken, using Python iterators extensively to separate the different parsing tasks and to avoid extra copies of data structures. CP2K also features a preprocessor which allows for including other files, conditional blocks and variable assignments. The `CP2KInputTokenizer` class responsible for parsing a single configuration key is a state machine, making the whole architecture a one-pass parser which uses a stack to track any preprocessor variables and state, building a kind of syntax tree of the input configuration. Implementations of the `CP2KInputParser` will then use this tree to provide different representations of the CP2K input configuration. We are constantly maintaining the source context to give the user proper feedback on syntax errors, as seen in the screenshot in Figure 3.6, including the specific line and column. Together with a base language server protocol (LSP), this made it possible to easily implement a CP2K language server, permitting syntax checking of CP2K input directly in any editor supporting the LSP.

To validate a CP2K input configuration against a specific version of CP2K, the parsing framework uses the CP2K input configuration XML, which is a description of the available options and the input structure that CP2K can generate with the `--xml`, initially intended for the CP2K manual generation. In contrast to most other quantum chemistry code configurations, the CP2K configuration consists of not only sections with keywords but also nested sections (a tree-like structure). This generally maps very well to maps (*dictionaries* in Python) and lists available in most programming languages, together with their primitive datatypes. Complications for a simple one-to-one mapping without further descriptive elements (e.g. explicit *section* or *keyword* tags) arise in the case of having in the same section the identical word for both a section and a keyword (which within CP2K is resolved by the `&` marker to indicate the start of a section). Further inconsistencies are introduced by repeatable (parametrised) sections like the usual `&KIND` section, which calls either for a list of maps or a map of maps since the parameters must still be unique. The latter requires a special being introduced to capture the section parameter. This leads to the following rules for a (canonical) representation of the CP2K input configuration in Javascript Object Notation (JSON) in terms of Python objects as illustrated in Figure 3.7[13]:

- repeatable sections are mapped to dictionaries
- keywords or subsections are key/value entries in sections
- all repeatable elements (sections and keywords) are mapped to lists of their respective mapped datatype
- section parameters are mapped to a special key named `_`
- default section keywords are mapped to a special key name `*`
- sections in JSON must be prefixed to avoid the double definition of a key in case of the same name for a section and a keyword (like the `POTENTIAL` in `KIND`), to avoid quotation marks, instead of CP2K's `&` we are using the `+` character

13: The following rules and the referenced code examples have also been published as part of the *cp2k-input-tools* package.

- keyword values are mapped based on their datatypes: a list of values is always mapped to a list of their respective datatypes

Based on the above restrictions, the format specification can be relaxed under the conditions followed below, leading to a simplified format as illustrated with JSON and YAML[14] (YAML Ain't Markup Language™) in Figure 3.8. A derivation of this simplified object notation (with small differences in how multi-valued values for keywords are specified) is used in the AiiDA CP2K Plugin. It can also be emitted by the *cp2k-input-tools* library and command line interfaces.

14: https://yaml.org/

- a section only has to be prefixed with a + if a keyword with the same name is present at the same time in the same section. Given the XML input specification, we can resolve any ambiguities while parsing
- if a repeated keyword or section contains only one entry, the list can be omitted. In case of ambiguity, priority is given to multiple values per keyword rather than keyword repetition
- sections with default parameters can be formulated as dictionaries as long as the default parameter values are unique and do not match section keywords or subsection names

Using a JSON or YAML-derived format rather than the original CP2K input format in orchestration tools like FLATMAN or AiiDA has other advantages than a direct mapping to language constructs. They can be syntactically validated before the logical validation, they can be stored directly in JSON database columns as implemented in PostgreSQL or SQLite (as opposed to opaque text objects on disk or in a textual column), and last, there are already tools available to manipulate structured data like JSON (e.g. *jq*) which facilitates the integration of CP2K itself in workflow orchestration tools.

The functionality described is implemented as a library, together with the command line tools (converters, language server, linters, prettifiers) on Github [15] under the MIT license. Release versions are automatically deployed on Pypi [16], after successful testing with code coverage above 90%. This library is used in Phonopy [80] to read and generate CP2K input and has been integrated into the *aiida-gaussian-datatypes* plugin to parse CP2K basis set and pseudopotential files.

15: https://github.com/cp2k/cp2k-input-tools

16: https://pypi.org/project/cp2k-input-tools/

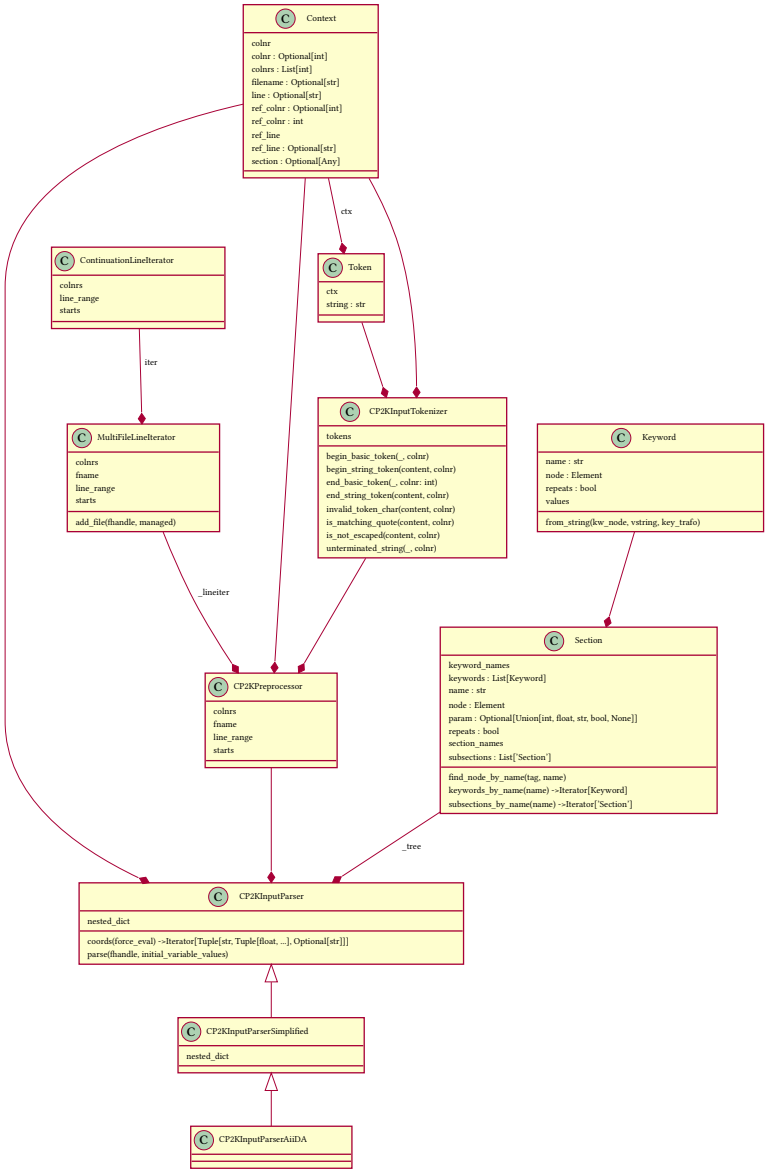


Figure 3.5.: Class diagram of the main CP2K input tools classes. Different core parts of the parser have been split into separate classes to use *separation of concerns*, using *Generators* to avoid full storage of parsed segments. The parsed options are stored in a tree-like structure with the `Section` and `Keyword` dataclasses. The `Context` class stores original source file, line and character numbers across the complete input, allowing exact determination and feedback of parsing errors, including tracking of references of preprocessor variables and respective sections of the input schema definition.

```
$ cp2klint mysystem.inp
Syntax error: unterminated string detected, in mysystem.inp:
line   14: BASIS_SET_FILE_NAME  "BASIS_MOLOPT_UCL
                                ~~~~~~~~~~~~~~~~~^
```

Figure 3.6.: Example of a `cp2klint` output returning exact line number and column indicator.

```
&GLOBAL
   PRINT_LEVEL MEDIUM
   PROJECT test
   RUN_TYPE ENERGY
&END GLOBAL
&FORCE_EVAL
   METHOD Quickstep
   &DFT
      BASIS_SET_FILE_NAME ./BASIS_SETS
      POTENTIAL_FILE_NAME ./POTENTIALS
      &XC
         &XC_FUNCTIONAL PBE
         &END XC_FUNCTIONAL
      &END XC
   &END DFT
   &SUBSYS
      &CELL
         A [angstrom] 4.07 0.0  0.0
         B [angstrom] 2.03 3.52 0.0
         C [angstrom] 2.03 1.17 3.32
         PERIODIC XYZ
      &END CELL
      &KIND Ge
         ELEMENT Ge
         POTENTIAL ALL-q32
         BASIS_SET ORB pob-TZVP
      &END KIND
      &TOPOLOGY
         COORD_FILE ./struct.xyz
         COORD_FILE_FORMAT XYZ
      &END TOPOLOGY
   &END SUBSYS
&END FORCE_EVAL
```

```
{
  "+global": {
    "print_level": "medium",
    "project_name": "test",
    "run_type": "energy"
  },
  "+force_eval": [
    {
      "method": "quickstep",
      "+DFT": {
        "basis_set_file_name": [
          "./BASIS_SETS"
        ],
        "potential_file_name":
↪  "./POTENTIALS"
      },
      "+XC": {
        "+xc_functional": {
          "_": "PBE"
        }
      },
      "+subsys": {
        "cell": {
          "A": [ 4.07, 0, 0 ],
          "B": [ 2.03, 3.52, 0 ],
          "C": [ 2.03, 1.17, 3.32],
          "periodic": "XYZ"
        },
        "+kind": [
          {
            "_": "Ge",
            "element": "Ge",
            "potential": "ALL-q32",
            "basis_set": [
              [ "ORB", "pob-TZVP" ]
            ]
          }
          ],
        "+topology": {
          "coord_file_name":
↪  "./struct.xyz",
          "coord_file_format": "XYZ"
        }
      }
    }
  ]
}
```

Figure 3.7.: Comparison of a CP2K input and its canonical JSON representation.

```
{
  "global": {
    "print_level": "medium",
    "project_name": "test",
    "run_type": "energy"
  },
  "force_eval": {
    "method": "quickstep",
    "DFT": {
      "basis_set_file_name":
↪   "./BASIS_SETS",
      "potential_file_name":
↪   "./POTENTIALS"
    },
    "xc": {
      "xc_functional": {
        "_": "PBE"
      }
    },
    "subsys": {
      "cell": {
        "A": [ 4.07, 0.0, 0.0 ],
        "B": [ 2.03, 3.52, 0.0],
        "C": [ 2.03, 1.17, 3.32],
        "periodic": "XYZ"
      },
      "kind": {
        "_": "Ge",
        "element": "Ge",
        "potential": "ALL-q32",
        "basis_set": [ "ORB", "pob-TZVP"
↪   ]
      },
      "topology": {
        "coord_file_name":
↪   "./struct.xyz",
        "coord_file_format": "XYZ"
      }
    }
  }
}
```

```
global:
  print_level: medium
  project_name: test
  run_type: energy
force_eval:
  DFT:
    basis_set_file_name: ./BASIS_SETS
    potential_file_name: ./POTENTIALS
  XC:
    xc_functional:
      _: PBE
  method: quickstep
  subsys:
    cell:
      A: [ 4.07, 0.0, 0.0]
      B: [ 2.03, 3.52, 0.0]
      C: [ 2.03, 1.17, 3.32]
      periodic: XYZ
    kind:
      Ge:
        basis_set: [ORB, pob-TZVP]
        element: Ge
        potential: ALL-q32
    topology:
      coord_file_format: XYZ
      coord_file_name: ./struct.xyz
```

Figure 3.8.: Example of the same CP2K input as in Figure 3.7, but with simplification rules applied. The YAML format (used for the example on the right) as a superset of JSON can contain the same information, but due to its rules for structured indentation and quoting, it becomes even more condensed as the CP2K original in Figure 3.7 but retains the same legibility.

3.5. CP2K and Gaussian Output Tools Python Packages

In high-throughput calculations, being able to parse the output of codes efficiently, correctly and in a fault-tolerant way is vital as it directly affects further calculations and hence compute resources. While some codes like BigDFT [81] have implemented a human and machine-readable output by adopting YAML as their primary format, this becomes quickly unfeasible with a highly complex code as CP2K and impossible for a closed source code like Gaussian [82].

With the implementation of the *cp2k-output-tools*[17] package, we provide a Python library and command line tools which implement parsing the most common blocks of CP2K output in an efficient way using regular expressions (regex). The package is organised as a set of parser functions acting on a string and which can be reused individually by any Python project as needed. The parsed data is stored in Python dataclasses as they provide a clear advantage over simple dictionaries in the form of semantic completion within text editors and, by integrating typing annotations, allow for additional correctness checks. Based on the integration of the Pint[18] library, the data fields carry physical units to incorporate unit documentation in the code, making unit conversion easier and avoiding unit conversion errors in the process. In addition to the parsing blocks, the library also contains a more structured parser which returns a tree structure containing the various data objects. An example of such a tree structure in textual representation is shown in Figure 3.9. This tree structure corresponds to the nested method calls of CP2K, for example, employed in a geometry optimisation where each geometry optimisation step consists of an SCF loop, which may be split into an inner and outer loop. Monitoring quantities and outputs across levels allows for quick analysis in high-throughput calculations, even in the presence of errors.

17: https://github.com/cp2k/cp2k-output-tools

18: https://pint.readthedocs.io

Additional tools – for example, Projected Density of States (PDOS) interpolation or bandstructure conversions – have also been implemented. Still, among those tools, the technique behind the *trajectory restart cleaner* is worth highlighting. When running MD simulations, CP2K continuously writes trajectory files in XYZ format, which can be used to evaluate observables statistically. Upon restarts of the calculation, the restoration point may be several frames before the last entry in the trajectory, leading to multiple duplicated frames and statistically noticeable noise. As

```
CP2K:
    started at: 2021-02-23 22:57:53.427000
    ended at: 2021-02-23 22:58:11.587000
    CellInfoType.top cell volume: 39.167875 angstrom ** 3
    CellInfoType.default cell volume: 39.167875 angstrom ** 3
    CellInfoType.reference cell volume: 39.167875 angstrom ** 3
    SCF:
        converged: True
        Total FORCE_EVAL energy: -7.944253454478329 hartree
        Mulliken Population Analysis:
            (present)
        [warning]: Print MO Cubes not implemented for k-point calculations!!
        [warning]: Localization and MO related output not implemented for k-point
         ↪ calculations!
        Inner SCF:
            converged: True
            number of steps: 8
```

Figure 3.9.: Shortened example of a parsed CP2K output, showing proper Inner and Outer SCF loop detection with correct warning assignment.

CP2K includes a step number in the XYZ comment line, filtering the duplicated frames is possible. The challenge is the trajectories' size, which can easily grow to a couple of gigabytes. Handling huge text files efficiently is still an ongoing research topic [83]. However, the challenges we face here are limited computer memory and avoiding costly conversion operations and string handling by Python due to its extra conversion steps, often a performance bottleneck. To avoid the costly Python string operations, we are again resorting to regular expressions to find frames in the output, implemented behind Python iterator objects. The regex library is implemented in C and acts directly on text buffers, with string conversion to Python delayed until needed. To avoid loading the entire content of the trajectory into memory as a Python string and work around limited memory, we are directly using the operating systems (OS) memory mapping capabilities (*mmap*) in which the file gets mapped to an OS-managed buffer. As the regex engine sequentially accesses bytes in the array, the operating system automatically loads only required parts of the file in page units into memory, evicting previous pages. Buffering the loaded frames before writing until a restart point is detected (or a maximum number of frames has been buffered in memory) prevents high-frequency seeking, which for network filesystems like NFS works around the issue of higher latency for quick file system access. Combining those two techniques allowed us to clean gigabyte-sized trajectories several times faster than a previously implemented naive parsing approach based on line-based file iteration.

The *cp2k-output-tools* library and its accompanying command line utilities are published under the MIT license on Github, including a Continuous Integration setup with high coverage and released as an installable Python package on Pypi. It has been integrated into the *aiida-cp2k* plugin to parse additional data from CP2K output files in AiiDA workflows.

Since we also needed to parse Gaussian output as part of this thesis (see Chapter 4), a package following the same design principles as the *cp2k-output-tools* was implemented, the *gaussian-output-tools*[19].

19: https://github.com/dev-zero/gaussian-output-tools

3.6. Software Engineering in Computational Science

Software engineering techniques have not only played a role in the development of the FLATMAN workflow manager and its component-based architecture, which allowed the reuse of components across different projects. With long-running open-source projects like CP2K with many contributors, organisation and code quality become increasingly important to enable continued development. To that end the author contributed significantly to three aspects: the migration of the CP2K from the Subversion to the Git version control system, making the embedded Distributed Block Compressed Sparse Row (DBCSR) Matrix library a standalone project, and the introduction of the CMake build system first for said DBCSR library and then CP2K itself.

On Version Control Systems While collaboration and development do not necessarily rely on a Version Control System (VCS) to manage the

project's source code or parameter data, using a VCS has significant advantages. Tracking changes to the source code alongside meaningful comments not only serves as a documentation of labour but also as an implicit form of documentation since most VCS provide an annotated source code view in which source code lines are shown next to the comment entered for that code change (e.g. for Git: `git blame`). This, in turn, allows tracing bugs back across releases and identifying copyright holders for individual files. The former is important for the reproducibility of results. At the same time, the latter becomes important in the case of relicensing, sometimes required to ensure collaboration with industry, which relies on more permissive licensing than potentially chosen at the beginning of a project (e.g. BSD-style license instead of GPL). The continuous recording of authorship also permits easy tracking down people for collaborative work within a specific area. Something which becomes increasingly difficult for a large project like CP2K.

For new projects, the choice of VCS is often not done consciously but decided based on what the original authors are familiar with, which VCS is most popular or for which free online resources for sharing are available. For CP2K, the collaborative platform chosen at the time was Sourceforge, which initially offered the hosting of CVS and later Subversion code repositories. Both VCS have a clear client-server model in which the server hosts the master version, and each commit done by a contributor is uploaded to the server (on the master version or a branch). Any conflicts arising due to the master version having advanced while the changes in a commit still rely on a previous state of the source code have to be resolved at the time of commit by the user. While this client-server paradigm makes for a linear development model with a low entry barrier, it has shortcomings when it comes to collaboration and ensuring coding standards. The dedicated master service requires everyone who wants to commit on either the master or a separate branch to have an account with the service, often leading to patches being sent by email to the maintainers who then commit the changes with their credentials. Besides the time required by the maintainers to adapt and incorporate such patches, it also meant the occasional loss of authorship since credentials for committing and author of the commit are coincident in such systems. A second challenge arises from the fact that branching in either CVS or Subversion – while possible – is rather cumbersome and again requires the person who wishes to create such a branch certain access rights to the repository. It furthermore makes running tests such as code style compliance checks before a commit unfeasible, often requiring a-posteriori work.

Distributed Version Control Systems (DVCS) like Git, together with publicly hosted services like Github or Gitlab, solve the issues mentioned above. The distributed aspect of a DVCS means that every participant will automatically get a fully separate copy of a project's complete versioned source code. On this copy, a contributor can then create branches and submit commits. The implication of such a model is that conflicts arising when merging changes from one repository to another (in whatever fashion) need to be resolved at the time of the merge rather than at the time of commit, potentially requiring multiple conflict resolutions, one for each commit. This further implies that the immutability of commits from non-distributed VCS is relaxed so that commits can be rear-

ranged and altered. Each commit then also carries both an author and a committer, solving the problem of losing authorship. Services like Github not only host repositories, but they also provide workflows to send and incorporate commits from other copies (called *forks*) in what the services call *Pull* or *Merge* Requests. Furthermore, they provide tasks to be run on such requests, allowing for a priori checks to be run, solving the other mentioned challenge of non-distributed VCS. Altogether this allows for a much more dynamic development in which different parties can work largely independently while making their changes publicly available once they see fit. Being able to commit independently results in commits being more atomic rather than lumping multiple unrelated changes together into one, helping with code review and therefore code quality.

The migration of CP2K from Subversion to Git involved not only a simple import (for which tools are available) but to preserve copyright holders and history. Care had to be taken to mirror each commit. This required several passes in which usernames were mapped, directory reorganizations had to be correctly tracked, and tags and branches to be recreated. With the service migration, existing workflows (both automated and manual) had to be adjusted, and users were guided towards using the new system. It must also be noted that even though Git is centred around the concept of *merging* in which changes can be pulled into a copy with many conflicts being resolved automatically, this has the negative effect of rendering the commit history of a project quickly unreadable as it becomes unclear where a specific commit originated from. To avoid this cluttering of the history, it was decided for the main repository only to permit a linear history, requiring authors to *rebase* their work onto the latest master before sending a *Pull Request*. Incidentally, this also prevents the issue common in DVCS in which tests on *Pull Requests* succeed, but once merged into the *main* branch, they start to fail despite not having generated any conflicts during the said merge. In the *rebase* development workflow, the *Pull Request* becomes the new state of the *main* branch once merged and thus prevents this corner case as well as shifts the integration work to the submitter of the *Pull Request*. Unfortunately, this *rebasing* workflow was, and occasionally still is, the source of some confusion within the developer community as it often requires more manual adjustment of commits than the more prominent *merge* strategy.

On Build Systems The scientific software ecosystem has seen drastic changes within the last twenty years. While most HPC software like CP2K was previously buildable with just BLAS/LAPACK, MPI and ScaLAPACK, modularization and consolidation have led to large dependency trees. While this has allowed the field to advance more quickly by allowing to easily share common code like implementations of integration routines, correlation-exchange functionals and others, building software has become much more difficult. This can easily be illustrated by a dependency tree in Figure 3.10, generated with the Spack package manager for the CP2K package, which the author maintains. Such a dependency tree depends not only on the available packages on a system (compiler, scientific libraries) but also on the project's configuration. Advancing language standards while providing the developers with more advanced

constructs and simplifying development also have the potential of breaking unexpectedly with untested compilers.

A build system is therefore not only used to invoke the compiler and linker to build an executable reproducibly but also to configure the software (e.g. enable or disable code passages) based on features requested by the user via flags, check the consistency of the requested features, and to check whether the environment (compiler, provided libraries) satisfy the requirements. CP2K has been using a custom build system based on GNU Make and Python, focusing primarily on the building. The user then writes a configuration file specifying the preprocessor directives and compiler and linker flags. The separation of the DBCSR library into a separate project opened up the possibility of introducing the CMake build system for DBCSR, which is a slightly smaller codebase than CP2K itself, before adapting it for CP2K itself.

The CMake-based build system for DBCSR has the following additional features compared to the custom CP2K build system. It checks at configuration time for a series of required Fortran language constructs. This helps the user to identify potentially incompatible compilers before starting the compilation with an understandable error message. Required libraries are automatically discovered based on multiple mechanisms (pkg-config, CMake, common library paths), accommodating most setups, including the Cray Programming Environment. Preprocessing the source code with the Fortran preprocessor `Fypp` has been integrated. The use of accelerator software development kits (CUDA, HIP/ROCm, OpenCL) has been integrated, and their use can be controlled via a single build configuration, which triggers respective discovery mechanisms. Likewise, for different small matrix multiplication libraries. The discovered BLAS library is checked for OpenMP compatibility, avoiding hard-to-debug numerical errors due to concurrency issues when using non-thread-safe libraries. The build system automatically builds complete API documentation with the FORD documentation tool, which gets automatically deployed via a CI/CD system on GitHub. This makes it possible that most users can get a working configuration of DBCSR with a single command, avoiding common mistakes and potentially devastating inconsistencies in the library configuration.

Based on the above principles of the CMake build system adoption for the DBCSR library, a CMake-based build system for the whole CP2K project has been recently added.

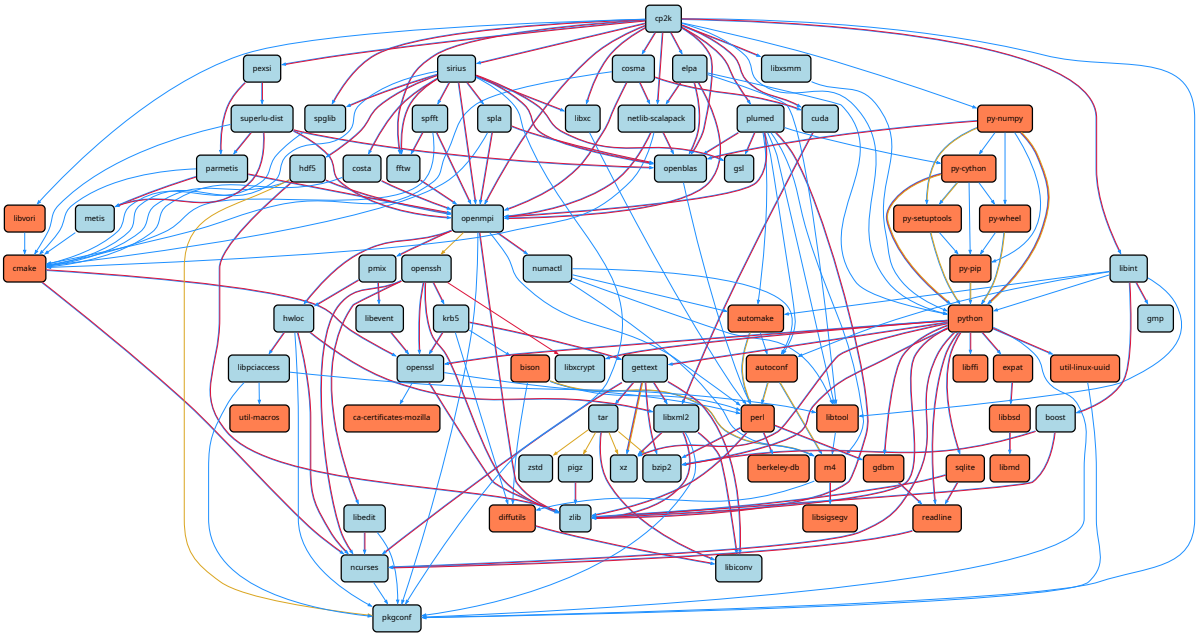


Figure 3.10.: Dependency tree for CP2K 2022.1 as generated with the Spack package manager for the *spec* `cp2k +cosma +cuda +elpa +libint +libvori +libxc +mpi +openmp +pexsi +plumed +sirius +spglib cuda_arch=70 smm=libxsmm`. The edges of the graph are colour coded depending on the type of dependency (build, link, run, test). Such a graph depends on available compilers and scientific libraries, in this case on OpenBLAS, Netlib ScaLAPACK, and OpenMPI with the GNU Compilers.

MOLOPT Basis Sets and Pseudopotentials Benchmarks

4.

4.1. Motivation

While localised basis sets like the Gaussian basis sets have been successfully used for decades in quantum chemistry and materials science, they do have some limitations. Unlike with plane-waves, it is not possible to increase the precision at will by simply increasing a cutoff, but larger basis sets must be explicitly generated. This is done by optimising contraction coefficients and exponents in the Gaussian basis sets such that calculations adequately reproduce specific properties (often the energy), obtained for example from databases of molecular energies or other quantities. Since optimising a larger number of exponents is still difficult, multiple approaches exist to directly construct them via recursion formulas [51, 52, 84, 85]. By now, a vast number of of basis sets have been published [86, 87]; the *Basis Set Exchange* [88] lists almost 700 families of basis sets. Although many of them are attuned to specific systems or methods, and often cover only a subset of the elements on the periodic table. Yet, new basis sets are still developed [89, 90].

Additionally, they are closely tied to the underlying method of handling the core electrons and the charge distribution near the nucleus, segmenting them into either *all-electron* or *pseudopotential* (resp. *valence*) basis sets. Finally, the specific method (Density Functional Theory or Hartree-Fock and post-HF) to approximate the wave function is inherently reflected in the generated coefficients. This becomes clearer when looking at *correlation-consistent* basis set families attuned to post-Hartree-Fock calculations, but also for DFT basis sets this may add a dependency on the functional being used. Failure to use the correct basis sets for the respective methods [40] may then in fact lead to unreasonably large errors [41]. Finally, optimising basis sets with respect to energy alone does not guarantee convergence for other properties as they may not be correlated [91], thus careful validation is required.

A basis set *family* describes a collection of basis sets for a number of chemical elements. Often with multiple sets per element, where the different sets contain an increasing number of exponents ζ for the same momenta and/or cover increasingly higher momenta. A *family* basis set on the other hand describes a generally contracted basis set in which the exponents are shared between the contractions.

In this work we will benchmark a revised edition of the *MOLOPT* family of basis sets and pseudopotentials [46], particularly quantifying their improvements for increasing numbers of exponents and added polarisation terms, covering three rungs of Jacob's ladder (see Chapter 2). This revised MOLOPT set now also features an AE basis set, generated in the tradition of the *Karlsruhe def2* basis sets[53]. A special focus will also be given towards assessing how well such a benchmark can be automated.

These MOLOPT basis sets and pseudopotentials are extensively used in, and shipped with the CP2K software package [1].

4.2. Methodology

To assess the quality of the MOLOPT basis sets and pseudopotentials we are looking at both a quantum chemical and a solid-state setting. There exist many different data bases of molecules along with referenced numbers for various use cases, quantities, functionals, or methods (of which [92–98] cover a mere fraction), sometimes collected into even larger sets [99]. For this work we used as database for the quantum chemical case the already mentioned SMDB by Weigend and Ahlrichs [53], which consists of more than 300 small molecules. The reason is that it has a good coverage of most of the periodic table (excluding the Lanthanides, see also Figure 3.1 for coverage of elements by structures) and has been used in a similar function before.

This set of small molecules was also instrumental in the generation of this basis set family, but given the optimisation procedure outlined in Section 3.1, there is no direct correlation between the calculated properties of those molecules and the basis set itself and it can therefore still serve as a reference database for performance comparison.

Table 4.1.: Overview of available calculation sets for the Small Molecules Database

	G16	CP2K GPW	CP2K GAPW
PBE	•	•	•
TPSS	•	•	
PBE0	•	•	
ADMM		•	

For the molecular database, we compare multiple properties for the GGA functional PBE, the meta-GGA TPSS and the hybrid PBE0 functionals of DFT which are described in Chapter 2, each with the three different sizes of the basis set *DZVP*, *TZVP* and *TZV2P* and their respective pseudopotentials. The properties range from simple geometric comparisons of how well bond lengths and angles are reproduced to electronic properties like dipole moments, polarisability, as well as vibrational frequencies. For PBE this is done in a three-way comparison between G16 reference calculations, all-electron and pseudopotential calculations with CP2K, while the comparison for TPSS and PBE0 is between the G16 reference and the CP2K pseudopotential-based approach as shown in table Table 4.1.

The external reference values for these molecules have been obtained from calculations with the G16 software package [82] and the all-electron *Def2-QZVP* basis set of Ahlrichs [53] with *tight* SCF conditions and an *ultrafine* integration grid.

To run the calculations with CP2K a cutoff and relative cutoff of 720 Ry and 60 Ry were used, with a decreased default epsilon of 1×10^{-14} for improved precision in integration grids. The convergence criteria for the energy was set to 1×10^{-7}.

To assess the quality of the basis set and pseudopotentials for the solid-state use case, we relied on the Δ-test database of elemental solids [100] and the Δ-gauge[101] given by

$$\Delta_i(a,b) := \sqrt{\frac{\int_{0.94V_{0,i}}^{1.06V_{0,i}} \left(E_{b,i}(V) - E_{a,i}(V)\right)^2 dV}{0.12V_{0,i}}}, \tag{4.1}$$

and where i denotes a chemical element and a and b for two results obtained with different codes, methods or parameter sets.

The function $E(V)$ is the internal energy function derived from the Birch-Murnaghan equation of states [102]

$$P(V) = \frac{3}{2}B_0\left[\left(\frac{V_0}{V}\right)^{7/3} - \left(\frac{V_0}{V}\right)^{5/3}\right]\left\{1 + \frac{3}{4}(B_1 - 4)\left[\left(\frac{V_0}{V}\right)^{2/3} - 1\right]\right\} \quad (4.2)$$

by integration over P, as

$$\begin{aligned} E(V) = E_0 + \frac{9}{16}V_0B_0\Bigg\{&\left[\left(\frac{V_0}{V}\right)^{2/3} - 1\right]^3 B_1 \\ &+ \left[\left(\frac{V_0}{V}\right)^{2/3} - 1\right]^2\left[6 - 4\left(\frac{V_0}{V}\right)^{2/3}\right]\Bigg\}, \end{aligned} \quad (4.3)$$

with $B_1 = \frac{\partial B}{\partial P}\big|_{P=0}$ the derivative of the bulk modulus B_0. B_0, B_1 and V_0 are determined from a fit on the energy of a bulk system for different volumes and the energy E_0 aligned such that the curves have their minimum at the same energy. The difference expressed in the Δ-gauge, therefore, comes from a shift in V_0 and skewness and opening of the slopes.

While this metric has been proven useful as a way to broadly assess differences in method implementations, it can not be used as a fitness function for optimisation, nor are values between different elements meaningfully comparable. The latter is a result of the naturally varying energies per atom around V_0 (and thus the integral value) for the different elements and the missing normalisation in Equation 4.1. Hence a direct comparison of the equilibrium volumes or bulk moduli may still be required to recover the full picture or to verify the quality of a pseudopotential or basis set for the required quantity to be reproduced.

For the calculations with CP2K (both GPW and GAPW), we have been using a high cutoff of 1000 Ry and relative cutoff of 200 Ry (for GAPW CP2K automatically reduces this, the stated numbers are the specified values in the input). To avoid spurious numerical artefacts in the integration, we are using a fixed reference cell for all the volume points based on 120 % of the anticipated V_0 cell. For GAPW we used 100 and 250 grid points for the Lebedev and radial grids and decreased the default epsilon to 1×10^{-14}. The energy convergence threshold was set to 1×10^{-8} except for a small number of elements which would otherwise not have succeeded. The **k**-point-mesh densities for the different elements are based on the initial suggestions of the Δ-test protocol used for the Wien2k reference calculations.

To obtain pure pseudopotential calculations without Gaussian basis sets we used the CP2K-integrated SIRIUS plane-wave code which was able to make use of the GTH pseudopotentials. The aforementioned reference grid was not employed for these calculations. The G_k and plane-wave cutoffs were set based on plane-wave Δ-test calculations with Quantum Espresso [103] leading up to the SSSP library of pseudopotentials [104], the same holds for the cold smearing and the **k**-point-mesh density.

To test the full potential linearized augmented plane wave (FP-LAPW+lo) implemention in SIRIUS as a means to obtain a full set of reference calculations from within one code (CP2K + SIRIUS), we used settings based

on the EXCITING code as published in [105], since SIRIUS' FP-LAPW facilities were carefully verified against the EXCITING code. At this point it must be noted that the effectively used values for the calculation differ from the summarised report at [1], the effectively employed values were taken for this work and *IORA* [106] was used as the relativistic scheme. The Muffin-tin basis functions were generated with SIRIUS' `atom` command line utility with `--auto_enu --order=2 --type=lo1,lo2,lo3,LO1,LO2` to allow an automatic search of the linearisation energy and include the maximum order of available local orbitals. While the EXCITING basis used for the Δ-test reference calculations could be used after a format conversion, practical usage failed due to unsupported higher-order energy derivatives within SIRIUS at the point of the experiment.

1: https://molmod.ugent.be/sites/default/files/deltadftcodes/supplmat/SupplMat-Exciting.pdf

For the PBE GPW calculations, we used the CP2K built-in implementation of the Exchange-Correlation functional, while the rest of the calculations (for CP2K with GAPW, SIRIUS) use the implementation provided by the *libxc* library [107].

For the extraction of the data from CP2K and G16 output files the cp2k-output-tools and gaussian-output-tools from Section 3.5 have been used, and the procedure to automate the Δ-test calculations is described in Chapter 3.

