. SCUOLA INTERNAZIONALE SUPERIORE DI STUDI AVANZATI

SISSA Digital Library

Advanced capabilities for materials modelling with Quantum ESPRESSO

Original
Advanced capabilities for materials modelling with Quantum ESPRESSO / Giannozzi, P., Andreussi, O., Brumme, T., Bunau, O., Buongiorno Nardelli, M., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Cococcioni, M., Colonna, N., Carnimeo, I., Dal Corso, A., De Gironcoli, S., Delugas, P., Distasio, R.A., Ferretti, A., Floris, A., Fratesi, G., Fugallo, G., et al.. - In: JOURNAL OF PHYSICS. CONDENSED MATTER. - ISSN 0953-8984. - 29:46(2017), pp. 1-30. [10.1088/1361-648X/aa8f79]

Availability:
This version is available at: 20.500.11767/68925 since: 2018-09-24T21:11:25Z

Publisher:

Published
DOI:10.1088/1361-648X/aa8f79

Terms of use:

Testo definito dall'ateneo relativo alle clausole di concessione d'uso

Publisher copyright

note finali coverpage

(Article begins on next page)

13 October 2026

arXiv:1709.10010v1 [cond-mat.mtrl-sci] 28 Sep 2017

Advanced capabilities for materials modelling with QUANTUM ESPRESSO

P. Giannozzi[a], O. Andreussi[b,i], T. Brumme[c], O. Bunau[d], M. Buongiorno Nardelli[e], M. Calandra[d], R. Car[f], C. Cavazzoni[g], D. Ceresoli[h], M. Cococcioni[i], N. Colonna[i], I. Carnimeo[a], A. Dal Corso[j], S. de Gironcoli[j], P. Delugas[j], R. A. DiStasio Jr.[k], A. Ferretti[l], A. Floris[m], G. Fratesi[n], G. Fugallo[o], R. Gebauer[p], U. Gerstmann[q], F. Giustino[r], T. Gorni[j], J. Jia[k], M. Kawamura[s], H.-Y. Ko[f], A. Kokalj[t], E. Küçükbenli[j], M. Lazzeri[d], M. Marsili[u], N. Marzari[i], F. Mauri[v], N. L. Nguyen[j], H.-V. Nguyen[w], A. Otero-de-la-Roza[x], L. Paulatto[d], S. Poncé[r], D. Rocca[y,z], R. Sabatini[1], B. Santra[f], M. Schlipf[r], A. P. Seitsonen[2,3], A. Smogunov[4], I. Timrov[i], T. Thonhauser[5], P. Umari[u,6], N. Vast[7], X. Wu[8], S. Baroni [j]

(a) Dept. of Mathematical, Physical, and Computer Sciences, University of Udine, via delle Scienze 206, I-33100 Udine, Italy

(b) Institute of Computational Sciences, Università della Svizzera Italiana, Lugano, Svizzera

(c) Wilhelm-Ostwald-Institute of Physical and Theoretical Chemistry, Leipzig University, Linnéstr. 2, D-04103 Leipzig, Germany

(d) IMPMC, UMR CNRS 7590, Sorbonne Universités-UPMC University Paris 06, MNHN, IRD, 4 Place Jussieu, F-75005 Paris, France

(e) Department of Physics and Department of Chemistry, University of North Texas, Denton, USA

(f) Department of Chemistry, Princeton University, Princeton, NJ 08544, USA

(g) CINECA - Via Magnanelli 6/3, I-40033 Casalecchio di Reno, Bologna, Italy

(h) Institute of Molecular Science and Technologies (ISTM), National Research Council (CNR), I-20133 Milano, Italy

(i) Theory and Simulation of Materials (THEOS), and National Centre for Computational Design and Discovery of Novel Materials (MARVEL), Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

(j) SISSA-ISAS, via Bonomea, 265, I-34136 Trieste, Italy

(k) Department of Chemistry and Chemical Biology, Cornell University, Ithaca, NY 14853 USA

(l) CNR Istituto Nanoscienze, I-42125 Modena, Italy

(m) University of Lincoln, UK (n) Dipartimento di Fisica, Università degli Studi di Milano, via Celoria 16, I-20133 Milano, Italy (o) ETSF, Laboratoire des Solides Irradiés, Ecole Polytechnique, F-91128 Palaiseau cedex, France

(p) The Abdus Salam International Centre for Theoretical Physics (ICTP), Strada Costiera 11, I-34151 Trieste, Italy

(q) Department Physik, Universität Paderborn, D-33098 Paderborn, Germany
(r) Department of Materials, University of Oxford,
Parks Road, Oxford OX1 3PH, United Kingdom
(s) The Institute for Solid State Physics, Kashiwa, Japan
(t) Department of Physical and Organic Chemistry,
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
(u) Dipartimento di Fisica e Astronomia,
Università di Padova, via Marzolo 8, I-35131 Padova, Italy
(v) Dipartimento di Fisica, Università di Roma La Sapienza,
Piazzale Aldo Moro 5, I-00185 Roma, Italy
(w) Institute of Physics, Vietnam Academy of Science and Technology, 10 Dao Tan, Hanoi, Vietnam
(x) Department of Chemistry, University of British Columbia,
Okanagan, Kelowna BC V1V 1V7, Canada
(y) Université de Lorraine, CRM^2, UMR 7036, F-54506 Vandoeuvre-lès-Nancy, France
(z) CNRS, CRM^2, UMR 7036, F-54506 Vandoeuvre-lès-Nancy, France
(1) Orionis Biosciences, Boston (2) Institut für Chimie,
Universität Zurich, CH-8057 Zürich, Switzerland
(3) Département de Chimie, École Normale Supérieure, F-75005 Paris, France
(4) SPEC, CEA, CNRS, Université Paris-Saclay, F-91191 Gif-Sur-Yvette, France
(5) Department of Physics, Wake Forest University, Winston-Salem, NC 27109, USA
(6) CNR-IOM DEMOCRITOS, Istituto Officina dei Materiali, Consiglio Nazionale delle Ricerche
(7) Laboratoire des Solides Irradiés, École Polytechnique, CEA-DRF-IRAMIS,
CNRS UMR 7642, Université Paris-Saclay, F-91120 Palaiseau, France
(8) Department of Physics, Temple University, Philadelphia, PA 19122-1801, USA

(Dated: September 29, 2017)

QUANTUM ESPRESSO is an integrated suite of open-source computer codes for quantum simulations of materials using state-of-the art electronic-structure techniques, based on density-functional theory, density-functional perturbation theory, and many-body perturbation theory, within the plane-wave pseudo-potential and projector-augmented-wave approaches. QUANTUM ESPRESSO owes its popularity to the wide variety of properties and processes it allows to simulate, to its performance on an increasingly broad array of hardware architectures, and to a community of researchers that rely on its capabilities as a core open-source development platform to implement theirs ideas. In this paper we describe recent extensions and improvements, covering new methodologies and property calculators, improved parallelization, code modularization, and extended interoperability both within the distribution and with external software.

CONTENTS

I. Introduction 4

II. Theoretical, algorithmic, and methodological extensions 6

A. Advanced functionals 7

1. Advanced implementation of exact (Fock) exchange and hybrid functionals 7

2. Dispersion interactions 10

3. Hubbard-corrected functionals: DFT+U 15

4. Adiabatic-connection fluctuation-dissipation theory 18

B. Linear response and excited states without virtual orbitals 20

1. Static perturbations and vibrational spectroscopy 21

2. Dynamic perturbations: optical, electron energy loss, and magnetic spectroscopies 22

3. Many-body perturbation theory 27

C. Other spectroscopies 29

1. `QE-GIPAW`: Nuclear magnetic and electron paramagnetic resonance 29

2. `XSpectra`: $L_{2,3}$ X-ray absorption edges 31

D. Other lattice-dynamical and thermal properties 32

1. `thermo_pw`: Thermal properties from the quasi-harmonic approximation 32

2. `thermal2`: phonon-phonon interaction and thermal transport 33

3. `EPW`: Electron-phonon coefficients from Wannier interpolation 35

4. Non-perturbative approaches to vibrational spectroscopies 37

E. Multi-scale modeling 38

1. `Environ`: Self-Consistent Continuum Solvation embedding model 38

2. QM-MM 40

F. Miscellaneous feature enhancements and additions 41

1. Fully relativistic projector augmented-wave method 41

2. Electronic and structural properties in field-effect configuration 42

3. Cold restart of Car-Parrinello molecular dynamics 43

4. Optimized tetrahedron method 45

5. Wyckoff positions 46

III. Parallelization, modularization, interoperability and stability 47

A. New parallelization levels 47

B. Aspects of interoperability 49
C. Input/Output and data file format 50
1. XML files with schema 51
2. Large-record data file format 51
D. Organization of the distribution 52
1. Package re-organization and modularization 53
2. Modular parallelism 53
3. Reorganization of linear-response codes 54
E. Quantum ESPRESSO and scripting languages 54
1. `AiiDA`: a python materials' informatics infrastructure 54
2. `Pwtk`: a toolkit for `PWscf` 55
3. `QE-emacs-modes` 56
F. Continuous Integration and testing 57

IV. Outlook and conclusions 59

References 60

I. INTRODUCTION

Numerical simulations based on density-functional theory (DFT) [1, 2] have become a powerful and widely used tool for the study of materials properties. Many of such simulations are based upon the "plane-wave pseudopotential method", often using ultrasoft pseudopotentials [3] or the projector augmented wave method (PAW) [4] (in the following, all of these modern developments will be referred to under the generic name of "pseudopotentials"). An important role in the diffusion of DFT-based techniques has been played by the availability of robust and efficient software implementations [5], as is the case for Quantum ESPRESSO, which is an open-source software distribution—*i.e.*, an integrated suite of codes—for electronic-structure calculations based on DFT or many-body perturbation theory, and using plane-wave basis sets and pseudopotentials [6].

The core philosophy of Quantum ESPRESSO can be summarized in four keywords: openness, modularity, efficiency, and innovation. The distribution is based on two core packages, `PWscf` and `CP`, performing self-consistent and molecular-dynamics calculations respectively, and on additional packages for more advanced calculations. Among these we quote in particular: `PHonon`, for linear-response calculations of vibrational properties; `PostProc`, for data analysis and postprocessing;

`atomic`, for pseudopotential generation; `XSpectra`, for the calculation of X-ray absorption spectra; `GIPAW`, for nuclear magnetic resonance and electron paramagnetic resonance calculations.

In this paper we describe and document the novel or improved capabilities of QUANTUM ESPRESSO up to and including version 6.2. We do not cover features already present in v.4.1 and described in Ref. 6, to which we refer for further details. The list of enhancements includes theoretical and methodological extensions but also performance enhancements for current parallel machines and modularization and extended interoperability with other software.

Among the theoretical and methodological extensions, we mention in particular:

- Fast implementations of exact (Fock) exchange for hybrid functionals [7, 42–44]; implementation of non-local van der Waals functionals [9] and of explicit corrections for van der Waals interactions [10–13]; improvement and extensions of Hubbard-corrected functionals [14, 15].
- Excited-state calculations within time-dependent density-functional and many-body perturbation theories.
- Relativistic extension of the PAW formalism, including spin-orbit interactions in density-functional theory[16, 17].
- Continuum embedding environments (dielectric solvation models, electronic enthalpy, electronic surface tension, periodic boundary corrections) via the `Environ` module [18, 19] and its time-dependent generalization [20].

Several new packages, implementing the calculation of new properties, have been added to QUANTUM ESPRESSO. We quote in particular:

- `turboTDDFT` [21–24] and `turboEELS` [25, 26], for excited-state calculations within time-dependent DFT (TDDFT), without computing virtual orbitals, also interfaced with the `Environ` module (see above).
- `QE-GIPAW`, replacing the old `GIPAW` package, for nuclear magnetic resonance and electron paramagnetic resonance calculations.
- `EPW`, for electron-phonon calculations using Wannier-function interpolation [27].
- `GWL` and `SternheimerGW` for quasi-particle and excited-state calculations within many-body perturbation theory, without computing any virtual orbitals, using the Lanczos bi-orthogonalization [28, 29] and multi-shift conjugate-gradient methods [30], respectively.

- `thermo_pw`, for computing thermodynamical properties in the quasi-harmonic approximation, also featuring an advanced master-slave distributed computing scheme, applicable to generic high-throughput calculations [31].

- 3q and `thermal2`, for the calculation of anharmonic 3-body interatomic force constants, phonon-phonon interaction and thermal transport [32, 33].

Improved parallelization is crucial to enhance performance and to fully exploit the power of modern parallel architectures. A careful removal of memory bottlenecks and of scalar sections of code is a pre-requisite for better and extending scaling. Significant improvements have been achieved, in particular for hybrid functionals [34, 35].

Complementary to this, a complete pseudopotential library, `pslibrary`, including fully-relativistic pseudopotentials, has been generated [36, 37]. A curation effort [38] on all the pseudopotential libraries available for Quantum ESPRESSO has led to the identification of optimal pseudopotentials for efficiency or for accuracy in the calculations, the latter delivering an agreement comparable to any of the best all-electron codes [5]. Finally, a significant effort has been dedicated to modularization and to enhanced interoperability with other software. The structure of the distribution has been revised, the code base has been re-organized, the format of data files re-designed in line with modern standards. As notable examples of interoperability with other software, we mention in particular the interfaces with the `LAMMPS` molecular dynamics (MD) code [39] used as molecular-mechanics "engine" in the Quantum ESPRESSO implementation of the QM-MM methodology [40], and with the `i-PI` MD driver [41], also featuring path-integral MD.

All advances and extensions that have not been documented elsewhere are described in the next sections. For more details on new packages we refer to the respective references.

The paper is organized as follows. Sec. II contains a description of new theoretical and methodological developments and of new packages distributed together with Quantum ESPRESSO. Sec. III contains a description of improvements of parallelization, updated information on the philosophy and general organization of Quantum ESPRESSO, notably in the field of modularization and interoperability. Sec. IV contains an outlook of future directions and our conclusions.

II. THEORETICAL, ALGORITHMIC, AND METHODOLOGICAL EXTENSIONS

In the following, CGS units are used, unless noted otherwise.

A. Advanced functionals

1. *Advanced implementation of exact (Fock) exchange and hybrid functionals*

Hybrid functionals are already the *de facto* standard in quantum chemistry and are quickly gaining popularity in the condensed-matter physics and computational materials science communities. Hybrid functionals reduce the self-interaction error that plagues lower-rung exchange-correlation functionals, thus achieving more accurate and reliable predictive capabilities. This is of particular importance in the calculation of orbital energies, which are an essential ingredient in the treatment of band alignment and charge transfer in heterogeneous systems, as well as the input for higher-level electronic-structure calculations based on many-body perturbation theory. However, the widespread use of hybrid functionals is hampered by the often prohibitive computational requirements of the exact-exchange (Fock) contribution, especially when working with a plane-wave basis set. The basic ingredient here is the action $(\hat{V}_x\phi_i)(\mathbf{r})$ of the Fock operator $\hat{V}_x$ onto a (single-particle) electronic state ϕ_i, requiring a sum over all occupied Kohn-Sham (KS) states $\{\psi_j\}$. For spin-unpolarized systems, one has:

$$(\hat{V}_x\phi_i)(\mathbf{r}) = -e^2 \sum_j \psi_j(\mathbf{r}) \int d\mathbf{r}' \frac{\psi_j^*(\mathbf{r}')\phi_i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \tag{1}$$

where $-e$ is the charge of the electron. In the original algorithm [6] implemented in `PWscf`, self-consistency is achieved *via* a double loop: in the inner one the ψ's entering the definition of the Fock operator in Eq. (1) are kept fixed, while the outer one cycles until the Fock operator converges to within a given threshold. In the inner loop, the integrals appearing in Eq. (1):

$$v_{ij}(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho_{ij}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}, \qquad \rho_{ij}(\mathbf{r}) = \psi_i^*(\mathbf{r})\phi_j(\mathbf{r}), \tag{2}$$

are computed by solving the Poisson equation in reciprocal space using fast Fourier transforms (FFT). This algorithm is straightforward but slow, requiring $\mathcal{O}\big((N_bN_k)^2\big)$ FFTs, where N_b is the number of electronic states ("bands" in solid-state parlance) and N_k the number of $\mathbf{k}$ points in the Brillouin zone (BZ). While feasible in relatively small cells, this unfavorable scaling with the system size makes calculations with hybrid functionals challenging if the unit cell contains more than a few dozen atoms.

To enable exact-exchange calculations in the condensed phase, various ideas have been conceived and implemented in recent Quantum ESPRESSO versions. Code improvements aimed at either optimizing or better parallelizing the standard algorithm are described in Sec. III A. In this section we describe two important algorithmic developments in Quantum ESPRESSO, both

entailing a significant reduction in the computational effort: the *adaptively compressed exchange* (ACE) concept [7] and a linear-scaling ($\mathcal{O}(N_b)$) framework for performing hybrid-functional *ab initio* molecular dynamics using maximally localized Wannier functions (MLWF) [42–44].

a. Adaptively compressed exchange The simple formal derivation of ACE allows for a robust implementation, which applies straightforwardly both to isolated or aperiodic systems ($\Gamma-$only sampling of the BZ, that is, $\mathbf{k} = 0$) and to periodic ones (requiring sums over a grid of $\mathbf{k}$ points in the BZ); to norm conserving and ultrasoft pseudopotentials or PAW; to spin-unpolarized or polarized cases or to non-collinear magnetization. Furthermore, ACE is compatible with, and takes advantage of, all available parallelization levels implemented in QUANTUM ESPRESSO: over plane waves, over $\mathbf{k}$ points, and over bands.

With ACE, the action of the exchange operator is rewritten as

$$|\hat{V}_x\phi_i\rangle \simeq \sum_{jm} |\xi_j\rangle (M^{-1})_{jm} \langle \xi_m|\phi_i\rangle, \tag{3}$$

where $|\xi_i\rangle = \hat{V}_x|\psi_i\rangle$ and $M_{jm} = \langle\psi_j|\xi_m\rangle$. At self-consistency, ACE becomes exact for ϕ_i's in the occupied manifold of KS states. It is straightforward to implement ACE in the double-loop structure of `PWscf`. The new algorithm is significantly faster while not introducing any loss of accuracy at convergence. Benchmark tests on a single processor show a $3\times$ to $4\times$ speedup for typical calculations in molecules, up to $6\times$ in extended systems [45].

An additional speedup may be achieved by using a reduced FFT cutoff in the solution of Poisson equations. In Eq. (1), the exact FFT algorithm requires a FFT grid containing G-vectors up to a modulus $G_{max} = 2G_c$, where G_c is the largest modulus of G-vectors in the plane-wave basis used to expand ψ_i and ϕ_j, or, in terms of kinetic energy cutoff, up to a cutoff $E_x = 4E_c$, where E_c is the plane-wave cutoff. The presence of a $1/G^2$ factor in the reciprocal space expression suggests, and experience confirms, that this condition can be relaxed to $E_x \sim 2E_c$ with little loss of precision, down to $E_x = E_c$ at the price of increasing somewhat this loss [46]. The kinetic-energy cutoff for Fock-exchange computations can be tuned by specifying the keyword `ecutfock` in input.

Hybrid functionals have also been extended to the case of ultrasoft pseudopotentials and to PAW, following the method of Ref. 47. A large number of integrals involving augmentation charges q_{lm} are needed in this case, thus offsetting the advantage of a smaller plane-wave basis set. Better performances are obtained by exploiting the localization of the q_{lm} and computing the related terms in real space, at the price of small aliasing errors.

These improvements allow to significantly speed up a calculation, or to execute it on a larger number of processors, thus extending the reach of calculations with hybrid functionals. The bot-

tleneck represented by the sum over bands and by the FFT in Eq. (1) is however still present: ACE just reduces the number of such expensive calculations, but doesn't eliminate them. In order to achieve a real breakthrough, one has to get rid of delocalized bands and FFT's, moving to a representation of the electronic structure in terms of localized orbitals. Work along this line using the *selected column density matrix* localization scheme [48, 49] is ongoing. In the next section we describe a different approach, implemented in the `CP` code, based on maximally localized Wannier functions (MLWF).

b. Ab-initio molecular dynamics using maximally localized Wannier functions The `CP` code can now perform highly efficient hybrid-functional *ab initio* MD using MLWFs [50] $\{\overline{\varphi}_i\}$ to represent the occupied space, instead of the canonical KS orbitals $\{\psi_i\}$, which are typically delocalized over the entire simulation cell. The MLWF localization procedure can be written as a unitary transformation, $\overline{\varphi}_i(\mathbf{r}) = \sum_j U_{ij}\psi_j(\mathbf{r})$, where U_{ij} is computed at each MD time step by minimizing the total spread of the orbitals via a second-order damped dynamics scheme, starting with the converged U_{ij} from the previous time step as initial guesses [51].

The natural sparsity of the exchange interaction provided by a localized representation of the occupied orbitals (at least in systems with a finite band gap) is efficiently exploited during the evaluation of exact-exchange based applications (*e.g.*, hybrid DFT functionals). This is accomplished by computing each of the required pair-exchange potentials $\overline{v}_{ij}(\mathbf{r})$ (corresponding to a given localized pair-density $\overline{\rho}_{ij}(\mathbf{r})$) through the numerical solution of the Poisson equation:

$$\nabla^2\overline{v}_{ij}(\mathbf{r}) = -4\pi\overline{\rho}_{ij}(\mathbf{r}), \qquad \overline{\rho}_{ij}(\mathbf{r}) = \overline{\varphi}_i^*(\mathbf{r})\overline{\varphi}_j(\mathbf{r}) \tag{4}$$

using finite differences on the real-space grid. Discretizing the Laplacian operator (∇^2) using a 19-point central-difference stencil (with an associated $\mathcal{O}(h^6)$ accuracy in the grid spacing h), the resulting sparse linear system of equations is solved using the conjugate-gradient technique subject to the boundary conditions imposed by a multipolar expansion of $\overline{v}_{ij}(\mathbf{r})$:

$$\overline{v}_{ij}(\mathbf{r}) = 4\pi\sum_{lm}\frac{Q_{lm}}{2l+1}\frac{Y_{lm}(\theta,\phi)}{r^{l+1}}, \qquad Q_{lm} = \int d\mathbf{r}Y_{lm}^*(\theta,\phi)r^l\overline{\rho}_{ij}(\mathbf{r}) \tag{5}$$

in which the Q_{lm} are the multipoles describing $\overline{\rho}_{ij}(\mathbf{r})$ [42–44].

Since $\overline{v}_{ij}(\mathbf{r})$ only needs to be evaluated for *overlapping pairs* of MLWFs, the number of Poisson equations that need to be solved is substantially decreased from $\mathcal{O}(N_b^2)$ to $\mathcal{O}(N_b)$. In addition, $\overline{v}_{ij}(\mathbf{r})$ only needs to be solved on a subset of the real-space grid (that is in general of fixed size) that encompasses the overlap between a given pair of MLWFs. This further reduces the overall computational effort required to evaluate exact-exchange related quantities and results in a linear-

scaling ($\mathcal{O}(N_b)$) algorithm. As such, this framework for performing exact-exchange calculations is most efficient for non-metallic systems (*i.e.*, systems with a finite band gap) in which the occupied KS orbitals can be efficiently localized.

The MLWF representation not only yields the exact-exchange energy $E_{\rm xx}$,

$$E_{\rm xx} = -e^2 \sum_{ij} \int d\mathbf{r}\, \overline{\rho}_{ij}(\mathbf{r}) \overline{v}_{ij}(\mathbf{r}), \tag{6}$$

at a significantly reduced computational cost, but it also provides an amenable way of computing the exact-exchange contributions to the (MLWF) wavefunction forces, $\overline{D}^i_{xx}(\mathbf{r}) = e^2 \sum_j \overline{v}_{ij}(\mathbf{r}) \overline{\varphi}_j(\mathbf{r})$, which serve as the central quantities in Car-Parrinello MD simulations [53]. Moreover, the exact-exchange contributions to the stress tensor are readily available, thereby providing a general code base which enables hybrid DFT based simulations in the NVE, NVT, and NPT ensembles for simulation cells of any shape [44]. We note in passing that applications of the current implementation of this MLWF-based exact-exchange algorithm are limited to Γ–point calculations employing norm-conserving pseudo-potentials.

