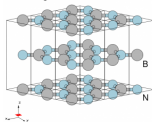


Quantum Espresso Tutorial

Michael Atambo - Technical University of Kenya RASESMA 2023

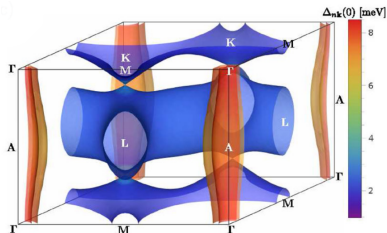


Hexagonal Boron Nitride - hBN



Quantum Espresso Tutorial : Background

- ▶ DFT is a very popular approach to materials modelling at the atomic scale.
- ▶ It strikes a balance between computational cost and accuracy.
- ▶ What can we compute with DFT?
 - ▶ Total energy
 - ▶ Band structure
 - ▶ Forces
 - ▶ Elastic properties
 - ▶ Phonons
 - ▶ Electric polarizability, Raman and infrared Activity
 - ▶ Electron-Phonon coupling, superconducting T_c



Quantum Espresso Tutorial : Background

- ▶ There are limits to DFT course, since its a ground state theory,
 - ▶ Time dependent properties
 - ▶ Localized d & f states
 - ▶ ... ? hint:(**why are we here?**)

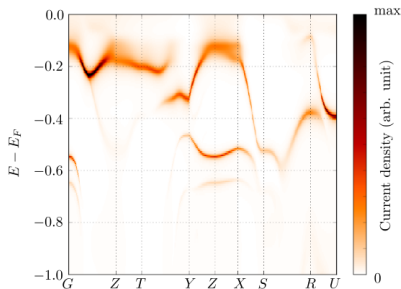


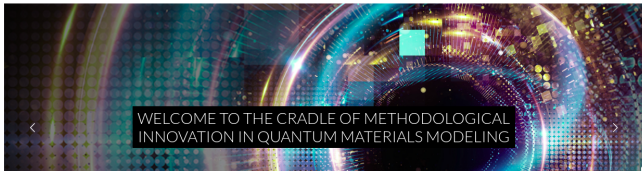
Figure 1: ARPES (Simulated) on FeSb₂ (bnl.gov)

Quantum Espresso Tutorial: the code

- ▶ Quantum Espresso is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the atomic scale
- ▶ It implements KS-DFT,
- ▶ Uses the Plane-wave and Pseudopotential method.



[home](#) [the project](#) [download](#) [users](#) [developers](#) [feats](#) [events](#) [news](#) [support us](#)



Quantum Espresso Tutorial: the distribution

- ▶ The code can be obtained freely from the QE website,
- ▶ Development versions are at: [<https://gitlab.com/QEF/q-e/tags>]



- ▶ The code is FOSS, community contributions.
- ▶ [<https://www.quantum-espresso.org/manifesto/>]

Quantum Espresso Tutorial: Packages

- ▶ Since its a suite of integrated codes, what are they?
 - ▶ pw.x SCF, NSCF, Force minimization,
 - ▶ dos.x Density of states (post processing)
 - ▶ bands.x Band structure
 - ▶ ph.x DFPT
 - ▶ neb.x reaction pathways
 - ▶ atomic.x atomic calculations



Quantum Espresso Tutorial: Packages and plugins

- ▶ Advanced
 - ▶ gipaw
 - ▶ WanT a plugin
 - ▶ Plumed



Quantum Espresso Tutorial: Outputs

- ▶ What can QE do?
 - ▶ Ground state calculations
 - ▶ Structural optimization, MD,
 - ▶ Electrochemistry and special boundary conditions
 - ▶ Response properties (DFPT)
 - ▶ ...
- ▶ Full list: [<https://www.quantum-espresso.org/what-can-qe-do/>]



Quantum Espresso Tutorial: Installing

- ▶ QE provide a standard approach to compiling and installing the code

```
1 ./configure
2 make pw pp
3 make install
```

- ▶ You of course need to have the prerequisites, such as a **compiler**, **FFTW library**, **LAPACK/BLAS**...
- ▶ We wont need to do the installation for the tutorial, this has been done for you.



Quantum Espresso Tutorial: Running a Calculation

- ▶ The code's executable are run from a typical shell environment,

```
1 ~$ pw.x <scf_input.in> scf_output.out
```

- ▶ It requires some preparation ahead of time
 - ▶ choosing pseudopotentials
 - ▶ the structure
 - ▶ input parameters ...