Table 4.2.: Overview of available CP2K (+SIRIUS) calculation sets for the solid-state benchmark in this work.

	PBE	PBEsol
LAPW	•	•
PW	•	•
GPW, MOLOPT		
DZVP	•	•
TZVP	•	•
TZV2P	•	•
GAPW, MOLOPT		
SVP	•	
TZVPP	•	
QZVPP	•	
GAPW, POB		
DZVP-rev2	•	
TZVP-rev2	•	
TZVPP	•	

4.3. Solid state benchmark results

To assess the quality of the newly generated MOLOPT basis sets and pseudopotentials and to some extent, the CP2K and SIRIUS codes themselves for the solid-state case we are relying on the Δ-test-metric [101] and the Δ-test-database [100] since it provides a well-researched and curated database of values to compare against. Table 4.2 shows the different test sets generated for this evaluation, based on variations of the functional, method and basis set. In the following we are conducting a more thorough discussion of the results. The raw data is available at [108] and tables with the fitted Birch-Murnaghan coefficients can be found in Chapter B. To complete the picture we also compare our all-electron basis sets with the second revised *pob* basis set family [109–112] which has been optimised for the use in, and verified against, crystalline systems.

Table 4.3.: Mean Δ-values across the periodic table (rows 1-4 for GAPW) against the official WIEN2K and SIRIUS LAPW and PW references from this work. Where multiple pseudopotentials with different numbers of valence electrons are available, the better one (usually the smaller core) has been used.

Functional	Method	Reference Basis	Wien2k meV	LAPW meV	PW meV
PBE	LAPW		4.3151		
	GAPW	SVP	54.7378	68.3858	
		TZVPP	8.5953	2.5056	
		QZVPP	0.8970	1.2412	
	PW		2.9916	5.9902	
	GPW	DZVP	6.5652	8.4550	4.2647
		TZVP	4.6304	6.7797	2.3271
		TZV2P	4.3104	6.5978	1.9027
PBEsol	PW			5.9654	
	GPW	DZVP		10.6000	5.0073
		TZVP		7.8192	2.8048
		TZV2P		7.3738	2.4362

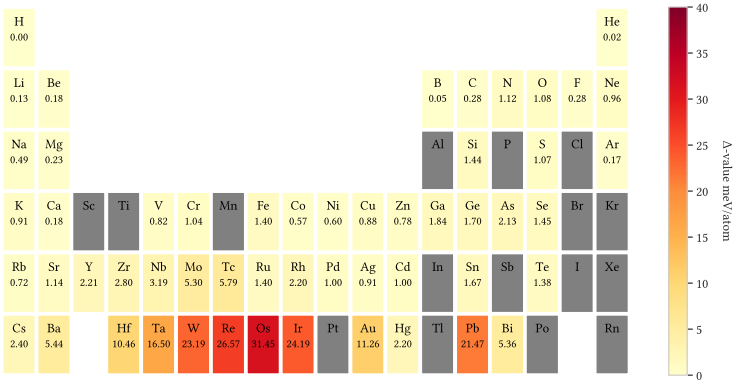


Figure 4.1.: Δ-values for FP-LAPW obtained with SIRIUS compared against the published WIEN2K reference.

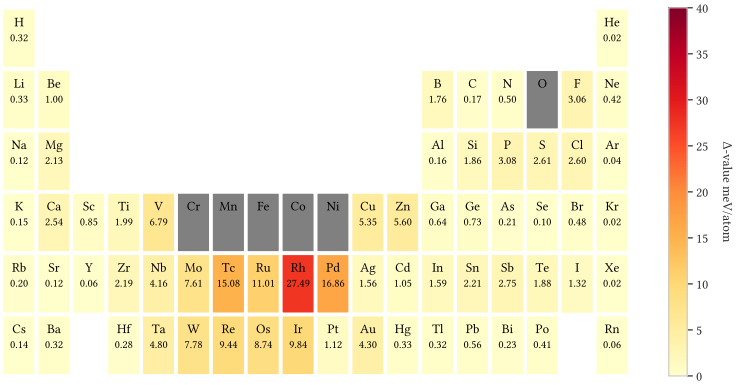


Figure 4.2.: Δ-values for the GTH pseudopotentials obtained with the plane-wave code of SIRIUS integrated within CP2K compared against the published WIEN2K reference. O, Cr-Ni are unavailable due to a bug in the interface between CP2K and SIRIUS while Sc, Ti are outliers under investigation.

4.3.1. FP-LAPW

The breakdown of the Δ-values for the FP-LAPW method provided by SIRIUS within CP2K with the automatically generated muffin tin basis can be seen in Figure 4.1. For most of the periodic table, an excellent match can be observed, especially for the first four rows which are well within the range of best-in-class codes with usually < 1 meV /atom. For the heavier elements in the sixth row of the transition metals, an unusual discrepancy can be observed, which could be attributed to either the generated muffin tin basis (and missing higher-order energy derivatives) or differences in how the relativistic effects are calculated. Another oddity is the failures in convergence for the lower part of the noble gases due to numerical issues, which, together with aluminium, phosphorus, chlorine and bromine, should be investigated further.

4.3.2. GTH Pseudopotentials

The verification of the GTH pseudopotentials is done via the plane-wave capabilities within CP2K provided by the SIRIUS integration and shown in Figure 4.2, with mean and median values of 3.04 meV /atom and 1.08 meV /atom respectively. When comparing this to the best-in-class GTH pseudopotentials at [2] which are around in average 1 meV /atom, with non-linear core correction (NLCC) [113], we fail to reproduce this value when including all converged and fitted systems. Removing the most obvious outliers (V, Cu, Zn, Nb-Pd, Ta-Ir, Au) from the comparison as similarly done in the published values, we can reproduce an average of 1 meV /atom

2: https://molmod.ugent.be/deltacodesdft

Table 4.4.: For most systems a pseudopotential with a larger number of valence electrons (q number) improves the agreement with the high-precision calculations as indicated for the selected elements with the WIEN2K reference. Likely due to the more accurate modelling of delocalized electrons in the crystals. The opposite can be observed for a minor number of elements, likely due to effects previously captured in their pseudopotentials.

Element	Pseudopotential	1. meV /atom	2. meV /atom	3. meV /atom
Li	q1, q3	3.7306	0.3292	
Ca	q2, q10	17.0983	2.5430	
Ti	q4, q12	19.8556	1.9903	
Ga	q3, q13, q21	1.0839	1.6320	0.6406
Au	q1, q11, q19	1298.8153	13.9142	4.3021
Mg	q2, q10	2.1262	11.8316	
Zr	q4, q12	2.1899	3.1396	
Hg	q12, q20	0.3268	5.1893	
Po	q6, q16, q24	0.4112	2.7833	5.9985

without applying NLCC. Limiting our analysis to the main group alone even yields an average of 0.93 meV /atom. As Table 4.4 shows, the pseudopotential selection can make a significant difference for the Δ-test value and possibly condensed systems in general.

4.3.3. MOLOPT Basis Sets

The results for the different sizes of the newly generated MOLOPT basis sets are shown in Figure 4.4, against the WIEN2K and SIRIUS plane-wave references from Lejaeghere et al. [100] and Section 4.3.2 respectively.

Table 4.5 shows a deviation of less than 10 % for the largest TZV2P basis sets compared to the plane-wave results. It is furthermore clear that the errors originating from the GTO basis sets are in the same order of magnitude as from the pseudoization approach. This is further confirmed by the detailed analysis in Figure 4.4, often showing a minor error as in the H-C, Si-Ti, Kr-Sn ranges. For some element ranges, an "overcorrection" can be observed, as in F case or the Cu-Ge series or in the stark case of Vanadium. This is not entirely surprising as the Δ-test does not allow a normalisation on the single atom energy and a small shift of V_0 can have a disproportionate effect.

Nonetheless, this analysis indicates elements for which further optimisation may be required to achieve more accurate results in the solid-state case. We can easily identify two categories: the first is elements with a significant difference but which improve measurably when increasing the basis set size. In the first three rows, Nitrogen and Oxygen fall into this category. The second is elements with a significant deviation from the mean but which do not improve with a larger basis, like the series As-Br.

Fluorine and Iridium require a short discussion as they allow additional insight. The first is an example of the *overcorrection* in the GPW approach compared to the plane-wave approach as illustrated in Figure 4.3. In the plane-wave approach, the volume is underestimated while the GPW approach overestimates by almost the same percentage. This leads to error accumulation when comparing the GPW values against the PW values directly as verified in Figure 4.4. It also serves as an example in which the bulk modulus and its derivative deviate significantly from the reference, yet this is not reflected in the Δ-value (Table 4.6). The GPW results for Iridium on the other hand have been omitted from the

comparison in Figure 4.4 due to its large error on the Δ-value as reported in Table 4.6. The likely cause of this poor behaviour on the basis sets is the underrepresentation of Iridium in the Small Molecules Database (Figure 3.1). Comparing the Δ-value for Iridium with the TZV2P basis set and Flourine in the plane-wave basis unearths a more critical deficiency in the Δ-test itself: despite a similar error in the equilibrium volume V_0 between Flourine and Iridium and smaller errors in B_0 and B_1 for Iridium is the Δ-value for Iridium an order of magnitude larger. The root cause is an absent energy normalisation which can not be implemented due to the missing general reference energy within electron structure theory. This leads to larger deviations in the Δ-value than would be expected due to the change in one of the parameters, and questions the meaning of an unweighted global average of Δ-values across the periodic table. Finally, for the practitioner, this again highlights the importance of direct quantity comparison rather than relying solely on Δ-values.

	all elements		main group	
Reference	WIEN2K	PW	WIEN2K	PW
Basis	meV /atom	meV /atom	meV /atom	meV /atom
DZVP	5.2984	3.0119	3.3017	2.9722
TZVP	4.0642	1.8537	2.1544	1.8687
TZV2P	3.9651	1.6758	1.8911	1.5983
PW	3.6331		1.2960	

Table 4.5.: Mean Δ-values for the MOLOPT basis sets and pseudopotentials across all elements, respectively only the main group. The values for systems with non-vanishing magnetic moments are missing for the plane-wave values and reference due to initialisation issues in the code.

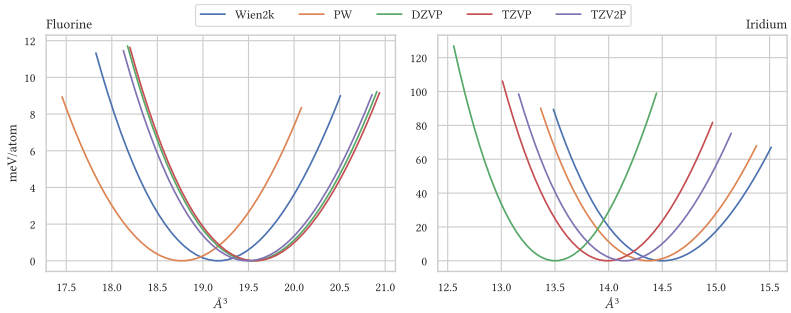


Figure 4.3.: Energy-Volume curve for Fluorine and Iridium, for WIEN2K, SIRIUS and CP2K with the MOLOPT basis set for the *q7*, *q17* pseudopotential variants respectively.

	Fluorine				Iridium			
	δV_0	δB_0	δB_1	Δ	δV_0	δB_0	δB_1	Δ
PW	2.08	11.87	88.00	3.06	0.88	2.00	2.69	9.84
DZVP	1.95	0.94	4.66	2.83	6.89	55.55	16.72	96.07
TZVP	2.09	0.14	4.36	3.02	3.52	24.72	11.69	43.16
TZV2P	1.69	0.80	2.55	2.42	2.40	13.98	8.13	28.11

Table 4.6.: Percent error for V_0, B_0 and B_1, and Δ-values with respect to WIEN2K for Fluorine and Iridium for SIRIUS plane-wave and CP2K GPW for the *q7*, *q17* pseudopotential variants respectively.

4.3.4. All-Electron MOLOPT Basis Sets

Besides the pseudopotential-based MOLOPT basis sets explored in the previous section, we also report the behaviour of the newly generated MOLOPT all-electron basis sets SZV, TZVPP and QZVPP. As the naming indicates, these basis sets use the same contraction patterns as the *def2* basis sets by Weigend and Ahlrichs [53] and Weigend [54]. For CP2K, the GAPW method [36] must be employed to model electron density around the atom centre with high enough accuracy without requiring an impractical amount of plane waves. At this point we are only interested

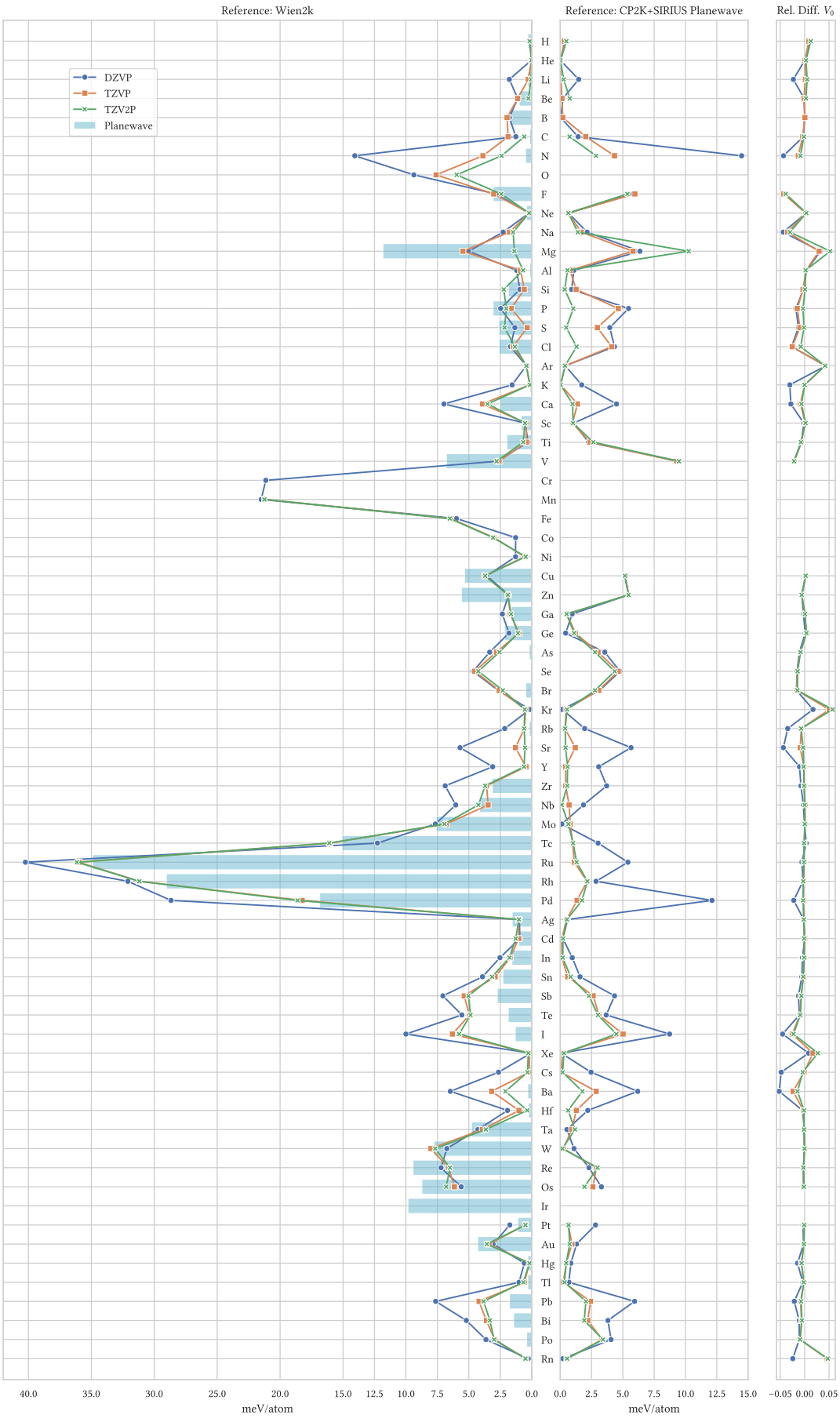


Figure 4.4.: PBE Δ-values for the MOLOPT basis sets against the Wien2k and CP2K + SIRIUS plane-wave references.

Family	GAPW Mean	GAPW Median	Family	GPW Mean	GPW Median
SVP	3.779	1.599	DZVP	3.599	1.796
TZVPP	1.116	0.689	TZVP	2.575	1.718
QZVPP	0.897	0.517	TZV2P	2.358	1.512

Table 4.7.: Average Δ-values for the first four rows of the periodic table for the all-electron MOLOPT basis set families SVP, TZVPP & QZVPP in meV /atom and the respective values of the GPW MOLOPT families DZVP, TZVP and TZV2P for the same subset of elements (see also Figure 4.5a), both against the WIEN2K reference.

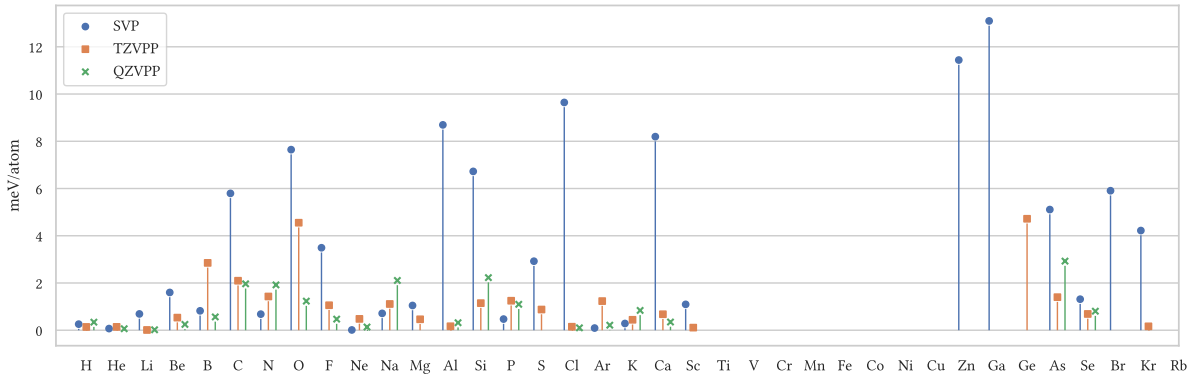
in the all-electron basis sets, hence we limit ourselves to the first four rows of the periodic table since for rows five and six a pseudoization approach is used. Some transition metal values are further excluded due to challenges in reaching convergence.

As with the GPW approach before, we can first note the consistent improvement of the Δ-values with larger basis sets, with a minor number of exceptions like nitrogen and sodium as shown in Figure 4.5a. In particular, the averages for the QZVPP family are below 1 meV /atom as indicated in Table 4.7, placing this family and method amongst the best-performing codes of the Δ-test benchmark. Comparing averages for the same subset of elements furthermore shows a significant improvement over the (GPW) families DZVP, TZVP & TZV2P. One can further observe the significantly larger error for the SVP family, drastically corrected at the TZVPP level.

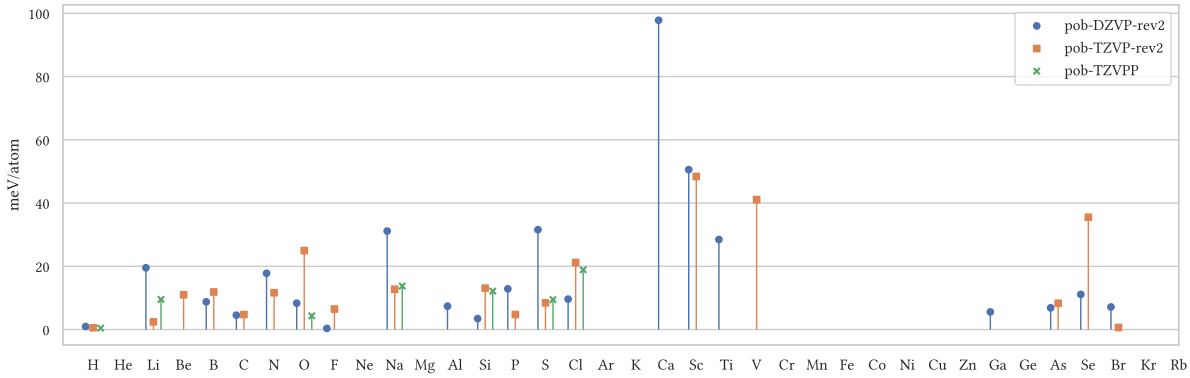
4.3.5. All-Electron POB Basis Set

To obtain an additional external point of reference for the improvements of our MOLOPT family, we selected the revised POB basis set families DZVP and TZVP, and TZVPP [111, 112]. Besides the fact that these basis sets have been optimised for use in solid-state calculations, they also follow the same contraction scheme as the *def2* families and thus our own MOLOPT basis sets, but they use less diffuse primitives in general. We again restrict the comparison to the first four periods as the fifth and sixth periods employ effective core potentials (ECP), for which CP2K has only basic support, and which has been shown to require further improvements. Figure 4.5b contains the overview of the converged calculations. The optimization procedure behind the generation of the POB basis set relied on a variational optimization approach for a database of compounds with single point energies obtained with the CRYSTAL09 code with the hybrid functional PW1PW. The verification was done against the CRYSTAL09-provided basis set by comparing lattice constants and other properties for both the PW1PW functional and Hartree-Fock calculations.

We observe an error which is on average an order of magnitude larger than obtained with any of the MOLOPT basis set families. Additionally, we do not observe the same systematic improvement for larger basis sets seen from the MOLOPT family. For the TZVPP variant, this may not be necessarily expected since it has not yet been revised, but also a comparison between the revised DZVP and TZVP does not exhibit this behaviour. A reason for this difference could be the optimization for hybrid calculations rather than plain DFT functionals.



(a) Δ-test values for the MOLOPT SVP, TZVPP, QZVPP all-electron basis sets against the WIEN2K reference.



(b) Δ-test values for the POB DZVP-rev2, TZVP-rev2 and TZVPP all-electron basis sets against the WIEN2K reference.

Figure 4.5.: GAPW-based all-electron Δ-test calculation results for MOLOPT and POB basis sets for the first 4 rows of the periodic table.

4.3.6. PBEsol

The support for FP-LAPW calculations directly available within CP2K via the SIRIUS integration opens up a route to a complete chain of verification for arbitrary supported functionals within the same code (Figure 4.6).

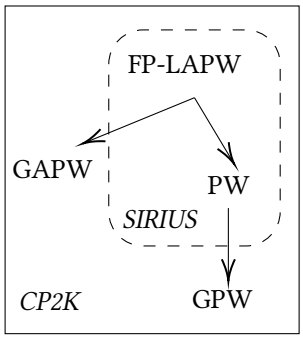


Figure 4.6.: The route to a fully automated GTH pseudopotential and GTO basis set verification within a single code.

As a first example, we have chosen PBEsol due to its similarity to the well-validated PBE functional. As the overview in Table 4.3 illustrates, the difference in the mean Δ-value between plane-wave and LAPW values is less than 0.05 meV /atom. Unfortunately, this hides the fact that numerical instabilities in the SIRIUS LAPW code have thinned out the number of converged calculations across the periodic table from 56 to 35 elements. The results are therefore now limited to rows 1-3, 5 and the first two groups of the periodic table as shown in Figure 4.7. Casual testing has shown that some of those instabilities vanish with smaller cutoffs than employed with our high accuracy settings.

The exact source of these instabilities could not be determined as for all calculations involving SIRIUS the LIBXC library has been used for the implementation of the exchange-correlation functional. With PBEsol being a reparametrisation of PBE and similar numerical issues having occurred for a smaller number of elements with the PBE functional, this may hint to a more fundamental issue with the numerics in the SIRIUS LAPW implementation in combination with the employed high precision settings

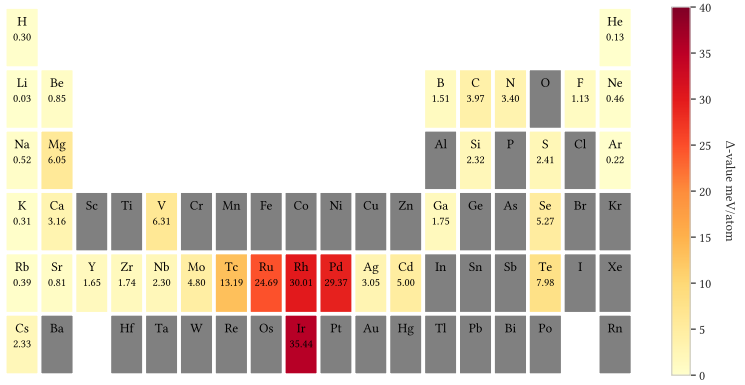


Figure 4.7.: PBEsol Δ-values for plane-wave with MOLOPT pseudopotentials with SIRIUS LAPW as a reference.

or the automatically generated muffin tin basis. Further investigation of these issues, which have to be resolved to succeed at a fully automated benchmark with SIRIUS within CP2K, will be part of future work.

4.4. Molecular benchmark results

While for the solid-state benchmark only the alignment of the energy-volume curve was evaluated as a proxy for the overall performance of code and parameters, does the molecular benchmark permit a much more nuanced analysis.

4.4.1. Geometry

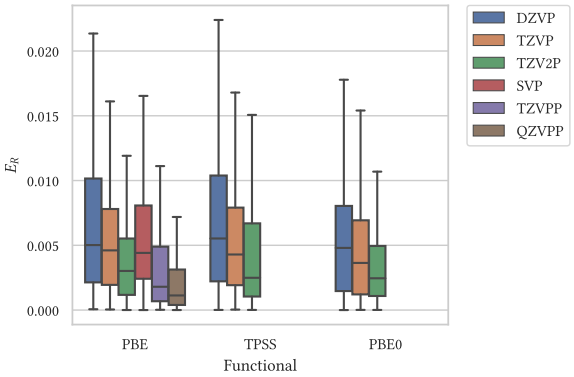
For the comparison of purely geometric properties, we are looking at relative errors for the bond lengths, and at absolute errors for the angles. Due to convergence issues when obtaining the reference values with G16 or comparison values with CP2K, not all molecules from the database are taken into account in the statistics. The frequency analysis in the following Section 4.4.3 has furthermore been used to filter molecules where the calculation inadvertently converged to a transient state rather than a minimum. Appendix Chapter A and the data set [114] contain the details of the setup and the raw data of the included molecules.

Analysis of the relative errors of bonds to their respective all-electron reference for each of the functionals PBE, TPSS and PBE0 in Figure 4.8 shows a consistent improvement for larger basis sets, reflected in lower median values and narrower interquartile ranges. The median values are consistently below 6 ‰ which corresponds to a difference of 0.01 Å on the average bond length of 1.73 Å in the database.

To verify this behaviour across different chemically important bonds, the same analysis has been repeated in Figure 4.13 for hydrogen and carbon bonds, the first 2 rows of the oxygen, nitrogen and fluorine groups, and metal complexes. The results from the inter-group comparison do not show a significant deviation for any particular element. We nonetheless observe larger errors for the oxygen and fluorine groups for the smaller basis sets *DZVP* and *TZVP*, which are only recovered when introducing an additional polarisation term in the *TZV2P* set. Carbon bonds are in general reproduced at a relative error of an order of magnitude lower

functional basis	PBE $\cdot 10^{-3}$	TPSS $\cdot 10^{-3}$	PBE0 $\cdot 10^{-3}$
DZVP	7.2864	7.9210	6.8635
TZVP	6.3310	6.8225	6.3875
TZV2P	5.3635	5.5192	4.8449
SVP	7.8845		
TZVPP	6.1766		
QZVPP	5.0116		

(a) Mean relative errors across all bonds.



(b) Distribution of relative errors across all bonds. The outliers have been hidden to focus on the behaviour around the median, whiskers are drawn at $1.5x$ of the interquartile range $Q3 - Q1$.

(c) Histograms are normalised to account for different availability of calculations across basis sets.

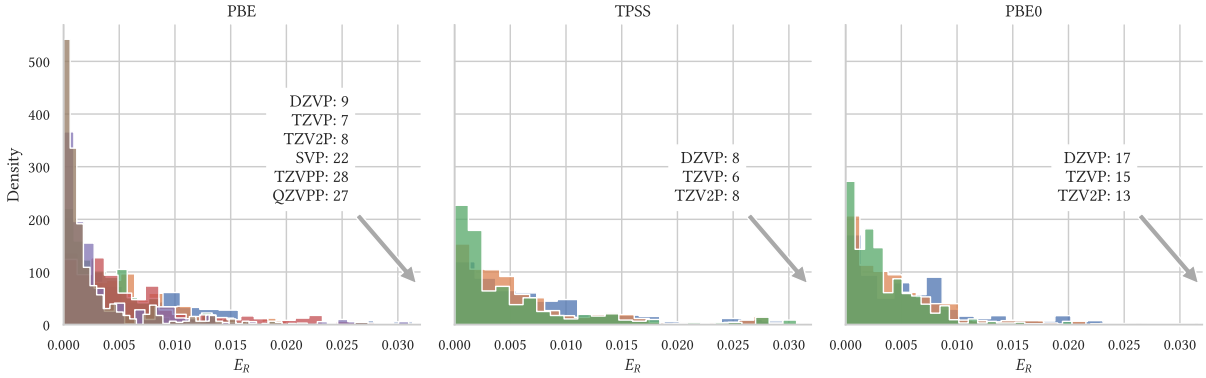


Figure 4.8.: Both (b) and (c) show the same trend, larger basis sets leading to a narrower spread as they reproduce the all-electron bond lengths much more accurately.

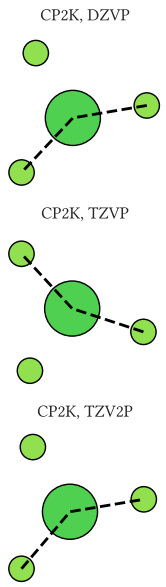
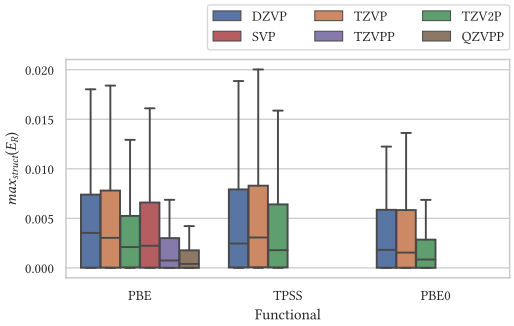


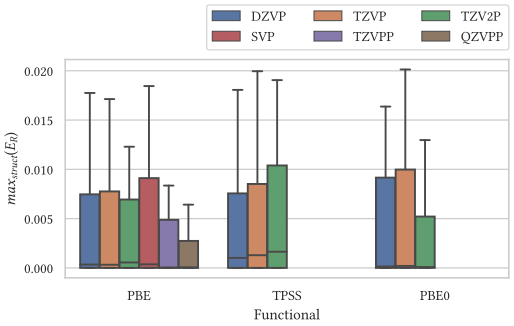
Figure 4.9.: For NiF_3 the largest angle may not always be located between the same Fluorine atoms, applying also to other highly symmetric molecules in the database like $AuCl_3$, AlH_3, etc.

than the average, although the number of such bonds within the database is rather low.

To automatically identify non-linear and dihedral angles we relied on the facilities of the G16 software package. As shown in Figure 4.10 the previously observed and anticipated trend for larger basis sets leading to narrower spread continues, in particular for the all-electron basis sets. The one notable exception here would be the TPSS functional, where larger basis sets tended to reproduce dihedral angles slightly worse. Comparing angles in a fully automated fashion and without additional chemical information has its difficulties: for highly constraint or even linear systems, many if not all angles are essentially fixed due to symmetries. Comparing such angles is then rather akin to a sanity check but its expressive power is reduced. Furthermore, for almost symmetrical systems, numerical noise can lead to obtuse angles occurring between different sites of the same kind, even with the same functionals but different basis sets, as the example of NiF_3 in Figure 4.9 illustrates. This could be solved on a case-by-case basis by simple permutation of equivalent sites if this information would be available in the database. We, therefore, limited ourselves to looking at the distribution of the maximum relative angle error and keeping trivial angles in the statistics.



(a) Distribution of the maximum relative error of angles per structure.



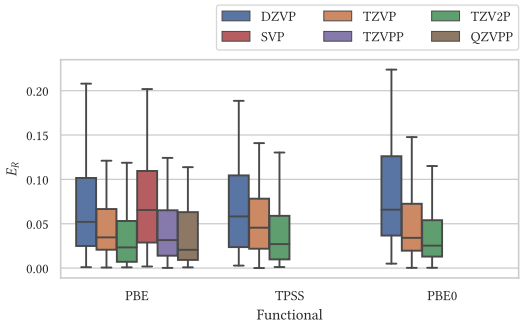
(b) Distribution of the maximum relative error of dihedral angles per structure.

Figure 4.10.: Relative errors of angles (non-linear and dihedral) for the small molecules database across functionals and basis sets.

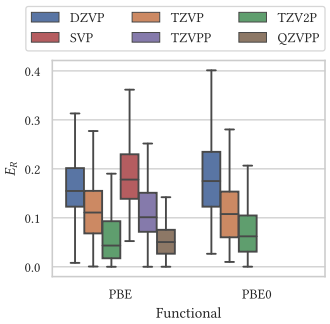
4.4.2. Dipole moments and polarizability

For a number of selected molecules with non-vanishing dipole (and availability across all functionals), we compare the magnitude of the dipole moments with the resulting statistics of the relative errors plotted in Figure 4.11a. Comparing just the order of the relative errors to the purely geometric quantities from Section 4.4.1 we see a much larger variance in the results, which is not completely resolved by employing the GAPW method in the same manner as for the vibrational modes discussed in Section 4.4.3. The general trend for larger basis sets to improve the error continues nonetheless here, leading to median relative errors well below 2 %.

The comparison of the average molecular polarizability reveals the effect of basis set quality and size in the most striking manner, as illustrated in Figure 4.11b. The error continuously and significantly improves going from smallest to largest basis set within the same functional and method (pseudopotential vs. all-electron). On the other hand, the sensitivity of this quantity due to second-order derivatives and high-order geometric dependency leads to the largest relative errors of the studied properties.



(a) Distribution of the relative error of dipole moments (for 87 converged structures with non-vanishing dipole moment).



(b) Distribution of the relative error of the trace, TPSS missing due to unavailability of this property in CP2K for meta-GGA.

Figure 4.11.: Relative errors of dipole moments & average molecular polarizability.

4.4.3. Vibrational Spectra

When calculating and comparing vibrational spectra in a high-throughput fashion, the difficulty lies in correctly identifying and pruning low frequencies and matching the remaining ones. Pruning is required because different codes may not identify the same degenerate frequencies, and collapse them into one. Matching is needed since the list of frequencies produced by the codes are unlikely to agree in order or exact value. One reason for such mismatches to occur is the numerical differentiation, which may amplify differences in geometric properties or lead to non-negligible low frequencies, indicating either an underlying problem with the structure itself or requiring even finer integration grids. Others are when a simulation inadvertently converged to a transient state instead of the true minimum, identifiable by significant negative frequencies. In either way, we have used a pre-screening on the vibrational spectra to exclude such structures completely from the analysis, assuming any good agreement in geometry (or other properties) would have been incidental and further manual investigation would be required to assess whether or not a structure is still meaningfully converged. Doing this in an automated fashion would require more elaborate machinery and likely more than just geometrical information in the database (e.g. validated chemical information). To compare the frequencies we are thus looking at the minimal relative error per frequency (when compared to the list of frequencies obtained from the reference), with the final comparison shown in Figure 4.14 based on the mean of the relative errors of the matched frequencies.

What can be observed in Figure 4.14 is the much larger average relative error as compared to the bond length comparisons, and while the median gradually improves for TPSS and PBE0, the same can not be observed for PBE with GPW (e.g. pseudopotentials), or for the spread for all functionals with GPW. With the all-electron basis sets SVP, TZVPP and QZVPP, which employ the GAPW method, we can recover the usual behaviour of significantly improved spread and median when going to larger basis sets. While the aforementioned difficulties for obtaining frequencies could be at play here, it is also possible that the improved precision by using GAPW – which gives a finer radial grid around the atom to model the core charge distribution – resolves some of the numerical issues often plaguing frequency calculations.

