

CPC 50th anniversary article

The ABINIT project: Impact, environment and recent developments[☆]



Xavier Gonze^{a,b,*}, Bernard Amadon^c, Gabriel Antonius^d, Frédéric Arnardi^c, Lucas Baguet^{c,e,f}, Jean-Michel Beuken^{a,b}, Jordan Bieder^{g,c}, François Bottin^c, Johann Bouchet^c, Eric Bousquet^g, Nils Brouwer^c, Fabien Bruneval^h, Guillaume Brunin^a, Théo Cavignac^a, Jean-Baptiste Charraud^c, Wei Chen^a, Michel Côtéⁱ, Stefaan Cottenier^{j,k}, Jules Denier^c, Grégory Geneste^c, Philippe Ghosez^g, Matteo Giantomassi^{a,b}, Yannick Gillet^a, Olivier Gingrasⁱ, Donald R. Hamann^{l,m}, Geoffroy Hautier^a, Xu He^g, Nicole Helbig^{g,b}, Natalie Holzwarthⁿ, Yongchao Jia^a, François Jollet^c, William Lafargue-Dit-Hauret^g, Kurt Lejaeghere^k, Miguel A.L. Marques^o, Alexandre Martin^{c,g}, Cyril Martins^c, Henrique P.C. Miranda^a, Francesco Naccarato^a, Kristin Persson^{p,q}, Guido Petretto^a, Valentin Planes^c, Yann Pouillon^{r,b}, Sergei Prokhorenko^g, Fabio Ricci^g, Gian-Marco Rignanese^{a,b}, Aldo H. Romero^s, Michael Marcus Schmitt^g, Marc Torrent^c, Michiel J. van Setten^{a,t,b}, Benoit Van Troeye^a, Matthieu J. Verstraete^{g,b}, Gilles Zérah^c, Josef W. Zwanziger^u

^a Institute of Condensed Matter and Nanoscience, UCLouvain, B-1348 Louvain-la-Neuve, Belgium^b European Theoretical Spectroscopy Facility, Belgium¹^c CEA DAM-DIF, F-91297 Arpajon, France^d Département de Chimie, Biochimie et Physique, Institut de recherche sur l'hydrogène, Université du Québec à Trois-Rivières, Trois-Rivières, Canada^e IMPMC, UMR 7590 of Sorbonne Université/CNRS/MNHN/IRD, Paris, France^f CMLA, ENS Cachan, CNRS, Université Paris-Saclay, Cachan, France^g Q-MAT Center, CESAM Research Unit, University of Liège, Belgium^h DEN, Service de Recherches de Métallurgie Physique, CEA, Université Paris-Saclay, F-91191 Gif-sur Yvette, Franceⁱ Département de Physique, Université de Montréal, C.P. 6128, Succursale Centre-Ville, Montréal H3C 3J7, Canada^j Department of Electrical Energy, Metals, Mechanical Constructions and Systems, Ghent University, Technologiepark 46, B-9052 Zwijnaarde, Belgium^k Center for Molecular Modeling, Ghent University, Technologiepark 46, B-9052 Zwijnaarde, Belgium^l Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854-8019, USA^m Mat-Sim Research LLC, P.O. Box 742, Murray Hill, NJ, 07974, USAⁿ Department of Physics, Wake Forest University, Winston-Salem, NC 27109, USA^o Institut für Physik, Martin-Luther-Universität Halle-Wittenberg, Halle, Saale, 06120, Germany^p Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA^q Department of Materials Science and Engineering, University of California, Berkeley, CA 94720, USA^r Departamento de Ciencias de la Tierra y Física de la Materia Condensada, Universidad de Cantabria, Avenida de los Castros^{s/n}, 39005 Santander, Spain^t Physics Department, West Virginia University, Morgantown, USA^u Imec, Kapeldreef 75, B-3001 Leuven, Belgium^v Department of Chemistry and Clean Technologies Research Institute, Dalhousie University, Halifax, Canada

ARTICLE INFO

Article history:

Received 24 June 2019

Received in revised form 4 November 2019

Accepted 7 November 2019

Available online 18 November 2019

Keywords:

First-principles calculation

ABSTRACT

ABINIT is a material- and nanostructure-oriented package that implements density-functional theory (DFT) and many-body perturbation theory (MBPT) to find, from first principles, numerous properties including total energy, electronic structure, vibrational and thermodynamic properties, different dielectric and non-linear optical properties, and related spectra. In the special issue to celebrate the 40th anniversary of CPC, published in 2009, a detailed account of ABINIT was included [Gonze et al. (2009)], and has been amply cited. The present article comes as a follow-up to this 2009 publication. It includes an analysis of the impact that ABINIT has had, through for example the bibliometric indicators of the

[☆] The review of this paper was arranged by Prof. N.S. Scott.

* Corresponding author at: Institute of Condensed Matter and Nanoscience, UCLouvain, B-1348 Louvain-la-Neuve, Belgium.

E-mail address: xavier.gonze@uclouvain.be (X. Gonze).¹ www.etsf.eu

Electronic structure
Density-functional theory
Many-body perturbation theory
ABINIT

2009 publication. Links with several other computational materials science projects are described. This article also covers the new capabilities of ABINIT that have been implemented during the last three years, complementing a recent update of the 2009 article published in 2016. Physical and technical developments inside the `abinit` application are covered, as well as developments provided with the ABINIT package, such as the MULTIBINIT and A-TDEP projects, and related ABINIT organization developments such as ABIPY. The new developments are described with relevant references, input variables, tests, and tutorials.

Program summary

Program Title: ABINIT

Program Files doi: <http://dx.doi.org/10.17632/csvdrr4d68.1>

Licensing provisions: GPLv3

Programming language: Fortran2003, Python

Journal reference of previous version: X. Gonze et al, Comput. Phys. Commun. 205 (2016) 106–131

Does the new version supersede the previous version?: Yes. The present 8.10.3 version is now the up-to-date stable version of `abinit`, and supercedes the 7.10.5 version.

Reasons for the new version: New developments

Summary of revisions:

- Many new capabilities of the main `abinit` application, related to density-functional theory, density-functional perturbation theory, GW, the Bethe-Salpeter equation, dynamical mean-field theory, etc.
- New applications in the package: `multibinit` (second-principles calculations) and `tdep` (temperature-dependent properties)

Nature of problem: Computing accurately material and nanostructure properties: electronic structure, bond lengths, bond angles, primitive cell, cohesive energy, dielectric properties, vibrational properties, elastic properties, optical properties, magnetic properties, non-linear couplings, electronic and vibrational lifetimes, etc. For large-scale systems, second-principles calculations, building upon the first-principles results, are also possible.

Solution method: Software application based on density-functional theory and many-body perturbation theory, pseudopotentials, with plane waves or wavelets as basis functions. Different real-time algorithms are implemented for second-principles calculations.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

The search for materials and nanostructures with desirable properties and functionalities, constrained by cost, durability and environmental concerns, is interestingly evolving toward a combined experimental-theoretical endeavor [1]. Progress in computers and algorithms, combined with many years of development of software applications, has largely driven this evolution, with simulations becoming increasingly fast and accurate and allowing to target more and more properties. Although many software applications contribute to the enormous number of first-principles calculations of material and nanostructure properties [2], the ABINIT project stands out as having pioneered the open-source approach, about twenty years ago, and still relies on the same, community-driven, development model. The description of the ABINIT project in the Computer Physics Communication 40th anniversary issue [3] is one of the most cited CPC articles. In this context, for the CPC 50th anniversary, the present article has the goal to provide an overview of recent developments and ongoing efforts within the ABINIT project, but also, to provide a brief assessment of the impact of this previous anniversary publication [3], and to highlight the links of the ABINIT project with several other computational material science projects.

The main ABINIT application implements density-functional theory (DFT) [4–6], density-functional perturbation theory (DFPT) [7–12], many-body perturbation theory [13,14], including the GW approximation and the Bethe–Salpeter equation (BSE), and other first-principles formalisms. It calculates the total energy, electronic structure, and numerous electronic, optical, vibrational, magnetic, mechanical, transport and thermodynamical properties of materials and nanostructures. The description of recent developments within the ABINIT project in the present article will cover the period between 2016, version 7.10.5 of the ABINIT package,

and 2019, version 8.10.3 of the ABINIT package, respectively the last v7 and v8 ABINIT package releases. ABINIT v8.10.3 is uploaded in the CPC software library with the present publication. Ongoing developments, not available in ABINIT v8.10.3, but already integrated and tested in the ABINIT trunk at the time of writing, will also be briefly described and should be made available soon with ABINIT v9.x.y, possibly at the end of 2019 or beginning of 2020. The description of other capabilities of ABINIT (either prior to v7.10.5 or, on the contrary, to be integrated in the trunk in the future) will not be covered here. Similarly, the numerous software-engineering techniques and its multifaceted documentation system, on which ABINIT has built its success and its sustainability, will only be marginally covered. We refer the reader to other previous incremental descriptions of the ABINIT project, in Refs. [3,15–17], to complement the present work.

The structure of the article is as follows. After the present introductory section, we will first describe in Section 2 the impact of Ref. [3], summarize the results of a user survey performed in 2016, and then present the four projects mentioned explicitly on the right-hand side of Fig. 1, for which the link with ABINIT has been particularly strong in the 2016–2019 period.

In Section 3, we will focus on the inner ‘ABINIT’ box in Fig. 1: the new and ongoing developments in the main `abinit` application. They are numerous. We present them using different acronyms and keywords inside this ‘ABINIT’ box. We will focus first on the generic improvements in the DFT part of `abinit` (Section 3.1), including work on algorithms for geometry optimization or dynamics, LDA-1/2 band gap evaluation, interpolation of electronic structure, PAW orbital magnetization and magnetic shielding, and hybrid functionals. Then we will focus on DFPT (Section 3.2) with the treatment of non-collinear systems and Zeeman magnetic field perturbation, the inclusion of van der Waals interactions, and Raman intensities computed with the

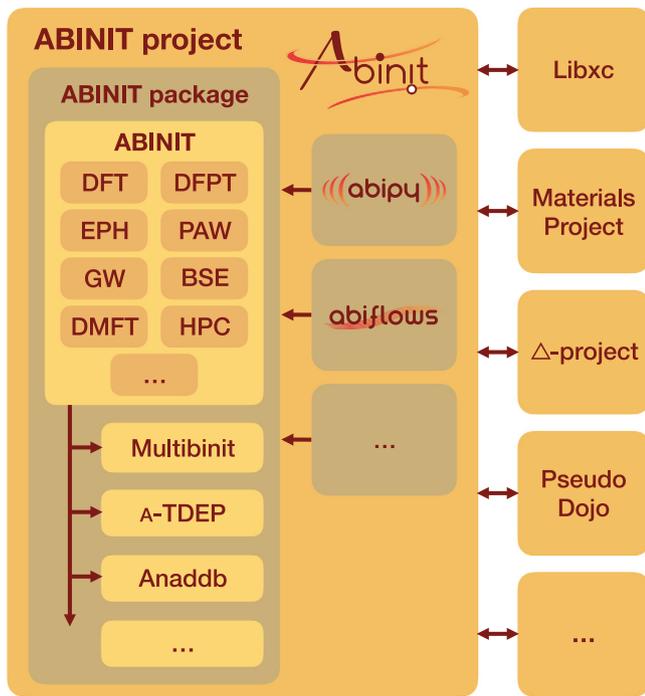


Fig. 1. The scope of the present publication. In the biggest box, the major components of the ABINIT project are represented with respect to each other. Related projects are indicated on the right-hand side. The ABINIT project includes components delivered in the ABINIT package (dark box on the left), as well as components delivered on the ABINIT organization github (for example ABIPy and ABIFlows). The ABINIT package contains a series of applications, including abinit, multibinit, tdep, as well as numerous scripts, documentation, a build system, and so forth. The abinit application has been improved in different aspects, represented by relevant acronyms and keywords in the inner light ABINIT box.

projector-augmented wave method (PAW). The new dedicated electron-phonon driver is presented in Section 3.3. Specific PAW-related developments are grouped in Section 3.4. We will also detail the improvements and new capabilities of the GW driver (Section 3.5), the BSE driver (Section 3.6), progress related to dynamical mean-field theory (DMFT, Section 3.7), and some improvements concerning the high-performance computing with abinit (HPC, Section 3.8).

In Section 4, we present new applications delivered inside the ABINIT package, from the MULTIBINIT project, for second-principles calculations of materials properties, and from the A-TDEP project, the ABINIT implementation of the TDEP algorithm, for temperature-dependent materials properties beyond the quasi-harmonic approximation. They are noted in Fig. 1 the inner 'ABINIT' box.

We complete the description of the new developments delivered in the ABINIT package in Section 5, with those related to the post-processor anadddb and the ElectronPhononCoupling script as well as with improvements in the documentation, build and testing systems. This completes the topics in the 'ABINIT package' box of Fig. 1.

In Section 6, we focus on the additional components of the ABINIT project that are not delivered directly in the ABINIT package, thus shown outside the left box of Fig. 1, but made available through the ABINIT Github site (<https://github.com/abinit>). In this respect, ABIPy provides a scripting environment for preparing abinit runs as well as post-processing tools for the graphical analysis of their results, while the ABIFlows package offers an infrastructure for workflows aimed at high-throughput applications. Section 7 presents perspectives for the project.

The text often refers to input variables and input files inside the ABINIT package. The input variable descriptions can be found in the on-line documentation of ABINIT, <https://docs.abinit.org/variables>, but can also be found, in the package, in the abimkdocs directory, for example in abimkdocs/variables.abinit.py for the main abinit application. The input files can be found in the packages inside the tests directory, with generic path tests/name_of_group/Input, where name_of_group is one of 27 different tests groups, such as v1-v8, para1, libxc, wan-nier90, and so forth. Each group of tests has at most 100 input files, each labeled with an index from 00 to 99. Reference output files can be found in the directory with generic path tests/name_of_group/Refs. We adopt the notation name_of_group#index_of_test in the present publication to refer to a test. Moreover, in the on-line or electronic.pdf version of this publication, a hyperlink is provided to the on-line version of such files.

2. ABINIT: impact and related projects

2.1. The 2009 CPC publication

ABINIT is one of the most used electronic structure codes by the materials science community. While distinct publications have reported the different developments within this package, this section focuses on the impact produced by our 2009 ABINIT publication [3]. For the analysis published here, we will use the citations gathered by the Web of Science [18] (WoS) up to June 2019. Though Google Scholar reports a higher number of citations, we used the results from the WoS, as the bibliometric analysis can be performed using established and well-developed software or even in WoS itself.

The 2009 ABINIT publication [3] has been cited 1530 times as of June 19, 2019, by research papers that have themselves been cited 24500 times (without self-citations) to the same date. These data demonstrate that the ABINIT package is recognized by leaders in the field, as the impact of citing publications is very noticeable. The paper's H-index is 67, which indicates that a large number of high-impact papers cite ABINIT as one of the main packages used in electronic structure characterization. While the majority of citations comes from the United States (450) followed by France (313), Germany (170) and China (161), citations are diverse, and researchers from 52 different countries in all continents have cited this particular ABINIT publication. Since 2010, the citation count per year has rapidly increased to 230 in 2016. Since then the citation count is decreasing, with 175 in 2018. This citation change leads us to predict that this paper will have a significant citation half-life as, after the maximum, the citations decrease very slowly compared to the initial rapid growth. We note that the decrease observed in recent years is likely due to the publication of a more recent article on ABINIT [17].

To provide a measure of the impact of ABINIT in different computational science communities, the number of fields that cite ABINIT can be used. In particular, the 2009 ABINIT publication has been cited by 34 distinct research areas (according to the WoS classification), where "Physics" is the research area with the highest citation count, with 1007 citations. Along the same lines, but now using the Web of Science Categories, we find that 63 different categories have cited this paper, with "Materials Science Multidisciplinary" being the highest citing category with 672 citations.

To understand the diversity of groups invested in ABINIT, we have counted the different affiliations reported in the papers citing our 2009 ABINIT publication [3]. We found 1220 different institutions, with "CEA France" and "Université catholique de Louvain" the institutions with the largest count: 84 and 75

publications respectively. Similarly, the supporting agencies are also included in the acknowledgments of most publications and we can study the impact of this computer package in relation to the funding support received by different research groups. In that respect, WoS data show that the ABINIT publication has been cited in publications that acknowledge 1931 different financial institutions, with the US National Science Foundation the largest one with 189 citations. It is important to note a shortcoming of this analysis, as the WoS tools are unable to merge institutions that are identified with different names, such as, for example, NSF and National Science Foundation, or the DOE and Office of Science of the US Department of Energy. Therefore the numbers per supporting agency should be larger, but here we only report the results obtained directly from the WoS analysis.

To characterize the community of users of the ABINIT package, we have performed an analysis of the coauthor network, which is the list of authors from works citing the ABINIT publication. This network corresponds to a graph obtained from the author field "AU" from the WoS data and relates authors within the same publication. A full record of all papers citing our 2009 ABINIT publication [3] was obtained from WoS, analyzed with the package Tethne [19], and visualized with Cytospace [20]. The results, illustrated in Fig. 2, show a large number of clusters of connected components. The social interpretation of these series of networks is that there are a large number of ABINIT user research groups that work quite independently of each other, and that there are a few groups that collaborate quite intensively. In this network plot, a node is associated with each author and a link exists between two nodes if the two authors are coauthors on a publication. In this plot, we only include clusters that have at least eight linked nodes, which means that at least eight different coauthors have collaborated on publications. Each node is embedded in a rectangle with color proportional to the number of publications involved. Many of the nodes with darker colors correspond to ABINIT developers. Some of them are present in different clusters, which indicates that they work in collaboration with different research groups. At the same time, we can see that several of the more massive clusters involved several ABINIT developers, demonstrating the close collaboration between them. There is also a large number of small clusters with no developer involved in the publications; this shows that the code is used by groups with electronic structure training but no ABINIT developers. In some clusters, the color allocated for an ABINIT developer is faint, which also demonstrates that there are cases, where the developer started collaborating with a research group but over time the group became independent and developed the corresponding expertise.

2.2. User survey

Starting from the initial release of ABINIT version 8, in May 2016, all users downloading the new version from the website were asked to complete a user survey. The responses were collected during one month, yielding a total of 507 replies. The survey showed that most people found ABINIT via a colleague (34%) or from the internet (31%), while 24% found ABINIT via a scientific paper. The remainder indicated other channels.

The basic features of DFT (total energy, structure relaxation, DOS, and so forth) are by far the most commonly cited reason given for obtaining ABINIT. Vibrational properties, GW, optical properties, electron-phonon coupling and dielectric properties are also mentioned with a relatively high frequency.

The survey also provided valuable information on the use of the build system. The Intel and Gnu compilers are used in almost equal ratios, all other compilers together make up only 3%. For the core libraries (mpi, fft, and linear algebra), all available

options are used to a more or less similar amount. The (optional) plugin libraries (WANNI90, LIBXC, NETCDF, etc.) are also used systematically.

In terms of MPI parallelization, the users indicate that any amount from 2 to 100 cores are used on a regular basis (81% of the responses). Running abinit in serial makes up about 10% of the calculations. About 7% of the users indicate running ABINIT on between 100 and 500 cores. Calculations on more than 500 cores are mentioned in less than 2% of the responses.

2.3. The MATERIALS PROJECT

We now present several projects for which the link with ABINIT has been particularly strong in the 2016–2019 period. The first of these, the MATERIALS PROJECT, is an online database of material properties. In the past, such databases were collections of experimental data which were often incomplete, as measurements of properties were not available for all materials. As calculations of material properties have become more accurate, nowadays we see databases that harvest this new source of information and aim to provide complete calculated properties. The MATERIALS PROJECT has recently incorporated data from ABINIT to increase its material properties coverage.

The goal of the MATERIALS PROJECT (MP, online at <http://www.materialsproject.org>) [21] is to accelerate materials discovery and education through advanced scientific computing and innovative design methods, provide computational data for all known inorganic compounds, and finally, to disseminate that information and design tools to the materials community. Founded in 2011 at Lawrence Berkeley National Laboratory, the MATERIALS PROJECT is known worldwide as a leading computational effort within the Materials Genome Initiative. Through this effort, materials researchers and designers now have access to a comprehensive database of predicted properties of materials.

This constantly growing resource contains computations on over 130,000 inorganic crystalline materials, including total energy, X-ray diffraction spectra, crystal structure, volume, density, etc. Additionally, MP disseminates a range of complex quantities that describe the fundamental behavior of materials including, for example, more than 13,000 full elastic tensors [22], over 500,000 k-edge XANES spectra [23], more than 3000 piezoelectric tensors [24], some 6000 dielectric tensors [25], over 58,000 bandstructures [26], over 1000 elemental surface energies [27], and more than 3000 phonon band structures [28]. The ability to generate such materials property data quickly and accurately relies on a software infrastructure that combines well-maintained first-principles DFT codes such as VASP, ABINIT, and QChem [2] with automated workflows, analysis, and database management.

