Intro to DFT & QE

Emine Kucukbenli
Feat. Sara Laporte, Matteo Ferri, Yusuf Shaidu
SISSA, Italy

Before you start...

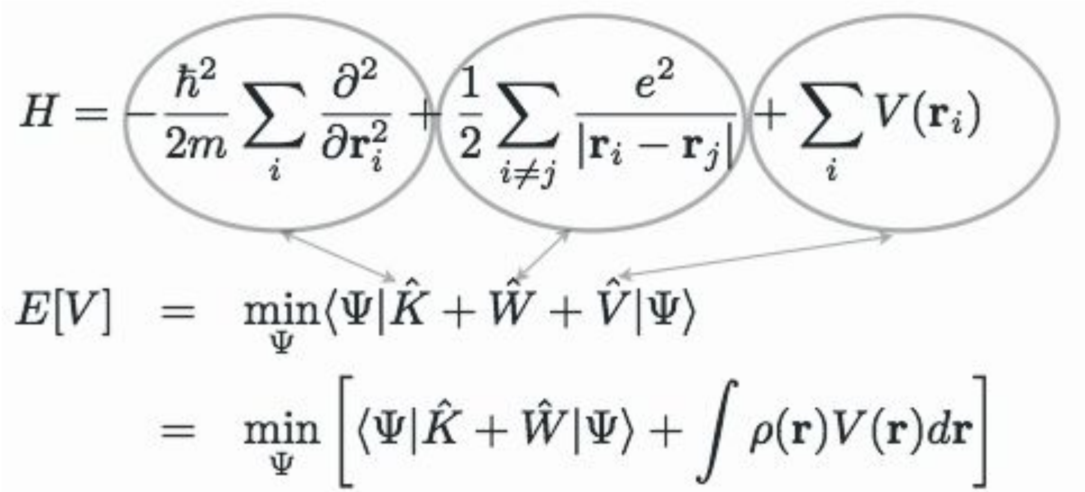
- The tutorial is designed as an assisted self-study, not a lecture.
  Start when you are ready.
- First, you will see a quick reminder of how DFT solves a many body problem using non-interacting electron picture. For more detailed notes see: http://stefano.baroni.me/presentations/2018/2018_Penn_State_DFT.pdf
- Then move on to hands-on section: 1) Silicon examples: How to calculate total energy and perform geometry optimization with Quantum Espresso? What are the convergence parameters that might be relevant during crystal structure prediction calculations? 2) A selection of standard exercises from previous QE-schools. Focus on the system type relevant to you (metals, magnetic, 2D etc.)
- When in doubt ask any one of us or www.quantum-espresso.org

Electronic Many Body Hamiltonian

Born Oppenheimer approximation: Ionic positions become parameters to the electronic Hamiltonian (and often gets omitted in equations)

$$\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial \mathbf{r}^2}+V(\mathbf{r},\mathbf{R})\right)\Psi(\mathbf{r}|\mathbf{R})=E(\mathbf{R})\Psi(\mathbf{r}|\mathbf{R})$$

The Electronic Hamiltoanian and the ground state energy can be written as:



$$H=-\frac{\hbar^2}{2m}\sum_i \frac{\partial^2}{\partial \mathbf{r}_i^2}+\frac{1}{2}\sum_{i\neq j}\frac{e^2}{|\mathbf{r}_i-\mathbf{r}_j|}+\sum_i V(\mathbf{r}_i)$$

$$\begin{aligned} E[V] &= \min_{\Psi}\langle\Psi|\hat{K}+\hat{W}+\hat{V}|\Psi\rangle \\ &= \min_{\Psi}\left[\langle\Psi|\hat{K}+\hat{W}|\Psi\rangle+\int\rho(\mathbf{r})V(\mathbf{r})d\mathbf{r}\right] \end{aligned}$$

http://stefano.baroni.me/presentations/2018/2018_Penn_State_DFT.pdf

DFT as Legendre Transform

The famous Hohenberg-Kohn theorem demonstrates that, there is a one-to-one mapping between the nuclei potential and electron density of the ground state.

$$V(\mathbf{r}) \rightleftarrows \rho(\mathbf{r})$$

A different way of expressing this is that ground state energy, E[V], could be a convex/concave functional of nuclei potential.

