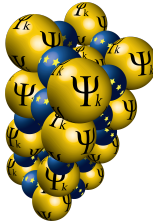


ICTP/Psi-k/CECAM School on Electron-Phonon Physics from First Principles

Trieste, 19-23 March 2018



Lecture Monday 19

Introduction to QUANTUM ESPRESSO

Paolo Giannozzi

Dip. Matematica Informatica e Fisica, Università di Udine
CNR-IOM Democritos, Trieste, Italy

The QUANTUM ESPRESSO distribution

QUANTUM ESPRESSO (QE) stands for *Quantum opEn-Source Package for Research in Electronic Structure, Simulation, and Optimization*

QE is a distribution (an integrated suite) of software for *first-principle simulations*, i.e., atomistic calculations based on electronic structure, using density-functional theory, a plane-wave basis set, pseudopotentials.

QE is freely available under the terms of the GNU General Public License

Main goals of QE are

- *innovation* in theoretical methods and numerical algorithms
- *efficiency* on modern computer architectures

A great effort is also devoted to *user friendliness* and to the formation of a *users' and developers' community*

QE exists since 2002, resulting from the merge of pre-existing packages; some core components have been under development for ~ 30 years

QUANTUM ESPRESSO contributors

QE is one of the community codes of H2020 project MaX – *Materials at the Exascale*, receives contributions from many individuals and partner institutions in Europe and worldwide. Who “owns” QE ... ?

... the QUANTUM ESPRESSO Foundation: a non-profit (“limited by guarantee”) company, based in London, that

- coordinates and supports research, education, and outreach within the QE community
- owns the trademarks and protects the open-source character of QE
- raises funds to foster the QE project and its development

Current members of the Foundation:

SISSA, EPFL, ICTP, IOM-CNR, Cineca, North Texas University, Oxford University

Users' community: factoids

- 1600+ registered users for the users' mailing list
- An average of ~ 8 messages a days on the mailing list
- Latest version downloaded 9500 times in less than two months [*]
- 30+ Schools or tutorials since 2002, attended by ~ 1200 users
- 4 developers' schools since 2013, latest in 2017
- Annual developers' meeting since 2010

[*] Number may be inflated by bots, failed or repeated downloads, etc.

QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials

**Paolo Giannozzi^{1,2}, Stefano Baroni^{1,3}, Nicola Bonini⁴,
Matteo Calandra⁵, Roberto Car⁶, Carlo Cavazzoni^{7,8},
Davide Ceresoli⁴, Guido L Chiarotti⁹, Matteo Cococcioni¹⁰,
Ismaila Dabo¹¹, Andrea Dal Corso^{1,3}, Stefano de Gironcoli^{1,3},
Stefano Fabris^{1,3}, Guido Fratesi¹², Ralph Gebauer^{1,13},
Uwe Gerstmann¹⁴, Christos Gougoussis⁵, Anton Kokalj^{1,15},
Michele Lazzeri⁵, Layla Martin-Samos¹, Nicola Marzari⁴,
Francesco Mauri⁵, Riccardo Mazzarello¹⁶, Stefano Paolini^{3,9},
Alfredo Pasquarello^{17,18}, Lorenzo Paulatto^{1,3}, Carlo Sbraccia^{1,†},
Sandro Scandolo^{1,13}, Gabriele Sclauzero^{1,3}, Ari P Seitsonen⁵,
Alexander Smogunov¹³, Paolo Umari¹ and
Renata M Wentzcovitch^{10,19}**

This is the main documenting paper. After 8 years and ~ 6000 citations ...