To advance such infrastructure, the MATERIALS PROJECT pioneered open-source software libraries for setting up, executing, analyzing, and deriving insights from calculations. These libraries include PYMATGEN [29], CUSTODIAN, FIREWORKS [30], and ATOMATE [31], which in themselves attract a cohort of active developers and users worldwide. In order to calculate automatically a specific property of any inorganic material, the MP infrastructure provides the necessary I/O, runtime management, and choice of specific numeric and physical parameters to provide a balance of accuracy, robustness and computing cost. In addition to data and software infrastructure, the MATERIALS PROJECT web site provides access to applications that combine and visually present the data for specific analyses such as phase diagram generation, aqueous stability, reaction energies, XAS spectral matching or battery electrode evaluation. Engagement with the community is evidenced by the MP's rapidly growing user base which, at the time of writing, includes over 85,000 registered users, consisting of a diverse set of researchers and students from academia and

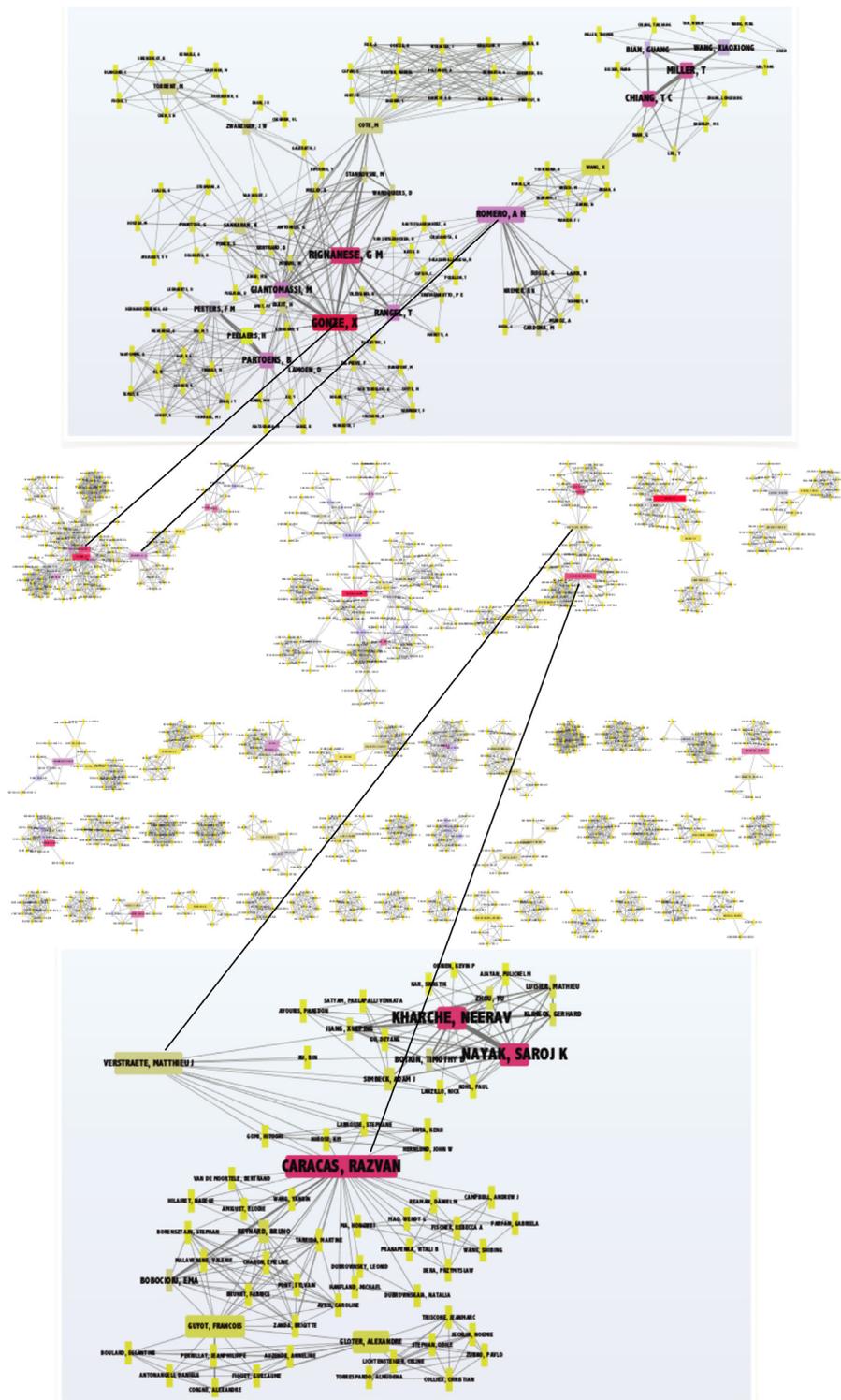


Fig. 2. Middle zone: coauthors networks based on the publications citing the 2009 ABINIT publication [3] with nodes connected to at least eight other nodes. Each author is framed within a colored rectangle, the darker the color the larger the number of associated coauthors with joint publications. Three primary clusters are identified. There are also 38 isolated clusters. Upper zone: magnified view of one primary cluster, in which one finds many of the core developers. Lower zone: magnified view of a primary cluster with few core developers present.

industry. Every day, more than 1500 distinct users log on to the site and on average, 600,000 data items are downloaded through its API. The diversity of the audience base showcases the usefulness of a first-principles materials database across the spectrum of education, research, and development activities.

As the state of the art in theory and computing advances, the specifics of the MATERIALS PROJECT data, scope, capabilities,

and infrastructure will no doubt change as well. New algorithms, software, functionals and implementations become available, and the MATERIALS PROJECT will adapt and strive to expose state-of-the-art, well-benchmarked materials data and associated algorithms. In this vein, the MATERIALS PROJECT recently partnered with ABINIT to include vibrational properties of materials [28]. In

2018, the joint effort launched the full first-principles phonon dispersion and vibrational density of states for 1521 semiconductor compounds in the harmonic approximation based on density-functional perturbation theory [28]. The data are collected along with derived dielectric and thermodynamic properties, providing a rich opportunity for automated learning of phase stability and dynamical stability.

2.4. The Δ -project

The `abinit` application is one of the 40–50 DFT codes currently available capable of dealing with periodic crystals [32]. Its core task is to solve the Kohn–Sham equations, which are at the heart of DFT. It is clearly important to assess whether `abinit`, and indeed all of the available codes, is a reliable implementation of DFT. Such an assessment is not straightforward, as there are no analytical solutions available to compare with. The best one can do in this respect is to compare the output of a standard set of tasks for several independently written DFT codes. As long as they all produce essentially the same results, one can have confidence that there is no intrinsic deviation or crucial bug in any of them. Such an approach is the goal of the Δ -project [33,34], a community effort by code developers and expert users to run the same set of tasks through a list of DFT methods and codes. This exercise tests the *reproducibility* of DFT implementations, that is, whether for a given exchange–correlation functional all implementations find a sufficiently similar numerical solution to the Kohn–Sham equations. It does not test the *accuracy* of DFT (or rather of the exchange–correlation functional used), namely how close the predictions are to experimental values [35,36].

The Δ -project concentrates on the most basic quantity that every DFT code calculates: the total energy. The total energy and its derivatives determine the phase and properties of a crystal, and are therefore of fundamental importance. The set of tasks assigned to all codes focus on the energy–volume behavior. The total energy is calculated for 71 elemental crystals, for 7 volumes around the equilibrium volume, to construct the energy–volume curve $E(V)$. For every element, the ground state crystal structure (or a structure very similar to the ground state) is used. This choice guarantees some structural variety, while scanning a considerable part of the periodic table. Therefore, the same test can be used as well to examine the performance of libraries of atomic datasets for both projector-augmented wave (PAW) and pseudopotential.

The major conclusion from this exercise is that all mainstream DFT codes produce total energies that are indistinguishable from each other: $E(V)$ curves predicted by two independent DFT codes vary less from each other than $E(V)$ curves determined by two independent experiments. This finding is true for all-electron DFT codes, but also for PAW and pseudopotential codes. `abinit`, for example, has been tested using a wide array of PAW datasets (JTH [37], GPAW [37,38], and GBRV [39]), and norm-conserving pseudopotential libraries (PseudoDojo ONCVSP [40,41], HGHK [42,43], and Troullier–Martins FHI [44]). The results of these calculations agree very well with implementations in other codes [33,34]. The ability to test multiple PAW or pseudopotential libraries using the same DFT code, or to test the same PAW or pseudopotential library with two different DFT codes, is also useful in the development of better PAW potentials and pseudopotentials.

Further efforts are currently ongoing, and include expanding the test set, by examining the same element in multiple structurally and chemically different environments, and examining properties beyond total energies. Such tests are more stringent, but also more representative of production runs of DFT on real materials.

2.5. The LIBXC project

In any implementation of DFT, the specific exchange–correlation functional employed is a crucial ingredient. The development of functionals is an active line of research, and the availability of recent functionals in a particular code is desirable to serve its user community better. Although `abinit` implements some functionals internally, most of the available functionals come from the LIBXC project, a library [45,46] of functionals for DFT. LIBXC includes functionals of the first 3 rungs of Jacob’s ladder of density functionals [47], i.e. local-density approximations (LDAs), generalized gradient approximations (GGAs), and meta-GGAs. It also includes the semi-local components of many hybrid functionals that belong to the fourth rung.

There were many objectives when developing this library. The first was to make all relevant functionals available consistently to scientists in different fields, from solid-state to atomic and molecular physics, from quantum chemistry to theoretical biochemistry. This goal, of course, was only possible if LIBXC was included in a variety of density-functional theory codes, an objective largely attained, as LIBXC is by now used in almost 30 different projects from several communities. The second was to provide a detailed historical account of the development of density functionals. For this, the literature of the past 55 years was sieved and the expressions implemented. Most of the historical (and sometimes obscure) functionals that were found were documented. The final objective was to allow for an easy comparison of different density-functional programs and a straightforward way to benchmark functionals.

At present LIBXC includes a grand total of 486 functionals, of which 59 are LDAs, 306 GGAs, and 121 meta-GGAs. By looking at the year that each of these functionals was proposed one can easily trace the history of this field, from its beginning in the 1960s, to its exponential growth starting around 1990. The field seems to have reached its peak at around 2010, with some years witnessing the publication of 20 articles proposing new functionals. The research seems to be still rather active nowadays, with 5–10 new functionals put forward each year.

Technically, LIBXC provides the energy functional, together with its derivatives with respect to all arguments (densities, gradients etc.). A standard Kohn–Sham equation only requires the first derivatives, which are enough to build the exchange–correlation potential v_{xc} . However, the calculation of linear and non-linear response properties within DFPT requires higher-order derivatives. This is relatively simple for the LDAs, as they are only written in terms of the spin-densities n_{\uparrow} and n_{\downarrow} . However, the number of derivatives increases dramatically for the GGAs and the meta-GGAs. Also, the complexity of the mathematical expressions typically increases going up the rungs of Jacob’s ladder.

This problem is tackled in LIBXC by using a mathematical symbolic language, specifically MAPLE. The workflow is then: (i) functionals are written in MAPLE language (that is very similar to traditional mathematical notation); (ii) MAPLE is then used to perform the symbolic derivation of all required derivatives and to translate the result to the C language; (iii) the resulting code is post-processed into a form suitable for inclusion in LIBXC. In this way, one is able to provide up to the third derivatives for the majority of the functionals. Incidentally, the use of MAPLE also allows one to easily produce specialized routines for spin-unpolarized, ferromagnetic, and spin-polarized densities. This is particularly important not only for efficiency reasons, but also to circumvent numerical problems in many functionals.

LIBXC is, and will certainly continue to be, a work in progress. Not only are several new functionals added each year, but also new functionalities make their way into the code. In the near

future, it is planned, for example, to have the possibility of using non-collinear densities, or the creation of a large test set reproducing key calculations performed in the literature.

LIBXC has been especially useful for `abinit` in providing the exchange–correlation kernels for DFPT, the results of which (phonon frequencies, thermodynamic quantities) have been compared in Ref. [48]. At present LIBXC versions 2.2.3 and 3.0.0 are recommended for use with `abinit` v8.10.3. LIBXC version 4.3 is interfaced with the forthcoming `abinit` v9. This new version addresses directly the C API of LIBXC, so that additional Fortran mod files will no longer be needed. All the tests related to LIBXC are provided in the `tests/libxc` directory, in which hundreds of functionals are indeed used, at the level of total energy calculations (also for hybrid functionals), as well as DFPT calculations.

2.6. The PSEUDOJOJO project

The `abinit` application employs a planewave basis set to describe the electronic states. This approach has many advantages but also some inconveniences, the most severe being the inevitable use of pseudopotentials. These reduce the required number of basis functions by replacing the bare nuclear potential by a softer pseudized one. In the past, the availability of “standardized” accurate pseudopotentials was problematic. The PSEUDOJOJO project was created to address this situation. This project had its roots in ABINIT but now serves the broader electronic structure community.

Norm-conserving pseudopotentials and PAW datasets are collected and maintained within the PSEUDOJOJO project [41], which provides several tools to generate pseudopotentials and validate them using benchmark sets based on all-electron calculations [34,35,39]. The pseudopotential data files and the corresponding test results are available via the web-interface at pseudodojo.org. The norm-conserving pseudopotentials generated with the ONCVSP code by Hamann [40] are available in `psp8`, `upf`, and `psml` [49] formats. These formats are compatible with several codes, including ABINIT, SIESTA, QUANTUM ESPRESSO, CP2K, QBOX, and QUANTUMATK. The PAW datasets [37] are available in the `pawxml` format shared by ABINIT and GPAW.

The PSEUDOJOJO provides LDA, PBE and PBEsol norm-conserving pseudopotentials both in a scalar relativistic and a fully relativistic version including spin–orbit coupling. These tables all come in two versions, standard and stringent. The standard version is sufficient for ground-state and DFPT calculations. For complicated systems, mostly magnetic materials or calculations requiring an accurate description of the scattering properties in the unoccupied state energy range, which are important for GW calculations, the stringent table provides alternatives with full shells in the valence. The pseudopotentials in the stringent table tend to have smaller core radii and/or include more semi-core electrons. For the lanthanide series, a special table of norm-conserving pseudopotentials, with *f*-electrons frozen in the core, is also available. These pseudopotentials are intended *only* to be used to study systems in which the lanthanide is in the 3+ oxidation state.

The pseudopotentials at the PSEUDOJOJO have been subjected to a series of tests including the Δ -project [34] and the GBRV benchmark set [39], phonon mode calculations at the Γ -point, and ghost-state detection. A full description of the tests is given in Ref. [41]. Based on the convergence behavior of the results of these tests, hints for the plane-wave energy cutoff are provided.

2.7. Other projects with a significant link to ABINIT

The ecosystem in which ABINIT grows is very rich, and does not simply include the four above-mentioned projects, for which the links were particularly strong in the 2016–2019 period. Without being exhaustive, several other projects are described below, some of which are also mentioned in other sections of the present article.

In addition to the files made available by the PSEUDOJOJO project, generators of pseudopotentials and PAW datasets are highly useful for ABINIT users. Two particularly notable projects are ONCVSP [40], for norm-conserving pseudopotentials, and ATOMPAW [50] for PAW datasets.

The software applications inside the ABINIT package are often linked to several libraries besides LIBXC, including WANNIERTO [51], ELPA [52], TRIQS [53], the ESL bundle [54], and others. Interfacing of `multibinit` with SCALE-UP [55] (which implements coupled tight binding and interatomic potential dynamics) is functional and ongoing. Similarly, the LibPAW library, present inside the ABINIT package, is used by BigDFT [56] as well as a recently developed DFT code based on Gaussian orbitals [57].

Finally, other projects can employ wavefunctions and other data generated by `abinit`, including FEFF [58] (that uses `abinit` wavefunctions, dynamical matrices and electron–phonon couplings through the OCEAN [59] and DMDW [60] applications), DP [61], EXC [62,63], BERKELEYGW [64] and YAMBO [65], implementing many-body perturbation theory, as well as the databases NOMAD, materialscloud.org and the above-mentioned MATERIALS PROJECT.

3. New capabilities of the main `abinit` application

As mentioned in the introduction, we will describe the new capabilities by grouping them around the following keywords: properties obtained by standard DFT or generalized DFT (Section 3.1), including work on algorithms for geometry optimization or dynamics, LDA-1/2 band gap evaluation, interpolation of electronic structure, PAW orbital magnetization and magnetic shielding, and hybrid functionals; density-functional perturbation theory (Section 3.2) with the treatment of non-collinear systems and Zeeman magnetic field perturbation, the inclusion of van der Waals interactions, and Raman intensities computed within PAW; the electron–phonon driver (Section 3.3); new PAW (Section 3.4) and GW (Section 3.5) capabilities; the Bethe–Salpeter equation (Section 3.6); dynamical mean-field theory (Section 3.7); and high-performance computing capabilities (Section 3.8).

3.1. DFT and generalized DFT

3.1.1. Algorithms for dynamics, geometry optimization, ensemble of images, and transition states

The capability to compute the forces and stresses of a given ion configuration in a periodic cell opens different possibilities: the equilibrium geometry can be obtained (possibly under constraints), the Born–Oppenheimer hypersurface can be constructed, and the ion dynamics can be examined or sampled. Moreover, replicating a set of ions in different configurations, and exchanging information between configurations also opens a large class of algorithms based on “images” or “replicas”, allowing for example transition state searches and path-integral molecular dynamics.

Many geometry optimization and ion dynamics algorithms were already available in `abinit` v7.10.3. These were selected by the `ionmov` input variable, complemented by the `optcell` input variable for defining specifically the cell optimization and dynamics, the `iatfix` input variable for optionally fixing the position of

given atoms, and the **strtarget** input variable for imposing an external stress. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm to optimize the geometry was implemented with **ionmov** = 2 in `abinit` v7.10.3. A new variant of this algorithm, the so-called Limited-memory BFGS [66] has been made available, originating from the `NETLIB` implementation (<https://www.netlib.org/>), restructured and rewritten in modern Fortran. It is activated by using **ionmov** = 22, see v8#02. It is at least as efficient as the **ionmov** = 2 implementation, and even outperforms it in the many cases where the latter was either slowing down or stagnating (typically after 10 or 20 computations of forces). It is now the recommended value for **ionmov** for performing geometry optimization, as of `abinit` v8.10.3. The FIRE algorithm [67], also for geometry optimization, has been implemented, and is activated with **ionmov** = 15, see v8#17. The original reference describes it as more stable than the BFGS, at the expense of a reasonable increase in the number of evaluation of forces.

The ion dynamics can now be computed using the velocity Verlet algorithm [68], activated with **ionmov** = 24, see v8#12, as well as the isokinetic algorithm for fixed-temperature simulation [69] with **ionmov** = 12, see v8#21 or v8#22, which is an alternative to the thermostat approaches (for instance the Nosé thermostat, **ionmov** = 8). The Born–Oppenheimer hypersurface can be sampled by using a Hybrid Monte Carlo (HMC) approach [70], with **ionmov** = 25, see v8#34. Within the HMC algorithm, the trial states are generated via short *NVE* trajectories (ten **ionmov** = 24 steps in the current implementation). The initial momenta for each trial are randomly sampled from a Boltzmann distribution, and the final trajectory state is either accepted or rejected based on the Metropolis criterion. Such a strategy allows simultaneously updating all reduced coordinates, and achieves a higher acceptance ratio than classical Metropolis Monte Carlo and better sampling efficiency for shallow energy landscapes [71]. As a side note, all these algorithms can be used by the second-principles `multibinit` main application, see Section 4.1.

The class of “image” algorithms, governed by the additional **imgmov** input variable, already included well-known algorithms to find transition states, such as the NEB algorithm with **imgmov** = 5, or the simplified String Method with **imgmov** = 2, and Path-Integral Molecular Dynamics, **imgmov** = 9 or 13.

The Path-Integral Molecular Dynamics algorithms have been complemented by the “Blue Moon Ensemble” method [72,73], a constrained Path-Integral Molecular Dynamics, activated by the additional input variable **pimd_constraint** = 1, v8#05, which allows one to examine rare events. Practically, the equations of motion are modified by additional forces (including a Lagrange multiplier) that maintain the constrained quantity to its initial value. At present, the only possible constraint is a linear combination of the centroids of the atomic positions (input variables: **nconeq**, **natcon**, **watcon**, **iatcon**), but this will be complemented in the near future by other constraints such as the distance between two atoms, or the difference of two interatomic distances. Moreover, currently it is not possible to impose two constraints simultaneously, (**nconeq** = 1). The idea behind this method is to force the system into high-energy configurations along a reaction coordinate (the occurrence of which constitutes a “rare event”), and compute the free-energy profile along the reaction coordinate. At each time step, `abinit` prints the quantity whose time-average over the constrained trajectory is the derivative of the free energy with respect to the reaction coordinate. If the reaction coordinate is sampled by a large enough number of constrained Molecular-Dynamics trajectories, corresponding to sufficiently dense values of the reaction coordinate, then the free energy profile can be obtained by integration. This constrained Path-Integral Molecular Dynamics approach is implemented in

the canonical ensemble and coupled to a Langevin thermostat, (**imgmov** = 9).

Additionally, a “linear combination of images” algorithm [74] has been implemented. It is activated by **imgmov** = 6, see v8#20. In this method, the different images have the same ionic configuration and cell, but the electronic state is different for each image. The global total energy is a fixed linear combination of the total energies of each image, and the dynamics of the ions and cell is governed by this global total energy, from which forces and stresses are computed: they are simple linear combinations of the forces and stresses for each image. This approach allows one to perform ensemble density-functional theory calculations, multiplet structure calculations [75], or to determine optimized configuration paths thanks to a Lagrange multiplier approach, as in Ref. [74].

3.1.2. LDA-1/2 band gap evaluation

It is well known that the band gaps of semi-conductors and insulators are poorly reproduced by LDA or GGA DFT calculations. Many methods have been proposed in the literature to overcome this problem, most of them relying on the framework of many-body perturbation theory, which are, however, considerably more time consuming than the underlying DFT calculation. However, Slater proposed years ago the method of half occupation to calculate ionization and affinity energies for molecules and atoms. A simple scheme to extend this method to infinite systems has been proposed by Ferreira et al. [76] It is named the LDA-1/2 method, and consists in adding a local potential of negative half charge to the system. It has been shown to be very competitive in terms of accuracy for band gaps, for the same computational cost as a LDA/GGA calculation [77].

