Recent developments in the ABINIT software package

X. Gonze [a,b], F. Jollet [c,*], F. Abreu Araujo [a,1], D. Adams [c,2], B. Amadon [c], T. Applencourt [c,3], C. Audouze [c,4], J.-M. Beuken [a,b], J. Bieder [c], A. Bokhanchuk [k,l], E. Bousquet [e], F. Bruneval [f], D. Caliste [g], M. Côté [h], F. Dahm [c], F. Da Pieve [a,b], M. Delaveau [c], M. Di Gennaro [e,b], B. Dorado [c], C. Espejo [w], G. Geneste [c], L. Genovese [g], A. Gerossier [c], M. Giantomassi [a,b], Y. Gillet [a,b], D.R. Hamann [p,q], L. He [m,n], G. Jomard [r], J. Laflamme Janssen [a,b], S. Le Roux [a,b], A. Levitt [c,5], A. Lherbier [a,b], F. Liu [o], I. Lukačević [u], A. Martin [c], C. Martins [c,6], M.J.T. Oliveira [j,b], S. Poncé [a,b,7], Y. Pouillon [i], T. Rangel [c,8], G.-M. Rignanese [a,b], A.H. Romero [v], B. Rousseau [h,9], O. Rubel [t], A.A. Shukri [f], M. Stankovski [s], M. Torrent [c], M.J. Van Setten [a,b], B. Van Troeye [a,b], M.J. Verstraete [e,b], D. Waroquiers [a,b], J. Wiktor [r], B. Xu [e,b,10], A. Zhou [m], J.W. Zwanziger [d]

[a] Université Catholique de Louvain, Louvain-la-Neuve, Belgium
[b] European Theoretical Spectroscopy Facility (ETSF), Belgium
[c] CEA DAM-DIF, F-91297 Arpajon, France
[d] Department of Chemistry and Institute for Research in Materials, Dalhousie University, Halifax, Canada
[e] Q-Mat, Department of Physics, University of Liège, Belgium
[f] CEA, DEN, Service de Recherches de Métallurgie Physique, F-91191 Gif-sur Yvette, France
[g] Laboratoire de Simulation Atomistique (L_Sim) SP2M, UMR-E CEA/UJF-Grenoble 1, INAC, Grenoble, F-38054, France
[h] Physics Department, University of Montreal, Montreal, Canada
[i] Euskal Herriko Unibertsitatea & Materials Evolution, Donostia-San Sebastian, Spain
[j] CFisUC, Department of Physics, University of Coimbra, 3004-516 Coimbra, Portugal
[k] Thunder Bay Regional Research Institute, 980 Oliver Road, Thunder Bay, Ontario, Canada
[l] Confederation College, 1450 Nakina Dr., Thunder Bay, Ontario, Canada
[m] LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[n] Supercomputing Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China
[o] School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 100081, China
[p] Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854-8019, United States
[q] Mat-Sim Research LLC, P.O. Box 742, Murray Hill, NJ 07974, United States
[r] CEA, DEN, DEC, Centre de Cadarache, 13108, Saint-Paul-lez-Durance, France
[s] Lund University, LU OPEN, Box 117, SE-221 00 Lund, Sweden
[t] Department of Materials Science and Engineering, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada
[u] Department of Physics, University J. J. Strossmayer, Osijek, Croatia
[v] Physics Department, West Virginia University, Morgantown, USA

* Corresponding author.
    E-mail address: francois.jollet@cea.fr (F. Jollet).
1 Unité Mixte de Physique, CNRS, Thales, University Paris-Sud, Université Paris-Saclay, 91767, Palaiseau, France.
2 nanotech@surfaces Laboratory, Empa, Swiss Federal Laboratories for Materials Science and Technology, Dübendorf, Switzerland.
3 Laboratoire de Chimie et Physique Quantiques, UMR 5626, IRSAMC, CNRS et Université de Toulouse (UPS), 118 route de Narbonne, 31062 Toulouse, France.
4 University of Toronto Institute for Aerospace Studies (UTIAS), 4925 Dufferin Street, Toronto, Ontario M3H 5T6 Canada.
5 Université Paris Est, CERMICS (ENPC), INRIA, F-77455 Marne-la-Vallée, France.
6 Laboratoire de Chimie et Physique Quantiques, UMR 5626, IRSAMC, CNRS et Université de Toulouse (UPS), 118 route de Narbonne, 31062 Toulouse, France.
7 Department of Materials, University of Oxford, Parks Road, OX1 3PH, United Kingdom.
8 Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.
9 Via Science, Montreal, H3J 1R4, Canada.
10 Physics Department and Institute for Nanoscience and Engineering, University of Arkansas, Fayetteville, AR 72701, USA.

^w *Departamento de Ciencias Básicas, Universidad Jorge Tadeo Lozano, Bogotá, Colombia*

A B S T R A C T

ABINIT is a package whose main program allows one to find the total energy, charge density, electronic structure and many other properties of systems made of electrons and nuclei, (molecules and periodic solids) within Density Functional Theory (DFT), Many-Body Perturbation Theory (GW approximation and Bethe–Salpeter equation) and Dynamical Mean Field Theory (DMFT). ABINIT also allows to optimize the geometry according to the DFT forces and stresses, to perform molecular dynamics simulations using these forces, and to generate dynamical matrices, Born effective charges and dielectric tensors. The present paper aims to describe the new capabilities of ABINIT that have been developed since 2009. It covers both physical and technical developments inside the ABINIT code, as well as developments provided within the ABINIT package. The developments are described with relevant references, input variables, tests and tutorials.

Program summary

Program title: ABINIT

Catalogue identifier: AEEU_v2_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEEU_v2_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU General Public License, version 3

No. of lines in distributed program, including test data, etc.: 4845789

No. of bytes in distributed program, including test data, etc.: 71340403

Distribution format: tar.gz

Programming language: Fortran2003, PERL scripts, Python scripts.

Classification: 7.3, 7.8.

External routines: (all optional) BigDFT [2], ETSF_IO [3], libxc [4], NetCDF [5], MPI [6], Wannier90 [7], FFTW [8].

Catalogue identifier of previous version: AEEU_v1_0

Journal reference of previous version: Comput. Phys. Comm. 180 (2009) 2582

Does the new version supersede the previous version?: Yes. The abinit-7.10.5 version is now the up to date stable version of ABINIT

Nature of problem:
This package has the purpose of computing accurately material and nanostructure properties: electronic structure, bond lengths, bond angles, primitive cell size, cohesive energy, dielectric properties, vibrational properties, elastic properties, optical properties, magnetic properties, non-linear couplings, electronic and vibrational life-times, and others.

Solution method:
Software application based on Density Functional Theory, Many-Body Perturbation Theory and Dynamical Mean Field Theory, pseudopotentials, with plane waves or wavelets as basis functions.

Reasons for new version:
Since 2009, the abinit-5.7.4 version of the code has considerably evolved and is not yet up to date. The abinit- 7.10.5 version contains new physical and technical features that allow electronic structure calculations impossible to carry out in the previous versions.

Summary of revisions:

- new physical features: quantum effects for the nuclei treated by the Path-integral Molecular Dynamics; finding transition states using image dynamics (NEB or string methods); two component DFT for electron-positron annihilation; linear response in a Projector Augmented-Wave approach - PAW-, electron-phonon interactions and temperature dependence of the gap; Bethe Salpeter Equation -BSE- ; Dynamical Mean Field Theory (DMFT).
- new technical features: development of a PAW approach for a wavelet basis; parallelisation of the code on more than 10,000 processors; new build system.
- new features in the ABINIT package: tests; test farm; new tutorials; new pseudopotentials and PAW atomic data tables; GUI and postprocessing tools like the AbiPy and APPA libraries.

Running time:
It is difficult to answer to the question as the use of ABINIT is very large. On one hand, ABINIT can run on 10,000 processors for hours to perform quantum molecular dynamics on large systems. On the other hand, tutorials for students can be performed on a laptop within a few minutes.

References:

1 http://www.gnu.org/copyleft/gpl.txt
2 http://bigdft.org
3 http://www.etsf.eu/fileformats

4 http://www.tddft.org/programs/octopus/wiki/index.php/Libxc
5 http://www.unidata.ucar.edu/software/netcdf
6 https://en.wikipedia.org/wiki/Message_Passing_Interface
7 http://www.wannier.org
8 M. Frigo and S.G. Johnson, Proceedings of the IEEE, 93, 216–231 (2005).

1. Introduction

The ABINIT software application allows one to compute many physical properties of systems made of electrons and nuclei (molecules and periodic solids) thanks to a "first-principles" approach, i.e. without adjustable parameters. The ground state properties are calculated in the frame of the Density-Functional Theory (DFT) as proposed by Hohenberg and Kohn [1] and Kohn and Sham [2], using many different exchange–correlation (XC) functionals, whereas the GW approximation (GW) proposed by Hedin [3] and Bethe–Salpeter equation [4] can be used for excited states. Generally speaking, for solids and systems structured at the nanoscale, even weakly bonded, these give access to cohesive properties, geometry predictions, vibrational, magnetic, elastic, thermodynamical, thermoelectric and dielectric properties, electronic structure, optical properties, spectroscopic responses, and several non-linear properties of solids.

The ABINIT project started in 1997 and the first publicly available version of ABINIT was released in December 2000 under the GNU GPL [5]. It has already been described in 2002 [6], 2005 [7] and 2009 [8]. The last stable version of the package (7.10.5) occupies about 70 MBytes, and gathers nearly 1400 files written in F90 (830 000 lines), including documentation, tutorials and more than one thousand tests. The code is developed by an open community (around fifty people) and it is used by more than a thousand researchers worldwide. A Web site [9] provides the latest version of the package, the on-line documentation and many entry points for the tutorial [10]. A forum is available for questions with many different topics and all levels of complexity [11] and a wiki has been recently developed [12] to house documentation and FAQ extracted from the forum.

The capabilities of ABINIT have been described in detail in the last general paper on ABINIT published in 2009 [8]. The aim of this new paper is to describe the new capabilities than have been developed since that time:

- Section 2 is devoted to a brief summary of the evolution of ABINIT concerning the structure of the package, and how to run the code.
- In Section 3, we shall detail the new physical features developed in the code these last 6 years. We focus on major developments and only briefly mention the other ones. Major developments include quantum effects for the nuclei treated by the Path-integral Molecular Dynamics, finding transition states using image dynamics (NEB or string methods), some developments made in the ground state calculations (finite electric fields and two component DFT for electron–positron annihilation), developments made in linear and non-linear responses (linear response in a Projector Augmented-Wave approach -PAW-, electron–phonon interactions and temperature dependence of the gap), developments made in excited state calculations (Bethe–Salpeter Equation -BSE-) and the Dynamical Mean Field Theory (DMFT).
- In Section 4, recently developed technical features will be described, particularly the advances concerning the development

of a PAW approach for a wavelet basis, the parallelization of the code and the build system.
- Other developments concerning the ABINIT package will be presented in Section 5, especially those which concern the tests, the test farm, the tutorials, pseudopotentials and PAW atomic data, the GUI and postprocessing tools like the AbiPy and APPA libraries or the supercell unfolding.
- Finally, on-going developments, which will be included in the next version of the package, will be briefly presented in Section 6. These include an interpolation technique for the BSE calculations, the availability of hybrid DFT and Van der Waals functionals, a new G_0W_0 implementation, the DFPT computation of effective masses and improvements related to RPA dielectric susceptibility calculations (electro-optic effect and second harmonic generation with anti-resonant contributions and/or scissor correction).

In each section, a brief presentation of the underlying theory of the new feature will be given, followed by the input variables for activation and references, tests, and tutorials if any. Tests are specified by the subdirectory of `tests/` in which they reside and the number of the test, *e.g.*, v7#80.

2. Overview of ABINIT

2.1. Structure of the main ABINIT program, and other main programs

From the user's perspective, the main ABINIT program, `abinit` the top-level routine of which is `src/98_main/abinit.F90`, is a "driver" attacking different "processing units". This structure is unchanged with respect to ABINITv5.8, and is properly described in Ref. [8]. One important processing unit, the "Bethe–Salpeter equation" that allows accurate computation of optical properties and resonant Raman intensities, has been added, and will be described in Section 3.5.1. The other ones (ground state calculations, including geometry optimization; linear responses; non-linear responses; GW computations) are operational, with added capabilities with respect to ABINITv5.8.

The sources of the different F90 programs of the ABINIT package are present in the `src/` directory. Compared to ABINITv5.8, the `src/` directory has been restructured. Most of the subdirectories names now start with a two-digit radix, from 01 to 98, followed by an underscore. Source files in one such subdirectory can only rely on source files in the same directory or from a directory with a lower two-digit characteristics. In particular, the top-level routines, for independent executables (not only `abinit`), are contained in `src/98_main`.

The set of major executables in `src/98_main` is also unchanged with respect to ABINITv5.8 (`abinit`, `anaddb`, `cut3d`, `conducti`, `optics`, `ujdet`), although some post processors, or conversion tools, that had become obsolete due to other developments, or lack of maintenance, have been removed (`abinetcdf`, `compare_interpol`, `lwf`). Also, utilities have been added:

`src/98_main/bsepostproc.F90` for the post-processing of BSE results,

`src/98_main/kss2wfk.F90` for the conversion of _KSS files to _WFK files. Some new profiling tools are present as well, `src/98_main/fftprof.F90`,

`src/98_main/ioprof.F90`, and `src/98_main/lapackprof.F90`. A Van der Waals "DF" type of computation can be prepared by generating the kernel with

`src/98_main/vdwkernelgen.F90`, see Section 6.4.

2.2. Structure of the package

During the last seven years, there has been a steady move towards a more modular structure of the ABINIT package. The motivation and main ideas have been explained in Ref. [8], with noticeable changes for specific directories. Although the `src/` directory still contains the source code (with however its own build system), the so-called "plug-ins" and "prerequisites" have been superseded by "fallbacks", `fallbacks/`, with the change of philosophy that the mandatory components (BLAS, LAPACK) should not be shipped by default with ABINIT package, but the user should use the best compiled libraries available on his/her computer for the execution of the computations. Among the optional external libraries of ABINIT, only FoX has been replaced by PSML, while for the others (BigDFT, NetCDF, ETSF_IO, LibXC, Wannier90), rather recent versions are interfaced with ABINIT. The `tests/` directory has been split in two directories, the tests of capabilities of ABINIT being kept in `tests/`, while the tests of fulfillment of coding rules, memory leaks, documentation of variables, *etc*, (see Section 5.1) have been moved to the directory `special/`. The directories `utils/` and `extras/`, have been suppressed, and their content (with additional files) has been restructured in the two new directories `developers/` and `scripts/`. The directory `developers/` contains tools, scripts and miscellaneous information for developers. Scripts that might be useful for the users are contained in `scripts/`. Finally, the additional directories `coverage/`, `packages/` and `watch/` are useful for developers only.

In recent years, a separate Python package has been developed for the management and post processing of the output of ABINIT, called AbiPy. Although not part of the ABINIT package, it is intimately linked to ABINIT, and it is described in this document as well. It is especially useful in the context of high-throughput calculations. Also, AbinitGUI, a GUI for ABINIT, written in Java, allows one to launch and manage ABINIT runs, and has numerous capabilities to help beginners set up ABINIT runs and analyze their output. This project is described in Section 5.4.3. In view of their size and post- or pre-processing characteristics with respect to the ABINIT package, it has been chosen not to make them part of the main ABINIT package (as it is itself already quite large).

2.3. How to build and run ABINIT

The steps hereafter apply to the build and run of ABINIT under Linux, in the simple case where the mandatory libraries (BLAS, LAPACK) are installed already. ABINIT will also try to determine whether other (optional) libraries are available. If these simple instructions fail, or if the user wants more control on the build of ABINIT, one should refer to the installation notes and help files, that are available on the ABINIT Web site, or in the `doc` directory of the package. Also, compilation/build help can be found on the ABINIT forum http://forum.abinit.org/.

First step: download, gunzip and untar the ABINIT package from the ABINIT Web site (or from the CPC site), then change the current directory to the top of the directory that was created.

```
gunzip abinit-7.10.5.tar.gz
tar -xvf abinit-7.10.5.tar
cd abinit-7.10.5
```

Second step: configure, make and (optionally) make install ABINIT, as follows.

```
configure or ./configure
(or first create a tmp directory,
then cd tmp,
then ../configure)
make (or make mj4 , or make multi multi_nprocs=n
for using ''n''
      processors on a SMP machine where the user
      has to replace ''n'' by its value)
make install  # optional
```

The last (optional) command will install `abinit` in the `/usr/local` directory, otherwise, it is available in `src/98_main`

Third step: run ABINIT, as follows, presuming the `abinit` executable is in the path.

```
abinit < Filenames
(or optionally   abinit < Filenames > log)
```

where the content of the Filenames file is explained on the ABINIT Web site, or in the `doc/users/abinit_help.html` file of the package.

New production versions of ABINIT will be made available as time goes on. We do not expect these build and run steps to be modified within the next few years. Of course, for versions of ABINIT other than v7.10.5, the user should change the package name `abinit-7.10.5.tar.gz` to the one of the new released package, like `abinit-8.1.1.tar.gz`, and modify the above-mentioned instructions accordingly.

Ref. [8] contains more detailed explanations of the build of ABINITv5.8, most of which still apply. Replace "plugins" by "fallbacks", and also note that in v7.10.5, there is only a single main executable, named `abinit`, instead of the separate executable for the sequential and parallel cases, that were named `abinis` and `abinip` in ABINITv5.8. The latter change is actually the only command line change to run ABINIT, as described in Sec. 2.5 of Ref. [8].

3. Recently developed physical features in ABINIT

3.1. Path-integral molecular dynamics

Path-Integral Molecular Dynamics (PIMD) is a technique allowing to simulate the quantum fluctuations of the nuclei at thermodynamic equilibrium [13]. It is implemented in ABINIT in the NVT ensemble since v7.8.2.

In the Path-Integral formalism of quantum statistical mechanics, the (quantum) nuclei are replaced by a set of images (beads) treated by means of classical mechanics, and interacting with each other through a specific effective potential. In the limit of an infinite number of beads, the quantum system and this many-beads classical system have the same partition function, and thus the same static observables. In PIMD, the classical system of beads is simulated by standard Molecular Dynamics. The PIMD equations of motion are integrated by using the Verlet algorithm. At each time step, a ground state DFT calculation is performed for each image. PIMD can be used with any XC functional and works in the PAW framework as well as in the norm-conserving pseudopotential (NCPP) case.

PIMD in ABINIT follows the set of numerical schemes developed by several authors in the 90's [13,14]. PIMD in the canonical ensemble needs specific thermostats to ensure that the trajectories

are ergodic: the Nose–Hoover chains are implemented, as well as the Langevin thermostat (controlled by the value of **imgmov**). Also, it is possible to use coordinate transformations (staging and normal mode), that are controlled by the keyword **pitransform**. In standard equilibrium PIMD, only static observables are relevant (quantum time-correlation functions are not accessible): the masses associated to the motion of the beads are controlled by the keyword **pimass**, whereas the true masses of the atoms are given by **amu**. The values given in **pimass** are used to fix the so-called fictitious masses [13]. In the case where a coordinate transformation is used, the fictitious masses are automatically fixed in the code to match the so-called staging masses or normal mode masses. The number of time steps of the trajectory is controlled by **ntimimage**, the initial and thermostat temperature by **mdtemp(2)**. Except if specified, the images in the initial configuration are assumed to be at the same position, and a random distribution of velocities is applied (governed by **mdtemp(1)**) to start the dynamics.

At each time step, ABINIT delivers in the output file:

(i) information about the ground state DFT calculation of the ground state for each image

(ii) the instantaneous temperature, the instantaneous energy as given by the primitive and virial estimators, and the pressure tensor as given by the primitive estimator.

An automatic test is given with the code (v7#08) and a tutorial will be provided in a next version of the package.

PIMD in ABINIT enters a larger family of algorithms that use images of the system (controlled by the keyword **imgmov**), such as the String method or the Nudged Elastic Band method, both described in the next section (Section 3.2). The number of images (keyword **nimage**) is associated to a specific parallelization level (keyword **npimage**).

PIMD has been used with ABINIT to reproduce the large isotope effect on the phase transition between phase I and phase II of dense hydrogen [15], and also some aspects of diffusion at low and room temperature in proton-conducting oxides for fuel cells [16]. PIMD in the NPT ensemble is not available yet.

3.2. Transition states from image algorithms

Similarly to the PIMD, finding minimum energy paths and in particular transition states for chemical transformations is of great importance in many different fields. In ABINIT we have implemented two different flavors based on interpolation methods [17–19] and controlled by the keyword **imgmov**. The calculation starts with the knowledge of the initial and final state (local minima in the configuration space) and an educated guess for the reaction pathway. If the reaction path is not given, a linear interpolation between the reactants and final products is constructed by a series of images (configurations) that connect the two states, which are given by the keyword **nimage**. The energy path that joins the series of images is then modified at each step to allow the search over the lowest energy path joining the reactants and products.