4.4.4. ADMM

When applying Hartree-Fock exchange as part of a hybrid-functional calculation like PBE0, the computation becomes more expensive than a pure DFT calculation. Indeed, a $\mathcal{O}(N^4)$ dependency on the system size as previously outlined in Section 2.5 is introduced. This makes methods like the Auxiliary Density Matrix Method (ADMM) worthwhile to consider as it allows us to replace the basis used to calculate the Hartree-Fock exchange energy with a smaller one, provided it is able to reproduce the required observables accurately enough. Even though the systems in the SMDB will not profit from a significant speedup, they will allow us to assess the quality of ADMM and the newly generated ADMM basis. This

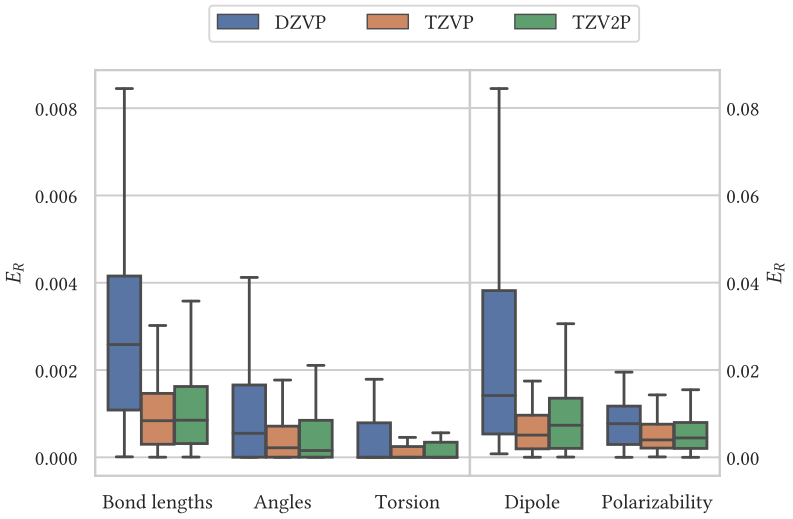


Figure 4.12.: Comparison of relative errors for PBE0+ADMM calculations against PBE0, with CP2K.

new basis was generated with a matching size for each of the Hybrid subsets of the *DZVP*, *TZVP* and *TZV2P* of MOLOPT basis sets. Interesting in this case is therefore not only the error against the G16 reference value but also the error compared to the CP2K PBE0 calculation.

Figure 4.15 contains the comparison of a subset of 200 molecules for which ADMM calculation data was available for both PBE0 and PBE0 with ADMM, against the G16 reference values. While minor deviations and shifts in spread and median can be be observed for the smaller DZVP basis set, this effect vanishes for the larger TZVP and TZV2P basis sets completely. This is confirmed by a direct comparison of the ADMM data against the CP2K PBE0 calculations in which the average errors for purely geometric values like bond lengths and angles drop below 2 ‰ and the more sensitive electronic properties below 1 %, as shown in Figure 4.12.

The indicated sizes describe the main basis set. For the auxiliary basis set we used the *admm-dzp* together with *DZVP*, and *admm-tzp* for both *TZVP* and *TZV2P*.

4.4.5. GAPW/All-Electron comparison

While the first sections have been focused on comparing CP2K and its basis sets and pseudopotentials against reference values obtained from G16 with an all-electron *def2* basis set, we are now focusing on an intra-CP2K comparison by comparing the pseudopotential MOLOPT basis sets for PBE against AE *QZVPP* reference values. Since the previous evaluations have shown a very narrow error between *QZVPP* and the G16 values we believe that *QZVPP* can itself serve as a reference and will allow us to verify the consistency of the data within CP2K itself. Indeed, the side-by-side comparison in Figure 4.16 shows only small statistical deviations between the *MOLOPT* basis sets DZVP, TZVP, TZV2P and G16 on the left and CP2K with the QZVPP as a reference on the right.

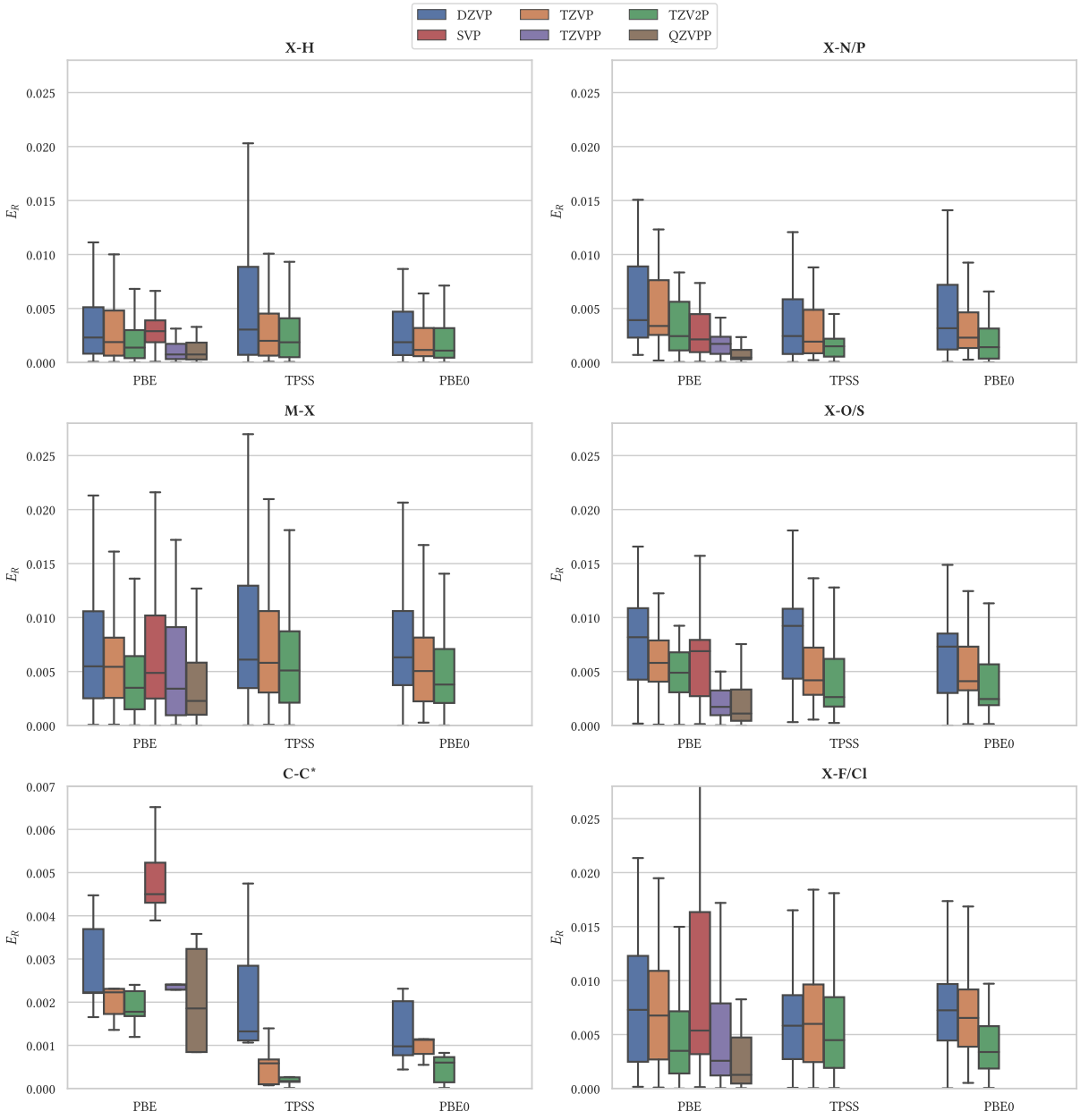
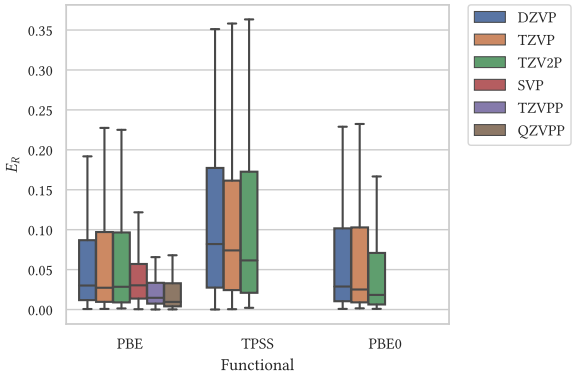


Figure 4.13.: Distribution of relative errors across all bonds (with outliers hidden) for different groups of bonds, comparing different basis set sizes. For *C-C* bonds the range has been adjusted as their error is at an order of magnitude lower, but given the low number of such bonds, the results are skewed, especially for the all-electron basis sets. A similar situation occurs at the distribution for the SVP basis and bonds involving Fluorine or Chlorine where the spread is higher than for all other basis sets and functionals. For easier comparison and legibility reasons, we refrained from rescaling the axis again to accompany this specific boxplot where the upper whisker is at $E_R = 0.033$.

functional basis	PBE	TPSS	PBE0
DZVP	0.1205	0.1790	0.1203
TZVP	0.1300	0.1455	0.1274
TZV2P	0.1398	0.1556	0.1667
SVP	0.1956		
TZVPP	0.1149		
QZVPP	0.1898		

(a) Mean relative errors across a mean of minimal relative errors of frequencies.



(b) Distribution of minimal relative errors. The outliers have been hidden to focus on the behaviour around the median, whiskers are drawn at $1.5x$ of the interquartile range $Q3 - Q1$.

(c) Histograms are normalised to account for different availability of calculations across basis sets.

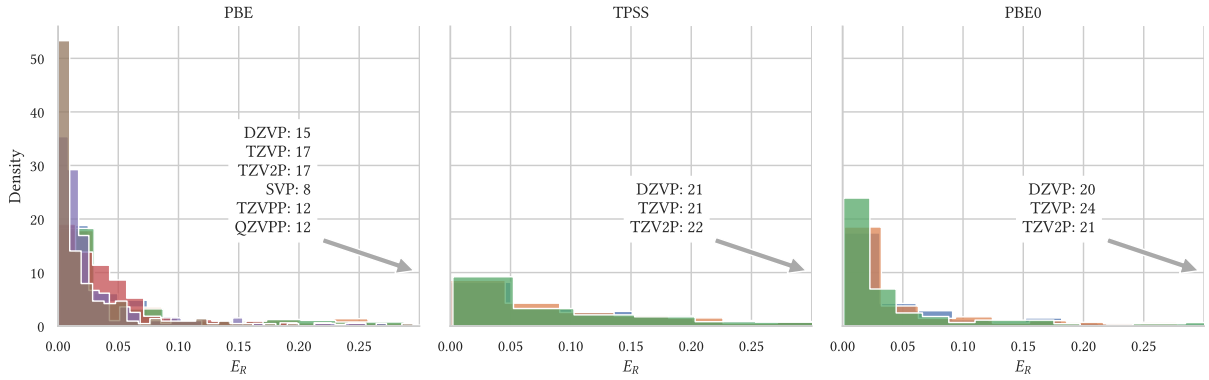


Figure 4.14.: Distribution of averages of minimal relative errors of vibrational modes.

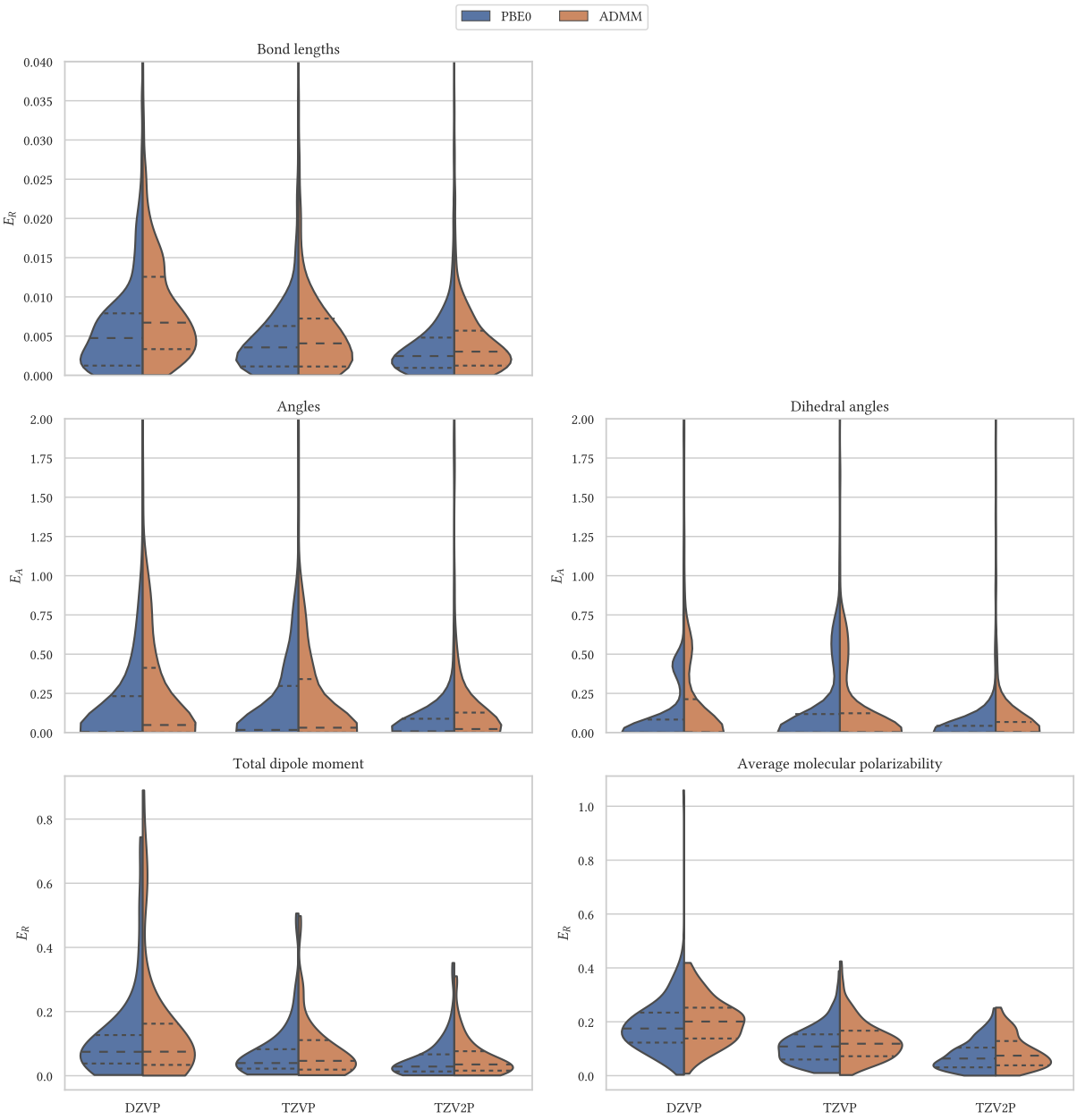


Figure 4.15.: Comparison of relative (for angles: absolute) errors between PBE0/PBE0+ADMM and G16.

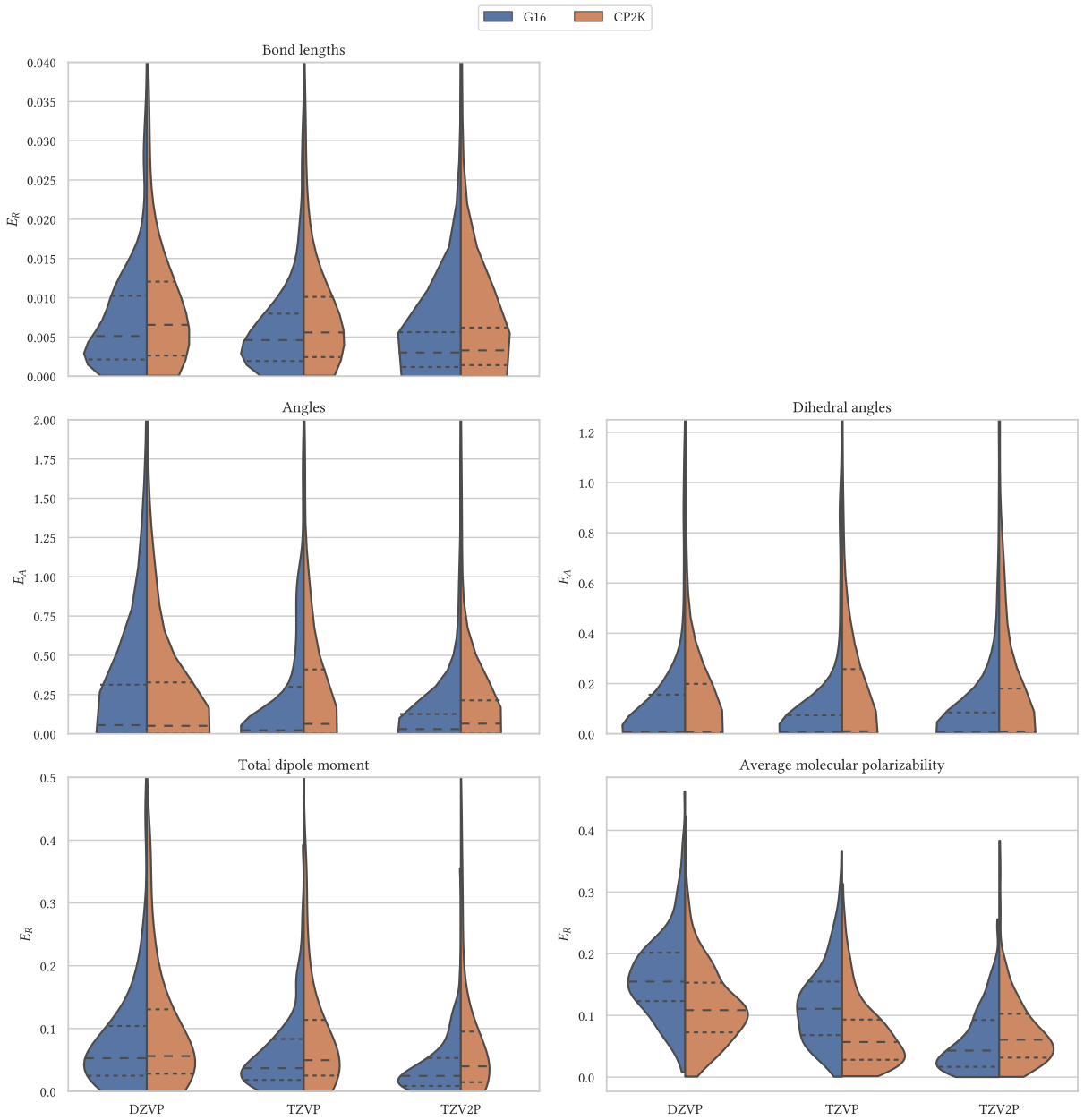


Figure 4.16.: Comparison of relative (for angles: absolute) errors between the MOLOPT basis sets DZVP, TZVP, TZV2P and the G16 and CP2K QZVPP reference values. Axis is truncated to focus on the distribution around the median.

4.5. Conclusion

We have shown that the revised *MOLOPT* basis sets are well-defined for a broad number of quantities and for both solid-state and molecular systems. As expexted, the results improve as one adds more functions or angular momenta. This behaviour has been shown to hold across the three functionals PBE, TPSS and PBE0, as well as with an ADMM calculation for PBE0. For the meta-GGA TPSS, it even shows that a pseudopotential and basis set optimised for a different functional of the same rung (SCAN) performs at a high enough precision to warrant its usage despite being optimised for a different functional.

For the solid-state case, we observed that semi-core pseudopotentials tend to lead to better results, likely due to the delocalization of electrons. The all-electron MOLOPT basis sets perform exceptionally well, also in cross-comparison with other basis sets based on similar principles of generation. This may indicate that the pseudopotentials can be a substantial source of deviation in such a high-accurracy calculation.

To be able to automate this analysis further, several improvements to either the database itself or the analysis tools must be made. For the database one likely has to follow the tradition of quantum chemistry data bases (like the GMTKN) and include validated reference data. This is in particular relevant for integer data, such as the number of vibrational frequencies and their degeneracies. But also identifying only (chemical) relevant frequencies would help to discover transition states. Other required data are symmetries, which can then be used to automatically validate the configuration and to match results between codes by e.g. automatic rotation of the molecules to minimize the RMSE of the position differences. Relying only on heuristic approaches, such as the ones we have implemented, is ultimately error-prone.

5. ADMM with k-point support in periodic Hartree-Fock Exchange

In the previous Chapter 4 we have been using PBE0 as an example for a hybrid DFT functional, testing the portability of basis sets and pseudopotentials otherwise optimized for the GGA functional PBE on a set of molecular structures. It is in this context of Hybrid functionals that HFX – despite its early introduction in Computational Chemistry[115, 116] – is still a very relevant target for optimization as it becomes the rate-limiting factor in the calculation of larger systems.

One aspect which has not been discussed previously in Section 2.4 is the challenge arising in the calculation of the HFX energy in the reciprocal space, i.e. the sum of the diagonal terms of the ERI

$$E_{\mathrm{X}} = -\frac{1}{N^2} \sum_{\mu\nu\mathbf{k}\mathbf{k}'} P^{\mathbf{k}}_{\mu\nu} P^{\mathbf{k}'}_{\nu\mu} \iint \phi^*_\mu(\mathbf{r}_1;\mathbf{k}) \phi_\nu(\mathbf{r}_1;\mathbf{k}') \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \phi^*_\nu(\mathbf{r}_2;\mathbf{k}') \phi_\mu(\mathbf{r}_2;\mathbf{k}) d\mathbf{r}_1 d\mathbf{r}_2, \tag{5.1}$$

which when expanded with Bloch functions for the ϕ_μ as in Equation 2.56 becomes a divergent series due to non-vanishing co-densities for $\mathbf{k} = \mathbf{k}'$, e.g.

$$\int e^{(\mathbf{k}'-\mathbf{k})\cdot\mathbf{r}_1} \, \mathrm{u}(\mathbf{r}_1). \tag{5.2}$$

There are different ways to resolve this divergence, one of them is based on adding an auxiliary function to the Exchange energy which cancels the divergent term and separately subtract it [117]. The disadvantage of this auxiliary function is that it must match the crystal class for which E_{X} is to be calculated. Instead, we follow the Γ-point periodic HFX implementation of CP2K, which directly replaces the Coulomb metric in the integration with a truncated Coulomb operator, based on the work by Spencer and Alavi [118]. That is, the $1/\|\mathbf{r}_2 - \mathbf{r}_1\|$ in Equation 5.1 gets replaced by

$$g_{\mathrm{TC}}(\mathbf{r}) = \begin{cases} \frac{1}{|\mathbf{r}|} & \text{if } |\mathbf{r}| \leq R_c, \\ 0 & \text{otherwise .} \end{cases} \tag{5.3}$$

A well established value for this truncation radius is found to be at half the cell distance $L/2$. For the DFT correction in ADMM (the DFT Exchange functional, see Equation 2.80) it is then only natural to apply the same truncation as for the Exchange energy. This is also the default within CP2K.

As we have shown in Chapter 2, actually calculating the HFX energy in a Gaussian-Type basis set based code for a periodic system with **k**-point sampling is not entirely trivial and computationally expensive. Yet, there are some codes which have supported this for some time [119] or have gained experimental support as of late [120]. Our contribution, which

we are going to discuss here is not only the addition of HFX implementation which supports **k**-point sampling to CP2K, but also its novel combination with ADMM. The latter allows us to use a significantly smaller basis without a noticeable reduction in accuracy, which is crucial to reduce the number of four-centre integrals to calculate.

We consider the implementations above, as well as our own, canonical implementations. In the sense that besides employing usual identities like $P^T_{\mu\nu} = P^{-T}_{\nu\mu}$ (and corresponding for the Exchange matrix) as well as applying required truncation schemes to avoid the divergence and limit the problem to a finite number of neighbour cells, no other optimisations are being used to tackle the actual computational complexity.

This is in contrast to Resolution of Identity (RI) approaches with suitable auxiliary basis sets to reduce the ERIs from 4- to 3 or 2-centre integrals in HF [121, 122] or Post-HF methods [123–125], as well as the more refined approaches developed in the group of Head-Gordon [126, 127]. In their occupied orbital RI-K (occ-RI-K) approach, several components are combined to obtain a very efficient (in both compute and memory usage) algorithm. The focus herein lies on the efficient construction of the exchange matrix K since fast methods for the J matrix which covers the classical Coulomb interaction have already been established prior [128–130]. The RI approach reduces the the ERI to a 2-centre quantity by introducing an auxiliary basis in which to expand the co-densities

$$\phi^*_\mu(\mathbf{r}_1)\,\phi_\lambda(\mathbf{r}_1) = \sum_Q C^{\mu\lambda}_Q\,\chi_Q(\mathbf{r}_1)\,, \tag{5.4}$$

leading to a simpler expression for K

$$K_{\mu\nu} = \sum_{\lambda\sigma}\sum_{PQ} C^{\mu\lambda}_P C^{\nu\sigma}_Q (P\,|\,Q) P_{\lambda\sigma r}. \tag{5.5}$$

The improvement then comes by exploiting the fact that for both the energy (which requires only the diagonal elements of the exchange matrix in the first place) and gradient only the occupied MO must be taken into account. When employing the Direct Inversion of the Iterative Subspace (DIIS) method, this finds direct application as in that case only the product of the Fock and density matrix are required to construct the DIIS error vector, and thus only elements corresponding to occupied MO contribute. The combination of this *compression* of the K matrix together with RI approach then loads to a reported speed-up from 3-5x compared to other RI approaches [131] in a molecular setting. This can then be extended to the solid state use case [126, 132] by following the same construction of the periodic HFX in a GTO framework. Going further, the group then progresses by combining this with the Tensor Hypercontraction framework [133–135] (respectively, interpolative separable density fitting (ISDF)) into a *THC-oo-K* algorithm. The main idea being the expansion of products of atomic orbitals into a sum of interpolation vectors, weighted by the orbitals

$$\phi_\mu(\mathbf{r})^*\phi_\nu(\mathbf{r}) \approx \sum_P^{N_{\mathrm{ISDF}}} \phi_\mu(\mathbf{r}_P)^*\,\phi_\nu(\mathbf{r}_P)\,\xi_P^{[nn]}(\mathbf{r}), \tag{5.6}$$

which then leads to a simplified (approximated) exchange matrix which only needs the basis functions at a set of interpolation points N_{ISDF} and integration between the the interpolation vectors $\xi_P(r)$

$$M_{PQ} = \iint d\mathbf{r}_1 d\mathbf{r}_2 \frac{\xi_P(\mathbf{r}_1)\,\xi_Q(\mathbf{r}_2)}{r_{12}}. \tag{5.7}$$

This is then extended towards a periodic **k**-point ISDF and adapted to fit only occupied orbitals.

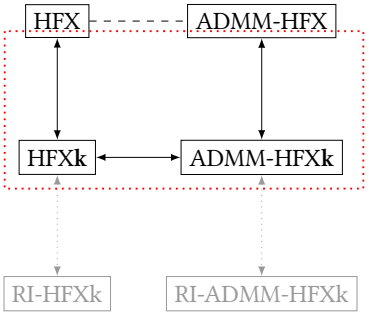


Figure 5.1.: With the addition of the Hartree-Fock Exchange with **k**-point (HFX**k**) implementation together with the ADMM integration we are validating the different code parts as illustrated. While the the practical use of (ADMM-)HFX**k** may be limited for larger systems, it serves as a stepping stone for other methods within CP2K.

5.1. Verification of HFXk

As base for the verification path as depicted in Figure 5.1 we are using the well-studied Hexagonal boron nitride (h-BN) and Lithiumhydrid (LiH) systems as examples for 2D and 3D systems. We are furthermore using the fact that when taking the supercell or superlattice of a primitive unit cell and calculating the system at the Γ-point, this corresponds to a sampling of the reciprocal space. Due to the Fourier transformation between real and reciprocal space, identifying special points relevant for studying bulk or surface materials is not as easily done in the supercell approach as when employing solely **k**-point sampling [136]. Likewise, band structures obtained from supercell calculations are folded upon themselves, requiring extra treatment [137, 138]. To sample the **k**-point space we are using a regular Monkhorst-Pack [139] grid. The issue of the band-folding or explicitly identifying relevant symmetry points in reciprocal space is not relevant for achieving converged total energies given fine enough mesh grids, number of supercells respectively. The detailed results of this can be seen in Figure 5.2 and Figure 5.3, respectively. We can conclude that an MKP grid of 9x9(x9), respectivly a supercell of 5x5(x5) is enough to achieve convergence for both systems as well as an agreement between the total energies of 1.106×10^{-5} E_h for LiH and 1.254×10^{-4} E_h for h-BN, using the revised all-electron POB DZVP basis set [112] employing the GAPW method, and a moderate `EPS_SCF` of 1×10^{-6}. The truncation radius was kept constant across the different supercells. As the timing plots in Figure 5.4 illustrate do the different calculations exhibit different scaling behaviour for constant resources. As long as the primitive integrals fit into the cache, only the first SCF cycle of the periodic HFX takes a significant amount of time. But once the cache is exhausted, the scaling of the HFX**k** implementation becomes significantly more favourable in terms of sampling of the reciprocal space.

5.1.1. Schwarz-Screening

While we are not relying on additional screening methods, including screening on the density matrix, for now in this reference implementation, we are still using the Schwarz-inequality to reduce the number of 4-center ERI to calculate [60, 140–143].

As Figure 5.5 illustrates, the default value of CP2K for `EPS_SCHWARZ` of 1×10^{-10} which has been used throughout the calculations is still justified. Making it larger leads to unphysical energies, while lowering it increases

Mesh	E_{tot} [a.u.]	Supercell	E_{tot} [a.u.]
3x3x3	-7.718231	1x1x1	-7.94597
6x6x6	-7.717316	2x2x2	-7.697665
9x9x9	-7.717312	3x3x3	-7.718219
12x12x12	-7.717312	4x4x4	-7.717115
15x15x15	-7.717312	5x5x5	-7.717323
18x18x18	-7.717312		
21x21x21	-7.717312		
24x24x24	-7.717312		
27x27x27	-7.717312		
30x30x30	-7.717312		

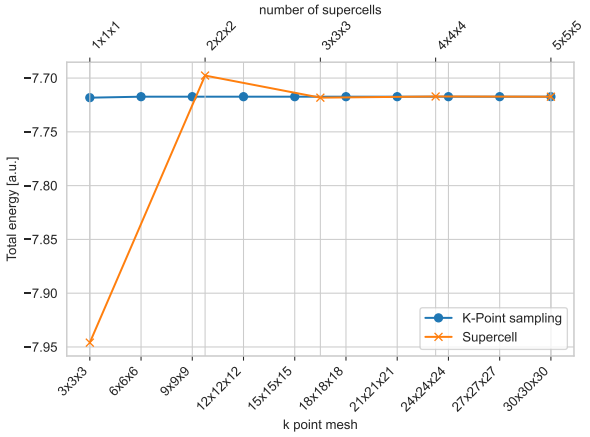


Figure 5.2.: Convergence of the total energy for ever finer **k**-point meshes, larger supercell configurations respectively, of LiH. The *mesh* column refers to the MKP grid, while the *supercell* refers to the supercell configuration. The respective energies (per cell) are given next to it. There is no direct correspondence between supercell configuration and **k**-point mesh.

the time to solution significantly without any significant improvement on the total energy.

5.1.2. ADMMk

For the verification of the ADMM implementation with **k**-point support with HFX we are limiting ourselves to the h-BN system. As shown in Figure 5.6, the energy converges continuously with the increased size of the ADMM basis up to an agreement of $-0.035\,\text{E}_\text{h}$ with *admm-tz2p*. We also show the speedup and total timing in Figure 5.6 and conclude that the time savings by using smaller ADMM basis sets are significant given the small improvement in total energy. Detailed analysis during calculations has shown that the simple parallelisation scheme occasionally leads to sub-optimal core utilisation, which becomes more important for larger basis sets. While the timing for the reference calculation at the full *TZVP* basis could therefore be improved, the speedups seen between the the different ADMM basis sets will remain.

Mesh	E_{tot} [a.u.]	Superlatt.	E_{tot} [a.u.]
3x3	-78.73841	1x1	-78.54089
6x6	-78.74149	2x2	-78.69401
9x9	-78.74149	3x3	-78.73841
12x12	-78.74149	4x4	-78.74085
15x15	-78.74149	5x5	-78.74136
18x18	-78.74149		
21x21	-78.74149		
24x24	-78.74149		
27x27	-78.74149		
30x30	-78.74149		

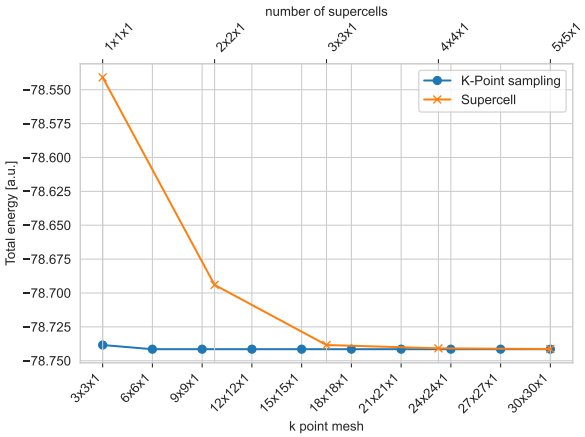
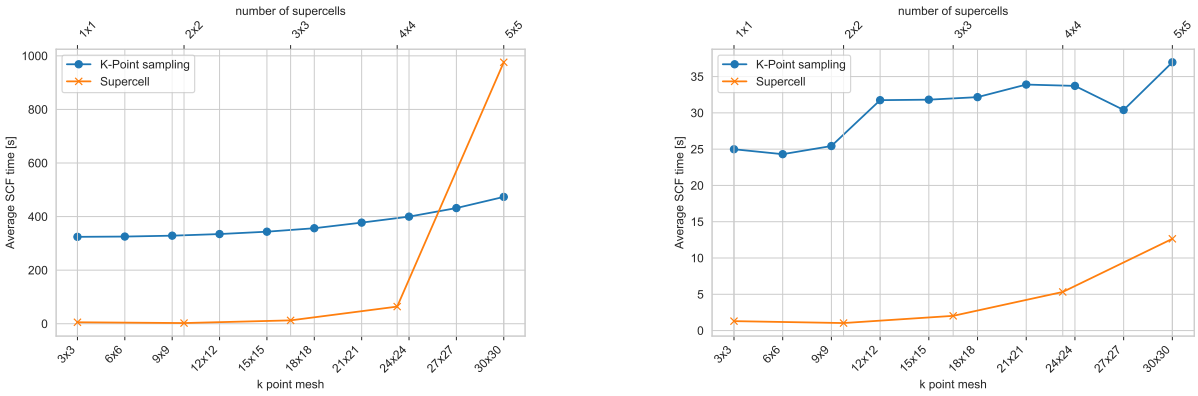


Figure 5.3.: Convergence of the total energy for ever finer **k**-point meshes, larger supercell configurations respectively, of h-BN. The *mesh* column refers to the MKP grid, while the *superlattice* refers to the superlattice configuration. The respective energies (per cell) are given next to it. There is no direct correspondence between superlattice configuration and **k**-point mesh.



(a) LiH **(b)** h-BN

Figure 5.4.: Average SCF timings for each system, with an integral cache of 128 GB.