The LDA-1/2 method has been implemented in `abinit`. Following [76], the LDA-1/2 pseudopotential is obtained adding a local part, V_S , to the usual local pseudopotential. It is defined by:

$$V_S = V(-1/2, r) - V(0, r), \quad (1)$$

in which $V(0, r)$ is the atomic potential for the neutral atom and $V(-1/2, r)$ is the potential of an ion with half an electron removed. This potential is trimmed with a function $\Theta(r) = [1 - (r/r_{\text{CUT}})^8]^3$ where r_{CUT} is a cutoff radius, that is given in Ref. [76] for ten elements (Si, N, As, O, Ga, Ge, P, Zn, S, In).

The $V(0, r)$ and $V(-1/2, r)$ potentials are obtained from the PAW atomic data generator `atompaw` [50] for the neutral and ionic configurations. A new XML tag has been added in the JTH PAW atomic data files of the ten elements so that V_S will be available in the next version (v1.2) of the JTH table [37].

In `abinit`, V_S is treated as an external potential. To activate the LDA-1/2 calculation the new keyword **ldaminushalf** must be set to 1, as in v8#32.

3.1.3. Star function interpolation of electronic band structure

Despite all the advancements in *ab initio* techniques to describe crystalline solids, it is often still difficult to achieve the

high \mathbf{k} -space resolution required to converge certain integrals in the Brillouin zone (BZ). Besides, hybrid-functional calculations or beyond-DFT methods such as GW can only compute single-particle energies on a discrete set of \mathbf{k} -points belonging to a homogeneous mesh, thus rendering the computation of band structures along a high-symmetry path computationally expensive. To address these problems, significant efforts have been spent to develop accurate techniques for interpolating eigenvalues and/or matrix elements in \mathbf{k} -space. Unfortunately, reaching high accuracy requires complicated interpolation schemes involving wavefunctions and/or several input parameters that must be specified by the user. For this reason, `abinit`v8 provides the user with the Shankland–Koelling–Wood (SKW) interpolation method [78–82], which requires as input a set of eigenvalues in the IBZ and a single parameter defining the basis set for the interpolation.

In the SKW method, the single-particle energies are expressed in terms of the (symmetrized) Fourier sum

$$\varepsilon_{n\mathbf{k}} = \sum_{\mathbf{R}} c_{n\mathbf{R}} S_{\mathbf{R}}(\mathbf{k}). \quad (2)$$

where the star function, $S_{\mathbf{R}}(\mathbf{k})$, is defined by

$$S_{\mathbf{R}}(\mathbf{k}) = \frac{1}{N} \sum_{\mathcal{O}} e^{i\mathbf{k} \cdot \mathcal{O}\mathbf{R}}, \quad (3)$$

\mathbf{R} is a lattice vector and the sum is over the N rotations of the crystallographic point group. By construction, the expansion in Eq. (2) fulfills the basic symmetry properties of the single-particle energies:

$$\varepsilon_{n\mathbf{k}} = \varepsilon_{n\mathbf{k}+\mathbf{G}}, \quad (4)$$

$$\varepsilon_{n\mathbf{k}} = \varepsilon_{n\mathcal{O}\mathbf{k}}. \quad (5)$$

In principle, the expansion coefficients in Eq. (2) can be uniquely determined by using a number of star functions equal to the number of *ab initio* \mathbf{k} -points but this usually leads to sharp oscillations between the input eigenvalues. To avoid this problem, one uses more star functions than *ab initio* \mathbf{k} -points and constrains the fit so that the interpolant function passes through the input energies and a roughness function is minimized.

This `ainterp` variable activates the interpolation of the electronic eigenvalues. The user can specify the number of star functions per *ab initio* \mathbf{k} -point and an optional Fourier filtering as proposed in [83]. `ainterp` can be used to interpolate KS eigenvalues at the end of the ground state calculation (`v8#42`) or to interpolate GW energies (`libxc#42`) when `optdriver` = 4. For GW band structures, however, we found that interpolating the GW corrections instead of the quasi-particle energies helps improve the fit [84]. In this case, one can employ the Python interface provided by `ABIPY` to automate the procedure. An example can be found in this [jupyter notebook](#).

3.1.4. Orbital magnetization and nuclear magnetic shieldings: ongoing developments

The presence of an external homogeneous magnetic field induces orbital electronic currents and hence additional effects in the energy, even in the absence of electron spin. At first order in the magnetic field, the induced current is observed through its coupling to nuclear magnetic dipoles, that is, the nuclear magnetic shielding, measured in nuclear magnetic resonance spectra. At second order, the field is detected through the magnetic susceptibility. Such effects may be computed by exploiting the so-called Modern Theory of Magnetization, which takes advantage of magnetic translation symmetry [85]. Two of us

developed this approach to all orders in the magnetic field [86], leading to the perturbed energy

$$E^{(n)} = \int_{\text{BZ}} d\mathbf{k} \text{Tr} \left[\left(\rho_{\text{kCC}}^{(n)} + \rho_{\text{kVV}}^{(n)} \right) H_{\mathbf{k}}^{(0)} \right], \quad (6)$$

where CC and VV refer to the conduction and valence subspaces respectively (an insulator is assumed), ρ is the density matrix, and $H^{(0)}$ is the magnetic-field-free Hamiltonian. Thus all field dependence is transferred to the density operator. The perturbed density operators are developed based on a mixed field and \mathbf{k} -space expansion, necessary to express operator products with magnetic translation symmetry.

This approach yields directly all-electron expressions. To implement this approach in `abinit` we have generalized the operator product formula to products of three terms, so that the idempotency condition in a PAW context, namely $\rho = \rho S \rho$, where S is the overlap operator, may be used to develop a perturbative treatment of the density operator in the presence of magnetic translation symmetry. As in the original treatment [86], the magnetic perturbation is reflected in the density operator through products of \mathbf{k} -derivatives, for example $\varepsilon_{\alpha\beta\gamma} B_{\alpha}(\partial_{\mathbf{k}_{\beta}} \rho)(\partial_{\mathbf{k}_{\gamma}} \rho)$. We treat the derivatives with a finite-difference formula in ρ , in contrast to finite differences of the wavefunctions, which leads to much more tractable formulae that do not require re-phasing in order to compute overlaps between neighboring \mathbf{k} -points. Our implementation includes a full PAW treatment of the orbital magnetization and Berry curvature, and does not require special PAW datasets (for example, it is not necessary to use norm-conserving PAW sets).

We have also implemented nuclear magnetic dipoles, as vector potential terms in the electronic energy. Computing $E^{(1)}$ from Eq. (6) in the presence of nuclear dipoles returns the nuclear magnetic shielding, as in the work of Thonhauser et al. [87]. We have in parallel implemented the computation of the Berry curvature and its integral over the Brillouin zone, returning the vector Chern invariant. In insulators with time-reversal invariance the Chern number is zero, but in the presence of nuclear magnetic dipoles it is non-zero and this provides for a useful additional check on the system response.

3.1.5. Hybrid functionals: ongoing developments

Hybrid functional calculations (generalized Kohn–Sham DFT) can be performed in `abinit` with different values of the input variable `ixc` or `ixc_sigma`: PBE0 (41), HSE06 (-428), HSE03 (-427), B3LYP (-402), where the negative values correspond to functionals defined through the `LIBXC` library [45]. Also, simple Hartree–Fock (40) is available.

Two calculation procedures have been implemented: (1) using a Kohn–Sham orbital basis set, as for GW calculations; (2) using a plane wave basis set, as for usual self-consistent calculations. The first implementation follows a prior Kohn–Sham calculation, and delivers the eigenenergies (but not the total energy, forces and stresses). It can be activated using `optdriver` = 3 or 4, with `gwcaltyp` = 5, 15 or 25 and the above-mentioned different values for `ixc_sigma`. The second implementation can be activated with the usual `optdriver` = 1 and the above-mentioned values for `ixc`. Both work with norm-conserving pseudopotentials (including non-linear core corrections) as well as PAW, for insulators and metals, for spin-unpolarized and collinear spin-polarized cases, while the spin-non-collinear case is under development.

The plane wave basis set implementation is based on a double self-consistent loop. Inside the outer self-consistent loop, an “adaptively compressed exchange” (ACE) operator is first generated, following Ref. [88]. This is the most time-consuming step of the methodology, the occurrence of which, for this reason, has to be minimized. The Kohn–Sham equation and the density

self-consistent updates are performed in an inner loop, with the ACE operator being kept constant. This inner loop, with fixed ACE operator, is very similar to the usual SCF loop of DFT, for which the choice of the algorithm for mixing the density or potential is governed by the input variable **iscf**.

The outer loop methodology involves combining input and output wavefunctions from previous steps. It is done thanks to a biorthogonalization procedure, that allows abinit to combine linearly such wavefunctions, bypassing the phase-matching problem between different sets of wavefunctions. More precisely, the last set of wavefunctions $|\psi_n^m\rangle$ is chosen as reference, $|\phi_n^{\text{ref}}\rangle = |\psi_n^m\rangle$. The other sets $|\psi_n^{m'}\rangle$ (input or output of previous steps) are biorthogonalized to it by computing the overlap matrix with the reference set, then inverting this matrix and applying it to the wavefunctions $|\psi_n^{m'}\rangle$, to deliver $|\psi_{n\perp}^{m'}\rangle$ with the biorthogonal property:

$$\langle \psi_{n\perp}^{m'} | \phi_n^{\text{ref}} \rangle = \delta_{nm'}. \quad (7)$$

Such biorthogonalized sets of wavefunctions $|\psi_{n\perp}^{m'}\rangle$ can then be linearly combined using coefficients α_j determined from standard iterative algorithms (for example, the Pulay DIIS method [89,90]) to deliver new biorthogonalized wavefunctions,

$$|\psi_{n\perp}^{m+1}\rangle = \sum_j^m \alpha_j |\psi_{n\perp}^j\rangle, \quad (8)$$

then orthonormalized, and used as input for the next step in the outer loop.

The plane wave implementation includes the possibility to downsample the wavefunction wavevector grid [91], to speed up the computation of the ACE operator. Parallelism over wavevectors is available. Forces and stresses are computed.

On top of such plane wave basis set hybrid functional calculations, GW calculations can be performed, both non-self-consistently (for example `G_0W_0@HSE06`) or self-consistently, with the different types of self-consistency that are allowed by the GW computation implementation in abinit governed by **gwcalctyp**. Note that the facilities for GW calculations in ABINIT are already relatively mature, and described in previous publications [3,16,17].

Currently, the tests of the implementation are as follows: `v7#65-v7#72`; `libxc#51-libxc#53`; `libxc#67-libxc#74`; `paral#09`; `paral#93-paral#94`.

3.2. Progress in density-functional perturbation theory calculations

3.2.1. DFPT for non-collinear systems and Zeeman magnetic field perturbation

While the implementation of DFPT in abinit already included the collinear spin case as well as the spin-orbit coupling case [92], it has now been extended to the case of non-collinear magnetic systems [93].² The main problem to face for such an extension

² As a side note, in ABINIT, the determination of the group of symmetries of a system explicitly takes into account not only the types and positions of all atoms, but also their magnetic characteristics, as defined by the user using the **spinat** input variable and optionally the **symafm** or **spgroupma** input variables: in the collinear case, abinit possibly finds and handles spin-flipping symmetries, needed to deal with antiferromagnetic (Shubnikov) space groups, while in the non-collinear case, abinit selects the symmetry operations that are compatible with atomic types, positions and magnetizations (but no magnetic operations of symmetry are yet allowed in that case). Such characteristics of the system are then preserved during the whole simulation, i.e. the charge and magnetization are symmetrized using the determined set of symmetries, including the spin-flipping ones, and the wavevector sampling is reduced to the corresponding irreducible Brillouin zone. This amounts to an interesting direct halving of the CPU time in the case of antiferromagnetic space groups, since abinit only computes the spin-up wavefunctions and densities, and deduces from symmetry operations the spin-down wavefunctions and densities. Such treatments are also available in the DFPT case. By the way, abinit can handle all 1191 “black-white” Shubnikov antiferromagnetic space groups (type III and type IV, see the input variable **spgroupma**), in addition to the usual 230 Fedorov space groups.

is to treat the exchange-correlation term, for which most of the present functionals have been built through a collinear framework. With such collinear functionals, the off-diagonal derivatives (containing the magnetization densities along the x and y directions) cannot be calculated since these components are neglected. One solution would be to use non-collinear exchange-correlation functionals recently proposed for LDA and GGA functionals [94–96]. Here, we have implemented approaches that allow to nevertheless exploit the collinear exchange-correlation functional derivatives to solve a non-collinear problem. The main idea is to transform locally the non-collinear derivatives to a collinear regime by aligning the local magnetization with the local quantization axis for each real space point. This alignment permits the computation of the derivatives locally in the usual collinear framework and then recovery of the global non-collinear orientation by performing the inverse transformation.

We have implemented two main methods to do the back and forth transformations, selected by the **ixcrot** input variable. The first method consists in a Taylor expansion of the spin rotation matrix (**ixcrot** = 1): it is independent of the choice of exchange-correlation potential (although only LDA is available at the moment). The second method uses an analytical expression for the LDA exchange-correlation derivatives (**ixcrot** = 2). As an alternative, the first method has also been implemented using an analytical expression of the transformation matrix (**ixcrot** = 3). These methods give the same results but we observed that the second one could be slower to converge than the other two (for the SCF cycle), though it might be system dependent. The default value has thus been set to **ixcrot** = 1. About a dozen of tests of the implementation are present, grouped in `tests/v8`, from `v8#68` to `v8#80`.

As of abinit v.8.10.3, the implementation has been completed for norm-conserving pseudopotentials and LDA functionals only; the atomic and electric field perturbations are treated only at the Γ point. Further developments will extend this implementation to the PAW case and to the \mathbf{q} -dependent atomic perturbation as well as the strain perturbation. Eventually, GGA and hybrid functionals will be considered.

Within this formalism, we have also extended the DFPT implementation of abinit to the homogeneous magnetic field perturbation applied to the electronic spins, yielding the electron spin part of the Zeeman interaction. Note that this does not include the coupling between the homogeneous magnetic field and the electronic orbital motion, which is considered separately and currently only for the ground state (Section 3.1.4). The non-collinear spin DFPT framework naturally allows for the linear response calculation of the electron spin susceptibility χ^m [97,98]. The magnetic susceptibility components can be estimated either through the second derivatives of the total energy, $\chi_{\alpha\beta}^m = \frac{\partial^2 E}{\partial B_\alpha \partial B_\beta}$, or, equivalently, by integrating the first derivative of the magnetization density with respect to the magnetic field perturbation, $m_\alpha^{(1)} = \frac{\partial E}{\partial B_\alpha}$, to B_β perturbation. This implementation has been done for LDA functionals and within the norm-conserving formalism and for static \mathbf{q} -dependent first potential derivative $v_{\text{ext}}^{(1)} = (\mathbf{B}e^{i\mathbf{q}\cdot\mathbf{r}} + \text{c.c.}) \cdot \boldsymbol{\sigma}$ where \mathbf{B} is the magnetic field perturbation, $\boldsymbol{\sigma}$ the Pauli matrices and c.c. stands for complex conjugate.

The input variable to activate the DFPT electron spin magnetic field perturbation is **rfmagn** = 1, the direction of the field is given by **rfdir** and the \mathbf{q} vector by the **qpt** input variable. The implementation has been done for both collinear and noncollinear cases, the **rfdir** flag being meaningless for the collinear case. We note that care should be taken for the transverse response at $\mathbf{q} = 0$, which could give divergent spin magnetic susceptibility for the transverse direction of the magnetic field in ferromagnetic cases where no anisotropy term is present (in this case there is no energy cost to turn all the spins together, which is often the

case in the absence of spin-orbit coupling). The implementation is tested in `v8#66`, `v8#68`, `v8#69`, and `v8#70`.

As noted, the magnetic field perturbation has been implemented for LDA functionals and norm-conserving pseudopotentials. In the future it will be extended to the PAW formalism and to GGA and hybrid functionals.

3.2.2. DFPT including van der Waals interactions

The implementation of the DFT-D dispersion schemes [99–101], introduced briefly in a previous ABINIT publication [17], has been completed. These DFT-D methods, namely DFT-D2, DFT-D3 and DFT-D3(BJ), add a pairwise term to some common functionals such as GGA-PBE, in order to capture more accurately long-range electron–electron correlation. They can be activated using the `vdw_xc` variable, with precision controlled using the `vdw_tol` variable. A three-body term can also be introduced using `vdw_tol_3bt`, but its use is questionable for periodic systems [102]. The DFT-D pairwise terms not only can be added to ground-state and relaxation computations, but also to response functions calculated by DFPT, including phonons and strain. The corresponding theoretical derivations and implementation validations can be found in Ref. [102] for interatomic force constants and in Ref. [103] for elastic constants, respectively. Examples of usage and of application can be found in the tests `vdwxc#10`, `v7#97` to `v7#99`, in the previously-cited references, or in Ref. [104].

3.2.3. Raman intensities in PAW

Within the Placzek’s approximation, the Raman intensity of the Stokes process associated to a phonon mode m depends on experimental parameters and the Raman tensor α^m [105–107]:

$$\alpha_{ij}^m = \sum_{\kappa,\beta} \frac{d\chi_{ij}}{d\tau_{\kappa\beta}} u_m(\kappa\beta), \quad (9)$$

where i and j stand for spatial directions, χ is the electric susceptibility, $\tau_{\kappa\beta}$ is the coordinate of the atom κ in the direction β and u_m is the displacement of the atom for the mode m . As χ describes the linear response of the system with respect to the application of an external electric field, which can be obtained from DFPT at second-order, the Raman tensor is obtained from the third-order level of perturbation.

The computation of the Raman tensor was already implemented in `abinit` in the so-called “Berry phase” formalism [107], but was limited to norm-conserving pseudopotentials. Here we present a new implementation, valid for both norm-conserving pseudopotentials and the PAW formalism, in which both TO and LO modes can be treated. We follow the scheme presented by Miwa [108], where the electric field is treated analytically in a full DFPT formalism, leading to the resolution of a non-self-consistent second-order Sternheimer equation.

For the user the workflow is as follows. After having obtained the ground-state and first-order wavefunctions, one has to add two supplementary datasets to compute the second-order wavefunction derivatives, the first one with the keyword `rf2_dkdk` = 1 and the second one with `rf2_dkde` = 1. `getwfk` (resp. `getddk`, `get1den`, `getdelfd`) are used to read ground-state (resp. first-order) quantities in both datasets. The second dataset also needs `getdkdk`, as it uses the second-order derivative of wavefunctions computed by the first one. The directions to be computed are given by `rf2_pert1_dir` and `rf2_pert2_dir`. However, when computing third-order derivatives of the energy, symmetries can be used to reduce the number of directions to be actually computed. `prepanl` = 1 can be used in `rf2_dkdk` = 1 or `rf2_dkde` = 1 datasets to automatically compute the needed directions only. A final dataset is needed to compute third derivatives of the energy,

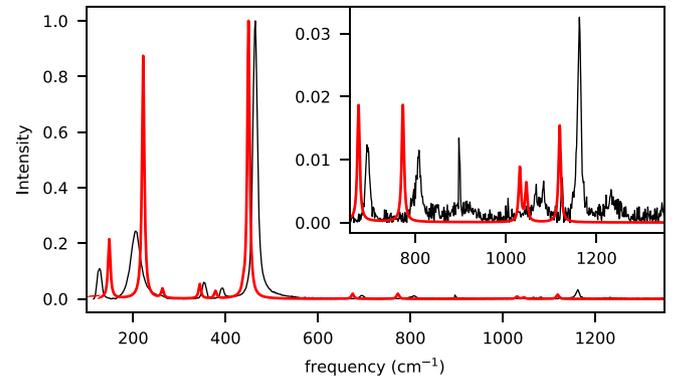


Fig. 3. Raman relative intensity line of polycrystalline quartz, α -SiO₂. The black line shows experimental data [109], while the red line is theoretical, using Lorentzian broadening. Phonon frequencies (peak positions) and Raman intensities are obtained from DFPT and PAW pseudopotentials. The peak width is arbitrary set to 3 cm⁻¹. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

activated with `optdriver` = 5. The new implementation is activated with `usepead` = 0, and the required second-order wavefunction derivatives are read using `getdkde`. The tests `v8#81` to `v8#89` are examples of such a workflow.

The Raman tensor is obtained with the `mrqddb` and `anaddb` ABINIT tools, as explained in the [DFPT/NLO tutorial](#). One obtains relative intensities with [105,106]:

$$I_m \propto (n_m + 1) \frac{(\omega_0 - \omega_m)^4}{\omega_m} |\mathbf{e}_s \cdot \alpha^m \cdot \mathbf{e}_l|^2, \quad (10)$$

where ω_0 and ω_m are the laser and the phonon frequencies respectively, \mathbf{e}_l and \mathbf{e}_s are the polarization direction of the incident and scattered light respectively, and $n_m = (e^{\hbar\omega_m/k_B T} - 1)^{-1}$ is the Bose occupation factor. The `anaddb` output can be analyzed to produced Raman intensities using Eq. (10) with the post-processing Python script `Raman_spec.py`. Powder spectra can also be computed, see [Fig. 3](#).

For this version of `abinit`, the computation of Raman intensities is now available for both norm-conserving and PAW pseudopotentials. In both cases, only LDA functionals can be used, and the system must be unpolarized or collinear spin-polarized (`nspinor` = 1 and `nspol` = 1 or = 2). The extension to the PAW+U formalism is under development.

