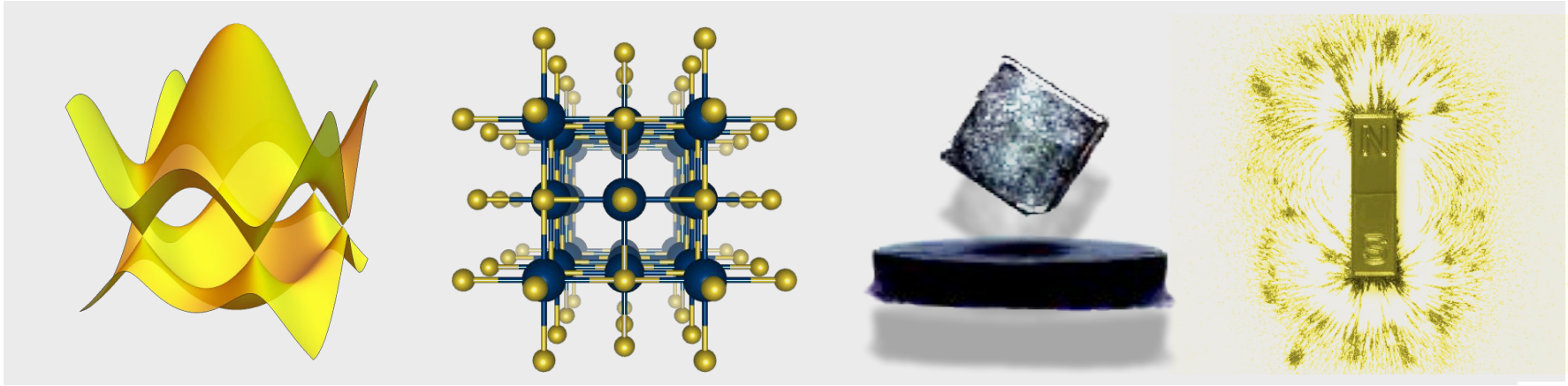
ESCOLA SÉRGIO MASCARENHAS DE FÍSICA DA MATÉRIA CONDENSADA

Hands-On 1 - Computational



SBF

UFPE

Program

Day 1 15/07	Day 2 16/07	Day 3 17/07	Day 4 18/07	Day 5 19/07
Introdução: HPC DFT Unix Quantum Espresso. **Cálculos:** Energia total Relaxação estrutural.	**Estrutura eletrônica:** Estrutura de Bandas Densidade de Estados	**Vibrações cristalinas:** Fónons	**Propriedades magnéticas.** Estado fundamental Temperatura de transição	**Propriedades óticas:** Constante Dielétrica Absorção ótica

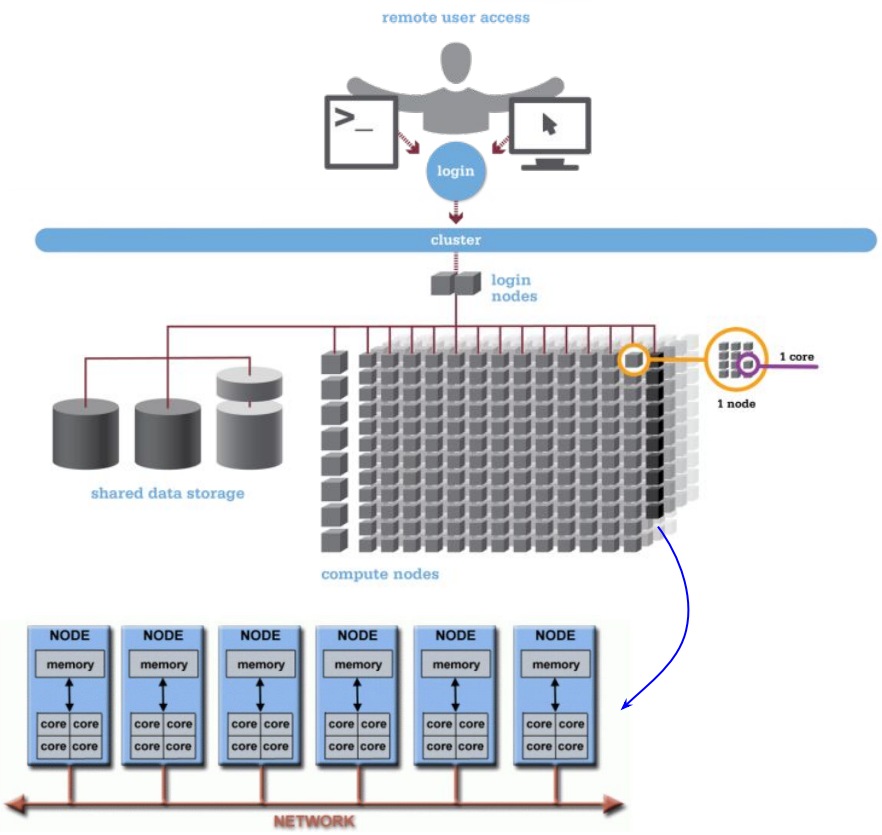
High Performance Computing (HPC)

Santos Dumont Supercomputer

Basic characteristics:

- Original machine in 2015: ~750 CPUs (~1.1 PFlops).
- Sequana expansion in 2019: ~280 CPUs + 100 GPUs (~1.8 Pflops).
- New expansion in early 2025: ~17 PFlops.

Basic HPC architecture



Basic components:

- Login node (a couple).
- Compute nodes (hundreds).
- Shared data storage.
- Communication network.

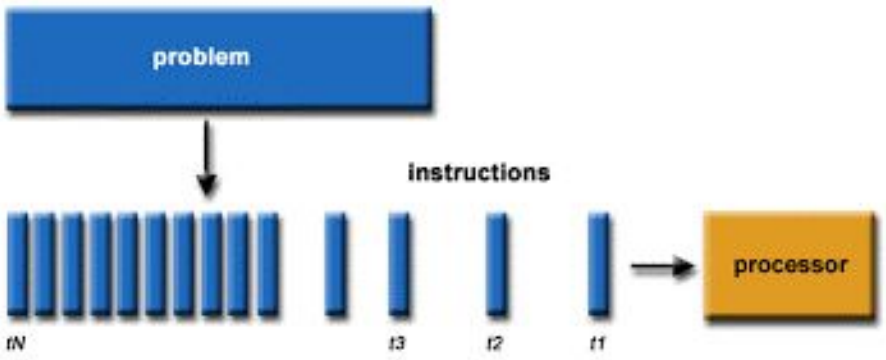
Compute nodes have several CPU cores which share memory. Intra-node communication (i.e. between cores of a node) is **fast**.

Nodes are connected by network. Inter-node communication (i.e. between different nodes) is **much slower**.

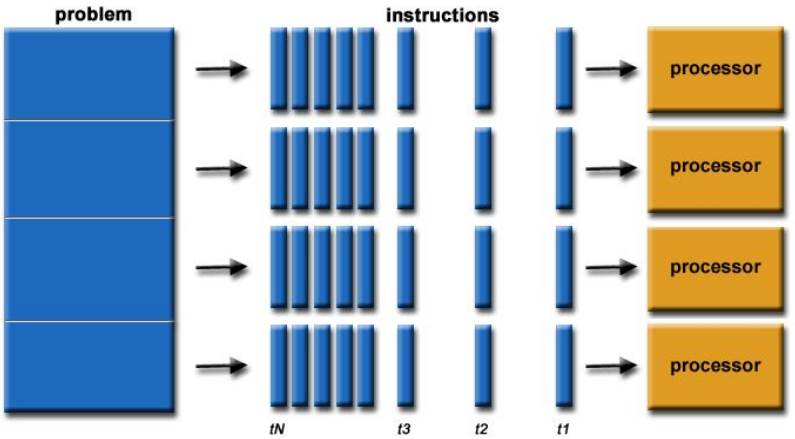
See here for SDumont technical specifications.

Parallelization in a nutshell

Serial programming



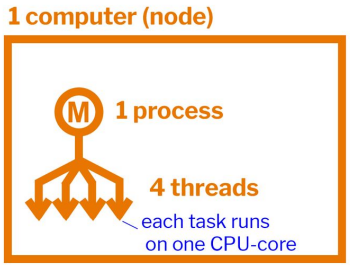
Parallel programming



The calculations we will be doing require parallelization!

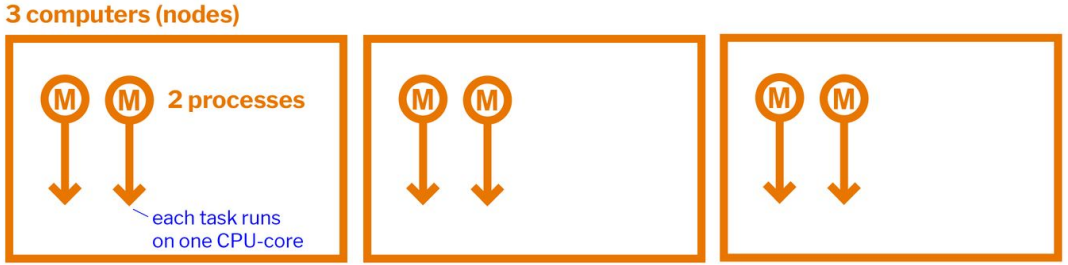
Parallelization in a nutshell

Shared-Memory Parallelism (Multithreading)



Computational tasks associated with the program being run are executed in parallel on different CPUs of the same node. These CPUs share memory, thus communication between tasks (threads) is very fast. Most common method implementing this paradigm is OpenMP.

Distributed-Memory Parallelism (Multiprocessing)

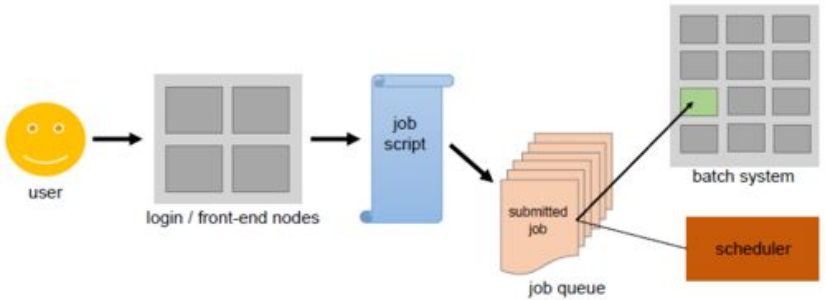


Computational tasks are executed in parallel on different nodes (and/or CPUs). These processes do not share memory. This is good if we want to spread computation over several nodes since communication between nodes is slow. Most common method implementing this paradigm is MPI.

