



Computational Solid State Physics, part II

- **Instructor:** Lilia Boeri
- **Email:** lilia.boeri@uniroma1.it
- **Webpage:** <https://lboeri.wordpress.com>



SAPIENZA
UNIVERSITÀ DI ROMA

Lab2: Setting up a DFT calculation

➤ Introduction

- **Recap of DFT basics:** Kohn-Sham Equations
- Kohn Sham Equations using **plane waves** and **pseudopotentials**

➤ Part I: Set up and run a SCF calculation for FCC Neon

- **Quantum Espresso Input & Output files** for scf calculations.
- **Setting up and visualizing the crystal structure.**
- Convergence parameters.
- Visualizing the charge density in real space.

➤ Part II: Self-consistent calculation for cubic and hexagonal silicon.

What you will need for **today's** lecture

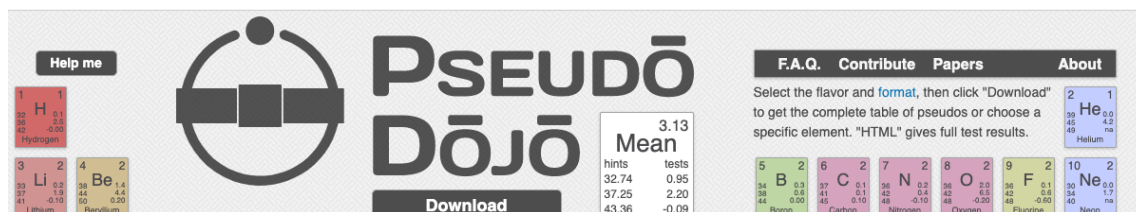
- For the practical part of our lecture we will use the code **Quantum Espresso (QE)**, which can be **downloaded** from: <https://www.quantum-espresso.org/>.



- For the Crystal structure visualization we will use VESTA, which can be **downloaded** from: <https://jp-minerals.org/vesta/en/download.html>



- Pseudopotential files can be downloaded from: <http://www.pseudo-dojo.org/>





Recap of DFT basics:

Density Functional Theory (**DFT**)

1) Density is the main variable: the ground-state properties of a system of interacting electrons are uniquely determined by its **ground-state charge Density**.

$$n^0(\mathbf{r}) = N \int d^3r_2 \dots d^3r_N |\Psi_{el}^0(\mathbf{r}, \mathbf{r}_2, \dots, \mathbf{r}_N)|^2$$

P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Phys. Rev. **136**, B864 (1964).

2) The ground-state charge density can be obtained solving a system of **self-consistent equations** for an **auxiliary system** of non-interacting quasi-particles (**Kohn-Sham electrons**):

$$n^0(\mathbf{r}) = \sum_i |\phi_i^{KS}(\mathbf{r})|^2$$

$$h^{KS} [n^0(\mathbf{r})] \phi_i^{KS}(\mathbf{r}) = \varepsilon_i \phi_i^{KS}(\mathbf{r})$$

W. Kohn and L. J. Sham, *Self-Consistent Equations Including Exchange and Correlation Effects*, Phys. Rev. **140** A1133 (1965).

Kohn-Sham Equations:

Kohn-Sham quasi-particles have eigenfunctions and eigenenergies satisfying the single-particle Schrodinger's equations:

$$\left(-\frac{1}{2}\nabla^2 + v_{ext}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + v_{xc}[n(\mathbf{r})] \right) \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r})$$