3.3. Progress in electron–phonon calculations with ABINIT

3.3.1. Recent developments

Electron–phonon (EPH) calculations have been available in ABINIT for a long time with the help provided by the `anaddb` tool designed as a post-processing step of the EPH matrix elements computed at the end of the DFPT calculation. On the one hand, this approach was relatively easy to implement as most of the work, in particular the computation of the EPH matrix elements, was already performed by the DFPT code. On the other hand, the resulting implementation was too rigid as several important dimensions such as the number of \mathbf{k} -points, \mathbf{q} -points and bands in the EPH matrix elements had to be fixed at the level of the DFPT calculation. Performing convergence studies with respect to the \mathbf{k} -point sampling, for instance, required performing new (and more expensive) DFPT calculations with denser \mathbf{k} -meshes. Similarly, convergence studies for the \mathbf{q} -points required additional DFPT computations, possibly on meshes that were multiples of the initial sampling so to reuse the \mathbf{q} -points computed previously. To address these limitations, `abinit` v8 provides a new driver explicitly designed to compute the EPH matrix elements and

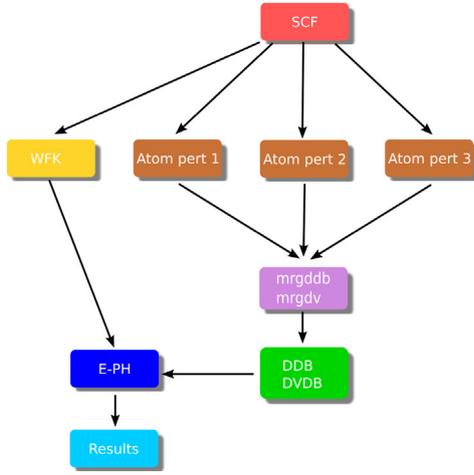


Fig. 4. Schematic representation of a typical EPH workflow. Each box type represents a different kind of calculation performed with `abinit`: self-consistent and non-self-consistent calculations to obtain wavefunctions, atomic perturbations with respect to specific atoms and directions. The DFPT potentials and the blocks of the dynamical matrix are merged and stored in two distinct files at the end of the DFPT part. Arrows indicate dependencies between the different steps of the calculation.

related physical properties. A different philosophy is used, in which EPH matrix elements are computed directly starting from the basic ingredients, namely, the ground-state wavefunctions stored in the WFK file, and the first-order change of the Kohn-Sham (KS) potential produced by the DFPT code. This approach allows for more flexibility because electron and phonon calculations are now partly decoupled: the \mathbf{k} -mesh can be densified by performing non-self-consistent calculations, thus bypassing the DFPT part, and interpolation schemes for the linear-response in \mathbf{q} -space can be readily implemented.

Unlike the previous algorithms implemented in `anaddb`, the new driver is directly interfaced with the `abinit` executable. This means that important `anaddb` variables related to the computation and diagonalization of the dynamical matrix such as `asr` and `dipdip` have been added to the `abinit` input file as well. A typical EPH calculation with the new driver is schematically depicted in Fig. 4. The brown boxes represent standard DFPT calculations done with relatively coarse \mathbf{k} - and \mathbf{q} -meshes. These calculations produce Derivative DataBase (DDB) files with dynamical matrix elements, and POT files with the local part of the first-order change of the KS potential (referred to as the DFPT potential below). A new utility, `mrgdv`, has been added to merge the DFPT potentials in a single “Derivative of $V(\mathbf{r})$ DataBase” DVDB file, while the partial DDB files can be merged, as in previous versions, with the `mrgddb` tool. The EPH driver (blue box) receives as input the two databases and a ground-state WFK file that may have been produced with a different \mathbf{k} -mesh (or even with a different number of bands) and uses these ingredients to compute the EPH matrix elements and associated physical properties. As the calculation of the DFPT potentials represents a significant fraction of the overall computational time, especially when compared with the non-self-consistent computation of the WFK file, the new EPH driver allows the user to densify the \mathbf{q} -mesh for phonons using the Fourier-based interpolation technique proposed by Eiguen and Ambrosch-Draxl in Ref. [110]. In this method, one computes the Fourier transform

$$W_{\kappa\alpha}(\mathbf{r}, \mathbf{R}) = \frac{1}{N_{\mathbf{q}}} \sum_{\mathbf{q}} e^{-i\mathbf{q}\cdot(\mathbf{R}-\mathbf{r})} \partial_{\kappa\alpha\mathbf{q}} v^{\text{scf}}(\mathbf{r}), \quad (11)$$

where the sum is over the \mathbf{q} -points belonging to the coarse grid used for the DFPT calculation and $\partial_{\kappa\alpha\mathbf{q}} v^{\text{scf}}$ represents the

(lattice-periodic) first order derivative of the local part of the KS potential associated to atom κ along the cartesian direction α . Once $W_{\kappa\alpha}(\mathbf{r}, \mathbf{R})$ is known, one can interpolate the potential at an arbitrary point $\tilde{\mathbf{q}}$ using the inverse Fourier transform

$$\partial v_{\tilde{\mathbf{q}}\alpha}^{\text{scf}}(\mathbf{r}) \approx \sum_{\mathbf{R}} e^{i\tilde{\mathbf{q}}\cdot(\mathbf{R}-\mathbf{r})} W_{\kappa\alpha}(\mathbf{r}, \mathbf{R}). \quad (12)$$

The accuracy of this interpolation technique depends on two factors: the density of the initial \mathbf{q} -mesh defining the Born-von Karman supercell, and the localization in \mathbf{R} -space of W . The long-range behavior of the DFPT potential in polar semiconductors and insulators is therefore problematic and a special numerical treatment is needed to enforce localization. As discussed in Refs. [111, 112], the long-range part associated to the displacement of atom κ along the cartesian direction α can be modeled with

$$V_{\kappa\alpha\mathbf{q}}^{\mathcal{L}}(\mathbf{r}) = i \frac{4\pi}{\Omega} \sum_{\mathbf{G} \neq -\mathbf{q}} \frac{(\mathbf{q} + \mathbf{G})_{\beta} \cdot \mathbf{Z}_{\kappa\beta\alpha}^* e^{i(\mathbf{q}+\mathbf{G})\cdot(\mathbf{r}-\tau_{\kappa})}}{(\mathbf{q} + \mathbf{G}) \cdot \boldsymbol{\epsilon}^{\infty} \cdot (\mathbf{q} + \mathbf{G})}, \quad (13)$$

where τ_{κ} is the atom position, Ω is the volume of the unit cell, \mathbf{Z}^* and $\boldsymbol{\epsilon}^{\infty}$ are the Born effective charge tensor and the dielectric tensor, respectively, and summation over the cartesian directions β is implied. Inspired by the approach used for the Fourier interpolation of the dynamical matrix in polar materials [10], we subtract the long-range part from the DFPT potentials before computing Eq. (11).

The EPH calculation is activated by `optdriver = 7` while `eph_task` defines the physical properties to be computed. In `abinitv8`, one can obtain phonon linewidths in metals `v7#88`, compute EPH matrix elements and save the results in `NETCDF` files that can be used by external codes, as illustrated in `v7#89`. Last but not least, one can estimate the correction to the zero-point renormalization of the band gap in polar materials using a generalized Fröhlich model based on *ab initio* effective masses [113]. An example input file is available in `v7#88`. Note that not all the new EPH features are available in `abinitv8`. Some of the ongoing developments, which will be made available in version 9, are briefly discussed in the following.

3.3.2. Ongoing developments

The electron-phonon self-energy [112] describes the renormalization of the single-particle energies due to the EPH interactions as well as the lifetime of the excitations. First-principles calculations of the EPH self-energy are therefore crucial to understand the temperature-dependence of band gaps, including the correction due to zero-point motion, as well as for computing phonon-limited mobilities within the Boltzmann transport equation. In `abinitv9`, it will be possible to compute the EPH self-energy in the Kohn-Sham representation using the EPH matrix elements. The code employs optimized algorithms to compute either the full self-energy (needed for QP corrections and spectral functions) or just the imaginary part that is then used to evaluate mobilities within the self-energy relaxation time approximation (SERTA) [112, 114]. The computation of the mobility will be fully integrated inside `abinit`, and will be an automatic output of the computation of the imaginary part of the self-energy, bypassing the need to post-process results. When computing the full self-energy, it is possible to reduce the number of empty states required for convergence by using the first-order wavefunctions obtained by solving the relevant Sternheimer equation. In the case of lifetime computations, the code takes advantage of the tetrahedron method to filter contributing \mathbf{q} -points, a double-grid integration technique to accelerate the convergence at marginal additional computational cost, and samples the relevant regions in the Brillouin zone contributing to transport properties thus leading to a significant reduction of the computational effort.

Crystalline symmetries are used throughout the code in order to reduce the number of \mathbf{k} - and \mathbf{q} -points that must be explicitly included in the integrals. To achieve good parallel efficiency, the most CPU demanding parts are parallelized with MPI employing a distribution schemes over \mathbf{q} -points, perturbations and bands (the band level is available only when computing the full self-energy).

3.4. PAW-related ongoing developments

PAW-related improvements and features are currently in various stages of development, including improvements for the optical and X-ray spectroscopy within the PAW framework, the possibility to relax core wavefunctions and the option to use meta-GGA functionals within the PAW method. These are described below.

Within the PAW framework, it has been possible for quite some time to calculate dipole matrix elements between (explicitly treated) valence/conduction electronic states (`prtnbla` = 1) and between such states and core states (`prtnbla` = 3). They can be used by the post-processing tool `conducti` to calculate the optical conductivity and X-ray absorption spectra, respectively [115]. Current development is focused on ensuring full compatibility of these spectroscopy features with spinor calculations (`nspinor` = 2). In addition, in spinor calculations the consideration of spin-orbit coupling effects in the dipole matrix element $M_{i,j,\mathbf{k}}$ using a semi-relativistic current operator [116] will also be possible:

$$M_{i,j,\mathbf{k}} = \langle \psi_{j,\mathbf{k}} | \mathbf{v} | \psi_{i,\mathbf{k}} \rangle \\ = \langle \psi_{j,\mathbf{k}} | -i\hbar\nabla + \frac{\hbar}{4m^2c^2} \boldsymbol{\sigma} \times \frac{\mathbf{r}}{r} \frac{dV(r)}{dr} | \psi_{i,\mathbf{k}} \rangle, \quad (14)$$

where \mathbf{v} is the velocity operator, $V(r)$ is the spherical potential and $\boldsymbol{\sigma}$ is the vector of Pauli matrices. The use of spinors in the PAW framework will also allow for the use of spinor core wavefunctions in the core-valence dipole matrix elements. The spinor core wavefunction, which is calculated Dirac-relativistically in a new version of the PAW dataset generator `atompaw` [50], has the following structure [6]:

$$\Psi_{njm_j}^l = \begin{pmatrix} g_{nj}(r) \varphi_{jm_j}^l \\ if_{nj}(r) \frac{\hat{\sigma} \cdot \mathbf{r}}{r} \varphi_{jm_j}^l \end{pmatrix}, \quad (15)$$

where $g_{nj}(r)$ and $f_{nj}(r)$ are radial wavefunctions and $\varphi_{jm_j}^l$ is a two dimensional analytic expression given by:

$$\varphi_{jm_j}^l = \sqrt{\frac{l + \frac{1}{2} \pm m_j}{2l + 1}} Y_l^{m_j - \frac{1}{2}} \chi_{\frac{1}{2}} \pm \sqrt{\frac{l + \frac{1}{2} \mp m_j}{2l + 1}} Y_l^{m_j + \frac{1}{2}} \chi_{-\frac{1}{2}}, \quad (16)$$

with $j = l \pm \frac{1}{2}$. The analytic expression is directly implemented in the calculation of the matrix elements, while the radial wavefunctions are read from `atompaw` output files. Note that only the particle part (upper components) of the Dirac 4-spinor will be used in `abinit`. The use of spinor wavefunctions will allow for the prediction of spin-orbit splittings and branching ratios within X-ray absorption spectroscopy. All the aforementioned matrix elements will also be available as `NETCDF` files for use in other codes (`iomode` = 3). The improvements to spectroscopy within the PAW framework are currently in the final testing and documentation stage and will be released soon.

The implementation of the relaxed core PAW (RCPAW) method is currently in an earlier stage of development. Originally proposed by Marsman and Kresse [117], this method allows for the relaxation of core wavefunctions in between SCF cycles, which is a compromise between a costly all-electron code and the frozen core approximation. Core relaxation is especially of interest for X-ray spectroscopy under warm dense matter conditions (high temperature and pressure).

Finally, in order to have a good estimation of electronic gaps at an economical computation cost, the possibility of using the exchange-correlation meta-generalized gradient approximation (meta-GGA) will be extended to the PAW approach following the work of Sun et al. [118]. It will be necessary to include the meta-GGA functional in the formalism implemented in `abinit` based on a Taylor series around the spherical contribution to the density (see Section 4.3.1 of [119]). A possible application of meta-GGA is to associate it with methods of energy landscape exploration. In this context, it will also be necessary to access the dynamic stability of materials (vibrational modes).

3.5. GW: bootstrap approximation and Monte-Carlo integration

While the core GW functionality in `ABINIT` is mature [3,16,17], there are some new developments.

GW [120] vertex corrections in the screened interaction ($W = \varepsilon^{-1}v$) can be taken into account in `abinit` through the bootstrap approximation of the exchange-correlation kernel [121,122]:

$$f_{xc,GG'}(\mathbf{q}, \omega) = \frac{v_{\mathbf{G}}^{1/2}(\mathbf{q}) \varepsilon_{GG'}^{-1}(\mathbf{q}, 0) v_{\mathbf{G}'}^{1/2}(\mathbf{q})}{1 - \varepsilon_{00}^{\text{RPA}}(\mathbf{q}, 0)}, \quad (17)$$

where ε^{RPA} is the dielectric matrix within the random-phase approximation (RPA), and v the bare Coulomb interaction. The target quantity, namely the inverse dielectric function ε^{-1} in Eq. (17), is determined self-consistently along with the f_{xc} kernel. For homogeneous systems, the bootstrap kernel can be simplified by keeping only the head of the matrix element so that self-consistent iterations are avoided [123]. A related one-shot kernel proposed in Refs. [124,125] is also available in `abinit` through the keyword `gwgamma`. Examples of such calculations can be found in the test `v67mbpt#36`.

Recently, an effort concerning the consistency between pseudopotential and plane-wave-based GW codes has been carried out [126]. In this study, the `abinit` GW code was compared to the `YAMBO` code [127] and the `BERKELEYGW` code [64]. It was realized that the main source of discrepancy among the codes is the method used to integrate the divergence in the exchange operator.

The expectation value of the exchange operator, Σ_x , in a plane-wave basis reads

$$\langle \mathbf{k} | \Sigma_x | \mathbf{k} \rangle = -\frac{1}{N_k \Omega} \sum_{\mathbf{q}, \mathbf{G}} v_{\mathbf{G}}(\mathbf{q}) |M_{\mathbf{k}ij}(\mathbf{q} + \mathbf{G})|^2, \quad (18)$$

where the matrix elements are defined as

$$M_{\mathbf{k}ij}(\mathbf{q} + \mathbf{G}) = \langle \mathbf{k} - \mathbf{q} | e^{-i(\mathbf{q} + \mathbf{G}) \cdot \mathbf{r}} | \mathbf{k} \rangle. \quad (19)$$

Whereas the matrix elements smoothly converge to δ_{ij} when $|\mathbf{q} + \mathbf{G}|$ goes to zero, the Coulomb interaction $v_{\mathbf{G}}(\mathbf{q}) = 4\pi/|\mathbf{q} + \mathbf{G}|^2$ requires a specific treatment since it is an “integrable divergence”. In `abinit`, the treatment of this delicate point is governed by the `icutcoul` input variable. `abinit` already contains numerous techniques for handling this term, such as the ones described in Refs. [128–130].

However, the comparison with other codes in Ref. [126] highlighted the performance of a simple Monte-Carlo integration of the Coulomb interaction $v_{\mathbf{q}}(\mathbf{G})$ in the “mini-Brillouin zone” around $\mathbf{q} + \mathbf{G} = \mathbf{0}$. This method transforms the \mathbf{q} -point summation in Eq. (18) back into an integral for the mini-Brillouin zone around the origin. Then, the integral is performed by brute force Monte-Carlo. In `abinit`, we further avoid the divergence at the origin by considering a small sphere around it, in which the Monte-Carlo points are discarded and replaced by an analytical integration. The number of Monte-Carlo sampling points is hard-coded to a very large value, which we found adequate for all the cases we have studied so far.

In the future, we plan to extend this technique into the hybrid functional calculations in the ground-state part of `abinit`.

3.6. Bethe–Salpeter equation: interpolation and temperature dependence

The computation of the dielectric function thanks to the Bethe–Salpeter equation (BSE) allows one to obtain optical absorption with excitonic effects. This capability of `abinit` v7, described in Ref. [17], Section 3.5.1, has been improved noticeably in `abinit` v8, following two directions.

First of all, a large speed up can often be obtained by the use of an interpolation technique, as described in Ref. [131]. This was presented briefly already in Ref. [17], Section 6.2. The relevant keywords are `bs_interp_method` and `bs_interp_mode`, while tests `v67mbpt#32` to `v67mbpt#35` present examples of such calculations

Furthermore, the temperature-dependent optical absorption can now be computed. This is described in Ref. [132], Section 5.2. The approximation described in Ref. [133] is used. Namely, the electron–phonon interaction renormalizes the diagonal part of the BSE Hamiltonian according to:

$$H_{v\mathbf{c}\mathbf{k},v'\mathbf{c}'\mathbf{k}'}(T) = H_{v\mathbf{c}\mathbf{k},v'\mathbf{c}'\mathbf{k}'}^{FA} + [\Delta\epsilon_{\mathbf{c}\mathbf{k}}(T) - \Delta\epsilon_{v\mathbf{k}}(T)]\delta_{v\mathbf{v}'}\delta_{\mathbf{c}\mathbf{c}'}\delta_{\mathbf{k}\mathbf{k}'}, \quad (20)$$

where H^{FA} is the “frozen-atom” expression for the BSE Hamiltonian introduced in Ref. [17], Section 3.5.1. The diagonalization of Eq. (20) gives complex excitonic eigenenergies and therefore finite (temperature-dependent) lifetimes, and therefore requires a much more careful treatment than the usual Bethe–Salpeter equation. Indeed, the matrix is not Hermitian anymore. It can be diagonalized using a generic non-Hermitian algorithm, but also an iterative Bi-Lanczos algorithm can be used to avoid the direct diagonalization of Eq. (20). More details are given in Ref. [132], Section 5.2 and App. F of that article. An example of such calculations is presented in Fig. 5. Relevant keywords for the temperature dependence are `bs_calc_type = 1`, `bs_calc_algo = 1` or `2`. Also, a QPS file that includes the temperature-dependent electronic eigenenergies must be included. The tests `v67mbpt#50` and `v67mbpt#51` do not really test such capability, but feed in a BSE calculation quasi-particle energy file which comes from GW, that might alternatively come from a temperature-dependent electronic structure calculation, described in Sections 3.3 and 5.1.

3.7. DMFT: Progress in electronic correlations

Strongly correlated materials exhibit a wide variety of phenomena like magnetism, unconventional superconductivity, metal–insulator transitions and multiferroicity. Strong correlation generally originates from Coulomb repulsion in partially filled localized d or f shells, and is not very well described by current exchange–correlation functionals. Including these effects can be accomplished in a first-principles approach by adding explicitly in the Hamiltonian the Coulomb interaction for the localized correlated orbitals. This Hamiltonian can be solved in the static mean-field approximation, giving the DFT+ U method. A more accurate solution is provided by Dynamical Mean-Field Theory (DMFT) [134]. In this approach, local correlations are exactly solved by mapping each correlated atom to an effective Anderson impurity model (AIM), which is solved using an impurity solver such as the continuous-time quantum Monte Carlo method (CT-QMC) [135–137]. The combination of DFT with DMFT [138] and the calculation of U with constrained Random Phase Approximation were already present in `abinit` v7 and are described in Ref. [17], Sections 3.6 and 3.5.2. In the following, we detail recent advances in these implementations.

First, the DFT+ U implementation was extended by including the possibility of performing “charge only-DFT”+ U calculations as proposed in Ref. [139,140] in order to improve the description of exchange. In this scheme, the exchange and correlation potential

and energy are computed using the total charge density and not the spin resolved densities. This possibility can be enabled for the full localized double counting using the input variable `usepawu = 4`. An example of such a calculation is given in test `v8#35`. This implementation was used in Ref. [141].

Secondly, the DFT+DMFT parallelism was improved for large systems. In particular, it is now possible to parallelize the calculation on both \mathbf{k} -points and bands/ \mathbf{g} -vectors by using the input variable `paral_kgb = 1` and related input variables. See examples of these calculations in tests `paral#84`, `paral#85`, and `paral#86`.

Finally, two new approaches to CT-QMC have been added to solve the AIM. In the first one, the density–density CT-QMC code available in `abinit` [17,142] was generalized in order to take into account off-diagonal elements of the hybridization function. This implementation is activated with the input variable `dmft_solv = 8`. Spin–orbit coupling calculations are possible, but using a real valued imaginary time hybridization function. An example of such a calculation is available in test `paral#83`. This solver was used in Refs. [143,144].