The MLWF-based exact-exchange algorithm in `CP` employs a hybrid MPI/OpenMP parallelization strategy that has been extensively optimized for use on large-scale massively-parallel (super-) computer architectures. The required set of Poisson equations—each one treated as an independent task—are distributed across a large number of MPI ranks/processes using a task distribution scheme designed to minimize the communication and to balance computational workload. Performance profiling demonstrates excellent scaling up to 30,720 cores (for the α-glycine molecular crystal, see Fig. 1) and up to 65,536 cores (for $(H_2O)_{256}$, see Ref. 43) on *Mira* (BG/Q) with extremely promising efficiency. In fact, this algorithm has already been successfully applied to the study of long-time MD simulations of large-scale condensed-phase systems such as $(H_2O)_{128}$ [43, 52]. For more details on the performance and implementation of this exact-exchange algorithm, we refer the reader to Ref. 44.

2. Dispersion interactions

Dispersion, or van der Waals, interactions arise from dynamical correlations among charge fluctuations occurring in widely separated regions of space. The resulting attraction is a non-local correlation effect that cannot be reliably captured by any local (such as local density approximation, LDA) or semi-local (generalized gradient approximation, GGA) functional of the electron density [54]. Such interactions can be either accounted for by a truly non-local exchange-correlation

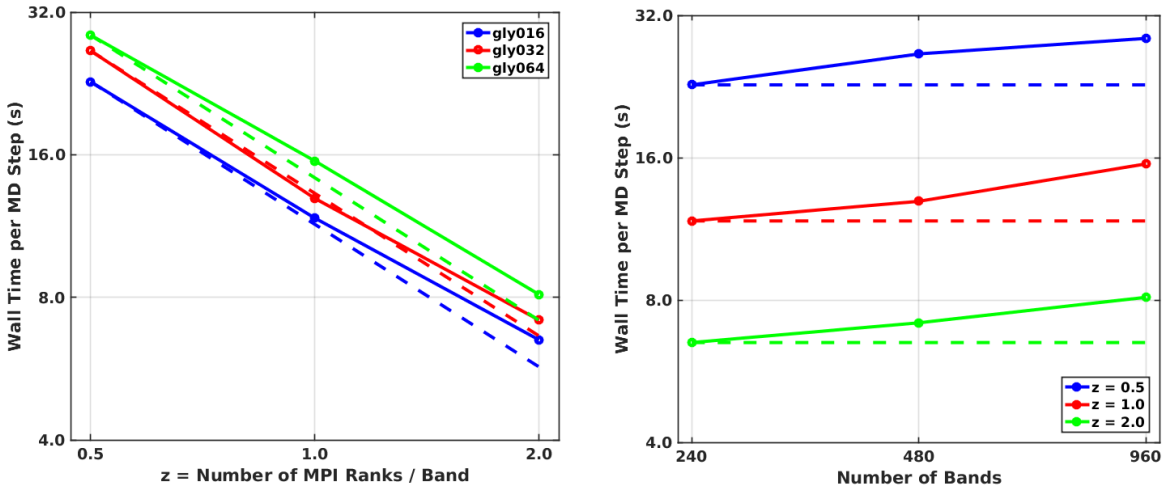

FIG. 1. Strong (*left*) and weak (*right*) scaling plots on *Mira* (BG/Q) for hybrid-DFT simulations of the α-glycine molecular crystal polymorph using the linear-scaling exact-exchange algorithm in `CP`. In these plots, unit cells containing 16-64 glycine molecules (160-640 atoms, 240-960 bands) were considered as a function of `z`, the number of MPI ranks per band (`z` = 0.5-2). On *Mira*, 30,720 cores (1920 MPI ranks $\times$ 16 OpenMP threads/rank $\times$ 1 core/OpenMP thread) were utilized for the largest system (gly064, `z` = 2), retaining over 88% (strong scaling) and 80% (weak scaling) of the ideal efficiencies (dashed lines). Deviations from ideal scaling are primarily due to the FFT (which scales non-linearly) required to provide the MLWFs in real space.

(XC) functional, or modeled by effective interactions amongst atoms, whose parameters are either computed from first principles or estimated semi-empirically. In QUANTUM ESPRESSO both approaches are implemented. Non-local XC functionals are activated by selecting them in the `input_dft` variable, while explicit interactions are turned on with the `vdw_corr` option. From the latter group, DFT-D2 [10], Tkatchenko-Scheffler [11], and exchange-hole dipole moment models [12, 13] are currently implemented (DFT-D3 [55] and the many-body dispersion (MBD) [56–58] approaches are already available in a development version).

a. Non-local van der Waals density functionals A fully non-local correlation functional able to account for van der Waals interactions for general geometries was first developed in 2004 and named vdW-DF [59]. Its development is firmly rooted in many-body theory, where the so-called adiabatic connection fluctuation-dissipation theorem (ACFD) [60] provides a formally exact expression for the XC energy through a coupling constant integration over the response function—see Sec. II A 4. A detailed review of the vdW-DF formalism is provided in Ref. 9. The overall XC energy given by the ACFD theorem—as a functional of the electron density n—is then split in vdW-DF into a

GGA-type XC part $E_{\rm xc}^0[n]$ and a truly non-local correlation part $E_{\rm c}^{\rm nl}[n]$, *i.e.*

$$E_{\rm xc}[n] = E_{\rm xc}^0[n] + E_{\rm c}^{\rm nl}[n] \;, \tag{7}$$

where the non-local part is responsible for the van der Waals forces. Through a second-order expansion in the plasmon-response expression used to approximate the response function, the non-local part turns into a computationally tractable form involving a universal kernel $\Phi(\mathbf{r},\mathbf{r}')$,

$$E_{\rm c}^{\rm nl}[n] = \frac{1}{2}\int d\mathbf{r}\; d\mathbf{r}'\; n(\mathbf{r})\;\Phi(\mathbf{r},\mathbf{r}')\;n(\mathbf{r}') \;. \tag{8}$$

The kernel $\Phi(\mathbf{r},\mathbf{r}')$ depends on $\mathbf{r}$ and $\mathbf{r}'$ only through $q_0(\mathbf{r})|\mathbf{r}-\mathbf{r}'|$ and $q_0(\mathbf{r}')|\mathbf{r}-\mathbf{r}'|$, where $q_0(\mathbf{r})$ is a function of $n(\mathbf{r})$ and $\nabla n(\mathbf{r})$. As such, the kernel can be pre-calculated, tabulated, and stored in some external file. To make the scheme self-consistent, the XC potential $V_{\rm c}^{\rm nl}(\mathbf{r}) = \delta E_{\rm c}^{\rm nl}[n]/\delta n(\mathbf{r})$ also needs to be computed [61]. The evaluation of $E_{\rm c}^{\rm nl}[n]$ in Eq. (8) is computationally expensive. In addition, the evaluation of the corresponding potential $V_{\rm c}^{\rm nl}(\mathbf{r})$ requires one spatial integral for each point $\mathbf{r}$. A significant speedup can be achieved by writing the kernel in terms of splines [62]

$$\begin{aligned}\Phi(\mathbf{r},\mathbf{r}') &= \Phi\big(q_0(\mathbf{r}), q_0(\mathbf{r}'), |\mathbf{r}-\mathbf{r}'|\big) \\ &\approx \sum_{\alpha\beta} \Phi(q_\alpha, q_\beta, |\mathbf{r}-\mathbf{r}'|)\; p_\alpha\big(q_0(\mathbf{r})\big)\; p_\beta\big(q_0(\mathbf{r}')\big) \;,\end{aligned} \tag{9}$$

where q_α are fixed values and p_α are cubic splines. Equation (8) then becomes a convolution that can be simplified to

$$\begin{aligned}E_{\rm c}^{\rm nl}[n] &= \frac{1}{2}\sum_{\alpha\beta}\int d\mathbf{r}\; d\mathbf{r}'\; \theta_\alpha(\mathbf{r})\;\Phi_{\alpha\beta}(|\mathbf{r}-\mathbf{r}'|)\;\theta_\beta(\mathbf{r}') \\ &= \frac{1}{2}\sum_{\alpha\beta}\int d\mathbf{k}\; \theta_\alpha^*(\mathbf{k})\;\Phi_{\alpha\beta}(k)\;\theta_\beta(\mathbf{k}) \;.\end{aligned} \tag{10}$$

Here $\theta_\alpha(\mathbf{r}) = n(\mathbf{r}) p_\alpha\big(q_0(\mathbf{r})\big)$ and $\theta_\alpha(\mathbf{k})$ is its Fourier transform. Accordingly, $\Phi_{\alpha\beta}(k)$ is the Fourier transform of the original kernel $\Phi_{\alpha\beta}(r) = \Phi(q_\alpha, q_\beta, |\mathbf{r}-\mathbf{r}'|)$. Thus, two spatial integrals are replaced by one integral over Fourier transformed quantities, resulting in a considerable speedup. This approach also provides a convenient evaluation for $V_{\rm c}^{\rm nl}(\mathbf{r})$.

The vdW-DF functional was implemented in QUANTUM ESPRESSO version 4.3, following Eq. (10). As a result, in large systems, compute times in vdW-DF calculations are only insignificantly longer than for standard GGA functionals. The implementation uses a tabulation of the Fourier transformed kernel $\Phi_{\alpha\beta}(k)$ from Eq. (10) that is computed by an auxiliary code, `generate_vdW_kernel_table.x`, and stored in the external file `vdW_kernel_table`. The file then has to be placed either in the directory where the calculation is run or in the directory where the

corresponding pseudopotentials reside. The formalism for vdW-DF stress was derived and implemented in Ref. 63. The proper spin extension of vdW-DF, termed svdW-DF [64], was implemented in QUANTUM ESPRESSO version 5.2.1.

Although the ACFD theorem provides guidelines for the total XC functional in Eq. (7), in practice $E_{\rm xc}^0[n]$ is approximated by simple GGA-type functional forms. This has been used to improve vdW-DF—and correct the often too large binding separations found in its original form—by optimizing the exchange contribution to $E_{\rm xc}^0[n]$. The naming convention for the resulting variants is that the extension should describe the exchange functional used. In this context, the functionals vdW-DF-C09 [65], vdW-DF-obk8 [66], vdW-DF-ob86 [67], and vdW-DF-cx [68] have been developed and implemented in QUANTUM ESPRESSO. While all of these variants use the same kernel to evaluate $E_{\rm c}^{\rm nl}[n]$, advances have also been made in slightly adjusting the kernel form, which is referred to and implemented as vdW-DF2 [69]. A corresponding variant, *i.e.*, vdW-DF2-b86r [70], is also implemented. Note that vdW-DF2 uses the same kernel file as vdW-DF.

The functional VV10 [71] is related to vdW-DF, but adheres to fewer exact constraints and follows a very different design philosophy. It is implemented in QUANTUM ESPRESSO in a form called rVV10 [72] and uses a different kernel and kernel file that can be generated by running the auxiliary code `generate_rVV10_kernel_table.x`.

b. Interatomic pairwise dispersion corrections An alternative approach to accounting for dispersion forces is to add to the XC energy $E_{\rm xc}^0$ a dispersion energy, $E_{\rm disp}$, written as a damped asymptotic pairwise expression:

$$E_{\rm xc} = E_{\rm xc}^0 + E_{\rm disp}, \qquad E_{\rm disp} = -\frac{1}{2}\sum_{n=6,8,10}\sum_{I\neq J}\frac{C_{IJ}^{(n)} f_n(R_{IJ})}{R_{IJ}^n} \tag{11}$$

where I and J run over atoms, $R_{IJ} = |\mathbf{R}_I - \mathbf{R}_J|$ is the interatomic distance between atoms I and J, and $f_n(R)$ is a suitable damping function. The interatomic dispersion coefficients $C_{IJ}^{(n)}$ can be derived from fits, as in DFT-D2 [10], or calculated non-empirically, as in the Tkatchenko-Scheffler (TS-vdW) [11] and exchange-hole dipole moment (XDM) models [12, 13].

In XDM, the $C_{IJ}^{(n)}$ coefficients are calculated assuming that dispersion interactions arise from the electrostatic attraction between the electron-plus-exchange-hole distributions on different atoms [12, 13]. In this way, XDM retains the simplicity of a pairwise dispersion correction, like in DFT-D2, but derives the $C_{IJ}^{(n)}$ coefficients from the electronic properties of the system under study. The damping functions f_n in Eq. (11) suppress the dispersion interaction at short distances, and serve the purpose of making the link between the short-range correlation (provided by the XC functional) and the long-range dispersion energy, as well as mitigating erroneous behavior

from the exchange functional in the representation of intermolecular repulsion [13]. The damping functions contain two adjustable parameters, available online [73] for a number of popular density functionals. Although any functional for which damping parameters are available can be used, the functionals showing best performance when combined with XDM appear to be B86bPBE [74, 75] and PW86PBE [75, 76], thanks to their accurate modeling of Pauli repulsion [13]. Both functionals have been implemented in QUANTUM ESPRESSO since version 5.0.

In the canonical XDM implementation, recently included in QUANTUM ESPRESSO and described in detail elsewhere [77], the dispersion coefficients are calculated from the electron density, its derivatives, and the kinetic energy density, and assigned to the different atoms in the system using a Hirshfeld atomic partition scheme [78]. This means that XDM is effectively a meta-GGA functional of the dispersion energy whose evaluation cost is small relative to the rest of the self-consistent calculation. Despite the conceptual and computational simplicity of XDM, and because the dispersion coefficients depend upon the atomic environment in a physically meaningful way, the XDM dispersion correction offers good performance in the calculation of diverse properties, such as lattice energies, crystal geometries, and surface adsorption energies. XDM is especially good for modeling organic crystals and organic/inorganic interfaces. For a recent review, see Ref. 13.

The XDM dispersion calculation is turned on by specifying `vdw_corr='xdm'` and optionally selecting appropriate damping function parameters (with the `xdm_a1` and `xdm_a2` keywords). Because the reconstructed all-electron densities are required during self-consistency, XDM can be used only in combination with a PAW approach. The XDM contribution to forces and stress is not entirely consistent with the energies because the current implementation neglects the change in the dispersion coefficients. Work is ongoing to remove this limitation, as well as to make XDM available for Car-Parrinello MD, in future QUANTUM ESPRESSO releases.

In the TS-vdW approach (`vdw_corr='ts-vdw'`), all vdW parameters (which include the atomic dipole polarizabilities, α_I, vdW radii, R_I^0, and interatomic $C_{IJ}^{(6)}$ dispersion coefficients) are functionals of the electron density and computed using the Hirshfeld partitioning scheme [78] to account for the unique chemical environment surrounding each atom. This approach is firmly based on a fluctuating quantum harmonic oscillator (QHO) model and results in highly accurate $C_{IJ}^{(6)}$ coefficients with an associated error of approximately 5.5% [11]. The TS-vdW approach requires a single empirical range-separation parameter based on the underlying XC functional and is recommended in conjunction with non-empirical DFT functionals such as PBE and PBE0. For a recent review of the TS-vdW approach and several other vdW/dispersion corrections, please see Ref. 79.

The implementation of the density-dependent TS-vdW correction in QUANTUM ESPRESSO is

fully self-consistent [80] and currently available for use with norm-conserving pseudo-potentials. An efficient linear-scaling implementation of the TS-vdW contribution to the ionic forces and stress tensor allows for Born-Oppenheimer and Car-Parrinello MD simulations at the DFT+TS-vdW level of theory; this approach has already been successfully employed in long-time MD simulations of large-scale condensed-phase systems such as $(H_2O)_{128}$ [43, 52]. We note in passing that the QUANTUM ESPRESSO implementation of the TS-vdW correction also includes analytical derivatives of the Hirshfeld weights, thereby completely reflecting the change in all TS-vdW parameters during geometry/cell optimizations and MD simulations.

3. Hubbard-corrected functionals: DFT+U

Most approximate XC functionals used in modern DFT codes fail quite spectacularly on systems with atoms whose ground-state electronic structure features partially occupied, strongly localized orbitals (typically of the d or f kind), that suffer from strong self-interaction effects and a poor description of electronic correlations. In these circumstances, DFT+U is often, although not always, an efficient remedy. This method is based on the addition to the DFT energy functional $E_{\rm DFT}$ of a correction E_U, shaped on a Hubbard model Hamiltonian: $E_{\rm DFT+U} = E_{\rm DFT} + E_U$. The original implementation in QUANTUM ESPRESSO, extensively described in Refs. 81 and 82, is based on the simplest rotationally invariant formulation of E_U, due to Dudarev and coworkers [83]:

$$E_U = \frac{1}{2}\sum_I U^I \sum_{m,\sigma}\left\{ n^{I\sigma}_{mm} - \sum_{m'} n^{I\sigma}_{mm'} n^{I\sigma}_{m'm} \right\}, \tag{12}$$

where

$$n^{I\sigma}_{mm'} = \sum_{\mathbf{k},\nu} f^{\sigma}_{\mathbf{k}\nu} \langle \psi^{\sigma}_{\mathbf{k}\nu} | \phi^I_m \rangle \langle \phi^I_{m'} | \psi^{\sigma}_{\mathbf{k}\nu} \rangle, \tag{13}$$

$|\psi^{\sigma}_{\mathbf{k}\nu}\rangle$ is the valence electronic wave function for state $\mathbf{k}\nu$ of spin σ, $f^{\sigma}_{\mathbf{k}\nu}$ the corresponding occupation number, $|\phi^I_m\rangle$ is the chosen Hubbard manifold of atomic orbitals, centered on atomic site I, that may be orthogonalized or not. The presence of the Hubbard functional results in extra terms in energy derivatives such as forces, stresses, elastic constants, or force-constant (dynamical) matrices. For instance, the additional term in forces is

$$\mathbf{F}^U_{I\alpha} = -\frac{1}{2}\sum_{I,m,m',\sigma} U^I \left(\delta_{mm'} - 2n^{I\sigma}_{mm'}\right) \frac{\partial n^{I\sigma}_{mm'}}{\partial R_{I\alpha}} \tag{14}$$

where $R_{I\alpha}$ is the α component of position for atom I in the unit cell,

$$\frac{\partial n^{I\sigma}_{mm'}}{\partial R_{I\alpha}} = \sum_{\mathbf{k},\nu} f^{\sigma}_{\mathbf{k}\nu} \left(\left\langle \psi^{\sigma}_{\mathbf{k}\nu} \middle| \frac{\partial \phi_{Im}}{\partial R_{I\alpha}} \right\rangle \langle \phi_{Im'} | \psi^{\sigma}_{\mathbf{k}\nu} \rangle + \langle \psi^{\sigma}_{\mathbf{k}\nu} | \phi_{Im} \rangle \left\langle \frac{\partial \phi_{Im'}}{\partial R_{I\alpha}} \middle| \psi^{\sigma}_{\mathbf{k}\nu} \right\rangle \right). \tag{15}$$

a. Recent extensions of the formulation As a correction to the total energy, the Hubbard functional naturally contributes an extra term to the total potential that enters the KS equations. An alternative formulation [14] of the DFT+U method, recently introduced and implemented in QUANTUM ESPRESSO for transport calculations, eliminates the need of extra terms in the potential by incorporating the Hubbard correction directly into the (PAW) pseudopotentials through a renormalization of the coefficients of their non-local terms.

A simple extension to the Dudarev functional, DFT+U+J0, was proposed in Ref. 15 and used to capture the insulating ground state of CuO. In CuO the localization of holes on the d states of Cu and the consequent on-set of a magnetic ground state can only be stabilized against a competing tendency to hybridize with oxygen p states when on-site exchange interactions are precisely accounted for. A simplified functional, depending upon the on-site (screened) Coulomb interaction U and the Hund's coupling J, can be obtained from the full second-quantization formulation of the electronic interaction potential by keeping only on-site terms that describe the interaction between up to two orbitals and by approximating on-site effective interactions with the (orbital-independent) atomic averages of Coulomb and exchange terms:

$$E_{U+J} = \sum_{I,\,\sigma} \frac{U^I - J^I}{2} \mathrm{Tr}\Big[\mathbf{n}^{I\,\sigma}\,(\mathbf{1} - \mathbf{n}^{I\,\sigma})\Big] + \sum_{I,\,\sigma} \frac{J^I}{2} \mathrm{Tr}\Big[\mathbf{n}^{I\,\sigma}\,\mathbf{n}^{I\,-\sigma}\Big]. \tag{16}$$

The on-site exchange coupling J^I not only reduces the effective Coulomb repulsion between like-spin electrons as in the simpler Dudarev functional (first term of the right-hand side), but also contributes a second term that acts as an extra penalty for the simultaneous presence of anti-aligned spins on the same atomic site and further stabilizes ferromagnetic ground states.

The fully rotationally invariant scheme of Liechtenstein *et al.* [84], generalized to non-collinear magnetism and two-component spinor wave-functions, is also implemented in the current version of QUANTUM ESPRESSO. The corrective energy term for each correlated atom can be quite generally written as:

$$E_U^{full} = \frac{1}{2} \sum_{\alpha\beta\gamma\delta} U_{\alpha\beta\gamma\delta} \langle c_\alpha^\dagger c_\beta^\dagger c_\delta c_\gamma \rangle_{\mathrm{DFT}} = \frac{1}{2} \sum_{\alpha\beta\gamma\delta} \big(U_{\alpha\beta\gamma\delta} - U_{\alpha\beta\delta\gamma}\big) n_{\alpha\gamma} n_{\beta\delta}, \tag{17}$$

where the average is taken over the DFT Slater determinant, $U_{\alpha\beta\gamma\delta}$ are Coulomb integrals, and some set of orthonormal spin-space atomic functions, $\{\alpha\}$, is used to calculate the occupation matrix, $n_{\alpha\beta}$, via Eq. (13). These basis functions could be spinor wave functions of total angular momentum $j = l \pm 1/2$, originated from spherical harmonics of orbital momentum l, which is a natural choice in the presence of spin-orbit coupling. Another choice, adopted in our implementation, is to use the standard basis of separable atomic functions, $R_l(r)Y_{lm}(\theta,\phi)\chi(\sigma)$, where $\chi(\sigma)$ are spin up/down

projectors and the radial function, $R_l(r)$, is an eigenfunction of the pseudo-atom. In the presence of spin-orbit coupling, the radial function is constructed by averaging between the two radial functions $R_{l\pm1/2}$. These radial functions are read from the file containing the pseudopotential, in this case a fully-relativistic one. In this conventional basis, the corrective functional takes the form:

$$E_U^{full} = \frac{1}{2}\sum_{ijkl,\sigma\sigma'} U_{ijkl} n_{ik}^{\sigma\sigma} n_{jl}^{\sigma'\sigma'} - \frac{1}{2}\sum_{ijkl,\sigma\sigma'} U_{ijlk} n_{ik}^{\sigma\sigma'} n_{jl}^{\sigma'\sigma}, \tag{18}$$

where $\{ijkl\}$ run over azimuthal quantum number m. The second term contains a spin-flip contribution if $\sigma' \neq \sigma$. For collinear magnetism, when $n_{ij}^{\sigma\sigma'} = \delta_{\sigma\sigma'} n_{ij}^{\sigma}$, the present formulation reduces to the scheme [84] of Liechtenstein *et al.* All Coulomb integrals, U_{ijkl}, can be parameterized by few input parameters such as U (s-shell); U and J (p-shell); U, J and B (d-shell); U, J, E_2, and E_3 (f-shell), and so on. We note that if all parameters but U are set to zero, the Dudarev functional is recovered.

b. Calculation of Hubbard parameters The Hubbard corrective functional E_U depends linearly upon the effective on-site interactions, U^I. Therefore, using a proper value for these interaction parameters is crucial to obtain quantitatively reliable results from DFT+U calculations. The QUANTUM ESPRESSO implementation of DFT+U has also been the basis to develop a method for the calculation of U [81], based on linear-response theory. This method is completely ab initio and provides values of the effective interactions that are consistent with the system and with the ground state that the Hubbard functional aims at correcting. A comparative analysis of this method with other approaches proposed in the literature to compute the Hubbard interactions has been initiated in Ref. 82 and will be further refined in forthcoming publications by the same authors.