In the Nudge Elastic Band method (NEB), the images are connected through springs, with a spring constant that has to be chosen such that the images are uniformly spaced during the path search. The forces on each image come from the potential energy surface of that configuration and a spring force from the two closest configurations. The change in images is calculated by projecting out the true force perpendicular to the path and the parallel projection of the spring force with respect to the path [18]. The spring constant is obtained from the keyword **neb_spring** and the number of iterations is given by **ntimimage**. In the String method, the system set up is exactly the same as in the NEBM with the

difference that no spring constant needs to be defined. In this case, the forces are obtained as in the NEB method from the true force perpendicular but now the configurations are equally redistributed along the path at each iteration [17]. In both methods, the search stops if the number of predefined iterations (**ntimimage**) or the tolerance convergence criteria (**tolimg**) is reached.

As in the PIMD, each of the images can be treated in parallel and the requested parallelization mode is set with the keyword **npimage**. Automatic tests for the string method are provided: paral#08, tutoparal#03 and tutoparal#4, v6#24, v6#25. Moreover a tutorial for the use of the string method in parallel is available (see [10] or `doc/tutorial/lesson_parall_string.html`).

3.3. Developments in ground state calculations

3.3.1. Finite electric field

The effect of an homogeneous static electric field on an insulator may be treated in ABINIT from two perspectives. One is perturbative, and yields the susceptibility in the form of the second derivative of the total energy with respect to the electric field, at zero field strength. This approach is discussed below in Section 3.4.1. ABINIT can also be used to compute the effect of an electric field of finite amplitude, using techniques from the Modern Theory of Polarization [20–22]. In this approach, the total energy to minimize includes the contribution due to the interaction of the external electric field with the material polarization $\mathbf{P}_{\text{Tot}}$, as follows:

$$E = E_0 - \Omega \mathbf{P}_{\text{Tot}} \cdot \mathbf{E}, \tag{1}$$

where E_0 is the usual ground state energy obtained from Kohn–Sham DFT in the absence of the external field $\mathbf{E}$, $\mathbf{P}_{\text{Tot}}$ is the polarization, made up of an ionic contribution and an electronic contribution, and Ω the volume of the unit cell. The Modern Theory of Polarization provides a formula for the electronic part $\mathbf{P}_{\text{el}}$ as

$$\mathbf{P}_{\text{el}} \cdot \mathbf{b}_{\mathbf{i}} = \frac{fe}{\Omega} \frac{1}{N_{\perp}^i} \sum^{N_{\perp}^i} \text{Im} \ln \prod^{N_{\parallel}^i} \det M^{\mathbf{k_i}, \mathbf{k_i} + \Delta \mathbf{k_i}} \tag{2}$$

where f is the spin degeneracy and e the charge (-1 for electrons). Here the component of $\mathbf{P}_{\text{el}}$ in direction $\mathbf{b_i}$ of the reciprocal lattice is computed, by discretizing a line integral through the Brillouin zone in the $\mathbf{b_i}$ direction and summing over points in the plane perpendicular to this line. The integrand, which is the Berry connection, is discretized in terms of the matrix $M^{\mathbf{k_i}, \mathbf{k_i} + \Delta \mathbf{k_i}}$ [23], with matrix elements

$$M_{mn}^{\mathbf{k_i}, \mathbf{k_i} + \Delta \mathbf{k_i}} = \langle u_{m\mathbf{k_i}} | u_{n\mathbf{k_i} + \Delta \mathbf{k_i}} \rangle, \tag{3}$$

that is, the overlap of Bloch states $u_{m\mathbf{k_i}}$ at neighboring k-points along the integration path.

In ABINIT, the polarization of the ground state may be computed using Eq. (2), and the effect of an external electric field is included by minimizing the total energy in Eq. (1). The latter case proceeds as usual by expressing the energy in terms of matrix elements of the Kohn–Sham Hamiltonian (and the polarization term), and functionally differentiating with respect to the bra of interest $\langle u_{m\mathbf{k_i}} |$. The discretized form in Eq. (2) is handled by use of the matrix identity $d(\ln \det M)/d\lambda = \text{Tr}(M^{-1} dM/d\lambda)$. Either the NCPP or the PAW scheme may be used [24]. In the case of PAW, the matrix elements of Eq. (3) have contributions from both the planewave expansion of the Bloch functions and from the atom-centered projectors, as follows:

$$\langle u_{m\mathbf{k_i}} | u_{n\mathbf{k_i} + \Delta \mathbf{k_i}} \rangle = \langle \tilde{u}_{m\mathbf{k_i}} | \tilde{u}_{n\mathbf{k_i} + \Delta \mathbf{k_i}} \rangle$$
$$+ \sum_{qrI} \langle \tilde{u}_{m\mathbf{k_i}} | \tilde{p}_{q\mathbf{k_i}}^I \rangle Q_{qr}^I (\Delta \mathbf{k_i}) \langle \tilde{p}_{r\mathbf{k_i} + \Delta \mathbf{k_i}}^I | \tilde{u}_{n\mathbf{k_i} + \Delta \mathbf{k_i}} \rangle, \tag{4}$$

where the sum runs over ions, labeled by I, and atomic-like orbitals φ_q^I on each ion labeled by q, the $\tilde{p}$ are the PAW projectors, $\tilde{u}_{m\mathbf{k_i}}$ the pseudized Bloch states, and

$$Q_{qr}^I(\boldsymbol{\Delta}\mathbf{k_i}) = e^{-il\cdot\boldsymbol{\Delta}\mathbf{k_i}}\left[\langle\varphi_q^I|e^{-i\boldsymbol{\Delta}\mathbf{k_i}\cdot(\mathbf{r}-\mathbf{I})}|\varphi_r^I\rangle - \langle\tilde{\varphi}_q^I|e^{-i\boldsymbol{\Delta}\mathbf{k_i}\cdot(\mathbf{r}-\mathbf{I})}|\tilde{\varphi}_r^I\rangle\right] \quad (5)$$

is the on-site PAW contribution to the overlap of states at adjacent $\mathbf{k}$-points. Expansion of Q in terms of $\boldsymbol{\Delta}\mathbf{k}$ leads to a first-order contribution that is effectively an on-site dipole moment, as identified already in the ultra-soft pseudopotential treatment of polarization [25]. However the exponential in Q can be treated exactly to all orders in Eq. (4), in which case the dipole should not be additionally included (as it erroneously was in Ref. [24]). In ABINIT the exact treatment is used, and the dipole is also reported for reference.

In the NCPP case, the electric field has no additional contribution to the Hellmann–Feynman forces, because the electronic states do not depend explicitly on ionic position [22]. In the PAW case however, as the projectors do depend on ion location, an additional force term arises when Eq. (2) is differentiated with respect to ion position, and similarly for stress [24]. The same identity mentioned above is used to differentiate $\ln\det M$ with respect to ion position or strain.

The polarization and finite electric field calculation in ABINIT is accessed through the variables **berryopt** and **efield**. In addition, displacement fields and mixed boundary conditions (a mix of electric field and displacement field) can be computed as well. A tutorial is also provided for these calculations (see [10] or `doc/tutorial/lesson_ffield.html`).

3.3.2. Two component Density Functional Theory: electron–positron annihilation

Two-component Density Functional Theory (TCDFT) [26–28] is a generalization of the Density Functional Theory, which allows one to determine densities and wavefunctions of interacting positrons and electrons. These quantities, in turn, can be used to model two positron annihilation characteristics, the lifetime of a positron and the momentum distribution of annihilating electron–positron pairs (Doppler broadening), which are measured in positron annihilation spectroscopy (PAS) [29]. Numerous coupled experimental and theoretical positron annihilation studies [30–35] on defects in solids prove the importance of first principles calculations for defect identification. A fully self-consistent TCDFT scheme along with the positron lifetime calculations has been implemented in ABINIT in the PAW [36] framework since v6.0.1. Methods for Doppler broadening of the annihilation radiation calculations are available since v7.10.1. The implementation is described in detail in Ref. [37].

The TCDFT is implemented as a double loop on the electronic and positronic densities: during each subloop, one of the two densities (and Hamiltonians) is kept constant while the other is being converged. There are two ways of performing positron calculations: In the first, manual, way, one needs to perform electronic ground-state calculation (**positron = 0**), then a positronic ground-state calculation (**positron = 1**) with the electronic density automatically read from a _DEN file, followed by an electronic calculation in the presence of the positron (**positron = 2**). The second and the third steps can be then repeated. The procedure can be also performed automatically using **positron = −1** (electronic and positronic wavefunctions not stored in memory), **positron = −10** (electronic and positronic wavefunctions stored in memory). The convergence of the total energy (resp. the forces) for the positrons is controlled by the use of the **postoldfe** (resp. **postoldff**) and **posnstep** input keywords.

Forces and stresses, including contributions from the electrons and the positron, can be evaluated, and the calculation continued with a new atomic configuration and/or new geometry. In this implementation we use a unified formalism for the positron and the electrons: the wavefunctions of the electrons and the positron in the system are expressed on the same mixed basis (plane-waves and atomic orbitals). Thanks to this, we can use the same PAW datasets for electrons and positron. The PAW datasets, however, are originally generated to describe electronic wavefunctions and not the positronic ones. Because of the nature of the electron–ion and positron–ion interactions, the shapes of the corresponding wavefunctions differ strongly. Therefore, in some cases, a standard PAW dataset can be inappropriate for the positron description. Two ways of overcoming this problem are proposed [37]: First, the number of valence electrons in the PAW dataset can be increased. The second, less time consuming solution, is to add the partial waves corresponding to the semicore electrons in the basis used for the positron wavefunction description, while keeping the initial number of valence electrons. The PAW datasets can be generated using a modified version of the ATOMPAW generator [38].

Several electron–positron correlation functionals, including LDA and GGA zero-positron density limit and LDA full electron–positron correlation functional, are available and are controlled by the keyword **ixpositron**.

At the end of each positron calculation, a positron lifetime (decomposed in several contributions) is printed. In order to output the momentum distributions of annihilating electron–positron pairs the **posdoppler** keyword should be set to 1. The three-dimensional momentum distributions can be then projected in various directions, giving Doppler spectra for comparison with experiments, using the `posdopspectra` script in the `scripts/post_processing` directory. An automatic test is given with the code (v7#35).

The TCDFT calculations have been parallelized on three levels, allowing one to use the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) [39] or the Chebyshev filtering algorithm [40]. This means that the processors can be distributed between the $\mathbf{k}$-points, bands and FFT grid points during the density, lifetime and momentum distribution calculations.

The TCDFT implementation in ABINIT has been used to contribute to identify defects in SiC [41,42,34] and UO_2 [35].

3.3.3. Other ground state developments

Kinetic energy density

The computation of the kinetic energy density

$$\tau(\mathbf{r}) = \frac{1}{2}\sum_{\mathbf{k}}\sum_n f(E_{n\mathbf{k}})|\nabla\psi_{n\mathbf{k}}(\mathbf{r})|^2 \quad (6)$$

where $f(E_{n\mathbf{k}})$ is the occupation number of the eigenstate $\psi_{n\mathbf{k}}(\mathbf{r})$, has been implemented in the ABINIT software from version 6 for the norm-conserving pseudopotential case. An implementation for the PAW case will become available in a future release of the code. The kinetic energy density is computed if the keyword **usekden** is set to 1. The kinetic energy density is mainly useful in two places in the code. It is used in the calculation of the electron localization function (ELF), and can also be required when using meta-GGA (MGGA) functionals.

ELF

The ELF [43,44] gives a dimensionless measure of the electron pairing probability, that is the probability of finding an electron in the vicinity of another reference electron. Similarly to the *Atoms in Molecules* (AIM) method [45], the ELF is mainly used in quantum chemical topology analysis to scrutinize the chemical bonding. The electron localization function is defined as

$$\text{ELF}(\mathbf{r}) = \frac{1}{1 + \left(\frac{D(\mathbf{r})}{D^0(\mathbf{r})}\right)^2} \quad (7)$$

and falls in the range [0, 1]. The ELF is equal to 1 for perfect localization and ELF $= 1/2$ when the Pauli kinetic energy density

$$D(\mathbf{r}) = \tau(\mathbf{r}) - \frac{1}{8}\frac{|\nabla n(\mathbf{r})|^2}{n(\mathbf{r})} \tag{8}$$

becomes equivalent to that of a homogeneous electron gas

$$D^0(\mathbf{r}) = \frac{3}{10}(3\pi^2)^{2/3}n^{5/3}(\mathbf{r}). \tag{9}$$

Using the keyword **prtelf**, the ELF can be output in a _ELF file which can be post processed and analyzed similarly to the _DEN file which contains the electronic density.

The LibXC library and Meta-GGA functions

The Libxc project [46] provides an implementation of almost all published exchange–correlation functionals, in a well tested, robust library which is released under the GNU Lesser General Public License (version 3.0). It is written in C, and includes bindings for Fortran.

In ABINIT, starting from version 5.7, it became possible to use Libxc to obtain the exchange–correlation energy E_{xc} and potential v_{xc} within LDA and GGA. It is now also possible to use it also within MGGA and for hybrid functionals (see Section 6.3 for the latter).

The MGGA functionals come as the third rung of the Jacob's ladder of DFT after LDA and GGA. The functionals from this family have an added dependence on either the kinetic energy density (τ) or the Laplacian of the electronic density ($\nabla^2 n$), or both. This means that τ and $\nabla^2 n$ are required in the calculation of the MGGA exchange–correlation energy E_{xc}. The calculations of τ and $\nabla^2 n$ are interfaced with Libxc such that many MGGA functionals are available in ABINIT. Automatic tests of the implementation are provided from v6#31 to v6#33, and the corresponding documentation can be found in the directory doc/theory/MGGA. However, the present implementation does not yet include an adapted mixing procedure in the self-consistent-field cycles. Therefore, only a restricted list of MGGA is currently available. We recommend to the MGGA-users to employ the V_{xc}−only MGGA based on an LDA correlation part (typically **ixc** −207 012, −208 012, and −209 012), and not those which allow total energy calculations (and the associated computation of forces). Consequently, structural optimization is not yet possible within a full MGGA approach. The use of MGGA functionals is expected to yield a better accuracy on ground state predictions although this can depend strongly on the chosen functional. As an example, it has been demonstrated that the TB09 MGGA functional allows a better prediction of the electronic band gaps [47]. However, it is worth noting that in this particular case this might be at the price of a worse band width and thus a worse description of the effective masses [48].

External magnetic field

An applied external magnetic field has been implemented in ABINIT by considering the Zeeman spin response only (*i.e.*, neglecting the orbital contribution). Following the procedure of Bousquet et al.,[49] the applied **B** field is introduced by adding the following term in the non-collinear Kohn–Sham potential:

$$V_{Zeeman} = -\frac{g}{2}\mu_B \begin{pmatrix} B_z & B_x + iB_y \\ B_x - iB_y & -B_z \end{pmatrix} \tag{10}$$

where g is the Landé factor for the spins, μ_B the Bohr magneton and B_i the component of the B-field along $i = x, y$ and z directions. This contribution is trivial to implement, and also dominant in amplitude, but has historically been neglected with respect to the orbital responses, which are rich in more complex physics. Unlike an applied electric field, such a Zeeman term in the potential is compatible with periodic boundary conditions. It is also compatible with collinear calculations by reducing its application on

"up" and "down" spin channels with $\mathbf{B} = B\vec{e}_z$. In ABINIT, the finite Zeeman field is controlled by the keyword **zeemanfield** which allows to control the amplitude of the applied **B**-field (in Tesla) along the three cartesian directions. Such an applied Zeeman field allows one to calculate the spin contribution of the magnetic and magnetoelectric susceptibilities, and to observe phase transitions under finite magnetic field, if present. The automatic test v6#17 shows an example of an applied Zeeman magnetic field on an isolated Bi atom.

Constrained atomic magnetic moments

A complementary magnetic constraint method has been implemented in the ABINIT code, wherein the magnetization around each atom is pushed to a desired (vectorial) value. The constraint can either be on the full vector quantity, $\vec{m}$, or only on the direction $\hat{m}$. This is mainly useful for non collinear systems, where the direction and amplitude of the magnetic moment can change. The method follows that used in the Quantum Espresso [50] and VASP [51] codes: a Lagrangian constraint is applied to the energy, and works through a resulting term in the potential, which acts on the different spin components. The magnetization in a sphere Ω_i around atom i at position $\vec{R}_i$ is calculated as:

$$\vec{m} = \int_{\Omega_i} \vec{m}(\vec{r})d\vec{r} \tag{11}$$

and the corresponding potential for spin component α is written as:

$$V_\alpha = 2\lambda f(|\vec{r} - \vec{R}_i|/r_s)\vec{c}_\alpha. \tag{12}$$

The function $f(x) = x^2(3 + x(1 + x(-6 + 3x)))$, is applied to smooth the transition near the edge of the sphere around $\vec{R}_i$, over a thickness r_s (by default 0.05 bohr, and f is set to 0 for $|\vec{r} - \vec{R}_i| > r_s$). This minimizes discontinuous variations of the potential from iteration to iteration.

The constraint is managed by the keyword **magconon**. Value 1 gives a constraint on the direction ($\vec{c} = \vec{m} - \hat{\vec{s}}_i(\hat{\vec{s}}_i \cdot \vec{m})$), value 2 gives a full constraint on the vector ($\vec{c} = \vec{m} - \vec{s}_i$), with respect to the keyword **spinat** ($\vec{s}_i$ above), giving a 3-vector for each atom. The latter is quite a stringent constraint, and often may not converge. The former value usually works, provided sufficient precision is given for the calculation of the magnetic moment (kinetic energy cutoff in particular).

The strength of the constraint is given by the keyword **magcon_lambda** (λ above−real valued). Typical values are 10^{-2} but vary strongly with system type: this value should be started small (here the constraint may not be enforced fully) and increased. A too large value leads to oscillations of the magnetization (the equivalent of charge sloshing) which do not converge. A corresponding Lagrange penalty term is added to the total energy, and is printed to the log file, along with the effective magnetic field being applied. In an ideal case the energy penalty term should go to 0 (the constraint is fully satisfied). The automatic test v7#05 runs different types of constraint combinations, for BCC iron.

Charged cells

It is well known that the electrostatic potential (arising from ion–ion, ion–electron, and electron–electron interactions) is ill-defined within periodic boundary conditions. However, it is less well known that the total energy of a charged cell is also ill-defined. In fact, after a careful derivation in Ref. [52], it was shown that the above two statements are tightly linked: when the number of electrons differs from the number of protons in a cell, the necessary compensating background that enforces the overall charge neutrality is sensitive to the arbitrary average electrostatic potential.

ABINITv7.10 offers the possibility to choose which convention to use for the average electrostatic potential with the keyword **usepotzero**. In PAW, one can choose among 3 options:

- the average of smooth electrostatic potential is set to zero;
- the average of all-electron electrostatic potential is set to zero;
- the average of smooth electrostatic potential is set to a finite value, which follows the Quantum Espresso implementation (see Ref. [53] for more details).

Only options 1 and 3 are valid for the NCPP case.

None of these conventions is intrinsically more correct than the other ones. This is just an arbitrary choice, but ABINIT now permits a straight comparison to the other codes.

3.4. Developments in linear and non-linear responses

3.4.1. Projector augmented-wave approach to Density-Functional Perturbation Theory

Density-Functional Perturbation Theory (DFPT) allows one to address a large variety of physical observables. Many properties of interest can be computed directly from the derivatives of the energy, without the use of finite differences: phonons modes, elastic tensors, effective charges, dielectric tensors, *etc*. Such DFPT capabilities have been implemented for years in ABINIT in the NCPP scheme [54,55], but they were not available in the framework of the PAW approach.

From version 6, linear responses started to be implemented within PAW. They can almost all be used in the current production version of ABINIT 7.10. The few missing mixed responses will be available with version 8.

Prior to implementation, it was necessary to derive the theoretical formalism [56,57]. Compared to NCCP, the equations become more complex because of: (1) the PAW "on-site" contributions, (2) the non-orthogonality of the pseudo wave-functions, (3) the introduction of a compensation charge density. A PAW-DFPT calculation workflow is similar to that of a NCPP calculation, except that one uses a PAW atomic dataset, and is conducted as follows:

- Run a Ground-State calculation in order to extract the Kohn–Sham pseudo wave-functions $\tilde{\psi}_n$; these must be extremely well converged.
- If necessary, *e.g.*, for the application of the derivative of the Hamiltonian with respect to an electric field, determine the derivatives of the wave functions with respect to the wave vector **k**, and keep them in a file. The keyword **rfddk** is used to perform this type of calculation.
- Compute the 2nd-order derivative matrix (*i.e.*, 2nd derivatives of the energy with respect to different perturbations λ). This can be done thanks to the keywords **rfphon** (λ = atomic displacement), **rfstr** (λ = strain) or **rfelfd** (λ = electric field). This calculation is performed in 3 steps with ABINIT:
  (1) Compute the constant-wavefunctions term.
  (2) Determine the derivatives of the wavefunctions by solving the *Sternheimer equation*. In the PAW approach, handling non-orthogonal pseudo wavefunctions, this equation, and the associated orthogonality condition (in the *parallel-transport gauge*), are:

$$P_c^* \left(\mathcal{H} - \epsilon_n \mathcal{S} \right) P_c \left| \frac{\partial \tilde{\psi}_n}{\partial \lambda} \right\rangle = -P_c^* \left(\frac{\partial \mathcal{H}}{\partial \lambda} - \epsilon_n \frac{\partial \mathcal{S}}{\partial \lambda} \right) |\tilde{\psi}_n\rangle \quad (13)$$

$$\left\langle \frac{\partial \tilde{\psi}_n}{\partial \lambda} \right| \mathcal{S} |\tilde{\psi}_n\rangle = -\frac{1}{2} \langle \tilde{\psi}_n | \frac{\partial \mathcal{S}}{\partial \lambda} |\tilde{\psi}_n\rangle \quad (14)$$

$\mathcal{H}$ (resp. $\mathcal{S}$) is the Hamiltonian (resp. Overlap) operator. P_c is the projector defined as $P_c = 1 - \sum_n |\tilde{\psi}_n\rangle\langle\tilde{\psi}_n| \mathcal{S}$.