Accelerator Parallelism

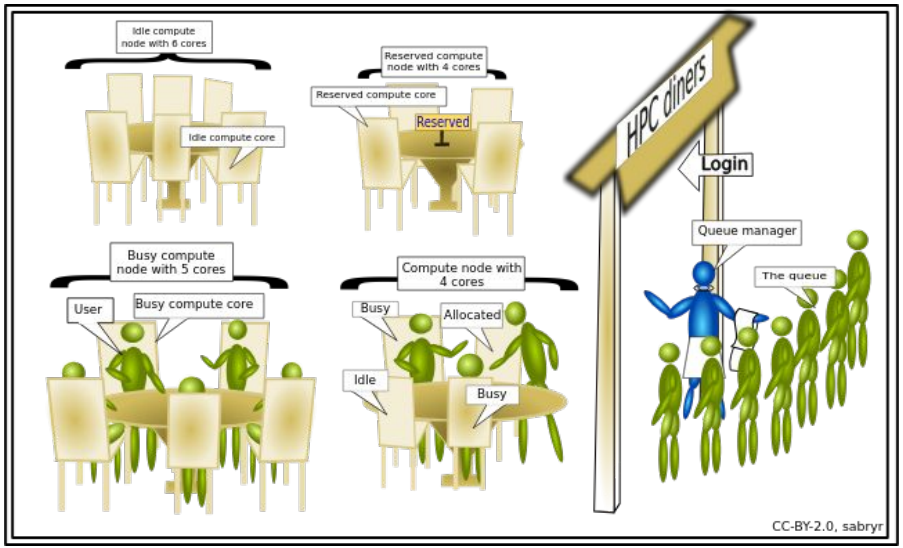
Takes advantage of extremely high parallelization efficiency of GPUs for very particular tasks (eg. matrix multiplications). CPUs are different (they are fast serial general purpose processors).

Basic workflow on HPC systems



HPC systems are multi-user machines. The basic workflow for a user is:

1. Remote access (ssh) from your machine.
2. Work on login node (read, copy, move, etc.).
3. Submit "jobs" to a "scheduler" (i.e. requests for allocation of resources for a time period) .
4. The "scheduler" puts the "job" on the queue waiting for resources to be available.
5. Once job starts you can check output files.
6. Use secure copy (scp) to move data in/from HPC cluster.



Specificities of SDumont

Scheduler: SDumont's scheduler is Slurm (more later). You can find its documentation here.

Queues: In this school we only have access to a couple queues with limited resources – see below.

Partition	Share	Priority	MaxJobs	MaxTRES	MaxTRESPerNode	MaxSubmit	MaxWall	MaxTRESMins
sequana_gpu_dev	1		1	cpu=192,node=4		1	00:20:00	
sequana_cpu_dev	1		1	cpu=192,node=4		1	00:20:00	
nvidia_dev	1		1	cpu=96,node=4		1	00:20:00	
cpu_dev	1		1	cpu=96,node=4		1	00:20:00	

Queue name

Maximum number of jobs running simultaneously

Maximum number of cpus and nodes requested

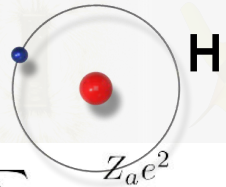
Maximum number of jobs on the queue

Maximum time requested for the job (20 minutes)

Scratch: In SDumont jobs can't be launched from your home folder at /scratch/efmc24/your_username. You have to launch the jobs from your scracth folder /home/efmc24/your_username. The scratch is a location where temporary large files are stored.

(Very!) Basic Notions of DFT

COMPUTATIONAL MODELING OF MATERIALS:
Density Functional Theory



Schrödinger Equation

$H\Psi = E\Psi$

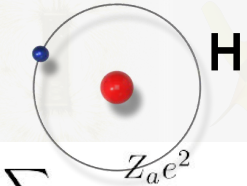
$$\hat{H} = -\sum_a \frac{\hbar^2}{2M_a}\nabla^2_{\vec{R}_a} - \frac{\hbar^2}{2m}\sum_i \nabla^2_{\vec{r}_i} - \sum_a \sum_i \frac{Z_a e^2}{4\pi\epsilon_0|\vec{R}_a - \vec{r}_i|} + \frac{1}{2}\sum_a \sum_{b\neq a} \frac{Z_a Z_b e^2}{4\pi\epsilon_0|\vec{R}_a - \vec{R}_b|} + \frac{1}{2}\sum_i \sum_{j\neq i} \frac{e^2}{4\pi\epsilon_0|\vec{r}_i - \vec{r}_j|}$$

Wave Function

Ψ

Properties of a Quantum System

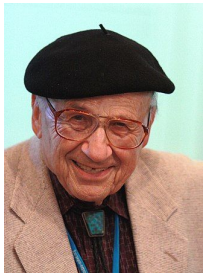
COMPUTATIONAL MODELING OF MATERIALS:
Density Functional Theory



$$\hat{H} = -\sum_a \frac{\hbar^2}{2M_a}\nabla^2_{\vec{R}_a} - \frac{\hbar^2}{2m}\sum_i \nabla^2_{\vec{r}_i} - \sum_a \sum_i \frac{Z_a e^2}{4\pi\epsilon_0|\vec{R}_a - \vec{r}_i|} + \frac{1}{2}\sum_a \sum_{b\neq a} \frac{Z_a Z_b e^2}{4\pi\epsilon_0|\vec{R}_a - \vec{R}_b|} + \frac{1}{2}\sum_i \sum_{j\neq i} \frac{e^2}{4\pi\epsilon_0|\vec{r}_i - \vec{r}_j|}$$

Schrödinger Equation

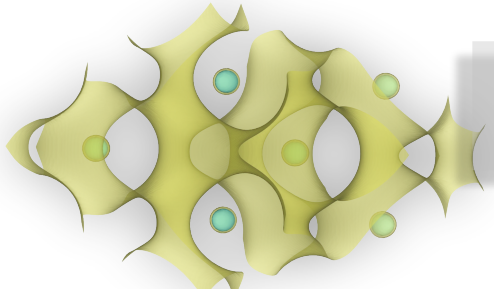
$H\Psi = E\Psi$



Walter Kohn

Electronic Density

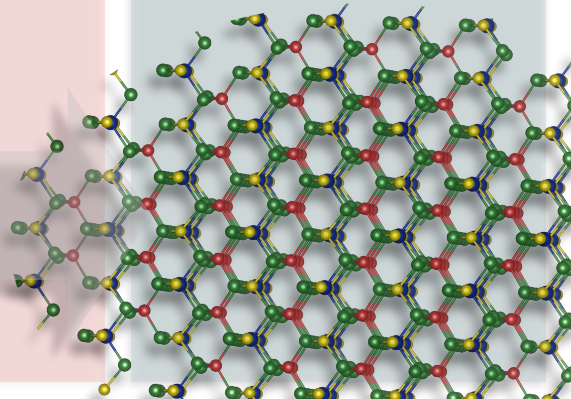
$\rho(r)$



Wave Function

Ψ

Properties of a Quantum System



Chemistry
1998

Walter Kohn

PHYSICAL REVIEW VOLUME 136, NUMBER 3B 9 NOVEMEBR 1964

Inhomogeneous Electron Gas*

P. HOHENBERG†
École Normale Superieure, Paris, France
AND
W. KOHN‡
École Normale Superieure, Paris, France and Faculté des Sciences, Orsay, France
and
University of California at San Diego, La Jolla, California
(Received 18 June 1964)

This paper deals with the ground state of an interacting electron gas in an external potential $v(\mathbf{r})$. It is proved that there exists a universal functional of the density, $F[n(\mathbf{r})]$, independent of $v(\mathbf{r})$, such that the expression $E\equiv\int v(\mathbf{r})n(\mathbf{r})d\mathbf{r}+F[n(\mathbf{r})]$ has as its minimum value the correct ground-state energy associated with $v(\mathbf{r})$. The functional $F[n(\mathbf{r})]$ is then discussed for two situations: (1) $n(\mathbf{r})=n_0+\tilde{n}(\mathbf{r})$, $\tilde{n}/n_0<<1$, and

PHYSICAL REVIEW VOLUME 140, NUMBER 4A 15 NOVEMBER 1965

Self-Consistent Equations Including Exchange and Correlation Effects*

W. KOHN AND L. J. SHAM
University of California, San Diego, La Jolla, California
(Received 21 June 1965)

From a theory of Hohenberg and Kohn, approximation methods for treating an inhomogeneous system of interacting electrons are developed. These methods are exact for systems of slowly varying or high density. For the ground state, they lead to self-consistent equations analogous to the Hartree and Hartree-Fock equations, respectively. In these equations the exchange and correlation portions of the chemical potential of a uniform electron gas appear as additional effective potentials. (The exchange portion of our effective

Hohenberg-Kohn Theorems

Chemistry
1998

Walter Kohn

I. *The external potential v(**r**), and hence the total energy, is a unique functional of the electron density.*

II. *The ground state energy can be obtained variationally: the density that minimizes the total energy is the exact ground state density.*

$$E_v[n] = T[n] + U[n] + \int d^3r\, n(\mathbf{r}) v(\mathbf{r})$$

$$n(\mathbf{r}) \equiv n_s(\mathbf{r}) = \sum_i^N f_i \left|\phi_i(\mathbf{r})\right|^2$$

Kohn-Sham formulation:

Map the interacting many-body system to an independent particle system.