Within linear-response theory, the Hubbard interactions are the elements of an effective interaction matrix, computed as the difference between bare and screened inverse susceptibilities [81]:

$$U^I = \left(\chi_0^{-1} - \chi^{-1}\right)_{II}. \tag{19}$$

In Eq. (19) the susceptibilities χ and χ_0 measure the response of atomic occupations to shifts in the potential acting on the states of single atoms in the system. In particular, χ is defined as $\chi_{IJ} = \sum_{m\sigma} \left(dn_{mm}^{I\sigma}/d\alpha^J\right)$ and is evaluated at self consistency, while χ_0 has a similar definition but is computed before the self-consistent re-adjustment of the Hartree and XC potentials. In the current implementation these susceptibilities are computed from a series of self-consistent total energy calculations (varying the strength α of the perturbing potential over a range of values)

performed on supercells of sufficient size for the perturbations to be isolated from their periodic replicas. While easy to implement, this approach is quite cumbersome to use, requiring multiple calculations, expensive convergence tests of the resulting parameters and complex post-processing tools.

These difficulties can be overcome by using density-functional perturbation theory (DFpT) to automatize the calculation of the Hubbard parameters. The basic idea is to recast the entries of the susceptibility matrices into sums over the BZ:

$$\frac{dn^{I\sigma}_{mm'}}{d\alpha^J} = \frac{1}{N_{\mathbf{q}}} \sum_{\mathbf{q}}^{N_{\mathbf{q}}} e^{i\mathbf{q}\cdot(\mathbf{R}_l - \mathbf{R}_{l'})} \Delta^{s'}_{\mathbf{q}} n^{s\,\sigma}_{mm'} \,, \tag{20}$$

where $I \equiv (l, s)$ and $J \equiv (l', s')$, l and l' label unit cells, s and s' label atoms in the unit cell, $\mathbf{R}_l$ and $\mathbf{R}_{l'}$ are Bravais lattice vectors, and $\Delta^{s'}_{\mathbf{q}} n^{s\,\sigma}_{mm'}$ represent the (lattice-periodic) response of atomic occupations to monochromatic perturbations constructed by modulating the shift to the potential of all the periodic replica of a given atom by a wave-vector $\mathbf{q}$. This quantity is evaluated within DFpT (see Sec. II B), using linear-response routines contained in `LR_Modules` (see Sec. III D 3). This approach eliminates the need for supercell calculations in periodic systems (along with the cubic scaling of their computational cost) and automatizes complex post-processing operations needed to extract U from the output of calculations. The use of DFpT also offers the perspective to directly evaluate inverse susceptibilities, thus avoiding the matrix inversions of Eq. (19), and to calculate the Hubbard parameters for closed-shell systems, a notorious problem for schemes based on perturbations to the potential. Full details about this implementation will be provided in a forthcoming publication [85] and the corresponding code will be made available in one of the next Quantum ESPRESSO releases.

4. Adiabatic-connection fluctuation-dissipation theory

In the quest for better approximations for the unknown XC energy functional in KS-DFT, the approach based on the adiabatic connection fluctuation-dissipation (ACFD) theorem [60] has received considerable interest in recent years. This is largely due to some attractive features: (i) a formally exact expression for the XC energy in term of density linear response functions can be derived providing a promising way for a systematic improvement of the XC functional; (ii) the method treats the exchange energy exactly, thus canceling out the spurious self-interaction error present in the Hartree energy; (iii) the correlation energy is fully non local and automatically includes long-range van der Waals interactions (see Sec. II A 2 a).

Within the ACFD framework a formally exact expression for the XC energy $E_{\rm xc}$ of an electronic system can be derived:

$$E_{\rm xc} = -\frac{\hbar}{2\pi}\int_0^1 d\lambda \int d\mathbf{r}d\mathbf{r}' \frac{e^2}{|\mathbf{r}-\mathbf{r}'|} \times \left[\int_0^\infty \chi_\lambda(\mathbf{r},\mathbf{r}',iu)du + \delta(\mathbf{r}-\mathbf{r}')n(\mathbf{r})\right], \tag{21}$$

where $\hbar = h/2\pi$ and h is the Planck constant, $\chi_\lambda(\mathbf{r},\mathbf{r}';iu)$ is the density response function at imaginary frequency iu of a system whose electrons interact via a scaled Coulomb interaction, *i.e.*, $\lambda e^2/|\mathbf{r}-\mathbf{r}'|$, and are subject to a local potential such that the electronic density $n(\mathbf{r})$ is independent of λ, and is thus equal to the ground-state density of the fully interacting system ($\lambda = 1$). The XC energy, Eq. (21), can be further separated into a KS exact-exchange energy $E_{\rm xx}$, Eq. (6), and a correlation energy $E_{\rm c}$. The former is routinely evaluated as in any hybrid functional calculation (see Sec. II A 1). Using a matrix notation, the latter can be expressed in a compact form in terms of the Coulomb interaction, $v_c = e^2/|\mathbf{r}-\mathbf{r}'|$, and of the density response functions:

$$E_{\rm c} = -\frac{\hbar}{2\pi}\int_0^1 d\lambda \int_0^\infty du\ {\rm tr}\big[{\rm v_c}[\chi_\lambda({\rm iu}) - \chi_0({\rm iu})]\big]. \tag{22}$$

For $\lambda > 0$, χ_λ can be related to the noninteracting density response function χ_0 via a Dyson equation obtained from TDDFT:

$$\chi_\lambda(iu) = \chi_0(iu) + \chi_0(iu)\big[\lambda v_c + f_{\rm xc}^\lambda(iu)\big]\chi_\lambda(iu). \tag{23}$$

The exact expression of the XC kernel $f_{\rm xc}$ is unknown, and in practical applications one needs to approximate it. In the `ACFDT` package, the random phase approximation (RPA), obtained by setting $f_{\rm xc}^\lambda = 0$, and the RPA plus exact-exchange kernel (RPAx), obtained by setting $f_{\rm xc}^\lambda = \lambda f_{\rm x}$, are implemented. The evaluation of the RPA and RPAx correlation energies is based on an eigenvalue decomposition of the non-interacting response functions and of its first-order correction in the limit of vanishing electron-electron interaction [86–88]. Since only a small number of these eigenvalues are relevant for the calculation of the correlation energy, an efficient iterative scheme can be used to compute the low-lying modes of the RPA/RPAx density response functions.

The basic operation required for the eigenvalue decomposition is a number of loosely coupled DFpT calculations for different imaginary frequencies and trial potentials. Although the global scaling of the iterative approach is the same as for implementations based on the evaluation of the full response matrices (N^4), the number of operation involved is 100 to 1000 times smaller [87], thus largely reducing the global scaling pre-factor. Moreover, the calculation can be parallelized very efficiently by distributing different trial potentials on different processors or groups of processors.

In addition, the local EXX and RPA-correlation potentials can be computed through an optimized effective potential (OEP) scheme fully compatible with the eigenvalue decomposition strategy

adopted for the evaluation of the EXX/RPA energy. Iterating the energy and the OEP calculations and using an effective mixing scheme to update the KS potential, a self-consistent minimization of the EXX/RPA functional can be achieved [89].

B. Linear response and excited states without virtual orbitals

One of the key features of modern DFT implementations is that they do not require the calculation of virtual (unoccupied) orbitals. This idea, first pioneered by Car and Parrinello in their landmark 1985 paper [53] and later adopted by many groups world-wide, found its way in the computation of excited-state properties with the advent of density-functional perturbation theory (DFpT) [90–93]. DFpT is designed to deal with static perturbations and its use is therefore restricted to those excitations that can be described in the Born-Oppenheimer approximation, such as lattice vibrations. The main idea underlying DFpT is to represent the linear response of KS orbitals to an external perturbation as generic orbitals satisfying an orthogonality constraint with respect to the occupied-state manifold and a self-consistent Sternheimer equation [94, 95], rather than as linear combinations of virtual orbitals (which would require the computation of all, or a large number, of them).

Substantial progress has been made over the past decade, allowing one to extend DFpT to the dynamical regime, and thus simulate sizable portions of the optical and loss spectra of complex molecular and extended systems, without making any explicit reference to their virtual states. Although the Sternheimer approach can be easily extended to time-dependent perturbations [96–98], its use is hampered in practice by the fact that a different Sternheimer equation has to be solved for each different value of the frequency of the perturbation. When the perturbation acting on the system vanishes, the frequency-dependent Sternheimer equation becomes a non-Hermitian eigenvalue equation, whose eigenvalues are the excitation energies of the system. In the TDDFT community, this equation is known as the Casida equation [99, 100], which is the immediate translation to the DFT parlance of the time-dependent Hartree-Fock equation [101]. This approach to excited states is optimal in those cases where one is interested in a few excitations only, but can hardly be extended to continuous spectra, such as those arising in extended systems or above the ionization threshold of even finite ones. In those cases where extended portions of a continuous spectrum is required, a new method has been developed, based on the Lanczos (bi-) orthogonalization algorithm, and dubbed the Liouville-Lanczos approach to time-dependent density-functional perturbation theory (TDDFpT). This method allows one to reuse intermediate products of an

iterative process, essentially identical to that used for static perturbations, to build dynamical response functions from which spectral properties can be computed for a whole wide spectral range at once [21, 22]. A similar approach to linear optical spectroscopy was proposed later, based on the multi-shift conjugate gradient algorithm [102], instead of Lanczos. This powerful idea has been generalized to the solution of the Bethe-Salpeter equation, which is formally very similar to the eigenvalue equations arising in TDDFpT [103–105], and to the computation of the polarization propagator and self-energy operator appearing in the *GW* equations [28, 29, 106]. It is presently exploited in several components of the QUANTUM ESPRESSO distribution, as well as in other advanced implementations of many-body perturbation theory [106].

1. Static perturbations and vibrational spectroscopy

The computation of vibrational properties in extended systems is one of the traditional fields of application of DFpT, as thoroughly described, *e.g.*, in Ref. 93. The latest releases of QUANTUM ESPRESSO feature the linear-response implementation of several new functionals in the van der Waals and DFT+U families. Explicit expressions of the XC kernel, implementation details, and a thorough benchmark are reported in Ref. 107 for the first case. As for the latter, DFpT+U has been implemented for both the Dudarev [83] and DFT+U+J0 functionals [15], allowing one to account for electronic localization effects acting *selectively* on specific phonon modes at arbitrary wave-vectors, thus substantially improving the description of the vibrational properties of strongly correlated systems with respect to "standard" LDA/GGA functionals. The current implementation allows for both norm-conserving and ultrasoft pseudopotentials, insulators and metals alike, also including the spin-polarized case. The implementation of DFpT+U requires two main additional ingredients with respect to standard DFpT [108]. First, the dynamical matrix contains an additional term coming from the second derivative of the Hubbard term E_U with respect to the atomic positions (denoted λ or μ), namely:

$$\Delta^\mu(\partial^\lambda E_U) = \sum_{I\sigma mm'} U^I \left[\frac{\delta_{mm'}}{2} - n^{I\sigma}_{mm'}\right] \Delta^\mu\left(\partial^\lambda n^{I\sigma}_{mm'}\right) - \sum_{I\sigma mm'} U^I \Delta^\mu n^{I\sigma}_{mm'} \partial^\lambda n^{I\sigma}_{mm'}, \qquad (24)$$

where the notations are the same as in Eq. (12). The symbols ∂ and Δ indicate, respectively, a *bare* derivative (leaving the KS wavefunctions unperturbed) and a *total* derivative (including also linear-response contributions). Second, in order to obtain a consistent electronic density response to the atomic displacements from the DFT+U ground state, the perturbed KS potential ΔV_{SCF}

in the Sternheimer equation is augmented with the Hubbard perturbed potential $\Delta^\lambda V_U$:

$$\Delta^\lambda V_U = \sum_{I\sigma mm'} U^I \left[\frac{\delta_{mm'}}{2} - n^{I\sigma}_{mm'}\right] \times \left[|\Delta^\lambda \phi^I_{m'}\rangle\langle\phi^I_m| + |\phi^I_{m'}\rangle\langle\Delta^\lambda\phi^I_m|\right] - \sum_{I\sigma mm'} U^I \Delta^\lambda n^{I\sigma}_{mm'} |\phi^I_{m'}\rangle\langle\phi^I_m|, \tag{25}$$

where the notations are the same as in Eq. (13). The unperturbed Hamiltonian in the Sternheimer equation is the DFT+U Hamiltonian (including the Hubbard potential V_U). More implementation details will be given in a forthcoming publication [109].

Applications of DFpT+U include the calculation of the vibrational spectra of transition-metal monoxides like MnO and NiO [108], investigations of properties of materials of geophysical interest like goethite [110, 111], iron-bearing [112, 113] and aluminum-bearing bridgmanite [114]. These results feature a significantly better agreement with experiment of the predictions of various lattice-dynamical properties, including the LO-TO and magnetically-induced TO splittings, as compared with standard LDA/GGA calculations.

2. Dynamic perturbations: optical, electron energy loss, and magnetic spectroscopies

Electronic excitations can be described in terms of the dynamical charge- and spin-density susceptibilities, which are accessible to TDDFT [115, 116]. In the linear regime the TDDFT equations can be solved using first-order perturbation theory. The time Fourier transform of the charge-density response, $\tilde{n}'(\mathbf{r},\omega)$, is determined by the projection over the unoccupied-state manifold of the Fourier transforms of the first-order corrections to the one-electron orbitals, $\tilde{\psi}'_{\mathbf{k}\nu}(\mathbf{r},\omega)$, [21–24, 117]. For each band index ($\mathbf{k}\nu$), two response orbitals $x_{\mathbf{k}\nu}$ and $y_{\mathbf{k}\nu}$ can be defined as

$$x_{\mathbf{k}\nu}(\mathbf{r}) = \frac{1}{2}\hat{Q}\left(\tilde{\psi}'_{\mathbf{k}\nu}(\mathbf{r},\omega) + \tilde{\psi}'^*_{-\mathbf{k}\nu}(\mathbf{r},-\omega)\right) \tag{26}$$

$$y_{\mathbf{k}\nu}(\mathbf{r}) = \frac{1}{2}\hat{Q}\left(\tilde{\psi}_{\mathbf{k}\nu}(\mathbf{r},\omega) - \tilde{\psi}'^*_{-\mathbf{k}\nu}(\mathbf{r},-\omega)\right), \tag{27}$$

where $\hat{Q}$ is the projector on the unoccupied-state manifold. The response orbitals $x_{\mathbf{k}\nu}$ and $y_{\mathbf{k}\nu}$ can be collected in so-called *batches* $X = \{x_{\mathbf{k}\nu}\}$ and $Y = \{y_{\mathbf{k}\nu}\}$, which uniquely determine the response density matrix. In a similar way, the perturbing potential $\hat{V}'$ can be represented by the batch $Z = \{z_{\mathbf{k}\nu}\} = \{\hat{Q}\hat{V}'\psi_{\mathbf{k}\nu}\}$. Using these definitions, the linear-response equations of TDDFpT take the simple form:

$$\left(\hbar\omega - \hat{\mathcal{L}}\right)\cdot\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 0 \\ Z \end{pmatrix}, \qquad \hat{\mathcal{L}} = \begin{pmatrix} 0 & \hat{D} \\ \hat{D}+\hat{K} & 0 \end{pmatrix}, \tag{28}$$

where the super-operators $\hat{D}$ and $\hat{K}$, which enter the definition of the *Liouvillian* super-operator, $\hat{\mathcal{L}}$, are defined in terms of the unperturbed Hamiltonian and of the perturbed Hartree-plus-XC potential [21–24, 117]. This implies that a Liouvillian build costs roughly twice as much as a single iteration in time-independent DFpT. It is important to note that $\hat{D}$ and $\hat{K}$, and therefore $\hat{\mathcal{L}}$, do not depend on the frequency ω. For this reason, when in Eq. (28) the vector on the right-hand side, $(0, Z)^\top$, is set to zero, a linear eigenvalue equation is obtained (Casida's equation).

The quantum Liouville equation (28) can be seen as the equation for the response density matrix operator $\hat{\rho}'(\omega)$, namely $(\hbar\omega - \hat{\mathcal{L}}) \cdot \hat{\rho}'(\omega) = [\hat{V}', \hat{\rho}^\circ]$, where $[\cdot, \cdot]$ is the commutator and $\hat{\rho}^\circ$ is the ground-state density matrix operator. With this at hand, we can define a generalized susceptibility $\chi_{AV}(\omega)$, which characterizes the response of an arbitrary one-electron Hermitian operator $\hat{A}$ to the external perturbation $\hat{V}'$ as

$$\chi_{AV}(\omega) = \mathrm{Tr}\left[\hat{A}\hat{\rho}'(\omega)\right] = \left\langle \hat{A} \,\middle|\, (\hbar\omega - \hat{\mathcal{L}})^{-1} \cdot [\hat{V}', \hat{\rho}^\circ] \right\rangle, \tag{29}$$

where $\langle \cdot | \cdot \rangle$ denotes a scalar product in operator space. For instance, when both $\hat{A}$ and $\hat{V}'$ are one of the three Cartesian components of the dipole (position) operator, Eq. (29) gives the dipole polarizability of the system, describing optical absorption spectroscopy [21, 22]; setting $\hat{A}$ and $\hat{V}'$ to one of the space Fourier components of the electron charge-density operator would correspond to the simulation of electron energy loss or inelastic X-ray scattering spectroscopies, giving access to plasmon and exciton excitations in extended systems [25, 26]; two different Cartesian components of the Fourier transform of the spin polarization would give access to spectroscopies probing magnetic excitations (*e.g.* inelastic neutron scattering or spin-polarized electron energy loss) [118], and so on. When dealing with macroscopic electric fields, the dipole operator in periodic boundary conditions is treated using the standard DFpT prescription, as explained in Refs. [119, 120].

The Quantum ESPRESSO distribution contains several codes to solve the Casida's equation or to directly compute generalized susceptibilities according to Eq. (29) and by solving Eq. (28) using different approaches for different pairs of $\hat{A}/\hat{V}'$, corresponding to different spectroscopies. In particular, Eq. (28) can be solved iteratively using the Lanczos recursion algorithm, which allows one to avoid computationally expensive inversion of the Liouvillian. The basic principle of how matrix elements of the resolvent of an operator can be calculated using a Lanczos recursion chain has been worked out by Haydock, Heine, and Kelly [121, 122] for the case of Hermitian operators and diagonal matrix elements. The quantity of interst can be written as

$$g_v(\omega) = \left\langle v \,\middle|\, (\hbar\omega - \hat{L})^{-1}\, v \right\rangle. \tag{30}$$

A chain of vectors is defined by

$$|q_0\rangle = 0 \tag{31}$$

$$|q_1\rangle = |v\rangle \tag{32}$$

$$\alpha_n = \langle q_n|\hat{L}q_n\rangle \tag{33}$$

$$\beta_{n+1}\,|q_{n+1}\rangle = (\hat{L} - \alpha_n)\,|q_n\rangle - \beta_n\,|q_{n-1}\rangle\,, \tag{34}$$

where β_{n+1} is given by the condition $\langle q_{n+1}|q_{n+1}\rangle = 1$. The vectors $|q_n\rangle$ created by this recursive chain are orthonormal. Furthermore, the operator $\hat{L}$, written in the basis of these vectors, is tridiagonal. If one limits the chain to the M first vectors $|q_0\rangle, |q_1\rangle, \cdots, |q_M\rangle$, then the resulting representation of $\hat{L}$ is a $M \times M$ square matrix T_M which reads

$$T_M = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \ddots & \vdots \\ 0 & \beta_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \alpha_{M-1} & \beta_M \\ 0 & \cdots & 0 & \beta_M & \alpha_M \end{pmatrix}. \tag{35}$$

Using such a truncated representation of $\hat{L}$, the resolvent in Eq. (30) can be approximated as

$$g_v(\omega) \approx \left\langle v \,\middle|\, (\hbar\omega - T_M)^{-1}\, v \right\rangle . \tag{36}$$

Thanks to the tridiagonal form of T_M, the approximate resolvent can finally be written as a continued fraction

$$g_v(\omega) \approx \cfrac{1}{\hbar\omega - \alpha_1 + \cfrac{\beta_2^2}{\hbar\omega - \alpha_2 + ...}}\,. \tag{37}$$

Note that performing the recursion (31) – (34) is the computational bottleneck of this algorithm, while evaluating the continued fraction in Eq. (37) is very fast. The recursion being independent of the frequency ω, a single recursion chain yields information about any desired number of frequencies, at negligible additional computational cost. It is also important to note that at any stage of the recursion chain, only three vectors need to be kept in memory, namely $|q_{n-1}\rangle$, $|q_n\rangle$, and $|q_{n-1}\rangle$. This is a considerable advantage with respect to the direct calculation of N eigenvectors of $\hat{L}$ where all N vectors need to be kept in memory in order to enforce orthogonality.

The Liouvillian $\hat{\mathcal{L}}$ in Eq. (28) is not a Hermitian operator. For this reason, the Lanczos algorithm presented above cannot be directly applied to the calculation of the generalized susceptibility (29). There are two distinct algorithms that can be applied in the non-Hermitian case. The

non-Hermitian Lanczos biorthogonalization algorithm [22, 23] amounts to recursively applying the operator $\hat{\mathcal{L}}$ and it Hermitian conjugate $\hat{\mathcal{L}}^\dagger$ to *two* previous Lanczos vectors $|v_n\rangle$ and $|w_n\rangle$. In this way, a pair of bi-orthogonal basis sets is created. The operator $\hat{\mathcal{L}}$ can then be represented in this basis as a tridiagonal matrix, similarly to the Hermitian case, Eq. (35). The Liouvillian $\hat{\mathcal{L}}$ of TDDFT belongs to a special class of non-Hermitian operators which are called pseudo-Hermitian [24, 123]. For such operators, there exists a second recursive algorithm to compute the resolvent – *pseudo-Hermitian Lanczos algorithm* – which is numerically more stable and requires only half the numbers of operations per Lanczos step [24, 123]. Both algorithms have been implemented in QUANTUM ESPRESSO. Because of its speed and numerical stability, the use of the pseudo-Hermitian method is recommended.