(3) Mix the wavefunction derivatives with the (non self-consistent) derivatives of the Hamiltonian to compute the complete derivative database.

- Launch the `anaddb` tool (distributed with ABINIT) to analyze the derivative database and compute relaxed tensors and thermodynamical properties.

Running a DFPT calculation within PAW requires no other additional parameter than using a PAW atomic dataset (available on the web site thanks to the PAW JTH table, see 5.3.2) and adding—as for the ground-state, the **pawecutdg** keyword to tune the auxiliary fine grid. The parameter **pawecutdg**, should be converged even more carefully than in the ground state (by testing the value of the desired response function) because it has a strong numerical impact on the second derivatives of the energy.

Note also that, when performing the post-processing with `anaddb`, it is recommended to include all the keywords enforcing the sum rules (acoustic sum and charge neutrality). Indeed the PAW formalism involves, for each atom, the calculation of a large number of real space integrals, whose numerical effect may be to break the translational invariance.

With ABINIT 7.10, it is possible to access to the following response properties with the PAW precision: (1) Phonon spectra for metals and insulators (including electric field contribution to LO–TO splitting), (2) dielectric tensor, (3) clamped-ion elastic tensor, (4) relaxed elastic tensor (including forces-internal strain coupling coefficients) for metals. Other responses (effective charges and piezoelectric tensor) are already planned for ABINIT v8, as well as some non-linear responses. `anaddb` then allows to compute many physical and thermodynamical properties containing derivatives of the energy.

Thanks to the locality provided by PAW partial wave basis, it is possible to perform response function calculations for correlated electron materials. The LDA+U formalism is usable without any restriction for the PAW + DFPT calculations.

Several automatic tests delivered in the package can be used as starting points to create input files for PAW-DFPT calculations: computation of wavefunction derivatives with respect to **k** (v5#05), computation of dielectric tensor (v5#30), computation of phonons at several **q** vectors for insulators (v6#62, libxc#82) and metals (v6#89). All the tutorials dedicated to response functions can be followed by substituting the NCPP files by PAW atomic datasets.

3.4.2. Electron–phonon coupling and transport

The calculation of electron–phonon coupling (EPC) quantities has been available for over 10 years in the ABINIT code. These will be very briefly summarized, and new implementations described, which go beyond the elastic Lowest Order Variational Approximation (LOVA).

The calculation of transport quantities (for the normal and superconducting states) dates back to Ziman [58] and before, but was pioneered in numerical approaches by P.B. Allen [59,60]. He applied an alternative method to the relaxation time approximation (RTA), based on an efficient basis set known as Fermi Surface Harmonics (FSH). These are orthogonalized combinations of polynomials in the group velocity components $\vec{v}_k$, which form a complete basis for the *k* variation of the electron–phonon coupling. The strong advantage is that, in the solution of the Boltzmann Transport Equations (BTE), only the 0th and 1st order terms appear explicitly in drift and diffusion. Thus, limiting the basis set to the 1st order gives simple, closed, and relatively accurate forms for the transport coefficients. The variational solution in the FSH basis gives the current/electric field/conductivity which maximize the production of entropy.

For a full solution of the BTE, the basis set FSH are multiplied by functions of the electronic energy $\sigma(\epsilon)$ (the dependency on k is thus double, through v_k and ϵ_k). Here too, orthogonalized polynomial functions are used, and their order n is limited to 0 and 1, which is enough to yield the conductivity and Seebeck coefficients.

The formulae for the resistivity and electrical thermal conductivity are well summarized in the book by Grimvall [61]. Their implementation in an ab-initio/LMTO framework by Savrasov is the standard reference within DFT [62], yielding the Eliashberg spectral function $\alpha^2 F(\omega)$ (with the related EPC coupling strength and phonon linewidths), and its "transport" versions $\alpha^2 F_{tr}(\omega)$ coined by Allen.

Basic calculations of electron–phonon interaction in ABINIT have been reviewed in previous papers, and are covered by an existing tutorial. One performs a normal ground state, then phonon calculations, with an added keyword `prtgkk` which saves the matrix elements to files suffixed GKK. The main change in this respect is that `prtgkk` now disables the use of symmetry in reducing k-points and perturbations. This avoids ambiguities in wave function phases due to band degeneracies. The resulting GKK files are merged using the `mrggkk` utility, and processed by `anaddb` if the flag `elphflag` is set to 1. With the implementation of phonons in PAW DFPT, the electron–phonon coupling is also available in PAW, though this has not yet been tested extensively.

The LOVA approximation for transport was first used by Allen with an additional approximation, that of *elastic processes*, where the central EPC kernel is evaluated neglecting its electron energy variation (elastic LOVA; set **ifltransport** = 1 in `anaddb`):

$$\alpha^2 F_{tr}(s, s'; \alpha, \beta; \epsilon, \epsilon'; \omega) \simeq \alpha^2 F_{tr}(s, s'; \alpha, \beta; \omega)$$
$$\simeq \alpha^2 F_{tr}(1, 1; \alpha, \beta; \omega) \qquad (15)$$

where $s, s' = \pm 1$ are akin to in/out scattering, ϵ, ϵ' are electron energies (if unspecified they are at the Fermi energy), ω is a phonon energy, and α, β are cartesian directions.

This is legitimate in simple metals, but will fail for more complex band structures, such as narrow bands metals, or doped semiconductors with Fermi levels near to or in the band gap. Furthermore, the Seebeck coefficient (S) vanishes in the elastic LOVA as the asymmetry of electrons and holes is completely lost.

Going beyond the elastic LOVA (input variable **ifltransport** = 2 for inelastic LOVA) can overcome the above-mentioned shortcomings, at the cost of significantly longer computation time and higher memory usage. The Seebeck tensor is obtained from the ratio of two scattering operator matrix elements: $S_{\alpha\beta} \approx \frac{\pi k_B}{\sqrt{3}e} \sum_\gamma (Q_{01})_{\alpha\gamma} (Q_{11})^{-1}_{\gamma\beta}$, where the Q matrix is composed of the product of the transport spectral function $\alpha^2 F_{tr}$, the energy polynomials, and occupation functions. The electron–phonon scattering has been fully accounted for by fulfilling the conservation of energy and $\mathbf{k}' - \mathbf{k} = \mathbf{q} + \mathbf{g}$ with $\mathbf{g}$ a reciprocal lattice vector. The sign and magnitude of S delicately relies on the energy dependence of these quantities, such that relatively dense k- and q-meshes are necessary. Note that only one fixed temperature is allowed per run, when setting **ifltransport** to 2.

The LOVA Seebeck calculation overcomes the limitations of the often adopted constant RTA approach. A peculiar example has been studied in Ref. [63], where the unusual positive Seebeck of the simple alkali metal Lithium is well reproduced and analyzed. By contrast, the constant RTA yields a negative sign of S.

When the Fermi energy is near to or in the band gap, the (non constant) relaxation time approach is more appropriate. By setting **ifltransport** to 3, the fully *ab initio* k-dependent electron life time is computed, as in the work of Restrepo [64] and others. It can be used to further calculate the electronic transport properties:

$$\frac{1}{\tau_{\mathbf{k}}} = \frac{2\pi}{\hbar} \sum_{\mathbf{q}} |g_{\mathbf{k}, \mathbf{k}+\mathbf{q}}|^2 \Big\{ [f(\epsilon_{\mathbf{k}+\mathbf{q}}) + n_{\mathbf{q}}] \delta(\epsilon_{\mathbf{k}} - \epsilon_{\mathbf{k}+\mathbf{q}} + \hbar\omega_{\mathbf{q}})$$
$$+ [1 + n_{\mathbf{q}} - f(\epsilon_{\mathbf{k}+\mathbf{q}})] \delta(\epsilon_{\mathbf{k}} - \epsilon_{\mathbf{k}+\mathbf{q}} - \hbar\omega_{\mathbf{q}}) \Big\} \qquad (16)$$

where g are the EPC matrix elements, f and n are the Fermi and Bose distributions. Within this (non constant) RTA, the energy variation of the relaxation time (weighted average over a uniform k-mesh) and mean free path (multiplied by the k-dependent velocities) are also calculated.

Other improvements include the consideration of the temperature dependence of the Fermi energy; cubic spline interpolation (**ep_nspline**) to linearly interpolate the transport arrays and reduce the memory usage. Besides setting the Fermi level with **elph_fermie** (in Hartree), it is also possible to specify the extra electrons per unit cell, (*i.e.*, the doping concentration often expressed in cm^{-3}) with **ep_extrael**. New tests are found in v7#90 to #94 concern electron–phonon calculations and transport for bcc Li.

3.4.3. Temperature dependence of the eigenenergies

The electronic structure changes with temperature. In most materials, such changes are mainly driven by the electron–phonon interaction, which is also present at zero Kelvin, inducing the so-called zero-point motion renormalization (ZPR) of the eigenvalues. These effects can be computed thanks to the Allen-Heine-Cardona (AHC) theory [65–67], which is based on diagrammatic method of Many-Body Perturbation Theory. An extension to the standard AHC theory also gives access to the electronic lifetime and decay rates. These physical properties are available from ABINIT since v7.10.4.

The ABINIT implementation of the AHC formalism relies on Density-Functional Perturbation Theory (DFPT) [68,54] and therefore allows for the calculation of the aforementioned quantities using the primitive cell only. It contains two main approximations: the harmonic approximation and the rigid-ion approximation (RIA). One can show that two Feynman diagrams are taken into account if these approximations are made: the Fan term (with two first-order electron–phonon interactions), and the Debye–Waller (with one second-order electron–phonon interaction). The impact of the electron–phonon coupling on the electronic eigenstate can be computed for semiconductors and insulators within an adiabatic or non-adiabatic framework. The AHC formalism and the implemented equations can be found in Ref. [69]. An extended verification and validation study (also versus other first-principle codes) of the ABINIT implementation can be found in Ref. [70]. The AHC implementation can be used with any XC functional working with the response-function (RF) part of the code, and requires the use of norm-conserving pseudopotentials. NetCDF support is mandatory. In v7.10.4, only non-spin-polarized calculations are feasible (this constraint has been removed in more recent version, see later).

The AHC implementation in ABINIT is built on a Sternheimer approach to efficiently compute the sum over highly energetic bands appearing in the AHC equations [71]. Such behavior is controlled by the input variable **ieig2rf**. The **k**-point convergence can be strongly improved by restoring the charge neutrality through the reading of the Born effective charge and dielectric tensor (controlled by the input variable **getddb**). More information on the importance of charge neutrality fulfillment can be found in Ref. [72]. The value of **elph2_imagden** sets the imaginary shifts used to smooth numerical instabilities in the denominator of the sum-over-states expression. We have checked that the implementation correctly holds for arbitrarily small **elph2_imagden**

parameters, see Ref. [72]. The input variable **smdelta** triggers the calculation of the electronic lifetime and the value of the smearing delta function can be specified through **esmear**. A double grid can be used to speed-up the calculations with **getwfkfine** or **irdwfkfine**. The variable **getgam_eig2nkq** gives the contribution at Γ so that the Debye–Waller term can be computed. This variable is only relevant for calculations of AHC using the `abinit` program only. It is nonetheless recommended to use the provided python post-processing script (`temperature_para.py` with its module `rf_mods.py` in the directory `scripts/post_processing/`) to allow for more flexibility. The python scripts support multi-threading.

The following steps are required to perform an AHC calculation:

(1) Perform a response function calculation at $\mathbf{q} = \Gamma$ with electric field perturbation.
(2) Perform phonon calculations and produce the EPC for a large set of wavevectors $\mathbf{q}$, reading the Born effective charge and dielectric tensor with **getddb**.
(3) Gather and compute the impact of the electron–phonon coupling on the electronic eigenenergies using the `temperature_para.py` python script.

The outputs of the script are provided in text and NetCDF format to allow for later reading inside ABINIT. This could be used in the future developments of ABINIT to compute temperature-dependent optical properties for example.

A family of automatic tests are given with the code (v7#50 to #59.in) and a tutorial is available on the ABINIT wiki [73].

The AHC implementation has been used with ABINIT to show that the RIA fails in molecules [71] but holds in periodic solids [69]. The ZPR, temperature-dependence of the gaps and electronic lifetime of diamond, silicon, boron nitride and two phases of aluminum nitride have been investigated using the non-adiabatic AHC theory [72]. The theory is quite successful, with perhaps a small systematic underestimation of the experimental data by ten or twenty percent. The impact of treating the electron–phonon coupling within GW instead of DFT–LDA has been investigated using finite differences, extrapolated with the AHC [74], in the case of diamond. This more advanced treatment has improved significantly the agreement with experiment. Finally, the spectral function and anharmonic effects on the electron–phonon coupling have been studied for diamond, boron nitride, lithium fluoride and magnesium oxide [75].

In future versions of the code, the computation of temperature-dependent electronic structure, zero-point motion renormalization and electronic lifetime, will be also possible in case of spin-collinear calculations. Moreover, the python post-processing scripts will enable calculations with memory optimized `_GKK.nc` files produced by the option **ieig2rf 5** of ABINIT.

3.4.4. Raman intensities by linear response calculation

Two noticeable restrictions on the computation of Raman intensities (in the low laser frequency limit, as described in Sec. 5.3 of Ref. [8]) are waived in ABINITv7.10.5. Indeed, it is now possible to perform such calculations in the case of spin-polarized materials, and also in the case where the sampling of wave vectors along some direction is as low as one (*e.g.*, Raman intensities can be computed even if only the Gamma point is used to sample the Brillouin zone). The restriction to NCPP is still present however. The frequency-dependent Raman spectrum can be computed thanks to frozen-phonon calculations coupled with independent-particle or Bethe–Salpeter evaluations of the dielectric response (described in Section 3.5.1). While the former is not very accurate, but computationally easy, the latter is quite involved, but proves to be very accurate. We refer the reader to Ref. [76] for more details. Test v6#66 presents a case where the sampling of the Brillouin zone along two directions is one, while v6#67 is a spin-polarized computation of a Raman intensity coefficient.

3.4.5. The LibXC library and exchange–correlation kernels

The exchange–correlation kernel f_{xc} (which is the second derivative of the energy with respect to the density) can also be obtained from Libxc for LDA and GGA described in Section 3.3.3. This has already allowed for the assessment of the validity of various XC functionals for computing the vibrational, dielectric, and thermodynamical properties of materials [77].

Automatic tests are provided in the directory `tests/libxc`.

3.5. Developments in excited state calculations

3.5.1. Bethe–Salpeter equation

Many-Body Perturbation Theory (MBPT [4]) defines a rigorous framework for the description of excited-state properties based on the Green's function formalism. Within MBPT, one can calculate charged excitations (*i.e.* electron addition and removal energies), using for example Hedin's GW approximation [3] for the electron self-energy. In the same framework, neutral excitations (experimentally related to optical absorption spectra) are also well described through the solution of the Bethe–Salpeter equation (BSE [4,78]). At present, the BSE represents the most accurate approach for the ab initio study of neutral excitations in crystalline systems as it includes the attractive interaction between electrons and holes thus going beyond the random-phase approximation (RPA) employed in the GW approximation. In what follows, we mainly focus on the BSE implementation as the features of the GW part have been already discussed in the last ABINIT paper [8] and in the review article [79].

As discussed in Ref. [4], the polarizability of the many-body system, L, is given by $L = [H - \omega]^{-1} F$ where H is an effective two-particle Hamiltonian describing the interaction between electron and hole, and F depends on the occupation factors. The equation is usually solved in the so-called electron–hole (e–h) space (products of single-particle orbitals) in which H assumes the form:

$$H_{(n_1 n_2)(n_3 n_4)} = (\epsilon_{n_2} - \epsilon_{n_1})\delta_{n_1 n_3}\delta_{n_2 n_4} + (f_{n_2} - f_{n_1})K_{(n_1 n_2)(n_3 n_4)} \quad (17)$$

where n_i is a shorthand notation to denote band, k-point and spin index, and f is the occupation of the state.

The BSE kernel, K, in the above equations is given by $\bar{v} - W$ where $\bar{v}$ is the bare Coulomb interaction without the long-range Fourier component at $\mathbf{G} = 0$, and W is the (static) screened interaction.

The macroscopic dielectric function is obtained via:

$$\epsilon_M(\omega) = 1 - \lim_{\mathbf{q} \to 0} v(\mathbf{q}) \langle P(\mathbf{q})|[H - \omega]^{-1} F|P(\mathbf{q})\rangle \quad (18)$$

where we have introduced the oscillator matrix elements $P(\mathbf{q})_{n_1 n_2} = \langle n_2|e^{i\mathbf{q}\cdot\mathbf{r}}|n_1\rangle$.

BSE calculations are activated by setting **optdriver 99**. The number of valence and conduction states in the e–h basis set is controlled by the variables **bse_loband** and **nband**, respectively. By default, the code solves the BSE in the so-called Tamm–Dancoff approximation (TDA) in which the coupling between resonant and anti-resonant transitions is ignored. A full BSE calculation including the coupling term can be done by setting **bs_coupling 1** in the input file. The variable **bs_coulomb_term** specifies the treatment of the screened interaction in the BSE kernel. By default, the code reads the W matrix from the **SCR** file generated by the GW code (**optdriver 3**). Alternatively, one can model the spatial dependency of W with the model dielectric function proposed by Cappellini in Ref. [80]. This option requires the specification of **mdf_coulomb_term 21** and **mdf_epsinf**. The contribution due

to the exchange term ($\bar{v}$ operator) can be optionally excluded by using **bs_exchange_term 0**. This is equivalent to computing the macroscopic dielectric function without local field effects. The frequency mesh for the macroscopic dielectric function is specified by **bs_freq_mesh** while **zcut** defines the complex shift to avoid the divergences due to the presence of poles along the real axis (from a physical standpoint, this parameter is related to the electron lifetimes). In addition, a scissor operator of energy **soenergy** can be used to correct the initial band energies and mimic the opening of the KS gap introduced by the GW approximation.

Three different algorithms for the solution of the BSE are implemented: (1) Direct diagonalization, (2) Haydock recursive algorithm, (3) Conjugate gradient eigensolver. The **bs_algorithm** input variable allows the user to choose among them. The Haydock algorithm [81,82] is the recommended approach for the computation of optical spectra since its computational cost is $\mathcal{O}(\mu n^2)$, where μ is the number of iterations required to converge and n is the size of the matrix. In general, μ is smaller than n (and nearly independent of the size of the system) and therefore the Haydock method is much more efficient than the direct diagonalization that scales as $\mathcal{O}(n^3)$. The variables **bs_haydock_niter** and **bs_haydock_tol** control the Haydock iterative method. Unfortunately, the Haydock solver does not give direct access to the eigenstates of the Hamiltonian, hence it cannot be used for the study of the excitonic wavefunctions. The conjugate gradient (CG) method employs standard iterative techniques [83] to compute the lowest eigenstates of the BSE Hamiltonian with a computational cost that scales as $\mathcal{O}(kn^2)$, where k is the number of eigenstates requested (**bs_nstates**). This solver is more memory demanding than the Haydock approach since the eigenstates must be stored in memory, but it gives direct access to the excitonic states. The CG algorithm should be preferred over the direct diagonalization especially when the number of eigenstates is much smaller than the size of the BSE Hamiltonian. Note, however, that CG has been implemented only for TDA calculations (Hermitian matrices).

The most important results of the calculation are saved in five different files. The **BSR** file stores the upper triangle of the BSE resonant block in Fortran binary format (**BSC** for the coupling matrix). The **HAYDR_SAVE** file contains the coefficients of the tridiagonal matrix and the three vectors employed in the iterative algorithm. This file can be used to restart the calculation if convergence has not been achieved (related input variables **gethaydock** and **irdhaydock**). Finally, the macroscopic dielectric function with excitonic effects is reported in the **EXC_MDF** file while **RPA_NLF_MDF** and **GW_NLF_MDF** contain the RPA spectrum without local field effects obtained with KS energies and the GW energies, respectively.

The BSE code uses the MPI parallel paradigm to distribute both the memory and the computation among the processors. The calculation is split into two steps: first the BSE matrix is constructed in parallel and saved on disk, then the code reads the matrix elements from file and invokes the solver selected by the user. Each step employs a different data distribution optimized for that particular task. The direct inversion of the BSE Hamiltonian, for example, can be performed with SCALAPACK while the Haydock and the CG solver use MPI primitives to implement matrix–vector operations on MPI-distributed matrices. Both NCPP as well as PAW are supported.

