



Progress in EPW

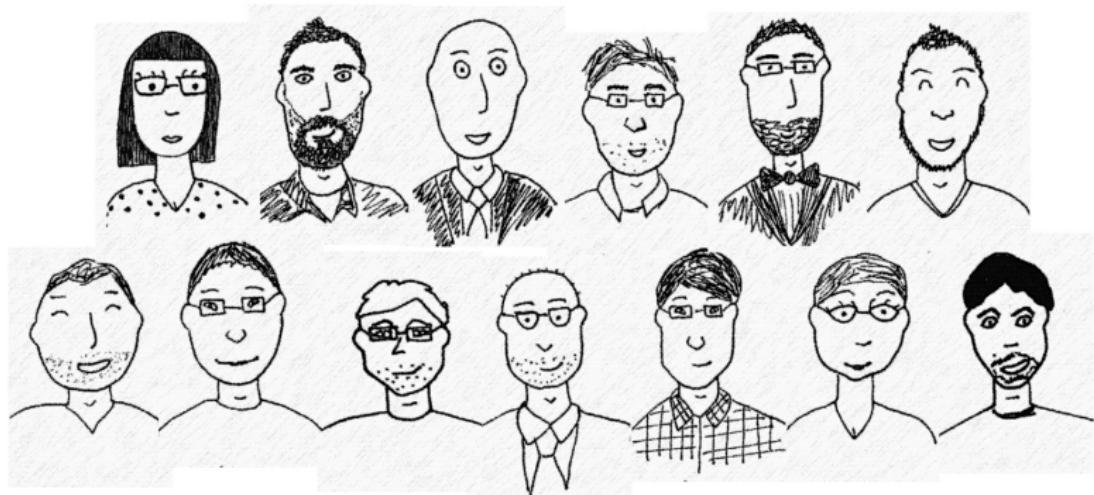
Samuel Poncé

QE Devel meeting 09-01-2017





The materials modeling and design group in Oxford

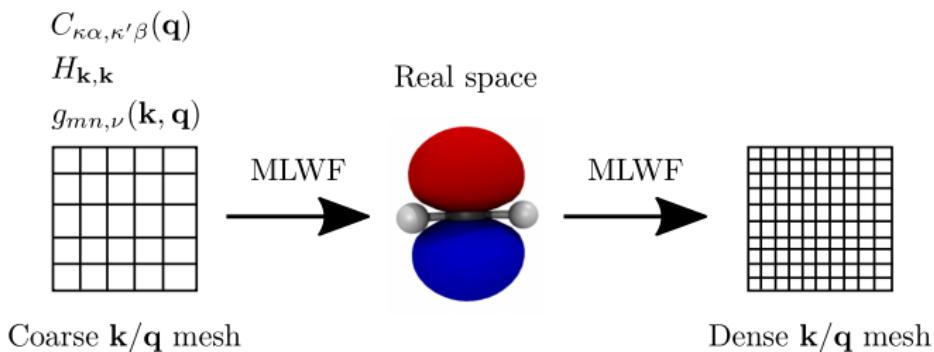


Carla, George, Feliciano, Miguel Angel, Samuel, Martin
Nourdine, Xiao-Long, Christoph, Fabio, Kun, Marina, Marios



Electron-phonon Wannier (EPW)

EPW relies on MLWF to interpolate electron-phonon matrix elements.





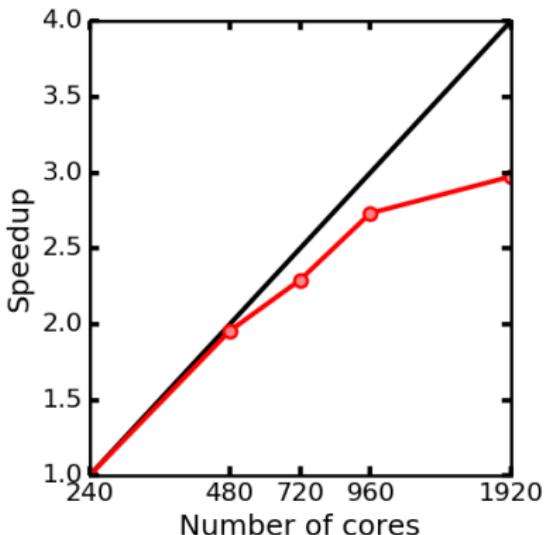
EPW developments in 2016

- ▶ Code beautification
- ▶ Correct Wannier localization with spinors
- ▶ Possibility to impose the ASR in real and reciprocal space
(require pp.x)
- ▶ New level of parallelization: band parallelism (only some part of the code). Can be extended easily.
- ▶ Speed/memory optimization
- ▶ Automatic documentation using Ford
- ▶ Increased test-farm using Buildbot + code coverage