This methodology has also been extended—presently only in the case of absorption spectroscopy—to employ hybrid functionals [24, 103, 104] (see Sec. II A 1). In this case the calculation requires the evaluation of the linear response of the non-local Fock potential, which is readily available from the response density matrix, represented by the *batches* of response orbitals. The corresponding hybrid-functional Liouvillian features additional terms with respect to the definition in Eq. (28), but presents a similar structure and similar mathematical properties. Accordingly, semi-local and hybrid-functional TDDFpT employ the same numerical algorithms in practical calculations.

a. Optical absorption spectroscopy The `turbo_lanczos.x` [23, 24] and `turbo_davidson.x` [24] codes are designed to simulate the optical response of molecules and clusters. `turbo_lanczos.x` computes the dynamical dipole polarizability [see Eq. (29)] of finite systems over extended frequency ranges without ever computing any eigenpairs of the Casida equation. This goal is achieved by applying a recursive non-Hermitian or pseudo-Hermitian Lanczos algorithm. The two flavours of the Lanczos algorithm implemented in `turbo_lanczos.x` are particularly suited in those cases where one is interested in the spectrum over a wide frequency range comprising a large number of individual excitations. In `turbo_davidson.x` a Davidson-like algorithm [124] is used to selectively compute a few eigenvalues and eigenvectors of $\hat{\mathcal{L}}$. This is useful when very few low-lying excited states are needed and/or when the excitation eigenvector is explicitly needed, *e.g.*, to compute ionic forces on excited potential energy surfaces, a feature that will be implemented in one of the forthcoming releases. Both `turbo_lanczos.x` and `turbo_davidson.x` are interfaced with the `Environ` module [18], to simulate the absorption spectra of complex molecules in solution using the self-consistent continuum solvation model [20] (see Sec. II E 1).

b. Electron energy loss spectroscopy The `turbo_eels.x` code [26] computes the response of extended systems to an incoming beam of electrons or X rays, aimed at simulating electron energy loss

(EEL) or inelastic X-ray scattering (IXS) spectroscopies, sensitive to collective charge-fluctuation excitations, such as plasmons. Similarly to the description of vibrational modes in a lattice by the `PHonon` package, here the perturbation can be represented as a sum of monochromatic components corresponding to different momenta, $\mathbf{q}$, and energy transferred from the incoming electrons to electrons of the sample. The quantum Liouville equation (28) in the batch representation can be formulated for individual $\mathbf{q}$ components of the perturbation, which can be solved independently [25]. The recursive Lanczos algorithm is used to solve iteratively the quantum Liouville equation, much like in the case of absorption spectroscopy. The entire EEL/IXS spectrum is obtained in an arbitrarily wide energy range (up to the core-loss region) with only one Lanczos chain. Such a numerical algorithm allows one to compute directly the diagonal element of the charge-density susceptibility, see Eq. (29), by avoiding computationally expensive matrix inversions and multiplications characteristic of standard methods based on the solution of the Dyson equation [125]. The current version of `turbo_eels.x` allows to explicitly account for spin-orbit coupling effects [126].

c. Magnetic spectroscopy The response of the system to a magnetic perturbation is described by a spin-density susceptibility matrix, see Eq. (29), labeled by the Cartesian components of the perturbing magnetic field and magnetization response, whose poles characterize spin-wave (magnon) and Stoner excitations. The methodology implemented in `turbo_eels.x` to deal with charge-density fluctuations has been generalized to spin-density fluctuations so as to deal with magnetic (spin-polarized neutron and electron) spectroscopies in extended systems. In the spin-polarized formulation of TDDFpT the time-dependent KS wave functions are two-component spinors $\{\psi^{\sigma}_{\mathbf{k}\nu}(\mathbf{r},t)\}$ (σ is the spin index), which satisfy a time-dependent Pauli-type KS equations and describe a time-dependent spin-charge-density, $n_{\sigma\sigma'}(\mathbf{r},t) = \sum_{\mathbf{k}\nu} \psi^{\sigma *}_{\mathbf{k}\nu}(\mathbf{r},t)\psi^{\sigma'}_{\mathbf{k}\nu}(\mathbf{r},t)$. Instead of using the latter quantity it is convenient to change variables and use the charge density $n(\mathbf{r},t) = \sum_{\sigma} n_{\sigma\sigma}(\mathbf{r},t)$ and the spin density $\mathbf{m}(\mathbf{r},t) = \mu_B \sum_{\sigma\sigma'} \boldsymbol{\sigma}_{\sigma\sigma'} n_{\sigma'\sigma}(\mathbf{r},t)$, where μ_B is the Bohr magneton and $\boldsymbol{\sigma}$ is the vector of Pauli matrices. In the linear-response regime, the charge- and spin-density response $n'(\mathbf{r},t)$ and $\mathbf{m}'(\mathbf{r},t)$ are coupled via the scalar and magnetic XC response potentials $V'_{\mathrm{xc}}(\mathbf{r},t)$ and $\mathbf{B}'_{\mathrm{xc}}(\mathbf{r},t)$, which are treated on a par with the Hartree response potential $V'_{\mathrm{H}}(\mathbf{r},t)$, depending only on $n'(\mathbf{r},t)$, and which all enter the linear-response time-dependent Pauli-type KS equations. The lack of time-reversal symmetry in the ground state means that the TDDFpT equations have to be generalized to treat KS spinors at $\mathbf{k}$ and $-\mathbf{k}$ and various combinations with the $\mathbf{q}$ vector. Moreover, this also implies that no rotation of batches is possible, as in Eqs. (26) and (27), and a generalization of the Lanczos algorithm to complex arithmetics is required. At variance with the cumbersome Dyson's equation approach, which requires the sep-

arate calculation and coupling of charge-charge, spin-spin, and charge-spin independent-electron polarizabilities, in our approach the coupling between spin and charge fluctuations is naturally accounted for via Lanczos chains for the spinor KS response orbitals. The current implementation supports general non-collinear spin-density distributions, which allows us to account for spin-orbit interaction and magnetic anisotropy. All the details about the present formalism will be given in a forthcoming publication [118] and the corresponding code will be made available in one of the next QUANTUM ESPRESSO releases.

3. Many-body perturbation theory

Many-body perturbation theory refers to a set of computational methods, based on quantum field theory, that are designed to calculate electronic and optical excitations beyond standard DFT [125]. The most popular among such methods are the *GW* approximation and the Bethe-Salpeter equation (BSE) approach. The former is intended to calculate accurate quasiparticle excitations, *e.g.*, ionization energies and electron affinities in molecules, band structures in solids, and accurate band gaps in semiconductor and insulators. The latter is employed to study optical excitations by including electron-hole interactions.

In the *GW* method the XC potential of DFT is corrected using a many-body self-energy consisting of the product of the electron Green's function G and the screened Coulomb interaction W [127, 128], which represents the lowest-order term in the diagrammatic expansion of the exact electron self-energy. In the vast majority of *GW* implementations, the evaluation of G and W requires the calculation of both occupied and unoccupied KS eigenstates. The convergence of the resulting self-energy correction with respect to the number of unoccupied states is rather slow, and in many cases it constitutes the main bottleneck in the calculations. During the past decade there has been a growing interest in alternative techniques which only require the calculation of occupied electronic states, and several computational strategies have been developed [29, 129–131]. The common denominator to all these strategies is that they rely on linear-response DFpT and the Sternheimer equation, as in the `PHonon` package.

In QUANTUM ESPRESSO the *GW* approximation is realized based on a DFpT representation of response and self-energy operators, thus avoiding any explicit reference to unoccupied states. There are two different implementations: the `GWL` (*GW*-Lanczos) package [28, 29] and the `SternheimerGW` package [30]. The former focuses on efficient *GW* calculations for large systems (including disordered solids, liquids, and interfaces), and also supports the calculations of optical

spectra via the Bethe-Salpeter approach [105]. The latter focuses on high-accuracy calculations of band structures, frequency-dependent self-energies, and quasi-particle spectral functions for crystalline solids. In addition to these, the WEST code [106], not part of the QUANTUM ESPRESSO distribution, relies on QUANTUM ESPRESSO as an external library to perform similar tasks and to achieve similar goals.

a. `GWL` The `GWL` package consists of four different codes. The `pw4gww.x` code reads the KS wave-functions and charge density previously calculated by `PWscf` and prepares a set of data which are used by code `gww.x` to perform the actual *GW* calculation. While `pw4gww.x` uses the plane-wave representation of orbitals and charges and the same QUANTUM ESPRESSO environment as all other linear response codes, `gww.x` does not rely on any specific representation of the orbitals. Its parallelization strategy is based on the distribution of frequencies. Only a few basic routines, such as the MPI drivers, are common with the rest of QUANTUM ESPRESSO.

`GWL` supports three different basis sets for representing polarisability operators: *i)* plane wave-basis set, defined by an energy cutoff; *ii)* the basis set formed by the most important eigenvectors (*i.e.*, corresponding to the highest eigenvalues) of the actual irreducible polarisability operator at zero frequency calculated through linear response; *iii)* the basis set formed by the most important eigenvectors of an approximated polarisability operator. The last choice permits the control of the balance between accuracy and dimension of the basis. The *GW* scheme requires the calculation of products in real space of KS orbitals with vectors of the polarisability basis. These are represented in `GWL` through dedicated additional basis sets of reduced dimensions.

`GWL` supports only the $\Gamma-$point sampling of the BZ and considers only real wave-functions. However, ordinary $\mathbf{k}$-point sampling of the BZ can be used for the long-range part of the (symmetrized) dielectric matrix. These terms are calculated by the `head.x` code. In this way reliable calculations for extended materials can be performed using quite small simulation cells (with cell edges of the order of 20 Bohr). Self-consistency is implemented in `GWL`, although limited to the quasi-particle energies; the so-called vertex term, arising in the diagrammatic expansion of the self-energy, is not yet implemented.

Usually ordinary *GW* calculations for transition elements require the explicit inclusion of semi-core orbitals in the valence manifold, resulting in a significantly higher computational cost. To cope with this issue, an approximate treatment of semicore orbitals has been introduced in `GWL` as described in Ref. 132. In addition to collinear spin polarization, `GWL` provides a fully relativistic non collinear implementation relying on the scalar relativistic calculation of the screened Coulomb interactions [133].

The `bse.x` code of the `GWL` package performs BSE calculations and permits to evaluate either the entire frequency-dependent complex dielectric function through the Lanczos algorithm or a discrete set of excited states and their energies through a conjugate gradient minimization. In contrast to ordinary implementations, `bse.x` scales as N^3 instead of N^4 with respect to the system size N (*e.g.*, the number of atoms) thanks to the use of maximally localized Wannier functions for representing the valence manifold [105]. The `bse.x` code, apart from reading the screened Coulomb interaction at zero frequency from a `gww.x` calculation, works as a separate code and uses the QUANTUM ESPRESSO environment. Therefore it could be easily be interfaced with other *GW* codes.

b. `SternheimerGW` The `SternheimerGW` package calculates the frequency-dependent *GW* self-energy and the corresponding quasiparticle corrections at arbitrary **k**-points in the BZ. This feature enables accurate calculations of band structures and effective masses without resorting to interpolation. The availability of the complete *GW* self-energy (as opposed to the quasiparticle shifts) makes it possible to calculate spectral functions, for example including plasmon satellites [134]. The spectral function can be directly compared to angle-resolved photoelectron spectroscopy (ARPES) experiments. In `SternheimerGW` the screened Coulomb interaction W is evaluated for wave-vectors in the irreducible BZ by exploiting crystal symmetries. Calculations of G and W for multiple frequencies ω rely on the use of multishift linear system solvers that construct solutions for all frequencies from the solution of a single linear system [131, 135]. This method is closely related to the Lanczos approach. The convolution in the frequency domain required to obtain the self energy from G and W can be performed either on the real axis or the imaginary axis. Padé functions are employed to perform approximate analytic continuations from the imaginary to the real frequency axis; the standard Godby-Needs plasmon pole model is also available to compare with literature results. The stability and portability of the `SternheimerGW` package are verified via a test-suite and a Buildbot test-farm (see Sec. III F).

C. Other spectroscopies

1. `QE-GIPAW`*: Nuclear magnetic and electron paramagnetic resonance*

The `QE-GIPAW` package allows for the calculation of various physical parameters measured in nuclear magnetic resonance (NMR) and electron paramagnetic resonance (EPR) spectroscopies. These encompass (i) NMR chemical shift tensors and magnetic susceptibility, (ii) electric field

gradient (EFG) tensors, (iii) EPR g-tensor, and (iv) hyperfine coupling tensor.

In `QE-GIPAW`, the NMR and EPR parameters are obtained from the orbital linear response to an external uniform magnetic field. The response depends critically upon the exact shape of the electronic wavefunctions near the nuclei. Thus, all-electron wavefunctions are reconstructed from the pseudo-wavefunctions in a gauge- and translationally invariant way using the gauge-including projector augmented-wave (GIPAW) method [136]. The description of a uniform magnetic field within periodic boundary conditions is achieved by the long-wavelength limit ($q \ll 1$) of a sinusoidally modulated field in real space. In practice, for each $\mathbf{k}$ point, we calculate the first order change of the wavefunctions at $\mathbf{k}+\mathbf{q}$, where $\mathbf{q}$ runs over a star of 6 points. The magnetic susceptibility and the induced orbital currents are then evaluated by finite differences, in the limit of small q. The induced magnetic field at the nucleus, which is the central quantity in NMR, is obtained from the induced current via the Biot-Savart law. In `QE-GIPAW`, the NMR orbital chemical shifts and magnetic susceptibility can be calculated both for insulators [34] and for metals [137] (the additional contribution for metals coming from the spin-polarization of valence electrons, namely the Knight shift, can also be computed but it is not yet ready for production at the time of writing). Similarly to the NMR chemical shift, the EPR g-tensor is calculated as the cross product of the induced current with the spin-orbit operator [138].

For the quantities defined in zero magnetic field, namely the EFG, Mössbauer and relativistic hyperfine tensors, the usual PAW reconstruction of the wavefunctions is sufficient and these are computed as described in Refs. [139, 140]. The hyperfine Fermi contact term, proportional to the spin density evaluated at the nuclear coordinates, however requires the relaxation of the core electrons in response to the magnetization of valence electrons. We implemented the core relaxation in perturbation theory, according to Ref. 141. Basically we compute the spherically averaged PAW spin density around each atom. Then we compute the change of the XC potential, ΔV_{XC}, on a radial grid, and compute in perturbation theory the core radial wavefunction, both for spin up and spin down. This provides an extra contribution to the Fermi contact, in most cases opposite in sign to and as large as that of valence electrons.

By combining the quadrupole coupling constants derived from EFG tensors and hyperfine splittings, electron nuclear double resonance (ENDOR) frequencies can be calculated. Applications highlighting all these features of the `QE-GIPAW` package can be found in Ref. 142. These quantities are also needed to compute NMR shifts in paramagnetic systems, like novel cathode materials for Li batteries [143]. Previously restricted to norm-conserving pseudopotentials only, all features are now applicable using any kind of pseudization scheme and to PAW, following the theory described

in [144]. The use of smooth pseudopotentials allows for the calculation of chemical shifts in systems with several hundreds of atoms [145].

The starting point of all `QE-GIPAW` calculations is a previous calculation of KS orbitals via `PWscf`. Hence, much like other linear response routines, the `QE-GIPAW` code uses many subroutines of `PWscf` and of the linear response module. As usually done in linear response methods, the response of the unoccupied states is calculated using the completeness relation between occupied and unoccupied manifolds [146]. As a result, for insulating as well as metallic systems, the linear response of the system is efficiently obtained without the need to include virtual orbitals.

As an alternative to linear response method, the theory of orbital magnetization via Berry curvature [147, 148] can be used to calculate the NMR [149] and EPR parameters [150]. Specifically, it can be shown that the variation of the orbital magnetization M^{orb} with respect to spin flip is directly related to the g-tensor: $g_{\mu\nu} = g_e - \frac{2}{\alpha S}\mathbf{e}_\mu \cdot \mathbf{M}^{\rm orb}(\mathbf{e}_\nu)$, where $g_e = 2.002319$, α is the fine structure constant, S is the total spin, $\mathbf{e}$ are Cartesian unit vectors, provided that the spin-orbit interaction is explicitly considered in the Hamiltonian. This *converse* method of calculating the g-tensor has been implemented in an older version of `QE-GIPAW`. It is especially useful in critical cases where linear response is not appropriate, *e.g.*, systems with quasi-degenerate HOMO-LUMO levels. A demonstration of this method applied to delocalized conduction band electrons can be found in Ref. 151.

The converse method will be shortly ported into the current `QE-GIPAW` and we will explore the possibility of computing in a converse way the Knight shift as the response to a small nuclear magnetic dipole. The present version of the code allows for parameter-free calculations of g-tensors, hyperfine splittings, and ENDOR frequencies also for systems with total spin $S > 1/2$. Such triplet or even higher-spin states give rise to additional spin-spin interactions, that can be calculated within the magnetic dipole-dipole interaction approximation. This interaction results in a fine structure which can be measured in zero magnetic field. This so-called zero-field splitting is being implemented following the methodology described in Ref. [152].

2. `XSpectra`*: $L_{2,3}$ X-ray absorption edges*

The `XSpectra` code [153, 154] has been extended to the calculation of X-ray absorption spectra at the $L_{2,3}$-edges [155]. The `XSpectra` code uses the self-consistent charge density produced by `PWscf` and acts as a post processing tool [153, 154, 156]. The spectra are calculated for the L_2 edge, while the L_3 edge is obtained by multiplying by two (single-particle statistical branching ratio) the

L_2 edge spectrum and by shifting it by the value of the spin-orbit splitting of the $2p_{1/2}$ core levels of the absorbing atom. The latter can be taken either from a DFT relativistic all-electron calculation on the isolated atom, or from experiments.

In practice, the L_3 edge is obtained from the L_2 with the `spectra_correction.x` tool. Such tool contains a table of experimental $2p$ spin-orbit splittings for all the elements. In addition to computing L_3 edges, `spectra_correction.x` allows one to remove states from the spectrum below a certain energy, and to convolute the calculated spectrum with more elaborate broadenings. These operations can be applied to any edge.

To evaluate the X-ray absorption spectrum for a system containing various atoms of the same species but in different chemical environments, one has to sum the contribution by each atom. This could be the case, for example of an organic molecule containing various C atoms in inequivalent sites. Such individual contributions can be computed separately by `XSpectra`, and the tool `molecularnexafs.x` allows one to perform their weighted sum taking into account the proper energy reference (initial and final state effects) [157, 158]. One should in fact notice that the reference for initial state effects will depend upon the environment (*e.g.*, the vacuum level for gas phase molecules, or the Fermi level for molecules adsorbed on a metal).

D. Other lattice-dynamical and thermal properties

1. `thermo_pw`: *Thermal properties from the quasi-harmonic approximation*

`thermo_pw` [31] is a collection of codes aimed at computing various thermodynamical quantities in the quasi-harmonic approximation. The key ingredient is the vibrational contribution, F_{ph}, to the Helmholtz free energy at temperature T:

$$F_{ph} = k_B T \sum_{\mathbf{q},\nu} \ln \left[2 \sinh \left(\frac{\hbar \omega_{\mathbf{q}\nu}}{2 k_B T} \right) \right], \tag{38}$$

where $\omega_{\mathbf{q},\nu}$ are phonon frequencies at wave-vector $\mathbf{q}$, k_B is the Boltzmann constant. `thermo_pw` works by calling QUANTUM ESPRESSO routines from `PWscf` and `PHonon`, that perform one of the following tasks: *i)* compute the DFT total energy and possibly the stress for a given crystal structure; *ii)* compute for the same system the electronic band structure along a specified path; *iii)* compute for the same system phonon frequencies at specified wave-vectors. Using such quantities, `thermo_pw` can calculate numerically the derivatives of the free energy with respect to the external parameters (*e.g.*, different volumes). Several calls to such routines, with slightly different geome-

tries, are typically needed in a run. All such tasks can be independently performed on different groups of processors (called *images*).

When the tasks carried out by different images require approximately the same time, or when the amount of numerical work needed to accomplish each task is easy to estimate a priori, it would be possible to statically assign tasks to images at the beginning of the run so that images do not need to communicate during the run. However, such conditions are seldom met in `thermo_pw` and therefore it would be impossible to obtain a good load balancing between images. `thermo_pw` takes advantage of an engine that controls these different tasks in an asynchronous way, dynamically assigning tasks to the images at run time.

At the core of `thermo_pw` there is a module `mp_asyn`, based on MPI routines, that allows for asynchronous communication between different images. One of the images is the "master" and assigns tasks to the other images (the "slaves") as soon as they communicate that they have accomplished the previously assigned task. The master image also accomplishes some tasks but once in a while, with negligible overhead, it checks if there is an image available to do some work; if so, it assigns to it the next task to do. The code stops when the master recognizes that all the tasks have been done and communicates this information to the slaves. The routines of this communication module are quite independent of the `thermo_pw` variables and in principle can be used in conjunction with other codes to set up complex workflows to be executed in a massively parallel environment. It is assumed that each processor of each image reads the same input and that the only information that the image needs to synchronize with the other images is which tasks to do. The design of `thermo_pw` makes it easily extensible to the calculation of new properties in an incremental way.

2. `thermal2`*: phonon-phonon interaction and thermal transport*

Phonon-phonon interaction (ph-ph) plays a role in different physical phenomena: phonon lifetime (and its inverse, the linewidth), phonon-driven thermal transport in insulators or semi-metals, thermal expansion of materials. Ph-ph is possible because the harmonic Hamiltonian of ionic motion, of which phonons are stationary states, is only approximate. At first order in perturbation theory we have the third derivative of the total energy with respect to three phonon perturbations, which we compute *ab-initio*. This calculation is performed by the 3̣q code via the $2n + 1$ theorem [32, 159]. The 3̣q code is an extension of the old D3 code, which only allowed the calculation of zone-centered phonon lifetimes and of thermal expansion. The current ver-

sion can compute the three-phonon matrix element of arbitrary wave-vectors $D^{(3)}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = \partial^3 E/\partial u_{\mathbf{q}_1} \partial u_{\mathbf{q}_2} \partial u_{\mathbf{q}_3}$, where u are the phonon displacement patterns, the momentum conservation rule imposes $\mathbf{q}_1+\mathbf{q}_2+\mathbf{q}_3 = 0$. The current version of the code can treat any kind of crystal geometry, metals and insulators, both local density and gradient-corrected functionals, and multi-projector norm-conserving pseudopotentials. Ultrasoft pseudopotentials, PAW, spin polarization and non-collinear magnetization are not implemented. Higher order derivative of effective charges[160] are not implemented.

The ph-ph matrix elements, computed from linear response, can be transformed, via a generalized Fourier transform, to the real-space three-body force constants which could be computed in a supercell by finite difference derivation:

$$D^{(3)}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = \sum_{\mathbf{R}', \mathbf{R}''} e^{-2i\pi(\mathbf{R}'\cdot\mathbf{q}_2 + \mathbf{R}''\cdot\mathbf{q}_3)} F^{(3)}(\mathbf{0}, \mathbf{R}', \mathbf{R}''), \tag{39}$$

where $F^3(\mathbf{0}, \mathbf{R}', \mathbf{R}'') = \partial^3 E/\partial\tau\partial(\tau' + \mathbf{R}')\partial(\tau'' + \mathbf{R}'')$ is the derivative of the total energy w.r.t. nuclear positions of ions with basis τ, τ', τ'' from the unit cells identified by direct lattice vectors $\mathbf{0}$ (the origin), $\mathbf{R}'$ and $\mathbf{R}''$. The sum over $\mathbf{R}'$ and $\mathbf{R}''$ runs, in principle, over all unit cells, however the terms of the sum quickly decay as the size of the triangle $\mathbf{0} - \mathbf{R}' - \mathbf{R}''$ increases. The real-space finite-difference calculation, performed by some external softwares[168], has some advantages: it is easier to implement and it can readily include all the capabilities of the self-consistent code; on the other hand it is much more computationally expensive than the linear-response method we use, its cost scaling with the cube of the supercell volume, or the 9^{th} power of the number of side units of an isotropic system. We use the real-space formalism to apply the sum rule corresponding to translational invariance to the matrix elements. This is done with an iterative method that alternatively enforces the sum rule on the first matrix index and restores the invariance on the order of the derivations. We also use the real-space force constants to Fourier-interpolate the ph-ph matrices on a finer grid, assuming that the contribution from triangles $\mathbf{0} - \mathbf{R}' - \mathbf{R}''$ which we have not computed is zero; it is important in this stage to consider the periodicity of the system.