In the second approach, we use the Toolbox for Research on Interacting Quantum System (TRIQS) library [53], which is an open-source project that provides a framework for many-body quantum physics and more specifically for strongly-correlated electronic systems. TRIQS provides an open source implementation of the continuous-time hybridization expansion quantum impurity solver (CT-HYB) [145], considered a state-of-the-art solver for multi-orbital AIM. An interface between `abinit` and the impurity solver TRIQS/CT-HYB is now available and will make use of the independent progress made by the TRIQS library. Relevant input variables for the choice of impurity solver are: `dmft_solv = 6` or `7` for the CT-HYB solver with the density–density part or the fully rotationally invariant formulation of the interacting Hamiltonian, respectively. TRIQS uses a linear frequency mesh, the `dmft_nwli` keyword specifies the number of frequencies for the entire mesh and the keyword `dmftqmc_l` for the number of time slices for the imaginary time Green function are constrained to `dmftqmc_l > 2 × dmft_nwli`. The TRIQS/CT-HYB library uses the Legendre polynomial basis to efficiently represent Green functions as in Ref. [146]. The input variable which fixes this basis size is `dmftctqmc_triqs_nleg`. Tests `paral#99` and `v8#01` present examples of such calculations.

3.8. Progress in high-performance computing

In the race to increase computing power, supercomputer architectures have recently changed considerably. Currently, a modern supercomputer can typically perform between dozens and hundreds of teraflops for a few megawatts. This increase in power is due to the use of a new type of processing units, taking advantage of multitasking parallelism, via dedicated graphical processing units, and vectorization in many-integrated-core processors.

To take advantage of this evolution of computing architectures, we included new features in the `abinit` code. Adaptations for GPUs and CPUs, although implemented a few years ago, were no longer suitable to take advantage of *many-core* processors.

The current implementation is based on a new abstract layer which manages the memory allocations/deallocations, and drives the linear algebra calculations with a hybrid MPI+OpenMP paradigm. The old *Locally Optimal Block Preconditioned Conjugate Gradient* (LOBPCG) implementation [147] has been replaced by a new one written from scratch, focusing on the memory footprint and the computational efficiency. The new wavefunction optimization algorithm (`wfoptalg = 114`) is automatically activated when the *band-FFT-k-point* parallelization is activated

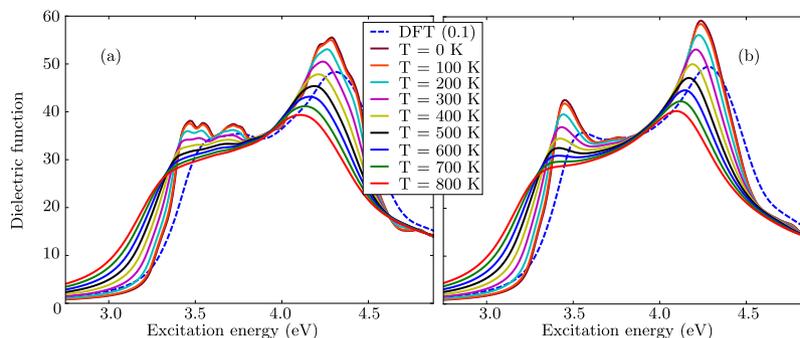


Fig. 5. Imaginary part of the temperature-dependent dielectric function of silicon obtained (a) from a $12 \times 12 \times 12$ k-point grid with $2 \times 2 \times 2$ multiple shifts and (b) from a $16 \times 16 \times 16$ k-point grid with $4 \times 4 \times 4$ multiple shifts. DFT (0.1) is the frozen-atoms results with a numerical broadening of 0.1 eV. From Ref. [132], Section 5.2.

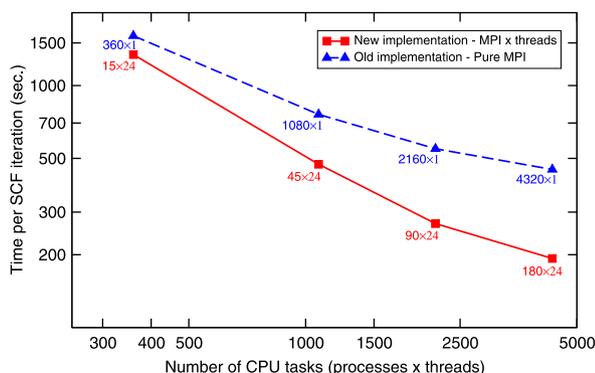


Fig. 6. Performances of *abinit* Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) algorithm on a supercomputer. Comparison of old (2008) and new (2018) implementations. The graph shows the time to solution for one iteration of the self-consistent cycle. The test case is a 1960-atom crystal of Ga_2O_3 (8640 electronic valence bands, 1 \mathbf{k} -point, no spin polarization). The underlying hardware is the French *Joliot-Curie* supercomputer (TGCC computing center), a cluster of *Intel Skylake* 48-core nodes. The old implementation uses only MPI processes whereas the new one is based on hybrid MPI-openMP parallelism (24 tasks per MPI process in the present test).

(`paral_kgb` = 1). A compilation of ABINIT with both MPI and OpenMP is recommended to get the best of this implementation. Currently, best performance is obtained with the use of *Intel MKL* multi-threaded library.³

To run a calculation with both OpenMP and MPI, first the user *should not* try to activate MPI FFT parallelism (setting `npfft` \neq 1), as this is enforced by the code when *abinit* runs in multi-threading mode. The parallelization over \mathbf{k} -point grid must be fully exploited, because it is the most efficient one. Then the number of MPI processes dedicated to band parallelism (`npband` keyword) and the number of bands per MPI process (`bandpp` keyword) should be chosen so that their product is a divisor of the number of bands (`nband`) and the closest to `nband`; ideally `npband` \times `bandpp` = `nband`. In the latter case, the algorithm reaches its highest efficiency and numerical stability with the inconveniences of an increase of the memory and longer self-consistent electronic steps. A comparison between the old algorithm and the new algorithm described here is illustrated in Fig. 6. In cases where *abinit* does not use multi-threading, a new implementation of ScalAPACK is used for some critical parts. The keyword `sil_krankpp` can be set to choose the size of a matrix per MPI process. Usually, a value between 100 and 1000 is fine; the advanced user should run tests to find out the best

³ link *abinit* with `-lmkl(intel|gnu)_thread` instead of `-lmkl_sequential` and invoke `--enable-openmp` during the configuration of *abinit*.

adequate value which depends on the architecture. Then, the code will automatically switch on/off the use of ScalAPACK and the number of MPI processes. To impose the number of MPI processes used, one can set the `np_silk` input keyword to the desired value. Note that the new ScalAPACK implementation is *not* compatible with multi-threading due to non *thread safety* of some critical functions, whatever the implementation.

Finally, note that ScalAPACK can be replaced by the *Eigenvalue solvers for Petaflop-Applications* (ELPA) library [52] to obtain even better performances.

4. New projects within ABINIT: MULTIBINIT and A-TDEP

4.1. MULTIBINIT: a second-principles approach to materials properties

Although DFT software applications, such as *abinit*, allow one to compute increasingly complex properties of increasingly complex systems, first-principles computations remain extremely demanding in terms of computer time, limiting their practical use to relatively small unit cells (a few hundred atoms) and time-scales (a few ps). In order to overcome this limitation, many efforts have been devoted to the development of so-called “second-principles” methods, which use first-principles data to build effective potentials [148–155]. These target the modeling of mesoscale systems (up to a few hundred thousand atoms) at operating conditions (finite temperature, appropriate mechanical constraints) while retaining most of the first-principles accuracy and predictive power.

The new *multibinit* application (Fig. 7) has the goals of automatizing and integrating (i) the construction of second-principles effective models for distinct degrees of freedom, such as lattice, spin or electrons; and (ii) the use of these models to access dynamical properties at finite temperatures, either independently or coupled together. *multibinit* has a modular design targeting high extensibility. A basic architecture, common to all degrees of freedom, was put in place so that the physics, including the potentials and equations of motion, can be expressed in a systematic way, allowing for an easier treatment of a combination of distinct degrees of freedom (for example, simultaneous spin and lattice). The first release of *multibinit* was restricted to the atomistic lattice model (Fig. 7, red part). The next release will include the spin model (Fig. 7, cyan part). Ongoing developments concern the spin-lattice coupling, the projection on lattice modes and the interface with the SCALE-UP electron module (Fig. 7, gray part).

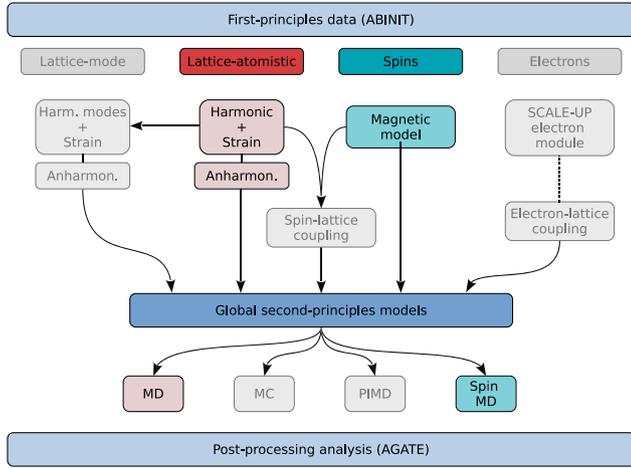


Fig. 7. Sketch of the structure of the integrated multibinit application. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.1. Lattice model

At the lattice level, multibinit relies on a Taylor series expansion of the Born–Oppenheimer potential energy surface (PES) around a reference structure, in powers of individual atomic displacements (R) and macroscopic strains (η), as proposed in Ref. [150] at both harmonic (*har*) and anharmonic (*anh*) levels:

$$E^{PES}[R, \eta] = (E_{phon}^{har}[R] + E_{phon}^{anh}[R]) + E_{ela}^{har}[\eta] + (E_{p-s}^{har}[R, \eta] + E_{p-s}^{anh}[R, \eta]). \quad (21)$$

multibinit can construct a lattice atomistic model from first-principles data provided in two separate input files, according to the following procedure.

The first input file is the ABINIT DDB containing all the second energy derivatives as calculated by linear response providing all harmonic terms in Eq. (21). $E_{phon}^{har}[R]$ is related to the real-space interatomic force constants, including a short-range part and, for insulators, a long-range dipole–dipole interaction. The latter is estimated from the Born effective charges and optical dielectric tensor, and treated explicitly [156,157]. $E_{ela}^{har}[\eta]$ is associated to the elastic constants. $E_{p-s}^{har}[R, \eta]$ is related to atom-strain couplings.

Determining the anharmonic part ($E_{phon}^{anh}[R] + E_{p-s}^{anh}[R, \eta]$) is more delicate: a second input file in the abinit HIST.nc (history) format provides DFT energies, forces and stresses for a set of ionic configurations, which will be used as a training set. multibinit generates all symmetry adapted terms (SAT) [150] compatible with the reference configuration, up to a given order and range (see input variables `fit_*`) of interactions provided by the user. Then, the most relevant terms are selected and their coefficients fitted to reproduce the training set of DFT forces and stresses, following the optimization procedure proposed by Escorihuela et al. [158]. The training set should contain a representative “unbiased” sampling of the PES, and can be generated in many different ways. Currently we favor the use of *ab initio* molecular dynamics runs (abinit, `ionmov` = 12, 13) at different temperatures, augmented by an automatic generation of distorted structures from random occupations of stable phonon modes (abinit, `ionmov` = 27).

In some situations, this automatically generated model will yield an energy which is not bound from below, for large displacements. An automatic bounding procedure was implemented (`bound_model` = 2), to identify and add appropriate high order terms with positive coefficients that improve the description of

the PES by confining the atomic displacements. A new more efficient procedure (`bound_model` = 3) is under development, which identifies the complete set of even high order terms with positive coefficients necessary to bound the potential.

Once the effective $E^{PES}[R, \eta]$ is known, it can be used for finite temperature simulations in much larger supercells (still imposing periodic boundary conditions). At this stage, multibinit is restricted to classical molecular dynamics simulations, but other options are under development (Monte-Carlo, path-integral molecular dynamics).

We note that such a lattice model is a direct generalization of the effective Hamiltonian approach introduced during the 1990s by Zhong, Vanderbilt and Rabe [159] to study structural phase transitions in perovskites. A new option (`eff_hamilt` = 1) is presently under development within multibinit, in order to project the full E^{PES} in the restricted subspace spanned by one or a few lattice Wannier functions [160]. This will enable simulations within a restricted subspace of vibrational modes, and with even larger supercells.

To help the user post-process the output of multibinit (as well as abinit and other software), a specific tool has been developed named Agate [161] (formerly APPA). The basic usage of Agate is through commands in a vim-like environment, and a more user-friendly GUI named qAgate [162] is also available. The main features include the calculation of mean square displacements, velocity autocorrelation functions, pair distribution function, or the graphical representation of the evolution of lattice and atomic positions, temperature, pressure, stress and other properties. One particularly useful function is the ability to project the trajectory onto phonon modes. More can be found at the website [161]. Invoking Agate with the `--help` option will also provide all the available features and functionalities.

4.1.2. Spin model

At the spin level, multibinit implements the most commonly used model for spin systems, via the Heisenberg Hamiltonian including magnetic exchange and Dzyaloshinskii–Moriya [163,164] interactions. Single ion anisotropy and dipole–dipole interactions are also included, and all terms bear a very strong similarity to the quadratic part of the lattice model Hamiltonian. A number of open source spin dynamics codes already exist, such as UPPASD [165], VAMPIR [166], OOMF [167]; the distinguishing features of multibinit are the integration with abinit, to fit parameters, and the simultaneous dynamics with other degrees of freedom (in particular using the inter-atomic force constants). The spin features of multibinit are functional in the forthcoming version of abinit, and continue to be developed.

The Heisenberg Hamiltonian parameters are generated using the magnetic force theorem, in which a local spin rotation is treated as a perturbation [168]. As abinit uses a plane wave basis set, the spin localized around a specific atom is not readily available. The localized perturbation is therefore applied to a Wannier-function based tight-binding Hamiltonian, which is mapped from the abinit results using the abinit–Wannier90 interface [169]. The method is implemented in a stand-alone Python code TB2, which requires a Wannier-function Hamiltonian built from a single ground-state DFT calculation and generates the inputs for multibinit directly.

The dynamics of the spin system can be simulated at finite temperature with either the Landau–Lifshitz–Gilbert equation [170] or the Monte-Carlo method. Subsequently, thermodynamic quantities can be calculated such as macroscopic magnetic moments, susceptibility, and specific heat. Both the generation of the parameters and the spin dynamics are described in a tutorial.

The coupling between the spin and lattice degrees of freedom is currently being implemented including the first two non-vanishing coupling terms: the first term is quadratic in the spin

variables and linear in the lattice distortion, and the second term is quadratic in both the spin and the lattice degrees of freedom. Other terms will be added at a later stage, in particular the bilinear one, which is linear in both spin and lattice variables (and is relevant when spin-orbit coupling is included).

4.1.3. Electron model

The advantage of an effective lattice model as described above is that it integrates out the electronic degrees of freedom, which strongly simplifies and accelerates the model. Although avoiding the explicit treatment of the electrons is typically seen as an advantage for low-cost lattice dynamics, it can be a drawback for addressing problems in which the system can naturally evolve on different PES, showing distinct spin, orbital or charge orderings, or is moved away from the initial PES, say by electronic excitation, or electron doping. The next release of `multibinit` will be interfaced with the external SCALE-UP module [55] which reintroduces the explicit treatment of a small number of selected electronic degrees of freedom and their coupling to the lattice degrees of freedom, while avoiding double counting, as described by Garcia et al. [151].

4.2. A-TDEP: Temperature dependent effective potential for ABINIT

When considering ground state calculations, numerous linear and non-linear response functions are available in `abinit`, in particular through an intensive use of DFPT. In most situations, the quasi-harmonic approximation (QHA) allows one to extrapolate the 0 K results obtained to finite temperature, see for example the study of thermal expansion and temperature-dependent elastic constants of Si in Ref. [171]. However, the QHA fails or cannot be applied in various particular cases: when the system is close to a phase transition, when the phase of interest is not stable at 0 K, or when one wants properties at high temperature.

In these situations, an implicit treatment of the temperature through a volume variation is no longer sufficient and a dedicated calculation, taking into account the temperature explicitly, is needed. In particular, by performing *ab initio* molecular dynamics (MD) or Monte-Carlo (MC) simulations, one can capture the anharmonic features involved in the phonon-phonon scattering and lacking in the QHA approximation.

One way to extract all the lattice dynamic properties from MD simulations was proposed by Esfarjani and Stokes [172] then developed by Hellman et al. [173–175]. Named “Temperature Dependent Effective Potential” (TDEP), this formalism uses a least-squares method to compute the temperature-dependent interatomic force constants (IFC). As highlighted in several publications, this method efficiently captures a large variety of explicit thermal effects [176–180].

Formally, at each time step t , the MD potential energy $U_{\text{MD}}(t)$ can be expanded around equilibrium positions $\tau_i(0)$ as:

$$U_{\text{MD}}(t) = U_0 + \sum_p \frac{1}{p!} \sum_{i\dots k, \alpha\dots\gamma} \Theta_{i\dots k}^{\alpha\dots\gamma}(p) u_{i,\text{MD}}^\alpha(t) \dots u_{k,\text{MD}}^\gamma(t), \quad (22)$$

with $u_{i,\text{MD}}^\alpha(t) = \tau_i^\alpha(t) - \tau_i^\alpha(0)$ the displacements and $\Theta_{i\dots k}^{\alpha\dots\gamma}(p)$, the IFC matrix at p th order. In this framework, the force acting on atom i in direction α can be written:

$$F_{i,\text{MD}}^\alpha(t) = \sum_{p\lambda} f_{i,p\lambda}^\alpha(\mathbf{u}_{\text{MD}}(t)) \theta^{p\lambda}, \quad (23)$$

with $\theta^{p\lambda}$ the λ IFC parameters at the p th order and $f_{i,p\lambda}^\alpha(\mathbf{u}_{\text{MD}}(t))$ a function gathering all the contributions coming from the atomic displacements. When the number of MD time steps is large, one can solve this system of equations by searching its least squares

solution. Let us define the residual $\mathcal{R} = \mathcal{F}_{\text{MD}} - \mathbf{f} \cdot \Theta$. To obtain the best solution, the measure $\mathcal{S} = \min(\mathcal{R} \cdot \mathcal{R}) = \|\mathcal{F}_{\text{MD}} - \mathbf{f} \cdot \Theta\|^2$ is minimized with respect to Θ . The solution giving the lowest residual is the least squares solution $\Theta = \mathbf{f}^\dagger \cdot \mathcal{F}_{\text{MD}}$, with \mathbf{f}^\dagger the pseudoinverse of the \mathbf{f} matrix.

This numerical process is efficient when the number of equations (proportional to the time steps) is large with respect to the unknowns (the number of IFC coefficients). But in practice, thousands of time steps would be needed to achieve a reasonable convergence. By using some invariance properties (translation and rotation) of the system and the symmetries of the crystal, it is possible to reduce the non-zero independent IFC coefficients to tens and decrease by several orders of magnitude the computational cost.

Once the temperature dependent IFC coefficients are obtained, a large number of dynamic, elastic and thermodynamic properties can be evaluated. The first one is the temperature dependent dynamical matrix:

$$D_{ij}^{\alpha\beta}(\mathbf{q}) = \sum_b \frac{\Theta_{ij}^{\alpha\beta}(0, b)}{\sqrt{M_i M_j}} \exp(i\mathbf{q} \cdot [\mathbf{R}(b) - \mathbf{R}(0)]), \quad (24)$$

with \mathbf{q} a wavevector, M_i the mass of atom i and $\mathbf{R}(b)$ the lattice vector going to cell b . The phonon eigenfrequencies, $\omega(\mathbf{q})$, and eigenvectors, $X_i^\alpha(\mathbf{q})$, are then defined by:

$$\sum_{\beta j} D_{ij}^{\alpha\beta}(\mathbf{q}) X_{js}^\beta(\mathbf{q}) = \omega_s^2(\mathbf{q}) X_{is}^\alpha(\mathbf{q}). \quad (25)$$

In this equation the quantum number s refers to the quantification of the eigenfrequencies $\omega_s(\mathbf{q})$. Using the (temperature dependent) phonon frequencies, it becomes easy to build the phonon density of states, $g(\omega)$, and various thermodynamic properties: the vibrational part of the free energy F_{vib} , the internal vibrational energy U_{vib} , the specific heat C_v and the entropy S_{vib} in a same manner as it is performed for the three-dimensional quantum harmonic crystal [181]. In these formulations, the thermodynamic properties no longer depend only on temperature through the statistics or volume $V(T)$ but also explicitly through the use of phonon frequencies, $\omega(T, V(T))$.

The temperature dependent elastic constants are evaluated using the second order IFC [182,183] such as:

$$C_{\alpha\beta\gamma\delta} = A_{\alpha\gamma\beta\delta} + A_{\beta\gamma\alpha\delta} - A_{\alpha\beta\gamma\delta}, \quad (26)$$

$$A_{\alpha\beta\gamma\delta} = \frac{1}{2V} \sum_{ij} \Theta_{ij}^{\alpha\beta} d_{ij}^\gamma d_{ij}^\delta, \quad (27)$$

with $d_{ij}^\gamma = \tau_i^\gamma - \tau_j^\gamma$, and where V is the volume. Using these elastic constants, it is straightforward to build the elastic moduli, the isothermal bulk modulus K_T , the shear G modulus, and others. Note that this method needs to have long-range IFCs in order to yield converged quantities [184,185].