The ground-state density of any interacting system can be reproduced by a non-interacting system.

$$H_{el}\Psi_{el}(\vec{r}_1, \vec{r}_2, ..., \vec{r}_N) = E_{el}\Psi_{el}(\vec{r}_1, \vec{r}_2, ..., \vec{r}_N)$$

$$\rho(\vec{r}) = N \int d\vec{r}_2 \int d\vec{r}_3 \ldots \int d\vec{r}_N \Psi^*(\vec{r}_2, ..., \vec{r}_N)\Psi(\vec{r}_2, ..., \vec{r}_N)$$

HK-I

A densidade eletrônica $\rho_0(\vec{r})$ do estado fundamental de um sistema de muitos elétrons na presença de um potencial externo $\nu(\vec{r}_i)$ determina unicamente este potencial.

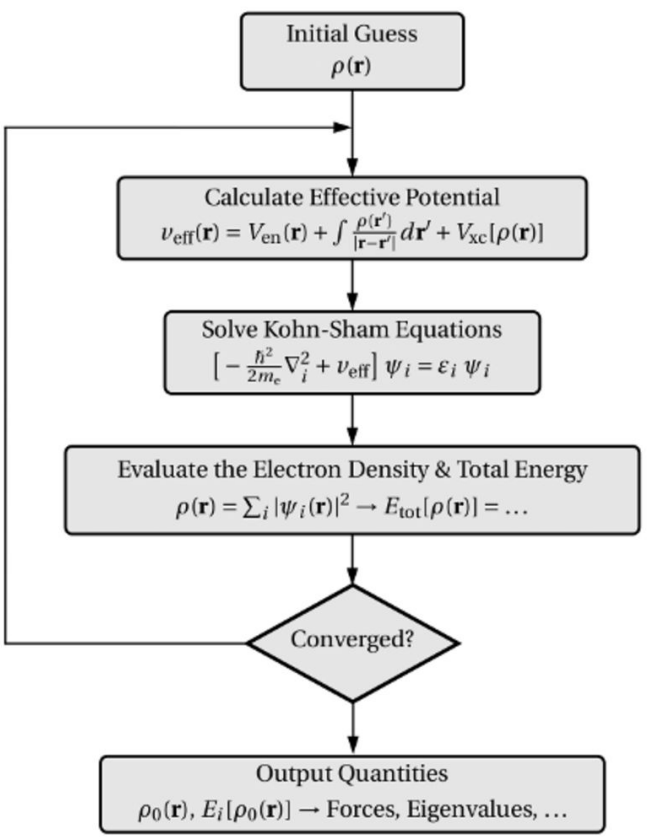
$$E[\rho] = T_e[\rho] + V_{ee}[\rho] + V_{en}[\rho]$$

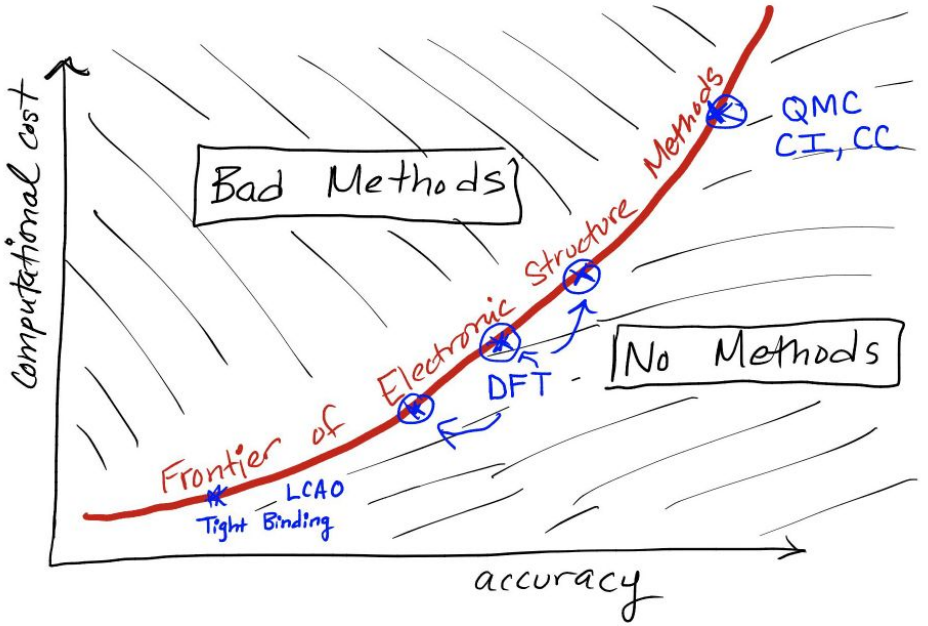
$$\rho(\vec{r}) = \sum_i^N \phi_i^*(\vec{r})\phi_i(\vec{r})$$

HK-II

A densidade que minimiza a energia total do sistema e fornece a energia do estado fundamental E_0 é justamente a densidade ρ_0 do estado fundamental

$$\left[-\frac{1}{2}\nabla^2 + \nu_{eff}(\vec{r})\right]\phi_i(\vec{r}) = \varepsilon_i\phi_i(\vec{r}) \qquad \nu_{eff} = \frac{1}{2}\int \frac{\rho(\vec{r})}{|\vec{r} - \vec{r}'|}d\vec{r} + \nu_{ext}[\rho] + \frac{\delta E_{xc}[\rho]}{\delta\rho}$$





ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Elif Ertekin

REVIEW ARTICLE | INSIGHT
https://doi.org/10.1038/s41563-021-01013-3
nature materials

Check for updates

Electronic-structure methods for materials design

Nicola Marzari[1]✉, Andrea Ferretti[2] and Chris Wolverton[3]

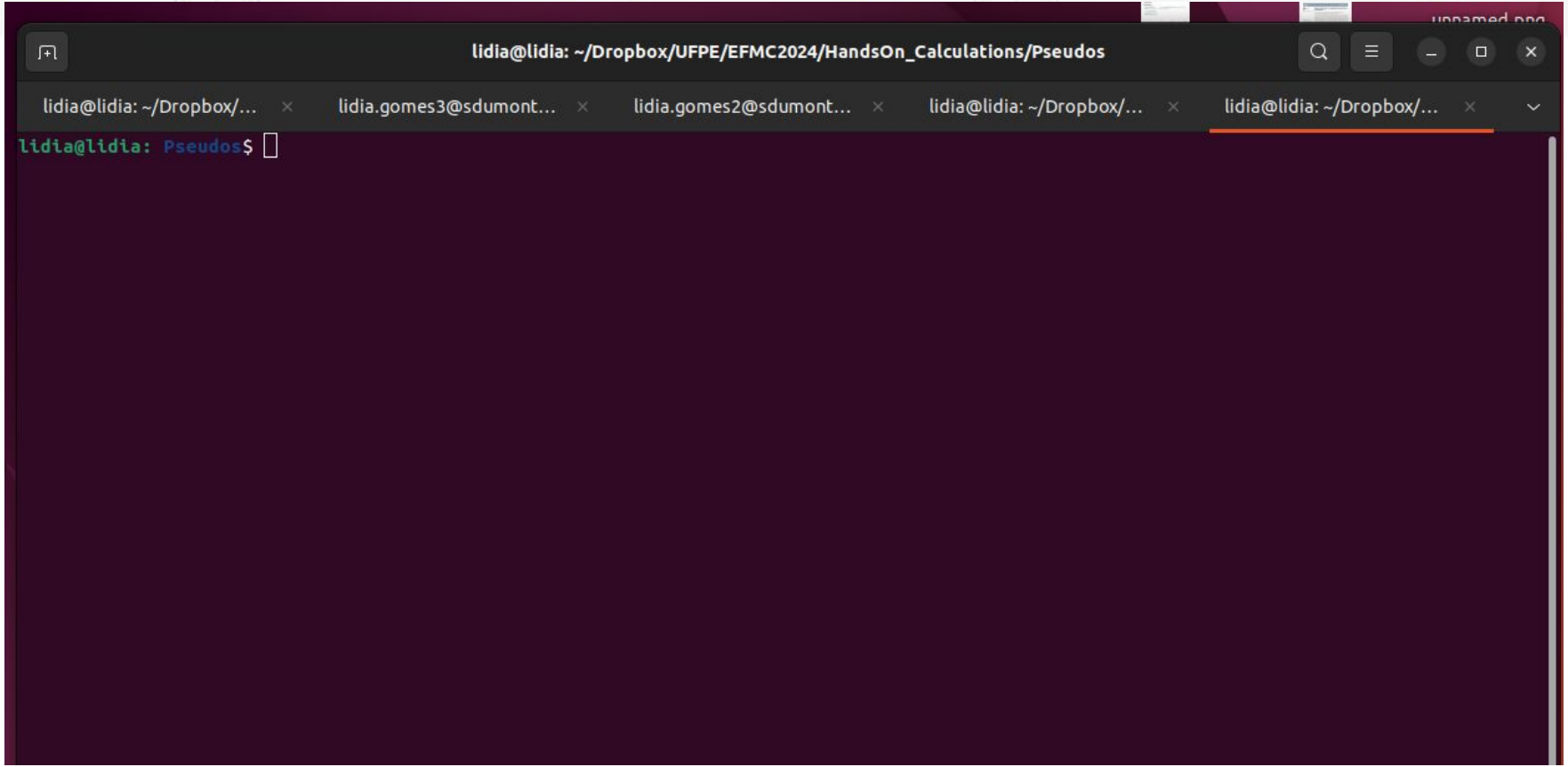
The accuracy and efficiency of electronic-structure methods to understand, predict and design the properties of materials has driven a new paradigm in research. Simulations can greatly accelerate the identification, characterization and optimization of materials, with this acceleration driven by continuous progress in theory, algorithms and hardware, and by adaptation of

Table 1 | An overview of selected materials properties that can be obtained from DFT gr