From many-body theory we get the first-order phonon linewidth[161] (γ_ν) of mode ν at $\mathbf{q}$, which is a sum over all the possible $N_\mathbf{q}$'s final and initial states ($\mathbf{q}'$,ν',ν'') with conservation of energy ($\hbar\omega$) and momentum ($\mathbf{q}'' = -\mathbf{q} - \mathbf{q}'$), Bose-Einstein occupations ($n(\mathbf{q}, \nu) = (\exp(\hbar\omega_{\mathbf{q},\nu}/k_B T) - 1)^{-1}$) and an amplitude $V^{(3)}$, proportional to the $D^{(3)}$ matrix element but renormalized with phonon

energies and ion masses:

$$\gamma_{\mathbf{q},\nu} = \frac{\pi}{\hbar^2 N_{\mathbf{q}}} \sum_{\mathbf{q}',\nu',\nu''} \left| V^{(3)}(\mathbf{q}\nu, \mathbf{q}'\nu', \mathbf{q}''\nu'') \right| \times \tag{40}$$
$$\left[(1 + n_{\mathbf{q}',\nu'} + n_{\mathbf{q}'',\nu''}) \delta(\omega_{\mathbf{q},\nu} - \omega_{\mathbf{q}',\nu'} - \omega_{\mathbf{q}'',\nu''}) + 2(n_{\mathbf{q}',\nu'} - n_{\mathbf{q}'',\nu''}) \delta(\omega_{\mathbf{q},\nu} + \omega_{\mathbf{q}',\nu'} - \omega_{\mathbf{q}'',\nu''}) \right].$$

This sum is computed in the `thermal2` suite, which is bundled with 3q. A similar expression can be written for the phonon scattering probability which appears in the Boltzmann transport equation. In order to properly converge the integral of the Dirac delta function, we express it as finite-width Gaussian function and use an interpolation grid. This equation can be solved either exactly or in the single mode approximation (SMA) [162]. The SMA is a good tool at temperatures comparable to or larger than the Debye temperature, but is known to be inadequate at low temperatures [163, 164] or in the case of 2D materials [165–167]. The exact solution is computed using a variational form, minimized via a preconditioned conjugate gradient algorithm, which is guaranteed to converge, usually in less than 10 iterations [33].

On top of intrinsic ph-ph events, the `thermal2` codes can also treat isotopic disorder and substitution effects and finite transverse dimension using the Casimir formalism. In addition to using our force constants from DFpT, the code supports importing 3-body force constants computed via finite differences with the `thirdorder.py` code [168]. Parallelization is implemented with both MPI (with great scalability up to thousand of CPUs) and OpenMP (optimal for memory reduction).

3. `EPW`*: Electron-phonon coefficients from Wannier interpolation*

The electron-phonon-Wannier (`EPW`) package is designed to calculate electron-phonon coupling using an ultra-fine sampling of the BZ by means of Wannier interpolation. `EPW` employs the relation between the electron-phonon matrix elements in the Bloch representation $g_{mn\nu}(\mathbf{k}, \mathbf{q})$, and in the Wannier representation, $g_{ij\kappa\alpha}(\mathbf{R}, \mathbf{R}')$ [169],

$$g_{mn}(\mathbf{k}, \mathbf{q}) = \sum_{\mathbf{R},\mathbf{R}'} e^{i(\mathbf{k}\cdot\mathbf{R} + \mathbf{q}\cdot\mathbf{R}')} \sum_{ij\kappa\alpha} U_{mi\mathbf{k}+\mathbf{q}}\, g_{ij\kappa\alpha}(\mathbf{R}, \mathbf{R}')\, U^{\dagger}_{jn\mathbf{k}}\, u_{\kappa\alpha,\mathbf{q}\nu}, \tag{41}$$

in order to interpolate from coarse **k**-point and **q**-point grids into dense meshes. In the above expression **k** and **q** represent the electron and phonon wave-vector, respectively, the indices m, n and i, j refer to Bloch states and Wannier states, respectively, and $\mathbf{R}, \mathbf{R}'$ are direct lattice vectors. The matrices $U_{mi\mathbf{k}}$ are unitary transformations and the vector $u_{\kappa\alpha,\mathbf{q}\nu}$ is the displacements of the atom κ along the Cartesian direction α for the phonon of wavevector **q** and branch ν. The

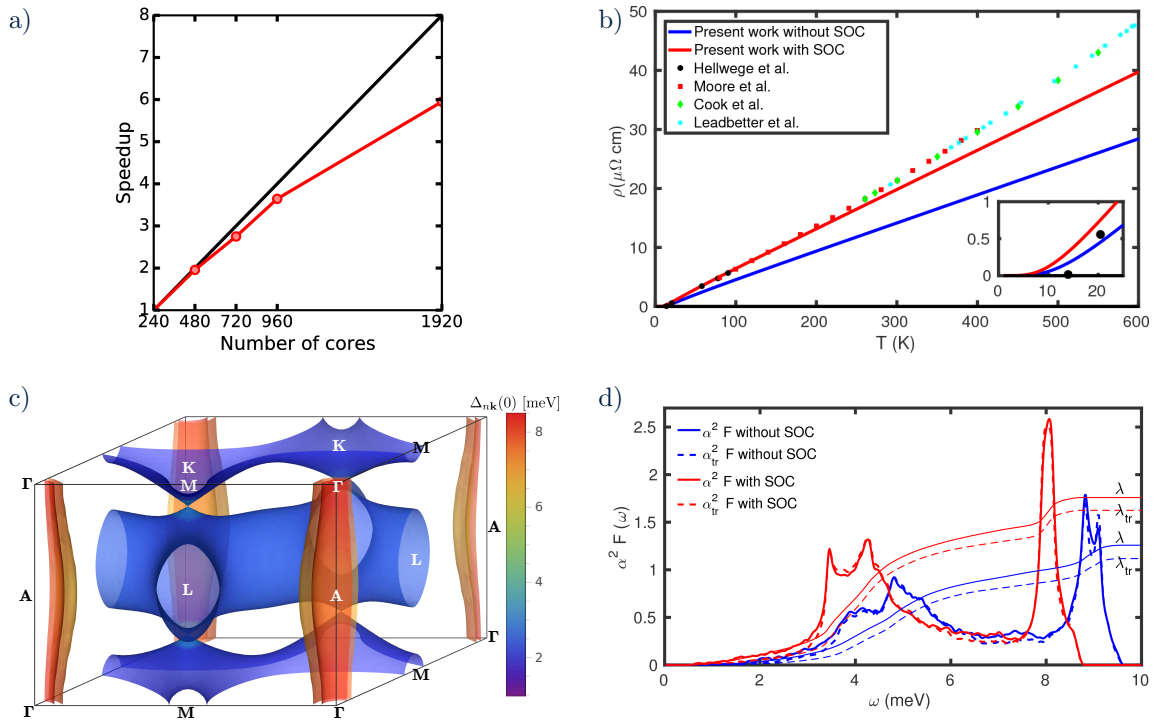

FIG. 2. Examples of calculations that can be performed using `EPW`. (a) Parallel scaling of `EPW` on ARCHER Cray XC30. This example corresponds to the calculation of electron-phonon couplings for wurtzite GaN. The parallelization is performed over **k**-points using MPI. (b) Calculated temperature-dependent resistivity of Pb by including/neglecting spin-orbit coupling. Reproduced from Ref. 27. (c) Calculated superconducting gap function of MgB_2, color-coded on the Fermi surface. Reproduced from Ref. 27. (d) Eliashberg spectral function α^2F and transport spectral function $\alpha^2F_{\rm tr}$ of Pb. Reproduced from Ref. 27.

interpolation is performed with *ab initio* accuracy by relying on the localization of maximally-localized Wannier functions [170]. During its execution `EPW` invokes the `Wannier90` software [171] in library mode in order to determine the matrices $U_{mi\mathbf{k}}$ on the coarse **k**-point grid.

`EPW` can be used to compute the following physical properties: the electron and phonon linewidths arising from electron-phonon interactions; the scattering rates of electrons by phonons; the total, averaged electron-phonon coupling strength; the electrical resistivity of metals, see Fig. 2(b); the critical temperature of electron-phonon superconductors; the anisotropic superconducting gap within the Eliashberg theory, see Fig. 2(c); the Eliashberg spectral function, transport spectral function, see Fig. 2(d) and the nesting function. The calculation of carrier mobilities using the Boltzmann transport equation in semiconductors is under development.

The `epw.x` code exploits crystal symmetry operations (including time reversal) in order to limit the number of phonon calculations to be performed using `PHonon` to the irreducible wedge of the BZ.

The code supports calculations of electron-phonon couplings in the presence of spin-orbit coupling. The current version does not support spin-polarized calculations, ultrasoft pseudopotentials nor the PAW method. As shown in Fig. 2(a), `epw.x` scales reasonably up to 2,000 cores using MPI. A test farm (see Sec. III F) was set up to ensure portability of the code on many architecture and compilers. Detailed information about the `EPW` package can be found in Ref. 27.

4. Non-perturbative approaches to vibrational spectroscopies

Although DFpT is in many ways the state of the art in the simulation of vibrational spectroscopies in extended systems, and in fact one of defining features of QUANTUM ESPRESSO, it is sometimes convenient to compute lattice-dynamical properties, the response to macroscopic electric fields, or combinations thereof (such as *e.g.*, the infrared or Raman activities), using non-perturbative methods. This is so because DFpT requires the design of dedicated codes, which have to be updated and maintained separately, and which therefore not always follow the pace of the implementation of new features, methods, and functionals (such as *e.g.*, DFT+U, vdW-DF, hybrid functionals, or ACBN0 [172]) in their ground-state counterparts. Such a non-perturbative approach is followed in the `FD` package, which implements the "frozen-phonon" method for the computation of phonons and vibrational spectra: the interatomic Force Constants (IFCs) and electronic dielectric constant are computed as finite differences of forces and polarizations, with respect to finite atomic displacements or external electric fields, respectively [173, 174]. IFC's are computed in two steps: first, code `fd.x` generates the symmetry-independent displacements in an appropriate supercell; after the calculations for the various displacements are completed, code `fd_ifc.x` reads the forces and generates IFC's. These are further processed in `matdyn.x`, where non-analytical long-ranged dipolar terms are subtracted out from the IFCs following the recipe of Ref. 175. The calculation of dielectric tensor and of the Born effective charges proceeds from the evaluation of the electronic susceptibility following the method proposed by Umari and Pasquarello [174], where the introduction of a non local energy functional $E^{\boldsymbol{\mathcal{E}}}_{tot}[\psi] = E^0[\psi] - \boldsymbol{\mathcal{E}} \cdot (\mathbf{P}^{ion} + \mathbf{P}^{el}[\psi])$ allows to compute the electronic structure for periodic systems under finite homogeneous electric fields. E^0 is the ground state total energy in the absence of external electric fields; $\mathbf{P}^{ion}$ is the usual ionic polarization, and $\mathbf{P}^{el}$ is given as a Berry phase of the manifold of the occupied bands [176]. The high-frequency dielectric tensor ϵ^∞ is then computed as $\epsilon^\infty_{ij} = \delta_{i,j} + 4\pi\chi_{ij}$, while Born effective-charge tensors $Z^*_{I,ij}$ are obtained as the polarization induced along the direction i by a unit displacement of the I-th atom in the j direction; alternatively, as the force induced on atom I by an applied electric field,

$\mathcal{E}$.

The calculation of the Raman spectra proceeds along similar lines. Within the finite-field approach, the Raman tensor is evaluated in terms of finite differences of atomic forces in the presence of two electric fields [177]. In practice, the tensor $\chi^{(1)}_{ijIk}$ is obtained from a set of calculations combining finite electric fields along different Cartesian directions. $\chi^{(1)}_{ijIk}$ is then symmetrized to recover the full symmetry of the structure under study.

E. Multi-scale modeling

1. `Environ`: *Self-Consistent Continuum Solvation embedding model*

Continuum models are among the most popular multiscale approaches to treat solvation effects in the quantum-chemistry community [178]. In this class of models, the degrees of freedom of solvent molecules are effectively integrated out and their statistically-averaged effects on the solute are mimicked by those of a continuous medium surrounding a cavity in which the solute is thought to dwell. The most important interaction usually handled with continuum models is the electrostatic one, in which the solvent is described as a dielectric continuum characterized by its experimental dielectric permittivity.

Following the original work of Fattebert and Gygi [179] , a new class of continuum models was designed, in which a smooth transition from the QM-solute region to the continuum-environment region of space is introduced and defined in terms of the electronic density of the solute. The corresponding free energy functional is optimized using a fully variational approach, leading to a generalized Poisson equation that is solved via a multi-grid solver[179]. This approach, ideally suited for plane-wave basis sets and tailored for MD simulations, has been featured in the QUANTUM ESPRESSO distribution since v. 4.1. This approach was recently revised[18], by defining an optimally smooth QM/continuum transition, reformulated in terms of iterative solvers[180] and extended to handle in a compact and effective way non-electrostatic interactions [18]. The resulting self-consistent continuum solvation (SCCS) model, based on a very limited number of physically justified parameters, allows one to reproduce experimental solvation energies for aqueous solutions of neutral [18] and charged[181] species with accuracies comparable to or higher than state-of-the-art quantum-chemistry packages.

The SCCS model involves different embedding terms, each representing a specific interaction with an external continuum environment and contributing to the total energy, KS potential, and

interatomic forces of the embedded QM system. Every contribution may depend explicitly on the ionic (rigid schemes) and/or electronic (self-consistent or soft schemes) degrees of freedom of the embedded system. All the different terms are collected in the stand-alone `Environ` module [182]. The present discussion refers to release 0.2 of `Environ`, which is compatible with QUANTUM ESPRESSO starting from versions 5.1. The module requires a separate input file with the specifications of the environment interactions to be included and of the numerical parameters required to compute their effects. Fully parameterized and tuned SCCS environments, *e.g.*, corresponding to water solutions for neutral and charged species, are directly available to the users. Otherwise individual embedding terms can be switched on and tuned to the specific physical conditions of the required environment. Namely, the following terms are currently featured in `Environ`:

- Smooth continuum dielectric, with the associated generalized Poisson problem solved via a direct iterative approach or through a preconditioned conjugate gradient algorithm [180].
- Electronic enthalpy functional, introducing an energy term proportional to the quantum-volume of the system and able to describe finite systems under the effect of an applied external pressure[183].
- Electronic cavitation functional, introducing an energy term proportional to the quantum-surface able to describe free energies of cavitation and other surface-related interaction terms[184].
- Parabolic corrections for periodic boundary conditions in aperiodic and partially periodic (slab) systems [19, 185].
- Fixed dielectric regions, allowing for the modelling of complex inhomogenous dielectric environments.
- Fixed Gaussian-smoothed distributions of charges, allowing for a simplified modelling of countercharge distributions, *e.g.*, in electrochemical setups.

Different packages of the QUANTUM ESPRESSO distribution have been interfaced with the `Environ` module, including `PWscf`, `CP`, `PWneb`, and `turboTDDFT`, the latter featuring a linear-response implementation of the SCCS model (see Sec.II B 2). Moreover, continuum environment effects are fully compatible with the main features of QUANTUM ESPRESSO, and in particular, with reciprocal space integration and smearing for metallic systems, with both norm-conserving and ultrasoft pseudopotentials and PAW, with all XC functionals.

2. QM-MM

QM-MM was implemented in v.5.0.2 using the method documented in Ref. 40. Such methodology accounts for both mechanical and electrostatic coupling between the QM (quantum-mechanical) and MM (molecular-mechanics) regions, but not for bonding interactions (*i.e.*, bonds between the QM and MM regions). In practice, we need to run two different codes, Quantum ESPRESSO for the QM region and a classical force-field code for the MM region, that communicate atomic positions, forces, electrostatic potentials.

LAMMPS [39] is the software chosen to deal with the classical (MM) degrees of freedom. This is a well-known and well-maintained package, released under an open-source license that allows redistribution together with Quantum ESPRESSO. The communications between the QM and MM regions use a "shared memory" approach: the MM code runs on a master node, communicates directly via the memory with the QM code, which is typically running on a massively parallel machine. Such approach has some advantages: the MM part is typically much faster than the QM one and can be run in serial execution, wasting no time on the HPC machine; there is a clear and neat separation between the two codes, and very small code changes in either codes are needed. It has however also a few drawbacks, namely: the serial computation of the MM part may become a bottleneck if the MM region contains many atoms; direct access to memory is often restricted for security reasons on HPC machines.

An alternative approach has been implemented in v.5.4. A single (parallel) executable runs both the MM and the QM codes. The two codes exchange data and communicate via MPI. This approach is less elegant than the previous one and requires a little bit more coding, but its implementation is quite straightforward thanks also to the changes in the logic of parallelization mentioned in Sec. III D. The coupling of the two codes has required some modifications also to the `qmmm` library inside LAMMPS and to the related fix `qmmm` (a "fix" in LAMMPS is any operation that is applied to the system during the MD run). In particular, electrostatic coupling has been introduced, following the approach described in Ref. 186. The great advantage of this approach is that its performance on HPC machines is as good as the separate performances of the QM and MM codes. Since LAMMPS is very well parallelized, this is a significant advantage if the MM region contains many atoms. Moreover, it can be run without restrictions on any parallel machine. This new QM-MM implementation is an integral part of the Quantum ESPRESSO distribution and will soon be included into LAMMPS as well (the "fix" is currently under testing) and it is straightforward to compile and execute it.

F. Miscellaneous feature enhancements and additions

1. Fully relativistic projector augmented-wave method

By applying the PAW formalism to the equations of relativistic spin density functional theory [187, 188], it is possible to obtain the fully relativistic PAW equations for four-component spinor pseudo-wavefunctions [16]. In this formalism the pseudo-wavefunctions can be written in terms of large $|\tilde{\Psi}^A_{i,\sigma}\rangle$ and small $|\tilde{\Psi}^B_{i,\sigma}\rangle$ components, both two-component spinors (the index σ runs over the two spin components). The latter is of order $\frac{v}{c}$ of the former, where v is of the order of the velocity of the electron and c is the speed of light. These equations can be simplified introducing errors of the order of the transferability error of the pseudopotential or of order $1/c^2$, depending on which is the largest. In the final equations only the large components of the pseudo-wavefunctions appear. The non relativistic kinetic energy $\mathbf{p}^2/2m$ (m is the electron mass) acts on the large component of the pseudo-wavefunctions $|\tilde{\Psi}^A_{i,\sigma}\rangle$ in the mesh defined by the FFT grid and the same kinetic energy is used to calculate the expectation values of the Hamiltonian between partial pseudo-waves $|\Phi^{I,PS,A}_{n,\sigma}\rangle$. The Dirac kinetic energy is used instead to calculate the expectation values of the Hamiltonian between all-electron partial waves $|\Phi^{I,AE}_{n,\eta}\rangle$ (η is a four-component index). In this manner, relativistic effects are hidden in the coefficients of the non-local pseudopotential. The equations are formally very similar to the equations of the scalar-relativistic case:

$$\sum_{\sigma_2}\left[\frac{\mathbf{p}^2}{2m}\delta^{\sigma_1,\sigma_2}+\sum_{\eta_1,\eta_2}\int d\mathbf{r}\tilde{V}^{\eta_1,\eta_2}_{\mathrm{LOC}}(\mathbf{r})\tilde{K}(\mathbf{r})^{\eta_1,\eta_2}_{\sigma_1,\sigma_2}-\varepsilon_i S^{\sigma_1,\sigma_2}\right.$$
$$\left.+\sum_{I,mn}(D^1_{I,mn}-\tilde{D}^1_{I,mn})|\beta^{I,A}_{m,\sigma_1}\rangle\langle\beta^{I,A}_{n,\sigma_2}|\right]|\tilde{\Psi}^A_{i,\sigma_2}\rangle=0, \tag{42}$$

where $D^1_{I,mn}$ and $\tilde{D}^1_{I,mn}$ are calculated inside the PAW spheres:

$$D^1_{I,mn}=\sum_{\eta_1,\eta_2}\langle\Phi^{I,AE}_{m,\eta_1}|T_D^{\eta_1,\eta_2}+V^{I,\eta_1,\eta_2}_{\mathrm{LOC}}|\Phi^{I,AE}_{n,\eta_2}\rangle, \tag{43}$$

$$\tilde{D}^1_{I,mn}=\sum_{\sigma_1,\sigma_2}\langle\Phi^{I,PS,A}_{m,\sigma_1}|\frac{\mathbf{p}^2}{2m}\delta^{\sigma_1,\sigma_2}+\tilde{V}^{I,\sigma_1,\sigma_2}_{\mathrm{LOC}}|\Phi^{I,PS,A}_{n,\sigma_2}\rangle$$
$$+\sum_{\eta_1,\eta_2}\int_{\Omega_I}d\mathbf{r}\hat{Q}^I_{mn,\eta_1,\eta_2}(\mathbf{r})\tilde{V}^{I,\eta_1,\eta_2}_{\mathrm{LOC}}(\mathbf{r}). \tag{44}$$

Here T_D is the Dirac kinetic energy:

$$T_D=c\boldsymbol{\alpha}\cdot\mathbf{p}+(\boldsymbol{\beta}-\mathbf{1}_{4\times4})mc^2, \tag{45}$$

written in terms of the 4×4 Hermitian matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and $V^{\eta_1,\eta_2}_{\mathrm{LOC}}$ is the sum of the local, Hartree, and XC potential (V_{eff}) together, in magnetic systems, with the contribution of the XC magnetic

field: $V_{\rm LOC}^{\eta_1,\eta_2}(\mathbf{r}) = V_{\rm eff}(\mathbf{r})\delta^{\eta_1,\eta_2} - \mu_B \mathbf{B}_{\rm xc}(\mathbf{r}) \cdot (\boldsymbol{\beta}\boldsymbol{\Sigma})^{\eta_1,\eta_2}$. We refer to Ref. 16 for a detailed definition of the partial waves $|\Phi_{n,\eta}^{I,AE}\rangle$, $|\Phi_{n,\sigma}^{I,PS,A}\rangle$ and projectors $|\beta_{m,\sigma}^{I,A}\rangle$, of the augmentation functions $\hat{Q}_{mn,\eta_1,\eta_2}^{I}(\mathbf{r})$ and $\tilde{K}(\mathbf{r})_{\sigma_1,\sigma_2}^{\eta_1,\eta_2}$, and of the overlap matrix S^{σ_1,σ_2} and for their rewriting in terms of projector functions that contain only spherical harmonics. Solving these equations it is possible to include spin-orbit coupling effects in electronic structure calculations. In QUANTUM ESPRESSO these equations are used when input variables `noncolin` and `lspinorb` are both `.TRUE.` and the PAW data sets are fully relativistic, as those available with the `pslibrary` project.

2. Electronic and structural properties in field-effect configuration

Since QUANTUM ESPRESSO v.6.0 it is possible to compute the electronic structure under a field-effect transistor (FET) setup in periodic boundary conditions [189]. In physical FETs, a voltage is applied to a gate electrode, accumulating charges at the interface between the gate dielectric and a semiconducting system (see Fig. 3). The gate electrode is simulated with a charged plate, henceforth referred to as the *gate*. Since the interaction of this charged plate with its periodic image generates a spurious nonphysical electric field, a dipolar correction, equivalent to two planes of opposite charge, is added [190], canceling out the field on the left side of the gate. In order to prevent electrons from spilling towards the gate for large electron doping [191], a potential barrier can be added to the electrostatic potential, mimicking the effect of the gate dielectric.