The BSE tutorial (see [10] or `doc/tutorial/lesson_bse .html`) contains a section with a brief introduction to the formalism and a lesson explaining step by step how to perform convergence studies for optical spectra (see tutorial#tbs_1). The automatic tests validating the different capabilities of the code can be found in v67mbpt (#11, #14, #15, #16. #29, and #29). The tests paral#76 and tutoparal#tmbt_5 show how to run BSE

calculations in parallel. See also Section 6.2 for a description of new developments in the BSE code that will be released in the forthcoming versions.

3.5.2. Other developments in electronic excitations
Random electronic stopping power

The slowing down of a swift charged particle inside condensed matter has been a subject of intense interest since the advent of quantum-mechanics. The Lindhard formula [84] that gives the polarizability of the free electron gas has been developed specifically for this purpose. The kinetic energy lost by the impinging particle by unit of path length is named the stopping power. For large velocities, the stopping power is dominated by its electronic contribution: the arriving particle induces electronic excitations in the target. These electronic excitations in the target can be related to the inverse dielectric function $\varepsilon_{\mathbf{G},\mathbf{G}'}^{-1}(\mathbf{q}, \omega)$ provided that linear response theory is valid. As a consequence, the electronic stopping power randomized over all the possible impact parameters reads

$$S(\mathbf{v}) = \frac{4\pi Z^2}{N_\mathbf{q} \Omega} \frac{1}{|\mathbf{v}|} \sum_\mathbf{q}^{\mathrm{BZ}} \sum_\mathbf{G} \mathrm{Im} \left\{ -\varepsilon_{\mathbf{G},\mathbf{G}}^{-1}[\mathbf{q}, \mathbf{v}.(\mathbf{q}+\mathbf{G})] \right\} \frac{\mathbf{v}.(\mathbf{q}+\mathbf{G})}{|\mathbf{q}+\mathbf{G}|^2}, \quad (19)$$

where Z and $\mathbf{v}$ are respectively the charge and the velocity of the impinging particle, Ω is the unit cell volume, $N_\mathbf{q}$ is the number of $\mathbf{q}$-points in the first Brillouin zone, and $\mathbf{G}$ are reciprocal lattice vectors. Apart from an overall factor of 2, Eq. (19) is identical to the formula published in Ref. [85].

The GW module of ABINIT gives access to the full inverse dielectric function $\varepsilon_{\mathbf{G},\mathbf{G}'}^{-1}(\mathbf{q}, \omega)$ for a grid of frequencies ω. Then, the implementation of Eq. (19) is a post-processing employing a spline interpolation of the inverse dielectric function $\varepsilon_{\mathbf{G},\mathbf{G}}^{-1}(\mathbf{q}, \omega)$ in order to evaluate it at $\omega = \mathbf{v}.(\mathbf{q}+\mathbf{G})$. The energy cutoff on $\mathbf{G}$ is governed by the keyword **ecuteps**, as in the GW module. The integer **npel** and the cartesian vector **pvelmax** control the discretization of the particle velocity. An example is provided in test v7#67. Note that the absolute convergence of the random electronic stopping power is a delicate matter that generally requires thousands of empty states together with large values of the energy cutoff.

Correlation energy within the Random Phase Approximation

In the adiabatic-connection fluctuation–dissipation framework, the correlation energy of an electronic system can be related to the density–density correlation function, also known as the reducible polarizability. When further neglecting the exchange–correlation contribution to the polarizability, one obtains the celebrated random-phase approximation (RPA) correlation energy. This expression for the correlation energy can alternatively be derived from Many-Body Perturbation Theory. In this context, the RPA correlation energy corresponds to the GW total energy.

The RPA correlation energy can be expressed as a function of the dielectric matrix $\varepsilon_{\mathbf{G},\mathbf{G}'}(\mathbf{q}, \omega)$:

$$E_c = \frac{1}{2\pi} \int_0^{+\infty} d\omega \sum_\mathbf{q}^{\mathrm{BZ}} \mathrm{Tr} \left\{ \ln \left[\varepsilon_{\mathbf{G},\mathbf{G}'}(\mathbf{q}, i\omega) \right] \right.$$
$$\left. + \delta_{\mathbf{G},\mathbf{G}'} - \varepsilon_{\mathbf{G},\mathbf{G}'}(\mathbf{q}, i\omega) \right\}, \quad (20)$$

where the trace symbol Tr implies a sum over the reciprocal lattice vectors $\mathbf{G}$. The integral over the frequencies is performed along the imaginary axis, where the integrand function is very smooth. Only a few sampling frequencies are then necessary. In ABINIT, the RPA correlation energy is triggered by setting the keyword **gwrpacorr** to 1. An example is provided in test v67mbpt#19. As in the previous paragraph about the stopping power, the RPA correlation energy is a post-processed quantity from the GW module of ABINIT, which

takes care of evaluating the dielectric matrix for several imaginary frequencies.

The RPA correlation has been shown to capture the weak van der Waals interactions [86] and to drastically improve defect formation energies [87]. The convergence versus empty states and energy cutoff is generally very slow. It requires a careful convergence study. The situation can be improved with the use of an extrapolation scheme [88,89].

Calculation of the effective Coulomb interaction

LDA+U as well as DFT+DMFT (see 3.6) requires as input values the effective Coulomb interaction. Two ways to compute them are available in ABINIT.

Firstly, the constrained Random Phase Approximation [90] (**ucrpa**) allows one to take into account the screening of the Coulomb interaction between correlated electrons, by non-interacting electrons. For non-entangled bands (**ucrpa 1**), the bands excluded from the polarizability can be specified either by a band index (**ucrpa_bands**) or an energy window (**ucrpa_window**) [91]. For entangled bands (**ucrpa 2**), the scheme used in ABINIT [92,93,91] uses a band and k-point dependent weight to define the polarizability, using Wannier orbitals as correlated orbitals. Examples are given in automatic tests v7#23-#24-#25 and in v7#68. This method is well adapted to compute the effective interaction for the same orbitals used in DFT + DMFT. To use the same orbitals as in DFT + U, the Wannier functions should be defined with a very large energy window [91,94].

Secondly, a linear response method [95] is implemented. The implementation is not yet in production. The implementation in ABINIT takes into account the truncated atomic orbitals from PAW and therefore differs from the original work [95] treating full atomic orbitals. In particular, considerably higher effective values for U are found. A tutorial is available (see [10] or `doc/tutorial/lesson_udet.html`).

3.6. The combination of DFT with dynamical mean field theory

DFT fails to describe the ground state and/or the excited states of strongly correlated systems such as many lanthanides, actinides or transition metals. Indeed, exchange–correlation functionals are not (yet) able to describe the strong repulsive Coulomb interactions occurring among electrons in partly filled localized d or f orbitals. A way to improve their description is to explicitly include these interactions in the Hamiltonian. Solving it in the static mean field approximation, gives the DFT + U method [96,97], implemented in ABINIT [98,8]. The Dynamical Mean Field Theory [99] (DMFT), goes beyond, by solving exactly the local correlations for an atom in an effective field (*i.e.*, an Anderson model). The effective field reproduces the effect of the surrounding correlated atoms and is thus self-consistently related to the solution of the Anderson model [99].

The combination of DFT with DMFT [100,101] (**usedmft 1**) relies on:

- The definition of correlated orbitals. In ABINIT, we use Wannier functions built using projected local orbitals [102]. Wannier functions are unitarily related to a selected set of Kohn–Sham (KS) wavefunctions, specified in ABINIT by band indices **dmftbandi** and **dmftbandf**. As empty bands are necessary to build Wannier functions, it is required in DMFT calculations that the KS Hamiltonian is correctly diagonalized: use high values for **nnsclo** and **nline**.

  In order to make a first rough estimation of the orbital character of KS bands and choose the band index, the band structure with highlighted atomic orbital character (so called *fatbands*) can be plotted, using the **pawfatbnd** variable. Band structures obtained from projected orbitals Wannier functions can also be

plotted using **plowan_compute** and related variables (see also automatic tests v7#71 and v7#72).

- The choice of the screened Coulomb interaction U (**upawu**) and J (**jpawu**). Note that up to version 7.10.5 (but not in later versions) **jpawu 0** is required if the density matrix in the correlated subspace is not diagonal.

- The choice of the double counting correction [94]. The current default choice in ABINIT is **dmft_dc 1** which corresponds to the full localized limit.

- The method of resolution of the Anderson model. In ABINIT, it can be the Hubbard I method [94] (**dmft_solv 2**), the Continuous time Quantum Monte Carlo (CTQMC) method [103, 104] (**dmft_solv 5**) or the static mean field method (**dmft_solv 1**, equivalent to usual DFT + U [94]).

The practical solution of the DFT + DMFT scheme is usually presented as a double loop over, first, the local Green's function, and second the electronic local density [94]. The number of iterations of the two loops are determined by **dmft_iter** and **nstep**. However, in the general case, the most efficient way to carry out fully consistent DFT + DMFT calculations is to keep only the loop governed by **nstep**, while **dmft_iter = 1** [104], **dmft_rslf 1** (to read the self-energy file at each step of the DFT loop) and **prtden -1** (to be able to restart the calculation of each step of the DFT loop from the density file). Lastly, one linear and one logarithmic grid are used for Matsubara frequencies [101] determined by **dmft_nwli** and **dmft_nwlo**. (Typical values are 10 000 and 100, but convergence should be studied). More information can be obtained in the log file by setting **pawprtvol = 3**.

The main output of the calculations are (1) the imaginary time Green's function, from which spectral functions can be obtained using an external maximum entropy code [105], self-energies, from which quasiparticle renormalization weight can be extracted, the density matrix of correlated orbitals, and the internal energies [106]. The electronic entropic contribution to the free energy can also be obtained using **dmft_entropy** and **dmft_nlambda**.

The efficient CTQMC code in ABINIT, which is the most time consuming part of DMFT, uses the hybridization expansion [107,103] with a *density–density* multiorbital interaction [103]. Moreover, the hybridization function [103] is assumed to be diagonal in the orbital (or flavor) index. This is exact for cubic symmetry without spin–orbit coupling but, in general, one should always check that the off-diagonal terms are much smaller than the diagonal ones. The non diagonal hybridization and coupling to the exact rotationally invariant interaction [103] is not available in version 7.10.5. As the CTQMC solver uses a Fourier transform, the time grid **dmftqmc_l** in imaginary space must be chosen so that the Nyquist frequency, defined by $\pi * $ **dmftqmc_l** $* $ **tsmear**, is around 2 or 3 Ha. A convergence study should be performed on this variable. Moreover, the number of imaginary frequencies (**dmft_nwlo**) has to be set to at least twice the value of **dmftqmc_l**. Typical numbers of steps for the thermalization (**dmftqmc_therm**) and for the Monte Carlo runs (**dmftqmc_n**) are 10^6 and 10^9 respectively. The random number generator can be initialized with the variable **dmftqmc_seed**. Several other variables (**dmftctqmc_***) are available. **dmftctqmc_order** gives a histogram of the perturbation orders during the simulation, **dmftctqmc_gmove** customizes the global move tries (mainly useful for systems with high/low spin configurations), and **dmftctqmc_meas** sets the frequency of measurement of quantities.

Finally, a DFT+DMFT tutorial is available in the current version of ABINIT (see [10] or `doc/tutorial/lesson_dmft.html`). It covers the study of the electronic structure of $SrVO_3$.

4. Recently developed technical features in ABINIT

4.1. Wavelets and PAW

Wavelets (WVLs) are wave-like oscillations which decay to zero, used in DFT simulations as a basis set to expand the Kohn–Sham (KS) orbitals. Amongst the different WVL kinds, Daubechies WVLs [108] form an optimal basis set to simulate low-dimensional systems due to their spatial localization, systematic convergence and adaptability to specific geometrical structures.

In ABINIT Version 5.5, wavelets were implemented via the BigDFT library, which originally supported only a Gaussian form of NCCPs: Goedecker–Teter–Hutter (GTH) and Hartwigsen–Goedecker–Hutter (HGH) [109,110].

In the present version of ABINIT, a wavelet-based PAW approach is implemented. The resulting formalism contains the above mentioned virtues of WVLs plus the PAW all-electron accuracy and ultrasoft PPs efficiency. The implementation was made possible by making a PAW library inside ABINIT and generalizing the BigDFT library to PAW.

The key feature of our WVL-PAW method is the analytical application of the non-local potential, $V_{NL} = \sum_{ij} |\tilde{p}_i\rangle D_{ij} \langle \tilde{p}_j|$ where $(\tilde{p}_i)$ are numerical PAW projectors. The application of the Hamiltonian onto KS wavefunctions expanded in WVL basis (ϕ_i), implies 3D integrals of the form $\int \tilde{p}(\mathbf{r})\phi_i(\mathbf{r})d\mathbf{r}$. Inspired by the BigDFT method [111], the 3D integrals are simplified into a sum of products of 1D integrals in the three cartesian directions,

$$\int \tilde{p}(\mathbf{r})\phi_i(\mathbf{r})d\mathbf{r} = \sum_j^{N_g} W_1^j W_2^j W_3^j, \qquad (21)$$

where W are convolutions of Gaussian functions and wavelets. This is achieved by fitting the PAW projectors to a sum of complex Gaussians:

$$\tilde{p}(\mathbf{r}) \approx x^{\ell_x} y^{\ell_y} z^{\ell_z} \sum_j^{N_g} a_j e^{b_j r^2}, \qquad (22)$$

where x_x^ℓ, y^{ℓ_y}, z^{ℓ_z} are polynomials, a_j and b_j are complex coefficients, and the number of Gaussian functions N_g is to be determined.

From the user's point of view, the WVL-PAW method is accessible with a few keywords: The keyword **usewvl** controls the choice of WVL or Plane Wave (PW) basis set. Since PAW atomic data are required in Gaussian form in Eq. (22), whenever the user provides with PAW atomic data in standard form, ABINIT converts them into Gaussian form via the PAW library. The number of Gaussians N_g is given by the user with the keyword **wvl_ngauss**. Typically, the more Gaussian functions are used, the more accurate the calculation is, and hence N_g is a convergence parameter. In the WVL-PAW method, the WVL basis is set by the same keywords as in the NC case: **wvl_crmult**, **wv_frmult** and **wvl_hgrid**, see the ABINIT web site for details. In addition, the BigDFT Poisson solver for free boundary conditions can now be used within plane waves and PAW, with the keyword **icoulomb**. This Poisson solver is useful for simulations of charged systems or surfaces.

Several automatic tests (bigdft#31 to #34) are provided and can serve as documentation. In particular, #31, #33 and #34 are simple examples of ground state calculations within WVL-PAW, and #32 tests the fitting of projectors to complex Gaussians. A more detailed explanation of the method and the PAW library is presented in Ref. [112].

4.2. Using high performance computers

The implementation of parallelism in ABINIT has made significant progress in recent years, and this on several fronts.

Several techniques have been implemented to accommodate the evolution of supercomputers and to help the user to make optimal choices to exploit computational resources. Part of this work has been supported by the "Partnership for Advanced Computing in Europe" (PRACE) consortium. Details about implementation and performance tests can be found on a dedicated website [113].

The code now makes extensive use of external libraries optimized for standard operations such as linear or matrix algebra and Fourier transforms. Wrappers to linear algebra routines have been introduced, which allow to automatically interface optimized versions such as Intel-MKL or architecture-dependent libraries such as MAGMA [114] (BLAS/LAPACK dedicated to GPU) or PLASMA [115] (BLAS/LAPACK for hybrid architectures). By default, when available, the ELPA library [116], optimized eigenvalue solvers, is now used. Substantial gains in pure performance were obtained. Similar work has been done with regard to the Fast Fourier Transforms. If software libraries such as FFTW or Intel-MKL are available in the environment, they are automatically used.

Part of the Ground-State computational workflow has been ported on NVidia Graphics Processing Unit (GPU). The most expensive code sections, namely the Hamiltonian application and the iterative diagonalization algorithm LOBPCG [39] have been adapted for use on GPUs. The basic operations use specialized external libraries: Cublas/cuFFT delivered with the CUDA package; MAGMA for matrix algebra operations; and a specialized kernel written in CUDA language has been developed for the application of the non-local operator. The resulting performances of ABINIT, when used on GPU, are similar to other DFT GPU implementations [117–119]: compared to a run on a single CPU, the GPU version of ABINIT (with as many GPUs as CPUs) is accelerated by a factor 3 to 5, depending on the computational load.

ABINIT can now benefit from hybrid architectures. The application of the Hamiltonian is fully compatible with a shared-memory parallelism (in addition to a distribution over MPI processes). The hybrid Fast Fourier Transforms are accessible through libraries like FFTW or Intel-MKL, or in a native version. The obtained performances depends strongly on the architecture (node size, shared memory performance...) but are significant especially when the number of threads remains below 8 [113].

Several additional parallelization levels have recently been added. It is now possible to perform calculations in the framework of non-collinear magnetism (including spin–orbit coupling or not) by distributing the computational load and the memory for each spinorial component of the wave-function. Computations using replicas of the simulation cell, such as for instance Minimal Energy Path determination or Path-Integral Molecular Dynamics (see 3.1) can benefit from a parallelization over the cell replicas. For such calculations, the number of potentially distributed processes becomes compatible with petascale (or higher) architectures.

In order to decrease the memory consumption, in the case of PAW calculations a distribution of the data structures over the atomic sites has also been introduced. The distribution of the computation of the "on site" PAW contributions is also automatically performed when the processors are available.

All the above-mentioned improvements are obviously effective, but make the use of ABINIT complex for the user. The latter should optimize the available computer resources and adapt them to the specificity of each simulation. To overcome this problem, a complete automation solution is now available for the ABINIT user, in the form of the **autoparal** input variable. In practice, the user does not need to provide any further parallelization keyword in the input file. The code starts in any case, using a process distribution determined by a set of simple rules. On demand, ABINIT can also perform a series of small benchmarks in order to refine the distribution of processes according to the system size and the computer architecture (hardware and software).

Three levels of automation can be used: (1) a simple heuristics based on empirical rules, (2) a series of benchmarks of the diagonalization/orthogonalization routines to refine the prediction of the simple heuristics, (3) a series of benchmarks to optimize the parameters of the Matrix Algebra library (number of processes, use of GPU).

Finally, the scalability of the inner core of Ground-State calculations has been significantly improved, as regards the "band" and "plane-wave" levels of parallelism. Indeed, for such parallelism, since 2008, ABINIT had been using the "Locally Optimal Block Preconditioned Conjugate Gradient" (LOBPCG) iterative diagonalization algorithm to determine the Hamiltonian eigenstates. This algorithm was limited in scalability on *petascale* architectures. It has been superceded by the Chebyshev filtering algorithm for cases involving more than a few hundred processors, as described in the next section Section 4.3.

4.3. The Chebyshev filtering algorithm for high performance computers

In the LOBPCG algorithm implemented in ABINIT, for a system in which N bands are requested, a $N \times N$ dense matrix has to be diagonalized at each step of the algorithm in order to provide an optimal subspace rotation (also known as the Rayleigh–Ritz step). Even using state-of-the-art linear algebra routines such as ELPA [116], the parallelization of a dense eigensystem of relatively modest size (typically, for large applications, a few thousands) involves communication between the processors, and stops scaling at about a hundred processors on most machines. This step quickly becomes the bottleneck, and the LOBPCG algorithm shows it limits when studying large systems which require thousands of processors for an acceptable time to solution.

To improve ABINIT's scaling into the thousands of processors, we have implemented the Chebyshev filtering algorithm [40]. This algorithm, mathematically classified as Chebyshev-accelerated subspace iteration, dates back to the 1970s [120], and has been applied to electronic structure computations in [121,122]. For an eigenvalue problem $Ax = \lambda x$, the algorithm applies a polynomial $p(A)$ to a set of basis vectors $(v_i)_{i=1,...,N}$ and uses a Rayleigh–Ritz procedure on the subspace spanned by $(p(A)v_i)_{i=1,...,N}$. This iteration is repeated until convergence. The polynomial p is a suitably shifted and rescaled Chebyshev polynomial, chosen to amplify the desired components of the spectrum, and damp the unwanted ones.

While inferior in terms of convergence rate to LOBPCG or block Davidson schemes (it uses a Rayleigh–Ritz procedure on a smaller subspace, and is unable to use preconditioning), it requires smaller and less frequent Rayleigh-Ritz steps, as well as orthogonalizations. For large systems, the added scalability greatly offsets the slower convergence rate (Fig. 1).

To implement the Chebyshev filtering algorithm in ABINIT, we kept the same two-level parallelization already used to implement the LOBPCG algorithm (bands, plane waves). The main difference is that, in the PAW framework, the eigenvalue problem is $H\psi = \lambda S\psi$, and adapting the Chebyshev filtering method requires the application of the matrix $S^{-1}H$. This can be computed efficiently because the PAW overlap matrix S is a small-rank perturbation of the identity, and the Woodbury matrix identity reduces the inversion of S to that of a much smaller matrix.