Advanced capabilities for materials modelling with QUANTUM ESPRESSO

P Giannozzi¹, O Andreussi^{2,9}, T Brumme³, O Bunau⁴, M Buongiorno Nardelli⁵, M Calandra⁴, R Car⁶, C Cavazzoni⁷, D Ceresoli⁸, M Cococcioni⁹, N Colonna⁹, I Carnimeo¹, A Dal Corso^{10,32}, S de Gironcoli^{10,32}, P Delugas¹⁰, R A DiStasio Jr¹¹, A Ferretti¹², A Floris¹³, G Fratesi¹⁴, G Fugallo¹⁵, R Gebauer¹⁶, U Gerstmann¹⁷, F Giustino¹⁸, T Gorni^{4,10}, J Jia¹¹, M Kawamura¹⁹, H-Y Ko⁶, A Kokalj²⁰, E Küçükbenli¹⁰, M Lazzeri⁴, M Marsili²¹, N Marzari⁹, F Mauri²², N L Nguyen⁹, H-V Nguyen²³, A Otero-de-la-Roza²⁴, L Paulatto⁴, S Poncé¹⁸, D Rocca^{25,26}, R Sabatini²⁷, B Santra⁶, M Schlipf¹⁸, A P Seitsonen^{28,29}, A Smogunov³⁰, I Timrov⁹, T Thonhauser³¹, P Umari^{21,32}, N Vast³³, X Wu³⁴ and S Baroni¹⁰

... a new version is out. What happened meanwhile?

Requirements on effective software for quantum simulations at the nanoscale

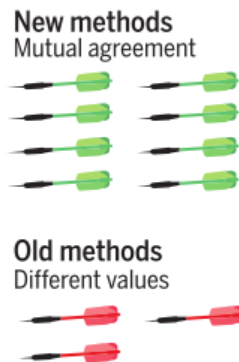
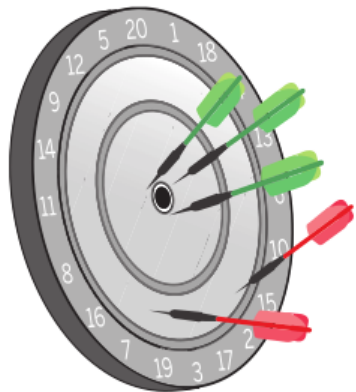
- Diffusion of first-principle techniques among non-specialists requires software that is **easy to use and (reasonably) error-proof**
- Challenging calculations stress the limits of available computer power: software should be **fast and efficient**
- Introducing innovation requires new ideas to materialize into new algorithms through codes: software should be **easy to extend and to improve**
- Complex problems require a mix of solutions coming from different approaches and methods: software should be **interoperable with other software**
- Finally, scientific ethics requires that **results should be reproducible and algorithms susceptible of validation**

Verification and Validation of electronic-structure codes

Systematic comparisons of different pseudopotential and all-electron DFT codes: *Reproducibility in density-functional theory calculations of solids*, K. Lejaeghere *et multis aliis*, Science **351** (6280), aad3000 (2016), DOI 10.1126/science.aad3000

Tests *precision* of the computational methods, not physical *accuracy* of results.
Main outcome: everybody is converging towards the same set of results.

Recent DFT methods yield reproducible results. Whereas older DFT implementations predict different values (red darts), codes have now evolved to mutual agreement (green darts). The scoreboard illustrates the good pairwise agreement of four classes of DFT implementations (horizontal direction) with all-electron results (vertical direction). Each number reflects the average difference between the equations of state for a given pair of methods, with the green-to-red color scheme showing the range from the best to the poorest agreement.