Materials properties	Models and theories
Atomic and cell geometries at fixed volume or pressure[13,166]	Hellmann–Feynman theorem
Zero-temperature stability[141–143]: formation energies, elastic constants, defects concentrations[167]	Equations of state (Murnaghan, Birch, …), convex hulls, self-consistent chemical potentials
Chemistry and reactivity[84,85], surface science[88]	Potential-energy surfaces, transition-state theory, volcano plots, kinetic Monte Carlo, rate equations, conical intersections, Marcus theory, Franck–Condon principle
Phonon dispersions and thermomechanical properties[18], thermal and electrical transport[19], superconductivity[168]	Linear-response theory, quasi-harmonic approximation, Grüneisen parameters, Boltzmann transport equation, equilibrium/non-equilibrium Green's functions, Allen–Dynes formula, Migdal–Eliashberg equations, superconducting DFT
Dielectric[18], magnetic[123] and topological properties[125], ferroelectrics[92] and multiferroics[169]	Linear-response theory, modern theory of polarization and of magnetization, electric enthalpy, model Hamiltonians, topological invariants
Magnetic phases[54], magnetic anisotropy, spin waves[55], skyrmions[56]	Spin Hamiltonians (Ising, Heisenberg, Dzyaloshinskii–Moriya), paramagnetism as ensemble average
Thermodynamic ensembles[71,77,78]: finite-temperature properties and Helmholtz or Gibbs free energies[81], transport coefficients[108]	Molecular dynamics, Monte Carlo, thermodynamic integration, metadynamics, path-integral molecular dynamics, Green–Kubo relations
Thermodynamic ensembles[88,129,130,170]: composition, chemical potential, partial pressure	Lattice Hamiltonians, Monte Carlo, mean-field approximation, cluster variation method, model entropies, special quasirandom structures
Electrochemistry[86,89], pH[90], operando studies	Grand-canonical simulations for electrons and ions, embedding, double-layer and diffuse-layer models
Macroscopic mechanical properties[109,110] (strength, fracture, and plasticity), soft matter, biomolecules	Multiscale simulations, QM-MM, dislocation dynamics, effective volumes
Microscopies[171]: STM, AFM, TEM	Tersoff–Hamann model, electron scattering

Getting started

Getting started: Terminal



Getting started

➔ Basic Unix Commands:

Unix Command	Description
ls	List directory contents
cp	Copy files
rm	Remove directory entries
which	Locate a program file in the user's path
diff	Find differences between two files
vi	Text editor
mv	Move and rename files
pwd	Print working directory name
cd	Change working directory
ln	Make a file link
mkdir	Make directories
rmdir	Remove directories
alias	Create a command alias

http://www.tech-faq.com/basic-unix-commands.html

Getting started: VPN

➔ On your linux **machine** edit the file **sdumont.conf** with your information:

```
>> sudo nano /etc/vpnc/sdumont.conf
```

```
IPSec gateway 146.134.0.14
IPSec ID sdumont
IPSec secret !$#Sdu#@mon!T321
Xauth username your_username
```

➔ Connecting to the VPN (Virtual Private Network)

```
>> sudo vpnc-connect /etc/vpnc/sdumont.conf --enable-weak-encryption
>> Enter password for login@146.134.0.14:
>> VPNC started in background (pid: ###)...
```

Getting started: SDumont

➔ Log in to SDummont:

```
>> ssh your_username@login.sdumont.lncc.br
>> password:
```

➔ Go to your scratch area:

```
>> cd /scratch/efmc24/your_username
```

➔ Copy files to you user area:

```
>> cp -r /scratch/efmc24/HandsOn_Material/day1 .

>> cd day1

>> ls
```

Getting Started

➔ Copy slides to your local machine:

```
>> scp -r your_username@login.sdumont.lncc.br:/scratch/
   efmc24/HandsOn_Material/day1/*pdf .
```

➔ Installing packages on SDumont:

```
>> module load python/3.9.12

>> pip install pandas

>> pip install matplotlib

>> pip install phonopy==2.11.0

>> pip install DFTtoolbox
```

Getting started

➔ SLURM (Simple Linux Utility for Resource Management) commands:

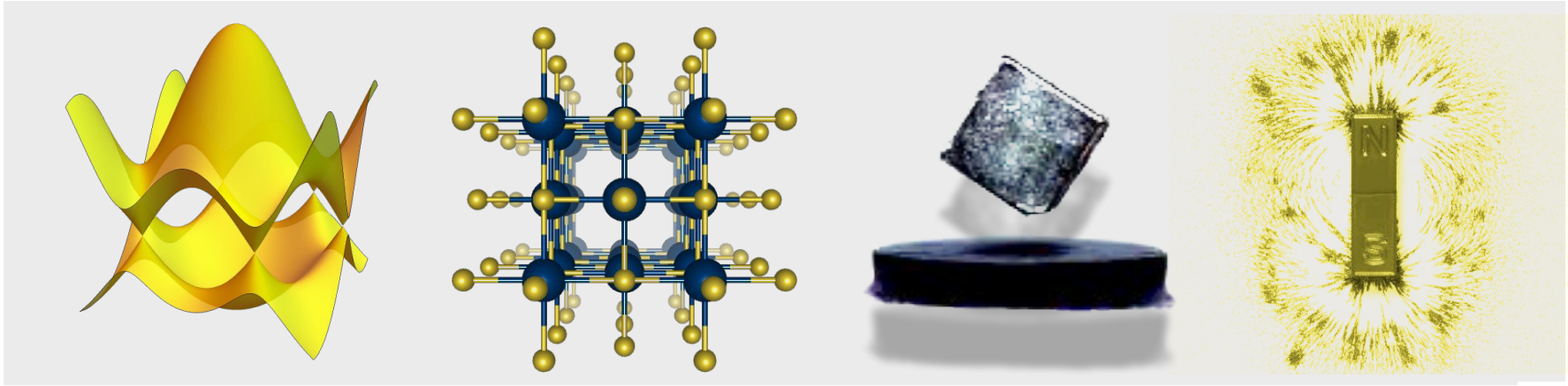
SLURM command	Sample command syntax	Meaning
scancel	scancel <jobid>	Cancel a job. `scancel` can also be used to kill job arrays or job steps.
sacct	sacct -j <jobid>	Check job accounting data. Running `sacct` is most useful for completed jobs.
scontrol	scontrol show job <jobid>	Look at a running job in detail. For more information about the job, add the `-dd` parameter.
scontrol	scontrol hold <jobid>	Pause a job
scontrol	scontrol release <jobid>	Release a held job (allow it to run)
sinfo	sinfo	See node and partition information. Use the `-N` parameter to see information per node.
srun	srun --pty -t 0-0:5:0 -p interactive /bin/bash	Start an interactive session for five minutes in the interactive queue.
sbatch	sbatch <jobscript>	Submit a batch job.
squeue	squeue -u <userid>	View status of your jobs in the queue. Only non-completed jobs will be shown.

https://wiki.rc.hms.harvard.edu/display/O2/Using+Slurm+Basic

ESCOLA SÉRGIO MASCARENHAS DE FÍSICA DA MATÉRIA CONDENSADA

Hands-On 1 - Computational

DFT: Structural Optimization

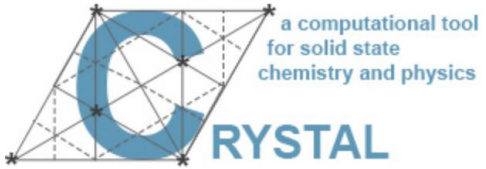


SBF

UFPE

Lesson 0

DFT: Many codes available!



https://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid-state_physics_software

What can QE do?

- ➔ Ground-state calculations.
- ➔ Structural Optimization.
- ➔ Molecular dynamics.
- ➔ Potential energy surfaces.
- ➔ Electrochemistry and special boundary conditions.
- ➔ Response properties (DFPT).
- ➔ Spectroscopic properties.
- ➔ Quantum Transport.
- ➔ and more...

http://www.quantum-espresso.org/

Today's goals

1. Familiarize with QE input/output main files.
   - ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Today's goals

1. Familiarize with QE input/output main files.
   - ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Input data

Input data format:

{ } = optional

[ ] = it depends

| = or

```
&CONTROL
  ...
/

&SYSTEM
  ...
/

&ELECTRONS
  ...
/

[ &IONS
  ...
 / ]

[ &CELL
  ...
 / ]

ATOMIC_SPECIES
 X  Mass_X  PseudoPot_X
 Y  Mass_Y  PseudoPot_Y
 Z  Mass_Z  PseudoPot_Z

ATOMIC_POSITIONS { alat | bohr | crystal | angstrom | crystal_sg }
  X 0.0  0.0  0.0  {if_pos(1) if_pos(2) if_pos(3)}
  Y 0.5  0.0  0.0
  Z 0.0  0.2  0.2

K_POINTS { tpiba | automatic | crystal | gamma | tpiba_b | crystal_b | tpiba_c | crystal_c }
if (gamma)
   nothing to read
if (automatic)
   nk1, nk2, nk3, k1, k2, k3
if (not automatic)
   nks
   xk_x, xk_y, xk_z,  wk

[ CELL_PARAMETERS { alat | bohr | angstrom }
   v1(1) v1(2) v1(3)
   v2(1) v2(2) v2(3)
```