The significance of convexity is that for functions that are convex everywhere, derivative is unique and can be used to represent the functional information equally well.
This is the basis of the Legendre transform.

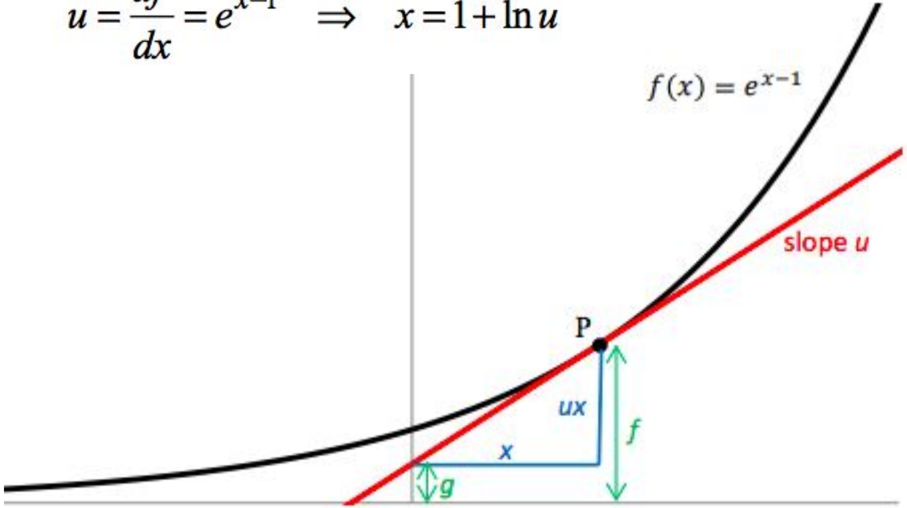
Legendre Transform

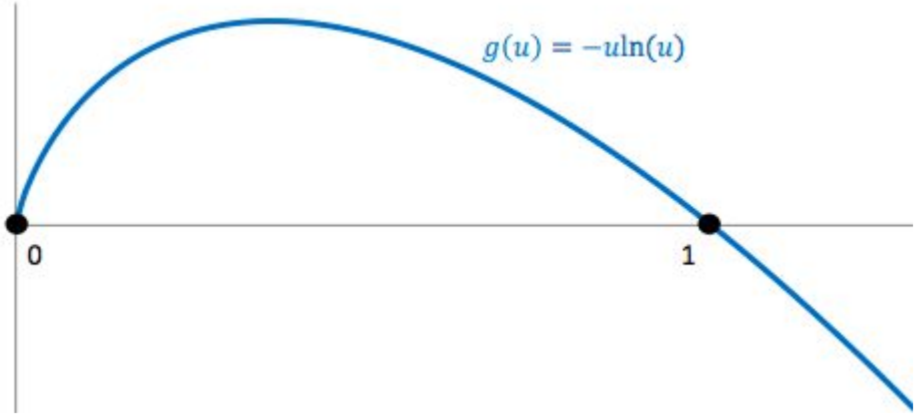
A geometrical demonstration of Legendre conjugates x and u.