This new scheme can be activated using **wfoptalg 1**. Compared to the LOBPCG algorithm, care has to be taken to include enough extra bands, to help the convergence of the algorithm (a wise tactic even when using LOBPCG, especially if the convergence threshold on the wavefunctions **tolwfr** is used). A number of "buffer" bands, that should not be considered as part of the termination criterion, can be specified with the `nbdbuf` input variable. The rest of the
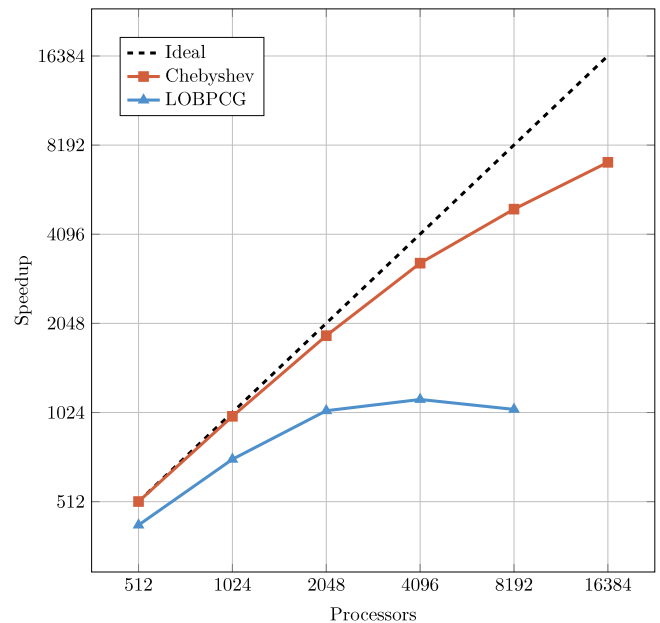

Fig. 1. Performance of the new "Chebyshev Filtering" algorithm compared to the previous LOBPCG algorithm. The scaling is strongly improved, and is not saturated even at over 16 kcores.

input file should follow the same guidelines as with the LOBPCG algorithm. Advice for optimal performance can be found in the ABINIT distribution, in the `doc/theory/howto_chebfi.pdf` file. For more details about the theory, implementation and performance of the Chebyshev algorithm in ABINIT, the reader is referred to [40].

4.4. Build system

The ABINIT Community and the context in which ABINIT is used are permanently evolving. Over the last few years, the major trends have been a growing use of ABINIT on massively parallel architectures, an increasing number of contributions to development, as well as the refinement of the development workflows and lifecycles. When it comes to preserve the usability, portability and interoperability of the code in this challenging environment, the build system stands on the very front lines and must provide a solid basis on which solutions to the corresponding issues can be developed.

Having to deal regularly with inexperienced new contributors is a common issue faced by growing projects. Documentation and tutorials can partially address it but involve in turn a growing amount of time and energy invested in the training and monitoring of newcomers. At some point, they have to be complemented by an increasing automation of some components. This is precisely what has been done for the build system of ABINIT, by providing developers with an interface based on config files, so that they do not have to modify the build system themselves. Instead, the latter rewrites itself as many times as necessary, and always remains consistent with the config files. To further simplify the process, the whole configuration is described using the Python ConfigParser format. At present, developers can perform various operations without knowing the internals of the build system, such as:

- including new files and directories into the build process in a flexible way;
- declaring new programs and their dependencies, both internal and external;

- defining options to the configure script, imposing constraints on their permitted values, and including the definition of custom preprocessing options;
- defining environment variables to be tracked by the build system;
- informing users about the latest changes in the options of the configure script;
- declaring custom shell commands and output files that the configure script will respectively execute and create;
- asking the build system to dump internal variables for debugging purposes;
- having example configurations automatically synchronized with the latest build template;
- tune parameters of the test farm;
- tune the behavior of key components of the build system.

This automation has also permitted the creation of a dedicated test suite for the build system, a missing component which had become critically necessary. It has also been made possible by the simultaneous enhancement of the ABINIT Test Farm. The tests we have developed so far enforce the consistency between the configuration files of the build system and the source code, in addition to ensuring that the test farm performs the actual tasks configured in the build system (doing the things right). In the future, we will include tests checking that the configured tasks are effectively the ones that should be executed (doing the right things).

An important challenge arising from the ever expanding variety of the contributions is to efficiently manage a growing number of external dependencies while respecting all the kinds of contributors involved: users, occasional developers, regular developers, maintainers, project managers, infrastructure managers, and external computer experts. Furthermore, those external dependencies can depend on each other, such as the ETSF_IO library, which relies on NetCDF, or Wannier90, which requires linear algebra. For these reasons, we have decided to reorganize the former set of "plugins" into a fully categorized ***connectors + fallbacks* mechanism**. External dependencies have thus been grouped by topic and labeled accordingly:

- TIMER: timing and benchmarking systems (ABINIT and PAPI);
- LINALG: linear algebra libraries (serial and parallel variants);
- ALGO: algorithmic enhancements (Levenberg–Marquardt);
- FFT: Fast Fourier Transform libraries (FFTW, MKL, and Goedecker);
- MATH: mathematical libraries (GSL and HP$^®$ Mlib);
- TRIO: TRansferable Input/Output libraries (FoX, LibYAML, NetCDF, and ETSF_IO);
- DFT: Density Functional Theory libraries (BigDFT, LibXC, and Wannier90).

The core build-system components have been restructured to provide all the relevant information to the connectors, now in charge of detecting external dependencies and adjusting the build parameters accordingly. The connectors take their decisions based on what they know about: the programming languages available and their peculiarities; architectural enhancements, such as the available levels of parallelism; or the presence of a benchmarking system. Fig. 2 describes the dependencies between the core build-system components, each of them being built consecutively. At configure-time, the detection process thus goes from bottom to top, leaving the connectors for the end of the process, so that they take the best possible informed decisions. When they want to enable particular features, users can provide useful hints to the build system through several options of the configure script. Each item of the above list has a corresponding set of -- *with-** options to pass the information, for example:
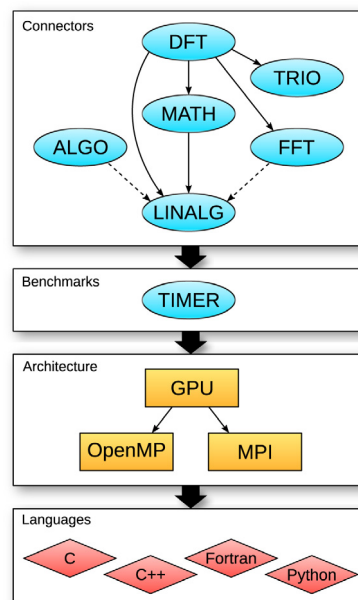

Fig. 2. Core components of the build system of ABINIT, from the most dependent (top) to the least (bottom). Each arrow in the diagram stands for "depends on". Dashed lines indicate indirect dependencies, *e.g.*, FFT routines indirectly depends on LINALG as they can be provided by the same library, for instance with Intel$^®$ MKL.

```
../configure \
  --with-dft-flavor="libxc" \
  --with-libxc-incs="-I/usr/local/include" \
  --with-libxc-libs="-L/usr/local/lib -lxcf90
  -lxc"
```

Since ABINIT is run in a wide variety of computing environments where installing software packages can sometimes be more complex, its source code is distributed with a fallback mechanism to provide external dependencies needed by the connectors. The *fallbacks/* subdirectory contains a meta-build-system, fully independent from ABINIT, able to download, build and install different stand-alone packages in a consistent way. It has been designed to let developers easily debug the most advanced ABINIT features even on exotic computer architectures, as well as test new versions of external dependencies and fix issues before updating the official list of ABINIT requirements. When the build system of ABINIT is unable to detect external dependencies, it can thus build fallback versions instead of stopping with an error, as described in Fig. 3. At present, the following fallbacks are available: BigDFT (wavelets for DFT), ETSF_IO (implementation of the ETSF File Format specifications), FoX (Fortran XML parser), LibXC (exchange–correlation functionals), NetCDF (platform-independent I/O), Netlib's BLAS and LAPACK, and Wannier90 (Maximally-Localized Wannier Functions). In addition, they provide a version of AtomPAW (PAW dataset generation) tested with the current version of Abinit.

4.5. Code beautification

Periodically, long term maintenance actions are performed by a subset of developers, in a concentrated time frame before release of a new version. Users are impacted either directly (documentation of input variables, tutorial cleaning...) or indirectly (source code documentation or pruning, build system simplification...). We refer to these as "beautifications" of ABINIT. As a first example of such "beautification" action, before the release of ABINITv7.10.5, we have activated a coverage check concerning the use of each of the source files, and even the use of individual lines of code, thanks to the software tool "gcov" (code covering). As a consequence,
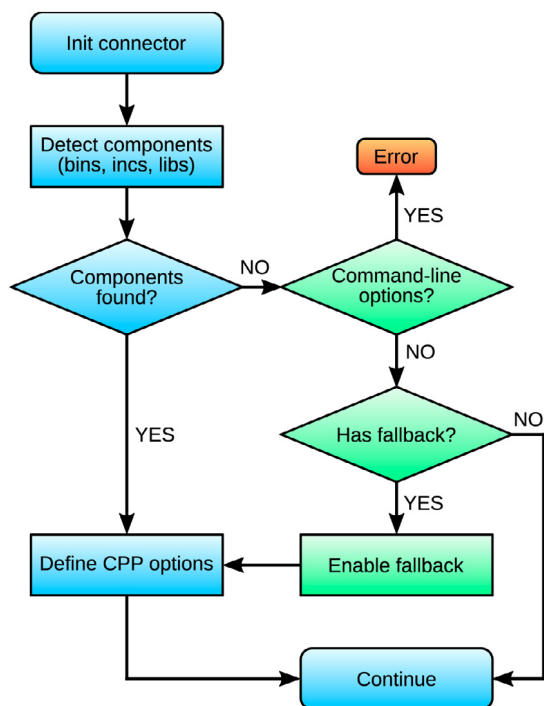
Fig. 3. Flowchart representing the internals of an ABINIT connector. Steps on the main path are colored in blue, while the fallback mechanism is shown in green. If the configure script has received user-specified options for the connector and is not able to find the corresponding external dependency, it will stop with an error. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

several sections of codes that were not tested by any test of the automatic suite have been detected. On one hand, a set of obsolete subroutines has been identified and removed. On the other hand, tests have been added to cover useful, but previously untested, sections of code. As another "beautification" operation, the documentation related to each of the input variables has been placed in a global database (instead of being present only in loosely structured html files). From this database, the html pages are automatically created, and automatic validation tools are now available (for instance in AbinitGUI, see Section 5.4.3). Other "beautification" operations have been carried out, in order to remove obsolete Fortran77 syntax occurrences (as detected by recent compilers) throughout the entire source code, or to use more systematically F2003 features. As mentioned earlier, there is a long-standing effort to modularize the ABINIT code, and to use high-level IO libraries, which eases the set up and interfacing of pre- and post-processing tools, and conversion tools.

5. Other developments in the ABINIT project

5.1. Documentation, tutorials and tests

The documentation of ABINIT is multifaceted. The source of most documentation files can be found in `doc/`, and most are processed and presented on the ABINIT Web site. One can distinguish three structured sets of information that guide the user: the new user guide and help files, the tutorial, and the description of input variables, with respective entry points on the Web

www.abinit.org/doc/helpfiles/for-v7.10/<X>,

with <X> being `users/new_user_guide.html`, `tutorial/welcome.html`, and `input_variables/keyhr.html`. In the package, they are located in `doc/users`, `doc/tutorial`, `doc/input_variables`, respectively. The `doc/` directory also

contains the installation notes, release notes, known problems, as well as some specific documents on the theory underlying ABINIT, information on the build configuration files, and other miscellaneous information for the use of ABINIT that are not present on the Web.

While the relatively low-level user guide and help files have only seen minor changes with respect to ABINITv5.7, the tutorial has been largely expanded, with twelve new lessons (for a total of thirty-three). Most of them are present in `doc/tutorial`. A flowchart representing the dependencies between the tutorials of ABINIT is presented Fig. 4.

Six new lessons, mentioned in the rightmost column of Fig. 4, are related to the use of parallel features of ABINIT

(`lesson_paral_gspw` - Parallel Ground state, `lesson_paral_moldyn` - MD, `lesson_paral_string` - Images, `lesson_paral_gswvl` - Parallel Ground state wavelets,

`lesson_paral_dfpt` - Parallel Response Functions,

`lesson_paral_mbt` - Parallel Many Body Perturb. Theory), and were demonstrated at a training school [123], where the participants could have access to more than one thousand computing cores for their tests. Other lessons focus on the BSE formalism, on dynamical mean field theory, on nuclear properties, on the comparison between PAW and all-electron calculations, and on the determination of the U value in the constrained LDA formalism (`lesson_bse.html`, `lesson_dmft.html`, `lesson_nuc.html`, `lesson_paw3.html`, `lesson_udet.html`). At variance with all the other lessons, the new lesson on the computation of the temperature-dependent electronic structure (including zero-point motion renormalization) is available only on the ABINIT Wiki [124].

The other lessons are available as well on the Wiki, although, at present, their official source is in the package or on the official ABINIT Web site.

Independently of the `doc/` directory, there is also a large set of example input files, with their reference output files, contained in `tests/*/Input` and `tests/*/Refs`. These are a very useful counterpart to the input variable description file, as they illustrate the use of every ABINIT input variable, at least in one case for each input variable. Between ABINITv5.10.8 and ABINITv7.10.5, the number of input files provided in `tests/*/Input` has increased from about 600 to about 1000.

Another noticeable source of documentation on ABINIT consists in the slides that are available on the Web, having been prepared for different schools or hands-on events. They are listed in http://www.abinit.org/community/events. As a recent side effort that aimed at investigating new ways to provide documentation for ABINIT, videos for the basic steps of ABINIT usage have been set up on YouTube, see www.youtube.com/<X>:

- How to install on Linux (<X> is `watch?v=DppLQ-KQA68`);
- How to install on Windows (<X> is `watch?v=EfJcYi1MNBg`);
- The first tutorial on H2 molecules

  (<X> is `watch?v=gcbfb_Mteo4`).

5.2. The test farm

The reference input and output files play an important role in securing the capabilities of ABINIT. Before a new development is included in the main version of ABINIT, there is a careful check that it does not destroy the existing ABINIT capabilities. At the time of ABINITv5.7, this was checked thanks to scripts, but the launch and analysis on different machines was done manually. Shortly after ABINITv5.7, a test farm was set up with appropriate software, and the whole checking procedure is now completely automatic.
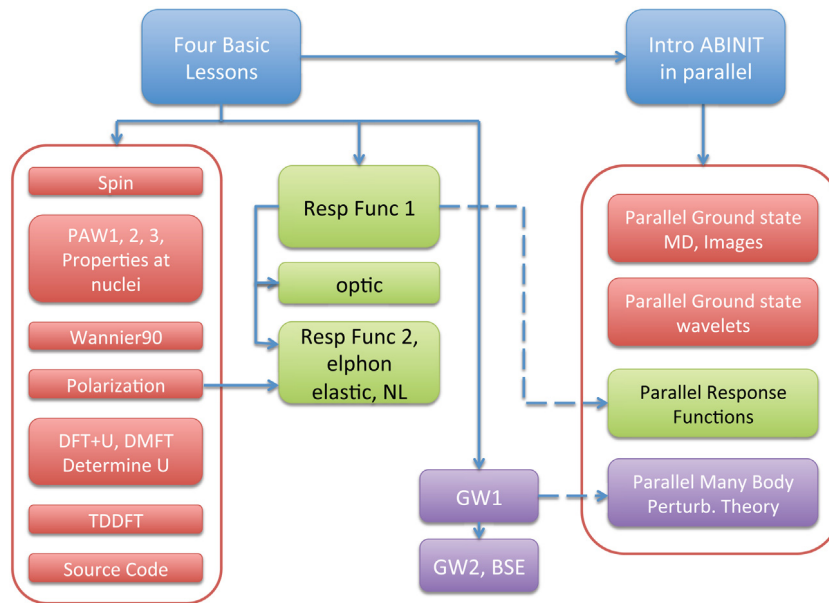
Fig. 4. Flowchart representing the dependencies between the tutorials of ABINIT.

Table 1
Several characteristics of the "builders" on which ABINIT v7.10.5 has been daily tested until September 2015.

Builder label	Compiler	Parallel support	Numerical libraries	Other characteristics
testf_gcc47	GNU 4.7.1	openmpi 1.6.1		
testf_gcc47_serial	GNU 4.7.1			Parallelism disabled
bigmac_gcc48_sl	GNU 4.8.2	mpich3 3.1.1 scalapack 2.0.2	atlas 3.10.1 fftw3 3.3.4	Packages installed with MacPorts
buda_gcc47_cuda	GNU 4.7.4	openmpi 1.6.5	mkl 10.3 magma 1.5.0	GPU (cuda 6.5) K40 NVIDIA card
buda_intel14	Intel 14.0.4	mpich3 3.1.3	mkl 11.0 fftw3	Compiler options -O3 -xhost
coba2_intel13	Intel 13.1.1	openmpi 1.6.5	mkl 11.0 fftw3	CentOS
cronos_gcc46_paral	GNU 4.6.3	openmpi 1.4.5	mkl 10.3 fftw3	Whole test suite executed on 2 procs
ibm6_xlf14	IBMxlf 14.1	poe		OS: AIX6.1
inca_gcc44_sdebug	GNU 4.4.7	mpich3 3.1.1	atlas 3.10.1	
ktulu_gcc48	GNU 4.8.2	openmpi 1.6.5	atlas 3.10.1 fftw3 3.3.3	Test Ubuntu 14.0.4 packages
max2_gcc46	GNU 4.6.2	mpich2 1.4.svn	acml 4.4.0	Memory profiling
petrus_nag	NAG 5.3.1	openmpi 1.6.5		openSUSE 12.1
shiva_gcc47_distchk	GNU 4.7.4	mpich2 1.3	openBLAS	Make distcheck
tikal_intel12_serial	Intel 12.1.4		mkl 10.3 fftw3	Parallelism disabled
tikal_gcc49	GNU 4.9.2	mpich3 3.1	mkl 11.1 fftw3	
toum_gcc46_openmp	GNU 4.6.3			2 OpenMP threads Fedora
vishnu_gcc47_nofbmisc	GNU 4.7.4	openmpi 1.6.5		
woopy_gcc48	GNU 4.8.3	mpich3 3.0.4		
yquem_g95	g95 0.94	openmpi 1.6.3		Tests with 1 proc only

This milestone for the ABINIT developer community is described in detail in several sections of Ref. [125]. The test farm infrastructure has evolved since that time, but not its philosophy. So, we suggest the interested reader to consult Ref. [125], instead of rephrasing it here. By contrast, a list of the architectures and compilers, on which ABINIT is daily tested, might be very useful for the user and is shown in Table 1. Note that older compilers which do not support fortran2003 are not able to compile the latest versions of ABINIT.

At variance with the Intel, GNU, G95, NAG and IBM compilers, for which the test suite executes correctly (at least for some recent or not-so-recent release(s) of these compilers), too many specific problems were encountered with the PGI, Pathscale and Open64 compilers. As a consequence, they are not officially supported, and no machine is dedicated to tests based on the executables that they generate. This does not mean that ABINIT will not work with some version of these compilers, but there is no guarantee ABINIT will compile and/or run correctly with them. Table 1 applies to

production version 7.10.5, while development versions of ABINIT have also been tested successfully with the GNU 5.1 and 5.2 compilers, as well as with the Intel 15 compilers.

In addition to the input files and corresponding reference files that test ABINIT computational capabilities, the test suite also includes memory checks and profiling (yquem_g95), as well as checks on the compilation (*e.g.*, full debugging options for inca_gcc44_sdebug and tikal_gcc49), build (*e.g.*, woopy_gcc48 relies on external libraries only) generation of package (shiva gcc47 distchk) and even the coherence of the documentation of ABINIT.

5.3. Atomic data files

The Norm-Conserving pseudopotentials (NCPP) and PAW methods are both available in ABINIT. The PAW method allows lower plane wave energy cutoffs and reproduces by construction all the nodes of the valence wavefunctions. It is therefore particularly suited to the study of localized electrons. That is why the local exact exchange, the LDA + U and DMFT approaches are available in ABINIT in the PAW framework only. Most of the time, new features are developed first in the NCPP framework, and are extended later to the PAW one because of the more complex coding required in the PAW framework. Some of the most recent features in ABINIT are therefore available only in the NCCP case, as mentioned in the text hereabove.

5.3.1. Norm-conserving pseudopotentials

At the time of ABINIT v5.8, the ABINIT project provided several NCCP tables [126,127] on the official website, but the majority of these tables had been generated without relying on optimization techniques, such as the RRKJ method [128], and, most importantly, no systematic validation of these tables had ever been performed. Last but not least, most of these legacy NCPPs employed only one projector per angular channel, hence it was difficult to find NCPPs with semi-core states treated as valence. For this reason, we have generated a new table of NCPPs based on the multiple-projector formalism recently proposed in Ref. [129]. These new pseudopotentials are usually softer than traditional NCPPs and more accurate, especially when semi-core states are included in the valence, since the additional projector allows one to employ the neutral atomic ground state as a reference configuration, instead of an ionized atom.