Performance Optimisation and Productivity

- ▶ European-funded project: free of charge
- ▶ Use Score-P and Scalasca
- ▶ Load imbalance & I/O was problematic





Automatic documentation using Ford

- ▶ Easy to set-up
- ▶ Only requires double exclamation marks '!!'
- ▶ Generates good looking graphs that supports modules, subroutines etc ...
- ▶ \LaTeX support using MathJax
- ▶ Use of Markdown to type-set documentation
- ▶ Is actively developed (in Oxford)
- ▶ Example: <http://epw.org.uk/uploads/Ford2>



Automatic documentation using Ford

Fortran Program EPW overview Source Files Modules Procedures Program

Search

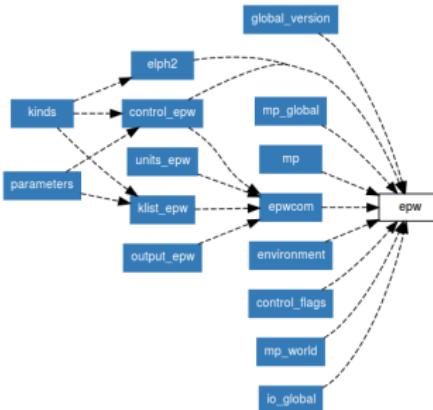
epw Program

Samuel Ponce', Roxana Margine, Carla Verdi, Feliciano Giustino GNU v4.0 | 78 statements | Source File epwf90 / epw

Variables

Uses: io_global mp mp_world mp_global control_flags control_epw global_version epwcom environment elph2

All Programs



Help

This is the main EPW driver which sets the phases on the wavefunctions, calls wann_run and elphon_shuffle_wrap

Note

8/14/08 Inscf is unnecessary, as is nqs,iq_start



Test-suite and Buildbot test-farm

Test-suite and Buildbot test-farm (Filippo@Cambridge + Samuel@Oxford) have been running for 1 year.

- ▶ Prove to be very useful in quickly detecting bugs
- ▶ Ensure long term stability of the code
- ▶ Rely on the python script testcode.py used by other codes





Test-suite and Buildbot test-farm

We need to test more parts of the code.

Important to know the code to set relevant accuracy tolerances.

Proposition:

- ▶ Iurii Timrov for TDDFT
- ▶ Paolo Umari for GWW
- ▶ Andrea Dal Corso for Phonon
- ▶ Carlo Cavazzoni for CPV
- ▶ Paolo Giannozzi for PP
- ▶ Samuel Poncé for EPW
- ▶ Others ?





Test-suite and Buildbot test-farm

I added some doc in Doc/developer_man.tex to add new tests to the QE test-suite

13.1 How to add tests for a new executable

Let us take the example of adding a new test for the TDDFPT module.

extract-PROG_NAME.x This script extracts the physical quantities from the output and parse it in a format for the testcode.py script. The script need to contain all the different output you want to parse (for chain calculations). For example, in this case we want to parse the output of `pw.x`, `turbo_lanczos.x` and `turbo_spectrum.x`. It is crucial to add as many parameter to be tested as possible to increase the code coverability.

run-PROG_NAME.sh This bash script contains the paths of the different programs and source the `ENVIRONMENT` file

jobconfig You need to edit this file to add all the new tests as well as the new program. You can chain different programs with different output in one test. In this case we added

```
[tddfpt_*]  
program = TDDFPT
```



Test-suite and Buildbot test-farm

The test-suite as a continuous integration tool.