In our case, V and density (rho) can be considered as Legendre conjugates.

$$u = \frac{df}{dx} = e^{x-1} \quad \Rightarrow \quad x = 1 + \ln u$$

$$g(u) \equiv f(x) - xu = u - (1 + \ln u)u = -u \ln u$$





https://www.aapt.org/docdirectory/meetingpresentations/sm14/mungan-poster.pdf

Legendre Transform of E[V]

This is a very simplified picture of real convex analysis of DFT, yet useful for understanding that E[V] and its convex-conjugate* F[rho] hold the same information

$$\rho(\mathbf{r}) = \frac{\delta E}{\delta V(\mathbf{r})} \quad \text{and} \quad E[V] = \min_{\rho}\left[F[\rho] + \int V(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}\right]$$

“The universal functional”

F[rho] turns out to be easier to work with and approximate:

$$F[\rho] = T_0[\rho] + \frac{e^2}{2}\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + E_{xc}[\rho]$$

Still a many-body problem that we do not know how to solve exactly for realistic systems, but at least the problem is represented in the 3-D space, not in N-particle coordinates

*More details here: https://doi.org/10.1016/j.theochem.2006.05.012

Kohn-Sham DFT recipe

Using the HK theory so far, find the stationary point for E taking derivative wrt density:

$$\frac{\delta T_0}{\delta \rho(\mathbf{r})} + \overbrace{e^2 \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + \frac{\delta E_{xc}}{\delta \rho(\mathbf{r})} + V(\mathbf{r})}^{v_{KS}[\rho](\mathbf{r})} = \mu$$

KS recipe suggests to map this problem on to an auxiliary system of non-interacting electrons that we can actually solve. Doing that allows to replace T[rho] with the trivial kinetic energy expectation value of one of a non-interacting system, and all e-e interaction and the error in T is buried inside the effective potential functional, i.e. KS potential, in particular the “exchange correlation functional”. The problem then becomes a single-particle one:

$$\left(-\frac{\hbar^2}{2m} \nabla^2 + v_{KS}[\rho](\mathbf{r}) \right) \psi_v(\mathbf{r}) = \epsilon_v \psi_v(\mathbf{r})$$

KS recipe: Solve the independent particle problem...

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + v_{KS}[\rho](\mathbf{r})\right)\psi_v(\mathbf{r}) = \epsilon_v\psi_v(\mathbf{r})$$

..in order to find the KS orbitals that minimize the energy of the many body system

$$\begin{aligned}E = \min\nolimits_\psi \Bigg(& -\frac{\hbar^2}{2m}\sum\nolimits_v \langle\psi_v|\nabla^2|\psi_v\rangle + \\ &\frac{e^2}{2}\int \frac{\rho(r)\rho(r')}{|r-r'|}dr\, dr' + \boxed{E_{xc}[\rho]} + \int V_I\,\rho(r)dr\Bigg)\end{aligned}$$

LDA, PBE etc. approximations

We started with a many body problem in N-coordinates;
Arrived at a tractable single particle problem in 3D.
If we knew Exc[rho], we would have had the exact solution.

- Question to check your DFT understanding: How do the sum of KS eigenvalues relate to the total energy of the many body system?

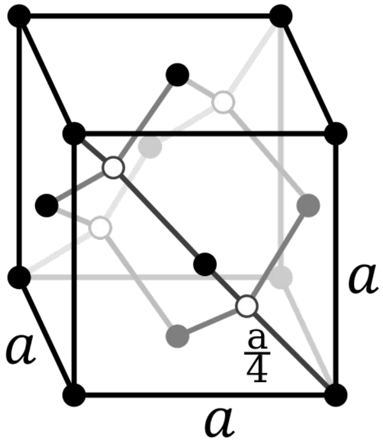
QE input must include information on DFT ingredients:

- Positions of nuclei, shape of the unit cell
- Choice of e-e interaction approximation, i.e. choice of XC functional
- Expansion of wavefunctions using a well converged basis set (PW)
- Expansion of density using a well converged basis set (PW)
- It is too costly, unnecessary and error inducing to try to simulate all electrons -> model the core electrons as a potential that the explicit electrons see (Pseudopotentials) - already requires an approximation for the e-e interaction
- Although the Brillouin zone is continuous, we need to discretize it for computational reasons.