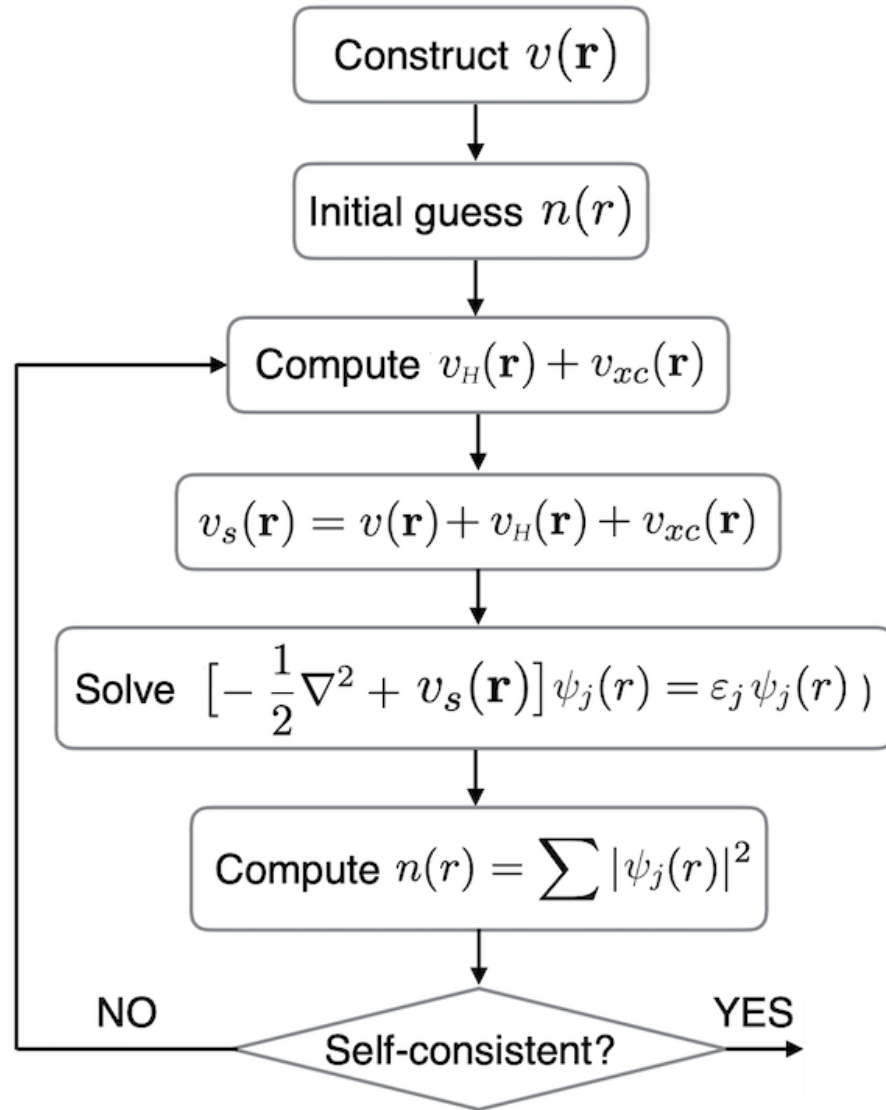
With the **self-consistency** condition:

$$n_0(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2$$

The **Total Energy** is given by:

$$E = 2 \sum_{j=1}^{N/2} \epsilon_j + E_{xc}[n(\mathbf{r})] - \frac{e^2}{2} \iint \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' - \int d\mathbf{r} v_{xc}(\mathbf{r}) n(\mathbf{r})$$

DFT: Self-Consistent Cycle



Kohn-Sham Equations for solids:

- Due to Bloch's theorem, the energy levels in a solid form **bands**:

$$i \longrightarrow (n, \mathbf{k})$$

- Since the Hamiltonian is lattice-periodic, KS equations are diagonal in \mathbf{k} space:

$$\left[-\frac{\nabla^2}{2} + V_{\text{eff}}(\mathbf{r}) \right] \psi_{n\mathbf{k}}(\mathbf{r}) = \epsilon_{n\mathbf{k}} \psi_{n\mathbf{k}}(\mathbf{r})$$

$$n(\mathbf{r}) = \frac{1}{N_k} \sum_{\mathbf{k}, n} f_{n\mathbf{k}} |\psi_{n\mathbf{k}}(\mathbf{r})|^2$$

The **sum** over occupied states is a **sum over occupied bands**; this requires additional care in the case of metals (->LAB3).