The setup for a system in FET configuration is shown in Fig. 3. The gate has a charge $n_{\rm dop}A$ and the system has opposite charge. Here $n_{\rm dop}$ is the number of doping electrons per unit area (*i.e.*, negative for hole doping), A is the area of the unit cell parallel to the surface. In practice the gate is represented by an external potential

$$V_{\rm gate}(\mathbf{r}) = -2\pi\, n_{\rm dop} \left(-|\overline{\rm z}| + \tfrac{\overline{\rm z}^2}{L} + \tfrac{L}{6} \right) \tag{46}$$

Here $\overline{\rm z} = z - z_{\rm gate}$ with $\overline{\rm z} \in [-L/2; L/2]$ measures the distance from the gate (see Fig. 3). The dipole of the charged system plus the gate is canceled by an electric dipole generated by two planes of opposite charge [190, 192, 193], placed at $z_{\rm dip} - d_{\rm dip}/2$ and $z_{\rm dip} + d_{\rm dip}/2$, in the vacuum region next to the gate ($V_{\rm dip}$ in Fig. 3). Additionally one may include a potential barrier to avoid charge spilling towards the gate, or as a substitute for the gate dielectric. $V_b(\mathbf{r})$ is a periodic function of z defined on the interval $z \in [0, L]$ as equal to a constant V_b for $z_1 < z < z_2$ and zero elsewhere. Figure 3 shows the resulting total potential (black line). The following additional variables are needed: $z_{\rm gate}$, z_1, z_2, and V_0. In the code these variables are named `zgate`, `block_1`, `block_2`,

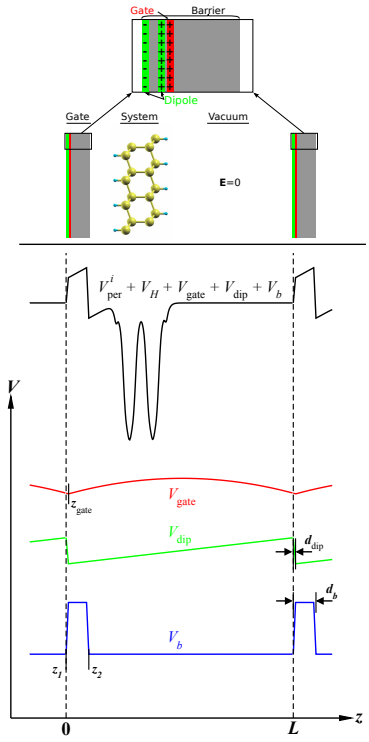

FIG. 3. Schematic picture of the planar averaged KS potential (without the exchange-correlation potential) for periodically repeated, charged slabs. The uppermost panel shows a sketch of a gated system. The different parts of the total KS potential are shown with different color: red – gate, $V_{\rm gate}$, green – dipole, $V_{\rm dip}$, blue – potential barrier, V_b. The position of the gate is indicated by $z_{\rm gate}$. The black line shows the sum of $V_{\rm gate}$, $V_{\rm dip}$, V_b, of the ionic potential $V^i_{\rm per}$, and of the Hartree potential V_H. The length of the unit cell along $\hat{\mathbf{z}}$ is given by L.

and `block_height`, respectively. The dipole corrections and the gate are activated by the options `dipfield=.true.` and `gate=.true.`. In order to enable the potential barrier and the relaxation of the system towards it, the new input parameters `block` and `relaxz`, respectively, have to be set to `.true.` More details about the implementation can be found in Ref. 189.

3. Cold restart of Car-Parrinello molecular dynamics

In the standard Lagrangian formulation of *ab initio* molecular dynamics [53], the coefficients of KS molecular orbitals over a given basis set (*i.e.*. their Fourier coefficients, in the case of plane waves) are treated as classical degrees of freedom obeying Newton's equations of motion that derive from a suitably defined extended Lagrangian. This Lagrangian is obtained from the Born-Oppenheimer total energy by augmenting it with a fictitious electronic kinetic-energy term and relaxing the constraint that the molecular orbitals stay at each instant of the trajectory in their

instantaneous KS ground state. The idea is that, by choosing a suitably small *fictitious electronic mass*, the thermalization time of the electronic degrees of freedom can be made much longer than the typical simulation times, so that if the system is prepared in its electronic KS ground state at the start of the simulation, the electronic dynamics would follow almost adiabatically the nuclear one all over the simulation, thus effectively mimicking a *bona fide* Born-Oppenheimer dynamics.

While in Car-Parrinello MD both the physical nuclear and fictitious electronic velocities are determined by the equations of motion on a par, the question still remains as to how choose them at the start of the simulation. Initial nuclear velocities are dictated by physical considerations (*e.g.*, thermal equilibrium) or may be taken from a previously interrupted MD run. Electronic velocities (*i.e.*, the time derivatives of the KS molecular orbitals), instead, are not available when the simulation is started from scratch. and are not independent of the physical nuclear ones, but are determined by the adiabatic time evolution of the system. Moreover, the projection over the occupied-state manifold of the electronic velocities, $\dot{\psi}_v^{\parallel} \doteq \hat{P}\dot{\psi}_v$ is ill-defined because the KS ground-state solution is defined modulo a unitary transformation within this manifold. This means that the starting electronic velocities may not be simply obtained as finite differences of KS orbitals at times $t = 0$ and $t = \Delta t$. Here and in the following $\hat{P}$ indicates the projector over the occupied-state manifold, and $\hat{Q} = 1 - \hat{P}$ its complement (*i.e.* the projector over the virtual-orbital manifold).

The component of the electronic velocities over the virtual-state manifold, $\dot{\psi}_v^{\perp} \doteq \hat{Q}\dot{\psi}_v$, is instead well defined and can be formally written using standard first-order perturbation theory:

$$\dot{\psi}_v^{\perp}(\mathbf{r}) = \sum_c \psi_c(\mathbf{r}) \frac{\langle \psi_c | \dot{V}_{KS} | \psi_v \rangle}{\epsilon_v - \epsilon_c}, \tag{47}$$

where v and c indicate occupied (*valence*) and virtual (*conduction*) states, respectively, ϵ_n the corresponding orbital energies, and $\dot{V}_{KS}$ is the time derivative of the KS potential, V_{KS}. $\dot{V}_{KS}$ is the linear response of V_{KS} to the perturbation in the external potential determined by an infinitesimal displacement of the nuclei along a MD trajectory: $\dot{V}_{\text{ext}}(\mathbf{r}) = \sum_{\mathbf{R}} \frac{\partial v_{\mathbf{R}}(\mathbf{r}-\mathbf{R})}{\partial \mathbf{R}} \cdot \dot{\mathbf{R}}$, where $v_{\mathbf{R}}(\mathbf{r}-\mathbf{R})$ is the bare ionic pseudopotential of the atom at position $\mathbf{R}$ and $\dot{\mathbf{R}}$ its velocity. Electronic velocities can conveniently be initialized to the values given by Eq. (47), which are those that minimize their norm and, hence, the initial *electronic temperature*, which is defined as the sum of the squared norms of the electronic velocities.

While this could in principle be done using density-functional perturbation theory [90, 93], it is more convenient to compute them numerically, following the procedure described below. At $t = 0$ the KS molecular orbitals are initialized from a ground-state computation, performed with whatever method is available or preferred (standard SCF calculation or global optimization, such as

e.g., with conjugate gradients [194]). The KS molecular orbitals that would result from a perfectly adiabatic propagation at $t = \Delta t$ are then determined from a second ground-state computation, performed after half a "velocity-Verlet" MD step, *i.e.*, at nuclear positions $\mathbf{R}(\Delta t) = \mathbf{R}(0)+\dot{\mathbf{R}}(0)\Delta t$. The initial velocities are then obtained from the relation:

$$\dot{\psi}_v^{\perp} = \hat{\dot{P}}\psi_v, \tag{48}$$

which is obtained by simply differentiating the definition of occupied-state projector, $\hat{P}\psi_v = \psi_v$. The right-hand side of Eq. (48) is finally easily computed by subtracting from each KS orbital at time $t = 0$, its component over the occupied-state manifold at $t = \Delta t$ and dividing by Δt.

4. Optimized tetrahedron method

The integration over **k**-points in the BZ is a crucial step in the calculation of the electronic structure of a periodic system, affecting not only the ground state but linear response as well. This is especially true for metallic systems where the integrand is discontinuous at the Fermi level. Even more problematic is the integration of Dirac delta functions, such as those appearing in the density of states (DOS), partial DOS and in the electron-phonon coupling constant.

Quantum ESPRESSO has always implemented a variety of "smearing" methods, in which the delta function is replaced by a function of finite width (*e.g.*, a Gaussian function, or more sophisticated choices). It has also always implemented the linear tetrahedron method [195] with the correction proposed by Blöchl [196], in which the BZ is divided into tetrahedra and the integration is performed analytically by linear interpolation of KS eigenvalues in each tetrahedron. Such method is however limited in its convenience and range of applicability: in fact the linear interpolation systematically overestimates convex functions, thus making the convergence against the number of **k**-points slow. The linear tetrahedron method was thus mostly restricted to the calculation of DOS and partial DOS.

Since Quantum ESPRESSO v.6.1, the optimized tetrahedron method [197] is implemented. Such method overcomes the drawback of the linear tetrahedron method using an interpolation that accounts for the curvature of the interpolated function. The optimized tetrahedron method has better convergence properties and an extended range of applicability: in addition to the calculation of the ground-state charge density, DOS and partial DOS, it can be used in linear-response calculation of phonons and of the electron-phonon coupling constant.

5. *Wyckoff positions*

In QUANTUM ESPRESSO the crystal geometry is traditionally specified by a Bravais lattice index (called `ibrav`), by the crystal parameters (`celldm`, or `a`, `b`, `c`, `cosab`, `cosac`, `cosbc`) describing the unit cell, and by the positions of *all* atoms in the unit cell, in crystal or Cartesian axis.

Since v.5.1.1, it is possible to specify the crystal geometry in crystallographic style[198], according to the notations of the International Tables of Crystallography (ITA)[199]. A complete description of the crystal structure is obtained by specifying the space-group number according to the ITA and the positions of *symmetry-inequivalent* atoms only in the unit cell. The latter can be provided either in the crystal axis of the conventional cell, or as *Wyckoff positions*: a set of special positions, listed in the ITA for each space group, that can be fully specified by a number of parameters, none to three depending upon the site symmetry. Table 1 reports a few examples of accepted syntax. The code generates the symmetry operations for the specified space group and

TABLE I. Examples of valid syntax for Wyckoff positions. *C* is the element name, followed by the Wyckoff label of the site (number of equivalent atoms followed by a letter identifying the site), followed by the site-dependent parameters needed to fully specify the atomic positions.

ATOMIC_POSITIONS sg

C	1a	
C	8g	x
C	24m	x y
C	48n	x y z
C		x y z

applies them to inequivalent atoms, thus finding all atoms in the unit cell.

For some crystal systems there are alternate descriptions in the ITA, so additional input parameters may be needed to select the desired one. For the monoclinic system the "c-unique" orientation is the default and `bunique=.TRUE.` must be specified in input if the "b-unique" orientation is desired. For some space groups there are two possible choices of the origin. The origin appearing first in the ITA is chosen by default, unless `origin_choice=2` is specified in input. Finally, for trigonal space groups the atomic coordinates can be referred to the rhombohedral or to the hexagonal Bravais lattices. The default is the rhombohedral lattice, so `rhombohedral=.FALSE.` must be specified in input to use the hexagonal lattice.

A final comment for centered Bravais lattices: in the crystallographic literature, the *conventional* unit cell is usually assumed. QUANTUM ESPRESSO however assumes the *primitive* unit cell,

having a smaller volume and a smaller number of atoms, and discards atoms outside the primitive cell. Auxiliary code `supercell.x`, available in `thermo_pw` (see Sec.II D 1), prints all atoms in the conventional cell when necessary.

III. PARALLELIZATION, MODULARIZATION, INTEROPERABILITY AND STABILITY

A. New parallelization levels

The basic modules of QUANTUM ESPRESSO are characterized by a hierarchy of parallelization levels, described in Ref.6. Processors are divided into groups, labeled by a MPI communicator. Each group of processors distributes a specific subset of computations. The growing diffusion of HPC machines based on nodes with many cores (32 and more) makes however pure MPI parallelization not always ideal: running one MPI process per core has a high overhead, limiting performances. It is often convenient to use mixed MPI-OpenMP parallelization, in which a small number of MPI processes per node use OpenMP threads, either explicitly (*i.e.*, with compiler directives) or implicitly (*i.e.*, via calls to OpenMP-aware library). Explicit OpenMP parallelization, originally confined to computationally intensive FFT's, has been extended to many more parts of the code.

One of the challenges presented by massively parallel machine is to get rid of both memory and CPU time bottlenecks, caused respectively by arrays that are not distributed across processors and by non-parallelized sections of code. It is especially important to distribute all arrays and parallelize all computations whose size/complexity increases with the dimensions of the unit cell or of the basis set. Non-parallelized computations hamper "weak" scalability, that is, parallel performance while increasing both the system size and the amount of computational resources, while non-distributed arrays may become an unavoidable RAM bottleneck with increasing problem size. "Strong" scalability (that is, at fixed problem size and increasing number of CPUs) is even more elusive than weak scalability in electronic-structure calculations, requiring, in addition to systematic distribution of computations, to keep to the minimum the ratio between time spent in communications and in computation, and to have a nearly perfect load balancing. In order to achieve strong scalability, the key is to add more parallelization levels and to use algorithms that permit to overlap communications and computations.

For what concerns memory, notable offenders are arrays of scalar products between KS states ψ_i: $O_{ij} = \langle\psi_i|\widehat{O}|\psi_j\rangle$, where $\widehat{O}$ can be either the Hamiltonian or an overlap matrix; and scalar products

between KS states and pseudopotential projectors β, $p_{ij} = \langle\psi_i|\beta_j\rangle$. The size of such arrays grows as the square of the size of the cell. Almost all of them are now distributed across processors of the "linear-algebra group", that is, the group of processors taking care of linear-algebra operations on matrices. The most expensive of such operations are subspace diagonalization (used in `PWscf` in the iterative diagonalization) and iterative orthonormalization (used by `CP`). In both cases, a parallel dense-matrix diagonalization on distributed matrix is needed. In addition to ScaLAPACK, QUANTUM ESPRESSO can now take advantage of newer ELPA libraries (Ref. 200), leading to significant performance improvements.

The array containing the plane-wave representation, $c_{\mathbf{k},n}(\mathbf{G})$, of KS orbitals is typically the largest array, or one of the largest. While plane waves are already distributed across processors of the "plane-wave group" as defined in Ref. 6, it is now possible to distribute KS orbitals as well. Such a parallelization level is located between the **k**-point and the plane-wave parallelization levels. The corresponding MPI communicator defines a subgroup of the "**k**-point group" of processors and is called "band group communicator". In the `CP` package, band parallelization is implemented for almost all available calculations. Its usefulness is better appreciated in simulations of large cells — several hundreds of atoms and more — where the number of processors required by memory distribution would be too large to get good scalability from plane-wave parallelization only.

In `PWscf`, band parallelization is implemented for calculations using hybrid functionals. The standard algorithm to compute Hartree-Fock exchange in a plane-wave basis set (see Sec. II A 1) contains a double loop on bands that is by far the heaviest part of computation. A first form of parallelization, described in Ref. 34, was implemented in v.5.0. In the latest version, this has been superseded by parallelization of pairs of bands, Ref. 35. Such algorithm is compatible with the "task-group" parallelization level (that is: over KS states in the calculation of $V\psi_i$ products) described in Ref. 6.

In addition to the above-mentioned groups, that are globally defined and in principle usable in all routines, there are a few additional parallelization levels that are local to specific routines. Their goal is to reduce the amount of non-parallel computations that may become significant for many-atom systems. An example is the calculation of DFT+U (Sec. II A 3) terms in energy and forces, Eqs. (12) and (14) respectively. In all these expressions, the calculation of the scalar products between valence and atomic wave functions is in principle the most expensive step: for N_b bands and N_{pw} plane waves, $\mathcal{O}(N_{pw}N_b)$ floating-point operations are required (typically, $N_{pw} \gg N_b$). The calculation of these terms is however easily and effectively parallelized, using standard matrix-matrix multiplication routines and summing over MPI processes with a `mpi_reduce` operation on

the plane-wave group. The sum over **k**-points can be parallelized on the **k**-point group. The remaining sums over band indices ν and Hubbard orbitals I, m may however require a significant amount of non-parallelized computation if the number of atoms with a Hubbard U term is not small. The sum over band indices is thus parallelized by simply distributing bands over the plane-wave group. This is a convenient choice because all processors of the plane-wave group are available once the scalar products are calculated. The addition of band parallelization speeds up the computation of such terms by a significant factor. This is especially important for Car-Parrinello dynamics, requiring the calculation of forces at each time step, when a sizable number of Hubbard manifolds is present.

B. Aspects of interoperability

One of the original goals of QUANTUM ESPRESSO was to assemble different pieces of rather similar software into an integrated software suite. The choice was made to focus on the following four aspects: input data formats, output data files, installation mechanism, and a common base of code. While work on the first three aspects is basically completed, it is still ongoing on the fourth. It was however realized that a different form of integration — interoperability, *i.e.*, the possibility to run QUANTUM ESPRESSO along with other software — was more useful to the community of users than tight integration. There are several reasons for this, all rooted in new or recent trends in computational materials science. We mention in particular the usefulness of interoperability for

1. excited-states calculations using many-body perturbation theory, at various levels of sophistication: *GW*, TDDFT, BSE (*e.g.*, `yambo` [201], `SaX` [202], or `BerkeleyGW` [203]);

2. calculations using quantum Monte Carlo methods;

3. configuration-space sampling, using such algorithms as nudged elastic band (NEB), genetic/evolutionary algorithms, meta-dynamics;

4. inclusion of quantum effects on nuclei via path-integral Monte Carlo;

5. multi-scale simulations, requiring different theoretical approaches, each valid in a given range of time and length scale, to be used together;

6. high-throughput, or "exhaustive", calculations (*e.g.*, `AiiDA` [204, 205] and AFLOWπ [206]) requiring automated submission, analysis, retrieval of a large number of jobs;

7. "steering", *i.e.*, controlling the computation in real time using either a graphical user interface (GUI) or an interface in a high-level interpreted language (*e.g.*, python).

It is in principle possible, and done in some cases, to implement all of the above into QUANTUM ESPRESSO, but this is not always the best practice. A better option is to use QUANTUM ESPRESSO in conjunction with external software performing other tasks.

Cases 1 and 2 mentioned above typically use as starting step the self-consistent solution of KS equations, so that what is needed is the possibility for external software to read data files produced by the main QUANTUM ESPRESSO codes, notably the self-consistent code `PWscf` and the molecular dynamics code `CP`.

Cases 3 and 4 typically require many self-consistent calculations at different atomic configurations, so that what is needed is the possibility to use the main QUANTUM ESPRESSO codes as "computational engine", *i.e.*, to call `PWscf` or `CP` from an external software, using atomic configurations supplied by the calling code.

The paradigmatic case 5 is QM-MM (Sec.II E 2), requiring an exchange of data, notably: atomic positions, forces, and some information on the electrostatic potential, between QUANTUM ESPRESSO and the MM code – typically a classical MD code.

Case 6 requires easy access to output data from one simulation, and easy on-the-fly generation of input data files as well. This is also needed for case 7, which however may also require a finer-grained control over computations performed by QUANTUM ESPRESSO routines: in the most sophisticated scenario, the GUI or python interface should be able to perform specific operations "on the fly", not just running an entire self-consistent calculation. This scenario relies upon the existence of a set of application programming interfaces (API's) for calls to basic computational tasks.

C. Input/Output and data file format

On modern machines, characterized by fast CPU's and large RAM's, disk input/output (I/O) may become a bottleneck and should be kept to a strict minimum. Since v.5.3 both `PWscf` and `CP` do not perform by default any I/O at run time, except for the ordinary text output (printout), for checkpointing if required or needed, and for saving data at the end of the run. The same is being gradually extended to all codes. In the following, we discuss the case of the final data writing.

The original organization of output data files (or more exactly, of the output data directory) was based on a formatted "head" file, with a XML-like syntax, containing general information on

the run, and on binary data files containing the KS orbitals and the charge density. We consider the basic idea of such approach still valid, but some improvements were needed. On one hand, the original head file format had a number of small issues—inconsistencies, missing pieces of relevant information—and used a non-standard syntax, lacking a XML "schema" for validation. On the other hand, data files suffered from the lack of portability of Fortran binary files and had to be transformed into text files, sometimes very large ones, in order to become usable on a different machine.

1. XML files with schema

Since v.6.0, the "head" file is a true XML file using a consistent syntax, described by a XML schema, that can be easily parsed with standard XML tools. It also contains complete information on the run, including all data needed to reproduce the results, and on the correct execution and exit status. This aspect is very useful for high-throughput applications, for databasing of results and for verification and validation purposes.

The XML file contains an input section and can thus be used as input file, alternative to the still existing text-based input. It supersedes the previous XML-based input, introduced several years ago, that had a non-standard syntax, different from and incompatible with the one of the original head file. Implementing a different input is made easy by the clear separation existing between the reading and initialization phases: input data is read, stored in a separate module, copied to internal variables.

The current XML file can be easily parsed and generated using standard XML tools and is especially valuable in conjunction with GUI's. The schema can be found at the URL:
`http://www.quantum-espresso.org/ns/qes/qes-1.0.xsd`.

2. Large-record data file format

Although not as I/O-bound as other kinds of calculations, electronic-structure simulations may produce a sizable amount of data, either intermediate or needed for further processing. The largest array typically contains the plane-wave representation of KS orbitals; other sizable arrays contain the charge and spin density, either in reciprocal or in real space. In parallel execution using MPI, large arrays are distributed across processors, so one has two possibilities: let each MPI process write its own slice of the data ("distributed" I/O), or collect the entire array on a single processor

before writing it ("collected" I/O). In distributed I/O, coding is straightforward and efficient, minimizing file size and achieving some sort of I/O parallelization. A global file system, accessible to all MPI processes, is needed. The data is spread into many files that are directly usable only by a code using exactly the same distribution of arrays, that is, exactly the same kind of parallelization. In collected I/O, the coding is less straightforward. In order to ensure portability, some reference ordering, independent upon the number of processors and the details of the parallelization, must be provided. For large simulations, memory usage and communication pattern must be carefully optimized when a distributed array is collected into a large array on a single processor.

In the original I/O format, KS orbitals were saved in reciprocal space, in either distributed or collected format. For the latter, a reproducible ordering of plane waves (including the ordering within shells of plane waves with the same module), independent upon parallelization details and machine-independent, ensures data portability. Charge and spin density were instead saved in real space and in collected format. In the new I/O scheme, available since v.6.0, the output directory is simplified, containing only the XML data file, one file per **k**-point with KS orbitals, one file for the charge and spin density. Both files are in collected format and both quantities are stored in reciprocal space. In addition to Fortran binary, it is possible to write data files in HDF5 format[207]. HDF5 offers the possibility to write structured record and portability across architectures, without significant loss in performances; it has an excellent support and is the standard for I/O in other fields of scientific computing. Distributed I/O is kept only for checkpointing or as a last-resort alternative.

In spite of its advantages, such a solution has still a bottleneck in large-scale computations on massively parallel machines: a single processor must read and write large files. Only in the case of parallelization over **k**-points is I/O parallelized in a straightforward way. More general solutions to implement parallel I/O using parallel extensions of HDF5 are currently under examination in view of enabling QUANTUM ESPRESSO towards "exascale" computing (that is: towards $\mathcal{O}(10^{18})$ floating-point operations per second).

D. Organization of the distribution

Codes contained in QUANTUM ESPRESSO have evolved from a small set of original codes, born with rather restricted goals, into a much larger distribution via continuous additions and extensions. Such a process - presumably common to most if not all scientific software projects - can easily lead to uncoordinated growth and to bad decisions that negatively affect maintainability.

1. Package re-organization and modularization

In order to make the distribution easier to maintain, extend and debug, the distribution has been split into

a. base distribution, containing common libraries, tools and utilities, core packages `PWscf`, `CP`, `PostProc`, plus some commonly used additional packages, currently: `atomic`, `PWgui`, `PWneb`, `PHonon`, `XSpectra`, `turboTDDFT`, `turboEELS`, `GWL`, `EPW`;

b. external packages such as `SaX` [202], `yambo` [201], `Wannier90` [171], `WanT` [208, 209], that are automatically downloaded and installed on demand.

The directory structure now explicitly reflects the structure of QUANTUM ESPRESSO as a "federation" of packages rather than a monolithic one: a common base distribution plus additional packages, each of which fully contained into a subdirectory.

In the reorganization process, the implementation of the NEB algorithm was completely rewritten, following the paradigm sketched in Sec. III B. `PWneb` is now a separate package that implements the NEB algorithm, using `PWscf` as the computational engine. The separation between the NEB algorithm and the self-consistency algorithm is quite complete: `PWneb` could be adapted to work in conjunction with a different computational engine with a minor effort.

The implementation of meta-dynamics has also been re-considered. Given the existence of a very sophisticated and well-maintained package[210] `Plumed` for all kinds of meta-dynamics calculations, the `PWscf` and `CP` packages have been adapted to work in conjunction with `Plumed` v.1.x, removing the old internal meta-dynamics code. In order to activate meta-dynamics, a patching process is needed, in which a few specific "hook" routines are modified so that they call routines from `Plumed`.