QE input also requires computation and software related information:

- “calculation” What kind of calculation are you running? (total energy? Molecular dynamics? Geometry optimization? etc)
- “prefix” the name that user gives to the calculation - for bookkeeping reasons
- “pseudo_dir” where to find the pseudopotential file
- “outdir” where to place the intermediary files of the calculation such as wavefunctions. (Needs fast access & can be large)

Example: Silicon - Input preparation



Download the Silicon and pseudo directories from the following link https://drive.google.com/open?id=1n4m3Sfg65XA7UnwsXoG77QFywTpor8p6

```
>cd Silicon
```

(I’m using > as the prompt of your terminal)

Examine the silicon input file in your tutorial directory “ si.scf.in “

Examine the input description using the user manual here: https://www.quantum-espresso.org/Doc/INPUT_PW.html

Note that there are many ways to describe the atomic positions and the unit cell.

Note that there are two ways of providing the discretization of Brillouin zone: i) you can provide the list of point coordinates ii) you can use a uniform mesh and provide the size of the mesh and the offset.

Example: Silicon - Visualization and run

Now let us see if the input file we have prepared really is an FCC Silicon in dimaond structure. Run the following in command line:

```
>xcrysden --pwi si.scf.in &
```

Make sure the interatomic distance and lattice parameter is what you expect to find. If you are satisfied with the structure, now you can run the simulation. Add the address of the executable `pw.x` to your path as below and run:

```
>export PATH=$PATH:/scratch/qe-6.3/bin:/scratch/VESTA-x86_64
>export LD LIBRARY PATH=/scratch/VESTA-x86_64
>pw.x -i si.scf.in > si.scf.out
```

Now examine the contents of the output file si.scf.out and the directory that you defined with “outdir” key.

Example: Silicon - Output analysis

Observe the iterative solution of the independent particle problem, as the optimization yields lower and lower energies at each step:

```
> grep -e ’total energy’ -e estimated si.scf.out
 total energy = -15.79103344 Ry
estimated scf accuracy < 0.06376674 Ry
total energy = -15.79409289 Ry
estimated scf accuracy < 0.00230109 Ry
total energy = -15.79447822 Ry
estimated scf accuracy < 0.00006291 Ry
total energy = -15.79449510 Ry
estimated scf accuracy < 0.00000448 Ry
! total energy = -15.79449593 Ry
estimated scf accuracy < 0.00000005 Ry
The total energy is the sum of the following terms:
```

Example: Silicon - What else is there in the output?

Take a look at the output file and try to locate the following information

Cell volume, number of electrons in the cell, KS orbitals, number of bands,
number of planewaves used, symmetry operations that were found,
number of iterations that were needed to find a converged solution,
convergence criteria, Hartree-energy, XC approximation used, CPU time spent

If you are curious about some output messages, feel free to ask.

Example: Silicon - Quality of convergence

- Is this a converged basis set for the wavefunctions ?
  - Try running calculation with `ecutwfc=`16, 20, 24, 28, 32 Ry - observe the effect on number of plane waves used, total energy, (force) and stress.
  - Plot the dependence of total energy as a function of cutoff by preparing a suitable data file. How is the trend? Is it as you expected?

    ```
    >plot 'si.etot_vs_ecut' using 1:2 with lines
    ```
- Is this a converged discretization for the BZ?
  - Try running calculations with using a uniform mesh of K points and observe the behavior of total energy and stress as a function of the mesh size. Try: nk=2,4,6 divisions in each direction i.e. the K_POINTS card of your input should look like, in the case of nk=4
  - `K_POINTS automatic`
  - `4 4 4 1 1 1`
  - Change the offset (the last integer) from 0 to 1 and observe the change in k points and the energy. Can you guess why the change of offset might be useful?