Scorecard

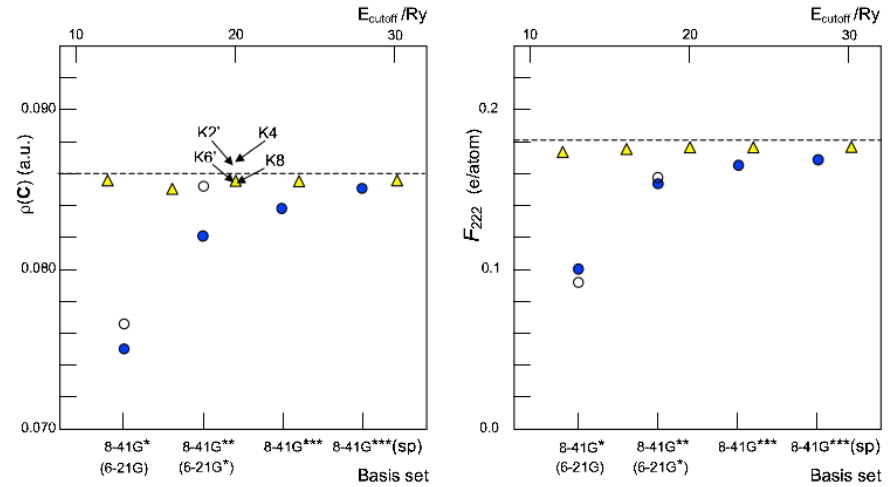
	0.3	0.3	0.6	1.0	0.9	0.3	1.5	0.6	0.9	0.4	1.0	0.4	0.4	6.3	13.5	1.1	2.1	0.7	1.4
0.3		0.1	0.5	0.9	0.8	0.2	1.5	0.6	0.8	0.4	1.0	0.5	0.3	6.3	13.4	1.1	2.1	0.7	1.4
0.3	0.1		0.5	0.9	0.8	0.2	1.5	0.6	0.8	0.4	0.9	0.5	0.3	6.3	13.4	1.1	2.1	0.7	1.4
0.6	0.5	0.5		0.8	0.6	0.4	1.5	0.6	0.8	0.6	1.0	0.7	0.5	6.3	13.2	1.0	1.9	0.6	1.3
1.0	0.9	0.9	0.8		0.9	0.9	1.8	0.9	1.3	1.0	1.4	1.0	0.9	6.4	13.0	1.2	1.8	1.0	1.6
0.9	0.8	0.8	0.6	0.9		0.8	1.7	0.7	1.1	0.8	1.3	1.0	0.8	6.5	13.2	1.1	1.8	0.8	1.5
0.3	0.2	0.2	0.4	0.9	0.8		1.5	0.5	0.8	0.3	1.0	0.5	0.3	6.2	13.4	1.0	2.0	0.6	1.4

Comparing QE with Gaussian-based code CRYSTAL

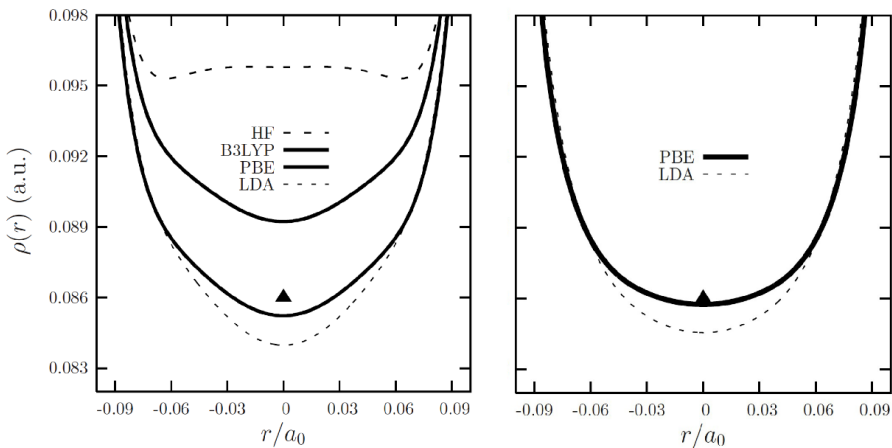
Chapter 2 Electron Densities and Related Properties from the *ab-initio* Simulation of Crystalline Solids