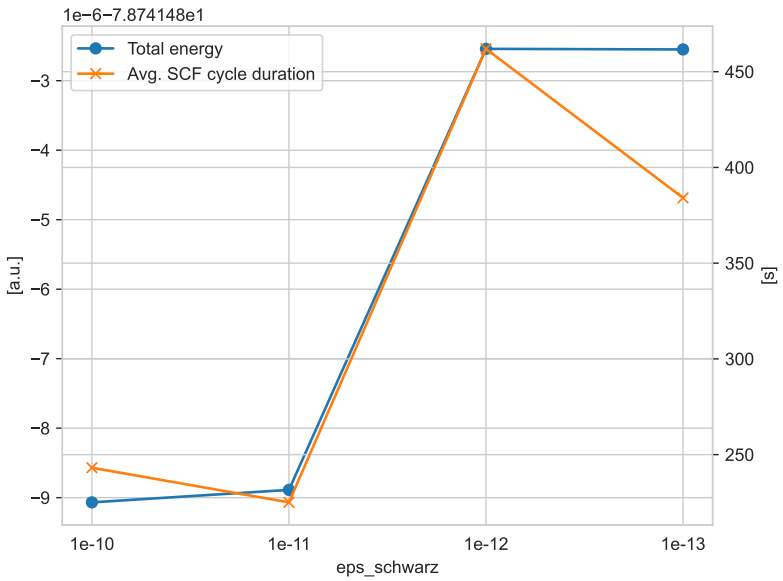


Figure 5.5.: Combined effect of the Schwarz-screening parameter on total energy and average SCF duration.

Basis Set Size	Energy [a.u.]	Avg. SCF [s]
admm-dz	−12.4270	46.8
admm-dzp	−12.4349	122.6
admm-tzp	−12.4768	300.3
admm-tz2p	−12.4784	1117
TZVP (ref)	−12.5133	43 207

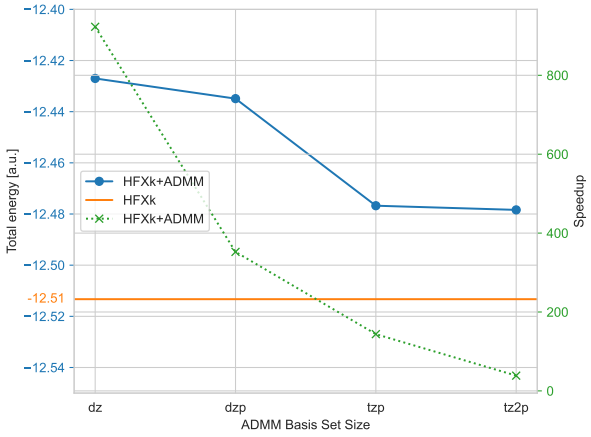


Figure 5.6.: Convergence of the total energy of h-BN with HFX, **k**-point mesh of 18x18, a cut-off of 2 Å and for different sizes of the respective ADMM *MOLOPT* basis set compared to the reference with a *MOLOPT TZVP* basis. Speedup is given in respect to the non-ADMM reference calculation.

5.2. Basis Set Convergence

As mentioned in Section 2.5 is the scaling of the number of primitive integrals to calculate with regard to the size of the basis set unfavourable. A demonstration of this can be seen in Figure 5.7, indicating that with increasing size of the MOLOPT basis, the total energy is indeed converging. At the same time is the time to solution (TTS) increasing, following the N^4 dependence. While TTS might not be the right quantity as it is dependent on algorithmic and implementation factors, as Chapter C illustrates, it is the measure the user is usually interested in. The respective total number of calculated ERIs is given in Table 5.1 for completeness. While caching the cartesian primitive integrals at TZVP level may still be feasible at single precision where it amounts to ≈10 TB, this becomes quickly unfeasible if higher precision and larger basis sets are needed or more complex systems are to be investigated.

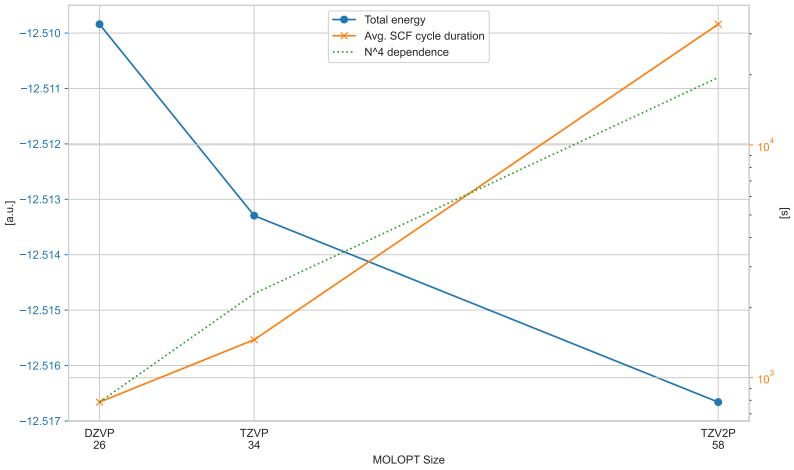


Figure 5.7.: Behaviour of the total energy for h-BN using HFX with the `MOLOPT-HYB-GTH` basis set family and its corresponding pseudopotential. On a dual AMD EPYC 7742 64-Core Processor, running with 256 OpenMP threads and an MKP grid of $18x18x1$. Slow convergence of the total energy is indicated. The scaling of the average SCF duration follows the expected N^4 scaling, although with some minor deviation.

For a comparison of different sizes of MOLOPT-ADMM basis sets, we are again using the Δ-test metric [101]. As an example for a hybrid functional, we selected *HSE06*, an improved parametrisation of the original Heyd-Scuseria-Ernzerhof (HSE) screened hybrid functional [144]. It is a range separated functional of the form

$$\begin{aligned} E_{\mathrm{xc}}^{\mathrm{HSE}} = a E_{\mathrm{x}}^{\mathrm{HF,SR}}(\omega) + (1-a) E_{\mathrm{x}}^{\mathrm{PBE,SR}}(\omega) \\ + E_{\mathrm{x}}^{\mathrm{PBE,LR}}(\omega) + E_{\mathrm{c}}^{\mathrm{PBE}}, \end{aligned} \tag{5.8}$$

where *SR* and *LR* refer to the short- and long-range components of the the *PBE* exchange functional, based on the partition of the Coulomb potential by

MOLOPT	Cartesian	Spherical
DZVP	3.04×10^{12}	6.26×10^{11}
TZVP	3.10×10^{12}	1.83×10^{12}
TZV2P	9.18×10^{13}	2.64×10^{13}

Table 5.1.: Total number of calculated cartesian primitive ERIs, resp. spherical ERIs for different sizes of the MOLOPT basis for h-BN.

$$\frac{1}{r} = \underbrace{\frac{1-\operatorname{erf}(\omega r)}{r}}_{\text{SR}} + \underbrace{\frac{\operatorname{erf}(\omega r)}{r}}_{\text{LR}}. \tag{5.9}$$

The system is again h-BN, and as the main orbital basis set we are using the MOLOPT TZVP, to account for the observation that hybrid functionals require basis sets at least at the triple-ζ (or larger) quality level [145, 146].

The results for this experiment can be seen in Figure 5.8. We have used the MOLOPT TZVP as the main basis for all calculations, with varying ADMM basis set sizes. The truncation radius was adjusted to the scaling of the volume.

The total energy converges towards the non-ADMM case with a delta of $0.0208\,\mathrm{E_h}$ with *admm-dz* down to $0.0089\,\mathrm{E_h}$ at *adm-tzp*. The equilibrium volume converges in a similar manner with a final Δ-value of 5.2 eV for the complete system.

For all the series not denoted with `corr` we used the *PBE* Exchange functional adapted to the truncated coulomb operator and its radius, while the `corr` series employs a full *PBE* Exchange operator in the ADMM correction term

$$\begin{aligned} E_x^{\mathrm{HFX}}[P] &= E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{HFX}}[P] - E_x^{\mathrm{HFX}}[\hat{P}]\right) \\ &\approx E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{DFT}}[\rho] - E_x^{\mathrm{DFT}}[\hat{\rho}]\right). \end{aligned} \tag{5.10}$$

Using the full *PBE* exchange operator improves the energy towards the non-ADMM reference, and better than the *admm-dzp*. It also increased the average SCF time significantly, but still remained below that of *admm-dzp*.

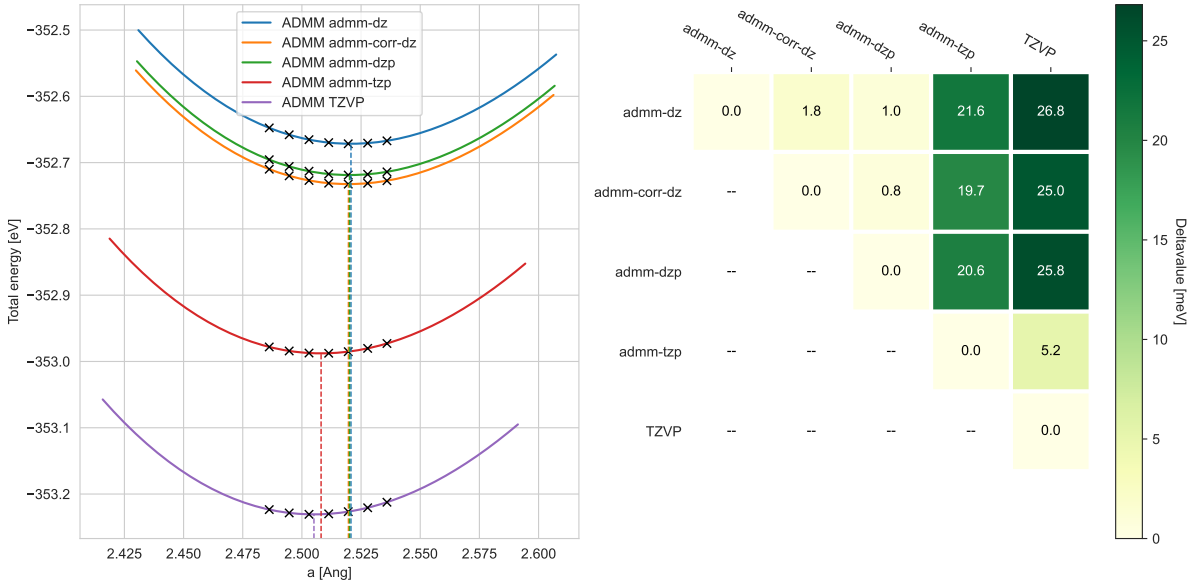


Figure 5.8.: Energy-volume curves fitted to the Birch-Murnaghan equation of states with scalings of $0.98, 0.99, \ldots, 1.04$ of the volume corresponding to the lattice parameter $a = 2.503$ Å. The Δ-values are given for the complete system rather than normalised for a single atom as in Section 4.3. The table shows the Δ-value between each of the ADMM basis sets (with the MOLOPT TZVP as primary basis), resp. between the ADMM basis sets and the non-ADMM reference with the TZVP basis set.

5.3. Conclusion

We have shown an implementation of the Hartree-Fock Exchange with **k**-point support for the Auxiliary Density Matrix Method, with verification using the LiH and h-BN systems. The comparison has shown a well-defined behaviour in terms of convergence with reasonable cutoffs for the Truncated Coulomb operator with a mix of direct density mixing and DIIS in the SCF iterations. Significant instabilities with the Truncated Coulomb (TC) operator as mentioned in Irmler, Burow, and Pauly [147] have not been observed. The ADMM implementation shows clear advantage in terms of number of ERI to calculate and therefore timing, compared to the reference HFX **k**-point implementation. An advantage in timing of the **k**-point-enabled implementation could also be observed in terms of convergence for sampling of the reciprocal space when compared to the Γ-point supercell approach. Although it must be stressed that this advantage occurs in the limited setup of a primitive cell and does not necessarily apply to cases where a supercell or a mixed approach is needed, such as adsorption and surface effect studies. Possible improvements to this reference implementation could be made in the following areas:

- Quantization and compression: the usual avenue of accelerating HFX calculations is the separation of the primitive integrals and the contraction with the density matrix, allowing to cache the result of the ERI calculation between SCF iterations. While it was still possible to implement this technique for the periodic implementation by Guidon, Hutter, and VandeVondele [148], the N^2 dependency in terms of neighbor cells makes this more challenging for the **k**-point implementation. Nonetheless, if a reduction of precision (Quantization) can be applied, together with improved screening

methods (possibly on the basis of density matrix for extended systems), newer memory sharing paradigms like disaggregated and Fabric-attached memory could be applied to successfully continue the implementation of caches.

- Using the occupation: approaches like the *Adaptively Compressed Exchange operator* by Lin [149] but also the *RI-occ-K* approach successfully exploit locality and reduce the computational complexity by limiting the calculation to occupied orbitals.
- Resolution of Identity approaches: the RI approaches already present within CP2K could be extended towards **k**-point-enabled HFX (with or without ADMM). As recent work shows [126, 127, 131] is the reduction of the ERI to a 3- or 2-center quantity a lucrative approach. While the choice of an additional basis may add an additional set of parameters which must be studied and controlled for well-defined behaviour, this basis must not necessarily follow the same construction scheme (safe for some properties) as a *full* basis sets but can be generated on-the-fly. We note here that the implementation of such an RI scheme is likely to be done in separate algorithm as it changes the algebraic form of the overall schema and provides different possibilities in terms of parallelisation. As shown in the aforementioned references a direct analysis and optimisation of the required quantities for specific iteration schemes proof fruitful as well.

There are also more recent proposals on accelerating HFX calculations, such as combining a range-separated operator with the advantages of the dual-nature of the GPW approach itself [150].

Overall this shows that despite the decades passed since the first implementations HFX in computer codes, the field is still very active and the final optimisation has not yet been reached. We think that this reference implementation serves as another stepping stone towards more efficient HFX implementations, in particular within CP2K.

6. Summary and Outlook

6.1. Summary

This work was divided into three parts, all evolving around the use of Gaussian-type Orbital (GTO) as the basis to solving the time-independent Schrödinger equation within electronic structure theory using the software package CP2K.

In the first part we have illustrated the generation of the revised MOLOPT pseudopotential and basis set family, the latter now being available as an all-electron basis set. From there we have analysed the performance of both pseudopotential and basis sets in a molecular setting, comparing the accuracy of both code and parameters against well-known references. In doing so, we were able to confirm once more the well-behaved nature of the MOLOPT basis set towards the complete basis set limit. This behaviour has been shown to hold consistently across the three functionals PBE, TPSS and PBE0, as well as with ADMM. With the meta-GGA TPSS we could also show the transferability of the basis set to other functionals in the same rung.

Since software packages like CP2K excel at the intersection of quantum chemistry and solid state physics, evaluating the performance of both the software package and its underlying simulation parameters in a condensed matter setting is crucial. To that end, we employed the Δ-test and not only compared against literature values, but also established a full workflow to obtain reference values from within CP2K, with help of the SIRIUS library for implementation of the Full-Potential Linearized Augmented Plane Wave method (FP-LAPW) and plane wave (PW) methods. This we have demonstrated for the PBEsol functional. We could again observe well-defined behaviour for most elemental crystals with some minor exceptions for elements with more difficult electronic structure (magnetisation) or some heavier elements. In particular with the newly generated all-electron basis set together with Gaussian and Augmented Planewave (GAPW), we were able to obtain results in very good agreement with other state-of-the-art codes.

These sort of benchmarks require a considerable amount of calculations, as well as proper tracking of the parameters such as the pseudopotentials and basis sets under test, as these may be revised during the testing. Hence, a significant effort was made towards automation and therefore counted as a separate part in this thesis. Molecular calculations were far easier to converge and computationally not as demanding, hence they could be run directly on local machines and thus required far less automation. Nonetheless, in the spirit of improving the software ecosystem around CP2K, we developed and released extendable open-source parsing libraries for both CP2K and G16. Automating the more complex solid-state benchmarks took more effort. While at the beginning of the project, the AiiDA framework was still in early development and lacked tracking functionalities for different versions of GTO basis sets, a bespoke workflow manager was written. To make the transition to AiiDA easier, its architecture was again based on separately released open-source libraries

to handle CP2K input and output. This workflow manager was used for the initial batch of Δ-test results, while the latter were obtained using an AiiDA workflow with additional plugins. We compared the different approaches of running benchmarks, from simple ad-hoc scripting over custom to adaption of a larger existing workflow manager. In that comparison it has become clear that established workflow managers hold a clear advantage when it comes to standardised tasks in a more complex workflow, such as data transport, scheduling, error recovery. On the contrary, the process of adoption can grow more complex as a consequence of the levels of abstraction and the idiosyncratic design of any given workflow management tool. We have also seen that the availability of well-designed code interface libraries, such as the *cp2k-input-tools* and *cp2k-output-tools* developed as part of this thesis, make it significantly easier to transition between the different approaches and to retain a high grade of automation for directly code related tasks (such as parsing or input generation). Despite appropriate tooling, endeavours such as large scale benchmarking still retain some of their complexity. Both the setup of a workflow management system, but also the full automation still require considerable amount of work and respective know-how. Based on our work, it should be possible to create a simple and fully automated validation suite based on small molecular benchmarks, such as the Small Molecules Database (SMDB) with additional information. More intricate benchmarks necessitating greater orchestration or compute resources still elude simple usability and execution. Several of the improvements as well as the libraries and tools developed as either effort are now directly available to the community, either as plugins for AiiDA or within the software itself [2, 3].

To be able to extend the aforementioned solid state benchmarks towards Hybrid functionals such as PBE0, a periodic **k**-point Hartree-Fock Exchange (HFX) implementation is necessary. As has been shown is the scaling of a naïve implementation in the order of N^4 of the system size. Even though this can be improved by appropriate screening of the integrals, many problems still remain out of reach. As a remedy, the existing Auxiliary Density Matrix Method (ADMM) implementation within CP2K was extended to support periodic calculations with **k**-points as well, and validated with systems commonly found in literature. This concluded the third and final part of this thesis.

6.2. Outlook

6.2.1. Benchmarking

The need for validating results in terms of accuracy and agreement amongst different electronic structure codes will remain high. For GTO based codes, this does not only include the implementation and any pseudopotential (PP) involved, but also the basis set employed in such calculations. While we have shown that the MOLOPT basis sets are qualitatively high performing basis sets for many different use cases able to reproduce various quantities accurately enough, further work may be needed to extend this in different directions. A next step towards automation of such benchmarks has already been taken with the *aiida-common-workflows*

package [151] which has been used for a new verification study encompassing the whole periodic table from Z=1 to 96 and characterising 10 prototypical cubic compounds for each element (4 unaries and 6 oxides), spanning a wide range of coordination numbers and oxidation states. This resulted in a vast collection of equations of state, verified between all-electron and PP-based approaches [3]. In doing so, this also expands the landscape towards system beyond simple solids. Another direction would be to try for different functionals. While this is often explored in the context of new basis sets or computational methods, it is rarely explored across different codes and approaches (PW, GTO). Finally, in particular towards the basis set limits, many codes exhibit unclear instabilities when it comes to convergence of the self-consistent field (SCF). Monitoring of specific observables during a calculation run could proof very useful to avoid losing compute time in such cases, in particular for high-throughput calculations. The difficulty lies in identifying which quantities to monitor and detecting anomalies reliably.

6.2.2. ADMM-HFXk Implementation

The primary intention of the implementation given in this work was that of a reference code, in the sense of correctness. As such it is thought to serve as a stepping stone to improved-scaling methods such as the mentioned Resolution of Identity (RI) approaches. Since these improvements will always have to start at the level of the **k**-point enabled HFX implementation itself, the combination with ADMM will most likely automatically benefit from the improvements as well. As outlined in the *occ-RI-K* approach, optimising for SCF methods which require only a subset of the full HFX matrix has some potential as well, in particular with larger or more diffuse basis sets such as the MOLOPT.

The reference implementation itself could also be improved. One such area of improvement could be the screening, for which we relied mostly only on the classical Schwarz-screening. Extending the integral estimates to Multipole-methods [141, 143, 152–154] as in the work of Irmler [155] has the potential of reducing the number of Electron Repulsion Integral (ERI) to calculate further by providing stricter bounds. Algorithmic improvements could be considered as well. Another avenue could be new computer architecture techniques of pooling memory between nodes using fabric-attached memory, which greatly enhances the amount of memory available to each node. Such an architecture would have the potential to simplify the otherwise explicit distribution of matrices and permit to cache the full list of ERI (whether obtained in an RI-approximation or with the reference four-centre approach) in a transparent fashion. Furthermore, with the data cached on memory-like storage, accessing it in a variable ordering without performance penalty as for example required by post-Hartree-Fock methods such as MP2 would be directly supported.

6.2.3. Tooling

Finally, the development of external integration libraries such as the *cp2k-input-tools* and *cp2k-output-tools* for CP2K has already proven useful. By

collecting common tasks, they can greatly accelerate the development of simple custom workflows, which despite the existence of larger and all-encompassing frameworks such as AiiDA still have their place, as we have seen in this thesis. While a proper understanding of the underlying physics and methods of electronic structure theory will always be necessary to setup and run calculations and interpret their results, the commoditisation of High-performance computing (HPC) makes software such as CP2K more accessible in general. By providing corresponding tooling around it, such as a Language Server Protocol implementation, which provides live feedback in a text editor when crafting the input for a calculation, adaption could be accelerated even further.

Small Molecules Database Calculation Details

A.

The full data set for our benchmark with the SMDB can be found in Hutter and Müller [114] and the original geometries in the appendix to Weigend and Ahlrichs [53].

The protocol is run in four steps

1. Gaussian reference calculation, with `OPT` and `FREQ` keywords to optimise the geometry and obtain the vibrational frequencies.
2. Geometry optimisation with CP2K.
3. Properties calculation with the wave function and positions from the previous step as initial guess, using the `LINEAR_RESPONSE` run type.
4. Vibrational Analysis with the `VIBRATIONAL_ANALYSIS` run type, again with initial guess for wave function and positions obtained from the previous steps.

In Listings 1 and 3 we are reproducing exemplary G16 and CP2K input configurations (for the latter only for the geometry optimisation).

The file included as `kinddef` contains CP2K basis set and PP definitions for all elements, for the basis set size and type under test. The `xcdef` contains the definition of the functional to be used for a specific test run, here we reproduce the one used for the ADMM as it the most complex and complete one in Listing 2. The ADMM input illustrates the use of *ADMM2* with the full *PBE* exchange functional.

Listing 1: CP2K input for the geometry optimisation of the Ag_2 structure from the SMDB

```
&GLOBAL
   PROJECT  ag2
   PRINT_LEVEL low
   RUN_TYPE GEO_OPT
   PREFERRED_DIAG_LIBRARY SL
&END GLOBAL
&MOTION
   &GEO_OPT
      MAX_ITER 50
   &END GEO_OPT
&END MOTION
&FORCE_EVAL
   METHOD Quickstep
   &DFT
      CHARGE            0
      MULTIPLICITY           1
      BASIS_SET_FILE_NAME BASIS_MOLOPT_UZH
      POTENTIAL_FILE_NAME POTENTIAL_UZH
      &MGRID
          CUTOFF     720
          REL_CUTOFF  60
      &END MGRID
      &QS
         EPS_DEFAULT 1.E-14
      &END QS
      &SCF
         SCF_GUESS ATOMIC
         NOTCONV_STOPALL
         MAX_SCF 15
         EPS_SCF 1.E-7
         &OT
            PRECONDITIONER FULL_ALL
            MINIMIZER DIIS
         &END OT
         &OUTER_SCF
           MAX_SCF 5
           EPS_SCF 1.E-7
         &END OUTER_SCF
      &END SCF
      &POISSON
         PERIODIC NONE
         POISSON_SOLVER MT
      &END POISSON
@include xcdef
   &END DFT
   &SUBSYS
      &CELL
        ABC      14.00000000    14.00000000    14.00000000
         PERIODIC NONE
      &END CELL
      &COORD
Ag          8.50000000     8.50000000     7.21259135
Ag          8.50000000     8.50000000     9.78740865
      &END COORD
      &TOPOLOGY
        &CENTER_COORDINATES
        &END CENTER_COORDINATES
      &END TOPOLOGY
      &PRINT
         &ATOMIC_COORDINATES
         &END ATOMIC_COORDINATES
      &END PRINT
@include kinddef
   &END SUBSYS
&END FORCE_EVAL
```

```
    BASIS_SET_FILE_NAME BASIS_ADMM_UZH
    &AUXILIARY_DENSITY_MATRIX_METHOD
      METHOD BASIS_PROJECTION
      ADMM_PURIFICATION_METHOD NONE
      EXCH_CORRECTION_FUNC PBEX
    &END

    &XC
      &XC_FUNCTIONAL
         &GGA_C_PBE
         &END GGA_C_PBE
         &GGA_X_PBE
            SCALE 0.75
         &END GGA_X_PBE
      &END XC_FUNCTIONAL
      &HF
         FRACTION 0.25
         &HF_INFO ON
         &END
         &SCREENING
           EPS_SCHWARZ 1.0E-10
         &END
         &MEMORY
            MAX_MEMORY 4000
         &END
      &END HF
    &END XC
```

Listing 2: A CP2K input segment for the ADMM calculations for the SMDB benchmark.

```
%mem=8GB
%NProcShare=8
#N RPBEPBE/Def2QZVPP scf=tight Int(Grid=ultrafine) OPT FREQ

Mol job 1

0 1
Ag       8.50000000       8.50000000       7.21259135
Ag       8.50000000       8.50000000       9.78740865
```

Listing 3: G16 input for the Ag_2 structure from the SMDB

B. Solid State Calculation Details

We have been using single point energy calculations at seven volume scales. Non-convergent single point energies were ignored if a fit could be obtained with the remaining points. The fitted coefficients can be found in Table B.2 and Table B.1, unphysical values were filtered out in a later stage.

In Listing 4 we have reproduced an exemplary input for the Δ-test with CP2K for Ag with the DZVP MOLOPT basis set and a corresponding GTH PP, run with Gaussian and Plane Waves Method (GPW). Inputs for other basis sets have been prepared in the same manner and with similar tolerances, with specific parameters as given in the main text. The raw data set can be found in Müller [108].

Table B.1.: Fitted Birch-Murnaghan coefficients for all PBEsol calculations.

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
H	LAPW			-15.8832	17.3904	10.2642	2.6260
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-15.8215	17.5038	10.2292	2.7158
			pob-DZVP-rev2	-15.8218	16.9765	11.2995	2.6064
			pob-TZVP-rev2	-15.8504	17.1372	10.8926	2.6248
			pob-TZVPP	-15.8577	17.1932	10.7337	2.6206
			TZVPP-MOLOPT-PBE-ae	-15.8745	17.4498	10.3691	2.6673
			QZVPP-MOLOPT-PBE-ae	-15.8756	17.2370	10.6106	2.6528
	PW	GTH-PBE-q1		-15.8606	17.5328	10.1536	2.6781
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-15.8600	17.3228	10.5428	2.6588
			TZVP-MOLOPT-PBE-GTH-q1	-15.8609	17.3825	10.4206	2.6382
			TZV2P-MOLOPT-PBE-GTH-q1	-15.8724	17.3135	10.6523	2.6499
He	LAPW			-78.7356	17.9370	1.0545	-8.2822
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-78.3849	17.4150	0.8403	5.7855
			TZVPP-MOLOPT-PBE-ae	-78.6670	17.0639	0.9659	6.3705
			QZVPP-MOLOPT-PBE-ae	-78.5936	17.4560	0.9606	6.0447
	PW	GTH-PBE-q2		-78.5467	17.9247	0.6953	2.5338
	GPW	GTH-PBE-q2	DZVP-MOLOPT-PBE-GTH-q2	-78.6792	17.9204	0.8878	6.4982
			TZVP-MOLOPT-PBE-GTH-q2	-78.6793	17.9139	0.8886	6.4985
			TZV2P-MOLOPT-PBE-GTH-q2	-78.6793	17.8734	0.8892	6.4816
Li	LAPW			-204.6818	20.2698	13.6981	2.5288
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-204.4918	20.4537	13.3782	3.3446
			pob-DZVP-rev2	-204.3497	17.0080	29.6925	-2.9236
			pob-TZVP-rev2	-204.5433	20.9549	16.1886	2.4483
			pob-TZVPP	-204.3579	17.8019	22.1851	2.8218
			TZVPP-MOLOPT-PBE-ae	-204.6549	20.2154	13.8489	3.3468
			QZVPP-MOLOPT-PBE-ae	-204.6583	20.2252	13.8801	3.3109
	PW	GTH-PBE-q1		-7.1285	18.9798	13.7378	3.1565
		GTH-PBE-q3		-202.5001	20.3413	14.1607	1.2658
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-7.0617	20.1839	13.4990	2.9430
			TZVP-MOLOPT-PBE-GTH-q1	-7.0620	20.1415	13.4348	3.0185
			TZV2P-MOLOPT-PBE-GTH-q1	-7.0834	19.4188	13.0134	2.8825
		GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-203.3115	20.8098	13.8358	3.4133
			TZVP-MOLOPT-PBE-GTH-q3	-203.3249	20.3104	13.8473	3.3894
			TZV2P-MOLOPT-PBE-GTH-q3	-203.3372	20.2359	13.8296	3.3715
Be	LAPW			-401.9007	7.9159	122.9020	3.2637
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-401.5361	7.8504	130.2762	3.3814
			pob-DZVP-rev2				
			pob-TZVP-rev2	-398.9713	7.4612	96.2436	2.8771
			pob-TZVPP				
			TZVPP-MOLOPT-PBE-ae	-401.7877	7.9288	123.8557	3.3172
			QZVPP-MOLOPT-PBE-ae	-401.8009	7.9004	124.8591	3.2406
	PW	GTH-PBE-q2		-30.9002	7.7154	117.2288	3.1137
		GTH-PBE-q4		-396.2395	7.9407	123.6585	5.1810
	GPW	GTH-PBE-q2	DZVP-MOLOPT-PBE-GTH-q2	-30.7234	7.6459	127.0702	3.1539
			TZVP-MOLOPT-PBE-GTH-q2	-30.7707	7.7486	121.2237	3.1072
			TZV2P-MOLOPT-PBE-GTH-q2	-30.8910	7.7063	118.1171	3.1517
		GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-398.9432	7.9481	128.7689	3.2860
			TZVP-MOLOPT-PBE-GTH-q4	-398.9580	7.9506	124.2816	3.2812
			TZV2P-MOLOPT-PBE-GTH-q4	-399.0773	7.9192	120.2699	3.1987
B	LAPW			-676.1914	7.2389	237.1675	3.5223
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-675.5680	7.2551	240.9741	3.4824
			pob-DZVP-rev2	-675.2857	7.0780	263.7309	3.3132
			pob-TZVP-rev2	-675.8314	7.0259	275.5157	3.1902
			TZVPP-MOLOPT-PBE-ae	-675.9011	7.1854	246.1147	3.4358
			QZVPP-MOLOPT-PBE-ae	-675.9672	7.2292	236.6390	3.4371
	PW	GTH-PBE-q3		-77.1467	7.2061	235.4008	3.4577
	GPW	GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-77.1033	7.2057	236.2925	3.4337
			TZVP-MOLOPT-PBE-GTH-q3	-77.1130	7.2022	235.9625	3.3985
			TZV2P-MOLOPT-PBE-GTH-q3				
C	LAPW			-1036.8699	11.6309	208.6473	3.5853
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-1035.2359	11.7654	206.5785	3.5702

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			pob-DZVP-rev2	-1035.1413	11.7338	215.3833	3.5070
			pob-TZVP-rev2	-1036.1545	11.5334	214.3296	3.5387
			TZVPP-MOLOPT-PBE-ae	-1036.2264	11.5912	214.7666	3.5411
			QZVPP-MOLOPT-PBE-ae	-1036.3725	11.5938	210.8468	3.5728
	PW	GTH-PBE-q4		-155.0295	11.6332	208.3696	3.6104
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-154.9481	11.6651	207.9134	3.5640
			TZVP-MOLOPT-PBE-GTH-q4	-154.9566	11.6789	206.9634	3.5502
			TZV2P-MOLOPT-PBE-GTH-q4	-154.9931	11.6503	206.3905	3.5524
N	LAPW			-1490.1516	28.7916	53.9455	3.6231
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-1486.6470	28.8358	57.1474	3.6915
			pob-DZVP-rev2	-1487.1295	30.4175	52.6102	3.6902
			pob-TZVP-rev2	-1488.6195	29.9396	47.8866	3.8477
			TZVPP-MOLOPT-PBE-ae	-1488.9546	29.0064	54.0176	3.7273
			QZVPP-MOLOPT-PBE-ae	-1489.2352	28.7228	54.4347	3.7044
	PW	GTH-PBE-q5		-270.8419	28.8465	53.6068	3.5190
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-270.7173	30.0906	53.0153	3.7058
			TZVP-MOLOPT-PBE-GTH-q5	-270.8029	29.2220	52.4930	3.6520
			TZV2P-MOLOPT-PBE-GTH-q5	-270.8311	29.0935	52.5927	3.6590
O	LAPW			-2045.8897	18.4622	51.9055	3.9102
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-2040.2986	19.2874	45.8991	3.8356
			pob-DZVP-rev2	-2041.6623	19.3064	50.7557	3.9718
			pob-TZVP-rev2	-2043.6676	21.1491	38.8501	3.6953
			pob-TZVPP	-2044.0588	18.9558	49.4192	3.9582
			TZVPP-MOLOPT-PBE-ae	-2043.6961	18.1665	56.0228	3.8381
			QZVPP-MOLOPT-PBE-ae	-2044.3144	18.6712	50.7753	3.7550
	PW	GTH-PBE-q6					
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-434.3855	19.4037	50.3916	3.8928
			TZVP-MOLOPT-PBE-GTH-q6	-434.4011	19.2592	48.7263	3.8354
			TZV2P-MOLOPT-PBE-GTH-q6	-434.4447	19.1079	49.0029	3.8161
F	LAPW			-2716.0340	19.1304	34.2757	3.9263
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-2705.9806	19.6422	33.2514	3.9768
			pob-DZVP-rev2	-2709.8371	19.2131	32.4534	4.1914
			pob-TZVP-rev2	-2713.1254	20.0473	33.2274	3.9851
			pob-TZVPP				
			TZVPP-MOLOPT-PBE-ae	-2712.1296	19.0564	31.4353	-0.0014
			QZVPP-MOLOPT-PBE-ae	-2713.3961	19.2264	34.8659	4.1113
	PW	GTH-PBE-q7		-649.5895	18.7676	30.2494	0.4722
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-657.4806	19.5405	34.6492	4.1183
			TZVP-MOLOPT-PBE-GTH-q7	-657.4835	19.5681	34.3740	4.1067
			TZV2P-MOLOPT-PBE-GTH-q7	-657.5103	19.4914	34.0509	4.0355
Ne	LAPW			-3510.6585	25.4049	12.7550	-24.0795
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-3503.6509	24.3358	1.4059	5.3171
			TZVPP-MOLOPT-PBE-ae	-3506.4273	22.8442	0.8975	-1.5239
			QZVPP-MOLOPT-PBE-ae	-3506.5521	23.8946	1.4581	6.7434
	PW	GTH-PBE-q8		-944.9968	23.7488	5.6261	22.7700
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-950.1891	23.6746	1.2345	8.3563
			TZVP-MOLOPT-PBE-GTH-q8	-950.1917	23.6994	1.2591	8.1334
			TZV2P-MOLOPT-PBE-GTH-q8	-950.1918	23.6668	1.2776	8.2233
Na	LAPW			-4420.0773	37.2015	7.9292	1.8615
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-4411.3506	37.8887	7.8512	3.6380
			pob-DZVP-rev2	-4412.0485	30.1301	22.5778	1.3088
			pob-TZVP-rev2	-4413.6100	32.3587	13.8176	1.1454
			pob-TZVPP	-4413.6488	32.1241	14.2638	1.1356
			TZVPP-MOLOPT-PBE-ae	-4413.6908	36.7999	7.9263	3.8570
			QZVPP-MOLOPT-PBE-ae	-4413.9719	36.2457	8.4849	3.6817
	PW	GTH-PBE-q1		-6.1950	37.9716	7.2285	3.6726
		GTH-PBE-q9		-1254.8853	37.3276	7.3523	15.3515
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-6.1735	38.6459	7.0372	3.6326
			TZVP-MOLOPT-PBE-GTH-q1	-6.1800	38.7735	7.2287	3.4908
			TZV2P-MOLOPT-PBE-GTH-q1	-6.1921	37.9859	7.2299	3.6448
		GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-1304.0790	38.9538	6.7858	3.5000
			TZVP-MOLOPT-PBE-GTH-q9	-1304.0817	38.6250	6.7616	3.5544
			TZV2P-MOLOPT-PBE-GTH-q9	-1304.0869	38.4423	6.6965	3.5513