Solution of K-S Equations: basis sets

Kohn–Sham orbitals are expanded in a chosen basis set, and the problem reduces to determining the corresponding expansion coefficients. This turns the **differential Kohn–Sham equation** into a secular (eigenvalue) problem.

$$\psi_i(\mathbf{r}) = \sum_{\alpha} c_i^{\alpha} \phi_{\alpha}(\mathbf{r})$$

$$\sum_{\beta} H_{\alpha\beta} c_i^{\beta} = \epsilon_n \sum_{\beta} S_{\alpha\beta} c_i^{\beta}$$

With:

$$H_{\alpha\beta} = \int \phi_{\alpha}^*(\mathbf{r}) \hat{H}_{\text{KS}} \phi_{\beta}(\mathbf{r}) d\mathbf{r}$$

$$S_{\alpha\beta} = \int \phi_{\alpha}^*(\mathbf{r}) \phi_{\beta}(\mathbf{r}) d\mathbf{r}$$

Solution of K-S Equations: basis sets

The expansion is in principle exact if the basis set is infinite. However, in order to implement it on a computer it is necessary to use a finite basis set.

$$\sum_{\beta} H_{\alpha\beta} c_i^{\beta} = \epsilon_n \sum_{\beta} S_{\alpha\beta} c_i^{\beta}$$

The size of the problem (speed of the calculation) depends on the number of basis set functions.

➤ Two possible strategies:

- **Smart basis sets:** Small, but may not generalize, complicated integrals.
- **Analytical basis sets:** (Plane waves, gaussians): simpler integrals, in general need many more WFs, very general.

QE: Planewaves:

Plane waves (PW): natural choice for solids (free electrons); **complete basis set**.

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

Kohn-Sham Equations:

$$\sum_{\mathbf{G}'} H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) c_{n\mathbf{k}}(\mathbf{G}') = \epsilon_{n\mathbf{k}} c_{n\mathbf{k}}(\mathbf{G})$$

$$H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) = \frac{1}{2} |\mathbf{k} + \mathbf{G}|^2 \delta_{\mathbf{G}\mathbf{G}'} + V_{\text{eff}}(\mathbf{G} - \mathbf{G}')$$

The **size** of the Hamiltonian matrix to be diagonalized is determined by the **number of Fourier components in the potential term**.

QE: Planewaves:

Plane waves (PW): natural choice for solids (free electrons); **complete basis set**.

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

Kohn-Sham Equations:

$$\sum_{\mathbf{G}'} H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) c_{n\mathbf{k}}(\mathbf{G}') = \epsilon_{n\mathbf{k}} c_{n\mathbf{k}}(\mathbf{G})$$

$$H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) = \frac{1}{2} |\mathbf{k} + \mathbf{G}|^2 \delta_{\mathbf{G}\mathbf{G}'} + V_{\text{eff}}(\mathbf{G} - \mathbf{G}')$$

The **size** of the Hamiltonian matrix to be diagonalized is determined by the **number of Fourier components in the potential term**.

The Pseudopotential trick

Pseudopotentials are a «trick» to ensure that the **planewave expansion** converges with a **reasonable number of basis set functions**.

The size of the problem is determined by the number of Fourier components in the potential expansion:

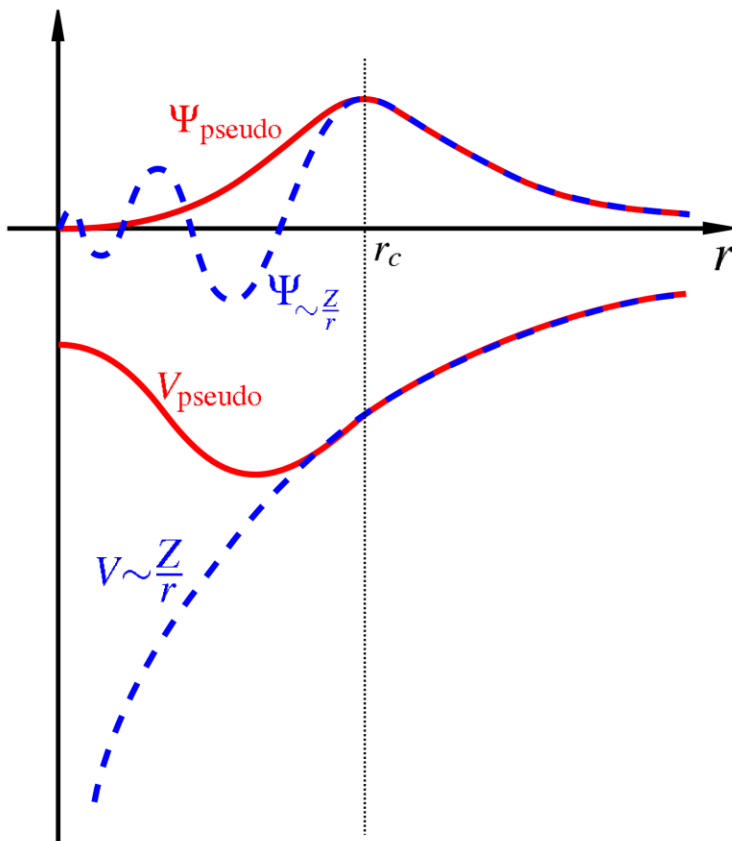
$$V_{\text{eff}}(\mathbf{G}) = \frac{1}{\Omega} \int V_{\text{eff}}(\mathbf{r}) e^{-i\mathbf{G}\cdot\mathbf{r}} d\mathbf{r} .$$

$$V_{\text{eff}}(\mathbf{r}) = V_{\text{ion}}(\mathbf{r}) + V_{\text{H}}(\mathbf{r}) + V_{\text{xc}}[n](\mathbf{r})$$

$$V_{\text{eff}}(\mathbf{G}) = V_{\text{ion}}(\mathbf{G}) + \frac{4\pi}{|\mathbf{G}|^2} \left[\frac{1}{N_k} \sum_{\mathbf{k},n} f_{n\mathbf{k}} \sum_{\mathbf{G}'} c_{n\mathbf{k}}^*(\mathbf{G}') c_{n\mathbf{k}}(\mathbf{G}' + \mathbf{G}) \right] + V_{\text{xc}}(\mathbf{G})$$

The Pseudopotential trick

The number of Fourier components needed to represent the potential is determined by the ionic potential. In the pseudopotential approximation, the full ionic potential is replaced by a **smoother potential** which correctly captures the behaviour of external (valence) electrons:

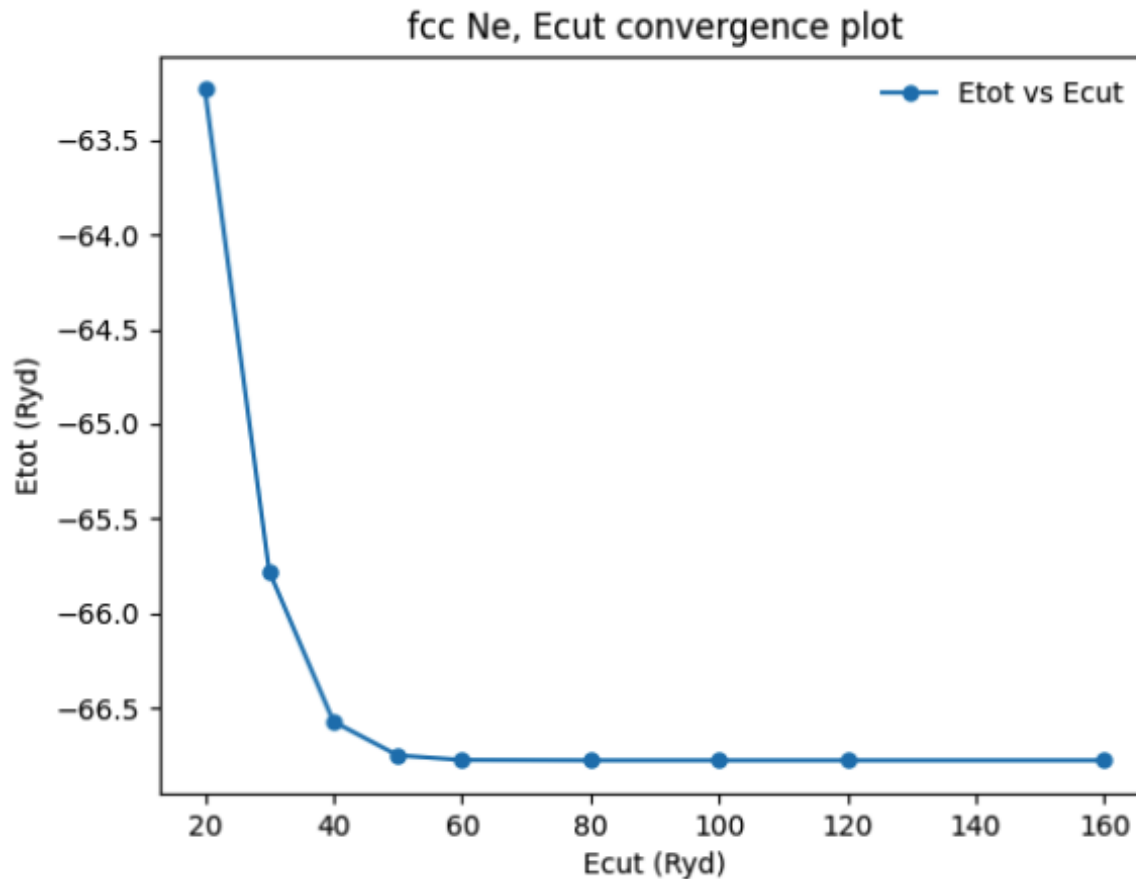


$$v_{\text{ion}}(\mathbf{r} - \mathbf{R}_{\alpha}) \rightarrow v_{\text{ps}}(\mathbf{r} - \mathbf{R}_{\alpha})$$

Pseudopotential files (one for each atom type) can be downloaded from **online repositories** like pseudodojo (<http://www.pseudo-dojo.org/>).

Convergence of the basis set size:

Since plane waves form a **complete basis set**, the convergence of the expansion to the exact solution is variational, i.e. increasing the number of the basis function the **ground-state energy converges to the exact solution from above**.





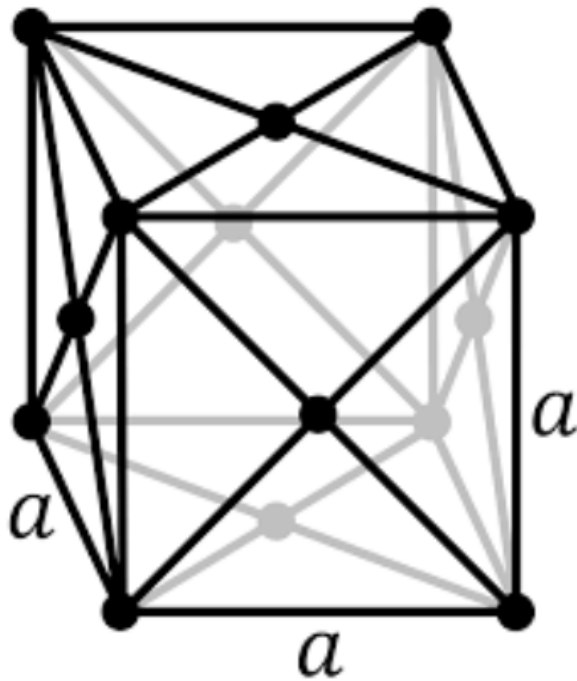
First Guided Example: FCC Neon



How to setup a DFT calculation:

- Setup (and visualize) the **crystal structure**.
- Select exchange and correlation potential (LDA, GGA, ...) and a suitable **pseudopotential**.
- Copy all relevant files in the appropriate directories.
- Run a **preliminary scf calculation**
- **Convergence test:** basis size (**plane wave cutoff**).
- **Convergence test: k points**.
- Once the basic scf calculation is converged, run property calculations!