Quantum Espresso: Manual

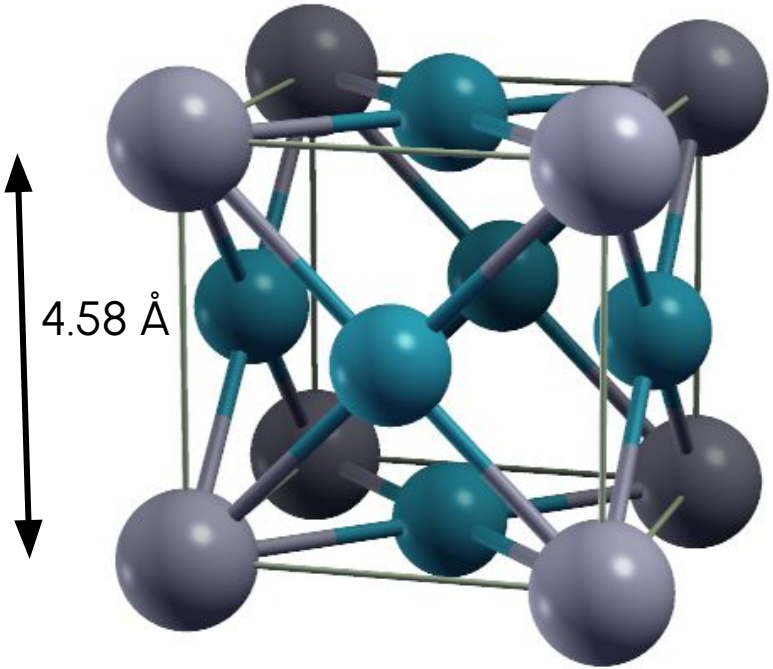
Input File Description

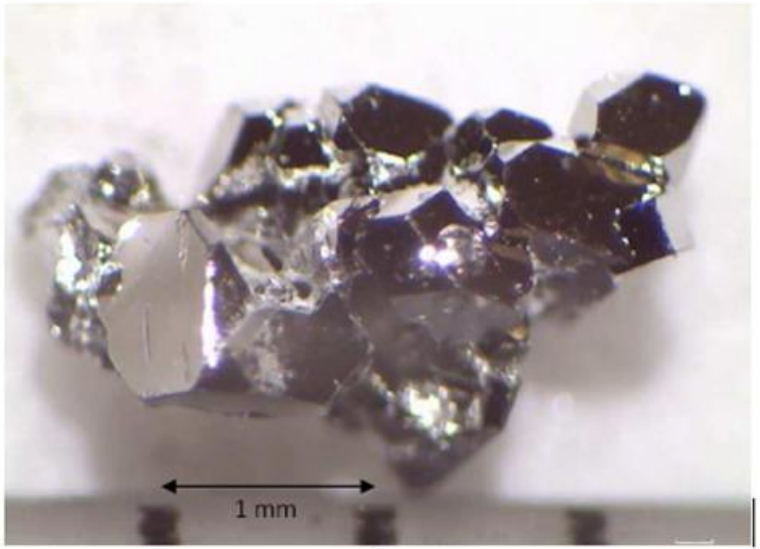
Program: pw.x / PWscf / Quantum Espresso (version: 6.4)

TABLE OF CONTENTS

https://www.quantum-espresso.org/Doc/INPUT_PW.html

YIn_3



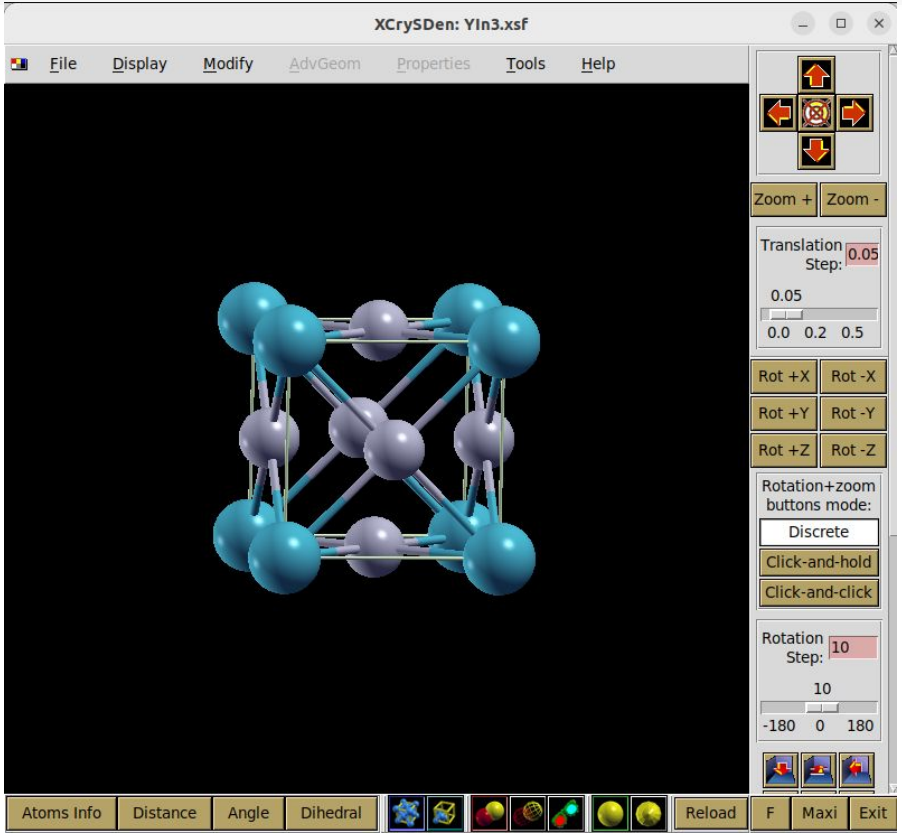


YIn_3

Xcrysden

On the terminal of your machine:

```
>> mkdir efmc24

>> scp -r
your_username@login.sdumont.lncc.br:/
scratch/your_username/day1/ .

>> cd day1/1_scf

>> xcrysden &
```



A simple input file: YIn_3.

```
 &CONTROL
  calculation = 'scf'
  outdir = './'
  prefix = 'yin3'
  pseudo_dir =
'/scratch/efmc2024/HandsOn_Material/Pseudos/'
/
&SYSTEM
  ecutwfc =   90
  ibrav = 0
  nat = 4
  ntyp = 2
/
&ELECTRONS
/
&IONS
/
&CELL
/
```

```
ATOMIC_SPECIES
Y      88.90  Y.pbe-spn-kjpaw_psl.1.0.0.UPF
In    114.82  In.pbe-dn-kjpaw_psl.1.0.0.UPF

ATOMIC_POSITIONS crystal
Y    0.00    0.00    0.00
In   0.50    0.50    0.00
In   0.50    0.00    0.50
In   0.00    0.50    0.50

CELL_PARAMETERS angstrom
  4.6256    0.0000    0.0000
  0.0000    4.6256    0.0000
  0.0000    0.0000    4.6256


K_POINTS automatic
2 2 2 0 0 0
```

A simple input file: YIn_3.

Describes the task to be performed.

Prepended to input/output filenames: prefix.wfc, prefix.rho, etc.

Directory containing pseudopotential and output files.

Crystal structure information.

Atomic specie and corresponding pseudopotential file name.

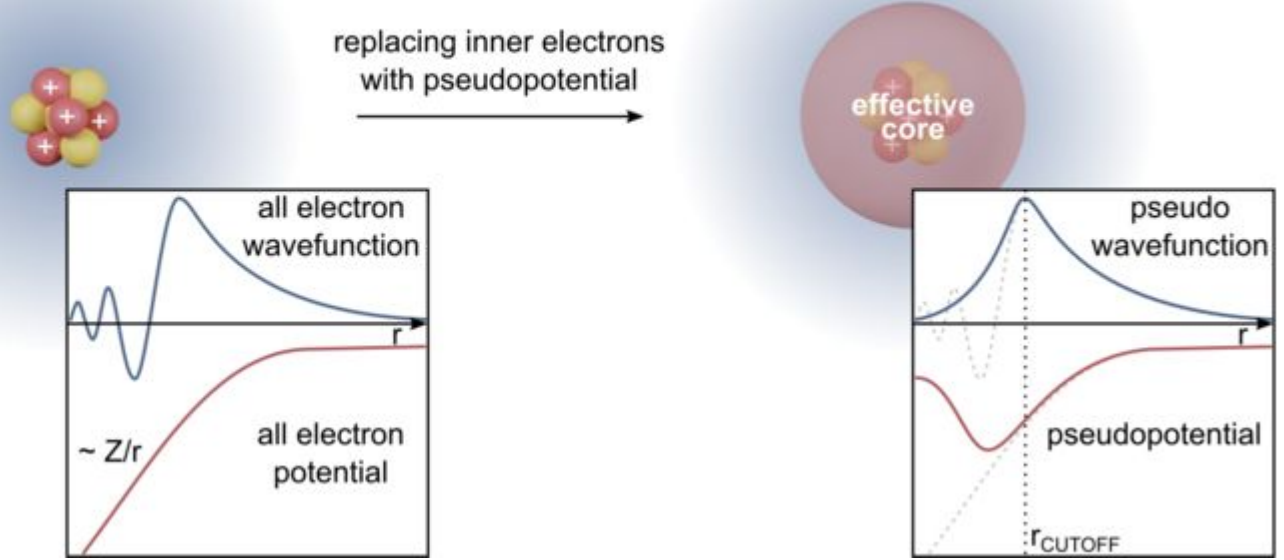
```
&CONTROL
  calculation = 'scf'
  outdir = './'
  prefix = 'ypb3'
  pseudo_dir = '/scratch/efmc2024/HandsOn_Material/Pseudos/'
/
&SYSTEM
  ecutwfc =   90
  ibrav = 0
  nat = 4
  ntyp = 2
/
&ELECTRONS
/
&IONS
/
&CELL
/

ATOMIC_SPECIES
Y      88.90  Y.pbe-spn-kjpaw_psl.1.0.0.UPF
In    114.82  In.pbe-dn-kjpaw_psl.1.0.0.UPF

CELL_PARAMETERS {angstrom}
```

Crystal structure information.

K-points grid.

```
/
&ELECTRONS
/
&IONS
/
&CELL
/

ATOMIC_SPECIES
In      114.82  In.pbe-dn-kjpaw_psl.1.0.0.UPF
Y       88.90585  Y.rel-pbe-spn-kjpaw_psl.1.0.0.UPF

ATOMIC_POSITIONS crystal
Y     0.00    0.00    0.00
In    0.50    0.50    0.00
In    0.50    0.00    0.50
In    0.00    0.50    0.50

CELL_PARAMETERS angstrom
  4.6256     0.0000     0.0000
  0.0000     4.6256     0.0000
  0.0000     0.0000     4.6256


K_POINTS automatic
10 10 10 0 0 0
```

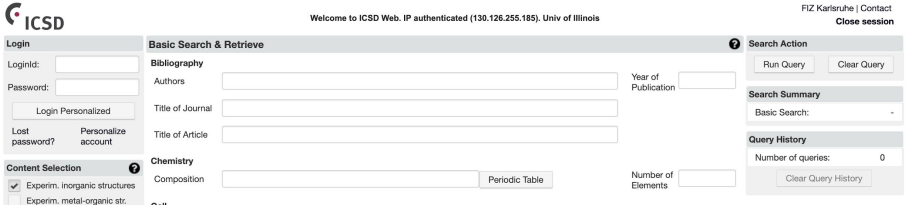
Pseudopotentials



https://commons.wikimedia.org/wiki/File:Pseudopotential.png

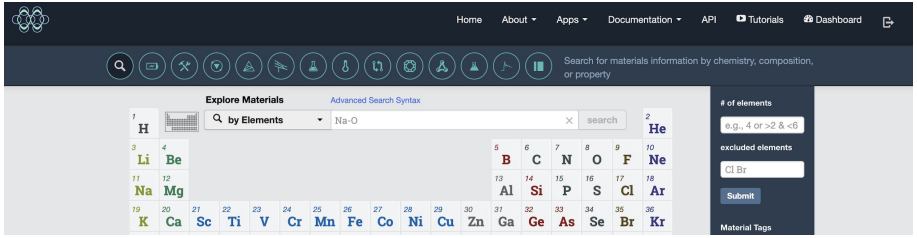
Materials' Structure

Where can we find structural information of materials?