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
Mg	LAPW			-5451.3968	22.9660	36.0507	4.0055
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-5440.4647	22.8030	36.9186	4.0303
			pob-DZVP-rev2	-5439.2867	17.8998	220.2267	7.2914
			pob-TZVP-rev2	-5440.4657	13.6408	48.5830	-14.6618
			TZVPP-MOLOPT-PBE-ae	-5442.2670	22.9949	35.9941	4.0405
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q2		-23.9916	23.2149	34.7533	3.9538
		GTH-PBE-q10		-1637.8867	24.3920	48.9855	-4.1965
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-1723.0894	23.6044	33.5330	4.0434
			TZVP-MOLOPT-PBE-GTH-q10	-1723.1057	23.6727	32.9018	3.9795
			TZV2P-MOLOPT-PBE-GTH-q10	-1723.2320	23.1172	34.4829	4.1657
Al	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-6592.3451	15.9798	83.5647	5.1990
			pob-DZVP-rev2	-6592.6524	16.9295	73.1317	4.6810
			pob-TZVP-rev2	-6593.4372	10.9805	40.2847	-26.1470
			TZVPP-MOLOPT-PBE-ae	-6594.7179	16.4696	78.8214	4.6754
			QZVPP-MOLOPT-PBE-ae	-6594.9780	16.4603	78.8063	4.7080
	PW	GTH-PBE-q3		-56.4348	16.4744	76.7742	4.5787
	GPW	GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-56.3601	16.4088	79.1315	4.6433
			TZVP-MOLOPT-PBE-GTH-q3	-56.3925	16.4269	78.9152	4.6809
			TZV2P-MOLOPT-PBE-GTH-q3	-56.4132	16.4388	78.4769	4.6959
Si	LAPW			-7892.2879	20.5224	86.4086	5.2519
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-7872.0034	20.1227	99.9992	4.1453
			pob-DZVP-rev2	-7872.1517	20.6279	91.3889	4.1109
			pob-TZVP-rev2	-7874.0270	19.7811	91.2406	4.6695
			pob-TZVPP	-7874.0889	19.8313	91.5886	4.5303
			TZVPP-MOLOPT-PBE-ae	-7874.6712	20.5116	89.1866	4.2638
			QZVPP-MOLOPT-PBE-ae	-7874.9390	20.3381	91.1498	4.2942
	PW	GTH-PBE-q4		-107.1747	20.3573	88.2776	4.2780
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-107.0801	20.4052	87.8711	4.2617
			TZVP-MOLOPT-PBE-GTH-q4	-107.0934	20.4246	87.1143	4.2758
			TZV2P-MOLOPT-PBE-GTH-q4	-107.1299	20.3384	88.1904	4.2975
P	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-9282.2740	21.5018	68.3008	4.4405
			pob-DZVP-rev2	-9282.1720	20.7033	87.4301	4.2061
			pob-TZVP-rev2	-9285.0416	21.7842	69.7421	4.2946
			TZVPP-MOLOPT-PBE-ae	-9285.2366	21.5551	68.1752	4.3025
			QZVPP-MOLOPT-PBE-ae	-9285.6387	21.3973	68.1825	4.3788
	PW	GTH-PBE-q5		-179.4299	21.2659	68.6570	4.1869
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-179.2151	21.6443	64.4372	4.3428
			TZVP-MOLOPT-PBE-GTH-q5	-179.2469	21.5854	65.2551	4.3119
			TZV2P-MOLOPT-PBE-GTH-q5	-179.3532	21.3362	67.5744	4.3244
S	LAPW			-10862.1886	17.1580	77.7122	16.7882
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-10826.9326	17.3399	85.4391	4.3051
			pob-DZVP-rev2	-10827.0847	18.9541	80.3028	4.1192
			pob-TZVP-rev2	-10827.8508	16.7690	105.9676	4.3706
			pob-TZVPP	-10828.2880	16.7244	109.3671	4.6993
			TZVPP-MOLOPT-PBE-ae	-10830.9052	17.2326	83.8572	4.0912
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q6		-277.4321	17.0411	83.4896	4.1901
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-277.2984	17.2578	83.7300	4.1460
			TZVP-MOLOPT-PBE-GTH-q6	-277.3067	17.2050	83.0642	4.1502
			TZV2P-MOLOPT-PBE-GTH-q6	-277.3644	17.0664	84.7653	4.0746
Cl	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-12512.8428	41.3032	17.7753	4.1451
			pob-DZVP-rev2	-12512.2673	40.9315	25.7571	-0.0162
			pob-TZVP-rev2	-12516.8495	44.2736	17.0903	4.4545
			pob-TZVPP	-12516.9784	43.4143	19.1358	4.0264
			TZVPP-MOLOPT-PBE-ae	-12517.5330	38.8550	19.4832	4.3934
			QZVPP-MOLOPT-PBE-ae	-12517.9438	38.8660	18.9442	4.3820
	PW	GTH-PBE-q7		-407.4189	38.2670	19.3079	4.3728
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-407.2015	39.2855	19.4587	4.4362
			TZVP-MOLOPT-PBE-GTH-q7	-407.2066	39.2340	19.4294	4.4545

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			TZV2P-MOLOPT-PBE-GTH-q7	-407.2847	38.5691	19.7077	4.3912
Ar	LAPW			-14401.4009	53.5168	0.7277	3.5107
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-14345.6766	52.0692	0.5060	3.6099
			TZVPP-MOLOPT-PBE-ae	-14338.3649	45.4402	0.5358	-5.1931
			QZVPP-MOLOPT-PBE-ae	-14349.3944	51.1558	0.7212	3.3586
	PW	GTH-PBE-q8		-573.4247	52.1382	0.7340	8.9418
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-573.4101	49.8795	0.9045	7.9615
			TZVP-MOLOPT-PBE-GTH-q8	-573.4115	49.9480	0.8729	7.6659
			TZV2P-MOLOPT-PBE-GTH-q8	-573.4116	49.9592	0.8689	7.6769
K	LAPW			-16384.7174	72.2149	6.7689	28.5580
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-16316.8670	74.0597	3.6624	3.5182
			pob-DZVP-rev2	-16315.7124	46.6820	74.3901	6.6250
			pob-TZVP-rev2	-16318.5581	53.4494	37.1556	4.9363
			pob-TZVPP	-16318.7868	54.1243	35.9046	4.5592
			TZVPP-MOLOPT-PBE-ae	-16319.4228	74.2621	3.6394	3.5773
			QZVPP-MOLOPT-PBE-ae	-16319.6802	72.6539	3.8371	3.7254
	PW	GTH-PBE-q1		-5.0872	75.3082	3.2991	3.6513
		GTH-PBE-q9		-769.2741	73.4869	3.6629	4.7954
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-769.2143	75.7127	3.5828	3.4459
			TZVP-MOLOPT-PBE-GTH-q9	-769.2290	73.4891	3.6015	3.4745
			TZV2P-MOLOPT-PBE-GTH-q9	-769.2297	73.5015	3.5927	3.9296
Ca	LAPW			-18514.6121	42.2549	16.8036	2.8342
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-18430.2829	44.3617	17.4465	3.6403
			pob-DZVP-rev2	-18428.4917	34.3641	69.8131	2.3605
			pob-TZVP-rev2	-18430.1094	34.1258	79.2187	0.7484
			pob-TZVPP	-18418.9901	27.2111	118.1119	6.2843
			TZVPP-MOLOPT-PBE-ae	-18433.1229	42.0201	17.7952	3.2745
			QZVPP-MOLOPT-PBE-ae	-18433.3207	42.1094	17.6075	3.1773
	PW	GTH-PBE-q2		-19.7780	37.5599	16.4737	3.0679
		GTH-PBE-q10		-999.2612	42.8601	17.4123	3.9946
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-999.1915	44.0666	17.1717	3.3888
			TZVP-MOLOPT-PBE-GTH-q10	-999.2492	43.2273	17.6843	3.4175
			TZV2P-MOLOPT-PBE-GTH-q10	-999.2639	43.1298	17.4132	3.4277
Sc	LAPW			-24752.5438	3.9153	3182.3725	-241.5389
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-20691.8456	24.7021	57.3891	3.2285
			pob-DZVP-rev2	-20692.4037	20.9823	51.9912	-7.3226
			pob-TZVP-rev2	-20694.9945	21.0576	46.2838	-10.2562
			TZVPP-MOLOPT-PBE-ae	-20695.2005	24.6296	54.0749	3.3800
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q3		-46.7663	25.8714	50.1385	3.2352
		GTH-PBE-q11		-1267.1305	24.6206	47.4213	5.4860
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1269.1217	24.6620	55.4702	3.3440
			TZVP-MOLOPT-PBE-GTH-q11	-1269.1782	24.5810	54.3681	3.3965
			TZV2P-MOLOPT-PBE-GTH-q11	-1269.1831	24.5750	54.5076	3.3983
Ti	LAPW			-24631.7885	17.8752	6385.1130	14116.1892
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-23100.3653	16.0752	587.6793	16.6887
			pob-DZVP-rev2	-23107.8611	16.3902	140.0910	0.0773
			pob-TZVP-rev2	-23107.8852	12.8275	777.2526	4.4044
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q4		-93.6653	18.2366	104.6754	3.4649
		GTH-PBE-q12		-1576.8055	17.2868	109.3539	7.5673
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1580.0234	17.3979	114.5387	3.5100
			TZVP-MOLOPT-PBE-GTH-q12	-1580.0558	17.4071	111.6865	3.6114
			TZV2P-MOLOPT-PBE-GTH-q12	-1580.0605	17.4182	111.5357	3.5995
V	LAPW			-25832.4142	13.4720	181.5373	3.8876
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-25667.9085	13.8161	12801.9119	-28.3937
			pob-DZVP-rev2	-25672.3607	14.2021	4789.2105	-23.9172
			pob-TZVP-rev2	-25683.5768	12.6031	259.2206	2.3610
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q5		-164.9700	14.0344	163.4427	3.9376
		GTH-PBE-q13		-1935.4361	13.2350	176.1863	14.1838

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1940.2561	13.5200	183.9794	3.6972
			TZVP-MOLOPT-PBE-GTH-q13	-1940.2964	13.5171	181.6041	3.8582
			TZV2P-MOLOPT-PBE-GTH-q13	-1940.2990	13.5222	181.2874	3.8545
Cr	LAPW			-28596.2273	11.7494	193.0643	6.8255
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-28397.3115	12.2635	11635.2963	-15.3822
			pob-DZVP-rev2	-28399.8687	11.3779	61578.7764	28.6375
			pob-TZVP-rev2	-28414.3671	11.4605	46514.7704	33.8671
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q6					
		GTH-PBE-q14					
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-2321.6068	12.3673	148.6561	6.6212
			TZVP-MOLOPT-PBE-GTH-q14				
			TZV2P-MOLOPT-PBE-GTH-q14				
Mn	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-31277.8960	9.3826	75.5822	-5.2540
			pob-DZVP-rev2	-31291.2666	9.7572	114.3377	-105.8516
			pob-TZVP-rev2	-31316.5241	11.3354	3548.6184	37.3484
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q7					
		GTH-PBE-q15					
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2816.5167	12.2273	116.7321	5.5686
			TZVP-MOLOPT-PBE-GTH-q15	-2816.5709	12.2322	111.5940	5.9163
			TZV2P-MOLOPT-PBE-GTH-q15	-2816.5717	12.2326	111.7111	5.9169
Fe	LAPW			-34634.7911	11.3153	199.1108	4.8661
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae				
			pob-DZVP-rev2	-34360.2679	6.7124	130.6761	-7.7296
			pob-TZVP-rev2				
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q8					
		GTH-PBE-q16					
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-3362.3163	11.4877	182.2544	6.4381
			TZVP-MOLOPT-PBE-GTH-q16	-3362.3958	11.4974	170.0512	8.2152
			TZV2P-MOLOPT-PBE-GTH-q16	-3362.4104	11.4997	168.7266	8.3425
Co	LAPW			-37918.4805	10.8516	212.6994	4.8522
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-37581.2149	11.3055	49208.8731	-32.3308
			pob-DZVP-rev2	-37586.8463	9.6822	4163.0632	10.4383
			pob-TZVP-rev2	-37520.5903	9.1949	465.6776	-26.1644
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q9					
		GTH-PBE-q17					
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-3955.6803	10.8407	228.0858	3.8444
			TZVP-MOLOPT-PBE-GTH-q17	-3955.7399	10.9248	211.5322	5.3751
			TZV2P-MOLOPT-PBE-GTH-q17	-3955.7412	10.9239	207.7918	6.2607
Ni	LAPW			-41384.0119	10.9056	198.1051	4.8247
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-41008.0059	11.0791	79951.8624	9.5212
			pob-DZVP-rev2				
			pob-TZVP-rev2				
			TZVPP-MOLOPT-PBE-ae				
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q10					
		GTH-PBE-q18					
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-4620.3982	10.8627	197.9677	4.8579
			TZVP-MOLOPT-PBE-GTH-q18	-4620.4496	10.8856	198.7912	3.3097
			TZV2P-MOLOPT-PBE-GTH-q18	-4620.4516	10.8839	194.6025	4.9001
Cu	LAPW			-45035.7586	11.9859	140.3096	4.8144
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-44616.8409	12.2348	9021.7156	-37.6415
			pob-DZVP-rev2	-44655.6882	12.2161	3332.6517	0.0525
			pob-TZVP-rev2				
			TZVPP-MOLOPT-PBE-ae				

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q1					
		GTH-PBE-q11		-1283.5713	12.0993	228.0067	-9.5291
		GTH-PBE-q19		-5246.4965	11.2939	22.1959	-251.1812
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1300.4918	12.0711	141.4677	5.0290
			TZVP-MOLOPT-PBE-GTH-q11	-1300.5211	12.0746	141.6981	5.3956
			TZV2P-MOLOPT-PBE-GTH-q11	-1300.5225	12.0747	142.1636	5.0380
Zn	LAPW			-48874.8419	15.2440	73.5165	5.2441
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-48389.4508	14.5649	88.7800	2.9605
			pob-DZVP-rev2	-48406.4643	12.6267	291.2088	5.3469
			pob-TZVP-rev2				
			TZVPP-MOLOPT-PBE-ae	-50950.8147	9.7724	16964.9520	-238.6189
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q2		-29.0509	13.4572	84.0158	5.0079
		GTH-PBE-q12		-1611.5535	15.2201	144.8093	5.1155
		GTH-PBE-q20		-6008.5549	14.4676	181.7431	-3.5463
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1648.0874	15.3072	74.1344	5.3337
			TZVP-MOLOPT-PBE-GTH-q12	-1648.1153	15.3098	74.5469	5.3061
			TZV2P-MOLOPT-PBE-GTH-q12	-1648.1168	15.3107	74.4922	5.3025
Ga	LAPW			-52902.1015	20.4824	47.5200	5.4402
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-52364.9438	19.2522	57.4039	-0.6197
			pob-DZVP-rev2	-52365.0445	19.8653	56.3976	-0.1934
			pob-TZVP-rev2	-51721.8374	20.5386	35161.5376	-54.3024
			TZVPP-MOLOPT-PBE-ae	-52371.3612	19.7894	438.8576	35.8659
			QZVPP-MOLOPT-PBE-ae	-51761.0279	21.9255	38708.1644	-16.7032
	PW	GTH-PBE-q3		-59.8629	20.4143	47.0110	5.0577
		GTH-PBE-q13		-2027.2111	20.4782	52.4422	1.2904
		GTH-PBE-q21		-6886.1801	20.3593	48.8396	6.4117
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-2029.8579	20.5291	47.6711	5.4149
			TZVP-MOLOPT-PBE-GTH-q13	-2029.8753	20.4708	47.4701	5.4622
			TZV2P-MOLOPT-PBE-GTH-q13	-2029.8769	20.4635	47.5036	5.4779
Ge	LAPW			-57119.0868	24.0484	58.2416	4.9688
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-56503.6368	21.3411	48.5815	-5.1204
			pob-DZVP-rev2				
			pob-TZVP-rev2				
			TZVPP-MOLOPT-PBE-ae	-56509.7772	23.5487	91.0848	15.3534
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q4		-107.0346	24.0876	57.3302	4.8027
		GTH-PBE-q14		-2408.0346	23.7966	57.3622	16.7275
		GTH-PBE-q22		-7750.3315	23.2924	100.7769	24.8427
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-106.9266	24.0560	59.3451	4.7464
			TZVP-MOLOPT-PBE-GTH-q4	-106.9638	23.9927	60.0263	4.7082
			TZV2P-MOLOPT-PBE-GTH-q4	-106.9678	24.0003	59.6854	4.7212
As	LAPW			-61527.3169	22.7726	68.6159	-0.6460
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-60825.8067	22.2598	75.1267	4.0104
			pob-DZVP-rev2	-60825.8247	22.5209	103.3910	-20.6797
			pob-TZVP-rev2	-60833.2179	22.0496	73.6982	3.9439
			TZVPP-MOLOPT-PBE-ae	-60833.4578	22.4930	70.4376	4.4930
			QZVPP-MOLOPT-PBE-ae	-60834.6710	22.3740	72.1998	6.8126
	PW	GTH-PBE-q5		-171.8218	22.5743	68.5830	4.2677
		GTH-PBE-q15		-2500.8113	20.7635	290.6755	14.4549
		GTH-PBE-q23		-8257.1996	18.9559	82.2489	-0.4831
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-171.7199	22.8143	67.8998	4.2170
			TZVP-MOLOPT-PBE-GTH-q5	-171.7269	22.7778	68.0080	4.2269
			TZV2P-MOLOPT-PBE-GTH-q5	-171.7318	22.7618	67.9921	4.2238
Se	LAPW			-66129.8018	29.8919	46.2620	4.0165
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-65332.4232	29.8645	48.3635	4.4760
			pob-DZVP-rev2	-65331.7064	30.8809	43.5990	4.0573
			pob-TZVP-rev2	-65338.8805	26.8947	71.4343	5.3258
			TZVPP-MOLOPT-PBE-ae	-65340.7182	29.8011	48.1094	4.8336
			QZVPP-MOLOPT-PBE-ae	-65341.9720	29.8219	46.8656	4.5314
	PW	GTH-PBE-q6		-256.7435	29.7341	47.1912	4.4446
		GTH-PBE-q16		-3275.9888	25.6804	9.0943	-15.5984

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
		GTH-PBE-q24		-9530.6610	25.9890	13.3552	-8.3344
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-256.5321	30.1975	47.0078	4.3976
			TZVP-MOLOPT-PBE-GTH-q6	-256.5405	30.1868	46.7190	4.4010
			TZV2P-MOLOPT-PBE-GTH-q6	-256.5480	30.1593	46.6705	4.4079
Br	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-70026.8755	40.7142	20.5932	4.8628
			pob-DZVP-rev2	-70027.1760	40.9208	21.8707	4.9529
			pob-TZVP-rev2	-70035.5050	39.3252	22.9714	4.0830
			TZVPP-MOLOPT-PBE-ae	-69245.3357	35.3325	72.9323	-92.5946
			QZVPP-MOLOPT-PBE-ae				
	PW	GTH-PBE-q7		-365.2062	39.3484	22.5617	4.8391
		GTH-PBE-q17		-3854.9810	37.2014	35.8380	11.3761
		GTH-PBE-q25		-10654.3520	37.0584	23.6454	-0.8834
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-364.9852	40.0007	21.8293	4.9208
			TZVP-MOLOPT-PBE-GTH-q7	-364.9934	39.9853	21.7494	4.9053
			TZV2P-MOLOPT-PBE-GTH-q7	-365.0078	39.9261	21.7632	4.9085
Kr	LAPW						
	GAPW	ALLELECTRON	SVP-MOLOPT-PBE-ae	-74909.6363	95.4197	-0.1345	1.0146
			TZVPP-MOLOPT-PBE-ae	-74922.6949	66.4989	0.1711	1.6345
			QZVPP-MOLOPT-PBE-ae	-73568.6689	57.3050	91.8538	-308.0157
	PW	GTH-PBE-q8		-500.6643	65.8763	0.6497	7.2689
		GTH-PBE-q18		-4456.9037	65.6646	9.0162	8.3833
		GTH-PBE-q26		-11700.6840	65.0890	15.9500	13.9862
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-500.5068	64.7460	0.5411	7.2646
			TZVP-MOLOPT-PBE-GTH-q8	-500.5106	62.5483	0.6201	6.1733
			TZV2P-MOLOPT-PBE-GTH-q8	-500.5112	62.1104	0.6437	6.2586
Rb	LAPW			-81129.8272	92.0981	2.7461	3.1845
	GAPW	GTH-PBE-q9	SVP-MOLOPT-PBE-GTH-q9	-651.5563	90.0933	2.8692	3.6736
			TZVPP-MOLOPT-PBE-GTH-q9	-653.0066	74.7063	3.6456	-0.0245
			QZVPP-MOLOPT-PBE-GTH-q9	-653.1290	78.6174	13.9912	3.2770
	PW	GTH-PBE-q1		-4.8116	93.0440	2.5693	3.6856
		GTH-PBE-q9		-654.9276	91.2049	2.7948	3.6838
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-654.9021	94.3547	2.8608	3.8769
			TZVP-MOLOPT-PBE-GTH-q9	-654.9198	91.8211	2.8448	3.8370
			TZV2P-MOLOPT-PBE-GTH-q9	-654.9202	91.8513	2.8233	3.7768
Sr	LAPW			-86527.6884	54.9955	10.6029	4.2525
	GAPW	GTH-PBE-q10	SVP-MOLOPT-PBE-GTH-q10	-831.7988	59.0842	11.4846	2.4754
			TZVPP-MOLOPT-PBE-GTH-q10	-832.4314	51.5009	13.6659	4.0255
			QZVPP-MOLOPT-PBE-GTH-q10	-831.6793	32.6719	3.9719	3.2589
	PW	GTH-PBE-q2		-17.8859	53.4071	10.7449	3.0206
		GTH-PBE-q10		-834.1201	54.5556	11.5063	3.9123
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-833.9005	56.9282	10.7992	2.9846
			TZVP-MOLOPT-PBE-GTH-q10	-833.9589	55.0517	11.4562	3.2321
			TZV2P-MOLOPT-PBE-GTH-q10	-833.9983	54.7255	11.5565	3.9040
Y	LAPW			-92130.4108	33.0776	40.5754	4.2465
	GAPW	GTH-PBE-q11	SVP-MOLOPT-PBE-GTH-q11	-1038.2627	32.2068	48.9834	2.6694
			TZVPP-MOLOPT-PBE-GTH-q11	-1038.5534	32.3690	43.4974	2.6356
			QZVPP-MOLOPT-PBE-GTH-q11				
	PW	GTH-PBE-q3		-42.3562	32.7431	38.7292	2.5899
		GTH-PBE-q11		-1038.8985	32.8459	41.2468	3.0675
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1038.9518	33.1871	41.4001	3.1381
			TZVP-MOLOPT-PBE-GTH-q11	-1039.0185	32.8953	41.2579	3.0320
			TZV2P-MOLOPT-PBE-GTH-q11	-1039.0246	32.9129	41.1240	3.0100
Zr	LAPW			-97940.1661	23.5350	93.7871	1.7957
	GAPW	GTH-PBE-q12	SVP-MOLOPT-PBE-GTH-q12	-1268.7395	23.5031	99.0107	2.8994
			TZVPP-MOLOPT-PBE-GTH-q12	-1268.6678	22.6655	97.7125	3.3004
			QZVPP-MOLOPT-PBE-GTH-q12	-1268.6676	22.9563	97.9211	3.2649
	PW	GTH-PBE-q4		-82.1015	23.2783	92.3712	3.1868
		GTH-PBE-q12		-1269.2119	23.5412	93.3070	3.0292
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1269.2911	23.7180	94.6828	3.3921
			TZVP-MOLOPT-PBE-GTH-q12	-1269.3507	23.5606	93.2574	3.2967
			TZV2P-MOLOPT-PBE-GTH-q12	-1269.3564	23.5670	93.0691	3.2958
Nb	LAPW			-103961.0230	18.2270	170.6344	2.9425

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	GAPW	GTH-PBE-q13	SVP-MOLOPT-PBE-GTH-q13	-1539.1646	17.9445	172.5274	3.4716
			TZVPP-MOLOPT-PBE-GTH-q13	-1539.5935	17.5570	179.5144	3.9489
			QZVPP-MOLOPT-PBE-GTH-q13				
	PW	GTH-PBE-q5		-137.9394	19.4597	147.3837	3.6098
		GTH-PBE-q13		-1540.9654	18.2507	167.5931	3.5326
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1540.7734	18.2987	170.1492	3.7053
			TZVP-MOLOPT-PBE-GTH-q13	-1540.8092	18.2300	169.4620	3.7214
			TZV2P-MOLOPT-PBE-GTH-q13	-1540.8215	18.2512	168.6857	3.7221
Mo	LAPW			-110195.7119	15.8871	261.6172	2.9066
	GAPW	GTH-PBE-q14	SVP-MOLOPT-PBE-GTH-q14	-1843.5188	15.6807	262.3867	4.0819
			TZVPP-MOLOPT-PBE-GTH-q14	-1844.5570	15.6839	263.6758	4.0858
			QZVPP-MOLOPT-PBE-GTH-q14				
	PW	GTH-PBE-q6		-212.4822	17.4596	233.4783	3.8265
		GTH-PBE-q14		-1845.7719	15.9182	258.1047	4.7304
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-1845.4385	15.9215	259.4148	4.2400
			TZVP-MOLOPT-PBE-GTH-q14	-1845.5112	15.9069	258.4687	4.2234
			TZV2P-MOLOPT-PBE-GTH-q14	-1845.5154	15.9095	258.4475	4.2147
Tc	LAPW			-116647.1203	14.5188	297.5740	5.7451
	GAPW	GTH-PBE-q15	SVP-MOLOPT-PBE-GTH-q15	-2173.4575	14.4925	310.4461	3.3336
			TZVPP-MOLOPT-PBE-GTH-q15	-2174.0060	14.1446	315.2489	4.5748
			QZVPP-MOLOPT-PBE-GTH-q15				
	PW	GTH-PBE-q7		-315.1877	15.4045	288.8086	4.5224
		GTH-PBE-q15		-2175.8540	14.6692	295.9129	4.4642
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2175.5391	14.6218	302.7981	4.5348
			TZVP-MOLOPT-PBE-GTH-q15	-2175.5771	14.6844	294.7529	4.5366
			TZV2P-MOLOPT-PBE-GTH-q15	-2175.5803	14.6849	294.9145	4.5320
Ru	LAPW			-123318.1865	13.7822	311.6958	5.0358
	GAPW	GTH-PBE-q16	SVP-MOLOPT-PBE-GTH-q16	-2522.6833	14.1666	359.4352	0.5194
			TZVPP-MOLOPT-PBE-GTH-q16	-2523.9103	13.9140	324.7799	4.7195
			QZVPP-MOLOPT-PBE-GTH-q16				
	PW	GTH-PBE-q8		-445.2143	13.9285	300.8632	4.8022
		GTH-PBE-q16		-2526.4513	14.2730	311.6089	4.7995
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-2525.9617	14.3525	309.9522	4.8172
			TZVP-MOLOPT-PBE-GTH-q16	-2526.0052	14.2894	310.5957	4.8525
			TZV2P-MOLOPT-PBE-GTH-q16	-2526.0084	14.2923	310.0475	4.8508
Rh	LAPW			-130213.8263	14.0854	256.9706	4.0888
	GAPW	GTH-PBE-q17	SVP-MOLOPT-PBE-GTH-q17	-74797.4289	17.6190	287638.2153	6563.6319
			TZVPP-MOLOPT-PBE-GTH-q17	-2931.5601	12.6873	387.9537	5.1031
			QZVPP-MOLOPT-PBE-GTH-q17				
	PW	GTH-PBE-q9		-593.4592	14.5411	246.1175	5.1288
		GTH-PBE-q17		-2936.7442	14.5646	249.2615	5.1672
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-2936.3840	14.6157	251.8141	5.1669
			TZVP-MOLOPT-PBE-GTH-q17	-2936.4552	14.6040	248.4729	5.1745
			TZV2P-MOLOPT-PBE-GTH-q17	-2936.4559	14.6044	248.2922	5.1727
Pd	LAPW			-137337.2312	15.3464	171.3886	3.4718
	GAPW	GTH-PBE-q18	SVP-MOLOPT-PBE-GTH-q18	-3400.2825	14.3175	265.3936	3.8631
			TZVPP-MOLOPT-PBE-GTH-q18	-3401.2674	15.2154	210.2143	4.7731
			QZVPP-MOLOPT-PBE-GTH-q18				
	PW	GTH-PBE-q10		-780.6251	15.8988	162.1933	5.5851
		GTH-PBE-q18		-3408.0960	15.7730	163.5727	5.5914
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-3407.6971	16.1237	155.0134	5.4432
			TZVP-MOLOPT-PBE-GTH-q18	-3407.7994	15.8102	163.9947	5.5818
			TZV2P-MOLOPT-PBE-GTH-q18	-3407.8024	15.8217	163.3482	5.5629
Ag	LAPW			-144691.3061	17.9062	88.4021	3.5726
	GAPW	GTH-PBE-q19	SVP-MOLOPT-PBE-GTH-q19	-3897.2426	18.1021	81.8125	5.0988
			TZVPP-MOLOPT-PBE-GTH-q19	-3910.0087	15.6485	200.5091	4.9903
			QZVPP-MOLOPT-PBE-GTH-q19	682944.0327	18.4175	1333982.5517	131638.9363
	PW	GTH-PBE-q1					
		GTH-PBE-q11		-1010.3986	17.7619	93.1423	6.3227
		GTH-PBE-q19		-3918.4894	18.4500	85.8810	5.7821
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1010.3439	17.7935	92.8240	5.9344
			TZVP-MOLOPT-PBE-GTH-q11	-1010.3616	17.7903	92.2488	5.9170
			TZV2P-MOLOPT-PBE-GTH-q11	-1010.3621	17.7916	92.2007	5.9189

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
Cd	LAPW			-152275.5616	22.9753	42.2624	3.0000
	GAPW	GTH-PBE-q20	SVP-MOLOPT-PBE-GTH-q20	-4439.9972	10.6294	40.3857	-18.7317
			TZVPP-MOLOPT-PBE-GTH-q20	-4434.7953	13.6216	45.2405	-4.4565
			QZVPP-MOLOPT-PBE-GTH-q20				
	PW	GTH-PBE-q2		-26.9872	18.6698	72.1597	5.7806
		GTH-PBE-q12		-1253.6335	22.9437	44.0012	6.7954
		GTH-PBE-q20		-4452.8249	23.2169	43.4445	8.1295
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1253.5483	22.9371	42.2800	7.0277
			TZVP-MOLOPT-PBE-GTH-q12	-1253.6046	22.9383	45.1234	6.7340
			TZV2P-MOLOPT-PBE-GTH-q12	-1253.6106	22.9657	44.4243	6.7494
In	LAPW						
	GAPW	GTH-PBE-q21	SVP-MOLOPT-PBE-GTH-q21	-4992.1593	27.4509	30.0089	5.2670
			TZVPP-MOLOPT-PBE-GTH-q21				
			QZVPP-MOLOPT-PBE-GTH-q21				
	PW	GTH-PBE-q3		-54.3153	27.8840	34.4413	4.6923
		GTH-PBE-q13		-1532.6001	27.6694	35.7758	5.0385
		GTH-PBE-q21		-5026.5513	28.2543	34.8599	4.9832
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1532.5661	27.7937	35.3973	5.0618
			TZVP-MOLOPT-PBE-GTH-q13	-1532.5844	27.6867	35.5252	5.0506
			TZV2P-MOLOPT-PBE-GTH-q13	-1532.5851	27.6957	35.4635	5.0530
Sn	LAPW			-168141.5470	37.0488	35.6577	3.3727
	GAPW	GTH-PBE-q22	SVP-MOLOPT-PBE-GTH-q22	-5604.0354	33.9814	40.1039	5.1818
			TZVPP-MOLOPT-PBE-GTH-q22	-5606.9888	27.6418	79.4478	4.5367
			QZVPP-MOLOPT-PBE-GTH-q22	-5641.3669	14.6281	48.7982	-26.1690
	PW	GTH-PBE-q4		-95.2024	37.1211	34.2577	4.6412
		GTH-PBE-q14		-1838.5250	36.5339	35.4103	4.9393
		GTH-PBE-q22		-5650.5889	37.5899	34.6881	4.6320
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-95.0111	37.3317	34.2034	4.6988
			TZVP-MOLOPT-PBE-GTH-q4	-95.0274	37.2004	34.2282	4.7028
			TZV2P-MOLOPT-PBE-GTH-q4	-95.0340	37.2335	33.9569	4.6971
Sb	LAPW						
	GAPW	GTH-PBE-q23	SVP-MOLOPT-PBE-GTH-q23	-6281.7467	9.6020	46.9886	-17.5695
			TZVPP-MOLOPT-PBE-GTH-q23				
			QZVPP-MOLOPT-PBE-GTH-q23	-6265.6061	15.6281	52.3215	-4.7131
	PW	GTH-PBE-q5		-150.7143	31.9805	50.1644	4.5045
		GTH-PBE-q15		-2179.7602	31.1943	50.4064	4.6437
		GTH-PBE-q23		-6316.8763	32.5797	49.7097	4.4594
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-150.4696	32.3834	49.0183	4.4650
			TZVP-MOLOPT-PBE-GTH-q5	-150.5113	32.2241	49.4707	4.4656
			TZV2P-MOLOPT-PBE-GTH-q5	-150.5259	32.1919	49.5701	4.4802
Te	LAPW			-184951.5381	35.1715	45.3929	0.2960
	GAPW	GTH-PBE-q24	SVP-MOLOPT-PBE-GTH-q24	-6996.5587	36.9860	41.6650	4.6448
			TZVPP-MOLOPT-PBE-GTH-q24	-6961.7302	16.8011	31.4536	-20.8773
			QZVPP-MOLOPT-PBE-GTH-q24				
	PW	GTH-PBE-q6		-222.2432	35.1684	44.7100	4.6997
		GTH-PBE-q16		-2560.8822	33.7584	43.5621	4.2202
		GTH-PBE-q24		-7023.0537	35.8682	42.8242	5.0089
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-221.8612	35.5427	44.8608	4.6523
			TZVP-MOLOPT-PBE-GTH-q6	-222.1210	35.4889	43.8980	4.7214
			TZV2P-MOLOPT-PBE-GTH-q6	-222.1344	35.4819	43.7561	4.7253
I	LAPW						
	GAPW	GTH-PBE-q25	SVP-MOLOPT-PBE-GTH-q25	-7709.0413	39.7665	32.8982	5.3693
			TZVPP-MOLOPT-PBE-GTH-q25	-7705.5637	33.0413	137.3011	5.6245
			QZVPP-MOLOPT-PBE-GTH-q25	-7701.6781	31.4296	147.0058	5.9464
	PW	GTH-PBE-q7		-312.2300	50.5566	18.5615	5.0544
		GTH-PBE-q17		-2962.2854	48.2506	18.2687	5.9143
		GTH-PBE-q25		-7788.2584	51.2825	17.5422	5.5375
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-311.8713	52.8342	16.7653	5.0700
			TZVP-MOLOPT-PBE-GTH-q7	-311.9248	51.8419	17.4081	5.0717
			TZV2P-MOLOPT-PBE-GTH-q7	-311.9408	51.7026	17.4937	5.0677
Xe	LAPW						
	GAPW	GTH-PBE-q26	SVP-MOLOPT-PBE-GTH-q26	-8492.9824	54.7299	3.9148	6.9710
			TZVPP-MOLOPT-PBE-GTH-q26	-8480.6895	41.0818	16.1830	6.5245