FCC Neon:



- **Neon**, noble Gas (weak bonding)
- **Z=10**
- **Electronic Configuration:** $(1s^2)2s^2sp^6$

Length

4.46

Angstrom

=

8.428179

Bohr radius

Table 3.1 Lennard-Jones potential parameters for noble-gas crystals

	Neon	Argon	Krypton	Xenon
Crystal structure	fcc	fcc	fcc	fcc
Lattice constant (\AA)	4.46	5.31	5.64	6.13
ϵ (10^{-20} J)	0.050	0.167	0.225	0.320
ϵ (eV)	0.031	0.0104	0.014	0.0200
σ (\AA)	2.74	3.40	3.65	3.98

QE Input file (fcc neon):

neon.scf.in

&CONTROL

```
calculation = 'scf'  
restart_mode = 'from_scratch'  
outdir = './tmp'  
pseudo_dir = './PSEUDO'  
prefix = 'neon'  
tstress = .true.  
tprnfor = .true.
```

/

&SYSTEM

```
ibrav = 2  
celldm(1) = 8.43  
nat = 1  
ntyp = 1  
ecutwfc = 160  
occupations = 'fixed'
```

/

&ELECTRONS

```
conv_thr = 1.0e-6
```

/

ATOMIC_SPECIES

```
Ne 20.18 Ne.LDA.upf
```

ATOMIC_POSITIONS crystal

```
Ne 0.00 0.00 0.00
```

K_POINTS automatic

```
4 4 4 0 0 0
```

CONTROL SECTION: Calculation type, information on input/output info & files.

SYSTEM SECTION: Specifies the system (lattice type, lattice vectors, etc).

ATOMIC SPECIES AND COORDINATES: Types and position of each atom.

QE Input file (fcc neon):

```
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'neon'
  tstress = .true.
  tprnfor = .true.
```

CONTROL SECTION: Calculation type, information on input/output info & files.

Pseudopotential files

```
/
&SYSTEM
  ibrav = 2
  celldm(1) = 8.43
  nat = 1
  ntyp = 1
  ecutwfc = 160
  occupations = 'fixed'
```

SYSTEM SECTION: Specifies the system (lattice type, lattice vectors, etc).

```
/
&ELECTRONS
  conv_thr = 1.0e-6
```

```
/
ATOMIC_SPECIES
  Ne 20.18 Ne.LDA.upf
ATOMIC_POSITIONS crystal
  Ne 0.00 0.00 0.00
K_POINTS automatic
  4 4 4 0 0 0
```

ATOMIC SPECIES AND COORDINATES: Types and position of each atom.

Setting up the Crystal Structure:

```
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'neon'
  tstress = .true.
  tprnfor = .true.
```

```
/
&SYSTEM
  ibrav = 2
  celldm(1) = 8.43
  nat = 1
  ntyp = 1
  ecutwfc = 160
  occupations = 'fixed'
```

Crystal Structure: Definition of the unit cell:
Bravais lattice, lattice constant, number of atoms in the unit cell, types of atoms.

```
/
&ELECTRONS
  conv_thr = 1.0e-6
```

```
/
ATOMIC_SPECIES
  Ne 20.18 Ne.LDA.upf
ATOMIC_POSITIONS crystal
  Ne 0.00 0.00 0.00
```

Crystal Structure: Types of atoms and Position of the atoms in the unit cell

```
K_POINTS automatic
  4 4 4 0 0 0
```

- The **ibrav** variable (1-14) permits to select one of the fourteen 3d Bravais lattices.
- https://en.wikipedia.org/wiki/Bravais_lattice
- If the crystal is non-primitive, you can specify the basis using the card ATOMIC POSITIONS.