Quantum Espresso Tutorial: Input Documentation

- ▶ The executable in QE will read data from an input file:

```
1 ~$ pw.x < scf_input.in
```

- ▶ The file has a very specific structure:

```
1 &NAMELIST1 ... /  
2 &NAMELIST2 ... /  
3 &NAMELIST3 ... /  
4 &INPUT_CARD1  
5 ...  
6 &INPUT_CARD2  
7 ...  
8 ...
```



Quantum Espresso Tutorial: Input Documentation

- ▶ **NAMELIST** are standard input constructs in F90
- ▶ They allow the specification of a value for an input variable when needed, and defaults otherwise.
- ▶ The variables inside the nameslist can appear in any order.

```
1 &NAMELIST
2   example_variable2=XX,
3   example_variable1=YY
4   ...
```

- ▶ **NAMELISTS** are read in a specific order.
- ▶ **NAMELISTS** that are not required are ignored.



Quantum Espresso Tutorial: Input Documentation

- ▶ **INPUT_CARDS** are specific to Quantum Espresso,
- ▶ They provide a means to specify data that is required, and is inconvenient to put in using NAMELIST format
- ▶ **INPUT_CARDS** Requires that data is in a specific order,

```
1 INPUT_CARDS card_format_specifier  
2 data(1,1) data(1,2) data(1,3)...
```



Quantum Espresso Tutorial: Input Documentation

► **Mandatory NAMELISTS** In Quantum Espresso are these three:

- 1 `&CONTROL` Input variables that control the calculation and the amount of I/O on disk and the screen
- 1 `&SYSTEM` input variables that specify the system under study
- 1 `&ELECTRON` input variables that control the algorithms used to reach self-consistent solution of KS equation for the electrons



Quantum Espresso Tutorial: Input Documentation

► **Mandatory INPUT_CARDS** In Quantum Espresso are three:

- 1 `ATOMIC_SPECIES` name, mass and pseudos for each species present
- 1 `ATOMIC_POSITIONS` type and coordinates of each atom in the unit cell
- 1 `K_POINTS` coordinates and weights of the k-points used for BZ integration



Quantum Espresso Tutorial: Understanding the calculation

- ▶ **Energy Cutoff**: controls the number of basis functions used to expand the wave function
- ▶ **Pseudopotential**: a modification of the ionic potential which allows one to greatly reduce the number of plane waves needed without changing the chemical properties of the atoms.
- ▶ **BZ sampling** we have to sample over the first BZ with a discrete grid.



Quantum Espresso Tutorial: Logging In to Your ICTP Machine,

- ▶ Over **SSH**:

```
1 ~$ ssh -i .ssh/smryambo ictpuser@insXXXX...it
```

- ▶ where XXXX is a number, this should have been sent to you.
- ▶ Over the **browser** chrome/firefox/safari: <https://insXXXXXX.ictp.it>
- ▶ enter the provided password.

Quantum Espresso Tutorial: Running Quantum espresso

- ▶ **Spack** commands will be used to load the environment to have access to the pwscf executable.

```
1 ~$ spack load quantum-espresso
```



Quantum Espresso Tutorial: Silicon Hands On Example

- ▶ We can run the Silicon example,
- ▶ Copy the files to your home directory and work from there:

```
1 ~$ cp -r /media/ictpuser/smr3694/ictptutor/YAMBO_TUTORIALS .
2 ~$ cd YAMBO_TUTORIALS/Silicon/PWSCF
```

- ▶ Follow the Tutorial on the Yambo Wiki: [Click this](https://www.yambo-code.eu/wiki/index.php?title=Silicon)
[<https://www.yambo-code.eu/wiki/index.php?title=Silicon>]

Quantum Espresso Tutorial: Silicon Hands On Example

- ▶ Great,
- ▶ We will have a break before we do the next step,
- ▶ Learning to generate Yambo Inputs.

Generating Yambo Databases from PWSCF

- ▶ We will use **Quantum Espresso** to generate KS-DFT eigenvalues and WFs,
- ▶ It implements **KS-DFT**, among other things.
- ▶ Freely available (FOSS).
- ▶ Yambo uses the results of your PWSCF calculations as the starting point.
- ▶ You need to be able to run SCF+NSCF calculations before getting to G0W0 with yambo.
- ▶ we will follow the bulk h-BN tutorial on the **Yambo Wiki**
[https://www.yambo-code.eu/wiki/index.php/Bulk_material:_h-BN]



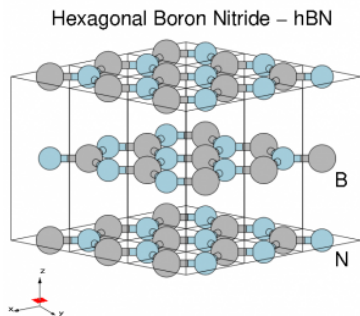
Prerequisites

- ▶ **Quantum Espresso**, it is installed on the ICTP machines.
- ▶ Input files and pseudopotentials : please copy them from the tutor directory.
- ▶ yambo: provides p2y, this is installed on the ICTP machines.