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			QZVPP-MOLOPT-PBE-GTH-q26				
	PW	GTH-PBE-q8		-423.3691	86.8302	0.5417	7.5104
		GTH-PBE-q18		-3405.4621	78.1091	0.7716	10.3010
		GTH-PBE-q26		-8595.2101	86.6970	0.6431	5.2668
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-423.1778	86.1253	0.6097	7.9132
			TZVP-MOLOPT-PBE-GTH-q8	-423.1787	85.3436	0.6253	7.8187
			TZV2P-MOLOPT-PBE-GTH-q8	-423.1796	84.4831	0.6352	7.6554
Cs	LAPW			-211979.2517	124.9737	0.2660	6.4129
	GAPW	GTH-PBE-q9	SVP-MOLOPT-PBE-GTH-q9	-546.6368	98.3904	3.6051	6.1249
			TZVPP-MOLOPT-PBE-GTH-q9	-546.4776	95.6529	2.7856	4.3104
			QZVPP-MOLOPT-PBE-GTH-q9	-546.9571	108.9744	2.4976	3.0382
	PW	GTH-PBE-q1		-4.3166	136.6352	1.4053	2.7757
		GTH-PBE-q9		-547.5845	117.3563	1.9514	3.6406
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-547.5302	122.9767	2.0686	3.2544
			TZVP-MOLOPT-PBE-GTH-q9	-547.5552	117.4911	2.0343	3.2840
			TZV2P-MOLOPT-PBE-GTH-q9	-547.5564	117.8092	1.9946	3.2983
Ba	LAPW			-221477.4001	66.6585	5.8304	2.7918
	GAPW	GTH-PBE-q10	SVP-MOLOPT-PBE-GTH-q10	-691.2548	65.0610	8.4637	2.6648
			TZVPP-MOLOPT-PBE-GTH-q10	-691.4556	64.8577	9.0192	2.9426
			QZVPP-MOLOPT-PBE-GTH-q10	-691.9766	64.1048	9.5054	2.8558
	PW	GTH-PBE-q2		-16.8503	57.1428	6.9943	1.5664
		GTH-PBE-q10		-692.2466	63.3266	8.7291	3.0289
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-691.5125	66.6178	8.6096	2.9763
			TZVP-MOLOPT-PBE-GTH-q10	-691.5637	64.8182	8.8545	3.2565
			TZV2P-MOLOPT-PBE-GTH-q10	-691.6413	64.2469	8.8632	3.0799
Hf	LAPW			-410832.4160	22.4965	16.0218	3.0000
	GAPW	GTH-PBE-q12	SVP-MOLOPT-PBE-GTH-q12	-1338.9596	22.9666	119.3659	4.3922
			TZVPP-MOLOPT-PBE-GTH-q12	-1341.0620	22.4446	117.3297	-0.2591
			QZVPP-MOLOPT-PBE-GTH-q12				
	PW	GTH-PBE-q12		-1342.9753	22.5220	107.0614	3.2738
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1341.1952	22.6148	107.2820	3.4417
			TZVP-MOLOPT-PBE-GTH-q12	-1341.2693	22.5766	106.9953	3.3876
			TZV2P-MOLOPT-PBE-GTH-q12	-1341.2819	22.5482	107.2377	3.3756
Ta	LAPW			-425207.6713	18.3251	16.0218	3.0000
	GAPW	GTH-PBE-q13	SVP-MOLOPT-PBE-GTH-q13	-1578.6407	18.4964	195.9132	3.6432
			TZVPP-MOLOPT-PBE-GTH-q13	-1581.2241	18.2119	213.9765	-8.8466
			QZVPP-MOLOPT-PBE-GTH-q13				
	PW	GTH-PBE-q5		-134.1937	19.6380	179.1351	3.5234
		GTH-PBE-q13		-1583.1241	18.1750	192.5867	3.4763
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1583.0439	18.1853	193.5176	3.7607
			TZVP-MOLOPT-PBE-GTH-q13	-1583.0901	18.1935	195.0192	3.7607
			TZV2P-MOLOPT-PBE-GTH-q13	-1583.0920	18.2000	194.6478	3.7555
W	LAPW			-439902.3659	16.2260	16.0218	3.0000
	GAPW	GTH-PBE-q14	SVP-MOLOPT-PBE-GTH-q14	-1850.6744	16.0180	315.5015	4.1259
			TZVPP-MOLOPT-PBE-GTH-q14				
			QZVPP-MOLOPT-PBE-GTH-q14				
	PW	GTH-PBE-q6		-208.2997	16.9495	283.0834	4.1167
		GTH-PBE-q14		-1854.5729	16.0212	303.2627	4.2467
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-1854.4879	16.0372	306.1920	4.1721
			TZVP-MOLOPT-PBE-GTH-q14	-1854.5220	16.0180	305.0462	4.1874
			TZV2P-MOLOPT-PBE-GTH-q14	-1854.5266	16.0232	304.7623	4.1716
Re	LAPW			-454919.9135	14.9325	16.0218	3.0000
	GAPW	GTH-PBE-q15	SVP-MOLOPT-PBE-GTH-q15	-2152.3892	14.7936	390.9635	4.3283
			TZVPP-MOLOPT-PBE-GTH-q15	-2153.4731	14.1772	392.2084	5.2668
			QZVPP-MOLOPT-PBE-GTH-q15				
	PW	GTH-PBE-q7		-297.6067	15.7396	349.3212	4.3292
		GTH-PBE-q15		-2157.6248	14.8398	366.5660	4.3848
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2157.5128	14.8676	369.3230	4.4219
			TZVP-MOLOPT-PBE-GTH-q15	-2157.5826	14.8750	368.5825	4.4421
			TZV2P-MOLOPT-PBE-GTH-q15	-2157.5854	14.8764	367.9852	4.4419
Os	LAPW			-470266.5384	14.0733	16.0218	3.0000
	GAPW	GTH-PBE-q16	SVP-MOLOPT-PBE-GTH-q16	-2485.9498	14.0249	436.6202	4.9334
			TZVPP-MOLOPT-PBE-GTH-q16	-2485.7841	13.6225	427.4195	4.1501

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			QZVPP-MOLOPT-PBE-GTH-q16				
	PW	GTH-PBE-q8		-414.2363	14.6642	387.1398	4.7663
		GTH-PBE-q16		-2492.4326	14.1797	402.1182	4.8265
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-2492.3064	14.2159	408.0839	4.7452
			TZVP-MOLOPT-PBE-GTH-q16	-2492.3516	14.2093	402.3626	4.8160
			TZV2P-MOLOPT-PBE-GTH-q16	-2492.3561	14.2020	400.7729	4.8142
Ir	LAPW			-485944.8264	14.5661	16.0218	3.0000
	GAPW	GTH-PBE-q17	SVP-MOLOPT-PBE-GTH-q17	-2851.0859	14.1109	386.7454	4.7022
			TZVPP-MOLOPT-PBE-GTH-q17	-2805.8626	15.0891	47793.1072	-15.8257
			QZVPP-MOLOPT-PBE-GTH-q17				
	PW	GTH-PBE-q9		-547.3022	15.2097	334.3132	5.0757
		GTH-PBE-q17		-2861.9186	14.3719	354.6361	5.0398
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-2861.2962	13.5010	540.8082	4.3130
			TZVP-MOLOPT-PBE-GTH-q17	-2861.6280	13.9899	433.6238	4.5734
			TZV2P-MOLOPT-PBE-GTH-q17	-2861.7318	14.1517	396.2893	4.7578
Pt	LAPW						
	GAPW	GTH-PBE-q18	SVP-MOLOPT-PBE-GTH-q18	-3268.9482	15.6568	258.2151	5.0652
			TZVPP-MOLOPT-PBE-GTH-q18	-3269.8516	15.3489	1559.0464	27.3655
			QZVPP-MOLOPT-PBE-GTH-q18				
	PW	GTH-PBE-q10		-708.8052	16.1918	244.8685	5.4822
		GTH-PBE-q18		-3271.1316	15.6211	252.1194	5.4815
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-3271.0095	15.6710	253.2527	5.5397
			TZVP-MOLOPT-PBE-GTH-q18	-3271.0310	15.6328	252.3665	5.4817
			TZV2P-MOLOPT-PBE-GTH-q18	-3271.0327	15.6331	252.2180	5.4904
Au	LAPW			-518320.7632	18.1815	16.0218	3.0000
	GAPW	GTH-PBE-q19	SVP-MOLOPT-PBE-GTH-q19	-3710.8648	17.3758	213.4867	4.7881
			TZVPP-MOLOPT-PBE-GTH-q19	-3704.7803	17.7289	449990.8956	43.1289
			QZVPP-MOLOPT-PBE-GTH-q19	-3714.4455	16.7995	206.3357	5.8839
	PW	GTH-PBE-q1		-19.5702	4.5026	50.1001	-20.1968
		GTH-PBE-q11		-901.8543	18.4337	136.4954	5.8700
		GTH-PBE-q19		-3715.5238	17.8331	146.1890	6.0326
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-901.7443	18.5287	132.7443	6.0189
			TZVP-MOLOPT-PBE-GTH-q11	-901.7653	18.5387	132.1700	6.0176
			TZV2P-MOLOPT-PBE-GTH-q11	-901.7689	18.5300	132.6993	6.0048
		GTH-PBE-q19	DZVP-MOLOPT-PBE-GTH-q19	-3715.4937	17.8730	146.7658	6.0269
			TZVP-MOLOPT-PBE-GTH-q19	-3715.5146	17.8628	145.8506	5.9381
			TZV2P-MOLOPT-PBE-GTH-q19	-3715.5178	17.8569	146.1006	5.9971
Hg	LAPW			-535026.2793	30.6494	11.1558	3.0000
	GAPW	GTH-PBE-q20	SVP-MOLOPT-PBE-GTH-q20	-4141.5733	24.0271	58.5666	6.7631
			TZVPP-MOLOPT-PBE-GTH-q20	-4142.4781	23.7489	18.7192	6.0556
			QZVPP-MOLOPT-PBE-GTH-q20	-4142.0620	19.6661	46.5061	2.6268
	PW	GTH-PBE-q2					
		GTH-PBE-q12		-1118.6168	29.5226	7.2612	2.1989
		GTH-PBE-q20		-4144.2943	27.4405	15.9500	10.7037
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1118.5914	29.9501	7.3837	10.4255
			TZVP-MOLOPT-PBE-GTH-q12	-1118.6017	29.7172	7.7093	10.3409
			TZV2P-MOLOPT-PBE-GTH-q12	-1118.6023	29.7055	7.7415	10.1985
Tl	LAPW						
	GAPW	GTH-PBE-q21	SVP-MOLOPT-PBE-GTH-q21				
			TZVPP-MOLOPT-PBE-GTH-q21	-4654.3469	29.7321	33.5858	5.1884
			QZVPP-MOLOPT-PBE-GTH-q21				
	PW	GTH-PBE-q3		-56.9485	29.8759	30.6071	5.1739
		GTH-PBE-q13		-1358.2761	31.4428	27.0857	5.5155
		GTH-PBE-q21		-4659.1376	31.7566	27.7756	3.5168
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1358.2302	31.5606	26.9708	5.4451
			TZVP-MOLOPT-PBE-GTH-q13	-1358.2431	31.4856	27.0847	5.4184
			TZV2P-MOLOPT-PBE-GTH-q13	-1358.2441	31.5021	27.0181	5.4152
Pb	LAPW			-569488.1221	35.7633	6.8160	0.0998
	GAPW	GTH-PBE-q22	SVP-MOLOPT-PBE-GTH-q22	-5195.0681	31.6539	43.1942	5.3418
			TZVPP-MOLOPT-PBE-GTH-q22	-5199.1969	27.9115	47.2267	0.0657
			QZVPP-MOLOPT-PBE-GTH-q22	-5197.5825	25.5870	190.6506	8.2109
	PW	GTH-PBE-q4		-96.2967	32.1911	40.4900	5.4012
		GTH-PBE-q14		-1627.6914	32.0706	39.5705	4.2420

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
		GTH-PBE-q22		-5202.3770	32.2811	39.6458	4.1789
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-96.2003	32.8723	39.9307	4.8842
			TZVP-MOLOPT-PBE-GTH-q4	-96.2508	32.4604	41.4390	5.0003
			TZV2P-MOLOPT-PBE-GTH-q4	-96.2528	32.4203	41.4017	4.9883
Bi	LAPW			-587253.2612	36.7274	16.0218	3.0000
	GAPW	GTH-PBE-q23	SVP-MOLOPT-PBE-GTH-q23	-5780.7539	36.6911	44.0538	4.7980
			TZVPP-MOLOPT-PBE-GTH-q23	-5783.6829	0.0115	42.8991	7.2225
			QZVPP-MOLOPT-PBE-GTH-q23	-5782.6815	29.7441	84.4283	4.3809
	PW	GTH-PBE-q5		-149.3625	37.0574	43.2011	4.5897
		GTH-PBE-q15		-1920.6651	36.9289	42.7744	4.6820
		GTH-PBE-q23		-5790.5350	36.7780	42.5205	4.6857
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-149.2726	37.4683	42.1366	4.5977
			TZVP-MOLOPT-PBE-GTH-q5	-149.2973	37.2946	42.4017	4.6188
			TZV2P-MOLOPT-PBE-GTH-q5	-149.3068	37.2634	42.4404	4.6372
Po	LAPW						
	GAPW	GTH-PBE-q24	SVP-MOLOPT-PBE-GTH-q24	-6448.9450	37.2959	44.0253	4.9844
			TZVPP-MOLOPT-PBE-GTH-q24	-6455.3571	34.2403	57.4580	4.7942
			QZVPP-MOLOPT-PBE-GTH-q24	-6455.7853	30.5736	103.2762	3.4301
	PW	GTH-PBE-q6		-219.2938	37.5453	45.8776	4.9142
		GTH-PBE-q16		-2248.5516	37.3075	45.2492	4.9807
		GTH-PBE-q24		-6460.9174	36.9941	45.3715	4.2765
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-219.1878	37.9580	44.4075	5.0600
			TZVP-MOLOPT-PBE-GTH-q6	-219.2102	37.8878	44.3168	5.0018
			TZV2P-MOLOPT-PBE-GTH-q6	-219.2145	37.8955	44.2307	4.9417
Rn	LAPW						
	GAPW	GTH-PBE-q26	SVP-MOLOPT-PBE-GTH-q26	-11681.7605	54.3012	502.0782	-992.0318
			TZVPP-MOLOPT-PBE-GTH-q26	-7977.6944	62.7742	2.3488	7.2259
			QZVPP-MOLOPT-PBE-GTH-q26				
	PW	GTH-PBE-q8		-407.1189	93.2580	0.5387	7.2186
		GTH-PBE-q18		-2975.9580	90.2585	0.2744	-18.1617
		GTH-PBE-q26		-7996.3985	95.1867	3.0496	-32.4445
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-407.0654	95.5170	0.3983	6.1548
			TZVP-MOLOPT-PBE-GTH-q8	-407.0758	88.9960	0.6043	6.3695
			TZV2P-MOLOPT-PBE-GTH-q8	-407.0788	88.8468	0.5827	6.4658

Table B.2.: Fitted Birch-Murnaghan coefficients for all PBE calculations.

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
H	LAPW			-15.6175	17.9351	9.7015	2.7374
	PW	GTH-PBE-q1		-15.5965	18.0790	9.5865	2.7121
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-15.5932	17.8161	9.9201	2.6930
			TZVP-MOLOPT-PBE-GTH-q1	-15.5949	17.9048	9.8088	2.6808
			TZV2P-MOLOPT-PBE-GTH-q1	-15.6069	17.8253	10.0549	2.6873
He	LAPW			-77.7728	21.0995	0.1208	20.2427
	PW	GTH-PBE-q2		-77.6020	19.8611	0.6708	1.5808
	GPW	GTH-PBE-q2	DZVP-MOLOPT-PBE-GTH-q2	-77.7294	19.9286	0.4849	5.9682
			TZVP-MOLOPT-PBE-GTH-q2	-77.7298	19.9464	0.4839	5.9702
			TZV2P-MOLOPT-PBE-GTH-q2	-77.7298	19.9122	0.4846	5.9453
Li	LAPW			-202.9954	20.2418	13.6170	3.2875
	PW	GTH-PBE-q1		-7.1200	19.0675	13.6417	3.1462
		GTH-PBE-q3		-200.9590	20.2482	13.8208	1.7343
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-7.0503	20.3347	13.4267	2.9375
			TZVP-MOLOPT-PBE-GTH-q1	-7.0508	20.2822	13.3653	3.0201
			TZV2P-MOLOPT-PBE-GTH-q1	-7.0722	19.5381	12.9309	2.9016
		GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-201.7400	20.7896	13.5594	3.4331
			TZVP-MOLOPT-PBE-GTH-q3	-201.7607	20.2511	13.7021	3.3640
			TZV2P-MOLOPT-PBE-GTH-q3	-201.7745	20.1708	13.7055	3.3566
Be	LAPW			-399.4924	7.8404	123.8039	3.2539
	PW	GTH-PBE-q2		-30.8790	7.7331	117.1984	3.1140
		GTH-PBE-q4		-394.1024	7.8651	126.3576	5.1439

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	GPW	GTH-PBE-q2	DZVP-MOLOPT-PBE-GTH-q2	-30.7010	7.6567	127.5143	3.1437
			TZVP-MOLOPT-PBE-GTH-q2	-30.7488	7.7630	121.2539	3.1036
			TZV2P-MOLOPT-PBE-GTH-q2	-30.8690	7.7223	118.1314	3.1498
		GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-396.7452	7.8694	130.4650	3.2672
			TZVP-MOLOPT-PBE-GTH-q4	-396.7628	7.8600	126.1365	3.2568
			TZV2P-MOLOPT-PBE-GTH-q4	-396.8834	7.8307	121.5060	3.1801
B	LAPW			-672.9501	7.1560	242.7251	3.5049
	PW	GTH-PBE-q3		-77.0245	7.1852	238.2562	3.4450
	GPW	GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-76.9809	7.1870	238.8967	3.4178
			TZVP-MOLOPT-PBE-GTH-q3	-76.9917	7.1805	238.8944	3.3839
			TZV2P-MOLOPT-PBE-GTH-q3				
C	LAPW			-1032.6189	11.5227	213.6417	3.5584
	PW	GTH-PBE-q4		-154.6064	11.6086	210.8686	3.6371
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-154.5262	11.6344	210.5249	3.5495
			TZVP-MOLOPT-PBE-GTH-q4	-154.5343	11.6464	209.7561	3.5349
			TZV2P-MOLOPT-PBE-GTH-q4	-154.5715	11.6217	209.1216	3.5357
N	LAPW			-1484.6057	28.6688	54.3998	3.5423
	PW	GTH-PBE-q5		-269.8368	28.9594	53.5525	3.4962
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-269.7178	30.1529	52.9613	3.6863
			TZVP-MOLOPT-PBE-GTH-q5	-269.8010	29.3106	52.3461	3.6316
			TZV2P-MOLOPT-PBE-GTH-q5	-269.8282	29.1882	52.4314	3.6411
O	LAPW			-2039.2165	17.9909	52.4608	3.4924
	PW	GTH-PBE-q6					
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-432.9234	19.0629	52.2028	3.8466
			TZVP-MOLOPT-PBE-GTH-q6	-432.9395	18.9150	50.3823	3.7796
			TZV2P-MOLOPT-PBE-GTH-q6	-432.9857	18.7784	50.5589	3.7576
F	LAPW			-2708.0879	18.4738	36.6774	3.8632
	PW	GTH-PBE-q7		-647.4913	18.3317	41.4198	5.4898
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-655.3982	18.9854	37.0424	4.1070
			TZVP-MOLOPT-PBE-GTH-q7	-655.4002	18.9928	36.8143	4.0992
			TZV2P-MOLOPT-PBE-GTH-q7	-655.4322	18.9149	36.4846	4.0223
Ne	LAPW			-3501.3537	24.4018	3.1154	41.0698
	PW	GTH-PBE-q8		-942.1426	23.9522	4.0881	24.9124
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-947.3654	24.5037	0.8311	8.4246
			TZVP-MOLOPT-PBE-GTH-q8	-947.3692	24.0489	0.9035	8.0568
			TZV2P-MOLOPT-PBE-GTH-q8	-947.3693	23.9730	0.9266	8.2408
Na	LAPW			-4409.5381	36.6559	8.0266	-7.5984
	PW	GTH-PBE-q1		-6.1841	38.2590	7.1647	3.6378
		GTH-PBE-q9		-1251.4325	36.5159	9.6095	13.9927
	GPW	GTH-PBE-q1	DZVP-MOLOPT-PBE-GTH-q1	-6.1637	38.8908	6.9908	3.6272
			TZVP-MOLOPT-PBE-GTH-q1	-6.1701	39.0175	7.1693	3.4953
			TZV2P-MOLOPT-PBE-GTH-q1	-6.1815	38.2764	7.1584	3.6362
		GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-1300.6153	37.9545	6.8924	3.4801
			TZVP-MOLOPT-PBE-GTH-q9	-1300.6203	37.5664	6.9084	3.5294
			TZV2P-MOLOPT-PBE-GTH-q9	-1300.6268	37.3703	6.8567	3.5051
Mg	LAPW			-5439.6553	22.5329	37.4777	4.1302
	PW	GTH-PBE-q2		-23.9696	23.3094	34.6972	3.9354
		GTH-PBE-q10		-1633.8860	24.0304	44.0289	-5.7322
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-1719.0342	23.1223	35.2117	4.0319
			TZVP-MOLOPT-PBE-GTH-q10	-1719.0504	23.1577	34.6195	3.9649
			TZV2P-MOLOPT-PBE-GTH-q10	-1719.1821	22.6237	36.3589	4.1597
Al	LAPW						
	PW	GTH-PBE-q3		-56.3944	16.4914	77.9540	4.4989
	GPW	GTH-PBE-q3	DZVP-MOLOPT-PBE-GTH-q3	-56.3197	16.4147	80.6000	4.5236
			TZVP-MOLOPT-PBE-GTH-q3	-56.3521	16.4416	80.2061	4.5778
			TZV2P-MOLOPT-PBE-GTH-q3	-56.3723	16.4533	79.7548	4.5872
Si	LAPW			-7877.9823	20.1109	92.7024	4.2244
	PW	GTH-PBE-q4		-106.9456	20.2276	90.8298	4.2316
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-106.8487	20.2592	90.6634	4.2288
			TZVP-MOLOPT-PBE-GTH-q4	-106.8634	20.2844	89.7687	4.2422
			TZV2P-MOLOPT-PBE-GTH-q4	-106.8992	20.2029	90.8387	4.2643
P	LAPW						
	PW	GTH-PBE-q5		-179.0315	20.7907	74.2039	4.3545

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-178.8067	21.1444	69.5697	4.3526
			TZVP-MOLOPT-PBE-GTH-q5	-178.8419	21.0912	70.4101	4.3201
			TZV2P-MOLOPT-PBE-GTH-q5	-178.9557	20.8559	72.9439	4.3294
S	LAPW			-10845.2256	16.4581	97.0646	4.6022
	PW	GTH-PBE-q6		-276.8883	16.3473	96.1688	4.2100
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-276.7497	16.5665	96.0646	4.1256
			TZVP-MOLOPT-PBE-GTH-q6	-276.7591	16.5043	95.3515	4.1302
			TZV2P-MOLOPT-PBE-GTH-q6	-276.8200	16.3806	96.8222	4.0594
Cl	LAPW						
	PW	GTH-PBE-q7		-406.2473	37.1298	20.4757	4.3820
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-406.0244	38.1667	20.6890	4.4608
			TZVP-MOLOPT-PBE-GTH-q7	-406.0302	38.0997	20.6758	4.4922
			TZV2P-MOLOPT-PBE-GTH-q7	-406.1159	37.4099	20.9964	4.4162
Ar	LAPW			-14381.0982	52.6788	0.3308	3.2626
	PW	GTH-PBE-q8		-571.7792	49.9793	0.3872	0.4509
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-571.7649	46.9049	0.8058	8.0586
			TZVP-MOLOPT-PBE-GTH-q8	-571.7697	45.4875	0.8458	7.7770
			TZV2P-MOLOPT-PBE-GTH-q8	-571.7698	45.4690	0.8460	7.7965
K	LAPW			-16362.9769	71.0903	3.7856	3.3906
	PW	GTH-PBE-q1		-5.0756	76.0087	3.2522	3.6329
		GTH-PBE-q9		-767.3459	70.7169	3.6685	3.4128
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-767.2825	72.9339	3.6660	3.3952
			TZVP-MOLOPT-PBE-GTH-q9	-767.3002	70.5635	3.7131	3.4259
			TZV2P-MOLOPT-PBE-GTH-q9	-767.3009	70.5768	3.7733	3.9003
Ca	LAPW			-18491.4956	40.6359	17.9798	3.1964
	PW	GTH-PBE-q2		-19.6947	37.9025	16.4181	2.9555
		GTH-PBE-q10		-997.0935	41.4349	18.1975	3.3767
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-997.0170	42.6066	17.8436	3.3615
			TZVP-MOLOPT-PBE-GTH-q10	-997.0771	41.8017	18.3571	3.3848
			TZV2P-MOLOPT-PBE-GTH-q10	-997.0930	41.6686	18.1139	3.3995
Sc	LAPW						
	PW	GTH-PBE-q3		-46.6933	25.8351	50.7226	3.2223
		GTH-PBE-q11		-1264.6834	23.5645	54.3509	5.0900
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1266.6951	23.7657	58.6067	3.3051
			TZVP-MOLOPT-PBE-GTH-q11	-1266.7444	23.6227	57.6284	3.3749
			TZV2P-MOLOPT-PBE-GTH-q11	-1266.7491	23.6200	57.7566	3.3747
Ti	LAPW						
	PW	GTH-PBE-q4		-93.4455	18.0504	109.6496	3.4541
		GTH-PBE-q12		-1574.1227	16.6575	139.9758	8.3389
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1577.3485	16.7942	121.6664	3.4811
			TZVP-MOLOPT-PBE-GTH-q12	-1577.3764	16.7650	119.8649	3.5968
			TZV2P-MOLOPT-PBE-GTH-q12	-1577.3804	16.7750	119.6347	3.5871
V	LAPW			-25805.3026	12.9481	172.6385	0.1113
	PW	GTH-PBE-q5		-164.5336	13.7699	179.5786	3.7780
		GTH-PBE-q13		-1932.4766	12.7939	254.6950	11.4765
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1937.2884	13.0575	197.9054	3.6741
			TZVP-MOLOPT-PBE-GTH-q13	-1937.3231	13.0287	197.2329	3.8610
			TZV2P-MOLOPT-PBE-GTH-q13	-1937.3255	13.0319	196.8748	3.8579
Cr	LAPW						
	PW	GTH-PBE-q6					
		GTH-PBE-q14					
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-2318.3581	11.8299	466.3770	-30.3701
			TZVP-MOLOPT-PBE-GTH-q14	-2318.3898	11.8655	519.2989	-28.1484
			TZV2P-MOLOPT-PBE-GTH-q14	-2318.3919	11.8665	521.6223	-28.0775
Mn	LAPW						
	PW	GTH-PBE-q7					
		GTH-PBE-q15					
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2812.6811	11.3465	140.6143	3.2451
			TZVP-MOLOPT-PBE-GTH-q15	-2812.7332	11.3200	141.4244	4.0526
			TZV2P-MOLOPT-PBE-GTH-q15	-2812.7338	11.3215	141.7251	4.0899
Fe	LAPW						
	PW	GTH-PBE-q8					
		GTH-PBE-q16					

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-3357.9116	10.9544	225.4500	5.7058
			TZVP-MOLOPT-PBE-GTH-q16	-3357.9835	10.9401	221.0841	5.5855
			TZV2P-MOLOPT-PBE-GTH-q16	-3357.9969	10.9412	232.3906	7.9224
Co	LAPW						
	PW	GTH-PBE-q9					
		GTH-PBE-q17					
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-3950.7802	10.3918	281.0843	6.3721
			TZVP-MOLOPT-PBE-GTH-q17	-3950.8330	10.4416	251.4248	5.3776
			TZV2P-MOLOPT-PBE-GTH-q17	-3950.8343	10.4436	252.8794	5.6283
Ni	LAPW						
	PW	GTH-PBE-q10					
		GTH-PBE-q18					
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-4614.6753	10.3396	233.3278	4.8767
			TZVP-MOLOPT-PBE-GTH-q18	-4614.7177	10.3516	230.0563	4.9095
			TZV2P-MOLOPT-PBE-GTH-q18	-4614.7191	10.3507	230.2043	4.9117
Cu	LAPW						
	PW	GTH-PBE-q1					
		GTH-PBE-q11		-1280.2576	11.3673	22.8738	-165.8010
		GTH-PBE-q19		-5240.1493	10.5634	36.6440	-75.2832
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1297.3689	11.4569	171.0017	5.0251
			TZVP-MOLOPT-PBE-GTH-q11	-1297.3962	11.4647	173.8712	5.3640
			TZV2P-MOLOPT-PBE-GTH-q11	-1297.3975	11.4621	171.5465	5.0430
Zn	LAPW						
	PW	GTH-PBE-q2		-29.0189	13.4850	83.3286	4.8929
		GTH-PBE-q12		-1607.4709	14.7094	161.0557	4.7928
		GTH-PBE-q20		-6001.5303	13.5664	50.1301	-38.8229
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1644.2525	14.3341	93.9999	5.3185
			TZVP-MOLOPT-PBE-GTH-q12	-1644.2804	14.3455	94.0666	5.3012
			TZV2P-MOLOPT-PBE-GTH-q12	-1644.2817	14.3464	93.9863	5.2978
Ga	LAPW			-52862.2938	19.1169	45.4918	-0.0029
	PW	GTH-PBE-q3		-59.7595	20.2426	50.0762	4.9269
		GTH-PBE-q13		-2022.7314	19.2521	59.6231	4.6890
		GTH-PBE-q21		-6878.4052	18.9490	43.3463	-2.0715
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-2025.4606	19.2735	59.9724	5.3336
			TZVP-MOLOPT-PBE-GTH-q13	-2025.4769	19.2172	59.9644	5.3762
			TZV2P-MOLOPT-PBE-GTH-q13	-2025.4783	19.2132	60.0368	5.3882
Ge	LAPW						
	PW	GTH-PBE-q4		-106.7798	23.8259	60.5061	4.6645
		GTH-PBE-q14		-2403.1292	22.9225	93.7680	13.0816
		GTH-PBE-q22		-7741.9624	22.7565	157.5015	18.2639
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-106.6692	23.8141	62.4032	4.6983
			TZVP-MOLOPT-PBE-GTH-q4	-106.7106	23.7546	63.0277	4.6664
			TZV2P-MOLOPT-PBE-GTH-q4	-106.7144	23.7579	62.7561	4.6793
As	LAPW						
	PW	GTH-PBE-q5		-171.4792	22.0478	74.7424	4.2257
		GTH-PBE-q15		-2495.9540	20.4965	341.8909	13.3175
		GTH-PBE-q23		-8250.2224	17.4025	52.4271	-5.5508
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-171.3765	22.2733	73.9281	4.1828
			TZVP-MOLOPT-PBE-GTH-q5	-171.3844	22.2388	74.0743	4.1925
			TZV2P-MOLOPT-PBE-GTH-q5	-171.3899	22.2243	74.0426	4.1886
Se	LAPW			-66084.9035	28.4390	40.1278	-2.3959
	PW	GTH-PBE-q6		-256.0970	28.8477	51.9211	4.4596
		GTH-PBE-q16		-3270.1771	23.5581	22.5093	0.5703
		GTH-PBE-q24		-9521.7618	24.0679	29.8087	2.3439
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-255.8821	29.3119	51.5117	4.4156
			TZVP-MOLOPT-PBE-GTH-q6	-255.8910	29.2946	51.2162	4.4206
			TZV2P-MOLOPT-PBE-GTH-q6	-255.8993	29.2683	51.1665	4.4270
Br	LAPW						
	PW	GTH-PBE-q7		-364.2143	37.6754	25.5437	4.8774
		GTH-PBE-q17		-3848.4697	54.5234	-16.1783	-0.1202
		GTH-PBE-q25		-10644.9084	36.5154	64.2625	11.9812
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-363.9828	38.2935	24.9086	4.9758
			TZVP-MOLOPT-PBE-GTH-q7	-363.9917	38.2817	24.7928	4.9579

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
			TZV2P-MOLOPT-PBE-GTH-q7	-364.0090	38.2104	24.8487	4.9577
Kr	LAPW						
	PW	GTH-PBE-q8		-499.2261	57.3510	0.5109	6.8811
		GTH-PBE-q18		-4450.1050	65.3846	8.8028	9.6459
		GTH-PBE-q26		-11691.0266	65.0551	16.7059	15.4033
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-499.0835	40.7535	1.2061	6.7366
			TZVP-MOLOPT-PBE-GTH-q8	-499.0894	42.3238	1.4494	6.8879
			TZV2P-MOLOPT-PBE-GTH-q8	-499.0909	42.0858	1.4985	6.8545
Rb	LAPW			-81079.4764	87.2984	2.7782	3.4483
	PW	GTH-PBE-q1		-4.8014	93.8370	2.5345	3.6011
		GTH-PBE-q9		-653.2606	86.6607	3.1129	4.7088
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-653.2324	90.0065	3.0333	3.8434
			TZVP-MOLOPT-PBE-GTH-q9	-653.2537	87.2269	3.0252	3.8037
			TZV2P-MOLOPT-PBE-GTH-q9	-653.2540	87.2196	2.9928	3.7347
Sr	LAPW			-86475.6853	52.0744	11.9494	3.0847
	PW	GTH-PBE-q2		-17.8745	53.6098	10.6095	2.8032
		GTH-PBE-q10		-832.3426	51.7589	12.4320	3.5327
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-832.1160	53.9982	11.1843	3.0002
			TZVP-MOLOPT-PBE-GTH-q10	-832.1801	52.2069	12.0680	3.2952
			TZV2P-MOLOPT-PBE-GTH-q10	-832.2230	51.9061	12.6600	3.9489
Y	LAPW			-92076.8136	31.5801	38.7281	-0.3092
	PW	GTH-PBE-q3		-42.3405	32.7850	38.7665	2.5970
		GTH-PBE-q11		-1037.1273	31.3668	42.4836	2.7939
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1037.1754	31.7876	43.4170	3.0491
			TZVP-MOLOPT-PBE-GTH-q11	-1037.2386	31.4510	43.0021	2.9122
			TZV2P-MOLOPT-PBE-GTH-q11	-1037.2446	31.4140	38.1368	-0.0251
Zr	LAPW			-97884.9628	22.5778	92.8066	1.6627
	PW	GTH-PBE-q4		-82.0506	23.2480	93.3018	3.2091
		GTH-PBE-q12		-1267.3340	22.6564	95.0606	2.2216
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1267.4105	22.8879	100.4964	3.4086
			TZVP-MOLOPT-PBE-GTH-q12	-1267.4679	22.6855	98.7341	3.2871
			TZV2P-MOLOPT-PBE-GTH-q12	-1267.4736	22.6895	98.5560	3.2853
Nb	LAPW			-103904.1870	17.5997	187.8443	4.7249
	PW	GTH-PBE-q5		-137.8071	19.3618	152.6901	3.4428
		GTH-PBE-q13		-1539.0512	17.6635	179.8685	3.7323
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1538.8687	17.7445	182.3243	3.7003
			TZVP-MOLOPT-PBE-GTH-q13	-1538.9045	17.6600	181.8824	3.7184
			TZV2P-MOLOPT-PBE-GTH-q13	-1538.9154	17.6776	181.0125	3.7203
Mo	LAPW			-110137.1497	15.4184	270.2258	3.3334
	PW	GTH-PBE-q6		-212.1923	17.3178	240.9368	3.9104
		GTH-PBE-q14		-1843.6086	15.4914	278.6786	4.1391
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-1843.2781	15.5114	280.1252	4.2749
			TZVP-MOLOPT-PBE-GTH-q14	-1843.3716	15.4877	279.2913	4.2131
			TZV2P-MOLOPT-PBE-GTH-q14	-1843.3752	15.4899	279.1844	4.2061
Tc	LAPW			-116586.7748	14.0856	285.7584	0.3745
	PW	GTH-PBE-q7		-314.7914	15.2337	304.3792	4.4865
		GTH-PBE-q15		-2173.4664	14.2529	324.0990	4.4690
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2173.1740	14.2270	331.3396	4.5264
			TZVP-MOLOPT-PBE-GTH-q15	-2173.2072	14.2717	323.6137	4.5309
			TZV2P-MOLOPT-PBE-GTH-q15	-2173.2103	14.2725	323.6970	4.5262
Ru	LAPW			-123255.9906	13.3392	355.4945	5.4654
	PW	GTH-PBE-q8		-444.4679	13.6780	326.2908	4.7612
		GTH-PBE-q16		-2523.7902	13.8417	346.5866	4.7916
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-2523.2969	13.9361	344.1972	4.8014
			TZVP-MOLOPT-PBE-GTH-q16	-2523.3613	13.8622	346.5705	4.8394
			TZV2P-MOLOPT-PBE-GTH-q16	-2523.3644	13.8642	346.1155	4.8369
Rh	LAPW			-130149.6632	13.5328	287.9859	4.4707
	PW	GTH-PBE-q9		-592.3543	14.1529	276.9023	5.0709
		GTH-PBE-q17		-2933.6317	14.0172	284.5624	3.8745
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-2933.2799	14.0705	291.3060	5.1458
			TZVP-MOLOPT-PBE-GTH-q17	-2933.3492	14.0428	287.4739	5.1602
			TZV2P-MOLOPT-PBE-GTH-q17	-2933.3497	14.0425	287.2747	5.1597
Pd	LAPW			-137271.0081	14.2561	85.0743	-13.0124