2. Modular parallelism

The logic of parallelism has also evolved towards a more modular approach. It is now possible to have all QUANTUM ESPRESSO routines working inside a MPI communicator, passed as argument to an initialization routine. This allows in particular the calling code to have its own parallelization level, invisible to QUANTUM ESPRESSO routines; the latter can thus perform independent calculations, to be subsequently processed by the calling code. For instance: the "image" parallelization level, used by NEB calculations, is now entirely managed by `PWneb` and no longer in the called `PWscf` routines. Such a feature is very useful for coupling external codes to

QUANTUM ESPRESSO routines. To this end, a general-purpose library for calling `PWscf` or `CP` from external codes (either Fortran or C/C++ using the Fortran 2003 ISO C binding standard) is provided in the directory `COUPLE/`.

3. Reorganization of linear-response codes

All linear-response codes described in Secs. II B and II A 4 share as basic computational step the self-consistent solution of linear systems $Ax = b$ for different perturbations b, where the operator A is derived from the KS Hamiltonian H and the linear-response potential. Both the perturbations and the methods of solution differ by subtle details, leading to a plethora of routines, customized to solve slightly different versions of the same problem. Ideally, one should be able to solve any linear-response problem by using a suitable library of existing code. To this end, a major restructuring of linear-response codes has been started. Several routines have been unified, generalized and extended. They have been collected into the same subdirectory, `LR_Modules`, that will be the container of "generic" linear-response routines. Linear-response-related packages now contain only code that is specific to a given perturbation or property calculation.

E. QUANTUM ESPRESSO and scripting languages

A desirable feature of electronic-structure codes is the ability to be called from a high-level interpreted scripting language. Among the various alternatives, python has emerged in the last years due to its simple and powerful syntax and to the availability of numerical extensions (NumPy). Since v.6.0, an interface between `PWscf` and the path integral MD driver i-PI [41] is available and distributed together with QUANTUM ESPRESSO. Various implementations of an interface between QUANTUM ESPRESSO codes and the atomic simulation environment (ASE) [211] are also available. In the following we briefly highlight the integration of QUANTUM ESPRESSO with `AiiDA`, the `pwtk` toolkit for `PWscf`, and the `QE-emacs-modes` package for user-friendly editing of QUANTUM ESPRESSO with the `Emacs` editor [212].

1. `AiiDA`: a python materials' informatics infrastructure

`AiiDA` [204] is a comprehensive python infrastructure aimed at accelerating, simplifying, and organizing major efforts in computational science, and in particular computational materials science, with a close integration with the QUANTUM ESPRESSO distribution. `AiiDA` is structured

around the four pillars of the ADES model (Automation, Data, Environment, and Sharing, Ref. 204)), and provides a practical and efficient implementation of all four. In particular, it aims at relieving the work of a computational scientist from the tedious and error-prone tasks of running, overseeing, and storing hundreds or more of calculations daily (Automation pillar), while ensuring that strict protocols are in place to store these calculations in an appropriately structured database that preserves the provenance of all computational steps (Data pillar). This way, the effort of a computational scientist can become focused on developing, curating, or exploiting complex workflows (Environment pillar) that calculate in a robust manner e.g. the desired materials properties of a given input structure, recording expertise in reproducible sequences that can be progressively perfected, while being able to share freely both the workflows and the data generated with public or private common repositories (Sharing). `AiiDA` is built using an agnostic structure that allows to interface it with any given code — through plugins and a plugin repository — or with different queuing systems, transports to remote HPC resources, and property calculators. In addition, it allows to use arbitrary object-relational mappers (ORMs) as backends (currently, Django and SQLAlchemy are supported). These ORMs map the `AiiDA` objects ("Codes", "Calculations" and "Data") onto python classes, and lead to the representation of calculations through Directed Acyclic Graphs (DAGs) connecting all objects with directional arrows; this ensures both provenance and reproducibility of a calculation. As an example, in Fig. 4 we present a simple DAG representing a `PWscf` calculation on $BaTiO_3$.

2. `Pwtk`: a toolkit for `PWscf`

The `pwtk`, standing for **PWscf ToolKit**, is a Tcl scripting interface for `PWscf` set of programs contained in the QUANTUM ESPRESSO distribution. It aims at providing a flexible and productive framework. The basic philosophy of `pwtk` is to lower the learning curve by using syntax that closely matches the input syntax of QUANTUM ESPRESSO. `Pwtk` features include: (i) assignment of default values of input variables on a project basis, (ii) reassignment of input variables on the fly, (iii) stacking of input data, (iv) math-parser, (v) extensible and hierarchical configuration (global, project-based, local), (vi) data retrieval functions (*i.e.*, either loading the data from pre-existing input files or retrieving the data from output files), and (vii) a few predefined higher-level tasks, that consist of several seamlessly integrated calculations. `Pwtk` allows to easily automate large number of calculations and to glue together different computational tasks, where output data of preceding calculations serve as input for subsequent calculations. `Pwtk` and related documentation

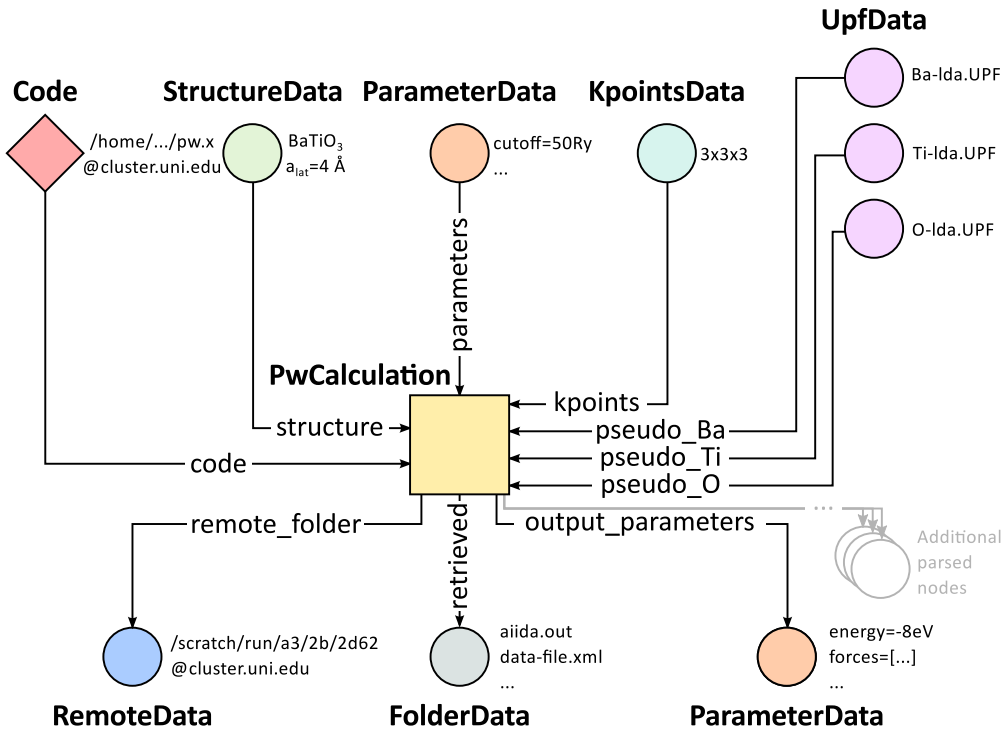

FIG. 4. A simple `AiiDA` directed acyclic graph for a QUANTUM ESPRESSO calculation using `PWscf` (square), with all the input nodes (data, circles; code executable, diamond) and all the output nodes that the daemon creates and connects automatically.

can be downloaded from `http://pwtk.quantum-espresso.org`.

3. *QE-emacs-modes*

The `QE-emacs-modes` package is an open-source collection of `Emacs` major-modes for making the editing of QUANTUM ESPRESSO input files easier and more comfortable with `Emacs`. The package provides syntax highlighting (see Fig. 5a), auto-indentation, auto-completion, and a few utility commands, such as `M-x` *prog*`-insert-template` that inserts a respective input file template for the *prog* program (*e.g.*, `pw`, `neb`, `pp`, `projwfc`, `dos`, `bands`). The `QE-emacs-modes` are aware of all namelists, variables, cards, and options that are explicitly documented in the `INPUT_`*PROG*`.html` files, which describe the respective input file syntax (see Fig. 5b), where *PROG* stands for the uppercase name of a given program of QUANTUM ESPRESSO. The reason for this is that both `INPUT_`*PROG*`.html` files and `QE-emacs-modes` are automatically generated by the internal `helpdoc` utility of QUANTUM ESPRESSO.

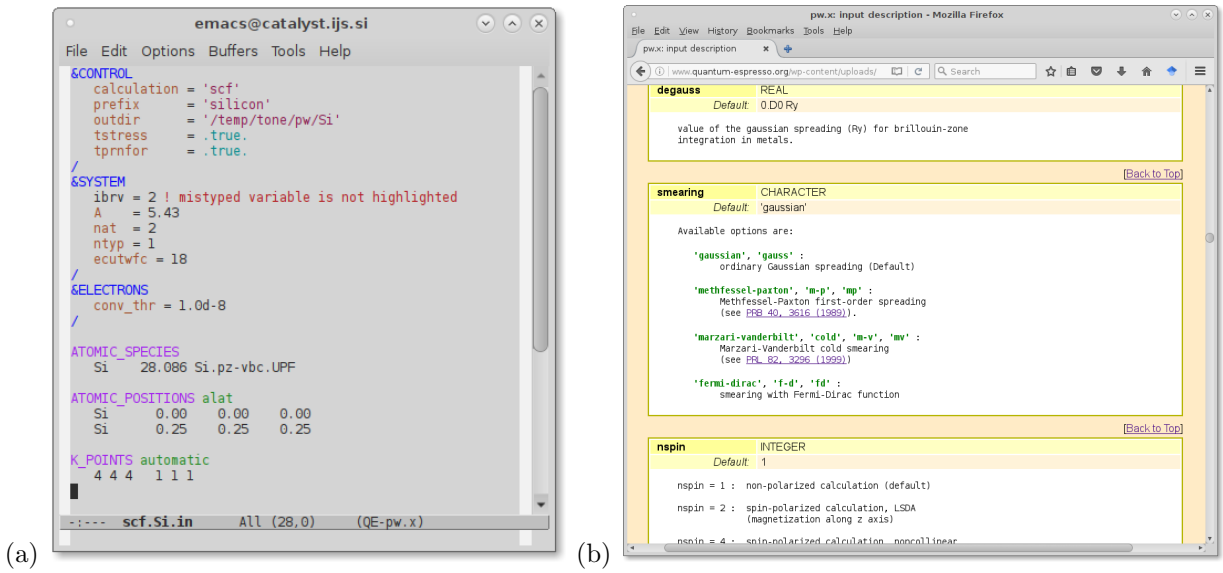

FIG. 5. (a) `pw.x` input file opened in `Emacs` with `pw-mode` highlighting the following elements: namelists and their variables (blue and brown), cards and their options (purple and green), comments (red), string and logical variable values (burgundy and cyan, respectively). A mistyped variable (*i.e.*, `ibrv` instead of `ibrav`) is not highlighted. (b) An excerpt from the `INPUT_PW.html` file, which describes the `pw.x` input file syntax. Both the `QE-emacs-modes` and the `INPUT_PW.html` are automatically generated from the QUANTUM ESPRESSO's internal definition of the input file syntax.

F. Continuous Integration and testing

The modularization of QUANTUM ESPRESSO reduces the extent of code duplication, thus improving code maintainability, but it also creates interdependencies between the modules so that changes to one part of the code may impact other parts. In order to monitor and mitigate these side effects we developed a test-suite for non-regression testing. Its purpose is to increase code stability by identifying and correcting those changes that break established functionalities. The test-suite relies on a modified version of python script `testcode` [213].

The layout of the test-suite is illustrated in Fig. 6. The suite is invoked via a Makefile that accepts several options to run sequential or parallel tests or to test one particular feature of the code. The test-suite runs the various executables of QUANTUM ESPRESSO, extracts the numerical data of interest, compares them to reference data, and decides whether the test is successful using specified thresholds. At the moment, the test-suite contains 181 tests for `PW`, 14 for `PH`, 17 for `CP`, 43 for `EPW`, and 6 for `TDDFpT` covering 43%, 30%, 29%, 63% and 25% of the blocks, respectively. Moreover, 60%, 44%, 47%, 76% and 32% of the subroutines in each of these codes are tested, respectively.

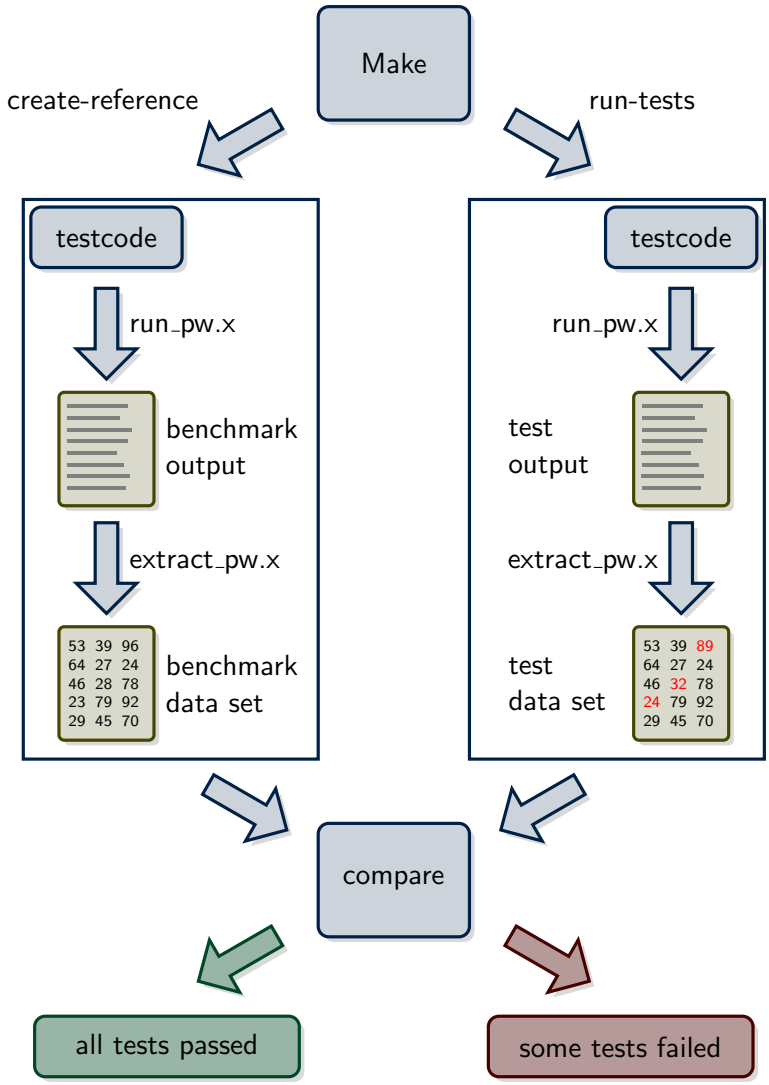

FIG. 6. (Color online) Layout of the QUANTUM ESPRESSO test-suite. The program `testcode` runs QUANTUM ESPRESSO executables, extracts numerical values from the output files, and compares the results with reference data. If the difference between these data exceeds a specified threshold, `testcode` issues an error indicating that a recent commit might have introduced a bug in parts of the code.

The test-suite also contains the logic to automatically create reference data by running the relevant executables and storing the output in a benchmark file. These benchmarks are updated only when new tests are added or bugfixes modify the previous behavior.

The test-suite enables automatic testing of the code using several `Buildbot` test farms. The test farms monitor the code repository continuously, and trigger daily builds in the night after every new commit. Several compilers (Intel, GFortran, PGI) are tested both in serial and in parallel (openmpi, mpich, Intel mpi and mvapich2) execution with different mathematical libraries

(LAPACK, BLAS, ScaLAPACK, FFTW3, MKL, OpenBlas). More information can be found at `test-farm.quantum-espresso.org`.

The official mirror of the development version of QUANTUM ESPRESSO (`https://github.com/QEF/q-e`) employs a subset of the test-suite to run `Travis CI`. This tool rapidly identifies erroneous commits and can be used to assist code review during a pull request.

IV. OUTLOOK AND CONCLUSIONS

This paper describes the core methodological developments and extensions of QUANTUM ESPRESSO that have become available, or are about to be released, after Ref. 6 appeared. The main goal of QUANTUM ESPRESSO to provide an efficient and extensible framework to perform simulations with well-established approaches and to develop new methods remains firm, and it has nurtured an ever growing community of developers and contributors.

Achieving such goal, however, becomes increasingly challenging. On one hand, computational methods become ever more complex and sophisticated, making it harder not only to implement them on a computer but also to verify the correctness of the implementation (for a much needed initial effort on verification of electronic-structure codes based on DFT, see Ref. 5). On the other hand, exploiting the current technological innovations in computer hardware can requires massive changes to software and even algorithms. This is especially true for the case of "accelerated" architectures (GPUs and the like), whose exceptional performance can translate to actual calculations only after heavy restructuring and optimization. The complexity of existing codes makes a rewrite for new architectures a challenging choice, and a risky one given the fast evolution of computer architectures.

We think that the main directions followed until now in the development of QUANTUM ESPRESSO are still valid, not only for new methodologies, but also for adapting to new computer architectures and future "exascale" machines. Namely, we will continue pushing towards code reusability, by removing duplicated code and/or replacing it with routines performing well-defined tasks, by identifying the time-intensive sections of the code for machine-dependent optimization, by having documented APIs with a predictable behavior and with limited dependency upon global variables, and we will continue to optimize performance and reliability. Finally, we will push towards extended interoperability with other software, also in view of its usefulness for data exchange and for cross-verification, or to satisfy the needs of high-throughput calculations.

Still, the investment in the development and maintenance of state-of-the-art scientific software

has historically lagged behind compared to the investment in the applications that use such software, and one wonders is this the correct or even forward-looking approach given the strategic importance of such tools, their impact, their powerful contribution to open science, and their full and complete availability to the entire community. In all of this, the future of materials simulations appear ever more bright[214], and the usefulness and relevance of such tools to accelerating invention and discovery in science and technology is reflected in its massive uptake by the community at large.

Acknowledgments. This work has been partially funded by the European Union through the MaX Centre of Excellence (Grant No. 676598) and by the Quantum ESPRESSO Foundation. SdG acknowledges support from the EU Centre of Excellence `E-CAM` (Grant No. 676531). OA, MC, NC, NM, NLN, and IT acknowledge support from the SNSF National Centre of Competence in Research MARVEL, and from the PASC Platform for Advanced Scientific Computing. TT acknowledges support from NSF Grant No. DMR-1145968. SP, MS, and FG are supported by the Leverhulme Trust (Grant RL-2012-001). MBN acknowledges support by DOD-ONR (N00014-13-1-0635, N00014-11-1-0136, N00014-15-1-2863) and the Texas Advanced Computing Center at the University of Texas, Austin. RD acknowledges partial support from Cornell University through start-up funding and the Cornell Center for Materials Research (CCMR) with funding from the NSF MRSEC program (DMR-1120296). This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. This research also used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

---

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[3] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).

[4] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).

[5] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. Dal Corso, S. de Gironcoli, T. Deutsch, J. K. Dewhurst, I. Di Marco, C. Draxl, M. Dułak, O. Eriksson, J. A. Flores-Livas, K. F. Garrity, L. Genovese, P. Giannozzi, M. Giantomassi, S. Goedecker, X. Gonze, O. Grånäs, E. K. U. Gross, A. Gulans, F. Gygi, D. R. Hamann, P. J. Hasnip, N. A. W. Holzwarth, D. Iuşan, D. B. Jochym, F. Jollet, D. Jones, G. Kresse, K. Koepernik, E. Küçükbenli, Y. O. Kvashnin, I. L. M. Locht, S. Lubeck, M. Marsman, N. Marzari, U. Nitzsche,

L. Nordström, T. Ozaki, L. Paulatto, C. J. Pickard, W. Poelmans, M. I. J. Probert, K. Refson, M. Richter, G.-M. Rignanese, S. Saha, M. Scheffler, M. Schlipf, K. Schwarz, S. Sharma, F. Tavazza, P. Thunström, A. Tkatchenko, M. Torrent, D. Vanderbilt, M. J. van Setten, V. Van Speybroeck, J. M. Wills, J. R. Yates, G.-X. Zhang, and S. Cottenier, Science **351**, 10.1126/science.aad3000 (2016).

[6] 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, and R. M. Wentzcovitch, J. Phys.: Condens. Matter **21**, 395502 (2009).

[7] L. Lin, J. Chem. Theory Comput. **12**, 2242 (2016).

[8] J. Jia, A. Vazquez-Mayagoitia, and R. A. DiStasio Jr., in preparation.

[9] K. Berland, V. R. Cooper, K. Lee, E. Schröder, T. Thonhauser, P. Hyldgaard, and B. I. Lundqvist, Reports Prog. Phys. **78**, 066501 (2015).

[10] S. Grimme, J. Comput. Chem. **27**, 1787 (2006).

[11] A. Tkatchenko and M. Scheffler, Phys. Rev. Lett. **102**, 073005 (2009).

[12] A. D. Becke and E. R. Johnson, J. Chem. Phys. **127**, 154108 (2007).

[13] E. R. Johnson, in *Non-covalent Interactions in Quantum Chemistry and Physics*, edited by A. Otero-de-la-Roza and G. DiLabio (Elsevier, in press, 2017) pp. 215–248.

[14] G. Sclauzero and A. Dal Corso, Phys. Rev. B **87**, 085108 (2013).

[15] B. Himmetoglu, R. M. Wentzcovitch, and M. Cococcioni, Phys. Rev. B **84**, 115108 (2011).

[16] A. Dal Corso, Phys. Rev. B **82**, 075116 (2010).

[17] A. Dal Corso, Phys. Rev. B **86**, 085135 (2012).

[18] O. Andreussi, I. Dabo, and N. Marzari, J. Chem. Phys. **136**, 064102 (2012).

[19] O. Andreussi and N. Marzari, Phys. Rev. B **90**, 245101 (2014).

[20] I. Timrov, O. Andreussi, A. Biancardi, N. Marzari, and S. Baroni, J. Chem. Phys. **142**, 034111 (2015).

[21] B. Walker, A. M. Saitta, R. Gebauer, and S. Baroni, Phys. Rev. Lett. **96**, 113001 (2006).

[22] D. Rocca, R. Gebauer, Y. Saad, and S. Baroni, J. Chem. Phys. **128**, 154105 (2008).

[23] O. B. Malcioğlu, R. Gebauer, D. Rocca, and S. Baroni, Comput. Phys. Commun. **182**, 1744 (2011).

[24] X. Ge, S. Binnie, D. Rocca, R. Gebauer, and S. Baroni, Comput. Phys. Commun. **185**, 2080 (2014).

[25] I. Timrov, N. Vast, R. Gebauer, and S. Baroni, Phys. Rev. B **88**, 064301 (2013), *ibid.* **91**, 139901(E) (2015).

[26] I. Timrov, N. Vast, R. Gebauer, and S. Baroni, Comput. Phys. Commun. **196**, 460 (2015).

[27] S. Poncé, E. Margine, C. Verdi, and F. Giustino, Comput. Phys. Commun. **209**, 116 (2016).

[28] P. Umari, G. Stenuit, and S. Baroni, Phys. Rev. B **79**, 201104 (2009).

[29] P. Umari, G. Stenuit, and S. Baroni, Phys. Rev. B **81**, 115104 (2010).

[30] M. Schlipf, H. Lambert, N. Zibouche, and F. Giustino, "SternheimerGW," `https://github.com/QEF/SternheimerGW` (2017).

[31] A. Dal Corso, `https://github.com/dalcorso/thermo_pw` ().