Cesare Pisani, Roberto Dovesi, Alessandro Erba, and Paolo Giannozzi

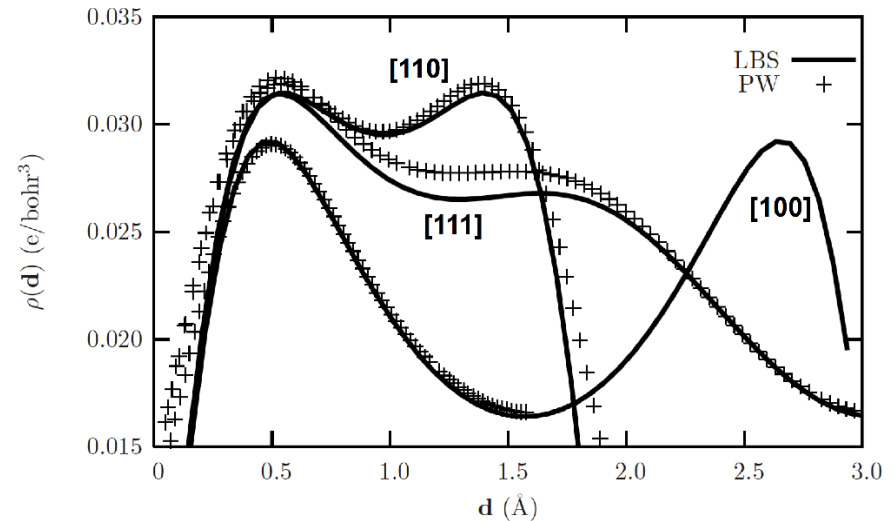
C. Gatti and P. Macchi (eds.), *Modern Charge-Density Analysis*,
DOI 10.1007/978-90-481-3836-4_2, © Springer Science+Business Media B.V. 2012



Effect of the basis set



Comparison of charge density in Si



Comparison of charge density in Al

Requirements on effective software for quantum simulations at the nanoscale

- Diffusion of first-principle techniques among non-specialists requires software that is **easy to use and (reasonably) error-proof**
- Challenging calculations stress the limits of available computer power: software should be **fast and efficient**
- Introducing innovation requires new ideas to materialize into new algorithms through codes: software should be **easy to extend and to improve**
- Complex problems require a mix of solutions coming from different approaches and methods: software should be **interoperable with other software**
- Finally, scientific ethics requires that **results should be reproducible and algorithms susceptible of validation**

Solutions for interoperability

- I/O with schema-based, standard-compliant XML file, plus binary files (optionally in portable HDF5 format) for large records (e.g. wavefunctions, charge density). Allows easy parsing and transferral of data both inside QE and between QE and external software.
- More modular code and parallelization logic. Allows to call QE code as a library and to execute it inside a MPI communicator provided by the external software

Applications:

- QM-MM, with LAMMPS for the MM part (Comput. Phys. Commun. *195*, 191 (2015), new version using MPI still under development)
- Advanced minimization algorithms (basin hopping, genetic algorithms)
- Path-Integral Molecular Dynamics with i-Pi (CPC *185*, 1019 (2014))
- High-throughput computing with AiiDA (Comput. Mater. Sci. *111* 218 (2016))

Requirements on effective software for quantum simulations at the nanoscale

- Diffusion of first-principle techniques among non-specialists requires software that is **easy to use and (reasonably) error-proof**
- Challenging calculations stress the limits of available computer power: software should be **fast and efficient**
- Introducing innovation requires new ideas to materialize into new algorithms through codes: software should be **easy to extend and to improve**
- Complex problems require a mix of solutions coming from different approaches and methods: software should be **interoperable with other software**
- Finally, scientific ethics requires that **results should be reproducible and algorithms susceptible of validation**

Major improvements and extensions

Mostly in the field of “advanced functionals”:

- New methods for van-der-Waals-bonded systems:
 - non-local functionals (“vdw-DF”)
 - semi-empirical corrections: Grimme’s DFT+D2, DFT+D3
 - non-so-empirical corrections; Tkatchenko-Scheffler, exchange-hole dipole moment model (XDM)
- New methods to deal with hybrid functionals:
 - Adaptively Compressed Exchange, also in conjunction with Selected Columns of Density Matrix localization: [arXiv:1801.09263](https://arxiv.org/abs/1801.09263)
 - Car-Parrinello dynamics with localized Wannier functions
- Usable meta-GGA functionals

Non-local functionals in molecular crystals

New non-local (vdW-DF) functionals allow to deal with molecular crystals without semi-empirical schemes, with a computational effort comparable to plain DFT:

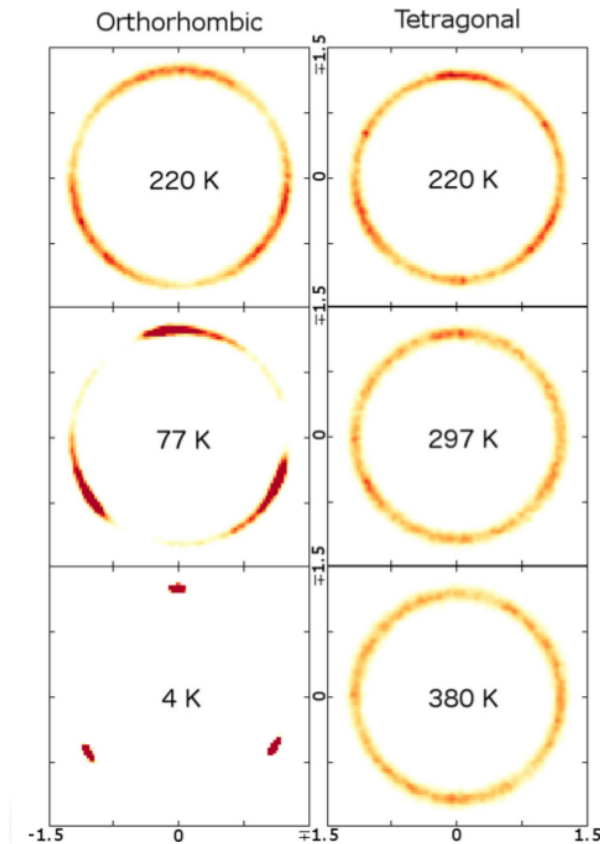
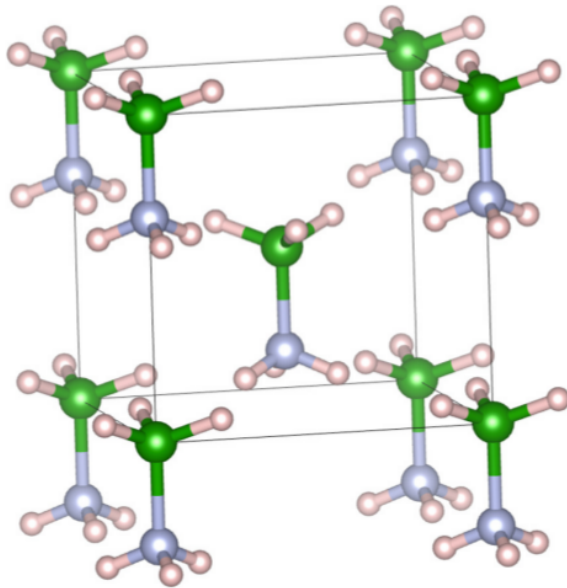
PHYSICAL REVIEW B **89**, 180101(R) (2014)

Positional disorder in ammonia borane at ambient conditions

E. Welchman,¹ P. Giannozzi,² and T. Thonhauser^{1,*}

¹Department of Physics, Wake Forest University, Winston-Salem, North Carolina 27109, USA

²Department of Chemistry, Physics, and Environment, University of Udine, 33100 Udine, Italy



First-principle molecular dynamics explains fourfold symmetry axis at ambient conditions, apparently inconsistent with three-fold symmetry of NH₃BH₃ molecules