By using the temperature dependent third-order IFC, it becomes possible to evaluate the thermodynamic Grüneisen parameter γ [186,187] defined as:

$$\gamma = \frac{\sum_{s,q} \gamma_s(\mathbf{q}) C_{v,s}(\mathbf{q})}{C_v} \quad \text{with} \quad (28)$$

$$\gamma_s(\mathbf{q}) = \sum_{ijk, bc, \alpha\beta\gamma} \frac{\Theta_{ijk}^{\alpha\beta\gamma}(0, b, c) X_{is}^{\alpha\alpha}(\mathbf{q}) X_{js}^\beta(\mathbf{q})}{6\omega_s^2(\mathbf{q}) \sqrt{M_i M_j}} \tau_k^\gamma \exp[i\mathbf{q} \cdot \mathbf{R}(b)]. \quad (29)$$

Using the Grüneisen parameter, a variety of properties become available: the thermal expansion, the isentropic compressibility, the constant pressure specific heat, the sound velocities, and so forth.

The ABINIT implementation of the TDEP algorithm, referred to as A-TDEP, has one main executable `tdep`. All these elastic and

thermodynamic quantities are available in various output files after a single run of `tdep`. The user needs only to provide an input file with around ten keywords and a `NETCDF` file with the MD trajectory. The `tdep` post-processor can be launched in the same manner as `abinit:tdep < tdep.files > tdep.log`. We recommend the user to read the `A-TDEP` documentation distributed in the `ABINIT` package before any calculations, and also to read the article dedicated to this implementation [188].

5. Other recent developments within the `ABINIT` package

5.1. Electron–phonon self-energies and spectral functions in the Python module `ElectronPhononCoupling`

Different modules within the `ABINIT` package allow one to compute the electronic self-energy due to electron–phonon interactions. The latest implementation that we describe here employs new strategies to accelerate its convergence properties. It is implemented in a Python module called `ELECTRONPHONONCOUPLING`, located in the directory `scripts/post_processing/ElectronPhononCoupling` of the `ABINIT` package. The module uses the data produced in DFPT calculations to compute the self-energies and spectral functions in a post-processing step.

The electron self-energies due to electron–phonon interactions contain information on the temperature dependence of the electronic energies as well as the lifetime of the electronic states. Two distinct diagrams contribute to the self-energy: the Fan term and the Debye–Waller term, written as $\Sigma = \Sigma^{\text{Fan}} + \Sigma^{\text{DW}}$, whose expressions are

$$\Sigma_{\mathbf{k}n}^{\text{Fan}}(T, \varepsilon) = \sum_{\mathbf{q}v} \sum_m w_{\mathbf{q}} |g_{m\nu}(\mathbf{k}, \mathbf{q})|^2 \times \left[\frac{n_{\mathbf{q}v}(T) + f_{\mathbf{k}+\mathbf{q}m}(T)}{\varepsilon - \varepsilon_{\mathbf{k}+\mathbf{q}m} + \hbar\omega_{\mathbf{q}v} + i\eta} + \frac{n_{\mathbf{q}v}(T) + 1 - f_{\mathbf{k}+\mathbf{q}m}(T)}{\varepsilon - \varepsilon_{\mathbf{k}+\mathbf{q}m} - \hbar\omega_{\mathbf{q}v} + i\eta} \right] \quad (30)$$

and

$$\Sigma_{\mathbf{k}n}^{\text{DW}}(T) = - \sum_{\mathbf{q}v} \sum_m w_{\mathbf{q}} \frac{|g_{m\nu}^{\text{DW}}(\mathbf{k}, \mathbf{q})|^2}{\varepsilon_{\mathbf{k}n} - \varepsilon_{\mathbf{k}m}} \left[n_{\mathbf{q}v}(T) + \frac{1}{2} \right]. \quad (31)$$

In these expressions, \mathbf{k} and n , m label an electron wavevector and band index, \mathbf{q} and v label a phonon wavevector and branch index, $\varepsilon_{\mathbf{k}+\mathbf{q}m}$ is an electronic eigenvalue with corresponding Fermi–Dirac occupation number $f_{\mathbf{k}+\mathbf{q}m}$, and $\omega_{\mathbf{q}v}$ is a phonon frequency with corresponding Bose–Einstein occupation number $n_{\mathbf{q}v}(T)$. The self-energy $\Sigma_{\mathbf{k}n}$ is defined for each electronic state $(\mathbf{k}n)$, and is a function of temperature (T) and energy (ε). The quantities $g_{m\nu}(\mathbf{k}, \mathbf{q})$ and $g_{m\nu}^{\text{DW}}(\mathbf{k}, \mathbf{q})$ are electron–phonon coupling matrix elements whose expressions can be found in Refs. [189,190]. To each phonon wavevector corresponds a weight $w_{\mathbf{q}}$, which is proportional to the fraction of the Brillouin zone it samples and the number of symmetry-equivalent wavevectors. The broadening parameter η is real and infinitesimal, and formally preserves causality in the Green function of the electron and the holes. In practice, η must be converged to the smallest value consistent with the number of \mathbf{q} -points, as discussed in Refs. [191–193].

The summation over intermediate bands m can be split using a certain band cutoff M , which defines a lower bands contribution ($m \leq M$) and an upper bands contribution ($m > M$). For the lower bands contribution, the summation over intermediate bands is carried explicitly using Eqs. (30) and (31). For the upper bands, the summation can be eliminated exactly by solving the Sternheimer equation for the perturbed electronic states, as described in Ref. [194]. Moreover, the upper bands contribution can be approximated with a static expression, as discussed in [191].

The most difficult aspect to the computation of the self-energy is the convergence with respect to \mathbf{q} -point sampling of phonon

wavevectors. This difficulty is circumvented by first defining two \mathbf{q} -point grids: a *coarse* grid and a *fine* grid. The electron–phonon coupling potential is interpolated onto the fine grid with the procedure described in Section 3.3. An important observation is that the fine \mathbf{q} -point grid is required only for the lower bands contribution to the self-energy, because the denominators in Eq. (30) may become small for certain intermediate bands. For the upper bands, however, these denominators are always large. Their contribution converges quickly with the number of \mathbf{q} -points, and may be evaluated on the coarse grid.

The workflow to compute the self-energy is as follows. First, a DFPT calculation is performed for each \mathbf{q} -point of the coarse grid and for each perturbation, that is, the displacement of each atom along the 3 cartesian directions. For polar materials, the electric field response along the 3 cartesian directions is also required. These steps are activated with the keyword `rfphon = 1`, and produce the response potential files with suffix `POTx` (where x is a number), as well as dynamical matrix files with suffix `DDB.nc` (where `.nc` indicates the `NETCDF` file format). The upper bands contribution to the self-energy is then computed with keyword `ieig2rf = 5`, and the result is stored in files with suffix `EIGR2D.nc`. Next, the response potential files are merged in a file with suffix `DVDB` using the utility `mrqdv`, while the dynamical matrix files are merged with the utility `mrqddb`. The interpolation of the response potentials onto the fine \mathbf{q} -grid is performed by the main `abinit` executable with keywords `optdriver = 7` and `eph_task = 5`, and the interpolation of the dynamical matrices is performed with `anaddb`. The g matrix elements are then computed in the cartesian basis and stored in files with suffix `GKK.nc`. This is done by `abinit` with input variables `optdriver = 7` and `eph_task = 2`, as and the different `GKK.nc` files are merged with the `merge_gkk.nc.py` script.

Finally, the self-energy is computed with the `ELECTRONPHONONCOUPLING` package, which reads in the files `DDB.nc`, `GKK.nc`, `EIGR2D.nc`. The list of \mathbf{q} -point weights ($w_{\mathbf{q}}$) must be provided by the user, which allows for maximum flexibility in the choice of the fine \mathbf{q} -points grid. For examples, one could generate a random set of \mathbf{q} -points to sample the Brillouin zone. Another approach is to use adaptative \mathbf{q} -point grids that increase the sampling near the center of the Brillouin zone. The `ELECTRONPHONONCOUPLING` package features scripts that generate such adaptative \mathbf{q} -point grids. The computation of each \mathbf{q} -point contribution to the self-energy is distributed with MPI parallelism.

5.2. Grüneisen parameters and DOS in the `anaddb` post-processor

The mode Grüneisen parameters are the logarithmic derivatives of the phonon frequencies with respect to the volume V :

$$\gamma_s(\mathbf{q}) = - \frac{\partial \ln \omega_s(\mathbf{q})}{\partial \ln V} = - \frac{V}{\omega_s(\mathbf{q})} \frac{\partial \omega_s(\mathbf{q})}{\partial V}, \quad (32)$$

where the phonon frequencies $\omega_s(\mathbf{q})$ and the eigenvectors $X_s(\mathbf{q})$ are solution of the eigenvalue problem:

$$D(\mathbf{q})X_s(\mathbf{q}) = \omega_s^2(\mathbf{q})X_s(\mathbf{q}), \quad (33)$$

involving the dynamical matrix $D(\mathbf{q})$. Using the normalization condition $\langle X_s(\mathbf{q}) | X_{s'}(\mathbf{q}) \rangle = \delta_{ss'}$ and the Hellmann–Feynman theorem, one obtains

$$\begin{aligned} \frac{\partial \omega_s^2(\mathbf{q})}{\partial V} &= 2\omega_s(\mathbf{q}) \frac{\partial \omega_s(\mathbf{q})}{\partial V} = \frac{\partial \langle X_s(\mathbf{q}) | D(\mathbf{q}) | X_s(\mathbf{q}) \rangle}{\partial V} \\ &= \langle X_s(\mathbf{q}) | \frac{\partial D(\mathbf{q})}{\partial V} | X_s(\mathbf{q}) \rangle, \end{aligned} \quad (34)$$

which allows one to express the Grüneisen parameter in terms of the matrix element of the *derivative* of the dynamical matrix.

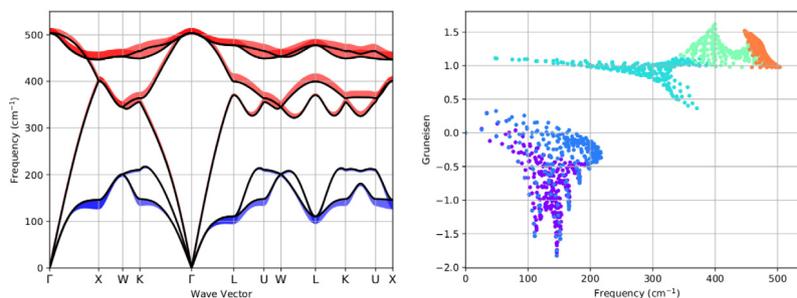


Fig. 8. Grüneisen parameters for bulk silicon, represented as fat bands (left) or a scatter plot (right). The different colors in the fat bands correspond to different signs, while the colors in the scatter plot indicate the band index.

The `anaddb` code evaluates $\partial D(\mathbf{q})/\partial V$ using central finite differences around the equilibrium volume V_0 ; the user is supposed to perform structural relaxations for different volumes followed by the DFPT computation of the vibrational spectrum and use the output DDB files for the finite difference. The list of DDB files is specified by `gruns_dpbs` while `gruns_ndpbs` gives the number of files (at present, only central finite differences of order 3, 5, 7 and 9 are supported). The code produces text and `NETCDF` files with the Grüneisen parameters along a \mathbf{q} -path, the Grüneisen DOS and the average value. The `NETCDF` files can be analyzed with the following `AbiPY` script to produce plots like those shown in Fig. 8. An example of an input file is provided in [v8#45](#). For an application of the technique, see [\[195\]](#).

5.3. Documentation and bibliography

Most of the `ABINIT` documentation is now written in [Markdown](#), a lightweight markup language with plain text formatting syntax. In addition to the basic Markdown syntax, the `ABINIT` documentation supports extensions and shortcuts to ease the inclusion of hyperlinks, bibliographic citations in `bibtex` format as well as `TeX` equations thanks to the [MathJax](#) JavaScript library. The website is automatically generated with `MkDocs`, a static site generator geared toward project documentation. `MkDocs` employs [Python-Markdown](#) to parse the Markdown documentation and uses a single `YAML` configuration file (`mkdocs.yml`) defining the organization of the pages on the website. The website uses `Mkdocs-Material`, a theme built using Google's [Material Design](#) guidelines. Navigation bars, header and footer are generated automatically by the framework using the [Jinja2](#) template engine.

As a net result, `ABINIT` developers can now write attractive documentation without having to use HTML explicitly while working in an environment that is well integrated with the other parts of the package (the Python database of input variables, the `ABINIT` test suite with the corresponding input and output files, the file with `bibtex` citations). From a user perspective, the new website appears more interactive and user-friendly thanks to the presence of several internal cross-references to important resources such as input files, the detailed description of input variables and internal links to the so-called *Topics page* that provides an introduction to a particular subject or feature of the code with links to the associated input variables and input files.

5.4. Build system, and environment for development

Over the last few years, the way the build system of `ABINIT` is used has changed dramatically. In particular, when counting the number of builds, human operators have increasingly been replaced by computer programs. Through the deployment of various build frameworks such as [EasyBuild](#) [\[196\]](#) and [Spack](#) [\[197\]](#),

as well as improved collaborations with vendor-specific platforms (e.g., [MacPorts](#)), more and more end users can access binary versions of `ABINIT` without having to compile them. For those who still need to compile the source code, it is desirable to improve the level of automation and have a better tolerance to faults from the build system. On the side of developers, a better integration between Fortran and C/C++ has become necessary, due to the migration of code blocks and features away from Fortran, both for portability reasons and for easier maintenance. All these changes have led to a whole new set of specifications for the build system, the implementation of which is still in progress at the time of writing.

Various efforts have been dedicated to address technical debt, among which the removal of a circular dependency between `ABINIT` and `BigDFT`, pressured by the release of the Poisson Solver of the latter as a standalone package [\[198\]](#). Up to now, `BigDFT` had contained copies of low-level components of `ABINIT`, which was greatly complicating the build of `ABINIT` when using `BigDFT` features. With the split of the source tree of `ABINIT` into a shared set of low-level components and a higher-level `ABINIT`-only source tree, this circular dependency will come to an end. This effort has already permitted the use of the low-level PAW components of `ABINIT` in another software package [\[57\]](#).

The decision of Canonical, Inc., to stop the development of `Bazaar`, as well as the scheduled end of Python 2 in 2020, have made necessary the transition of `ABINIT` to a new development model based on `Git` and `Gitlab`. In parallel, the Continuous Integration (CI) infrastructure has been thoroughly upgraded.

Collaborations between `ABINIT` and other developer teams have expanded within the electronic structure community and lead to new solutions regarding the management of software dependencies, see for example Ref. [\[54\]](#). The number of alternatives to the `ABINIT` fallbacks is constantly increasing and will soon supersede them. As a result, the fallbacks have been detached from the core of `ABINIT` and are now maintained within a standalone project, though snapshots are still distributed with the source package.

5.5. Test suite and related analysis: ongoing developments

Software testing represents a crucial part of any software project and this is especially true for scientific software in which the reproducibility of the physical results is of paramount importance. For this reason, the `ABINIT` package provides a collection of more than 1200 tests integrated with a `BUILDBOT` infrastructure for continuous integration, as described in Section 5.1 and 5.2 of Ref. [\[17\]](#). These tests are executed every time a new merge request is issued on the internal `gitlab` server in order to ensure that new developments maintain compatibility with the previous official implementation. A command-line interface is also provided through the `runtests.py` Python script so that users can validate their executable before running production calculations.

The standard algorithm compares reference and output files in a line-by-line fashion and all the floating-point numbers are compared using a single tolerance for the relative and the absolute variation. This method is easy to implement as it requires few configuration options, but it suffers from several limitations. First of all, this approach cannot cope with files containing different number of lines, a situation that usually occurs in iterative algorithms in which the number of iterations required to reach convergence may depend on several factors such as the number of MPI processes, the number of OpenMP threads, the input parameters as well as the optimization options used to compile the code. Moreover, output files usually contain many numerical results spanning a wide numerical range and each quantity has its own numerical noise that, in some cases, may be of the same order as the numerical value (this is especially true when the numerical values are close to machine precision, a typical example is given by the residuals of the wavefunctions that are usually converged within 10^{-20}). It is therefore clear that using a single tolerance to compare output results is far from optimal as developers are forced to employ large tolerances to account for the largest fluctuations in the most sensitive quantities.

To address these limitations, a new complementary approach designed to offer a high level of flexibility to compensate the rigidity of the original method has been developed. The new infrastructure is not based on the fragile line-by-line comparison but employs machine-readable YAML documents that are extracted from the main output file. The YAML documents store the most important physical results of the calculation in the form of dictionaries or lists that can be easily converted to Python objects. The Python function used to compare results is therefore data-aware and this allows one to implement specialized logic to compare scalars, vectors and matrices with customized tolerances specified in a YAML *configuration* file associated to the test. This new data-center methodology will facilitate the implementation of new testing patterns such as parametrized tests (tests in which multiple input parameters are changed and the internal consistency of the results is validated) as well as the validation of stochastic algorithms. Last but not least, these machine-readable YAML documents will gradually replace the standard printout in the main output file thus facilitating the implementation of parsers and post-processing tools.

6. Recent developments in ABIPY and high-throughput calculations

ABIPY is a Python package that provides a flexible scripting environment for the analysis and the post-processing of ABINIT calculations as well as tools for the automatic generation of input files and the submission of jobs on parallel architectures. Started as a mere set of scripts to automate the typical tasks needed during software development, ABIPY evolved gradually into a much more powerful and user-friendly toolkit that has been successfully employed in different domains including high-throughput DFPT applications [28], automatic GW calculations [199], generation and validation of pseudopotentials [41] as well as more conventional *ab initio* studies [131,200,201].

ABIPY is interfaced with PYMATGEN [29] which allows users to benefit from the different tools and Python objects available in the PYMATGEN ecosystem and reuse the results available on the MATERIALS PROJECT website [21]. ABIPY can be used in conjunction with matplotlib, pandas, scipy, seaborn, iPython and jupyter notebooks, thus providing a powerful and user-friendly environment for data analysis and visualization. The official documentation is hosted on [github pages](#) and includes a gallery of [plotting scripts](#) that can be easily reused to analyze data and a gallery of [ABIPY workflows](#) showing how to automate *ab initio* calculations

with ABINIT. The [abutorials](#) github repository provides jupyter notebooks with interactive tutorials and examples augmenting the Sphinx-based documentation. Some of the capabilities of the package are also exposed via command-line interfaces whose usage is documented on the official ABIPY [web page](#). The package is available on the PyPi portal and can be installed directly with pip (the official Python package installer) or, alternatively, through conda and the pre-compiled packages available on the [abiconda channel](#).

The ABIPY post-processing tools rely on NETCDF files storing data written according to the ETSF-IO specifications [202], although important text files such as the *abinit* input file or the DDB file with the dynamical matrix are supported as well. At the time of writing, ABIPY supports the majority of the NETCDF files produced by *abinit*. For further information about the Python API, we refer the interested reader to our [collections of jupyter notebooks](#).

A significant part of ABIPY is devoted to the programmatic interface for the automatic generation of input files. The package, indeed, provides a low-level API to create the Python equivalent of an *abinit* input file as well as *factory functions* to automatically generate input files with minimal user intervention. The low-level [interface](#), uses a dictionary-based approach to set explicitly the values of the input variables by name and can mimic the *abinit* dataset mode thus providing an interface that should look familiar to ABINIT users. [Factory functions](#), on the other hand, expose a high-level interface based on meta-variables and Python objects to facilitate the generation of input files in the high-throughput context. Both approaches can be used interchangeably to build the same *AbinitInput* Python object so that users can choose the approach that best suits their needs: users interested in conventional *ab initio* calculations and standard convergence studies may opt for the low-level interface while users interested in automatic calculations will likely rely on factory functions. The generation of input files, however, represents just one part of the more complicated logic required to implement first-principles calculations. *Ab initio* workflows, indeed, involve additional logic to connect the different steps, fix possible runtime errors as well as specialized code to prepare the calculation and post-process the output results. This logic is usually encapsulated in other Python objects that provide some sort of abstraction for the typical operations required to implement scientific workflows (job restart, propagation of output results to children, reception of input from parents, etc). Implementing such logic is not trivial, especially because it requires some sort of data and object persistence. In other words, the intermediate status of the *ab initio* calculations must be stored somewhere so that the framework can restart from it and the final results must be stored somewhere in a machine-readable format so that Python code can use these results to generate new calculations.

To address the problem of data/object persistence, ABIPY uses two different approaches. The first one is based on a lightweight implementation (*flowtk*) that provides a Python interface that is as close as possible to the one used in standard *abinit* calculations and relies on the *pickle* format implemented by the Python standard library to save the status of the calculation. The other approach takes advantage of the MongoDB database and of the Fireworks framework [30] to provide persistence and facilitate the programmatic generation of ABINIT workflows.

The two models share the same code base and have similar APIs but they target two different scenarios. The lightweight implementation is best suited for users who want to have full control of the input parameters or users who are mainly interested in small-scale calculations. The Fireworks-based infrastructure, on the contrary, is mainly designed for high-throughput applications in which the presence of a database with the associated query API

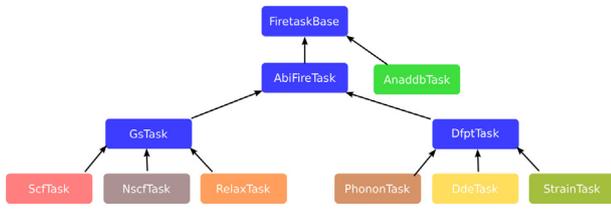


Fig. 9. Class hierarchy of the fireworks tasks implemented in ABIFlows. A *FiretaskBase* subclass is implemented for each kind of ABINIT calculation that can be performed. Arrows indicate the inheritance relationship between the classes.

represents a crucial ingredient. In general, the high-throughput implementation is advantageous when dealing with large number of calculations, where one relies on the previously mentioned factory functions with default options to generate the inputs of different types of calculations. The storage of the outputs is also automated thanks to standardized MongoDB documents that allow an easy and efficient filtering and analysis of the results. In this configuration, one can take full advantage of the features implemented in fireworks to easily distribute the calculations and monitor them through the associated database. All the components related to the firework workflows are implemented in the *ABIFlows* package which provides the basic building blocks used to compose all the fireworks workflows, i.e. the fireworks *Tasks*. These tasks are Python objects designed to address the requirements of the different kind of calculations. A graphical representation of the tasks' class hierarchy is shown in Fig. 9. The package also contains several utilities to generate and customize the ABINIT workflows and the standardized *mongoengine* documents used to store and retrieve the results from the MongoDB database.