Example: Silicon - Structural properties

We now know how to calculate total energy for a given structure. So we are now ready to calculate a mechanical property:
“Equation of State” is a functional form that describes the relationships of state variables such as temperature, pressure etc with each other for a given material. Here we will fit Birch–Murnaghan equation of state functional form to the “lattice parameter vs energy” data we will gather for Silicon. As a result of the fit, we will derive the equilibrium volume, bulk modulus and pressure derivative of bulk modulus for Silicon.

Gather 10 “lattice parameter vs energy” data points. Make sure that your modifications to the lattice only introduces uniform strain. Use ev.x code for the fitting. Observe the effect of convergence on the EoS parameters.

Crystal Structure Prediction related

In CSP methods that use evolutionary algorithm, high energy structures can be visited as initial configurations. Relaxation to the nearest local minimum might take many steps and can become the bottleneck of computations.

One way to speed up this process is to first perform a relaxation with a low quality convergence but computationally cheap setup, and then relax the cell again with tighter convergence parameters. Now we will study geometry optimization for atomic positions and lattice parameters and test the relevant threshold parameters.

Example: Silicon - Relaxing atomic positions

So far we have always placed Silicon atoms in their equilibrium positions such that the total force on them is zero by symmetry. Let us now modify the position of one atom out of the high symmetry point and perform relaxation to observe whether it will come back to zero force position.

Examine the input `si.rx.in` , visualize the cell, and run the calculation when satisfied. Note that the symmetry is reduced now, so we will expect more non-equivalent k points to appear in the same mesh size compared to symmetric case.

When the run is completed, visualize the relaxation using

```
> xcrysden --pwo si.rx.out
```

Example: Silicon - Relaxing atomic positions

```
> grep ! si.rx.out
!    total energy              =     -15.75361982 Ry
!    total energy              =     -15.78916339 Ry
!    total energy              =     -15.80569272 Ry
!    total energy              =     -15.80712183 Ry
!    total energy              =     -15.80724959 Ry
!    total energy              =     -15.80729247 Ry
!    total energy              =     -15.80731093 Ry
```

Energy goes down as relaxation continues. Similarly take a look at the evolution of the total force and atomic positions. Change the force and energy convergence thresholds to make reach more or less strictly relaxed configurations. Observe the total number of BFGS steps required at each case.

Example: Silicon - Variable-cell relaxation

Examine the input `si.vc.in` , visualize the cell, and run the calculation when satisfied.

When the run is completed, visualize the relaxation using

```
> xcrysden --pwo si.rx.out
```

In the output, grep “P=” to see the pressure at each relaxation step.

Note that positions and cell are optimized simultaneously and in order to do so, a fictitious mass is assigned to the cell (“`wmass`” in `&cell` namelist). If you observe an imbalance in how quickly forces and pressure reach to the target values, you can explore different values for wmass until desired behavior is reached.

Selected exercises of various systems

See http://www.fisica.uniud.it/~giannozz/QE-Tutorial/

Download “Using PWscf: basics” > “write-up” and “exercises” to reach exercises on

- Metals and smearing the Fermi-Dirac distribution - Aluminium
- Magnetic systems and different magnetic ordering - Fe (bcc and hcp)
- 2D structures: Graphene, graphane and graphene oxide
- Case of ultrasoft pseudopotentials

Since we do not have time to go through all in detail, choose a system that is of relevance to you and follow the write-up. If you need another class of systems, take a look at list of examples that come with the package: https://github.com/QEF/q-e/tree/master/PW/examples

Final remarks

- If you are a regular user of the code, remember to subscribe to user and developer mailing lists.
- Always use the latest stable version when you can.
- The code is open source, so feel free to look into it, modify it, contribute to it.
- There is a large user base, you are probably not the first one with a beginner’s issue, search the internet for resources.
- More and more automation tools are appearing (ASE, AiiDA, etc) take advantage of them.
- If you want to re-use your data in machine learning (for example with PANNA) keep the .xml files rather than the text-based output files.