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
	PW	GTH-PBE-q10		-779.2603	15.2670	194.1354	5.5288
		GTH-PBE-q18		-3404.4345	14.9781	201.3270	5.5761
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-3404.0124	15.2895	189.6958	5.4059
			TZVP-MOLOPT-PBE-GTH-q18	-3404.1374	15.0063	201.6153	5.5289
			TZV2P-MOLOPT-PBE-GTH-q18	-3404.1400	15.0158	200.6888	5.5108
Ag	LAPW			-144622.9715	16.6059	102.1208	3.7098
	PW	GTH-PBE-q1					
		GTH-PBE-q11		-1008.5747	16.6953	118.3687	5.6623
		GTH-PBE-q19		-3914.2429	17.1866	113.3724	5.7569
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-1008.5087	16.7656	119.8415	5.7801
			TZVP-MOLOPT-PBE-GTH-q11	-1008.5299	16.7468	119.8868	5.7905
			TZV2P-MOLOPT-PBE-GTH-q11	-1008.5305	16.7479	119.8436	5.7933
Cd	LAPW			-152205.1358	20.9190	51.2402	3.9962
	PW	GTH-PBE-q2		-26.9504	18.7169	70.1170	5.6191
		GTH-PBE-q12		-1251.3979	21.2508	64.1596	6.5342
		GTH-PBE-q20		-4448.1779	21.4432	67.5182	7.0484
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1251.3127	21.1842	63.4157	6.6632
			TZVP-MOLOPT-PBE-GTH-q12	-1251.3669	21.2756	64.8268	6.4416
			TZV2P-MOLOPT-PBE-GTH-q12	-1251.3724	21.2820	64.1235	6.4507
In	LAPW						
	PW	GTH-PBE-q3		-54.2434	27.5535	37.1747	4.5934
		GTH-PBE-q13		-1530.0407	25.8287	44.8185	5.0805
		GTH-PBE-q21		-5021.5032	26.2536	43.7301	5.0254
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1530.0042	25.9200	44.4394	5.0617
			TZVP-MOLOPT-PBE-GTH-q13	-1530.0228	25.8212	44.5937	5.0545
			TZV2P-MOLOPT-PBE-GTH-q13	-1530.0236	25.8284	44.5376	5.0573
Sn	LAPW						
	PW	GTH-PBE-q4		-94.9637	36.7148	36.1822	4.6394
		GTH-PBE-q14		-1835.6999	34.7032	41.7265	5.1229
		GTH-PBE-q22		-5645.2239	35.5901	40.2311	4.8382
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-94.7742	36.9422	36.1127	4.6485
			TZVP-MOLOPT-PBE-GTH-q4	-94.7932	36.8039	36.1735	4.6514
			TZV2P-MOLOPT-PBE-GTH-q4	-94.8009	36.8280	35.9099	4.6463
Sb	LAPW						
	PW	GTH-PBE-q5		-150.4313	31.3495	54.5435	4.4682
		GTH-PBE-q15		-2176.7607	29.7455	57.5408	4.6438
		GTH-PBE-q23		-6311.3136	31.0445	56.3745	4.4872
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-150.1820	31.7360	53.2781	4.4294
			TZVP-MOLOPT-PBE-GTH-q5	-150.2310	31.5770	53.7973	4.4322
			TZV2P-MOLOPT-PBE-GTH-q5	-150.2481	31.5441	53.9455	4.4473
Te	LAPW			-184873.0274	34.9826	16.0218	3.0000
	PW	GTH-PBE-q6		-221.7866	34.1284	50.1873	4.6780
		GTH-PBE-q16		-2557.6254	32.1137	51.5097	4.8549
		GTH-PBE-q24		-7017.2301	34.0991	50.5274	5.1612
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-221.3316	34.3778	50.6854	4.6567
			TZVP-MOLOPT-PBE-GTH-q6	-221.6616	34.4161	49.3824	4.7018
			TZV2P-MOLOPT-PBE-GTH-q6	-221.6775	34.4123	49.1878	4.7025
I	LAPW						
	PW	GTH-PBE-q7		-311.4269	48.2516	21.4603	5.0521
		GTH-PBE-q17		-2958.7020	45.3853	21.3345	5.3243
		GTH-PBE-q25		-7782.0489	48.2053	21.1116	5.2427
	GPW	GTH-PBE-q7	DZVP-MOLOPT-PBE-GTH-q7	-311.0450	50.5531	19.2914	5.1246
			TZVP-MOLOPT-PBE-GTH-q7	-311.1165	49.5022	20.1457	5.1271
			TZV2P-MOLOPT-PBE-GTH-q7	-311.1372	49.3280	20.3075	5.1218
Xe	LAPW						
	PW	GTH-PBE-q8		-422.1718	66.1123	0.7325	7.2037
		GTH-PBE-q18		-3401.5247	62.0912	0.6773	7.1358
		GTH-PBE-q26		-8588.6626	76.8252	0.8333	8.9859
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-421.9898	70.4662	0.8131	7.4066
			TZVP-MOLOPT-PBE-GTH-q8	-421.9922	68.9838	0.8611	7.3206
			TZV2P-MOLOPT-PBE-GTH-q8	-421.9943	67.2379	0.8763	7.1178
Cs	LAPW			-211894.2497	116.2106	0.6688	4.9213
	PW	GTH-PBE-q1		-4.3095	137.2055	1.3737	2.7735

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
		GTH-PBE-q9		-546.2145	109.0544	2.1394	4.0286
	GPW	GTH-PBE-q9	DZVP-MOLOPT-PBE-GTH-q9	-546.1550	115.4996	2.1214	3.1024
			TZVP-MOLOPT-PBE-GTH-q9	-546.1852	109.3304	2.0956	3.1057
			TZV2P-MOLOPT-PBE-GTH-q9	-546.1864	109.4804	2.0620	3.1332
Ba	LAPW						
	PW	GTH-PBE-q2		-16.8327	57.7823	7.1793	1.8076
		GTH-PBE-q10		-690.8517	58.2038	9.3978	3.0282
	GPW	GTH-PBE-q10	DZVP-MOLOPT-PBE-GTH-q10	-690.1286	61.8686	9.1308	3.0466
			TZVP-MOLOPT-PBE-GTH-q10	-690.1876	60.0507	9.6114	3.2527
			TZV2P-MOLOPT-PBE-GTH-q10	-690.2723	59.2595	9.5660	3.0405
Hf	LAPW						
	PW	GTH-PBE-q12		-1340.9196	21.7613	111.2478	3.0889
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1339.1732	21.8742	113.0428	3.4288
			TZVP-MOLOPT-PBE-GTH-q12	-1339.2399	21.8203	112.6487	3.3678
			TZV2P-MOLOPT-PBE-GTH-q12	-1339.2526	21.7986	112.7763	3.3482
Ta	LAPW						
	PW	GTH-PBE-q5		-134.1168	19.6094	181.7542	3.4651
		GTH-PBE-q13		-1581.1582	17.6611	203.8139	3.6137
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1581.0808	17.6987	204.6722	3.7898
			TZVP-MOLOPT-PBE-GTH-q13	-1581.1220	17.6986	206.5886	3.7854
			TZV2P-MOLOPT-PBE-GTH-q13	-1581.1237	17.7042	206.1931	3.7810
W	LAPW						
	PW	GTH-PBE-q6		-208.1176	16.8683	290.6481	4.0327
		GTH-PBE-q14		-1852.5367	15.6548	320.9545	4.2325
	GPW	GTH-PBE-q14	DZVP-MOLOPT-PBE-GTH-q14	-1852.4604	15.6947	324.2381	4.1605
			TZVP-MOLOPT-PBE-GTH-q14	-1852.4962	15.6664	323.7040	4.1713
			TZV2P-MOLOPT-PBE-GTH-q14	-1852.5004	15.6706	323.2913	4.1592
Re	LAPW						
	PW	GTH-PBE-q7		-297.3751	15.6420	361.3514	4.2858
		GTH-PBE-q15		-2155.4814	14.5076	392.7143	4.3863
	GPW	GTH-PBE-q15	DZVP-MOLOPT-PBE-GTH-q15	-2155.3762	14.5526	394.5936	4.4165
			TZVP-MOLOPT-PBE-GTH-q15	-2155.4367	14.5491	394.5329	4.4396
			TZV2P-MOLOPT-PBE-GTH-q15	-2155.4394	14.5498	393.9183	4.4387
Os	LAPW						
	PW	GTH-PBE-q8		-413.8500	14.5129	407.5752	4.7301
		GTH-PBE-q16		-2490.1047	13.8443	436.9916	4.7947
	GPW	GTH-PBE-q16	DZVP-MOLOPT-PBE-GTH-q16	-2489.9813	13.8982	441.9386	4.7257
			TZVP-MOLOPT-PBE-GTH-q16	-2490.0237	13.8785	437.4388	4.8068
			TZV2P-MOLOPT-PBE-GTH-q16	-2490.0297	13.8685	435.7268	4.8003
Ir	LAPW			-485807.8114	14.5661	16.0218	3.0000
	PW	GTH-PBE-q9		-546.6821	14.9576	362.6250	4.9087
		GTH-PBE-q17		-2859.3054	13.9634	396.8035	5.1958
	GPW	GTH-PBE-q17	DZVP-MOLOPT-PBE-GTH-q17	-2858.7421	13.2139	581.0155	4.2754
			TZVP-MOLOPT-PBE-GTH-q17	-2859.0330	13.6367	472.9057	4.5365
			TZV2P-MOLOPT-PBE-GTH-q17	-2859.1317	13.7768	434.7931	4.7368
Pt	LAPW						
	PW	GTH-PBE-q10		-707.8578	15.7577	278.4174	5.3845
		GTH-PBE-q18		-3267.9797	15.0483	293.5382	5.4551
	GPW	GTH-PBE-q18	DZVP-MOLOPT-PBE-GTH-q18	-3267.8435	15.1238	294.7404	5.4791
			TZVP-MOLOPT-PBE-GTH-q18	-3267.8764	15.0640	293.6531	5.4295
			TZV2P-MOLOPT-PBE-GTH-q18	-3267.8787	15.0648	293.5250	5.4351
Au	LAPW						
	PW	GTH-PBE-q1		-19.5271	4.5021	50.1124	-20.0740
		GTH-PBE-q11		-900.4920	17.6093	169.5067	5.8955
		GTH-PBE-q19		-3711.9232	16.8809	173.4895	5.1775
	GPW	GTH-PBE-q11	DZVP-MOLOPT-PBE-GTH-q11	-900.3741	17.6765	166.5365	5.9156
			TZVP-MOLOPT-PBE-GTH-q11	-900.3979	17.6788	165.9129	5.9262
			TZV2P-MOLOPT-PBE-GTH-q11	-900.4020	17.6740	166.3646	5.9115
		GTH-PBE-q19	DZVP-MOLOPT-PBE-GTH-q19	-3711.8849	16.9678	184.7695	5.9137
			TZVP-MOLOPT-PBE-GTH-q19	-3711.9103	16.9446	183.5796	5.8273
			TZV2P-MOLOPT-PBE-GTH-q19	-3711.9135	16.9423	184.3101	5.8857
Hg	LAPW						
	PW	GTH-PBE-q2					

Continued on next page

element	method	pseudo	basisset	E0 [eV]	V0 [Å^3/atom]	B0 [GPa]	B1
		GTH-PBE-q12		-1116.7468	22.9887	31.8898	6.7789
		GTH-PBE-q20		-4140.2559	23.7562	47.9253	7.9086
	GPW	GTH-PBE-q12	DZVP-MOLOPT-PBE-GTH-q12	-1116.6910	24.0524	33.7443	7.3404
			TZVP-MOLOPT-PBE-GTH-q12	-1116.7103	23.9250	34.7527	7.3427
			TZV2P-MOLOPT-PBE-GTH-q12	-1116.7116	23.8932	34.5859	7.3087
Tl	LAPW						
	PW	GTH-PBE-q3		-56.8631	29.3186	34.1118	5.0553
		GTH-PBE-q13		-1356.1317	28.7664	37.3614	5.5066
		GTH-PBE-q21		-4654.7803	28.8471	34.9340	4.8170
	GPW	GTH-PBE-q13	DZVP-MOLOPT-PBE-GTH-q13	-1356.0784	28.8904	36.8363	5.4402
			TZVP-MOLOPT-PBE-GTH-q13	-1356.0944	28.8016	36.9992	5.4167
			TZV2P-MOLOPT-PBE-GTH-q13	-1356.0977	28.2705	24.0547	0.0001
Pb	LAPW						
	PW	GTH-PBE-q4		-96.1621	31.6233	44.9474	5.4314
		GTH-PBE-q14		-1625.4417	30.1397	45.8825	4.2995
		GTH-PBE-q22		-5197.8551	30.2384	46.0360	4.2201
	GPW	GTH-PBE-q4	DZVP-MOLOPT-PBE-GTH-q4	-96.0540	32.3166	43.4638	4.7693
			TZVP-MOLOPT-PBE-GTH-q4	-96.1148	31.9029	45.3118	4.9262
			TZV2P-MOLOPT-PBE-GTH-q4	-96.1176	31.8559	45.2732	4.9143
Bi	LAPW						
	PW	GTH-PBE-q5		-149.0892	36.3388	46.8175	4.5666
		GTH-PBE-q15		-1918.2364	35.1900	49.1309	4.6638
		GTH-PBE-q23		-5785.8813	34.7090	39.2480	0.4069
	GPW	GTH-PBE-q5	DZVP-MOLOPT-PBE-GTH-q5	-148.9932	36.7425	45.6916	4.5726
			TZVP-MOLOPT-PBE-GTH-q5	-149.0207	36.5659	46.0046	4.5929
			TZV2P-MOLOPT-PBE-GTH-q5	-149.0316	36.5368	46.0508	4.6092
Po	LAPW						
	PW	GTH-PBE-q6		-218.9042	36.5769	51.3353	4.9146
		GTH-PBE-q16		-2245.9611	35.6352	52.5273	4.9445
		GTH-PBE-q24		-6456.1115	35.1901	52.0565	4.6517
	GPW	GTH-PBE-q6	DZVP-MOLOPT-PBE-GTH-q6	-218.7945	36.9430	50.0362	5.0178
			TZVP-MOLOPT-PBE-GTH-q6	-218.8170	36.8732	49.8237	4.9597
			TZV2P-MOLOPT-PBE-GTH-q6	-218.8216	36.8823	49.6270	4.8960
Rn	LAPW						
	PW	GTH-PBE-q8		-406.0117	61.5426	1.2324	6.7696
		GTH-PBE-q18		-2972.6854	74.2981	0.4654	0.5669
		GTH-PBE-q26		-7991.1482	100.7179	-2.1669	-7.5358
	GPW	GTH-PBE-q8	DZVP-MOLOPT-PBE-GTH-q8	-405.9687	55.0298	1.7489	6.8275
			TZVP-MOLOPT-PBE-GTH-q8	-405.9845	57.1931	2.0766	6.7812
			TZV2P-MOLOPT-PBE-GTH-q8	-405.9893	56.8314	2.1544	6.7971

```
&GLOBAL
   PROJECT aiida
   RUN_TYPE ENERGY
&END GLOBAL
&FORCE_EVAL
   &DFT
      BASIS_SET_FILE_NAME BASIS_SETS
      &KPOINTS
         FULL_GRID .FALSE.
         PARALLEL_GROUP_SIZE -1
         SCHEME MONKHORST-PACK 24 24 24
         SYMMETRY .FALSE.
      &END KPOINTS
      &MGRID
         CUTOFF 1000.0
         REL_CUTOFF 100.0
      &END MGRID
      POTENTIAL_FILE_NAME POTENTIAL
      &PRINT
         &OVERLAP_CONDITION ON
            1-NORM .TRUE.
            DIAGONALIZATION .TRUE.
         &END OVERLAP_CONDITION
      &END PRINT
      &QS
         EXTRAPOLATION USE_GUESS
         METHOD GPW
      &END QS
      &SCF
         ADDED_MOS 30
         EPS_SCF 1e-08
         MAX_SCF 200
         &MIXING
            ALPHA 0.4
            METHOD BROYDEN_MIXING
         &END MIXING
         &SMEAR
            ELECTRONIC_TEMPERATURE 300.0
            METHOD FERMI_DIRAC
         &END SMEAR
      &END SCF
      &XC
         &XC_FUNCTIONAL PBE
         &END XC_FUNCTIONAL
      &END XC
   &END DFT
   METHOD Quickstep
   &SUBSYS
      &CELL
         A 4.16424         0.0             0.0
         B 2.54986159e-16  4.16424         0.0
         C 2.54986159e-16  2.54986159e-16  4.16424
         &CELL_REF
            A 4.425165320132997 0.0 0.0
            B 2.7096322688430977e-16 4.425165320132997 0.0
            C 2.7096322688430977e-16 2.7096322688430977e-16 4.425165320132997
            PERIODIC XYZ
         &END CELL_REF
         PERIODIC XYZ
      &END CELL
      &KIND Ag
         BASIS_SET ORB DZVP-MOLOPT-PBE-GTH-q11
         ELEMENT Ag
         POTENTIAL GTH GTH-PBE-q11
      &END KIND
      &TOPOLOGY
         COORD_FILE_FORMAT XYZ
         COORD_FILE_NAME aiida.coords.xyz
      &END TOPOLOGY
   &END SUBSYS
&END FORCE_EVAL
```

Listing 4: CP2K input for the deltatest calculation of Ag with PBE for the DZVP MOLOPT basis set and GTH PP for the 1.00 scaling.

C. HFXk implementation details

In the development of the *k point enabled Hartree-Fock Exchange* reference implementation, scaling with regard to threads quickly became a challenge. The existing Γ-point code relied on a complex cost-model calculation together with a load-balancing algorithm to distribute tasks across both MPI and OpenMP parallelisation with both levels essentially being treated on an equal footing. For **k**-point support a different loop structure had to be employed in which the summation over the indices Q and T had to be pulled out of the primitive integrals calculation and only the summation over S remained:

$$K_{\mathrm{x}}^{\mathrm{T}} = -\frac{1}{2} \sum_{\mathrm{Q}} \sum_{\nu\lambda} P_{\nu\lambda}^{\mathrm{Q}} \sum_{\mathrm{S}} (\mu^{0} \nu^{\mathrm{S}} \mid \lambda^{\mathrm{T}} \sigma^{\mathrm{S+Q}}). \tag{C.1}$$

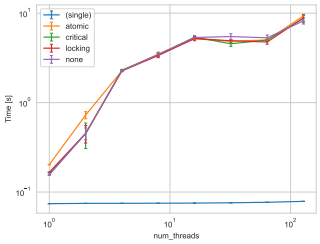
It is this change which makes caching an unresolved challenge with currently available single-node memory as a cache would now have to scale with $N_{\mathrm{img}}^2 \cdot N_{\mathrm{sgf}}^2$, e.g. requiring N_{img}^2 more memory than the periodic Γ-point implementation. This is equates to ~14 TB of memory at double precision for a two-atomic hBN system with ~1100 neighbour cells considered and a *MOLOPT-TZVP* basis.

Using an adequate (possibly lossy) compression scheme could help here, but care must be taken with regard to error propagation since this truncation would now be applied to individual values rather than a sum as before. From the infrastructure point of view, Fabric-attached memory together with a framework like OpenFAM[156] could be a solution as it would allow for memory pooling across nodes.

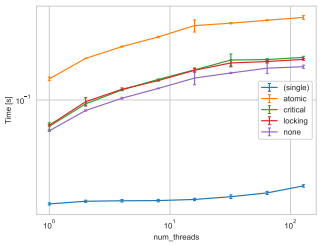
For the current implementation we therefore recalculate all primitive integrals on each iteration. Since we nowadays can assume that OpenMP runtimes, compilers together with modern computer architectures are capable of caching read-only data and that OpenMP threads are lightweight, we went with a simpler parallelisation scheme without active load-balacing but with much smaller tasks. Parallelising over the 10 loops (2 neighbour image, 4 atom, 4 basis set indices) with the `DO PARALLEL COLLAPSE` construct is not possible since it requires iteration boundaries to be known at the entry of the parallel section and no branching between the loops. This constraint is not straightforward satisfiable since the number of sets depends on the atom kind and we maintain several earlier screening checks on the individual iteration levels. Collapsing over only T, i and k to avoid locking to serialise access to the Exchange matrix has lead to limited core utilisation since some tasks take can take much longer than others.

Instead we are using task-based OpenMP parallelisation at the level of the contraction of the primitive integrals with the Fock matrix which leads to excellent core utilisation and associated speedup up to 384 threads (verified on AMD Genoa). To prevent corruption of the Fock matrix we employ a locking scheme. While benchmarking different locking schemes (`omp critical`, `omp_set_lock()`, `omp atomic`) we discovered that some OpenMP implementations do not scale well with larger

Figure C.1.: Timing behaviour of a model example using repeated squares with small tasks, intentionally underemploying cores to explore tasking and scheduling overhead.



(a) The GNU OpenMP implementation shows a larger overhead for a single-value atomic scheme for small number of threads, and a large linear scaling cost associated with threading.



(b) The Cray/LLVM OpenMP implementation shows the same overhead for the atomic scheme and the desired sublinear scaling.

number of threads as shown in Figure C.1 and for smaller systems this can become an issue.

Bibliography

[1] Thomas D. Kühne et al. "CP2K: An Electronic Structure and Molecular Dynamics Software Package - Quickstep: Efficient and Accurate Electronic Structure Calculations." In: *J. Chem. Phys.* 152.19 (May 21, 2020), p. 194103. DOI: 10.1063/5.0007045 (cited on pages v, 32, 47).

[2] Sebastiaan P. Huber et al. "AiiDA 1.0, a Scalable Computational Infrastructure for Automated Reproducible Workflows and Data Provenance." In: *Sci Data* 7.1 (1 Sept. 8, 2020), p. 300. DOI: 10.1038/s41597-020-00638-4 (cited on pages v, 29, 78).

[3] Giovanni Pizzi et al. "How to Verify the Precision of Density-Functional-Theory Implementations via Reproducible and Universal Workflows." In: *Nat Rev Phys* (2023). DOI: 10.1038/s42254-023-00655-3. in printing (cited on pages v, 78, 79).

[4] Tiziano Müller et al. "Survey of Adaptive Containerization Architectures for HPC." In: Supercomputing 23, CANOPIE-HPC. Denver, CO, Nov. 2023 (cited on page v).

[5] Tiziano Müller and Nina Mujkanovic. "Survey and Practical Taxonomy of HPC Workflow Managers." In: *tbd* (2024). in preparation (cited on page v).

[6] Robert G. Parr and Yang Weitao. *Density-Functional Theory of Atoms and Molecules*. International Series of Monographs on Chemistry. New York: Oxford University Press, 1995. 343 pp. (cited on pages 3, 7).

[7] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, July 2, 1996. 484 pp. (cited on page 3).

[8] Robert A. Evarestov. *Quantum Chemistry of Solids: LCAO Treatment of Crystals and Nanostructures*. Vol. 153. Springer Series in Solid-State Sciences. Berlin, Heidelberg: Springer, 2012 (cited on page 3).

[9] Carlos Fiolhais, Fernando Nogueira, and Miguel A. L. Marques, eds. *A Primer in Density Functional Theory*. 1st ed. Vol. 620. Lecture Notes in Physics. Springer Berlin Heidelberg, 2003 (cited on page 3).

[10] C. Pisani, R. Dovesi, and C. Roetti. *Hartree-Fock Ab Initio Treatment of Crystalline Systems*. Ed. by C. Pisani, R. Dovesi, and C. Roetti. Lecture Notes in Chemistry. Berlin, Heidelberg: Springer, 1988 (cited on page 3).

[11] Trygve Helgaker, Poul Jorgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory*. John Wiley & Sons, Aug. 11, 2014. 949 pp. (cited on page 3).

[12] Haoyu S. Yu, Shaohong L. Li, and Donald G. Truhlar. "Perspective: Kohn-Sham Density Functional Theory Descending a Staircase." In: *The Journal of Chemical Physics* 145.13 (Oct. 5, 2016), p. 130901. DOI: 10.1063/1.4963168 (cited on page 8).

[13] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas." In: *Phys. Rev.* 136 (3B Nov. 9, 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864 (cited on page 8).

[14] Lucas O. Wagner et al. "Kohn-Sham Calculations with the Exact Functional." In: *Phys. Rev. B* 90.4 (July 9, 2014), p. 045109. DOI: 10.1103/PhysRevB.90.045109 (cited on page 9).

[15] W. Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects." In: *Phys. Rev.* 140 (4A Nov. 15, 1965), A1133–A1138. DOI: 10.1103/PhysRev.140.A1133 (cited on page 9).

[16] John P. Perdew and Karla Schmidt. "Jacob's Ladder of Density Functional Approximations for the Exchange-Correlation Energy." In: *AIP Conference Proceedings* 577.1 (July 6, 2001), pp. 1–20. DOI: 10.1063/1.1390175 (cited on page 10).

[17] Philipp Haas et al. "Systematic Investigation of a Family of Gradient-Dependent Functionals for Solids." In: *Phys. Rev. B* 81.12 (Mar. 31, 2010), p. 125136. DOI: 10.1103/PhysRevB.81.125136 (cited on page 10).

[18] P. a. M. Dirac. "Note on Exchange Phenomena in the Thomas Atom." In: *Mathematical Proceedings of the Cambridge Philosophical Society* 26.3 (July 1930), pp. 376–385. DOI: 10.1017/S0305004100016108 (cited on page 11).

[19] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. "Generalized Gradient Approximation Made Simple." In: *Phys. Rev. Lett.* 77.18 (Oct. 28, 1996), pp. 3865–3868. DOI: 10.1103/PhysRevLett.77.3865 (cited on page 11).

[20] Elliott H. Lieb and Stephen Oxford. "Improved Lower Bound on the Indirect Coulomb Energy." In: *International Journal of Quantum Chemistry* 19.3 (1981), pp. 427–439. DOI: 10.1002/qua.560190306 (cited on page 12).

[21] Jianmin Tao et al. "Climbing the Density Functional Ladder: Nonempirical Meta–Generalized Gradient Approximation Designed for Molecules and Solids." In: *Phys. Rev. Lett.* 91.14 (Sept. 30, 2003), p. 146401. DOI: 10.1103/PhysRevLett.91.146401 (cited on page 12).

[22] John P. Perdew et al. "Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces." In: *Phys. Rev. Lett.* 100.13 (Apr. 4, 2008), p. 136406. DOI: 10.1103/PhysRevLett.100.136406 (cited on page 12).

[23] John P. Perdew, Matthias Ernzerhof, and Kieron Burke. "Rationale for Mixing Exact Exchange with Density Functional Approximations." In: *J. Chem. Phys.* 105.22 (Dec. 8, 1996), pp. 9982–9985. DOI: 10.1063/1.472933 (cited on page 13).

[24] J. Harris. "Adiabatic-Connection Approach to Kohn-Sham Theory." In: *Phys. Rev. A* 29.4 (Apr. 1, 1984), pp. 1648–1659. DOI: 10.1103/PhysRevA.29.1648 (cited on page 13).

[25] Marco Bernardi. "Physical Origin of the One-Quarter Exact Exchange in Density Functional Theory." In: *J. Phys.: Condens. Matter* 32.38 (June 2020), p. 385501. DOI: 10.1088/1361-648X/ab9409 (cited on page 13).

[26] Axel D. Becke. "A New Mixing of Hartree–Fock and Local Density-functional Theories." In: *The Journal of Chemical Physics* 98.2 (Jan. 15, 1993), pp. 1372–1377. DOI: 10.1063/1.464304 (cited on page 13).

[27] Axel D. Becke. "Density-functional Thermochemistry. III. The Role of Exact Exchange." In: *The Journal of Chemical Physics* 98.7 (Apr. 1, 1993), pp. 5648–5652. DOI: 10.1063/1.464913 (cited on page 13).

[28] Peter Elliott and Kieron Burke. *Non-Empirical 'derivation' of B88 Exchange Functional*. Feb. 26, 2009. DOI: 10.48550/arXiv.0902.1491. URL: http://arxiv.org/abs/0902.1491 (visited on 10/01/2023). preprint (cited on pages 13, 25).

[29] Gerald Lippert, Jürg Hutter, and Michele Parrinello. "A Hybrid Gaussian and Plane Wave Density Functional Scheme." In: *Molecular Physics* 92.3 (Oct. 1, 1997), pp. 477–488. DOI: 10.1080/002689797170220 (cited on page 14).

[30] Peter Schwerdtfeger. "The Pseudopotential Approximation in Electronic Structure Theory." In: *ChemPhysChem* 12.17 (2011), pp. 3143–3155. DOI: 10.1002/cphc.201100387 (cited on page 15).

[31] Michael Dolg and Xiaoyan Cao. "Relativistic Pseudopotentials: Their Development and Scope of Applications." In: *Chem. Rev.* 112.1 (Jan. 11, 2012), pp. 403–480. DOI: 10.1021/cr2001383 (cited on page 15).

[32] D. R. Hamann, M. Schlüter, and C. Chiang. "Norm-Conserving Pseudopotentials." In: *Phys. Rev. Lett.* 43.20 (Nov. 12, 1979), pp. 1494–1497. DOI: 10.1103/PhysRevLett.43.1494 (cited on page 15).

[33] S. Goedecker, M. Teter, and J. Hutter. "Separable Dual-Space Gaussian Pseudopotentials." In: *Phys. Rev. B* 54.3 (July 15, 1996), pp. 1703–1710. DOI: 10.1103/PhysRevB.54.1703 (cited on page 15).

[34] C. Hartwigsen, S. Goedecker, and J. Hutter. "Relativistic Separable Dual-Space Gaussian Pseudopotentials from H to Rn." In: *Phys. Rev. B* 58.7 (Aug. 15, 1998), pp. 3641–3662. DOI: 10.1103/PhysRevB.58.3641 (cited on page 15).

[35] E. Wimmer et al. "Full-Potential Self-Consistent Linearized-Augmented-Plane-Wave Method for Calculating the Electronic Structure of Molecules and Surfaces: ${\mathrm{O}}_{2}$ Molecule." In: *Phys. Rev. B* 24.2 (July 15, 1981), pp. 864–875. DOI: 10.1103/PhysRevB.24.864 (cited on page 16).

[36] Gerald Lippert, Jürg Hutter, and Michele Parrinello. "The Gaussian and Augmented-Plane-Wave Density Functional Method for Ab Initio Molecular Dynamics Simulations." In: *Theor Chem Acc* 103.2 (Dec. 1, 1999), pp. 124–140. DOI: 10.1007/s002140050523 (cited on pages 16, 53).

[37] Matthias Krack and Michele Parrinello. "All-Electron Ab-Initio Molecular Dynamics." In: *Phys. Chem. Chem. Phys.* 2.10 (Jan. 1, 2000), pp. 2105–2112. DOI: 10.1039/B001167N (cited on page 16).

[38] Marcella Iannuzzi et al. "Ground and Excited State Density Functional Calculations with the Gaussian and Augmented-Plane-Wave Method." In: *CHIMIA* 59.7-8 (7-8 July 1, 2005), pp. 499–499. DOI: 10.2533/000942905777676164 (cited on page 16).

[39] P. E. Blöchl. "Projector Augmented-Wave Method." In: *Phys. Rev. B* 50.24 (Dec. 15, 1994), pp. 17953–17979. DOI: 10.1103/PhysRevB.50.17953 (cited on page 16).

[40] Stig Rune Jensen et al. "The Elephant in the Room of Density Functional Theory Calculations." In: *J. Phys. Chem. Lett.* 8.7 (Apr. 6, 2017), pp. 1449–1457. DOI: 10.1021/acs.jpclett.7b00255 (cited on pages 16, 47).

[41] Frank Jensen. "How Large Is the Elephant in the Density Functional Theory Room?" In: *J. Phys. Chem. A* 121.32 (Aug. 17, 2017), pp. 6104–6107. DOI: 10.1021/acs.jpca.7b04760 (cited on pages 16, 47).

[42] Richard C. Raffenetti. "General Contraction of Gaussian Atomic Orbitals: Core, Valence, Polarization, and Diffuse Basis Sets; Molecular Integral Evaluation." In: *The Journal of Chemical Physics* 58.10 (Aug. 22, 2003), pp. 4452–4458. DOI: 10.1063/1.1679007 (cited on page 16).

[43] Michael J. Frisch, John A. Pople, and J. Stephen Binkley. "Self-consistent Molecular Orbital Methods 25. Supplementary Functions for Gaussian Basis Sets." In: *The Journal of Chemical Physics* 80.7 (Apr. 1, 1984), pp. 3265–3269. DOI: 10.1063/1.447079 (cited on page 16).

[44] J. Dziedzic, Q. Hill, and C.-K. Skylaris. "Linear-Scaling Calculation of Hartree-Fock Exchange Energy with Non-Orthogonal Generalised Wannier Functions." In: *The Journal of Chemical Physics* 139.21 (Dec. 2, 2013), p. 214103. DOI: 10.1063/1.4832338 (cited on page 17).

[45] Stefan Goedecker. "Linear Scaling Electronic Structure Methods." In: *Rev. Mod. Phys.* 71.4 (July 1, 1999), pp. 1085–1123. DOI: 10.1103/RevModPhys.71.1085 (cited on page 17).

[46] Joost VandeVondele and Jürg Hutter. "Gaussian Basis Sets for Accurate Calculations on Molecular Systems in Gas and Condensed Phases." In: *JCP* 127.11 (Sept. 18, 2007), p. 114105. DOI: 10.1063/1.2770708 (cited on pages 17, 27, 32, 47).

[47] Gerd Berghold, Michele Parrinello, and Jürg Hutter. "Polarized Atomic Orbitals for Linear Scaling Methods." In: *The Journal of Chemical Physics* 116.5 (Feb. 1, 2002), pp. 1800–1810. DOI: 10.1063/1.1431270 (cited on page 17).