The ABIFlows high-throughput infrastructure has been used to execute large number of calculations, from simple structural relaxations up to complex workflows like those required to perform DFPT calculations. A schematic representation of a generic DFPT workflow representing the dependencies between the different steps is depicted in Fig. 10. Its first application has been the realization of a large database of phonon band structures in collaboration with the MATERIALS PROJECT [28,203]. The results of the calculations in the DDB format have been collected and are available on the MATERIALS PROJECT website, along with all the postprocessed quantities like the phonon band structures and DOS, Born effective charges (BECS), dielectric tensors and thermodynamic properties. The web interface also allows users to visualize the crystalline structure with the phonon displacement pattern thanks to the *phononwebsite* [204] web application.

As shown in Fig. 10, the ABIFlows workflow can also be used to compute the third-order derivatives of the total energy with respect to an external electric field through the $(2n + 1)$ theorem [107]. These derivatives are related to the second order electric susceptibility (the nonlinear response of the material to an external electric field) and give access to the effective Second Harmonic Generation (SHG) nonlinear coefficients d_{eff} [205]. Nonlinear optical processes, such as SHG, play an important role in modern optics, especially in laser-related science and technology. They are at the core of a wide variety of applications ranging from optoelectronics to medicine [206]. In Fig. 11, we plot the effective nonlinear coefficient d_{eff} vs. the direct band gap E_g^d obtained by performing automatic DFPT calculations for more than 400 systems employing the above-mentioned ABIFlows workflow. The trend of the data points confirms the inverse relationship between these two quantities. Further analysis of these data is

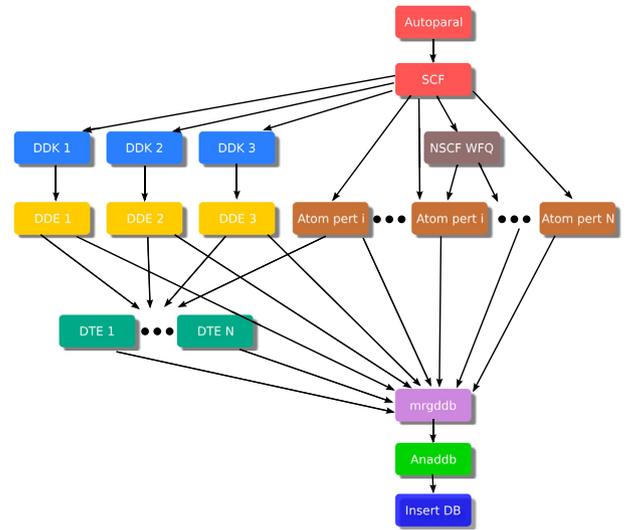


Fig. 10. Schematic representation of the DFPT workflow implemented in ABIFlows. Each box type represents a different kind of calculation performed with *abinit*: self-consistent and non self-consistent calculations to obtain wavefunctions, atomic perturbations with respect to specific atoms and directions, derivatives with respect to the wavevector (DDK) and the electric field (DDE) and third-order derivatives with respect to the electric field (DTE). All the contributions are merged, analyzed and stored in a database at the end of the workflow. Arrows indicate dependencies between the different steps of the workflow.

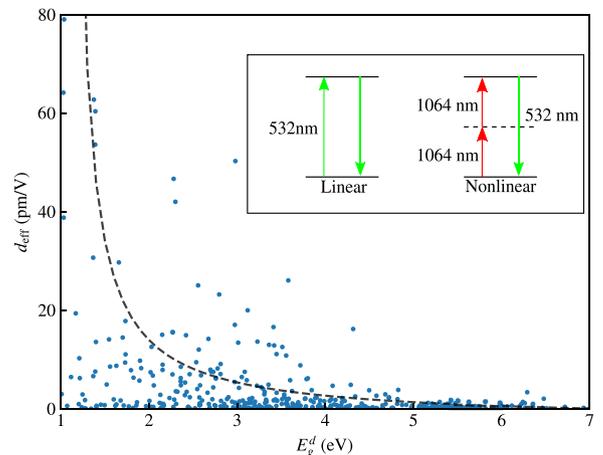


Fig. 11. Calculated effective nonlinear coefficient d_{eff} vs. direct band gap E_g^d for more than 400 materials represented by blue circles. The dashed line is a guide to the eye to indicate the inverse relationship between these two quantities. In linear optics, a material that absorbs an incoming photon frequency ω , emits a photon at the same frequency. In nonlinear optics, on the contrary, a SHG-active material (i.e. a system without inversion symmetry) can absorb two incoming photons at a frequency ω , emitting a single photon at a frequency 2ω provided the incoming light has sufficient intensity.

obviously needed to obtain more physical insights into the SHG mechanisms but this study will be presented in more detail in a future publication [207].

7. Conclusion and perspectives

In its more than twenty years of existence, the ABINIT project has contributed to thousands of publications, by its frontier-of-knowledge implementations of first-principles calculations based on density-functional theory, density-functional perturbation theory, many-body theory, and so forth. Over this period, several hundred developers have contributed to the project. The goal of

the present publication was to analyze its impact, situate it in its immediate environment, and provide an account of the recent and ongoing developments.

In the last few years, different layers within the ABINIT project have appeared: while the main abinit application has continued to grow and acquire interesting new computational capabilities with performance increases, new applications have gained importance, namely multibinit and tdep, both being delivered with the ABINIT package. In addition, a scripting environment, ABIPY, and the associate high-throughput flows collected in ABIFLOWS have been developed, and are now fully operational. They are delivered through the ABINIT organization on Github.

All these evolutions rely on, or are exploited by, a series of closely linked partner projects. Some have been singled out in the present publications: the MATERIALS PROJECT, the Δ -factor project, the LIBXC project and the PSEUDO-DOJO project, among others.

Such evolutions are far from being completed, and the work on ABINIT will continue for the coming decade at least. The axes of development of the ABINIT project for the forthcoming years will include, among others:

- Further development of second-principles calculations, to include the full palette of responses to atomic displacements, strains, local electric field, and local magnetic field, and possibly for more physical situations, including inhomogeneities, surfaces, interfaces, and so forth. In this respect, the implementation of spatial dispersion calculations, including flexoelectricity and dynamical quadrupole, following Refs. [208–211] should be ported inside the ABINIT distribution in the future.
- Corresponding developments of first-principles response calculations, to include more derivatives, and complete the capabilities of existing ones, also beyond the simplest local or semi-local exchange–correlation approximations.
- Adaptation to emerging computer architectures, not only for the mainstream DFT calculations, but also for DFPT or MBPT implementations, possibly relying on libraries.
- Further implementations and developments related to electron–phonon interactions, including development of high-throughput tools.

We look forward to providing further updates on the ABINIT project for the 60th anniversary of CPC.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Over the years, the ABINIT project has benefited from contributions of numerous scientists not present in the list of coauthors. We are grateful to these persons, and we want to acknowledge their contributions. Also, we thank the people who have expressed their appreciation of the ABINIT team work.

This work has been supported by the Fonds de la Recherche Scientifique (F.R.S.-FNRS Belgium) through the PdR Grants No. T.0238.13 - AIXPHO (X.G., M.G. G.-M. R.), HiT4FiT (Ph.G.), No. T.1071.15 - HTBaSE (H.P.C.M.), and No.T.0103.19 - ALPS (X.G., M.J.V.), an “Aspirant” mandate (G.B.), and a “Chargé de recherche” mandate (Y.J.). X.G. acknowledges the hospitality of the CEA DAM-DIF during the year 2017. G.-M.R. also acknowledges support from the Fonds de la Recherche Scientifique (F.R.S.-FNRS Belgium). It has also been supported (Ph.G.) by the EU and FRS-FNRS, Belgium through the M-ERA.NET project SIOX. G.P., M.G., X.G.,

Ph. G. and G.-M. R. acknowledge support from the Communauté française de Belgique, Belgium through the BATTAB project (ARC 14/19-057) and the the AIMED project. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL), by the University of Liège, the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the FRS-FNRS, Belgium under Grant No. 2.5020.11, and the Tier-1 supercomputer of the Fédération Wallonie-Bruxelles funded by the Walloon Region under the Grant No 1117545. S.C. acknowledges financial support from OCAS NV by an OCAS-endowed chair at Ghent University, Belgium. This work moreover benefited from the Research Foundation Flanders (FWO), Belgium through the personal postdoctoral fellowship of K.L. and project Nr. G0E0116N.

This work (G.A.) was supported by the National Science Foundation, United States under grant DMR-1508412 which provided for basic theory and formalism, and by the Center for Computational Study of Excited-State Phenomena in Energy Materials funded by the U. S. Department of Energy, Office of Basic Energy Sciences, United States, under Contract No. DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory which provided for algorithm and code developments and simulations.

This work (M.C., O.G.) has been supported by the Fonds de Recherche du Québec Nature et Technologie (FRQ-NT), Canada, the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants RGPIN-2016-06666. Computational resources have been provided by the Canadian Foundation for Innovation, the Ministère de l'Éducation des Loisirs et du Sport (Québec), Calcul Québec, and Compute Canada.

This work (J.Z.) has been supported by the Canada Research Chairs program, and was enabled in part by support provided by Westgrid (www.westgrid.ca) and Compute Canada (www.computeCanada.ca).

M.A.L.M acknowledges partial support from the German DFG through the project MA6787/6-1.

K.A.P. acknowledges support from the U.S. Department of Energy, Office of Science, United States, Office of Basic Energy Sciences, United States, Materials Sciences and Engineering Division under Contract DE-AC02-05-CH11231: Materials Project program KC23MP.

This work (Y.P.) has been supported by the RETOS Colaboración Funding Program of MINECO, Government of Spain (SIESTA-PRO, ref. RTC-2016-5681-7).

This work (A.R.) has been supported by the DMREF-NSF, Belgium 1434897, National Science Foundation, United States OAC-1740111 and U.S. Department of Energy, United States DE-SC00 16176 projects.

References

- [1] S. Curtarolo, G.L.W. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, *Nature Mater.* 12 (2013) 191.
- [2] In computational materials science, such a list of codes includes VASP, Quantum Espresso, Wien2K, CASTEP, SIESTA, GPAW, CP2K, CPMD, AIMS, Exciting, and others. See <https://www.nomad-coe.eu/externals/codes> for links to these codes. These codes can contribute to open-access databases such as NOMAD https://metainfo.nomad-coe.eu/nomadmetainfo_public/archive.html, the Materials Cloud <https://www.materialscloud.org/>, the Materials Project <https://www.materialsproject.org/>, AFLOW <http://afloplib.org/>, and OQMD <http://oqmd.org/> among others.
- [3] X. Gonze, B. Amadon, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Côté, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D.R. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M.J.T. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G.-M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M.J. Verstraete, G. Zerah, J.W. Zwanziger, *Comput. Phys. Comm.* 180 (2009) 2582–2615.
- [4] P. Hohenberg, W. Kohn, *Phys. Rev.* 136 (1964) B864–B871.

- [5] W. Kohn, L.J. Sham, *Phys. Rev.* 140 (1965) A1133–A1138.
- [6] R.M. Martin, *Electronic Structure, Basic Theory and Practical Methods*, Cambridge University Press, 2004.
- [7] S. Baroni, P. Giannozzi, A. Testa, *Phys. Rev. Lett.* 58 (1987) 1861–1864.
- [8] X. Gonze, J.-P. Vigneron, *Phys. Rev. B* 39 (1989) 13120–13128.
- [9] X. Gonze, P. Ghosez, R.W. Godby, *Phys. Rev. Lett.* 74 (1995) 4035–4038.
- [10] X. Gonze, *Phys. Rev. B* 55 (1997) 10337–10354.
- [11] S. Baroni, S. de Gironcoli, A. Dal Corso, P. Giannozzi, *Rev. Modern Phys.* 73 (2001) 515–562.
- [12] X. Gonze, G.-M. Rignanese, R. Caracas, Z. Kristallogr. 220 (2005) 458–472.
- [13] G.D. Mahan, *Many-particle physics*. Third edition., Springer US, ISBN: 978-1-4419-3339-3, 2000.
- [14] R.M. Martin, L. Reining, D.M. Ceperley, *Interacting Electrons. Theory and Computational Approaches*. Cambridge University Press, 2016.
- [15] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zérah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J.-Y. Raty, D.C. Allan, *Comput. Mater. Sci.* 25 (2002) 478–492.
- [16] X. Gonze, G.-M. Rignanese, M. Verstraete, J.-M. Beuken, Y. Pouillon, R. Caracas, F. Jollet, M. Torrent, G. Zerah, M. Mikami, P. Ghosez, M. Veithen, J.-Y. Raty, V. Olevano, F. Bruneval, L. Reining, R. Godby, G. Onida, D.C. Allan, D. Hamann, Z. Kristallogr. Cryst. Mater. 220 (2005) 558–562.
- [17] X. Gonze, F. Jollet, F. Abreu Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, E. Bousquet, F. Bruneval, D. Caliste, M. Côté, F. Dahm, F. Da Pieve, M. Delaveau, M. Di Gennaro, B. Dorado, C. Espejo, G. Geneste, L. Genovese, A. Grossier, M. Giantomassi, Y. Gillet, D.R. Hamann, L. He, G. Jomard, J. Laflamme Janssen, S. Le Roux, A. Levitt, A. Lherbier, F. Liu, I. Lukaevič, A. Martin, C. Martins, M.J.T. Oliveira, S. Poncé, Y. Pouillon, T. Rangel, G.-M. Rignanese, A.H. Romero, B. Rousseau, O. Rubel, A.A. Shukri, M. Stankovski, M. Torrent, M. Van Setten, B. Van Troeye, M.J. Verstraete, D. Waroquiers, J. Wiktor, B. Xu, A. Zhou, J.W. Zwanziger, *Comput. Phys. Comm.* 205 (2016) 106–131.
- [18] T. Reuters, Available in the internet at: <http://apps.isiknowledge.com>.
- [19] E. Peirson, et al., <https://diging.github.io/tethne/>, 2015.
- [20] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, *Genome Res.* 13 (2003) 2498–2504.
- [21] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, *APL Mater.* 1 (2013) 011002.
- [22] M. de Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. Krishna Ande, S. van der Zwaag, J.J. Plata, C. Toher, S. Curtarolo, G. Ceder, K.A. Persson, *M. Asta, Sci. Data* 2 (2015) 150009.
- [23] K. Mathew, C. Zheng, D. Winston, C. Chen, A. Dozier, J.J. Rehr, S.P. Ong, K.A. Persson, *Sci. Data* (2018) 180151.
- [24] M. de Jong, W. Chen, H. Geerlings, M. Asta, K.A. Persson, *Sci. Data* 2 (2015) 150053.
- [25] I. Petousis, W. Chen, G. Hautier, T. Graf, T.D. Schladt, K.A. Persson, F.B. Prinz, *Phys. Rev. B* 93 (2016) 115151.
- [26] F. Ricci, W. Chen, U. Aydemir, G.J. Snyder, G.-M. Rignanese, A. Jain, G. Hautier, *Sci. Data* 4 (2017) 170085.
- [27] R. Tran, Z. Xu, B. Radhakrishnan, D. Winston, W. Sun, K.A. Persson, S.P. Ong, *Sci. Data* 3 (2016) 160080 EP –.
- [28] G. Petretto, S. Dwaraknath, H.P.C. Miranda, D. Winston, M. Giantomassi, M.J. van Setten, X. Gonze, K.A. Persson, G. Hautier, G.-M. Rignanese, *Sci. Data* 5 (2018) 180065.
- [29] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, *Comput. Mater. Sci.* 68 (2013) 314–319.
- [30] A. Jain, S.P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, K.A. Persson, *Concurr. Comput.: Pract. Exper.* 27 (2015) 5037–5059.
- [31] K. Mathew, J.H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I. heng Chu, T. Smidt, B. Bocklund, M. Horton, J. Dagdelen, B. Wood, Z.-K. Liu, J. Neaton, S.P. Ong, K. Persson, A. Jain, *Comput. Mater. Sci.* 139 (2017) 140–152.
- [32] Wikipedia, https://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid-state_physics_software.
- [33] K. Lejaeghere, S. Cottenier, <https://molmod.ugent.be/deltacodesdft>. (Accessed 29 May 2019).
- [34] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I.E. Castelli, S.J. Clark, A. Dal Corso, S. de Gironcoli, T. Deutsch, J.K. Dewhurst, I. Di Marco, C. Draxl, M. Dułak, O. Eriksson, J.A. Flores-Livas, K.F. Garrity, L. Genovese, P. Giannozzi, M. Giantomassi, S. Goedecker, X. Gonze, O. Grånäs, E.K.U. Gross, A. Gulans, F. Gygi, D.R. Hamann, P.J. Hasnip, N.A.W. Holzwarth, D. Iuşan, D.B. Jochym, F. Jollet, D. Jones, G. Kresse, K. Koepnik, E. Küçükbenli, Y.O. Kvashnin, I.L.M. Locht, S. Lubeck, M. Marsman, N. Marzari, U. Nitzsche, L. Nordström, T. Ozaki, L. Paulatto, C.J. Pickard, W. Poelmans, M.J. Probert, K. Refson, M. Richter, G.-M. Rignanese, S. Saha, M. Scheffler, M. Schlipf, K. Schwarz, S. Sharma, F. Tavazza, P. Thunström, A. Tkatchenko, M. Torrent, D. Vanderbilt, M.J. van Setten, V. Van Speybroeck, J.M. Wills, J.R. Yates, G.-X. Zhang, S. Cottenier, *Science* 351 (2016) aad3000.
- [35] K. Lejaeghere, V. Van Speybroeck, G. Van Oost, S. Cottenier, *Crit. Rev. Solid State Mater. Sci.* 39 (2013) 1–24.
- [36] K. Lejaeghere, in: Y. Wang, D. McDowell (Eds.), *Uncertainty Quantification in Multiscale Materials Modeling*, Elsevier, 2020, in press.
- [37] F. Jollet, M. Torrent, N. Holzwarth, *Comput. Phys. Comm.* 185 (2014) 1246–1254.
- [38] GPAW, <https://wiki.fysik.dtu.dk/gpaw/setups/setups.html>. (Accessed 7 June 2019).
- [39] K.F. Garrity, J.W. Bennett, K.M. Rabe, D. Vanderbilt, *Comput. Mater. Sci.* 81 (2014) 446–452.
- [40] D.R. Hamann, *Phys. Rev. B* 88 (2013) 085117.
- [41] M.J. van Setten, M. Giantomassi, E. Bousquet, M.J. Verstraete, D.R. Hamann, X. Gonze, G.-M. Rignanese, *Comput. Phys. Comm.* 226 (2018) 39–54.
- [42] C. Hartwigsen, S. Goedecker, J. Hutter, *Phys. Rev. B* 58 (1998) 3641–3662.
- [43] M. Krack, *Theor. Chem Acc.* 114 (2005) 145–152.
- [44] M. Fuchs, M. Scheffler, *Comput. Phys. Comm.* 119 (1999) 67–98.
- [45] M.A.L. Marques, M.J.T. Oliveira, T. Burnus, *Comput. Phys. Comm.* 183 (2012) 2272–2281.
- [46] S. Lehtola, C. Steigemann, M.J.T. Oliveira, M.A.L. Marques, *SoftwareX* 7 (2018) 1–5.
- [47] J.P. Perdew, Jacob's Ladder of Density Functional Approximations for the Exchange–Correlation Energy, in: *AIP Conference Proceedings*, vol. 577, AIP, AIP, Belgium, 2001, pp. 1–20.
- [48] L. He, F. Liu, G. Hautier, M.J.T. Oliveira, M.A.L. Marques, F.D. Vila, J.J. Rehr, G.-M. Rignanese, A. Zhou, *Phys. Rev. B* 89 (2014) 064305.
- [49] A. Garcia, M.J. Verstraete, Y. Pouillon, J. Junquera, *Comput. Phys. Comm.* 227 (2018) 51–71.
- [50] N.A.W. Holzwarth, A.R. Tackett, G.E. Matthews, *Comput. Phys. Comm.* 135 (2001) 329–347.
- [51] A.A. Mostofi, J.R. Yates, Y.-S. Lee, I. Souza, D. Vanderbilt, N. Marzari, *Comput. Phys. Comm.* 178 (2008) 685.
- [52] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, *J. Phys.: Condens. Matter* 26 (2014) 213201.
- [53] O. Parcollet, M. Ferrero, T. Ayril, H. Hafermann, I. Krivenko, L. Messio, P. Seth, *Comput. Phys. Comm.* 196 (2015) 398–415.
- [54] <https://gitlab.com/ElectronicStructureLibrary/esl-bundle>.
- [55] <https://www.secondprinciples.unican.es>.
- [56] T. Rangel, D. Caliste, L. Genovese, M. Torrent, *Comput. Phys. Comm.* 208 (2016) 1–8.
- [57] X.-G. Xiong, T. Yanai, *J. Chem. Theory Comput.* 13 (2017) 3236–3249.
- [58] J. Rehr, J. Kas, F. Vila, M. Prange, K. Jorissen, *Phys. Chem. Chem. Phys.* 12 (2010) 5503–5513.
- [59] K. Gilmore, J. Vinson, E. Shirley, D. Prendergast, C. Pemmarajud, J. Kas, F. Vila, J. Rehr, *Comput. Phys. Comm.* 197 (2015) 109.
- [60] F.D. Vila, J.J. Rehr, H.H. Rossner, H.J. Krappe, *Phys. Rev. B* 76 (2007) 014301.
- [61] V. Olevano, M. Palummo, G. Onida, R.D. Sole, *Phys. Rev. B* 60 (1999) 14224–14233.
- [62] S. Albrecht, L. Reining, R. Del Sole, G. Onida, *Phys. Rev. Lett.* 80 (1998) 4510–4513.
- [63] F. Sottile, M. Marsili, V. Olevano, L. Reining, *Phys. Rev. B* 76 (2007) 161103.
- [64] J. Deslippe, G. Samsonidze, D.A. Strubbe, M. Jain, M.L. Cohen, S.G. Louie, *Comput. Phys. Comm.* 183 (2012) 1269–1289.
- [65] D. Sangalli, A. Ferretti, H. Miranda, C. Attaccalite, I. Marri, E. Cannuccia, P. Melo, M. Marsili, F. Paleari, A. Marrazzo, G. Prandini, P. Bonfà, M.O. Atambo, F. Affinito, M. Palummo, A. Molina-Sánchez, C. Hogan, M. Grüning, D. Varsano, A. Marini, *J. Phys.: Condens. Matter* 31 (2019) 325902.
- [66] J. Nocedal, *Math. Comp.* 35 (1980) 773.
- [67] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, P. Gumbusch, *Phys. Rev. Lett.* 97 (2006) 170201.
- [68] E. Hairer, C. Lubich, G. Wanner, *Acta Numer.* 12 (2003) 399–450.
- [69] P. Minyary, G.J. Martyna, M.E. Tuckerman, *J. Chem. Phys.* 118 (2003) 2510.
- [70] S. Duane, A.D. Kennedy, B.J. Pendleton, D. Roweth, *Phys. Lett. B* 195 (1987) 216.
- [71] S. Prokhorenko, K. Kalke, Y. Nahas, L. Bellaiche, *NPJ Comput. Mater.* 4 (2018) 80.
- [72] G.C.M. Sprink, *J. Chem. Phys.* 109 (1998) 7737.
- [73] Y. Komeiji, *Chem-bio Inf. J.* 7 (2007) 12–23.
- [74] Y. Jia, S. Poncé, A. Miglio, M. Mikami, X. Gonze, *Phys. Rev. B* 100 (2019) 155109.
- [75] U. von Barth, *Phys. Rev. A* 20 (1979) 1693.
- [76] L.G. Ferreira, M. Marques, L.K. Teles, *Phys. Rev. B* 78 (2008) 125116.
- [77] M. Landmann, E. Rauls, W.G. Schmidt, M. Röppischer, C. Cobet, N. Esser, T. Schupp, D.J. As, M. Feneberg, R. Goldhahn, *Phys. Rev. B* 87 (2013) 195210.
- [78] D.G. Shankland, *Int. J. Quantum Chem.* 5 (1971) 497–500.
- [79] R.N. Euwema, D.J. Stukel, T.C. Collins, J.S. Dewitt, D.G. Shankland, *Phys. Rev.* 178 (1969) 1419–1423.