This new table (ONCVPSP-PBE-PDv0.2) has been validated against all-electron calculations using the "Δ" estimator proposed in Refs. [130,131] ($\Delta = 0.7$ meV, $\Delta_1 = 1.6$ meV) as well as the GBRV benchmark set [132]. Moreover the convergence behavior of phonon modes within DFPT has been tested. A summary of the results of the validation tests is given in Fig. 5. The input files as well as the pseudopotential files are available on the ABINIT website, as well as on the website [133]. The python scripts used to validate the table are available on the PseudoDojo repository hosted on Gitlab [134].

5.3.2. PAW atomic datasets

In the Projector Augmented Wave (PAW) method developed by Blöchl (1994) [36], a PAW data file is needed for each element, which replaces the pseudopotential file used in the NCCP or USPP formalisms. We have developed a new set of PAW atomic data files for most of the stable elements in the periodic table (the JTH table [131]). They have been generated thanks to the ATOMPAW generator [38]. These files are provided in a standard XML format for use in any PAW electronic structure code, following the specifications given in Ref. [135]. The new datasets are available on the ABINIT web site [136] for the first 86 elements. They perform well as measured by the "Δ" evaluation criterion introduced by
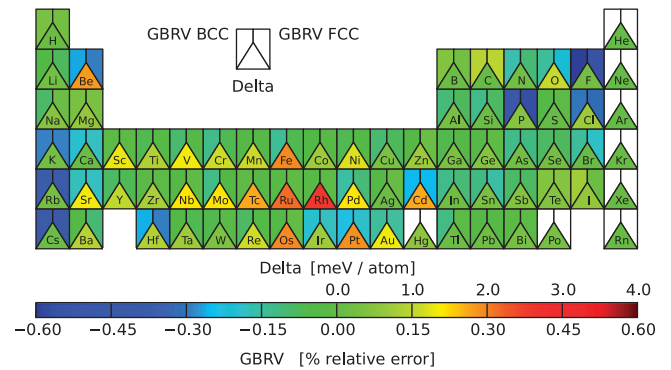

Fig. 5. Values of the Δ factor and relative error for the lattice constants of the bcc/fcc elemental solids included in the all-electron GBRV set for the ONCVPSP norm-conserving pseudopotentials. White rectangles (triangles) denote entries that are missing in the benchmark set. The mean absolute relative error for the tests are 0.12 and 0.13 for fcc and bcc, respectively.
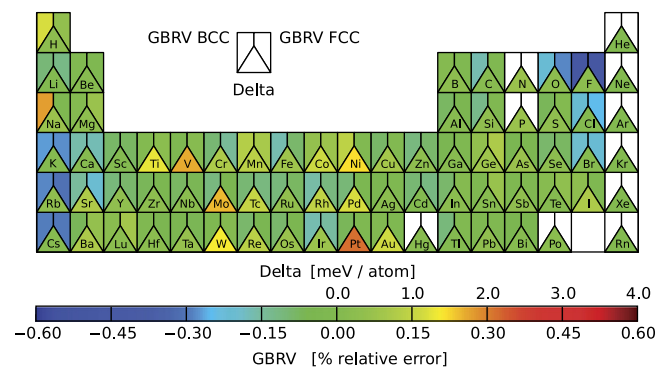

Fig. 6. Values of the Δ factor and relative error for the lattice constants of the bcc/fcc elemental solids included in the all-electron GBRV set for the JTHv1.0 PAW atomic data. White rectangles (triangles) denote entries that are missing in the benchmark set. The mean absolute relative error for the tests are 0.13 and 0.14 for fcc and bcc, respectively.

Lejaeghere *et al.* [130] (periodic table average $\Delta = 0.4$ meV), and also perform well in a modified evaluation scheme proposed in Ref.[131] (periodic table average $\Delta_1 = 1.0$ meV). The GBRV benchmark set [132] gives also very good results for the JTH table, as can be seen on Fig. 6.

5.4. Post-processing, GUIs, and scripting

5.4.1. AbiPy

The AbiPy software is a python package that provides a 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 massively parallel architectures. AbiPy is built on top of pymatgen [137], an open-source Python library that provides powerful tools to analyze large quantities of materials information [138] and a high-level interface to the Materials Project database [139]. The AbiPy package also contains several graphical interfaces that significantly simplify the repetitive work needed for extracting and visualizing the results. Fig. 7 shows a screen shot of the different capabilities. All the software components are hosted on Gitlab [140], a social coding platform that allows users to report bugs and contribute to development.

5.4.2. ABINIT Post Process Application (APPA)

Output files produced by large scale Molecular Dynamics (MD) simulations are not easy to handle. Such trajectories are stored in
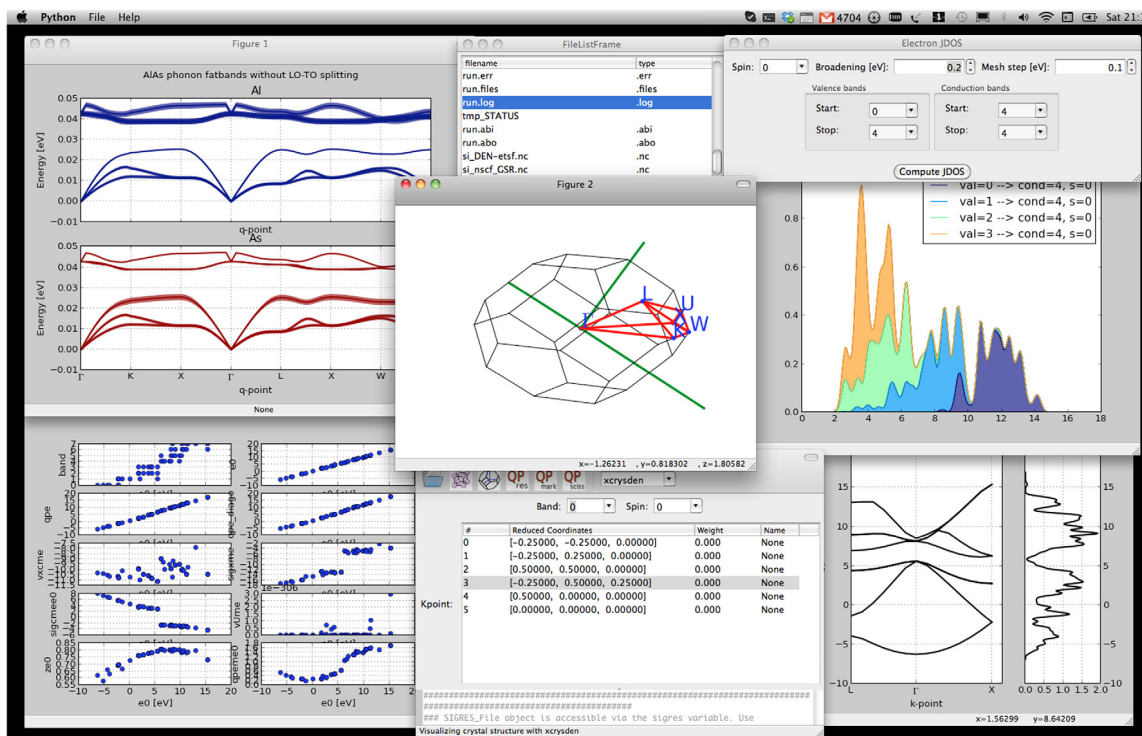
Fig. 7. Some of the graphical user interfaces provided by AbiPy for the visualization and the post-processing of ABINIT calculations. Phonon and electron band structures, GW eigenenergy corrections, a 3D Brillouin zone, and a transition decomposed optical spectrum are shown.

a file named HIST (a *NetCDF* file) from which it may be necessary to extract data in text format. The goal of the "APPA" utility is to allow statistical analysis of MD trajectories through an ergonomic graphical user interface.

APPA is written in Python, using graphical packages such as *PyQt* or *matplotlib*. Some parts of this tool use Fortran — thanks to the *F2PY* Python package, to handle large amounts of data more efficiently and to conserve numerical accuracy in post-processing. The APPA interface is intuitive and documented (tooltips) and allows the computation of variations or averages of several properties: total energy, pressure, temperature, velocity autocorrelation function, radial pair distribution, vibrational density of state, *etc.* APPA can manage trajectories obtained from normal Born Oppenheimer MD simulations or from Path-Integral Moleculars Dynamics (see Section 3.1).

APPA is delivered in the ABINIT package in the `scripts/post_processing/appa` directory. The user simply has to launch the `appa.py` script to access to the GUI. A command line version `appa_nw.py` is also available, to be used in scripts. Fig. 8 shows a screen shot of the different capabilities.

5.4.3. AbinitGUI: a simple graphic user interface for ABINIT

Running ABINIT simulations, as in other ab-initio packages, requires to manually process many tasks ranging from the writing of an input file to the analysis of the results. Many of these tasks are launched on remote computing centers (clusters), and are repetitive and time consuming. AbinitGUI has been developed to help users run calculations in a more user-friendly environment and to automate many repetitive tasks.

AbinitGUI allows one to launch simulations either locally or remotely. One of its handy features is to allow users to save several cluster configurations to send their simulation jobs. Furthermore, the GUI provides a complete SSH connection facility as well as a SSH terminal and a graphical SFTP client. A minimalistic interface is also provided to monitor the jobs (get current status, kill a specific job, . . .).

During the pre-processing step, AbinitGUI allows one to write the input file with a built-in editor, but also to test and analyze it thanks to an automatic parser, generated from structured documentation written in the YAML format. The geometric structure associated with each dataset can then be viewed in an embedded Jmol panel [141]. As a second step, the user edits the configuration for the specific remote job management system and launches the job. All the required files are automatically generated and submitted to the remote system.

Finally, the user can analyze the results through a set of post-processing scripts that are shipped with the GUI. The scripts are, for instance, band structure plotters, or links to the Python Library "Abipy" (see Section 5.4.1).

This project is developed in Java to be cross-platform and therefore does not require any installation. As the size of the package grew with time, the developers decided to extract the development from the main ABINIT package and hosted the package on Github [142]. A dedicated website has also been designed, and contains the latest stable version as well as documentation and supplementary information [143].

6. On-going developments

6.1. Unfolding supercell band structures

Supercells are often used in *ab initio* calculations to model compound alloys, surfaces and defects. Band structure plots obtained from supercell calculations are difficult to interpret and compare to the reference band structure of a corresponding primitive crystallographic unit due to the zone folding (see the band structure of a Si supercell in Fig. 9(a)). The purpose of `fold2Bloch` is to assist with interpretation of supercell electronic structure calculations by recovering the Bloch character of electronic eigenstates perturbed by disorder. In general, there is no single wave-vector **k** associated with a particular eigenvalue
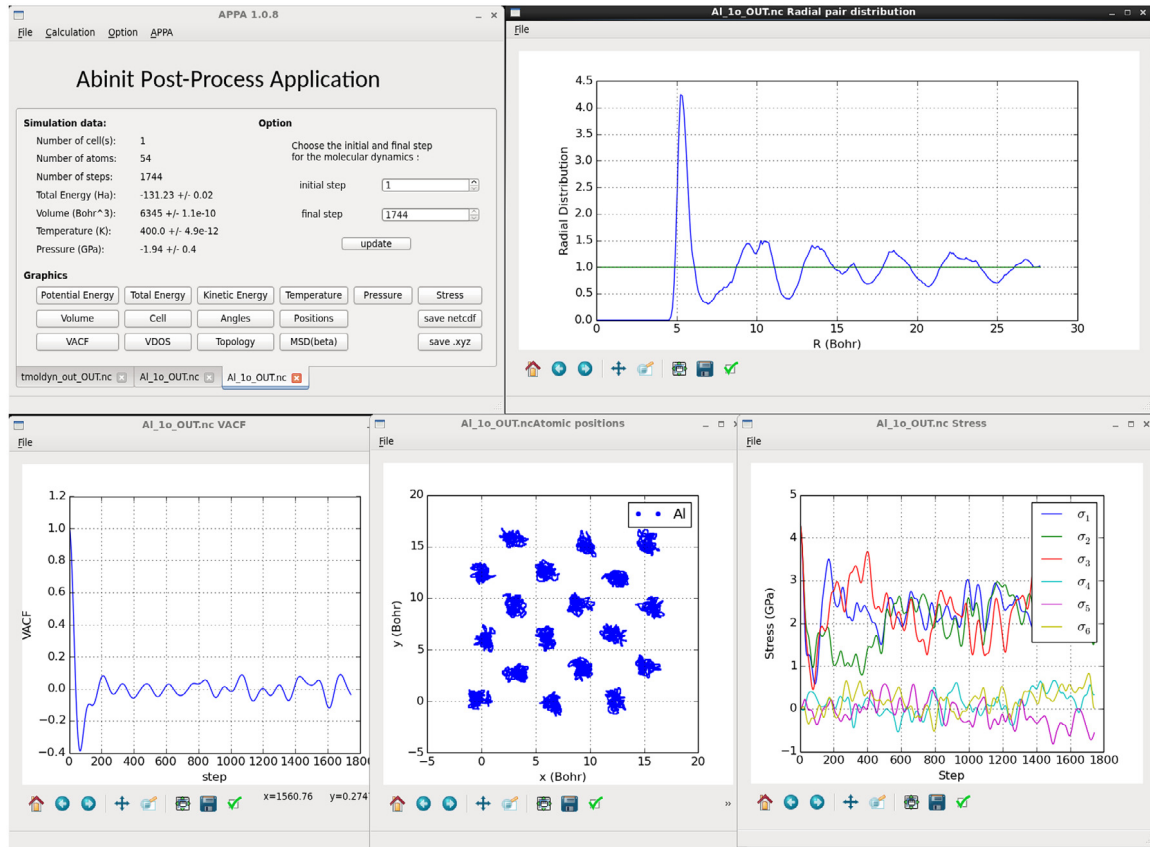
Fig. 8. Some of the graphical user interfaces provided by APPA for the visualization and the post-processing of ABINIT MD runs. A radial pair distribution function, velocity autocorrelation function, trajectory visualization, and the variation of the components of the total energy are shown.
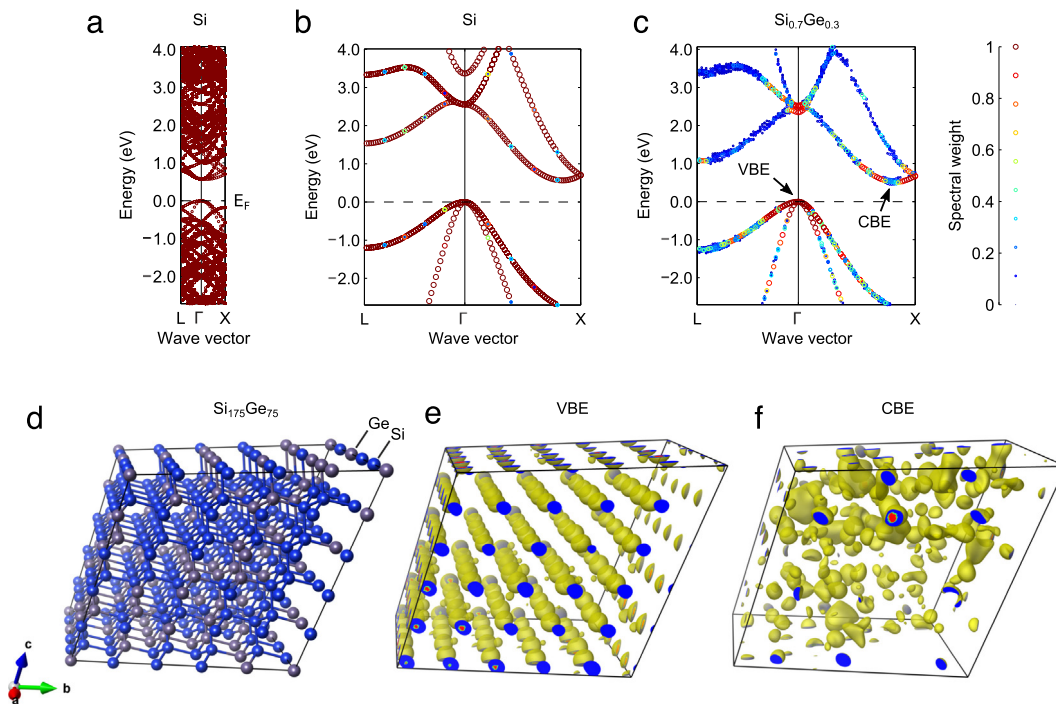

Fig. 9. Folded (a) and effective (b, c) band structure of 250-atoms supercell unfolded to the primitive Brillouin zone: pure Si (a, b) and random $Si_{0.7}Ge_{0.3}$ alloy (d). The Bloch character in $Si_{0.7}Ge_{0.3}$ alloy is well preserved at the top of the valence band (VBE), whereas the electronic states at the bottom of the conduction band (CBE) are more disordered. The bottom row illustrates $Si_{175}Ge_{75}$ supercell (d) and the Kohn–Sham orbitals $|\psi_{n,\mathbf{K}}(\mathbf{r})|^2$ that correspond to the top of the valence band (e) and the bottom of the conduction band (f). The structure and isosurface plots were created with the help of the ABINIT cut3d utility and VESTA 3 package [150]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

E, but rather a distribution of k's. This distribution is also known as a Bloch spectral density or spectral weight. The spectral weight $w_n(\mathbf{k})$ amounts to a Bloch $\mathbf{k}$-character of the n'th energy eigenstates ϵ_n and fulfills the normalization $\sum_{\mathbf{k}} w_n(\mathbf{k}) = 1$. In the case of a non-local basis set, such as plane waves, the spectral weight can be constructed from the Fourier expansion coefficients by gathering them in groups associated with a particular Bloch wave vector in the primitive cell [144–148]

$$w_n(\mathbf{k}) = \sum_{\mathbf{G}} |C_{n,\mathbf{K}}(\mathbf{G})|^2 \quad \text{for all } \mathbf{G} \text{ that fulfill } \mathbf{K} + \mathbf{G} = \mathbf{k} + \mathbf{g}, \quad (23)$$

where $C_{n,\mathbf{K}}(\mathbf{G})$ refers to plane wave coefficients for *n*th eigenstate with the wave vector $\mathbf{K}$ within the first Brillouin zone (BZ) of the supercell, $\mathbf{G}$ and $\mathbf{g}$ are the reciprocal lattice vectors of the supercell and primitive cell, respectively. More technical details on this unfolding procedure are available in Ref. [149]. It should be noted that the primitive and supercell reciprocal lattices should be commensurate as required by Eq. (23). This condition implies that a supercell needs to be generated by translation of the primitive cell along its lattice vectors in real space using a desired multiplicity $N_x \times N_y \times N_z$. Care should be taken when working with lattices whose conventional unit cell is different from the primitive cell (e.g., zinc-blende or rock salt structures).

A prerequisite for unfolding is the presence of wave function files (e.g., `abo_WFK`) that can be generated in a way similar to standard band structure calculations. The only difference from the conventional band structure calculation is the k-path selection approach. For instance, when aiming for the band structure plot with $\Gamma(0, 0, 0) - Z(0, 0, 1/2)$ path in the primitive Brillouin zone (BZ), the **kptbounds** in supercell BZ should include $\bar{Z}(0, 0, -1/2) - \Gamma(0, 0, 0) - Z(0, 0, 1/2)$. At first, the part $\bar{Z} - \Gamma$ seems redundant lying outside of the range of interest. However, it is easy to show that those wave vectors contribute to the range Γ–Z after unfolding. For instance, let us consider a k-point $(0, 0, -K)$ in the $1 \times 1 \times 3$ supercell. This point "unfolds" into 3 k-points in the primitive BZ: $(0, 0, (-K - 1)/3)$, $(0, 0, -K/3)$, and $(0, 0, (-K + 1)/3)$. The first two points are irrelevant for the purpose of the plot as they do not belong to the Γ–Z segment, but the last point falls into the range of interest and needs to be included.

Once the wave function file is prepared, the unfolding proceeds by invoking `fold2Bloch abo_WFK` N_x:N_y:N_z that generates an output file `abo_WFK.f2b`. The output file contains 5 columns: the list of unfolded wave vectors (k_x, k_y, k_z) in the primitive BZ, the corresponding eigenvalue ϵ_n (Ha) and the Bloch spectral weight w_n.

A MATLAB script `ubs_dots.m` is designed to assist with plotting the unfolded band structure. The unfolded band structure plot is represented as a scatter plot where the size and color of the markers carry information about the Bloch character of energy bands (see the unfolded band structure of Si in Fig. 9(b)). Users are asked to provide details about the k-path, name of the `fold2Bloch` output file `abo_WFK.f2b`, position of the Fermi energy, and reciprocal lattice vectors that can be found in the main output file `ab_out.out`.

In order to illustrate the performance of `fold2Bloch`, an effective band structure of a $Si_{0.7}Ge_{0.3}$ (an alloy extensively used in thermoelectric applications) is calculated (Fig. 9). The unfolded band structure of $Si_{0.7}Ge_{0.3}$ retains main features of the Si electronic structure, however not all states preserve their Bloch character. The valence band edge at Γ-point retains 90% of the Bloch character that reflects in the periodicity of Kohn–Sham orbitals. Contrary to the valence band, the conduction band edge shows signatures of disorder, namely "amorphous", non-periodic electronic states with the Bloch spectral weight of less than 50%.

This analysis allows us to conclude that the compositional disorder will result in diminished mobility of electrons, rather than holes.