[48] Michael S Lee and Martin Head-Gordon. "Absolute and Relative Energies from Polarized Atomic Orbital Self-Consistent Field Calculations and a Second Order Correction.: Convergence with Size and Composition of the Secondary Basis." In: *Computers & Chemistry* 24.3 (May 1, 2000), pp. 295–301. DOI: 10.1016/S0097-8485(99)00086-8 (cited on page 17).

[49] Michael S. Lee and Martin Head-Gordon. "Polarized Atomic Orbitals for Self-Consistent Field Electronic Structure Calculations." In: *The Journal of Chemical Physics* 107.21 (Dec. 1, 1997), pp. 9085–9095. DOI: 10.1063/1.475199 (cited on page 17).

[50] C. M. Reeves and M. C. Harrison. "Use of Gaussian Functions in the Calculation of Wavefunctions for Small Molecules. II. The Ammonia Molecule." In: *The Journal of Chemical Physics* 39.1 (1963), pp. 11–17. DOI: 10.1063/1.1733984 (cited on page 17).

[51] David F. Feller and Klaus Ruedenberg. "Systematic Approach to Extended Even-Tempered Orbital Bases for Atomic and Molecular Calculations." In: *Theoret. Chim. Acta* 52.3 (Sept. 1, 1979), pp. 231–251. DOI: 10.1007/BF00547681 (cited on pages 17, 47).

[52] Ira Cherkes, Shachar Klaiman, and Nimrod Moiseyev. "Spanning the Hilbert Space with an Even Tempered Gaussian Basis Set." In: *International Journal of Quantum Chemistry* 109.13 (2009), pp. 2996–3002. DOI: 10.1002/qua.22090 (cited on pages 17, 47).

[53] Florian Weigend and Reinhart Ahlrichs. "Balanced Basis Sets of Split Valence, Triple Zeta Valence and Quadruple Zeta Valence Quality for H to Rn: Design and Assessment of Accuracy." In: *Phys. Chem. Chem. Phys.* 7.18 (Aug. 30, 2005), pp. 3297–3305. DOI: 10.1039/B508541A (cited on pages 17, 28, 47, 48, 53, 81).

[54] Florian Weigend. "Accurate Coulomb-fitting Basis Sets for H to Rn." In: *Phys. Chem. Chem. Phys.* 8.9 (Feb. 23, 2006), pp. 1057–1065. DOI: 10.1039/B515623H (cited on pages 17, 53).

[55] H. Bernhard Schlegel and Michael J. Frisch. "Transformation between Cartesian and Pure Spherical Harmonic Gaussians." In: *International Journal of Quantum Chemistry* 54.2 (1995), pp. 83–87. DOI: 10.1002/qua.560540202 (cited on page 17).

[56] R. Dovesi et al. "The Periodic Hartree-Fock Method and Its Implementation in the Crystal Code." In: *physica status solidi (b)* 217.1 (2000), pp. 63–88. DOI: 10.1002/(SICI)1521-3951(200001)217:1<63::AID-PSSB63>3.0.CO;2-F (cited on page 19).

[57] Robert A. Evarestov. "Hartree–Fock LCAO Method for Periodic Systems." In: *Quantum Chemistry of Solids: LCAO Treatment of Crystals and Nanostructures*. Ed. by Robert A. Evarestov. Springer Series in Solid-State Sciences. Berlin, Heidelberg: Springer, 2012, pp. 109–155. DOI: 10.1007/978-3-642-30356-2_4 (cited on page 19).

[58] Manuel Guidon et al. "Ab Initio Molecular Dynamics Using Hybrid Density Functionals." In: *JCP* 128.21 (June 4, 2008), p. 214104. DOI: 10.1063/1.2931945 (cited on page 20).

[59] C. C. J. Roothaan. "New Developments in Molecular Orbital Theory." In: *Rev. Mod. Phys.* 23.2 (Apr. 1, 1951), pp. 69–89. DOI: 10.1103/RevModPhys.23.69 (cited on page 21).

[60] Marco Häser and Reinhart Ahlrichs. "Improvements on the Direct SCF Method." In: *Journal of Computational Chemistry* 10.1 (1989), pp. 104–111. DOI: 10.1002/jcc.540100111 (cited on pages 21, 69).

[61] Manuel Guidon, Jürg Hutter, and Joost VandeVondele. "Auxiliary Density Matrix Methods for Hartree–Fock Exchange Calculations." In: *J. Chem. Theory Comput.* 6.8 (Aug. 10, 2010), pp. 2348–2364. DOI: 10.1021/ct1002225 (cited on pages 22, 24).

[62] R. McWeeny. "Some Recent Advances in Density Matrix Theory." In: *Rev. Mod. Phys.* 32.2 (Apr. 1, 1960), pp. 335–369. DOI: 10.1103/RevModPhys.32.335 (cited on page 24).

[63] R. F. Rinehart. "The Derivative of a Matric Function." In: *Proc. Amer. Math. Soc.* 7.1 (1956), pp. 2–5. DOI: 10.1090/S0002-9939-1956-0077499-X (cited on page 24).

[64] Patrick Merlot et al. "Charge-Constrained Auxiliary-Density-Matrix Methods for the Hartree–Fock Exchange Contribution." In: *The Journal of Chemical Physics* 141.9 (Sept. 4, 2014), p. 094104. DOI: 10.1063/1.4894267 (cited on pages 25, 26).

[65] Nicholas C. Handy and Aron J. Cohen. "Left-Right Correlation Energy." In: *Molecular Physics* 99.5 (Mar. 10, 2001), pp. 403–412. DOI: 10.1080/00268970010018431 (cited on page 25).

[66] Wee-Meng Hoe, Aron J. Cohen, and Nicholas C. Handy. "Assessment of a New Local Exchange Functional OPTX." In: *Chemical Physics Letters* 341.3–4 (June 22, 2001), pp. 319–328. DOI: 10.1016/S0009-2614(01)00581-4 (cited on page 25).

[67] Charles H. Patterson. *Density Fitting in Periodic Systems: Application to TDHF in Diamond and Oxides*. arXiv.org. May 19, 2020. DOI: 10.1063/5.0014106. URL: https://arxiv.org/abs/2005.09291v1 (visited on 11/01/2023) (cited on page 26).

[68] M. Krack. "Pseudopotentials for H to Kr Optimized for Gradient-Corrected Exchange-Correlation Functionals." In: *Theor Chem Acc* 114.1 (Sept. 1, 2005), pp. 145–152. DOI: 10.1007/s00214-005-0655-y (cited on page 27).

[69] Takahito Nakajima and Kimihiko Hirao. "The Higher-Order Douglas–Kroll Transformation." In: *The Journal of Chemical Physics* 113.18 (Nov. 8, 2000), pp. 7786–7789. DOI: 10.1063/1.1316037 (cited on page 27).

[70] Markus Reiher. "Douglas–Kroll–Hess Theory: A Relativistic Electrons-Only Theory for Chemistry." In: *Theor Chem Acc* 116.1 (Aug. 1, 2006), pp. 241–252. DOI: 10.1007/s00214-005-0003-2 (cited on page 27).

[71] M. J. D. Powell. "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives." In: *The Computer Journal* 7.2 (Jan. 1, 1964), pp. 155–162. DOI: 10.1093/comjnl/7.2.155 (cited on page 27).

[72] Tiziano Müller, Nina Mujkanovic, and Alfio Lazzaro. "HPC Workflow Management - Current State and Trends." Presentation. PASC22 (Basel). June 28, 2022 (cited on pages 28, 32).

[73] Giovanni Pizzi et al. "AiiDA: Automated Interactive Infrastructure and Database for Computational Science." In: *Comp. Mat. Science* 111 (Jan. 2016), pp. 218–230. DOI: 10.1016/j.commatsci.2015.09.013 (cited on pages 29, 35).

[74] Ask Hjorth Larsen et al. "The Atomic Simulation Environment—a Python Library for Working with Atoms." In: *J. Phys.: Condens. Matter* 29.27 (June 2017), p. 273002. DOI: 10.1088/1361-648X/aa680e (cited on page 29).

[75] Shyue Ping Ong et al. "Python Materials Genomics (Pymatgen): A Robust, Open-Source Python Library for Materials Analysis." In: *Computational Materials Science* 68 (Feb. 1, 2013), pp. 314–319. DOI: 10.1016/j.commatsci.2012.10.028 (cited on page 29).

[76] M. J. van Setten et al. "Automation Methodologies and Large-Scale Validation for GW: Towards High-Throughput GW Calculations." In: *Phys. Rev. B* 96.15 (Oct. 17, 2017), p. 155207. DOI: 10.1103/PhysRevB.96.155207 (cited on page 29).

[77] Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures." PhD thesis. Irvine: University of California, 2000 (cited on page 30).

[78] Rafael Ferreira da Silva et al. "A Community Roadmap for Scientific Workflows Research and Development." In: *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)*. 2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS). Nov. 2021, pp. 81–90. DOI: 10.1109/WORKS54523.2021.00016 (cited on page 32).

[79] Laurens Versluis and Alexandru Iosup. "A Survey of Domains in Workflow Scheduling in Computing Infrastructures: Community and Keyword Analysis, Emerging Trends, and Taxonomies." In: *Future Generation Computer Systems* 123 (Oct. 1, 2021), pp. 156–177. DOI: 10.1016/j.future.2021.04.009 (cited on page 32).

[80] Atsushi Togo and Isao Tanaka. "First Principles Phonon Calculations in Materials Science." In: *Scripta Materialia* 108 (Supplement C Nov. 1, 2015), pp. 1–5. DOI: 10.1016/j.scriptamat.2015.07.021 (cited on pages 35, 37).

[81] Laura E. Ratcliff et al. "Flexibilities of Wavelets as a Computational Basis Set for Large-Scale Electronic Structure Calculations." In: *J. Chem. Phys.* 152.19 (May 21, 2020), p. 194110. DOI: 10.1063/5.0004792 (cited on page 41).

[82] *Gaussian˜16 {R}evision {C}.01*. In collab. with M. J. Frisch et al. Gaussian Inc. Wallingford CT, 2016 (cited on pages 41, 48).

[83] Daniel Lemire and Wojciech Muła. "Transcoding Billions of Unicode Characters per Second with SIMD Instructions." In: *Software: Practice and Experience* 52.2 (2022), pp. 555–575. DOI: 10.1002/spe.3036 (cited on page 42).

[84] J. Grant Hill. "Gaussian Basis Sets for Molecular Applications." In: *International Journal of Quantum Chemistry* 113.1 (2013), pp. 21–34. DOI: 10.1002/qua.24355 (cited on page 47).

[85] Frank Jensen. "Atomic Orbital Basis Sets." In: *WIREs Comput Mol Sci* 3.3 (May 1, 2013), pp. 273–295. DOI: 10.1002/wcms.1123 (cited on page 47).

[86] R. Ditchfield, W. J. Hehre, and J. A. Pople. "Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules." In: *J. Chem. Phys.* 54.2 (Jan. 15, 1971), pp. 724–728. DOI: 10.1063/1.1674902 (cited on page 47).

[87] Thom H. Dunning. "Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron through Neon and Hydrogen." In: *J. Chem. Phys.* 90.2 (Jan. 15, 1989), pp. 1007–1023. DOI: 10.1063/1.456153 (cited on page 47).

[88] Benjamin P. Pritchard et al. "New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community." In: *J. Chem. Inf. Model.* 59.11 (Nov. 25, 2019), pp. 4814–4820. DOI: 10.1021/acs.jcim.9b00725 (cited on page 47).

[89] Marcel Müller, Andreas Hansen, and Stefan Grimme. "ωB97X-3c: A Composite Range-Separated Hybrid DFT Method with a Molecule-Optimized Polarized Valence Double-ζ Basis Set." In: *J. Chem. Phys.* 158.1 (Jan. 7, 2023), p. 014103. DOI: 10.1063/5.0133026 (cited on page 47).

[90] Hong-Zhou Ye and Timothy C. Berkelbach. "Correlation-Consistent Gaussian Basis Sets for Solids Made Simple." In: *J. Chem. Theory Comput.* 18.3 (Mar. 8, 2022), pp. 1595–1606. DOI: 10.1021/acs.jctc.1c01245 (cited on page 47).

[91] Lars Goerigk and Stefan Grimme. "A Thorough Benchmark of Density Functional Methods for General Main Group Thermochemistry, Kinetics, and Noncovalent Interactions." In: *Phys. Chem. Chem. Phys.* 13.14 (Apr. 14, 2011), pp. 6670–6688. DOI: 10.1039/C0CP02984J (cited on page 47).

[92] Hannes Konrad Buchholz and Matthias Stein. "Accurate Lattice Energies of Organic Molecular Crystals from Periodic Turbomole Calculations." In: *Journal of Computational Chemistry* 39.19 (2018), pp. 1335–1343. DOI: 10.1002/jcc.25205 (cited on page 48).

[93] Lars Goerigk and Stefan Grimme. "A General Database for Main Group Thermochemistry, Kinetics, and Noncovalent Interactions – Assessment of Common and Reparameterized (Meta-)GGA Density Functionals." In: *J. Chem. Theory Comput.* 6.1 (Jan. 12, 2010), pp. 107–126. DOI: 10.1021/ct900489g (cited on page 48).

[94] Lars Goerigk et al. "A Look at the Density Functional Theory Zoo with the Advanced GMTKN55 Database for General Main Group Thermochemistry, Kinetics and Noncovalent Interactions." In: *Phys. Chem. Chem. Phys.* 19.48 (Dec. 13, 2017), pp. 32184–32215. DOI: 10.1039/C7CP04913G (cited on page 48).

[95] Stig Rune Jensen et al. *GGA-PBE and Hybrid-PBE0 Energies and Dipole Moments with MRChem, FHI-aims, NWChem and ELK*. Version 3. DataverseNO, Sept. 28, 2023. DOI: 10.18710/0EM0EL (cited on page 48).

[96] M. J. van Setten et al. *The PseudoDojo: Training and Grading a 85 Element Optimized Norm-Conserving Pseudopotential Table*. arXiv.org. Oct. 27, 2017. DOI: 10.1016/j.cpc.2018.01.012. URL: https://arxiv.org/abs/1710.10138v2 (visited on 11/01/2023) (cited on page 48).

[97] Igor Ying Zhang et al. "Main-Group Test Set for Materials Science and Engineering with User-Friendly Graphical Tools for Error Analysis: Systematic Benchmark of the Numerical and Intrinsic Errors in State-of-the-Art Electronic-Structure Approximations." In: *New J. Phys.* 21.1 (Jan. 2019), p. 013025. DOI: 10.1088/1367-2630/aaf751 (cited on page 48).

[98] Igor Ying Zhang et al. *Test Set for Materials Science and Engineering with User-Friendly Graphic Tools for Error Analysis: Systematic Benchmark of the Numerical and Intrinsic Errors in State-of-the-Art Electronic-Structure Approximations*. arXiv.org. Aug. 29, 2018. DOI: 10.1088/1367-2630/aaf751. URL: https://arxiv.org/abs/1808.09780v1 (visited on 11/01/2023) (cited on page 48).

[99] Pierpaolo Morgante and Roberto Peverati. "ACCDB: A Collection of Chemistry Databases for Broad Computational Purposes." In: *Journal of Computational Chemistry* 40.6 (2019), pp. 839–848. DOI: 10.1002/jcc.25761 (cited on page 48).

[100] Kurt Lejaeghere et al. "Reproducibility in Density Functional Theory Calculations of Solids." In: *Science* 351.6280 (Mar. 25, 2016), aad3000. DOI: 10.1126/science.aad3000 (cited on pages 48, 50, 52).

[101] Kurt Lejaeghere et al. "Error Estimates for Solid-State Density-Functional Theory Predictions: An Overview by Means of the Ground-State Elemental Crystals." In: *Crit. Rev. Solid State* 39.1 (Jan. 1, 2014), pp. 1–24. DOI: 10.1080/10408436.2013.772503 (cited on pages 48, 50, 73).

[102] Francis Birch. "Finite Elastic Strain of Cubic Crystals." In: *Phys. Rev.* 71.11 (June 1, 1947), pp. 809–824. DOI: 10.1103/PhysRev.71.809 (cited on page 48).

[103] Gianluca Prandini et al. *A Standard Solid State Pseudopotentials (SSSP) Library Optimized for Precision and Efficiency (Version 1.1, Data Download).* Materials Cloud, Nov. 8, 2018. DOI: 10.24435/MATERIALSCLOUD:2018.0001/V3 (cited on page 49).

[104] Gianluca Prandini et al. "Precision and Efficiency in Solid-State Pseudopotential Calculations." In: *npj Comput Mater* 4.1 (Dec. 2018), p. 72. DOI: 10.1038/s41524-018-0127-2 (cited on page 49).

[105] Andris Gulans. "NoMaD Repository Entry." In: (2016). DOI: 10.17172/NOMAD/2016.09.20-1 (cited on page 50).

[106] Kenneth G. Dyall and Erik van Lenthe. "Relativistic Regular Approximations Revisited: An Infinite-Order Relativistic Approximation." In: *The Journal of Chemical Physics* 111.4 (July 22, 1999), pp. 1366–1372. DOI: 10.1063/1.479395 (cited on page 50).

[107] Susi Lehtola et al. "Recent Developments in Libxc — A Comprehensive Library of Functionals for Density Functional Theory." In: *SoftwareX* 7 (Jan. 1, 2018), pp. 1–5. DOI: 10.1016/j.softx.2017.11.002 (cited on page 50).

[108] Tiziano M.A. Müller. *CP2K Calculations for Deltatest Structures with revised MOLOPT Basis Sets and Pseudopotentials.* DOI: 10.5281/zenodo.7254854 (cited on pages 50, 85).

[109] Joachim Laun and Thomas Bredow. "BSSE-corrected Consistent Gaussian Basis Sets of Triple-Zeta Valence with Polarization Quality of the Sixth Period for Solid-State Calculations." In: *Journal of Computational Chemistry* 42.15 (2021), pp. 1064–1072. DOI: 10.1002/jcc.26521 (cited on page 50).

[110] Joachim Laun and Thomas Bredow. "BSSE-corrected Consistent Gaussian Basis Sets of Triple-Zeta Valence with Polarization Quality of the Fifth Period for Solid-State Calculations." In: *Journal of Computational Chemistry* 43.12 (2022), pp. 839–846. DOI: 10.1002/jcc.26839 (cited on page 50).

[111] Michael F. Peintinger, Daniel Vilela Oliveira, and Thomas Bredow. "Consistent Gaussian Basis Sets of Triple-Zeta Valence with Polarization Quality for Solid-State Calculations." In: *J. Comput. Chem.* 34.6 (2013), pp. 451–459. DOI: 10.1002/jcc.23153 (cited on pages 50, 55).

[112] Daniel Vilela Oliveira et al. "BSSE-correction Scheme for Consistent Gaussian Basis Sets of Double- and Triple-Zeta Valence with Polarization Quality for Solid-State Calculations." In: *Journal of Computational Chemistry* 40.27 (2019), pp. 2364–2376. DOI: 10.1002/jcc.26013 (cited on pages 50, 55, 69).

[113] Santanu Saha. "Soft and Accurate Norm Conserving Pseudopotentials and Their Application for Structure Prediction." Thesis. University_of_Basel, 2017. 1 Online-Ressource (vi, 127 Seiten). DOI: 10.5451/unibas-006755697 (cited on page 51).

[114] Jürg Hutter and Tiziano M.A. Müller. *CP2K Calculations for Structures of the Small Molecule Database with revised MOLOPT Basis Sets and Pseudopotentials.* DOI: 10.5281/zenodo.7841955 (cited on pages 57, 81).

[115] J. C. Slater. "A Simplification of the Hartree-Fock Method." In: *Phys. Rev.* 81.3 (Feb. 1, 1951), pp. 385–390. DOI: 10.1103/PhysRev.81.385 (cited on page 67).

[116] Douglas Rayner Hartree and W. Hartree. "Self-Consistent Field, with Exchange, for Beryllium." In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 150.869 (Jan. 5, 1935), pp. 9–33. DOI: 10.1098/rspa.1935.0085 (cited on page 67).

[117] Pierre Carrier, Stefan Rohra, and Andreas Görling. "General Treatment of the Singularities in Hartree-Fock and Exact-Exchange Kohn-Sham Methods for Solids." In: *Phys. Rev. B* 75.20 (May 29, 2007), p. 205126. DOI: 10.1103/PhysRevB.75.205126 (cited on page 67).

[118] James Spencer and Ali Alavi. "Efficient Calculation of the Exact Exchange Energy in Periodic Systems Using a Truncated Coulomb Potential." In: *Phys. Rev. B* 77.19 (May 23, 2008), p. 193110. DOI: 10.1103/PhysRevB.77.193110 (cited on page 67).

[119] Roberto Dovesi et al. "Quantum-Mechanical Condensed Matter Simulations with CRYSTAL." In: *WIREs Computational Molecular Science* 8.4 (2018), e1360. DOI: 10.1002/wcms.1360 (cited on page 67).

[120] Sree Ganesh Balasubramani et al. "TURBOMOLE: Modular Program Suite for Ab Initio Quantum-Chemical and Condensed-Matter Simulations." In: *J. Chem. Phys.* 152.18 (May 14, 2020), p. 184107. DOI: 10.1063/5.0004635 (cited on page 67).

[121] Florian Weigend. "A Fully Direct RI-HF Algorithm: Implementation, Optimised Auxiliary Basis Sets, Demonstration of Accuracy and Efficiency." In: *Phys. Chem. Chem. Phys.* 4.18 (Sept. 6, 2002), pp. 4285–4291. DOI: 10.1039/B204199P (cited on page 68).

[122] Florian Weigend. "Hartree–Fock Exchange Fitting Basis Sets for H to Rn †." In: *Journal of Computational Chemistry* 29.2 (2008), pp. 167–175. DOI: 10.1002/jcc.20702 (cited on page 68).

[123] Peter Wind, Trygve Helgaker, and Wim Klopper. "Efficient Evaluation of One-Center Three-Electron Gaussian Integrals." In: *Theor Chem Acc* 106.4 (Sept. 1, 2001), pp. 280–286. DOI: 10.1007/s002140100281 (cited on page 68).

[124] Xinguo Ren et al. "Resolution-of-Identity Approach to Hartree–Fock, Hybrid Density Functionals, RPA, MP2 and GW with Numeric Atom-Centered Orbital Basis Functions." In: *New J. Phys.* 14.5 (May 2012), p. 053020. DOI: 10.1088/1367-2630/14/5/053020 (cited on page 68).

[125] Mauro Del Ben, Jürg Hutter, and Joost VandeVondele. "Electron Correlation in the Condensed Phase from a Resolution of Identity Approach Based on the Gaussian and Plane Waves Scheme." In: *J. Chem. Theory Comput.* 9.6 (June 11, 2013), pp. 2654–2671. DOI: 10.1021/ct4002202 (cited on page 68).

[126] Joonho Lee et al. "Faster Exact Exchange for Solids via Occ-RI-K: Application to Combinatorially Optimized Range-Separated Hybrid Functionals for Simple Solids with Pseudopotentials Near the Basis Set Limit." In: *J. Chem. Theory Comput.* 18.12 (Dec. 13, 2022), pp. 7336–7349. DOI: 10.1021/acs.jctc.2c00742 (cited on pages 68, 76).

[127] Adam Rettig, Joonho Lee, and Martin Head-Gordon. *Even Faster Exact Exchange for Solids via Tensor Hypercontraction*. Apr. 11, 2023. DOI: 10.48550/arXiv.2304.05505. URL: http://arxiv.org/abs/2304.05505 (visited on 08/09/2023). preprint (cited on pages 68, 76).

[128] Christopher A. White and Martin Head-Gordon. "A J Matrix Engine for Density Functional Theory Calculations." In: *The Journal of Chemical Physics* 104.7 (Feb. 15, 1996), pp. 2620–2629. DOI: 10.1063/1.470986 (cited on page 68).

[129] Yihan Shao and Martin Head-Gordon. "An Improved J Matrix Engine for Density Functional Theory Calculations." In: *Chemical Physics Letters* 323.5 (June 23, 2000), pp. 425–433. DOI: 10.1016/S0009-2614(00)00524-8 (cited on page 68).

[130] Yihan Shao, Christopher A. White, and Martin Head-Gordon. "Efficient Evaluation of the Coulomb Force in Density-Functional Theory Calculations." In: *The Journal of Chemical Physics* 114.15 (Apr. 15, 2001), pp. 6572–6577. DOI: 10.1063/1.1357441 (cited on page 68).

[131] Samuel Manzer et al. "Fast, Accurate Evaluation of Exact Exchange: The Occ-RI-K Algorithm." In: *The Journal of Chemical Physics* 143.2 (July 14, 2015), p. 024113. DOI: 10.1063/1.4923369 (cited on pages 68, 76).

[132] Sandeep Sharma, Alec F. White, and Gregory Beylkin. "Fast Exchange with Gaussian Basis Set Using Robust Pseudospectral Method." In: *J. Chem. Theory Comput.* 18.12 (Dec. 13, 2022), pp. 7306–7320. DOI: 10.1021/acs.jctc.2c00720 (cited on page 68).

[133] Edward G. Hohenstein et al. "Communication: Tensor Hypercontraction. III. Least-squares Tensor Hypercontraction for the Determination of Correlated Wavefunctions." In: *The Journal of Chemical Physics* 137.22 (Dec. 11, 2012), p. 221101. DOI: 10.1063/1.4768241 (cited on page 68).

[134] Edward G. Hohenstein, Robert M. Parrish, and Todd J. Martínez. "Tensor Hypercontraction Density Fitting. I. Quartic Scaling Second- and Third-Order Møller-Plesset Perturbation Theory." In: *The Journal of Chemical Physics* 137.4 (July 23, 2012), p. 044103. DOI: 10.1063/1.4732310 (cited on page 68).

[135] Robert M. Parrish et al. "Tensor Hypercontraction. II. Least-squares Renormalization." In: *The Journal of Chemical Physics* 137.22 (Dec. 11, 2012), p. 224106. DOI: 10.1063/1.4768233 (cited on page 68).

[136] Sverre Froyen. "Brillouin-Zone Integration by Fourier Quadrature: Special Points for Superlattice and Supercell Calculations." In: *Phys. Rev. B* 39.5 (Feb. 15, 1989), pp. 3168–3172. DOI: 10.1103/PhysRevB.39.3168 (cited on page 69).

[137] Wei Ku, Tom Berlijn, and Chi-Cheng Lee. "Unfolding First-Principles Band Structures." In: *Phys. Rev. Lett.* 104.21 (May 24, 2010), p. 216401. DOI: 10.1103/PhysRevLett.104.216401 (cited on page 69).

[138] M. X. Chen et al. "Effects of Magnetic Dopants in $({\mathrm{Li}}_{0.8}{M}_{0.2}\mathrm{OH})\mathrm{FeSe}$ ($M=\mathrm{Fe}$, Mn, Co): Density Functional Theory Study Using a Band Unfolding Technique." In: *Phys. Rev. B* 96.24 (Dec. 8, 2017), p. 245111. DOI: 10.1103/PhysRevB.96.245111 (cited on page 69).

[139] Hendrik J. Monkhorst and James D. Pack. "Special Points for Brillouin-zone Integrations." In: *Phys. Rev. B* 13.12 (June 15, 1976), pp. 5188–5192. DOI: 10.1103/PhysRevB.13.5188 (cited on page 69).

[140] Dieter Cremer and Jürgen Gauss. "An Unconventional SCF Method for Calculations on Large Molecules." In: *Journal of Computational Chemistry* 7.3 (1986), pp. 274–282. DOI: 10.1002/jcc.540070305 (cited on page 69).

[141] Daniel S. Lambrecht and Christian Ochsenfeld. "Multipole-Based Integral Estimates for the Rigorous Description of Distance Dependence in Two-Electron Integrals." In: *The Journal of Chemical Physics* 123.18 (Nov. 7, 2005), p. 184101. DOI: 10.1063/1.2079967 (cited on pages 69, 79).

[142] Daniel S. Lambrecht, Bernd Doser, and Christian Ochsenfeld. "Rigorous Integral Screening for Electron Correlation Methods." In: *The Journal of Chemical Physics* 123.18 (Nov. 7, 2005), p. 184102. DOI: 10.1063/1.2079987 (cited on page 69).

[143] Simon A. Maurer et al. "Distance-Dependent Schwarz-based Integral Estimates for Two-Electron Integrals: Reliable Tightness vs. Rigorous Upper Bounds." In: *The Journal of Chemical Physics* 136.14 (Apr. 11, 2012), p. 144107. DOI: 10.1063/1.3693908 (cited on pages 69, 79).

[144] Aliaksandr V. Krukau et al. "Influence of the Exchange Screening Parameter on the Performance of Screened Hybrid Functionals." In: *The Journal of Chemical Physics* 125.22 (Dec. 13, 2006), p. 224106. DOI: 10.1063/1.2404663 (cited on page 73).

[145] Nick X. Wang and Angela K. Wilson. "The Behavior of Density Functionals with Respect to Basis Set. I. The Correlation Consistent Basis Sets." In: *The Journal of Chemical Physics* 121.16 (Oct. 12, 2004), pp. 7632–7646. DOI: 10.1063/1.1792071 (cited on page 74).

[146] Nick X. Wang and Angela K. Wilson *. "Behaviour of Density Functionals with Respect to Basis Set: II. Polarization Consistent Basis Sets." In: *Molecular Physics* 103.2-3 (Jan. 20, 2005), pp. 345–358. DOI: 10.1080/00268970512331317264 (cited on page 74).

[147] Andreas Irmler, Asbjörn M. Burow, and Fabian Pauly. "Robust Periodic Fock Exchange with Atom-Centered Gaussian Basis Sets." In: *JCTC* 14.9 (Sept. 11, 2018), pp. 4567–4580. DOI: 10.1021/acs.jctc.8b00122 (cited on page 75).

[148] Manuel Guidon, Jürg Hutter, and Joost VandeVondele. "Robust Periodic Hartree–Fock Exchange for Large-Scale Simulations Using Gaussian Basis Sets." In: *JCTC* 5.11 (Nov. 10, 2009), pp. 3010–3021. DOI: 10.1021/ct900494g (cited on page 75).

[149] Lin Lin. "Adaptively Compressed Exchange Operator." In: *J. Chem. Theory Comput.* 12.5 (May 10, 2016), pp. 2242–2249. DOI: 10.1021/acs.jctc.6b00092 (cited on page 76).

[150] Qiming Sun. "Exact Exchange with Range-Separated Algorithm for Thermodynamic Limit of Periodic Hartree–Fock Theory." In: *The Journal of Chemical Physics* 159.2 (July 10, 2023), p. 024108. DOI: 10.1063/5.0155815 (cited on page 76).

[151] Sebastiaan P. Huber et al. "Common Workflows for Computing Material Properties Using Different Quantum Engines." In: *npj Comput Mater* 7.1 (1 Aug. 19, 2021), pp. 1–12. DOI: 10.1038/s41524-021-00594-6 (cited on page 79).

[152] Christopher A. White and Martin Head-Gordon. "Derivation and Efficient Implementation of the Fast Multipole Method." In: *The Journal of Chemical Physics* 101.8 (Oct. 15, 1994), pp. 6593–6605. DOI: 10.1063/1.468354 (cited on page 79).

[153] Christopher A. White et al. "Linear Scaling Density Functional Calculations via the Continuous Fast Multipole Method." In: *Chemical Physics Letters* 253.3 (May 3, 1996), pp. 268–278. DOI: 10.1016/0009-2614(96)00175-3 (cited on page 79).

[154] Daniel S. Lambrecht and Christian Ochsenfeld. "Erratum: "Multipole-based Integral Estimates for the Rigorous Description of Distance Dependence in Two-Electron Integrals" [J. Chem. Phys. 123, 184101 (2005)]." In: *The Journal of Chemical Physics* 136.14 (Apr. 11, 2012), p. 149901. DOI: 10.1063/1.3693966 (cited on page 79).

[155] Andreas Irmler. "Periodic Fock Exchange with Atom-Centered Gaussian Basis Sets." PhD thesis. Universität Konstanz, 2018 (cited on page 79).

[156] Sharad Singhal et al. "OpenFAM: A Library for Programming Disaggregated Memory." In: *OpenSHMEM and Related Technologies. OpenSHMEM in the Era of Exascale and Smart Networks*. Ed. by Stephen Poole et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 21–38. DOI: 10.1007/978-3-031-04888-3_2 (cited on page 107).

[157] Jean Marie André. "Self-Consistent Field Theory for the Electronic Structure of Polymers." In: *J. Chem. Phys.* 50.4 (Feb. 15, 1969), pp. 1536–1542. DOI: 10.1063/1.1671239.

[158] M. Causa et al. "Treatment of the Exchange Interactions in Hartree-Fock LCAO Calculations of Periodic Systems." In: *J. Phys. Chem.* 92.4 (Feb. 1, 1988), pp. 909–913. DOI: 10.1021/j100315a010.

[159] Raffaella Demichelis and Roberto Dovesi. "Discussion about Some Parts of the CRYSTAL Code - Work in Progress." In: (2008), p. 147.

[160] K. Doll, V. R. Saunders, and N. M. Harrison. "Analytical Hartree–Fock Gradients for Periodic Systems." In: *International Journal of Quantum Chemistry* 82.1 (2001), pp. 1–13. DOI: 10.1002/1097-461X(2001)82:1<1::AID-QUA1017>3.0.CO;2-W.

[161] Roman Łazarski, Asbjörn M. Burow, and Marek Sierka. "Density Functional Theory for Molecular and Periodic Systems Using Density Fitting and Continuous Fast Multipole Methods." In: *J. Chem. Theory Comput.* 11.7 (July 14, 2015), pp. 3029–3041. DOI: 10.1021/acs.jctc.5b00252.

[162] Sergey V. Levchenko et al. "Hybrid Functionals for Large Periodic Systems in an All-Electron, Numeric Atom-Centered Basis Framework." In: *Computer Physics Communications* 192 (July 1, 2015), pp. 60–69. DOI: 10.1016/j.cpc.2015.02.021.

[163] Joachim Paier et al. "Accurate Hartree-Fock Energy of Extended Systems Using Large Gaussian Basis Sets." In: *Phys. Rev. B* 80.17 (Nov. 20, 2009), p. 174114. DOI: 10.1103/PhysRevB.80.174114.

[164] C. Pisani and R. Dovesi. "Exact-Exchange Hartree–Fock Calculations for Periodic Systems. I. Illustration of the Method." In: *International Journal of Quantum Chemistry* 17.3 (1980), pp. 501–516. DOI: 10.1002/qua.560170311.

[165] Cesare Pisani, ed. *Quantum-Mechanical Ab-initio Calculation of the Properties of Crystalline Materials*. Lecture Notes in Chemistry. Berlin, Heidelberg: Springer, 1996.

[166] Wahyu Setyawan and Stefano Curtarolo. "High-Throughput Electronic Band Structure Calculations: Challenges and Tools." In: *Computational Materials Science* 49.2 (Aug. 1, 2010), pp. 299–312. DOI: 10.1016/j.commatsci.2010.05.010.

[167] *SQLAlchemy*. SQLAlchemy, Apr. 19, 2022.

[168] Ravishankar Sundararaman and T. A. Arias. "Regularization of the Coulomb Singularity in Exact Exchange by Wigner-Seitz Truncated Interactions: Towards Chemical Accuracy in Nontrivial Systems." In: *Phys. Rev. B* 87.16 (Apr. 17, 2013), p. 165122. DOI: 10.1103/PhysRevB.87.165122.

[169] C. J. Tymczak et al. "Linear Scaling Computation of the Fock Matrix. VIII. Periodic Boundaries for Exact Exchange at the Γ Point." In: *J. Chem. Phys.* 122.12 (Mar. 22, 2005), p. 124105. DOI: 10.1063/1.1869470.

[170] Xiao Wang, Cannada A. Lewis, and Edward F. Valeev. "Efficient Evaluation of Exact Exchange for Periodic Systems via Concentric Atomic Density Fitting." June 6, 2020.