Soft way:

- ▶ Developer receives an automatic email if their commit does not pass on the nightly Buildbot test-farm

Hard way:

- ▶ The contribution of a developer is only added to the SVN trunk if it passes the nightly Buildbot test-farm



NetCDF file format

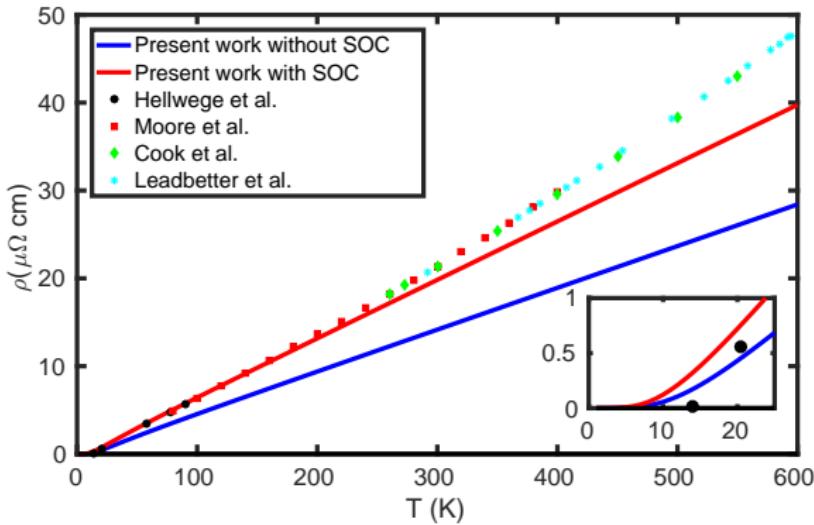
Including NetCDF support in QE:

- ▶ Code interoperability (e.g. can be read by python for scripting)
- ▶ Is portable through different architecture
- ▶ Has meta-data
- ▶ Is a type of binary file (small disk space)
- ▶ Has optimized parallel IO (netCDF-IO)
- ▶ Easier to code/use than HDF5. Do we really need all the extra HDF5 features?
- ▶ If we need HDF5, can we have a nice HDF5 wrapper?
- ▶ Having standardize functions like CALL
`write_crystal(file_name)`



New feature in EPW

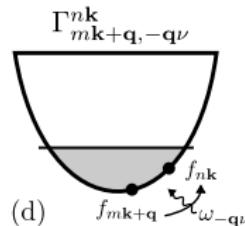
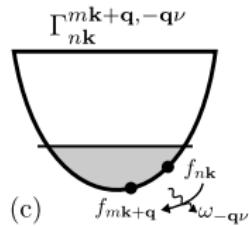
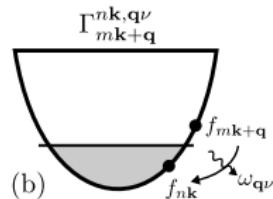
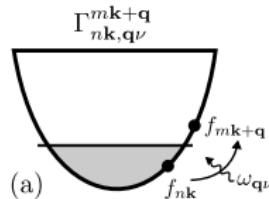
In development: carrier transport in semiconductors. Already in the code: electric resistivity and conductivity in metals.





Linearized Boltzmann transport equation

The goal is to describe the out of equilibrium distribution function f from transition rates



$$\Gamma_{n\mathbf{k},\mathbf{q}\nu}^{m\mathbf{k}+\mathbf{q}}(T) = 2\pi |g_{mn,\nu}(\mathbf{k}, \mathbf{q})|^2 f_{n\mathbf{k}}^0(T) [1 - f_{m\mathbf{k}+\mathbf{q}}^0(T)] n_{\mathbf{q}\nu}^0(T) \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_{m\mathbf{k}+\mathbf{q}} + \omega_{\mathbf{q}\nu})$$



Self-energy relaxation time approximation (SERTA)