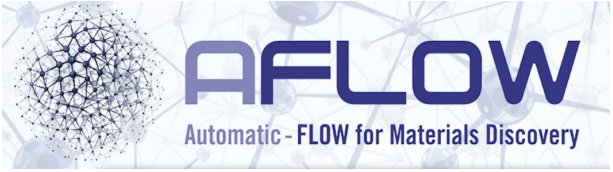


http://www2.fiz-karlsruhe.de/icsd_web.html

http://www.cryst.ehu.es/



https://materialsproject.org/



http://aflowlib.org/

Today's goals

1. Familiarize with QE input/output main files.
   - ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Today's goals

1. Familiarize with QE input/output main files.
   - ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

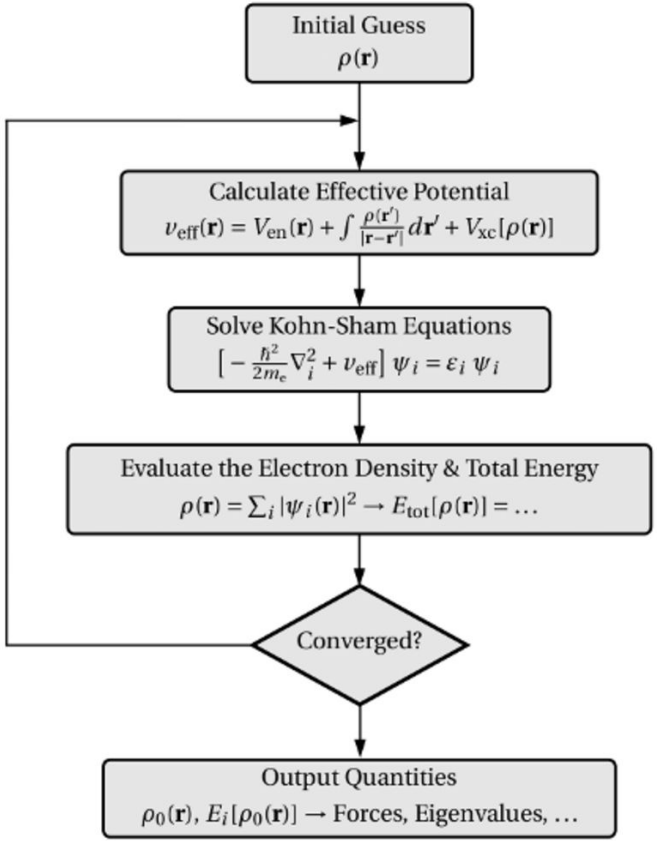
- ➟ k-grid.
- ➟ Energy cutoff.

Ciclo auto-consistente
(scf - self-consistent field)

$$\rho(\vec{r}) = \sum_i^N \phi_i^*(\vec{r})\phi_i(\vec{r})$$

$$\left[-\frac{1}{2}\nabla^2 + \nu_{eff}(\vec{r})\right]\phi_i(\vec{r}) = \varepsilon_i\phi_i(\vec{r})$$

$$\nu_{eff} = \frac{1}{2}\int \frac{\rho(\vec{r})}{|\vec{r}-\vec{r'}|}d\vec{r} + \nu_{ext}[\rho] + \frac{\delta E_{xc}[\rho]}{\delta\rho}$$



Self-Consistent Field (scf) calculation: YIn_3.

This is what you need:

1. Input file

`scf.in`

2. Pseudopotential files

`In.pbe-dn-kjpaw_psl.1.0.0.UPF`
`Y.rel-pbe-spn-kjpaw_psl.1.0.0.UPF`

3. Submission script

`sub.srm`

```
 &CONTROL
  calculation = 'scf'
  outdir = './'
  prefix = 'yin3'
  pseudo_dir =
'/scratch/efmc2024/HandsOn_Material/Pseudos/'
/
&SYSTEM
  ecutwfc =   90
  ibrav = 0
  nat = 4
  ntyp = 2
/
&ELECTRONS
/
&IONS
/
&CELL
/

ATOMIC_SPECIES
Y       88.90  Y.pbe-spn-kjpaw_psl.1.0.0.UPF
In     114.82  In.pbe-dn-kjpaw_psl.1.0.0.UPF
```

Pseudopotentials

https://pseudopotentials.quantum-espresso.org/legacy_tables

QUANTUMESPRESSO

HOME PROJECT DOWNLOAD RESOURCES PSEUDOPOTENTIALS CONTACTS NEWS & EVENTS

PSEUDOPOTENTIALS

- More about pseudopotentials
- SSSP on Materials Cloud
- Pseudo DoJo
- ONCV Potentials
- PSlibrary table
- Original QE PP table
- Hartwigesen-Goedecker-Hutter PP table
- Old FHI PP table

PSLIBRARY

Ready-to-use pseudopotentials from PSlibrary (*recommended*). For other ready-to-use tables, follow the links of the menu at the left. For more info, see here.

1 H																	2 He
3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
19 K	20 Ca		21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
37 Rb	38 Sr		39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
55 Cs	56 Ba	57-70 *	71 Lu	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
87 Fr	88 Ra	89-102 **	103 Lr	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt									

* Lanthanoids	57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb
** Actinoids	89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No

Elements for which at least a pseudopotential is available will appear in red in the periodic table. Click on the element entry and follow the link to access the pseudopotentials and a minimal description of their characteristics.

Pseudopotentials

http://www.quantum-espresso.org/pseudopotentials

http://www.pseudo-dojo.org/

https://www.materialscloud.org/discover/sssp/table/efficiency#0

http://theossrv1.epfl.ch/Main/Pseudopotentials

scf calculation: YIn_3.

1.
```
>> cd 1_scf
```

2. Run!
```
>> sbatch sub.srm
```

3. Check if the calculation finished properly:

```
>> tail scf.out

     PWSCF        :      0.06s CPU      0.28s WALL

   This run was terminated on:  18:54:28   7May2019

=------------------------------------------------------=
   JOB DONE.
=--------------------------------------=
```

4. Take your time to have a look at which files were produced after the calculation have finished.

5. Let's get used to the main output file.

scf calculation: YIn_3.

6. Checking some important information:

a. Convergence on energy:

```
>> grep "total energy" scf.out
     total energy              =   -1515.04651972 Ry
     total energy              =   -1515.04673690 Ry
     total energy              =   -1515.04677887 Ry
     total energy              =   -1515.04682267 Ry
!    total energy              =   -1515.04682360 Ry
```

b. Final total energy:

```
>> grep ! scf.out

!    total energy              =   -1515.04682360 Ry
```

scf calculation: YIn_3.

c. scf accuracy:

```
>> grep "accuracy" scf.out

     estimated scf accuracy    <       0.06489762 Ry
     estimated scf accuracy    <       0.00246557 Ry
     estimated scf accuracy    <       0.00005617 Ry
     estimated scf accuracy    <       0.00000053 Ry
```

7. Inside the **yin3.save** directory:

   a. Charge density file: charge-density.dat

   b. Wave functions file: wfc1.dat and wfc2.dat (one for each k-point).

   c. Info from PAW pseudopotentials: paw.txt.

Today's goals

1. Familiarize with QE input/output main files.
   - ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Today's goals

1. Familiarize with QE input/output main files.
   ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

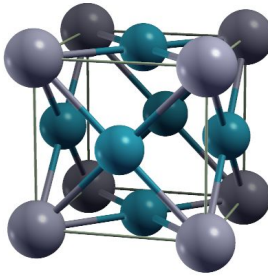
- ➟ k-grid.
- ➟ Energy cutoff.

Equilibrium lattice parameter: YIn_3

```
 &CONTROL
  calculation = 'relax'
  etot_conv_thr =   6.0000000000d-05
  forc_conv_thr =   1.0000000000d-04
  outdir = './'
  prefix = 'yin3'
  pseudo_dir =
'/scratch/efmc24/HandsOn_Material/Pseudos/'
  tprnfor = .true.
  tstress = .true.
  verbosity = 'high'
/
&SYSTEM
  degauss =   0.02
  ecutwfc =   100
  ibrav = 0
  celldm(1)= 17.489400
  nat = 4
  ntyp = 2
  occupations = 'smearing'
  smearing = 'gauss'
/
&ELECTRONS
  conv_thr =   1.d-06
  electron_maxstep = 80
```

(em a.u.)!

```
 &IONS
/
&CELL
/

ATOMIC_SPECIES
Y      88.90  Y.pbe-spn-kjpaw_psl.1.0.0.UPF
In    114.82  In.pbe-dn-kjpaw_psl.1.0.0.UPF

ATOMIC_POSITIONS (crystal)
Y     0.0000000000       0.0000000000       0.0000000000
In    0.5000000000       0.5000000000       0.0000000000
In    0.5000000000       0.0000000000       0.5000000000
In    0.0000000000       0.5000000000       0.5000000000

CELL_PARAMETERS alat
      0.5000000000       0.0000000000       0.0000000000
      0.0000000000       0.5000000000       0.0000000000
      0.0000000000       0.0000000000       0.5000000000

K_POINTS automatic
10 10 10 0 0 0
```



Equilibrium lattice parameter

1. Go to the working directory

```
>> cd 2.1_lattice
```

2. Change the crystallographic constant value at the relax.in file:

```
celldm(1) = 19.0
```

```
>> sbatch sub.srm
```

3. What will happen to the energy? It will increase or decrease? Why?

```
>> grep ! relax.out

!    total energy              =      -1514.9933 Ry
```

Equilibrium lattice parameter

1. Go to the working directory

```
>> cd 2.1_lattice
```

2. Change the crystallographic constant value at the relax.in file:

```
celldm(1) = 19.0
```

```
>> sbatch sub.srm
```

3. What will happen to the energy? It will increase or decrease? Why?

```
>> grep ! out

!    total energy              =      -1514.9933 Ry
```

Energy increases!