[32] L. Paulatto, F. Mauri, and M. Lazzeri, Phys. Rev. B **87**, 214303 (2013).

[33] G. Fugallo, M. Lazzeri, L. Paulatto, and F. Mauri, Phys. Rev. B **88**, 045430 (2013).

[34] N. Varini, D. Ceresoli, L. Martin-Samos, I. Girotto, and C. Cavazzoni, Comput. Phys. Commun. **184**, 1827 (2013).

[35] T. Barnes, T. Kurth, P. Carrier, N. Wichmann, D. Prendergast, P. R. C. Kent, and J. Deslippe, Comput. Phys. Commun. **241**, 52 (2017).

[36] A. Dal Corso, `http://pslibrary.quantum-espresso.org` ().

[37] A. Dal Corso, Comp. Material Science **95**, 337 (2015).

[38] I. Castelli, G. Prandini, A. Marrazzo, N. Mounet, and N. Marzari, `http://materialscloud.org/sssp/`.

[39] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[40] C. Ma, L. Martin-Samos, S. Fabris, A. Laio, and S. Piccinin, Comput. Phys. Commun. **195**, 191 (2015).

[41] M. Ceriotti, J. More, and D. E. Manolopoulos, Comput. Phys. Commun. **185**, 1019 (2014).

[42] X. Wu, A. Selloni, and R. Car, Phys. Rev. B **79**, 085102 (2009).

[43] R. A. DiStasio Jr., B. Santra, Z. Li, X. Wu, and R. Car, J. Chem. Phys. **141**, 084502 (2014).

[44] H.-Y. Ko, J. Jia, B. Santra, X. Wu, R. Car, and R. A. DiStasio Jr., J. Chem. Theory Comput. submitted.

[45] I. Carnimeo, P. Giannozzi, and S. Baroni, in preparation.

[46] M. Marsili and P. Umari, Phys. Rev. B **87**, 205110 (2013).

[47] J. Paier, R. Hirschl, M. Marsman, and G. Kresse, J. Chem. Phys. **122**, 234102 (2005).

[48] D. Anil, L. Lin, and Y. Lexing, J. Chem. Theory Comput. **11**, 1463 (2015).

[49] D. Anil, L. Lin, and Y. Lexing, (2017), SIAM J. Sci. Comput., in press.

[50] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997).

[51] M. Sharma, Y. Wu, and R. Car, Int. J. Quantum Chem. **95**, 821 (2003).

[52] B. Santra, R. A. DiStasio Jr., F. Martelli, and R. Car, Mol. Phys. **113**, 2829 (2015).

[53] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).

[54] R. H. French, V. A. Parsegian, R. Podgornik, R. F. Rajter, A. Jagota, J. Luo, D. Asthagiri, M. K. Chaudhury, Y.-m. Chiang, S. Granick, S. Kalinin, M. Kardar, R. Kjellander, D. C. Langreth, J. Lewis, S. Lustig, D. Wesolowski, J. S. Wettlaufer, W.-Y. Ching, M. Finnis, F. Houlihan, O. A. von Lilienfeld, C. J. van Oss, and T. Zemb, Rev. Mod. Phys. **82**, 1887 (2010).

[55] S. Grimme, J. Antony, S. Ehrlich, and S. Krieg, J. Chem. Phys. **132**, 154104 (2010).

[56] A. Tkatchenko, R. A. DiStasio Jr., R. Car, and M. Scheffler, Phys. Rev. Lett. **108**, 236402 (2012).

[57] A. Ambrosetti, A. M. Reilly, R. A. DiStasio Jr., and A. Tkatchenko, J. Chem. Phys. **140**, 18A508 (2014).

[58] M. A. Blood-Forsythe, T. Markovich, R. A. DiStasio Jr., R. Car, and A. Aspuru-Guzik, Chem. Sci.

7, 1712 (2016).

[59] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, Phys. Rev. Lett. **92**, 246401 (2004).

[60] D. C. Langreth and J. P. Perdew, Phys. Rev. B **15**, 2884 (1977).

[61] T. Thonhauser, V. R. Cooper, S. Li, A. Puzder, P. Hyldgaard, and D. C. Langreth, Phys. Rev. B **76**, 125112 (2007).

[62] G. Román-Pérez and J. M. Soler, Phys. Rev. Lett. **103**, 096102 (2009).

[63] R. Sabatini, E. Küçükbenli, B. Kolb, T. Thonhauser, and S. de Gironcoli, J. Phys. Condens. Matter **24**, 424209 (2012).

[64] T. Thonhauser, S. Zuluaga, C. A. Arter, K. Berland, E. Schröder, and P. Hyldgaard, Phys. Rev. Lett. **115**, 136402 (2015).

[65] V. R. Cooper, Phys. Rev. B **81**, 161104 (2010).

[66] J. Klimeš, D. R. Bowler, and A. Michaelides, J. Phys. Condens. Matter **22**, 022201 (2010).

[67] J. Klimeš, D. R. Bowler, and A. Michaelides, Phys. Rev. B **83**, 195131 (2011).

[68] K. Berland and P. Hyldgaard, Phys. Rev. B **89**, 035412 (2014).

[69] K. Lee, E. D. Murray, L. Kong, B. I. Lundqvist, and D. C. Langreth, Phys. Rev. B **82**, 081101 (2010).

[70] I. Hamada and M. Otani, Phys. Rev. B **82**, 153412 (2010).

[71] O. A. Vydrov and T. Van Voorhis, J. Chem. Phys. **133**, 244103 (2010).

[72] R. Sabatini, T. Gorni, and S. de Gironcoli, Phys. Rev. B **87**, 041108(R) (2013).

[73] `http://schooner.chem.dal.ca`.

[74] A. Becke, J. Chem. Phys. **85**, 7184 (1986).

[75] J. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[76] J. Perdew and W. Yue, Phys. Rev. B **33**, 8800 (1986).

[77] A. Otero-de-la-Roza and E. R. Johnson, J. Chem. Phys. **136**, 174109 (2012).

[78] F. L. Hirshfeld, Theor. Chim. Acta **44**, 129 (1977).

[79] J. Hermann, R. A. DiStasio Jr., and A. Tkatchenko, Chem. Rev. **117**, 4714 (2017).

[80] N. Ferri, R. A. DiStasio Jr., A. Ambrosetti, R. Car, and A. Tkatchenko, Phys. Rev. Lett. **114**, 176802 (2015).

[81] M. Cococcioni and S. de Gironcoli, Phys. Rev. B **71**, 035105 (2005).

[82] B. Himmetoglu, A. Floris, S. de Gironcoli, and M. Cococcioni, Int. J. Quantum Chem. **114**, 14 (2014).

[83] S. L. Dudarev, G. A. Botton, S. Y. Savrasov, C. J. Humphreys, and A. P. Sutton, Phys. Rev. B **57**, 1505 (1998).

[84] A. I. Liechtenstein, V. I. Anisimov, and J. Zaanen, Phys. Rev. B **52**, R5467 (1995).

[85] I. Timrov, M. Cococcioni, and N. Marzari, in preparation.

[86] H. F. Wilson, F. Gygi, and G. Galli, Phys. Rev. B **78**, 113303 (2008).

[87] H.-V. Nguyen and S. de Gironcoli, Phys. Rev. B **79**, 205114 (2009).

[88] N. Colonna, M. Hellgren, and S. de Gironcoli, Phys. Rev. B **90**, 125150 (2014).

[89] N. L. Nguyen, N. Colonna, and S. de Gironcoli, Phys. Rev. B **90**, 045138 (2014).

[90] S. Baroni, P. Giannozzi, and A. Testa, Phys. Rev. Lett. **58**, 1861 (1987).

[91] P. Giannozzi, S. de Gironcoli, P. Pavone, and S. Baroni, Phys. Rev. B **43**, 7231 (1991).

[92] X. Gonze, Phys. Rev. A **52**, 1096 (1995).

[93] S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, Rev. Mod. Phys. **73**, 515 (2001).

[94] R. M. Sternheimer, Phys. Rev. **96**, 951 (1954).

[95] G. D. Mahan, Phys. Rev. A **22**, 1780 (1980).

[96] C. Schwartz and J. Tiemann, Ann. Phys. **2**, 178 (1959).

[97] W. Zernik, Phys. Rev. **135**, A51 (1964).

[98] S. Baroni and A. Quattropani, Il Nuovo Cimento D **5**, 89 (1985).

[99] M. Casida, in *Recent Developments and Applications of Modern Density Functional Theory*, edited by J. Seminario (Elsevier, Amsterdam, 1996) p. 391.

[100] C. Jamorski, M. E. Casida, and D. R. Salahub, J. Chem. Phys. **104**, 5134 (1996).

[101] A. D. McLachlan and M. A. Ball, Rev. Mod. Phys. **36**, 844 (1964).

[102] H. Hübener and F. Giustino, J. Chem. Phys. **141** (2014).

[103] D. Rocca, D. Lu, and G. Galli, J. Chem. Phys. **133**, 164109 (2010).

[104] D. Rocca, Y. Ping, R. Gebauer, and G. Galli, Phys. Rev. B **85**, 045116 (2012).

[105] M. Marsili, E. Mosconi, F. De Angelis, and P. Umari, Phys. Rev. B **95**, 075415 (2017).

[106] M. Govoni and G. Galli, J. Chem. Theory Comput. **11**, 2680 (2015).

[107] R. Sabatini, E. Küçükbenli, C. H. Pham, and S. de Gironcoli, Phys. Rev. B **93**, 235120 (2016).

[108] A. Floris, S. de Gironcoli, E. K. U. Gross, and M. Cococcioni, Phys. Rev. B **84**, 161102(R) (2011).

[109] A. Floris, I. Timrov, N. Marzari, S. de Gironcoli, and M. Cococcioni, in preparation.

[110] M. Blanchard, E. Balan, P. Giura, K. Beneut, H. Yi, G. Morin, C. Pinilla, M. Lazzeri, and A. Floris, Physics and Chemistry of Minerals **41**, 289 (2014).

[111] M. Blanchard, N. Dauphas, M. Hu, M. Roskosz, E. Alp, D. Golden, C. Sio, F. Tissot, J. Zhao, L. Gao, R. Morris, M. Fornace, A. Floris, M. Lazzeri, and E. Balan, Geochimica et Cosmochimica Acta **151**, 19 (2014).

[112] G. Shukla, Z. Wu, H. Hsu, A. Floris, M. Cococcioni, and R. M. Wentzcovitch, Geophysical Research Letters **42**, 1741 (2015).

[113] G. Shukla and R. M. Wentzcovitch, Physics of the Earth and Planetary Interiors **260**, 53 (2016).

[114] G. Shukla, M. Cococcioni, and R. M. Wentzcovitch, Geophysical Research Letters **43**, 5661 (2016).

[115] E. Runge and E. Gross, Phys. Rev. Lett. **52**, 997 (1984).

[116] M. A. L. Marques, N. T. Maitra, F. M. S. Nogueira, E. K. U. Gross, and A. Rubio, eds., *Fundamentals of Time-Dependent Density Functional Theory*, Vol. 837 (Lecture Notes in Physics, Springer-Verlag, Berlin, Heidelberg, 2012).

[117] S. Baroni and R. Gebauer, *The Liouville-Lanczos Approach to Time-Dependent Density-Functional (Perturbation) Theory* (Ref. 116, chapter 19, p. 375-390).

[118] T. Gorni, I. Timrov, and S. Baroni, in preparation.

[119] S. Baroni and R. Resta, Phys. Rev. B **33**, 7017 (1986).

[120] J. Tobik and A. Dal Corso, J. Chem. Phys. **120**, 9934 (2004).

[121] R. Haydock, V. Heine, and M. J. Kelly, Journal of Physics C: Solid State Physics **5**, 2845 (1972).

[122] R. Haydock, V. Heine, and M. J. Kelly, Journal of Physics C: Solid State Physics **8**, 2591 (1975).

[123] M. Grüning, A. Marini, and X. Gonze, Computational Materials Science **50**, 2148 (2011).

[124] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).

[125] G. Onida, L. Reining, and A. Rubio, Rev. Mod. Phys. **74**, 601 (2002).

[126] I. Timrov, M. Markov, T. Gorni, M. Raynaud, O. Motornyi, R. Gebauer, S. Baroni, and N. Vast, Phys. Rev. B **95**, 094301 (2017).

[127] L. Hedin, Phys. Rev. **139**, A796 (1965).

[128] M. S. Hybertsen and S. G. Louie, Phys. Rev. Lett. **55**, 1418 (1985).

[129] L. Reining, G. Onida, and R. W. Godby, Phys. Rev. B **56**, R4301 (1997).

[130] H. F. Wilson, D. Lu, F. Gygi, and G. Galli, Phys. Rev. B **79**, 245106 (2009).

[131] F. Giustino, M. L. Cohen, and S. G. Louie, Phys. Rev. B **81**, 115105 (2010).

[132] P. Umari and S. Fabris, J. Chem. Phys. **136**, 174310 (2012).

[133] P. Umari, E. Mosconi, and F. De Angelis, Scientific Reports **4**, 4467 EP (2014).

[134] F. Caruso, H. Lambert, and F. Giustino, Phys. Rev. Lett. **114**, 146404 (2015).

[135] H. Lambert and F. Giustino, Phys. Rev. B **88**, 075117 (2013).

[136] C. J. Pickard and F. Mauri, Phys. Rev. B **63**, 245101 (2001).

[137] M. d'Avezac, N. Marzari, and F. Mauri, Phys. Rev. B **76**, 165122 (2007).

[138] C. J. Pickard and F. Mauri, Phys. Rev. Lett. **88**, 086403 (2002).

[139] H. M. Petrilli, P. E. Blöchl, P. Blaha, and K. Schwarz, Phys. Rev. B **57**, 14690 (1998).

[140] J. W. Zwanziger, J. Phys.: Condens. Matter **21**, 195501 (2009).

[141] M. S. Bahramy, M. H. F. Sluiter, and Y. Kawazoe, Phys. Rev. B **76**, 035124 (2007).

[142] H. J. von Bardeleben, J. L. Cantin, H. Vrielinck, F. Callens, L. Binet, E. Rauls, and U. Gerstmann, Phys. Rev. B **90**, 085203 (2014).

[143] R. Pigliapochi, A. J. Pell, I. D. Seymour, C. P. Grey, D. Ceresoli, and M. Kaupp, Phys. Rev. B **95**, 054412 (2017).

[144] J. R. Yates, C. J. Pickard, and F. Mauri, Phys. Rev. B **76**, 024401 (2007).

[145] E. Küçükbenli, K. Sonkar, N. Sinha, and S. de Gironcoli, J. Phys. Chem. A **116**, 3765 (2012).

[146] S. de Gironcoli, Phys. Rev. B **51**, 6773 (1995).

[147] D. Xiao, J. Shi, and Q. Niu, Phys. Rev. Lett. **95**, 137204 (2005).

[148] T. Thonhauser, D. Ceresoli, D. Vanderbilt, and R. Resta, Phys. Rev. Lett. **95**, 137205 (2005).

[149] T. Thonhauser, D. Ceresoli, A. A. Mostofi, N. Marzari, R. Resta, and D. Vanderbilt, J. Chem. Phys. **131**, 101101 (2009).

[150] D. Ceresoli, U. Gerstmann, A. P. Seitsonen, and F. Mauri, Phys. Rev. B **81**, 060409 (2010).

[151] B. M. George, J. Behrends, A. Schnegg, T. F. Schulze, M. Fehr, L. Korte, B. Rech, K. Lips, M. Rohrmüller, E. Rauls, W. G. Schmidt, and U. Gerstmann, Phys. Rev. Lett. **110**, 136803 (2013).

[152] Z. Bodrog and A. Gali, J. Phys.: Condens. Matter **26**, 015305 (2014).

[153] C. Gougoussis, M. Calandra, A. P. Seitsonen, and F. Mauri, Phys. Rev. B **80**, 075102 (2009).

[154] C. Gougoussis, M. Calandra, A. Seitsonen, C. Brouder, A. Shukla, and F. Mauri, Phys. Rev. B **79**, 045118 (2009).

[155] O. Bunău and M. Calandra, Phys. Rev. B **87**, 205105 (2013).

[156] M. Taillefumier, D. Cabaret, A.-M. Flank, and F. Mauri, Phys. Rev. B **66**, 195107 (2002).

[157] G. Fratesi, V. Lanzilotto, L. Floreano, and G. P. Brivio, J. Phys. Chem. C **117**, 6632 (2013).

[158] G. Fratesi, V. Lanzilotto, S. Stranges, M. Alagia, G. P. Brivio, and L. Floreano, Phys. Chem. Chem. Phys. **16**, 14834 (2014).

[159] M. Lazzeri and S. de Gironcoli, Phys. Rev. B **65**, 245402 (2002).

[160] G. Deinzer, G. Birner, and D. Strauch, Phys. Rev. B **67**, 144304 (2003).

[161] M. Calandra, M. Lazzeri, and F. Mauri, Physica C: Superconductivity **456**, 38 (2007), Recent Advances in MgB_2 Research.

[162] J. Callaway, Phys. Rev. **113**, 1046 (1959).

[163] M. Markov, J. Sjakste, G. Fugallo, L. Paulatto, M. Lazzeri, F. Mauri, and N. Vast, Phys. Rev. B **93**, 064301 (2016).

[164] M. Markov, J. Sjakste, G. Fugallo, L. Paulatto, M. Lazzeri, F. Mauri, and N. Vast, submitted.

[165] A. Ward, D. A. Broido, D. A. Stewart, and G. Deinzer, Phys. Rev. B **80**, 125203 (2009).

[166] G. Fugallo, A. Cepellotti, L. Paulatto, M. Lazzeri, N. Marzari, and F. Mauri, Nano Letters **14**, 6109 (2014).

[167] A. Cepellotti, G. Fugallo, L. Paulatto, M. Lazzeri, F. Mauri, and N. Marzari, Nat. Commun. **6** (2015).

[168] W. Li, J. Carrete, N. A. Katcho, and N. Mingo, Comp. Phys. Commun. **185**, 17471758 (2014).

[169] F. Giustino, Rev. Mod. Phys. **89**, 015003 (2017).

[170] N. Marzari, A. A. Mostofi, J. R. Yates, I. Souza, and D. Vanderbilt, Rev. Mod. Phys. **84**, 1419 (2012).

[171] A. A. Mostofi, J. R. Yates, G. Pizzi, Y. S. Lee, I. Souza, D. Vanderbilt, and N. Marzari, Comput. Phys. Commun. **185**, 2309 (2014).

[172] L. A. Agapito, S. Curtarolo, and M. Buongiorno Nardelli, Physical Review X **5**, 011006 (2015).

[173] A. Calzolari and M. Buongiorno Nardelli, Scientific Reports **3** (2013).

[174] P. Umari and A. Pasquarello, Phys. Rev. Lett. **89**, 157602 (2002).

[175] Y. Wang, S. Shang, Z.-K. Liu, and L.-Q. Chen, Phys. Rev. B **85**, 224303 (2012).

[176] R. King-Smith and D. Vanderbilt, Phys. Rev. B **47**, 1651 (1993).

[177] P. Umari, X. Gonze, and A. Pasquarello, Phys. Rev. Lett. **90**, 027401 (2003).

[178] J. Tomasi, B. Mennucci, and R. Cammi, Chem. Rev. **105**, 2999 (2005).

[179] J.-L. Fattebert and F. Gygi, J. Comput. Chem. **23**, 662 (2002).

[180] G. Fisicaro, L. Genovese, O. Andreussi, N. Marzari, and S. Goedecker, J. Chem. Phys. **144**, 014103

(2016).

[181] C. Dupont, O. Andreussi, and N. Marzari, J. Chem. Phys. **139**, 214110 (2013).

[182] O. Andreussi, I. Dabo, G. Fisicaro, S. Goedecker, I. Timrov, S. Baroni, and N. Marzari, "Environ 0.2: a library for environment effect in first-principles simulations of materials," `www.quantum-environ.org` (2016).

[183] M. Cococcioni, F. Mauri, G. Ceder, and N. Marzari, Phys. Rev. Lett. **94**, 145501 (2005).

[184] D. A. Scherlis, J. L. Fattebert, F. Gygi, M. Cococcioni, and N. Marzari, J. Chem. Phys. **124**, 074103 (2006).

[185] I. Dabo, B. Kozinsky, N. E. Singh-Miller, and N. Marzari, Phys. Rev. B **77**, 115139 (2008).

[186] A. Laio, J. VandeVondele, and U. Rothlisberger, J. Chem. Phys. **116**, 6941 (2002).

[187] A. H. MacDonald and S. H. Vosko, J. Phys. C **12**, 2977 (1979).

[188] A. K. Rajagopal and J. Callaway, Phys. Rev. B **7**, 1912 (1973).

[189] T. Brumme, M. Calandra, and F. Mauri, Phys. Rev. B **89**, 245406 (2014).

[190] L. Bengtsson, Phys. Rev. B **59**, 12301 (1999).

[191] M. Topsakal and S. Ciraci, Phys. Rev. B **85**, 045121 (2012).

[192] J. Neugebauer and M. Scheffler, Phys. Rev. B **46**, 16067 (1992).

[193] B. Meyer and D. Vanderbilt, Phys. Rev. B **63**, 205426 (2001).

[194] I. Štich, R. Car, M. Parrinello, and S. Baroni, Phys. Rev. B **39**, 4997 (1989).

[195] O. Jepsen and O. K. Andersen, Solid State Commun. **9**, 1763 (1971).

[196] P. E. Blöchl, O. Jepsen, and O. K. Andersen, Phys. Rev. B **49**, 16223 (1994).

[197] M. Kawamura, Y. Gohda, and S. Tsuneyuki, Phys. Rev. B **89**, 094515 (2014).

[198] F. Zadra and A. Dal Corso, unpublished.

[199] T. Hahn, ed., *International tables for crystallography Volume A: Space-group symmetry* (Springer, 2005).

[200] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, J. Phys.: Condens. Matter **26**, 213201 (2014).

[201] A. Marini, C. Hogan, M. Grüning, and D. Varsano, Comput. Phys. Commun. **180**, 1392 (2009).

[202] L. Martin-Samos and G. Bussi, Comput. Phys. Commun. **180**, 1416 (2009).

[203] J. Deslippe, G. Samsonidze, D. A. Strubbe, M. Jain, M. L. Cohen, and S. G. Louie, Comput. Phys. Commun. **183**, 1269 (2012).

[204] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, Computational Materials Science **111**, 218 (2016).

[205] N. Mounet, M. Gibertini, P. Schwaller, A. Merkys, I. E. Castelli, A. Cepellotti, G. Pizzi, and N. Marzari, (2016), arXiv:1611.05234 [cond-mat].

[206] A. R. Supka, T. E. Lyons, L. Liyanage, P. D'Amico, R. A. R. Al Orabi, S. Mahatara, P. Gopal, C. Toher, D. Ceresoli, A. Calzolari, S. Curtarolo, M. Buongiorno Nardelli, and M. Fornari, Computational Materials Science **136**, 76 (2017).

[207] The HDF Group, "Hierarchical data format version 5," `http://www.hdfgroup.org/HDF5` (2000-2010).

[208] A. Calzolari, I. Souza, N. Marzari, and M. Buongiorno Nardelli, Phys. Rev. B **69**, 035108 (2004).

[209] A. Ferretti, A. Calzolari, B. Bonferroni, and R. Di Felice, J. Phys.: Condens. Matter **19**, 036215 (2007).

[210] M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello, Comput. Phys. Commun. **180**, 1961 (2009).

[211] S. R. Bahn and K. W. Jacobsen, Comput. Sci. Eng. **4**, 56 (2002).

[212] D. A. Moon, G. L. Steele Jr., and R. Stallman et al., "Emacs: The extensible, customizable display editor," `https://www.gnu.org/software/emacs/` (2017).

[213] J. Spencer, "testcode," `https://github.com/jsspencer/testcode` (2017).

[214] N. Marzari, Nature Materials **15**, 381 (2016).