- [80] D. Koelling, J. Wood, *J. Comput. Phys.* 67 (1986) 253–262.
- [81] G.K.H. Madsen, D.J. Singh, *Comput. Phys. Comm.* 175 (2006) 67–71.
- [82] G.K.H. Madsen, J. Carrete, M.J. Verstraete, *Comput. Phys. Comm.* 231 (2018) 140–145.
- [83] K. Uehara, J.S. Tse, *Phys. Rev. B* 61 (2000) 1639–1642.
- [84] V.-A. Ha, G. Yu, F. Ricci, D. Dahliah, M.J. van Setten, M. Giantomassi, G.-M. Rignanese, G. Hautier, *Phys. Rev. Mater.* 3 (2019) 034601.
- [85] A.M. Essin, A.M. Turner, J.E. Moore, D. Vanderbilt, *Phys. Rev. B* 81 (2010) 205104.
- [86] X. Gonze, J.W. Zwanziger, *Phys. Rev. B* 84 (2011) 064445.
- [87] T. Thonhauser, D. Ceresoli, A.A. Mostofi, N. Marzari, R. Resta, D. Vanderbilt, *J. Chem. Phys.* 131 (2009) 101101.
- [88] L. Lin, *J. Chem. Theory Comput.* 12 (2016) 2242.
- [89] P. Pulay, *Chem. Phys. Lett.* 73 (1980) 393.
- [90] P. Pulay, *J. Comput. Chem.* 3 (1982) 556.
- [91] J. Paier, M. Marsman, K. Hummer, G. Kresse, I.C. Gerber, J.G. Ángyán, *J. Chem. Phys.* 124 (2006) 154709.
- [92] M.J. Verstraete, M. Torrent, F. Jollet, G. Zérah, X. Gonze, *Phys. Rev. B* 78 (2008) 045119.
- [93] F. Ricci, S. Prokhorenko, M. Torrent, M.J. Verstraete, E. Bousquet, *Phys. Rev. B* 99 (2019) 184404.
- [94] F.G. Eich, E.K.U. Gross, *Phys. Rev. Lett.* 111 (2013) 156401.
- [95] F.G. Eich, S. Pittalis, G. Vignale, *Phys. Rev. B* 88 (2013) 245102.
- [96] I.W. Bulik, G. Scalmani, M.J. Frisch, G.E. Scuseria, *Phys. Rev. B* 87 (2013) 035117.
- [97] S.Y. Savrasov, *Phys. Rev. Lett.* 81 (1998) 2570–2573.
- [98] K. Cao, H. Lambert, P.G. Radaelli, F. Giustino, *Phys. Rev. B* 97 (2018) 024420.
- [99] S. Grimme, *J. Comput. Chem.* 27 (2006) 1787–1799.
- [100] S. Grimme, J. Antony, S. Ehrlich, H. Krieg, *J. Chem. Phys.* 132 (2010) 154104.
- [101] S. Grimme, S. Ehrlich, L. Goerigk, *J. Comput. Chem.* 32 (2011) 1456–1465.
- [102] B. Van Troeye, M. Torrent, X. Gonze, *Phys. Rev. B* 93 (2016) 144304.
- [103] B. Van Troeye, M.J. van Setten, M. Giantomassi, M. Torrent, G.-M. Rignanese, X. Gonze, *Phys. Rev. B* 95 (2017) 024112.
- [104] N.A. Pike, A. Dewandre, B. Van Troeye, X. Gonze, M.J. Verstraete, *Phys. Rev. Mater.* 2 (2018) 063608.
- [105] P. Brüesch, *Phonons: Theory and Experiments II*, Springer, 1986.
- [106] M. Cardona, G. Güntherodt, *Light Scattering in Solids II*, Springer, 1982.
- [107] M. Veithen, X. Gonze, P. Ghosez, *Phys. Rev. B* 71 (2005) 125107.
- [108] K. Miwa, *Phys. Rev. B* 84 (2011) 094304.
- [109] B. Lafuente, R. Downs, H. Yang, N. Stone, *Highlights in Mineralogical Crystallography*, Walter de Gruyter GmbH, ISBN: 9783110417043, 2016, pp. 1–29, <http://rruff.info/>. (sample id. : R040031).
- [110] A. Eiguren, C. Ambrosch-Draxl, *Phys. Rev. B* 78 (2008) 045124.
- [111] C. Verdi, F. Giustino, *Phys. Rev. Lett.* 115 (2015) 176401.
- [112] F. Giustino, *Rev. Modern Phys.* 89 (2017) 015003.
- [113] J. Laflamme Janssen, Y. Gillet, S. Poncé, A. Martin, M. Torrent, X. Gonze, *Phys. Rev. B* 93 (2016) 205147.
- [114] S. Poncé, E. Margine, F. Giustino, *Phys. Rev. B* (2018) 121201(R).
- [115] S. Mazevet, M. Torrent, V. Recoules, F. Jollet, *High Energy Density Phys.* 6 (2010) 84–88.
- [116] J. Banhart, *Phil. Mag.* B 77 (1998) 85–103.
- [117] M. Marsman, G. Kresse, *J. Chem. Phys.* 125 (2006) 104101.
- [118] J. Sun, M. Marsman, G.I. Csonka, A. Ruzsinszky, P. Hao, Y.-S. Kim, G. Kresse, J.P. Perdew, *Phys. Rev. B* 84 (2011).
- [119] M. Torrent, F. Jollet, F. Bottin, G. Zérah, X. Gonze, *Comput. Mater. Sci.* 42 (2008) 337–351.
- [120] The status of the implementation of GW in ABINIT has been described in Refs. [3,16,17]. In line with the target of the present paper mentioned in the introduction, namely, to describe the improvements that have occurred in the 2016–2019 period, we do not dedicate a separate section to GW. The improvements are mentioned in Sections 3.1.5 and 3.5.
- [121] S. Sharma, J.K. Dewhurst, A. Sanna, E.K.U. Gross, *Phys. Rev. Lett.* 107 (2011) 186401.
- [122] W. Chen, A. Pasquarello, *Phys. Rev. B* 92 (2015) 041115.
- [123] W. Chen, F. Ambrosio, G. Miceli, A. Pasquarello, *Phys. Rev. Lett.* 117 (2016) 186401.
- [124] S. Rigamonti, S. Botti, V. Veniard, C. Draxl, L. Reining, F. Sottile, *Phys. Rev. Lett.* 114 (2015) 146402.
- [125] J.A. Berger, *Phys. Rev. Lett.* 115 (2015) 137402.
- [126] T. Rangel, M.D. Ben, D. Varsano, G. Antonius, F. Bruneval, F.H. da Jornada, M.J. van Setten, O.K. Orhan, D.D. O'Regan, A. Canning, A. Ferretti, A. Marini, G.-M. Rignanese, J. Deslippe, S.G. Louie, J.B. Neaton, 2019, arXiv: 1903.06865.
- [127] A. Marini, C. Hogan, M. Grüning, D. Varsano, *Comput. Phys. Comm.* 180 (2009) 1392–1403.
- [128] P. Carrier, S. Rohra, A. Görling, *Phys. Rev. B* 75 (2007) 205126.
- [129] F. Gygi, A. Baldereschi, *Phys. Rev. B* 34 (1986) 4405–4408.
- [130] J. Spencer, A. Alavi, *Phys. Rev. B* 77 (2008) 193110.
- [131] Y. Gillet, M. Giantomassi, X. Gonze, *Comput. Phys. Comm.* 203 (2016) 83–93.
- [132] Y. Gillet, (Ph.D. thesis), Université catholique de Louvain, Louvain-la-neuve, Belgium, 2017.
- [133] A. Marini, *Phys. Rev. Lett.* 101 (2008) 106405.
- [134] A. Georges, G. Kotliar, W. Krauth, M.J. Rozenberg, *Rev. Modern Phys.* 68 (1996) 13–125.
- [135] P. Werner, A. Comanac, L. de Medici, M. Troyer, A.J. Millis, *Phys. Rev. Lett.* 97 (2006) 076405.
- [136] P. Werner, A.J. Millis, *Phys. Rev. B* 74 (2006) 155107.
- [137] E. Gull, A.J. Millis, A.I. Lichtenstein, A.N. Rubtsov, M. Troyer, P. Werner, *Rev. Modern Phys.* 83 (2011) 349–404.
- [138] G. Kotliar, S.Y. Savrasov, K. Haule, V.S. Oudovenko, O. Parcollet, C.A. Marianetti, *Rev. Modern Phys.* 78 (2006) 865–951.
- [139] H. Park, A.J. Millis, C.A. Marianetti, *Phys. Rev. B* 92 (2015) 035146.
- [140] H. Chen, A.J. Millis, *Phys. Rev. B* 93 (2016) 045133.
- [141] B. Amadon, B. Dorado, *J. Phys.: Condens. Matter* 30 (2018) 405603.
- [142] J. Bieder, B. Amadon, *Phys. Rev. B* 89 (2014) 195132.
- [143] B. Amadon, A. Gerossier, *Phys. Rev. B* 91 (2015) 161103.
- [144] B. Amadon, *Phys. Rev. B* 94 (2016) 115148.
- [145] P. Seth, I. Krivenko, M. Ferrero, O. Parcollet, *Comput. Phys. Comm.* 200 (2016) 274–284.
- [146] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, O. Parcollet, *Phys. Rev. B* 84 (2011) 075145.
- [147] F. Bottin, S. Leroux, A. Knyazev, G. Zérah, *Comput. Mater. Sci.* 42 (2008) 329–336.
- [148] J. Behler, M. Parrinello, *Phys. Rev. Lett.* 98 (2007) 146401.
- [149] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, *Phys. Rev. Lett.* 104 (2010) 136403.
- [150] J.C. Wojdeł, P. Hermet, M.P. Ljungberg, P. Ghosez, J. Íñiguez, *J. Phys.: Condens. Matter* 25 (2013) 305401.
- [151] P. García-Fernández, J.C. Wojdeł, J. Íñiguez, J. Junquera, *Phys. Rev. B* 93 (2016) 195137.
- [152] A. Rohskopf, H.R. Seyf, K. Gordiz, T. Tadano, A. Henry, *NPJ Comput. Mater.* 3 (2017) 27.
- [153] D. Dragoni, T.D. Daff, G. Csányi, N. Marzari, *Phys. Rev. Mater.* 2 (2018) 013808.
- [154] A. Grisafi, D.M. Wilkins, G. Csányi, M. Ceriotti, *Phys. Rev. Lett.* 120 (2018) 036002.
- [155] F. Maresca, D. Dragoni, G. Csányi, N. Marzari, W.A. Curtin, *NPJ Comput. Mater.* 4 (2018) 69.
- [156] X. Gonze, C. Lee, *Phys. Rev. B* 55 (1997) 10355–10368.
- [157] P. Ghosez, J.-P. Michenaud, X. Gonze, *Phys. Rev. B* 58 (1998) 6224–6240.
- [158] C. Escorihuela-Sayalero, J.C. Wojdeł, J. Íñiguez, *Phys. Rev. B* 95 (2017) 094115.
- [159] W. Zhong, D. Vanderbilt, K.M. Rabe, *Phys. Rev. B* 52 (1995) 6301–6312.
- [160] K.M. Rabe, U.V. Waghmare, *Phys. Rev. B* 52 (1995) 13236–13246.
- [161] <https://github.com/abinit/abiot>.
- [162] <https://github.com/abinit/qAgate>.
- [163] I.E. Dzialoshinskii, *Sov. Phys. JETP* 5 (1957) 1259.
- [164] T. Moriya, *Phys. Rev.* 120 (1960) 91.
- [165] B. Skubic, J. Hellsvik, L. Nordström, O. Eriksson, *J. Phys.: Condens. Matter* 20 (2008) 315203.
- [166] R.F.L. Evans, W.J. Fan, P. Chureemart, T.A. Ostler, M.O.A. Ellis, R.W. Chantrell, *J. Phys.: Condens. Matter* 26 (2014) 103202.
- [167] M.J. Donahue, D.G. Porter, *Tech. Rep.*, National Institute of Standards and Technology, Gaithersburg, MD, 1999.
- [168] A.I. Liechtenstein, M.I. Katsnelson, V.P. Antropov, V.A. Gubanov, *J. Magn. Magn. Mater.* 67 (1987) 65–74.
- [169] N. Marzari, D. Vanderbilt, *Phys. Rev. B* 56 (1997) 12847–12865.
- [170] T.L. Gilbert, *IEEE Trans. Magn.* 40 (2004) 3443–3449.
- [171] G.-M. Rignanese, J.-P. Michenaud, X. Gonze, *Phys. Rev. B* 53 (1996) 4488–4497.
- [172] K. Esfarjani, H.T. Stokes, *Phys. Rev. B* 77 (2008) 144112.
- [173] O. Hellman, I.A. Abrikosov, S.I. Simak, *Phys. Rev. B* 84 (2011) 180301(R).
- [174] O. Hellman, P. Steneteg, I.A. Abrikosov, S.I. Simak, *Phys. Rev. B* 87 (2013) 104111.
- [175] O. Hellman, I.A. Abrikosov, *Phys. Rev. B* 88 (2013) 144301.
- [176] J. Bouchet, F. Bottin, *Phys. Rev. B* 92 (2015) 174108.
- [177] A. Dewaele, V. Stutzmann, J. Bouchet, F. Bottin, F. Occelli, M. Mezouar, *Phys. Rev. B* 91 (2015) 134108.
- [178] J. Bouchet, F. Bottin, *Phys. Rev. B* 95 (2017) 054113.
- [179] B. Dorado, F. Bottin, J. Bouchet, *Phys. Rev. B* 95 (2017) 104303.
- [180] J. Bouchet, F. Bottin, V. Recoules, F. Remus, G. Morard, R.M. Bolis, A. Benuzzi-Mounaix, *Phys. Rev. B* 99 (2019) 094113.
- [181] C. Lee, X. Gonze, *Phys. Rev. B* 51 (1995) 8610–8613.
- [182] G. Leibfried, W. Ludwig, in: F. Seitz, D. Turnbull (Eds.), *Solid State Physics Vol 12 : Theory of Anharmonic Effects in Crystals*, Academic Press, New York, 1961, pp. 275–444.

- [183] D.C. Wallace, *Statistical physics of crystals and liquids*, World Scientific Publishing Company, Singapore, 2003.
- [184] I. Schnell, M.D. Jones, S.P. Rudin, R.C. Albers, *Phys. Rev. B* 74 (2006) 054104.
- [185] E. Mozafari, N. Shulumba, P. Steneteg, B. Alling, I.A. Abrikosov, *Phys. Rev. B* 94 (2016) 054111.
- [186] J. Shiomi, K. Esfarjani, G. Chen, *Phys. Rev. B* 84 (2011) 104302.
- [187] C.H. Lee, C.K. Gan, *Phys. Rev. B* 96 (2017) 035105.
- [188] F. Bottin, J. Bouchet, J. Bieder, (2019) unpublished.
- [189] F. Giustino, M.L. Cohen, S.G. Louie, *Phys. Rev. B* 76 (2007) 165108.
- [190] S. Poncé, G. Antonius, Y. Gillet, P. Boulanger, J. Laflamme Janssen, A. Marini, M. Côté, X. Gonze, *Phys. Rev. B* 90 (2014) 214304.
- [191] G. Antonius, S. Poncé, E. Lantagne-Hurtubise, G. Auclair, X. Gonze, M. Côté, *Phys. Rev. B* 92 (2015) 085137.
- [192] S. Poncé, Y. Gillet, J. Laflamme Janssen, A. Marini, M. Verstraete, X. Gonze, *J. Chem. Phys.* 143 (2015) 102813.
- [193] J.P. Nery, P.B. Allen, G. Antonius, L. Reining, A. Miglio, X. Gonze, *Phys. Rev. B* 97 (2018) 115145.
- [194] X. Gonze, P. Boulanger, M. Côté, *Ann. Phys.* 523 (2010) 168–178.
- [195] W. Peng, G. Petretto, G.-M. Rignanese, G. Hautier, A. Zevalkink, *Joule* 2 (2018) 1879–1893.
- [196] M. Geimer, K. Hoste, R. McLay, in: 2014 First International Workshop on HPC User Support Tools, 2014, pp. 41–51.
- [197] T. Gamblin, M. LeGendre, M.R. Collette, G.L. Lee, A. Moody, B.R. de Supinski, S. Futral, SC15: International Conference for High-Performance Computing, Networking, Storage and Analysis, IEEE Computer Society, 2015, pp. 1–12.
- [198] G. Fiscaro, L. Genovese, O. Andreussi, N. Marzari, S. Goedecker, *J. Chem. Phys.* 144 (2016) 014103.
- [199] M.J. van Setten, M. Giantomassi, X. Gonze, G.-M. Rignanese, G. Hautier, *Phys. Rev. B* 96 (2017) 155207.
- [200] Y. Gillet, M. Giantomassi, X. Gonze, *Phys. Rev. B* 88 (2013) 094305.
- [201] Y. Gillet, S. Kontur, M. Giantomassi, C. Draxl, X. Gonze, *Sci. Rep.* 7 (2017) 7344.
- [202] D. Caliste, Y. Pouillon, M.J. Verstraete, V. Olevano, X. Gonze, *Comput. Phys. Comm.* 179 (2008) 748–758.
- [203] G. Petretto, X. Gonze, G. Hautier, G.-M. Rignanese, *Comput. Mater. Sci.* 144 (2018) 331–337.
- [204] H. P. C. Miranda, L. Wirtz, (2019) unpublished.
- [205] S.K. Kurtz, T.T. Perry, *J. Appl. Phys.* 39 (1968) 3798–3813.
- [206] J.M. Rondinelli, E. Kioupakis, *Annu. Rev. Mater. Res.* 45 (2015) 491–518.
- [207] F. Naccarato, et al., (2019) unpublished.
- [208] M. Stengel, *Nature Commun.* 4 (2013) 2693.
- [209] A. Schiaffino, C.E. Dreyer, D. Vanderbilt, M. Stengel, *Phys. Rev. B* 99 (2019) 085107.
- [210] C.E. Dreyer, M. Stengel, D. Vanderbilt, *Phys. Rev. B* 98 (2018) 075153.
- [211] M. Royo, M. Stengel, *Phys. Rev. X* 9 (2019) 021050.