X

```
celldm(1) = 17.4894 Ry  -1515.0468 Ry
```

Equilibrium lattice parameter

1. Do the same for many different values of celldm(1):

```
>> cd 2.2_lattice
```

```
celldm(1) = 0.80*celldm(1)0
celldm(1) = 0.85*celldm(1)0
celldm(1) = 0.90*celldm(1)0
celldm(1) = …
celldm(1) = 1.25*celldm(1)0
celldm(1) = 1.30*celldm(1)0
```

Changing the lattice vector to values varying from the original $celldm(1)_0$ =10.2.

2. Alternatively, just use the script:

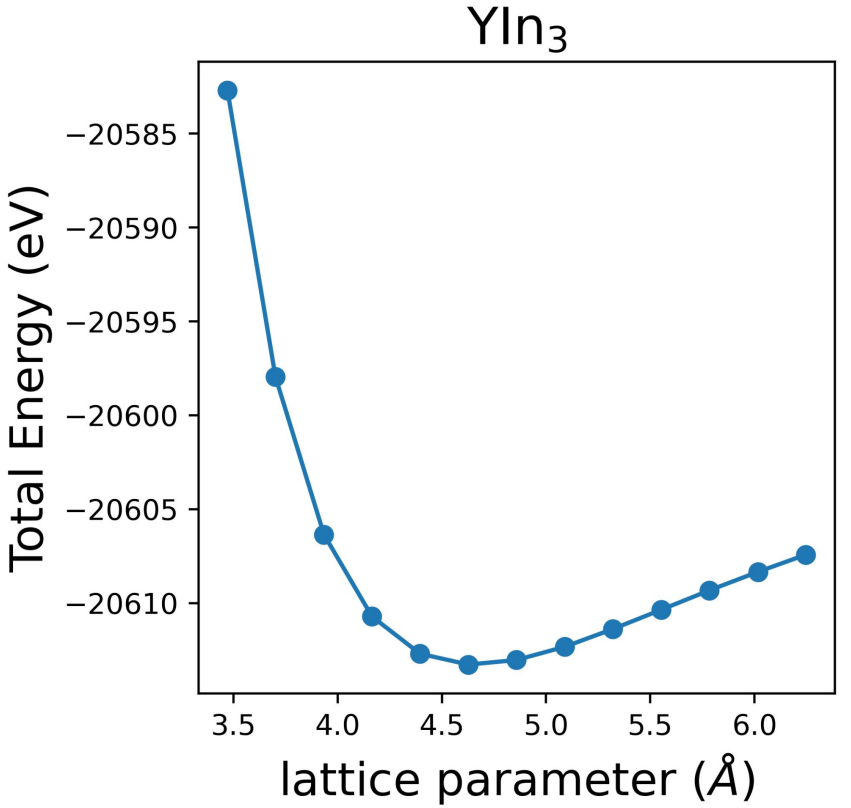
Ps: scripts are always helpful ;)

```
>> python qe_run_etotXcelldm.py
```

Equilibrium lattice parameter

3. Plot the total energies as a function of celldm(1):

```
>> python3 get_etot_X_celldm.py
>> python3 plot_two_columns.py
```



Today's goals

1. Familiarize with QE input/output main files.
   ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Today's goals

1. Familiarize with QE input/output main files.
   ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Full structural optimization

1. Go to the working directory

```
>> cd 3_vc-relax
```

2. Let's use one of the previous 'distorted' celldm(1) values:

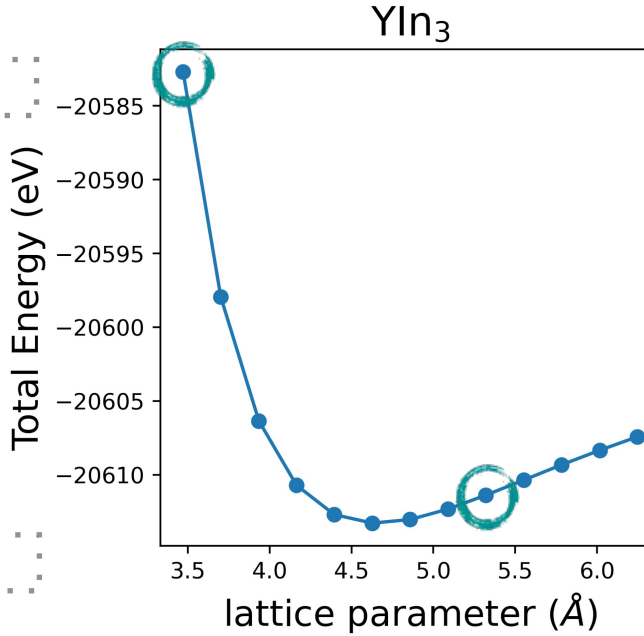
```
>> cp -r ../2.2_lattice/celldm_0.75 .

>> cp -r ../2.2_lattice/celldm_1.15 .
```

3. Edit the input file:

```
calculation = 'vc-relax'
```

4. Run!

```
>> sbatch celldm_0.75/sub.srm

>> sbatch celldm_1.15/sub.srm
```



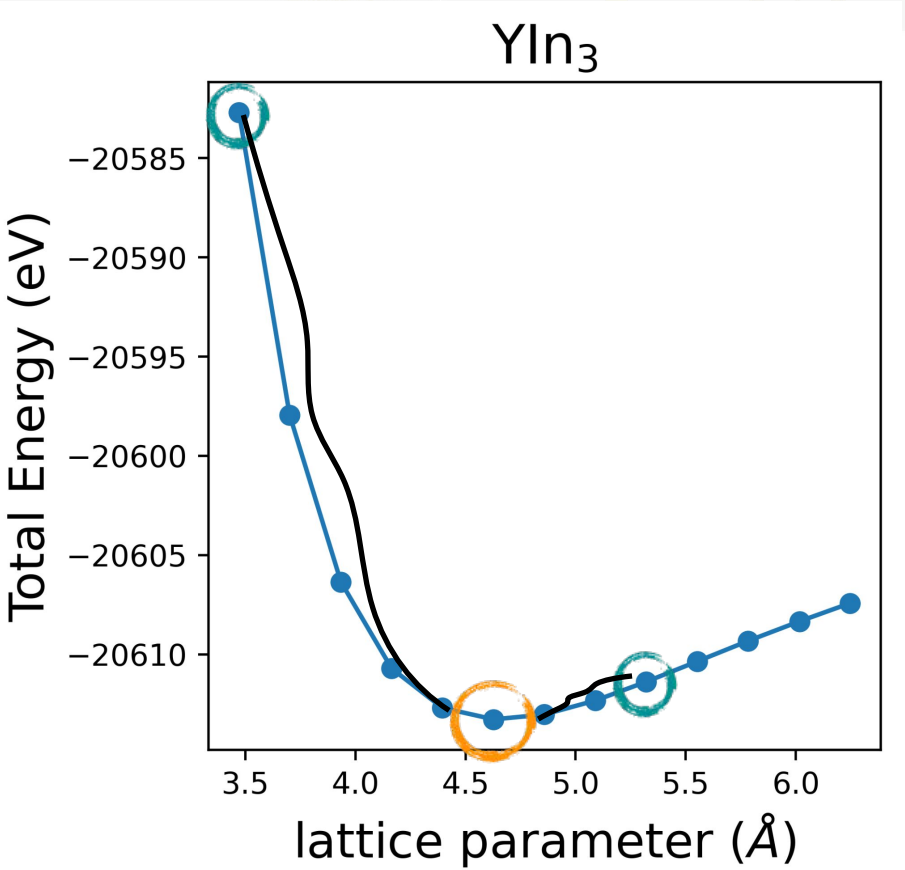
Full structural optimization

6. Compare the final lattice parameters:

```
>> grep -A10 Begin celldm_0.75/relax.out


>> grep -A12 Begin celldm_1.15/relax.out
```

What is happening?

Full structural optimization



Today's goals

1. Familiarize with QE input/output main files.
   ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Today's goals

1. Familiarize with QE input/output main files.
   ↳ Define the crystal structure structure of YIn_3.
2. Perform a simple self-consistent field (scf) calculation.
3. Compute lattice parameters of YIn_3.
4. Perform structural optimization of YIn_3.

Extra (DIY!) ➔ Basic (but fundamental!) total energy convergence tests:

- ➟ k-grid.
- ➟ Energy cutoff.