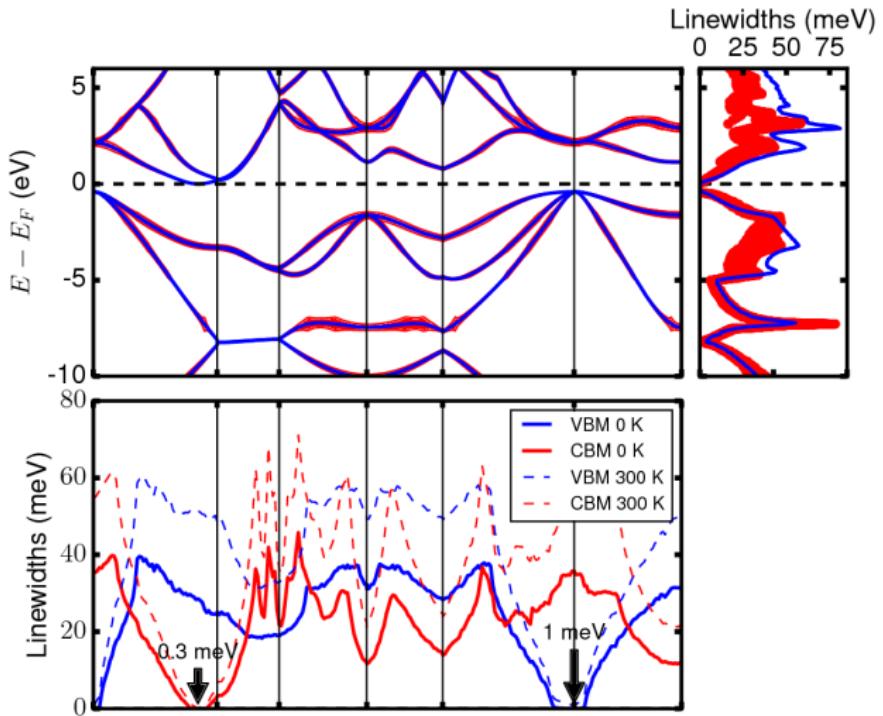
BTE has an iterative solution.

The SRTA consists in neglecting $\tau_{m\mathbf{k}+\mathbf{q}}$

$$\begin{aligned}\frac{1}{\tau_{n\mathbf{k}}^{\text{SERTA}}(T)} = 2\Sigma''_{n\mathbf{k}}(\varepsilon_{n\mathbf{k}}, T) &= \frac{1}{f_{n\mathbf{k}}^0(T)[1 - f_{n\mathbf{k}}^0(T)]} \\ &\times \sum_{m\nu} \int_{\text{BZ}} \frac{d\mathbf{q}}{\Omega_{\text{BZ}}} \left[\Gamma_{n\mathbf{k}, \mathbf{q}\nu}^{m\mathbf{k}+\mathbf{q}}(T) + \Gamma_{n\mathbf{k}}^{m\mathbf{k}+\mathbf{q}, -\mathbf{q}\nu}(T) \right]\end{aligned}$$

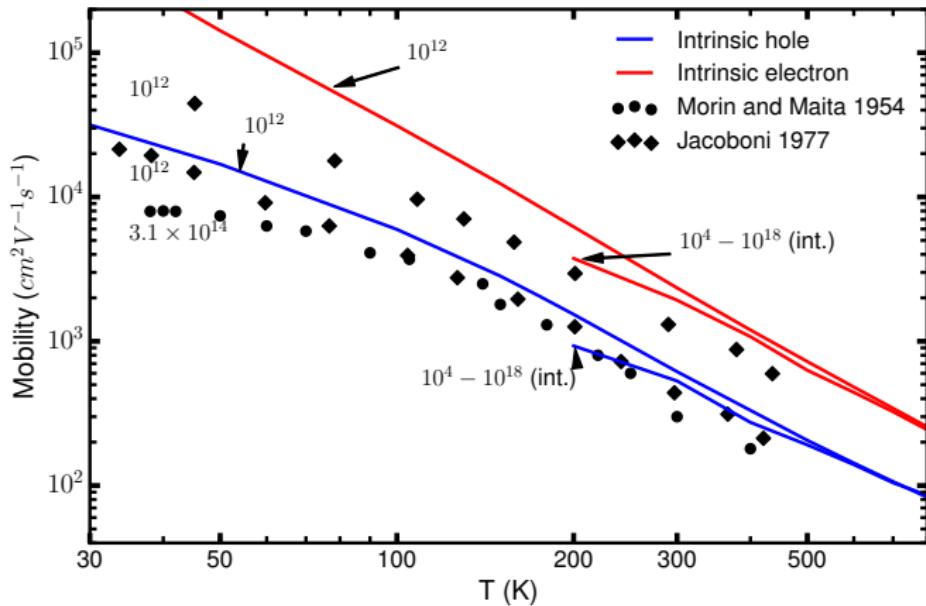


Carrier mobility in Silicon





Intrinsic carrier mobility in Silicon



In preparation (2017)



Acknowledgments





More info



<http://epw.org.uk>



<http://epwforum.uk>



[http://www.qe-
forge.org/gf/download](http://www.qe-forge.org/gf/download)