The System

- ▶ Bulk hBN is a HCP structured nitride,
- ▶ Four atoms per cell, 16 electrons in the unit cell,
- ▶ Lattice constants: $a=4.176a.u., c/a=2.582$



Loggin In to Your ICTP Machine,

- ▶ Over **SSH**:

```
1 ~$ ssh -i .ssh/smryambo ictpuser@insXXXX...it
```

- ▶ where XXXX is a number, this should have been sent to you.
- ▶ Over the **browser** chrome/firefox/safari: <https://insXXXXXX.ictp.it>
- ▶ enter the provided password.

Running Quantum espresso

- ▶ **Spack** commands will be used to load the environment to have access to the pwscf executable, and later p2y

```
1 ~$ spack load quantum-espresso  
2 ~$ spack load yambo
```



Retrieving the files:

- ▶ Copy the files to your home directory and work from there:

```
1 ~$ cp -r /media/ictpuser/smr3694/ictptutor/YAMBO_TUTORIALS .  
2 ~$ cd YAMBO_TUTORIALS/hBN/PWSCF
```



DFT Calculations

- ▶ From the **PWSCF** directory, you will observe that you have the following:

```
1 Inputs
2 Pseudos
3 References
4 hBN_2D_nscf.in
5 hBN_2D_scf.in
```

- ▶ These are the required files for this exercise.



Executing the steps

- ▶ Run the SCF and NSCF steps:

```
1 ~$ mpirun -np 2 pw.x < hBN_scf.in > hBN_scf.out
```

- ▶ The SCF generates the ground state $n(r)$, occupations, Fermi level...

```
1 ~$ mpirun -np 2 pw.x < hBN_nscf.in > hBN_nscf.out
```

- ▶ The NSCF calculation will compute the KS eigenvalues and eigenvectors for all the requested **nbnd**



DFT Calculations

- ▶ Some important entries in the **PWSCF** input file:

```
1 wf_collect=.true.  
2 force_symmorphic=.true.  
3 diago_thr_init=5.0e-6,  
4 diago_full_acc=.true.
```

- ▶ These are needed by yambo, see the wiki for more, and the QE documentation for more,



What do we have?

- ▶ **PWSCF** creates a **hBN.save** dir,
- ▶ This is where we will work from for the next task.

```
1 ls hBN.save/  
2 B.pz-vbc.UPF charge-density.dat wfc1.dat wfc11.dat wfc13.dat  
   wfc2.dat wfc4.dat wfc6.dat wfc8.dat  
3 N.pz-vbc.UPF data-file-schema.xml wfc10.dat wfc12.dat wfc14.dat  
   wfc3.dat wfc5.dat wfc7.dat wfc9.dat
```



Conversion to Yambo Format: P2Y

- ▶ Yambo provides the **p2y** executable that can convert **PWSCF** outputs to **YAMBO databases**.
- ▶ It requires no input specific input file. . .
- ▶ How to do that:

```
1 $ cd hBN.save/  
2 $ p2y  
3 ...  
4 ...  
5 <----> == DB3 (PseudoPotential) ... done ==  
6 <----> == P2Y completed ==  
7 $
```



SAVE Directory

- ▶ **p2y** generates a **SAVE** directory

```
1 $ ls
2 B.pz-vbc.UPF  SAVE                data-file-schema.xml  wfc10.dat
   ....
3 $ ls SAVE
4 ns.db1                ns.kb_pp_pwscf_fragment_13  ns.
   kb_pp_pwscf_fragment_6  ns.wf_fragments_11_1  ns.
   wf_fragments_3_1  ...
```

- ▶ What are these files?



yambo -D

- ▶ The **n*** files are netCDF formatted files, you need to use the yambo -D command to check the information they hold:

```
1 $ yambo -D
2
3 [RD./SAVE//ns.db1
4   ]-----
5 Bands                               : 100
6 K-points                            : 14
7 G-vectors                            : 8029 [RL space
8   ]
9 ...
10 ...
11 [RD./SAVE//ns.kb_pp_pwscf
12  ]-----
13 Fragmentation                       : yes
14 - S/N 004626 ----- v
15   .05.01.00 r.21422
16 $
```



What next?

- ▶ The convention for moving from here, is usually to take the **SAVE** directory somewhere else, where you will proceed with the rest of the G0W0 calculations



P2Y advanced usage

- ▶ If you need to know more about how P2Y can be used, refer to this page on the wiki: [Conversion to Yambo Format](#)



Next

- ▶ You now have the necessary inputs for G0W0 and more
- ▶ Before doing the G0W0 runlevel, you need to initialize first
- ▶ Then generate an input
- ▶ This will be done in the first G0W0 exercise



Thanks

- ▶ This is as much as we need for the **PWSCF** tutorial.
- ▶ Thank you!