Convergence test: k-points and ecutwfc

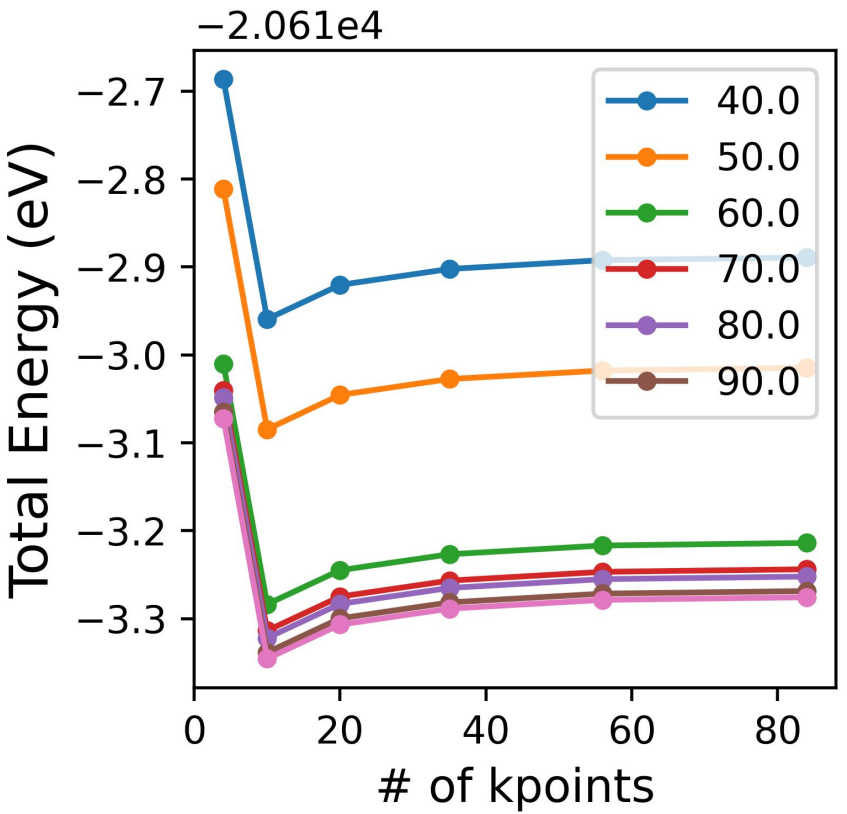
1. Go to the working directory

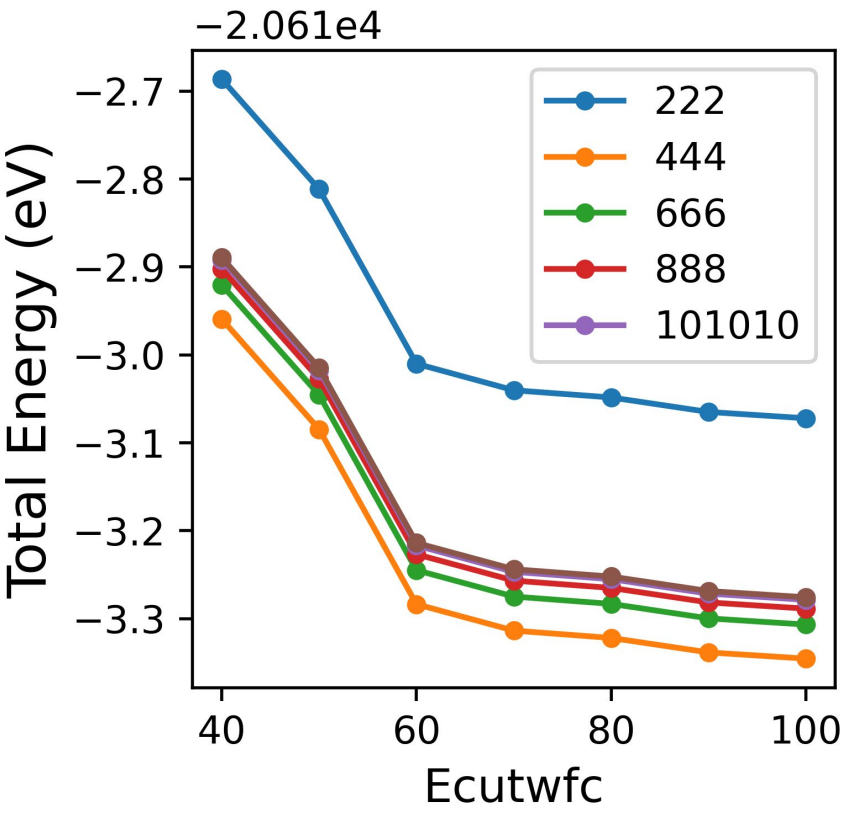
```
>> cd day1/reference/0_conv_kgrid-ecut

>> python qe_conv-ENCUT+KPOINTS.py
```

2. Collect and plot data:

```
>> python qe_get_etot_kpoints_ecut.py

>> python qe_plot_etot_kpoints_ecut.py
```

Convergence test: k-points and ecutwfc

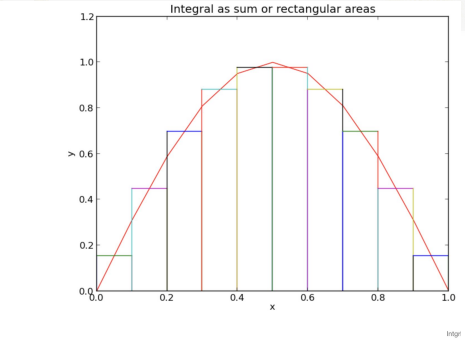




Convergence test 1: k-points

Choice of the k-point mesh:

- Periodic systems: integrals in real space over the (infinitely extended) system are replaced by integrals over the (finite) first Brillouin zone in reciprocal space, by virtue of Bloch's theorem.



Intergal approximated as 10 rectangular areas of width dx=0.1

- Such integrals are performed by summing the function values of the integrand (for instance: the charge density) at a finite number of points in the Brillouin zone, called the k-point mesh.

- Choosing a sufficiently dense mesh of integration points is crucial for the convergence of the results, and is therefore one of the major objectives when performing convergence tests.

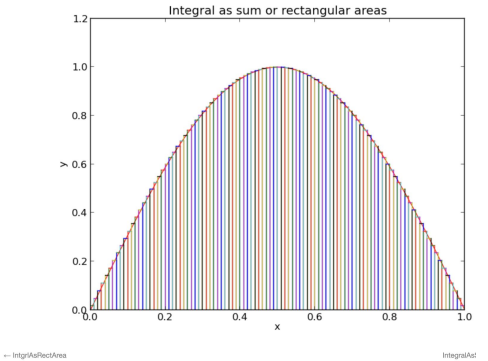


Fig. 2 The integral approximated as the sum of 100 rectangular area elements

Convergence test 1: k-points

```
&control
    prefix='silicon',
    calculation = 'scf',
    pseudo_dir = '/APS-workshop/handson/pseudos'
    outdir='./'
 /

…

K_POINTS automatic
  k1 k2 k3 1 1 1
```

Variables k1, k2 and k3.

Convergence test 1: k-points

You have to do the same k-grid convergence test with respect to EVERY property that you want to calculate:

- Total energy
- DOS
- Magnetic moment
- Others

For semiconductors, usually converged with a coarse k-mesh (4x4x4, 6x6x6).

Usually converged with a finer k-mesh (10x10x10 and above).

Convergence test 2: Energy Cutoff

```
 &control
    prefix='silicon',
    calculation = 'scf',
    pseudo_dir = '/APS-workshop/handson/pseudos'
    outdir='./'
 /

&system
    ibrav=  0
    celldm(1) = 10.2
    nat=  2
    ntyp= 1
    ecutwfc = 40
 /

 &electrons
 /

CELL_PARAMETERS alat
  -0.500000   0.000000   0.500000
   0.000000   0.500000   0.500000
  -0.500000   0.500000   0.000000

ATOMIC_SPECIES
 Si  28.086  Si.pbe-n-kjpaw_psl.0.1.UPF
ATOMIC_POSITIONS alat
 Si 0.00 0.00 0.00
 Si 0.25 0.25 0.25

K_POINTS automatic
  2 2 2 1 1 1
```

ecutwfc	REAL
Status:	REQUIRED

kinetic energy cutoff (Ry) for wavefunctions

Convergence test 2: Energy Cutoff

1. Input file `conv.in`:

Variable ECUTWFC.

```
&control
    prefix='silicon',
    calculation = 'scf',
    pseudo_dir = '/APS-workshop/handson/pseudos'
    outdir='./'
/

&system
    ibrav=  0
    celldm(1) = 10.2
    nat=  2
    ntyp= 1
    ecutwfc = ECUTWFC
/

&electrons
/

CELL_PARAMETERS alat
  -0.500000   0.000000   0.500000
   0.000000   0.500000   0.500000
  -0.500000   0.500000   0.000000

ATOMIC_SPECIES
 Si  28.086  Si.pbe-n-kjpaw_psl.0.1.UPF
```

More Convergence tests

As in the determination of the ground state of any system we need to check that the results are converged with respect to several technical parameters:

- The completeness of the plane wave expansion (parameters ecutwfc and ecutrho in &system);
- The completeness of the Brillouin zone sampling (K_POINTS card);
- The possibility of magnetism in the system;
- The convergence threshold in structural optimization (etot_conv_thr, forc_conv_thr)

See you tomorrow!

Extra

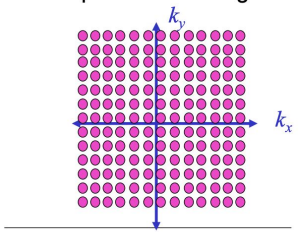
Plane Waves & Periodic Systems

- For a periodic system:

$$\psi_{\mathbf{k}}(\mathbf{r}) = \frac{1}{\Omega}\sum_{\mathbf{G}} c_{\mathbf{k},\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$

where **G** = reciprocal lattice vector

- The plane waves that appear in this expansion can be represented as a grid in k-space:



- Only true for periodic systems that grid is discrete.
- In principle, still need infinite number of plane waves.

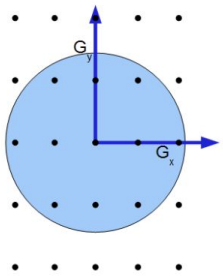
Cut-off Energy

Each of the Fourier basis functions $e^{i\mathbf{G}.\mathbf{r}}$ represents a plane-wave travelling in space, perpendicular to the vector **G**.

There are an infinite number of allowed **G**, but the coefficients c_{Gk} become smaller and smaller as $|\mathbf{G}|^2$ becomes larger and larger.

We define a cut-off energy $E^{cut} = \frac{\hbar^2}{2m}|\mathbf{G}|^2$ and only include plane-waves with energies less than this cut-off.

Cut-off Energy



Getting Started

➔ Editing packages:

```
>> nano /prj/efmc24/lidia.gomes3/.local/lib/python3.9/site-packages/DFTtoolbox/postproc.py

    np.int → int

    np.float → float
```