Requirements on effective software for quantum simulations at the nanoscale

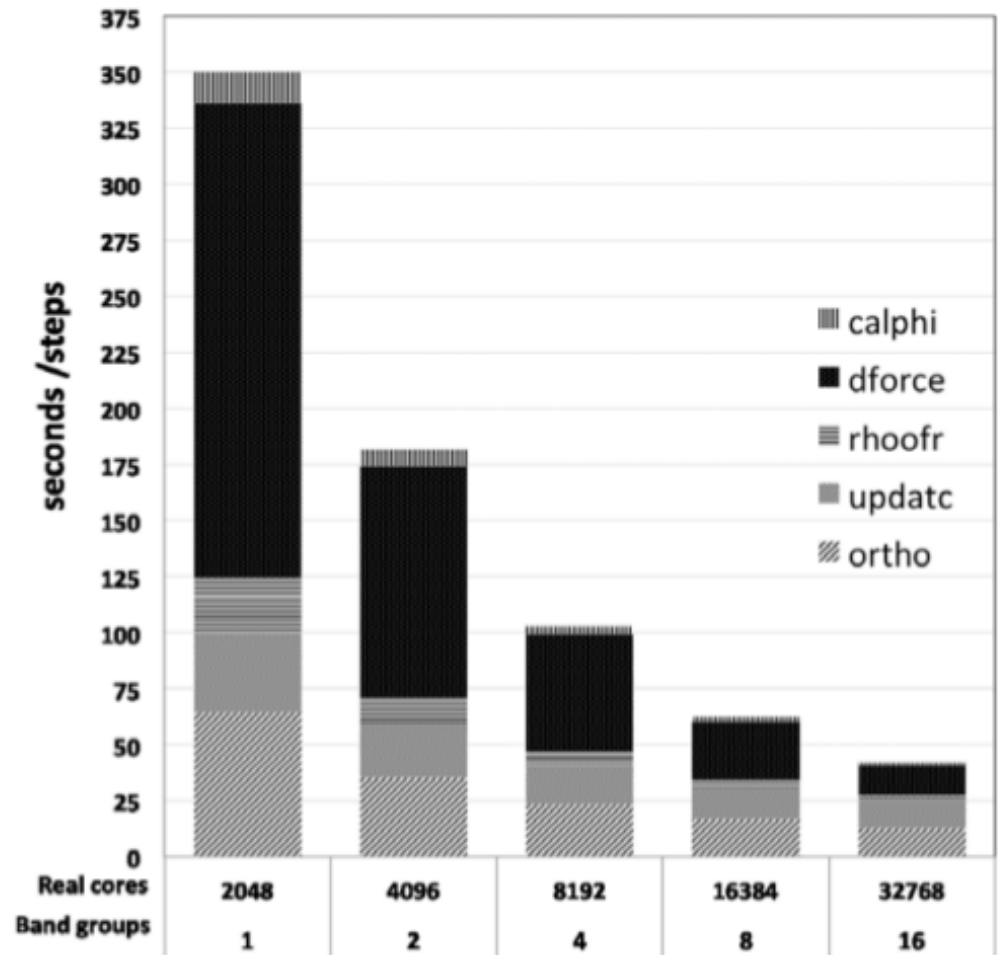
- Diffusion of first-principle techniques among non-specialists requires software that is **easy to use and (reasonably) error-proof**
- Challenging calculations stress the limits of available computer power: software should be **fast and efficient**
- Introducing innovation requires new ideas to materialize into new algorithms through codes: software should be **easy to extend and to improve**
- Complex problems require a mix of solutions coming from different approaches and methods: software should be **interoperable with other software**
- Finally, scientific ethics requires that **results should be reproducible and algorithms susceptible of validation**

Parallelization towards the exascale

Scalability of realistic calculations on up to tens of thousands cores, using mixed MPI-OpenMP parallelization, has been demonstrated.

Careful optimization of non-scalable RAM and computations required!
Scalability *strongly depends upon the kind and size of system!*

More and more *parallelization levels* are being implemented



CP Scalability on BG/Q, 1532-atom porphyrin-functionalized carbon nanotube
(data from N. Varini *et al.*, Comput. Phys. Commun. **184**, 1827 (2013))

New architectures: GPU

“Accelerated” architectures such as NVidia GPU’s are the current “big thing” in high-performance computing.

Problem: large code rewriting needed to obtain interesting performances.

Existing porting of QE to GPU’s, using NVidia’s CUDA language, has maintainability issues: as the code evolves, the GPU version lags behind.

Solution (tentative!): rewrite selected computational kernels using CUDA Fortran. Integrates much better into the Fortran-based code of QE.

Perspectives and Outlook

- More packages for advanced methodologies
- Better-structured distribution, with interfaces to external codes and to python scripting
- Porting to new hybrid and accelerated architectures
- More parallelization everywhere, communication-reducing and latency-hiding algorithms

Credits

- Thanks to all people whose slides and pictures I borrowed
- Thanks to all people who contributed to QE
- ...and thanks to you all