Analysis of the effective band structure of alloys can also be linked to their optical properties. For instance, it is possible to observe whether optical transitions, which are forbidden in the host material by the crystal momentum conservation, can become allowed in the presence of perturbations due to alloy elements.

6.2. Interpolation technique for BSE

An accurate description of dielectric properties within the BSE formalism presented in Section 3.5.1 usually requires the sampling of a large number of k-points in the Brillouin Zone. The large computational cost associated to this dense sampling renders well-converged BSE calculations prohibitive. This is especially true when derived properties e.g. Raman spectra are wanted [76].

Different numerical techniques have been proposed in the literature to improve the convergence rate. For example, Rohlfing & Louie [151] developed a double-grid technique in which the matrix elements of the BSE Hamiltonian evaluated on a coarse grid are then interpolated towards a denser grid. This scheme accelerates the convergence of the dielectric function with a computational cost that is significantly reduced with respect to a full calculation. In the spirit of this technique, we have generalized the Rohlfing & Louie approach allowing for multi-linear interpolation and we have designed efficient algorithms based on iterative Lanczos–Haydock algorithms. Different schemes with different degrees of accuracy and computational load are available in ABINIT. The interested reader is referred to Ref. [152] for a more detailed description of the interpolation schemes including benchmarks and tests on prototype semi-conductors.

These new interpolation schemes are available in ABINITv8 with the **bs_interp_*** input variables and examples of input files are available in v67mbpt#30-#35.

6.3. Hybrid functionals

In the 2009 ABINIT article [8], the local PBE0 approximation was presented: it was an application of exact exchange functional to a restricted subspace formed by the correlated electrons of a correlated system. A more general implementation of hybrid functionals has been carried out in ABINIT over the last two years. In the NCPP framework, the Fock operator has been coded. It is now available to perform a Hartree–Fock calculation (**ixc** = 40) or a PBE0 one (**ixc** = 41, 42). Only the total energy is available for the moment. A well converged LDA/GGA calculation must be performed first, to obtain good starting wavefunctions, before adding the Fock operator. Two tests are provided in v7#65 and #66. The **nphf** keyword allows for a local parallelization over k-points for the calculation of the Fock operator.

In the PAW context, the Fock operator has been coded following the work of Rostgaard [153] and will be available in ABINIT version 8. The exchange energy is divided in 3 terms. The first one is constant and comes from the interactions between core orbitals; the second one comes from the interactions between core and valence orbitals. It is calculated thanks to pre-computed PAW data that are provided in the PAW atomic datasets; the third one comes from interactions between valence electrons. This last term has a component coming from the charge compensation density, which is calculated "on the fly", and a component coming from on-site interactions that is already pre-calculated in the code. The implementation is now being extended to range-separated functionals.

6.4. Van der Waals interactions

It is well known that long range correlations responsible of van der Waals interactions are out of reach for both LDA and GGA approximations to the exchange–correlation energy in DFT. In recent years several methods have been devised to include such interactions, which can be grouped into two strategies, namely *ad hoc* methods and self-consistent approaches. Currently ABINIT can perform calculations based on either the DFT-D methods or the vdW-WF methods, as described later, both belonging to the first group.

A fully customizable implementation of the vdW-DF method [154], a self-consistent approach, and an adaptation of the strategy followed by G. Román-Pérez et al. [155] to the case of ABINIT are under development. It will offer around 25 adjustable parameters and be delivered with graphical tools to help users assess the quality of their kernels. It does not only aim at performing production calculations with vdW-DF, but also at helping researchers who develop new density functionals optimized for systems requiring van-der-Waals interactions.

The DFT-D methods have been implemented inside ABINIT, namely DFT-D2 [156], DFT-D3 [157] and DFT-D3(BJ) [158]. In these cases, pair-wise terms (and 3-body corrections for DFT-D3 and DFT-D3(BJ)) are added to the DFT energy, which are independent of the electronic density, in order to mimic the vdW interactions. The implementation includes the contributions of these methods to forces and stresses, in view of geometry optimization, as well as to first-order response functions like dynamical matrices, clamped elastic constants and internal strain coupling parameters.

To activate DFT-D dispersion correction, two keywords are in use: **vdw_xc = 5/6/7** to choose between DFT-D2, DFT-D3 and DFT-D3(BJ), and **vdw_tol**, to control the inclusion of largely distant pairs (those giving a contribution below **vdw_tol** are ignored). It is also possible to include 3-body corrections [157] (for ground-state only) with the keyword **vdw_tol_3bt**, which also controls the tolerance over this term. Four tests are available for this implementation: vdwxc#10 and v7#97 to v7#99.

Methods based on maximally localized Wannier functions (ML-WFs) to calculate vdW energy corrections have been implemented in ABINIT. In this case the pair-wise terms come from contributions of pairs of MLWFs rather than from atoms. Among the implemented methods in ABINIT it is found vdW-WF1 [159,160] vdW-WF2 [161] and vdW-QHO-WF [162]. A full description of the implementation of vdW-WF1 is reported in [163].

Selection of one of these 3 methods is achieved by using **vdw_xc = 10/11/14** respectively. Since vdW-WF1 and vdW-WF2 methods are approximations for the dispersion energy of non overlapping electronic densities, it is necessary to define the interacting fragments of the system whose dispersion energy is going to be calculated. The latter is achieved by using the input variables **vdw_nfrag** and **vdw_typfrag** to define the number of interacting fragments in the unit cell and to assign each atom to a fragment. A given MLWF belongs to the same fragment as its closer atom. The need for defining the interacting fragments is overridden in the vdW-QHO-WF, for which these input variables are not used. When dealing with periodic systems the input variable **vdw_supercell** controls the number of neighbor unit cells that will be included in the calculation. Each one of the 3 components of **vdw_supercell** indicates the maximum number of cells along both positive or negative directions of the corresponding primitive vector. This is useful for studying the spacial convergency of the vdW energy. It should be noticed that the user must set the variables associated to the calculation of MLWFs and that the resulting vdW energies strongly depend on the obtained Wannier functions. Three tests for these methods are available: wannier90#11, wannier90#12 and wannier90#13.

6.5. New G_0W_0 implementation

A high performance G_0W_0 implementation [164] has been developed within ABINIT. It is more efficient than the traditional implementation [8,79] thanks to the treatment of the two major bottlenecks: the summations over conduction states and the inversion of the dielectric matrix. The first bottleneck is circumvented by converting the summations into Sternheimer equations. Note that the introduction of approximations beyond the G_0W_0 (such as analytic continuation [165]) is avoided thanks to use of the simplified quasiminimal residual (SQMR) algorithm [166]. Then, the second bottleneck is assessed by expressing the dielectric matrix in a Lanczos basis, which reduces the matrix size by orders of magnitude with respect to a plane wave basis. A model dielectric operator has also been developed [164] and can optionally be used to further reduce the dielectric matrix size. The resulting implementation exhibits a 500-fold speedup for the silane molecule, without introducing approximations beyond the G_0W_0 formalism and the pseudopotential method. At the time of writing, the implementation supports simulations of molecular systems only (only one k-point). Furthermore, it does not supports PAW and spinors. However, it supports parallelism over bands and FFTs [39].

To compute G_0W_0 corrections to DFT eigenenergies, one needs to set **optdriver 66** in the input file. This type of calculation requires that the ground state density and wavefunctions be available from disk (using **getden** and **getwfk**). It also requires the user to specify the state to be corrected (**gwls_band_index**), the maximum allowed residual when solving the Sternheimer equations (**tolwfr**), and the number of Lanczos vectors used to express the dielectric matrix (**gwls_sternheimer_kmax**). Note that a convergence study on the latter value is required to check the accuracy of the results. Since only molecular systems are currently supported, the calculation also requires that the Coulomb potential be spherically truncated (**icutcoul 0**) with a radius **rcut** that should be validated with a convergence study. Relevant tests can be found in v67mbpt#15 (sequential) and paral#77 (parallel).

6.6. Effective masses

The direct estimation of effective masses from DFT band curvature using DFPT [167] has been implemented within the linear response part of ABINIT [168]. This method avoids the use of finite differences to estimate these masses, which eliminates the associated numerical noise and convergence study. To compute the effective masses, one has to set the keyword **efmas** to 1 within a calculation of the derivative of ground-state wavefunctions with respect to wavevector (**rfelfd 1** or **2**). The effective masses will then be computed for all k-points and bands present in the calculation. One can optionally specify the range of bands to be treated for each k-point with the keyword **efmas_nband**.

An additional feature of the effective mass implementation is the correct treatment of degenerate bands. Indeed, the concept of effective mass breaks down at degenerate band extrema since it is no longer possible to describe band curvature using a tensor [169,170]. However, using the concept of "transport equivalent effective mass" [170] and its adaptation to the **k · p** framework, the implementation is able to provide the user with effective mass tensors which, while not describing the band curvature, describe accurately the contribution of the individual bands to transport properties.

The implementation supports both NCPP and PAW schemes. Spin-polarized systems (**nspden 2**) as well as spinors (**nspinor 2**) can be treated, although the spin–orbit interaction can only be treated in the PAW case. The treatment of degenerescences is limited to the extremal points of the band structure (which are the

most relevant in any case). Relevant tests can be found in v7#80 (norm-conserving pseudopotentials) and #81 (PAW).

6.7. Improvements related to dielectric susceptibilities

The Random-Phase Approximation (RPA) to the calculation of the dielectric function can be expressed as a sum-over-states expression when local-fields effects are neglected. Starting from the momentum operators matrix elements and combining with the Kohn–Sham eigenenergies, several quantities such as first-order and second-order dielectric susceptibilities are then available[171–174].

These calculations are available in the ABINIT package via the optic plugin. First, a calculation of the ground-state wavefunctions and the derivatives with respect to the wavevectors needs to be performed within ABINIT on a grid of wavevectors in the Brillouin Zone. Then, the optic plugin reads these derivatives, the Kohn–Sham eigenenergies and calculates the dielectric suscepti-bilities.

In the last development version, the input file format has been changed to use fortran namelists instead of simple sequential fields. This allows for more user readability. Moreover, we have also implemented in the same framework the expressions derived by Hughes & Sipe [172] for the calculations of Linear-Electro Optic coefficients and we have included anti-resonant contributions as well as the scissors operator in the calculation of all second-order susceptibilities.

The description of the new input file format as well as a new version of the tutorial (tutorespfn#toptic_1-4) will be available in the forthcoming release version. Tests for the new routines are also available in v7#114-115.

6.8. Meta-GGA

The development of meta-GGA in ABINIT will continue with the implementation and testing of a mixing procedure for the kinetic energy density. Indeed, preliminary tests revealed persistent oscillations in the self-consistent field cycle. Then, the forces and the stress tensor will be added accounting for the new terms coming from the dependence of total energy on the Laplacian of the density and on the kinetic energy density. This requires in particular the calculation of the kinetic energy density tensor. This will allow automatic relaxation and geometry optimization with meta-GGA functionals. Finally, once these on-going developments will be achieved for the norm-conserving case, the case of PAW implementation will be considered.

7. Summary

In this paper, we have reviewed the most recent capabilities of the ABINIT code. They are available in version v7.10.5 on the ABINIT web site. The tests, tutorials and references mentioned here are a good starting point to be able to use these new capabilities. Some important information is also given concerning how to build and run ABINIT, the structure of the ABINIT package, the auxiliary files (NCPPs and PAW atomic data), and some post-processing capabilities. We have also covered new developments, that will be available in the forthcoming version of ABINIT: interpolation techniques for the Bethe–Salpeter equation, hybrid functionals, Van der Waals interactions, the direct computation of effective masses.

Acknowledgments

We thank Ph. Ghosez, R. Caracas and R. Shaltaf for their contributions to the life of the ABINIT community and dissemination of the ABINIT package, through the organization of workshops or tutorials.

References

[1] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, Phys. Rev. 136 (3B) (1964) B864–B871.
[2] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (4A) (1965) A1133–A1138.
[3] L. Hedin, New method for calculating the one-particle green's function with application to the electron-gas problem, Phys. Rev. A 139 (1965) 796.
[4] G. Onida, L. Reining, A. Rubio, Electronic excitations: density-functional versus many-body Green's-function approaches, Rev. Modern Phys. 74 (2002) 601–659.
[5] http://www.gnu.org/copyleft/gpl.txt.

[6] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, Ph. Ghosez, J.-Y. Raty, D.C. Allan, First-principles computation of material properties: the abinit software project, Comput. Mater. Sci. 25 (2002) 478–492.

[7] X. Gonze, M. Rignanese, M. Verstraete, J.-M. Beuken, Y. Pouillon, R. Caracas, F. Jollet, M. Torrent, G. Zerah, M. Mikami, Ph. Ghosez, M. Veithen, J.-Y. Raty, V. Olevano, F. Bruneval, L. Reining, R. Godby, G. Onida, D.R. Hamann, D.C. Allan, A brief introduction to the abinit software package, Z. Kristallogr. 220 (2005) 558–562.

[8] 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, Ph. 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, Abinit: First-principle approach to material and nanosystem properties, Comput. Phys. Comm. 180 (2009) 2582–2615.

[9] http://abinit.org.

[10] http://abinit.org/doc/helpfiles.

[11] http://forum.abinit.org.

[12] http://wiki.abinit.org/doku.php.

[13] D. Marx, M. Parrinello, Ab initio path integral molecular dynamics: Basic ideas, J. Chem. Phys. 104 (1996) 4077.

[14] M.E. Tuckerman, D. Marx, M.L. Klein, M. Parrinello, Efficient and general algorithms for path integral Car–Parrinello molecular dynamics, J. Chem. Phys. 104 (1996) 5579.

[15] G. Geneste, M. Torrent, F. Bottin, P. Loubeyre, Strong isotope effect in phase II of dense solid hydrogen and deuterium, Phys. Rev. Lett. 109 (2012) 155303.

[16] G. Geneste, A. Ottochian, J. Hermet, G. Dezanneau, Proton transport in barium stannate: classical, semi-classical and quantum regimes, Phys. Chem. Chem. Phys. 17 (2015) 19104.

[17] E. Weinan, W. Ren, E. Vanden-Eijnden, Simplified and improved string method for computing the minimum energy paths in barrier-crossing events, J. Chem. Phys. 126 (16) (2007) 164103.

[18] G. Henkelman, H. Jónsson, Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points, J. Chem. Phys. 113 (22) (2000) 9978–9985.

[19] G. Mills, H. Jónsson, Quantum and thermal effects in h 2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems, Phys. Rev. Lett. 72 (7) (1994) 1124.

[20] R. Resta, Macroscopic polarization in crystalline dielectrics: the geometric phase approach, Rev. Modern Phys. 66 (1994) 899–915.

[21] R.W. Nunes, X. Gonze, Berry-phase treatment of the homogeneous electric field perturbation in insulators, Phys. Rev. B 63 (15) (2001) 155107.

[22] I. Souza, J. Íñiguez, D. Vanderbilt, First-principles approach to insulators in finite electric fields, Phys. Rev. Lett. 89 (2002) 117602.

[23] R.D. King-Smith, D. Vanderbilt, Theory of polarization of crystalline solids, Phys. Rev. B 47 (3) (1993) 1651–1654.

[24] J.W. Zwanziger, J. Galbraith, Y. Kipouros, M. Torrent, M. Giantomassi, X. Gonze, Finite homogeneous electric fields in the projector augmented wave formalism: Applications to linear and nonlinear response, Comput. Mater. Sci. 58 (2012) 113–118.

[25] D. Vanderbilt, R.D. King-Smith, Electronic polarization in the ultrasoft pseudopotential formalism, http://arxiv.org/abs/cond-mat/9801177, 1998.

[26] R.M. Nieminen, E. Boroński, L.J. Lantto, Two-component density-functional theory: Application to positron states, Phys. Rev. B 32 (1985) 1377.

[27] E. Boroński, R.M. Nieminen, Electron-positron density-functional theory, Phys. Rev. B 34 (1986) 3820.

[28] M.J. Puska, R.M. Nieminen, Theory of positrons in solids and on solid surfaces, Rev. Modern Phys. 66 (1994) 841.

[29] F. Tuomisto, I. Makkonen, Defect identification in semiconductors with positron annihilation: Experiment and theory, Rev. Modern Phys. 85 (2013) 1583.

[30] A. Kawasuso, M. Yoshikawa, H. Itoh, T. Chiba, T. Higuchi, K. Betsuyaku, F. Redmann, R. Krause-Rehberg, Electron-positron momentum distributions associated with isolated silicon vacancies in 3 C-Si C, Phys. Rev. B 72 (4) (2005) 045204.

[31] P. Folegati, I. Makkonen, R. Ferragut, M.J. Puska, Analysis of electron-positron momentum spectra of metallic alloys as supported by first-principles calculations, Phys. Rev. B 75 (5) (2007) 054201.

[32] A. Calloni, A. Dupasquier, R. Ferragut, P. Folegati, M.M. Iglesias, I. Makkonen, M.J. Puska, Positron localization effects on the doppler broadening of the annihilation line: Aluminum as a case study, Phys. Rev. B 72 (5) (2005) 054112.

[33] C. Rauch, I. Makkonen, F. Tuomisto, Identifying vacancy complexes in compound semiconductors with positron annihilation spectroscopy: A case study of InN, Phys. Rev. B 84 (12) (2011) 125201.

[34] J. Wiktor, X. Kerbiriou, G. Jomard, S. Esnouf, M.-F. Barthe, M. Bertolus, Positron annihilation spectroscopy investigation of vacancy clusters in silicon carbide: Combining experiments and electronic structure calculations, Phys. Rev. B 89 (15) (2014) 155203.

[35] J. Wiktor, M.-F. Barthe, G. Jomard, M. Torrent, M. Freyss, M. Bertolus, Coupled experimental and DFT + U investigation of positron lifetimes in UO_2, Phys. Rev. B 90 (18) (2014) 184101.

[36] P.E. Blöchl, Projector augmented-wave method, Phys. Rev. B 50 (1994) 17953–17979.

[37] J. Wiktor, G. Jomard, M. Torrent, Two-component density functional theory within the projector augmented-wave approach: Accurate and self-consistent computations of positron lifetimes and momentum distributions, Phys. Rev. B 92 (2015) 125113.

[38] N.A.W. Holzwarth, A.R. Tackett, G.E. Matthews, A projector augmented wave (PAW) code for electronic structure calculations, part I: ATOMPAW for generating atom-centered functions, Comput. Phys. Comm. 135 (2001) 329.

[39] F. Bottin, S. Leroux, A. Knyazev, G. Zérah, Large-scale ab initio calculations based on three levels of parallelization, Comput. Mater. Sci. 42 (2) (2008) 329–336.

[40] A. Levitt, M. Torrent, Parallel eigensolvers in plane-wave density functional theory, Comp. Phys. Comm. 187 (2015) 98–105.

[41] J. Wiktor, G. Jomard, M. Torrent, M. Bertolus, Electronic structure investigation of energetics and positron lifetimes of fully relaxed monovacancies with various charge states in 3C-SiC and 6H-SiC, Phys. Rev. B 87 (2013) 235207.

[42] J. Wiktor, G. Jomard, M. Bertolus, Electronic structure calculations of positron lifetimes in SiC: Self-consistent schemes and relaxation effect, Nucl. Instrum. Methods 327 (2014) 63.

[43] A.D. Becke, K.E. Edgecombe, A simple measure of electron localization in atomic and molecular systems, J. Chem. Phys. 92 (9) (1990) 5397–5403.

[44] A. Savin, O. Jepsen, J. Flad, O.K. Andersen, H. Preuss, H.G. von Schnering, Electron localization in solid-state structures of the elements: the diamond structure, Angew. Chem., Int. Ed. Engl. 31 (2) (1992) 187–188.

[45] R.F.W. Bader, Atoms in Molecules: A Quantum Theory, Oxford University Press, ISBN: 978-0-19-855865-1, 1994.

[46] M.A.L. Marques, M.J.T. Oliveira, T. Burnus, Libxc: A library of exchange and correlation functionals for density functional theory, Comput. Phys. Comm. 183 (2012) 2272–2281.

[47] F. Tran, P. Blaha, Accurate band gaps of semiconductors and insulators with a semilocal exchange–correlation potential, Phys. Rev. Lett. 102 (2009) 226401.

[48] D. Waroquiers, A. Lherbier, A. Miglio, M. Stankovski, S. Poncé, M.J.T. Oliveira, M. Giantomassi, G.-M. Rignanese, X. Gonze, Band widths and gaps from the Tran-Blaha functional: Comparison with many-body perturbation theory, Phys. Rev. B 87 (2013) 075121.

[49] E. Bousquet, N.A. Spaldin, K.T. Delaney, Unexpectedly large electronic contribution to linear magnetoelectricity, Phys. Rev. Lett. 106 (2011) 107202.

[50] A. Mosca Conte, Quantum mechanical modeling of nano magnetism (Ph.D. thesis), SISSA, Trieste, Italy, 2007.

[51] Pui-Wai Ma, S.L. Dudarev, Constrained density functional for noncollinear magnetism, Phys. Rev. B 91 (2015) 054420.

[52] F. Bruneval, J.-P. Crocombette, X. Gonze, B. Dorado, M. Torrent, F. Jollet, Consistent treatment of charged systems within periodic boundary conditions: The projector augmented-wave and pseudopotential methods revisited, Phys. Rev. B 89 (2014) 045116.

[53] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G.L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A.P. Seitsonen, A. Smogunov, P. Umari, R.M. Wentzcovitch, QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials, J. Phys.: Condens. Matter 21 (39) (2009) 395502.

[54] X. Gonze, First-principles responses of solids to atomic displacements and homogeneous electric fields: Implementation of a conjugate-gradient algorithm, Phys. Rev. B 55 (1997) 10337–10354.

[55] X. Gonze, C. Lee, Dynamical matrices, born effective charges, dielectric permittivity tensors, and interatomic force constants from density-functional perturbation theory, Phys. Rev. B 55 (1997) 10355.

[56] C. Audouze, F. Jollet, M. Torrent, X. Gonze, Projector augmented-wave approach to density-functional perturbation theory, Phys. Rev. B 73 (2006) 235101.

[57] C. Audouze, F. Jollet, M. Torrent, X. Gonze, Comparison between projector augmented-wave and ultrasoft pseudopotential formalisms at the density-functional perturbation theory level, Phys. Rev. B 78 (2008) 035105.

[58] J.M. Ziman, Electrons and Phonons, Oxford University Press, 1960.

[59] P.B. Allen, New method for solving Boltzmann's equation for electrons in metals, Phys. Rev. B 17 (1978) 3725–3734.

[60] P.B. Allen, Boltzmann theory and resistivity of metals, in: J.R. Chelikowsky, S.G. Louie (Eds.), Quantum Theory of Real Materials, Klüwer, Boston, 1996, pp. 219–250.

[61] G. Grimvall, The Electron Phonon Interaction in Metals, North-Holland, Amsterdam, 1981.

[62] S.Y. Savrasov, D.Y. Savrasov, Electron–phonon interactions and related physical properties of metals from linear-response theory, Phys. Rev. B 54 (1996) 16487.

[63] B. Xu, M.J. Verstraete, First principles explanation of the positive seebeck coefficient of lithium, Phys. Rev. Lett. 112 (2014) 196603.

[64] O.D. Restrepo, K. Varga, S.T. Pantelides, First-principles calculations of electron mobilities in silicon: Phonon and coulomb scattering, Appl. Phys. Lett. 94 (21) (2009).

[65] P.B. Allen, V. Heine, Theory of the temperature dependence of electronic band structures, J. Phys. C: Solid State Phys. 9 (1976) 2305.

[66] P.B. Allen, M. Cardona, Theory of the temperature dependence of the direct gap of germanium, Phys. Rev. B 23 (1981) 1495–1505.

[67] P.B. Allen, M. Cardona, Temperature dependence of the direct gap of Si and Ge, Phys. Rev. B 27 (1983) 4760–4769.

[68] S. Baroni, P. Giannozzi, A. Testa, Green's-function approach to linear response in solids, Phys. Rev. Lett. 58 (1987) 1861–1864.

[69] S. Poncé, G. Antonius, Y. Gillet, P. Boulanger, J. Laflamme Janssen, A. Marini, M. Côté, X. Gonze, Temperature dependence of electronic eigenenergies in the adiabatic harmonic approximation, Phys. Rev. B 90 (2014) 214304.

[70] S. Poncé, G. Antonius, P. Boulanger, E. Cannuccia, A. Marini, M. Côté, X. Gonze, Verification of first-principles codes: Comparison of total energies, phonon frequencies, electronphonon coupling and zero-point motion correction to the gap between ABINIT and QE/Yambo, Comput. Mater. Sci. 83 (2014) 341–348.

[71] X. Gonze, P. Boulanger, M. Côté, Theoretical approaches to the temperature and zero-point motion effects on the electronic band structure, Ann. Phys. 523 (2011) 168.

[72] S. Poncé, Y. Gillet, J. Laflamme Janssen, A. Marini, M. Verstraete, X. Gonze, Temperature dependence of the electronic structure of semiconductors and insulators, J. Chem. Phys. 143 (2015) 102813.

[73] https://wiki.abinit.org/doku.php?id=howto:temperature.

[74] G. Antonius, S. Poncé, P. Boulanger, M. Côté, X. Gonze, Many-body effects on the zero-point renormalization of the band structure, Phys. Rev. Lett. 112 (2014) 215501.

[75] G. Antonius, S. Poncé, E. Lantagne-Hurtubise, G. Auclair, X. Gonze, M. Côté, Dynamical and anharmonic effects on the electron–phonon coupling and the zero-point renormalization of the electronic structure, Phys. Rev. B 92 (2015) 085137.

[76] Y. Gillet, M. Giantomassi, X. Gonze, First-principles study of excitonic effects in raman intensities, Phys. Rev. B 88 (2013) 094305.

[77] 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, Accuracy of generalized gradient approximation functionals for density-functional perturbation theory calculations, Phys. Rev. B 89 (2014) 064305.

[78] S. Albrecht, L. Reining, R. Del Sole, G. Onida, Ab initio calculation of excitonic effects in the optical spectra of semiconductors, Phys. Rev. Lett. 80 (1998) 4510–4513.

[79] M. Giantomassi, M. Stankovski, R. Shaltaf, M. Gruning, F. Bruneval, P. Rinke, G.M. Rignanese, Electronic properties of interfaces and defects from many-body perturbation theory: Recent developments and applications, Phys. Status Solidi b 248 (2) (2011) 275–289.

[80] G. Cappellini, R. Del Sole, L. Reining, F. Bechstedt, Model dielectric function for semiconductors, Phys. Rev. B 47 (1993) 9892–9895.

[81] R. Haydock, The recursive solution of the Schrödinger equation, Comput. Phys. Comm. 20 (1) (1980) 11–16.

[82] L.X. Benedict, E.L. Shirley, R.B. Bohn, Optical absorption of insulators and the electron–hole interaction: An *Ab Initio* calculation, Phys. Rev. Lett. 80 (1998) 4514–4517.

[83] M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients, Rev. Modern Phys. 64 (1992) 1045–1097.

[84] J. Lindhard, On the properties of a gas of charged particles, Mat. Fys. Medd. Dan. Vid. Selsk. 28 (1954) 8.

[85] I. Campillo, J.M. Pitarke, A.G. Eguiluz, Electronic stopping power of aluminum crystal, Phys. Rev. B 58 (1998) 10307–10314.

[86] S. Lebègue, J. Harl, T. Gould, J.G. Ángyán, G. Kresse, J.F. Dobson, Cohesive properties and asymptotics of the dispersion interaction in graphite by the random phase approximation, Phys. Rev. Lett. 105 (2010) 196401.

[87] F. Bruneval, Range-separated approach to the RPA correlation applied to the van der waals bond and to diffusion of defects, Phys. Rev. Lett. 108 (2012) 256403.

[88] F. Bruneval, X. Gonze, Accurate GW self-energies in a plane-wave basis using only a few empty states: Towards large systems, Phys. Rev. B 78 (2008) 085125.

[89] J. Harl, L. Schimka, G. Kresse, Assessing the quality of the random phase approximation for lattice constants and atomization energies of solids, Phys. Rev. B 81 (2010) 115126.

[90] F. Aryasetiawan, M. Imada, A. Georges, G. Kotliar, S. Biermann, A.I. Lichtenstein, Frequency-dependent local interactions and low-energy effective models from electronic structure calculations, Phys. Rev. B 70 (19) (2004) 195104–.

[91] B. Amadon, T. Applencourt, F. Bruneval, Screened coulomb interaction calculations: cRPA implementation and applications to dynamical screening and self-consistency in uranium dioxide and cerium, Phys. Rev. B 89 (2014) 125110.

[92] B.-C. Shih, Y. Zhang, W. Zhang, P. Zhang, Screened coulomb interaction of localized electrons in solids from first principles, Phys. Rev. B 85 (4) (2012) 045132.

[93] R. Sakuma, F. Aryasetiawan, First-principles calculations of dynamical screened interactions for the transition metal oxides MO (M = Mn, Fe, Co, Ni), Phys. Rev. B 87 (16) (2013) 165118.

[94] B. Amadon, A self-consistent DFT+DMFT scheme in the projector augmented wave method: applications to cerium, Ce_2O_3 and Pu_2O_3 with the Hubbard I solver and comparison to DFT + U, J. Phys.: Condens. Matter 24 (7) (2012) 075604.

[95] M. Cococcioni, S. de Gironcoli, Linear response approach to the calculation of the effective interaction parameters in the LDA + U method, Phys. Rev. B 71 (3) (2005) 035105.

[96] V.I. Anisimov, O. Gunnarsson, Density-functional calculation of effective coulomb interactions in metals, Phys. Rev. B 43 (10) (1991) 7570–7574.

[97] A.I. Liechtenstein, V.I. Anisimov, J. Zaanen, Density-functional theory and strong interactions: Orbital ordering in mott-hubbard insulators, Phys. Rev. B 52 (8) (1995) R5467–R5470.

[98] γ and β cerium: LDA + U calculations of ground-state parameters.

[99] A. Georges, G. Kotliar, W. Krauth, M.J. Rozenberg, Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions, Rev. Modern Phys. 68 (1996) 13–125.

[100] A. Georges, Strongly correlated electron materials: Dynamical mean-field theory and electronic structure, AIP Conf. Proc. 715 (1) (2004) 3–74.

[101] G. Kotliar, S.Y. Savrasov, K. Haule, V.S. Oudovenko, O. Parcollet, C.A. Marianetti, Electronic structure calculations with dynamical mean-field theory, Rev. Modern Phys. 78 (3) (2006) 865–951.

[102] B. Amadon, F. Lechermann, A. Georges, F. Jollet, T.O. Wehling, A.I. Lichtenstein, Plane-wave based electronic structure calculations for correlated materials using dynamical mean-field theory and projected local orbitals, Phys. Rev. B 77 (20) (2008) 205112.

[103] E. Gull, A.J. Millis, A.I. Lichtenstein, A.N. Rubtsov, M. Troyer, P. Werner, Continuous-time Monte Carlo methods for quantum impurity models, Rev. Modern Phys. 83 (2) (2011) 349–404.

[104] J. Bieder, B. Amadon, Thermodynamics of the $\alpha - \gamma$ transition in cerium from first principles, Phys. Rev. B 89 (2014) 195132.

[105] D. Bergeron, A.-M.S. Tremblay, Algorithms for optimized maximum entropy and diagnostic tools for analytic continuation, ArXiv e-prints, July, 2015.

[106] B. Amadon, S. Biermann, A. Georges, F. Aryasetiawan, The $\alpha - \gamma$ transition of cerium is entropy driven, Phys. Rev. Lett. 96 (2006) 066402.

[107] P. Werner, A. Comanac, L. de Medici, M. Troyer, A.J. Millis, Continuous-time solver for quantum impurity models, Phys. Rev. Lett. 97 (7) (2006) 076405.

[108] I. Daubechies, Lectures on Wavelets, Vol. 10, Society for Industrial and Applied Mathematics, SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, 1992.

[109] S. Goedecker, M. Teter, J. Hutter, Separable dual-space Gaussian pseudopotentials, Phys. Rev. B 54 (3) (1996) 1703–1710.

[110] C. Hartwigsen, S. Goedecker, J. Hutter, Relativistic separable dual-space Gaussian pseudopotentials from h to rn, Phys. Rev. B 58 (7) (1998) 3641–3662.

[111] L. Genovese, L. Neelov, S. Goedecker, T. Deutsch, S. Alireza Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, R. Schneider, Daubechies wavelets as a basis set for density functional pseudopotential calculations, J. Chem. Phys. 129 (2008) 014109.

[112] T. Rangel, D. Caliste, L. Genovese, M. Torrent, A wavelet-based projector augmented-wave (PAW) method: the PAW library, 2015, in press.

[113] http://hpcforge.org/plugins/mediawiki/wiki/abinit/index.php.

[114] http://icl.cs.utk.edu/magma.

[115] http://icl.cs.utk.edu/plasma.

[116] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science, J. Phys.: Condens. Matter 26 (21) (2014) 1–15.

[117] A. Harju, T. Siro, F. Federici Canova, S. Hakala, T. Rantalaiho, Computational physics on graphics processing units, Lecture Notes in Comput. Sci. 7782 (2013) 3–26.

[118] S. Maintz, B. Eck, R. Dronskowski, Speeding up plane-wave electronic-structure calculations using graphics-processing units, Comput. Phys. Comm. 182 (7) (2012) 1421–1427.

[119] L. Genovese, M. Ospici, T. Deutsch, J.-F. Méhaut, A. Neelov, S. Goedecker, Density functional theory calculation on many-cores hybrid CPU–GPU architectures in hybrid architecture, J. Chem. Phys. 131 (2009) 034103.

[120] H. Rutishauser, Simultaneous iteration method for symmetric matrices, Numer. Math. 16 (3) (1970) 205–223.

[121] Y. Zhou, Y. Saad, M.L Tiago, J.R. Chelikowsky, Self-consistent-field calculations using chebyshev-filtered subspace iteration, J. Comput. Phys. 219 (1) (2006) 172–184.

[122] Y. Zhou, Y. Saad, M.L. Tiago, J.R. Chelikowsky, Parallel self-consistent-field calculations via chebyshev-filtered subspace acceleration, Phys. Rev. E 74 (2006) 066704.

[123] http://sites.google.com/site/abinit2011tutorial.

[124] http://wiki.abinit.org/doku.php?id=howto:temperature.

[125] Y. Pouillon, J.M. Beuken, T. Deutsch, M. Torrent, X. Gonze, Organizing software growth and distributed development: the case of ABINIT, Comput. Sci. Eng. 13 (2011) 62.

[126] D.R. Hamann, M. Schlüter, C. Chiang, Norm-conserving pseudopotentials, Phys. Rev. Lett. 43 (1979) 1494–1497.

[127] G.B. Bachelet, D.R. Hamann, M. Schlüter, Pseudopotentials that work: From H to Pu, Phys. Rev. B 26 (1982) 4199–4228.

[128] A.M. Rappe, K.M. Rabe, E.s Kaxiras, J.D. Joannopoulos, Optimized pseudopotentials, Phys. Rev. B 41 (1990) 1227–1230.

[129] D.R. Hamann, Optimized norm-conserving vanderbilt pseudopotentials, Phys. Rev. B 88 (2013) 085117.

[130] K. Lejaeghere, V. Van Speybroeck, G. Van Oost, S. Cottenier, Error estimates for solid-state density-functional theory predictions: an overview by means of the ground-state elemental crystals, Crit. Rev. Solid State Mater. Sci. 39 (2014) 1.

[131] F. Jollet, M. Torrent, N. Holzwarth, Generation of projector augmented-wave atomic data: a 71 element validated table in the XML format, Comput. Phys. Comm. 185 (2014) 1246–1254.

[132] K.F. Garrity, J.W. Bennett, K.M. Rabe, D. Vanderbilt, Pseudopotentials for high-throughput DFT calculations, Comput. Mater. Sci. 81 (2014) 446–452.

[133] http://www.pseudo-dojo.org.

[134] http://gitlab.abinit.org/pseudo_dojo.

[135] http://esl.cecam.org/mediawiki/index.php/Pawxml.

[136] http://abinit.org/downloads/PAW2.
[137] http://github.com/materialsproject/pymatgen.
[138] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Python materials genomics (pymatgen) A robust, open-source python library for materials analysis, Comput. Mater. Sci. 68 (0) (2013) 314–319.
[139] http://www.materialsproject.org.
[140] http://gitlab.abinit.org/abipy.
[141] Jmol: an open-source Java viewer for chemical structures in 3D, http://www.jmol.org/.
[142] http://github.org/abinitgui/abinitgui.
[143] http://gui.abinit.org.
[144] L.-W. Wang, L. Bellaiche, S.-H. Wei, A. Zunger, Majority representation of alloy electronic states, Phys. Rev. Lett. 80 (1998) 4725.
[145] V. Popescu, A. Zunger, Effective band structure of random alloys, Phys. Rev. Lett. 104 (2010) 236403.
[146] V. Popescu, A. Zunger, Extracting e versus k effective band structure from supercell calculations on alloys and impurities, Phys. Rev. B 85 (2012) 085201.
[147] P.B. Allen, T. Berlijn, D.A. Casavant, J.M. Soler, Recovering hidden bloch character: Unfolding electrons, phonons, and slabs, Phys. Rev. B 87 (2013) 085322.
[148] Paulo V.C. Medeiros, Sven Stafström, Jonas Björk, Effects of extrinsic and intrinsic perturbations on the electronic structure of graphene: Retaining an effective primitive cell band structure by band unfolding, Phys. Rev. B 89 (2014) 041407.
[149] O. Rubel, A. Bokhanchuk, S.J. Ahmed, E. Assmann, Unfolding the band structure of disordered solids: from bound states to high-mobility kane fermions, Phys. Rev. B 90 (2014) 115202.
[150] K. Momma, F. Izumi, Vesta3 for three-dimensional visualization of crystal, volumetric and morphology data, J. Appl. Crystallogr. 44 (6) (2011) 1272–1276.
[151] M. Rohlfing, S.G. Louie, Electron–hole excitations and optical spectra from first principles, Phys. Rev. B 62 (2000) 4927–4944.
[152] Y. Gillet, M. Giantomassi, X. Gonze, Efficient interpolation technique for Bethe–Salpeter calculation of optical spectra, Comput. Phys. Comm. 203C (2016) 83–93.
[153] C. Rostgaard, Exact exchange in density functional calculations (M.S. thesis), Technical University of Denmark, 2006.
[154] M. Dion, H. Rydberg, E. Schröder, D.C. Langreth, B.I. Lundqvist, Van der waals density functional for general geometries, Phys. Rev. Lett. 92 (2004) 246401. Erratum: http://dx.doi.org/110.1103/PhysRevLett.95.109902.
[155] G. Román-Pérez, J.M. Soler, Efficient implementation of a Van der Waals density functional: Application to double-wall carbon nanotubes, Phys. Rev. Lett. 103 (2009) 096102.
[156] S. Grimme, Semiempirical GGA-type density functional constructed with a long-range dispersion correction, J. Comput. Chem. 27 (2006) 1787.
[157] S. Grimme, J. Anthony, S. Ehrlich, H. Krieg, A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu, J. Chem. Phys. 132 (2010) 154104.
[158] S. Grimme, S. Ehrlich, L. Goerigk, Effect of the damping function in dispersion corrected density functional theory, J. Comput. Chem. 32 (2011) 1456–1465.
[159] P.L. Silvestrelli, van der Waals interactions in DFT made easy by Wannier functions, Phys. Rev. Lett. 100 (2008) 053002.
[160] P.L. Silvestrelli, van der Waals interactions in density functional theory using Wannier functions, J. Phys. Chem. A 113 (2009) 5224.
[161] A. Ambrosetti, P.L. Silvestrelli, van der Waals interactions in density functional theory using wannier functions: Improved C6 and C3 coefficients by a different approach, Phys. Rev. B 85 (2012) 073101.
[162] P.L. Silvestrelli, van der Waals interactions in density functional theory by combining the quantum harmonic oscillator-model with localized Wannier functions, J. Chem. Phys. 139 (2013) 054106.
[163] C. Espejo, T. Rangel, Y. Pouillon, A.H. Romero, X. Gonze, Wannier functions approach to van der Waals interactions in ABINIT, Comput. Phys. Comm. 183 (2012) 480.
[164] J. Laflamme Janssen, B. Rousseau, M. Côté, Efficient dielectric matrix calculations using the Lanczos algorithm for fast many-body G_0W_0 implementations, Phys. Rev. B 91 (12) (2015) 125120.
[165] M. Rieger, L. Steinbeck, I.D. White, H.N. Rojas, R.W. Godby, The GW space–time method for the self-energy of large systems, Comput. Phys. Comm. 117 (3) (1999) 211–228.
[166] R.W. Freund, N.M. Nachtigal, Software for simplified Lanczos and QMR algorithms, Appl. Numer. Math. 19 (1995) 319.
[167] X. Gonze, Adiabatic density-functional perturbation theory, Phys. Rev. A 52 (2) (1995) 1096–1114.
[168] J. Laflamme Janssen, Y. Gillet, S. Poncé, A. Martin, M. Torrent, X. Gonze, Accurate effective masses from density functional perturbation theory, in preparation.
[169] J.M. Luttinger, W. Kohn, Motion of electrons and holes in perturbed periodic fields, Phys. Rev. 97 (4) (1955) 869–883.
[170] N.A. Mecholsky, L. Resca, I.L. Pegg, M. Fornari, Theory of band warping and its effects on thermoelectronic transport properties, Phys. Rev. B 89 (15) (2014) 155131.
[171] J.E. Sipe, E. Ghahramani, Nonlinear optical response of semiconductors in the independent-particle approximation, Phys. Rev. B 48 (1993) 11705–11722.
[172] J.L.P. Hughes, J.E. Sipe, Calculation of second-order optical response in semiconductors, Phys. Rev. B 53 (1996) 10751–10763.
[173] S. Sharma, J.K. Dewhurst, C. Ambrosch-Draxl, Linear and second-order optical response of III–V monolayer superlattices, Phys. Rev. B 67 (2003) 165332.
[174] S. Sharma, C. Ambrosch-Draxl, Second-harmonic optical response from first principles, Phys. Scr. 2004 (2004) 128.