- The meaning of all input variables is documented in the file **INPUT_PW.txt**:

```
Variable:      ibrav

Type:          INTEGER
Status:        REQUIRED
Description:    Bravais-lattice index. Optional only if space_group is set.
                If ibrav /= 0, specify EITHER [ "celldm"(1)-"celldm"(6) ]
                OR [ "A", "B", "C", "cosAB", "cosAC", "cosBC" ]
                but NOT both. The lattice parameter "alat" is set to
                alat = celldm(1) (in a.u.) or alat = A (in Angstrom);
                see below for the other parameters.
                For ibrav=0 specify the lattice vectors in "CELL_PARAMETERS",
                optionally the lattice parameter alat = celldm(1) (in a.u.)
                or = A (in Angstrom). If not specified, the lattice parameter is
                taken from "CELL_PARAMETERS"
                IMPORTANT NOTICE 1:
                with ibrav=0 lattice vectors must be given with a sufficiently large
                number of digits and with the correct symmetry, or else symmetry
                detection may fail and strange problems may arise in symmetrization
                IMPORTANT NOTICE 2:
                do not use celldm(1) or A as a.u. to Ang conversion factor,
                use the true lattice parameters or nothing,
                specify units in "CELL_PARAMETERS" and "ATOMIC_POSITIONS"

ibrav      structure      celldm(2)-celldm(6)
              or: b,c,cosbc,cosac,cosab

0          free
          crystal axis provided in input: see card "CELL_PARAMETERS"

1          cubic P (sc)
          v1 = a(1,0,0), v2 = a(0,1,0), v3 = a(0,0,1)

2          cubic F (fcc)
          v1 = (a/2)(-1,0,1), v2 = (a/2)(0,1,1), v3 = (a/2)(-1,1,0)
```

TODO: it's also possible to define the unit cell using space group and Wyckoff positions (see **sgroup** variable).

Self-
Consistent

QE Input file (fcc neon):

Calculation

```
&CONTROL  
  calculation = 'scf'  
  restart_mode = 'from_scratch'  
  outdir = './tmp'  
  pseudo_dir = './PSEUDO'  
  prefix = 'neon'  
  tstress = .true.  
  tprnfor = .true.
```

CONTROL SECTION: Calculation type, information on input/output info & files.

Pseudopotential files

```
/  
&SYSTEM  
  ibrav = 2  
  celldm(1) = 8.43  
  nat = 1  
  ntyp = 1  
  ecutwfc = 160  
  occupations = 'fixed'
```

SYSTEM SECTION: Specifies the system (lattice type, lattice vectors, etc).

```
/  
&ELECTRONS  
  conv_thr = 1.0e-6
```

ATOMIC SPECIES AND COORDINATES: Types and position of each atom.

```
ATOMIC_SPECIES  
  Ne 20.18 Ne.LDA.upf  
ATOMIC_POSITIONS crystal  
  Ne 0.00 0.00 0.00  
K_POINTS automatic  
  4 4 4 0 0 0
```

Crystal structure

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

Run a **preliminary** DFT calculation:

1. To run Quantum Espresso, open a terminal and type:

```
pw.x < neon.scf.in > neon.scf.out
```

2. If no error message appears, the output of your calculation will be found in `neon.scf.out`.

3. You can check your crystal structure by using a post-processing tool that works on QE output files and creates a file that can be visualized with VESTA: `pp.x <rho.pp.in > rho.pp.out`.

This generates the file `neon.rho.xsf` which can be read by VESTA.

4. If for some reason VESTA doesn't work on your pc, you can open the xsf output files using `xcrysden`.



Convergence Tests: basis set size

QE Input file (fcc neon):

&CONTROL

```
calculation = 'scf'
restart_mode = 'from_scratch'
outdir = './tmp'
pseudo_dir = './PSEUDO'
prefix = 'neon'
tstress = .true.
tprnfor = .true.
```

CONTROL SECTION: Calculation type, information on input/output info & files.

/

&SYSTEM

```
ibrav = 2
celldm(1) = 8.43
nat = 1
ntyp = 1
ecutwfc = 160
occupations = 'fixed'
```

SYSTEM SECTION: Specifies the system (lattice type, lattice vectors, etc).

Plane wave cutoff: Defines the number of plane waves used for K-S expansion

/

&ELECTRONS

```
conv_thr = 1.0e-6
```

/

ATOMIC_SPECIES

```
Ne 20.18 Ne.LDA.upf
```

ATOMIC_POSITIONS crystal

```
Ne 0.00 0.00 0.00
```

K_POINTS automatic

```
4 4 4 0 0 0
```

ATOMIC SPECIES AND COORDINATES: Types and position of each atom.

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

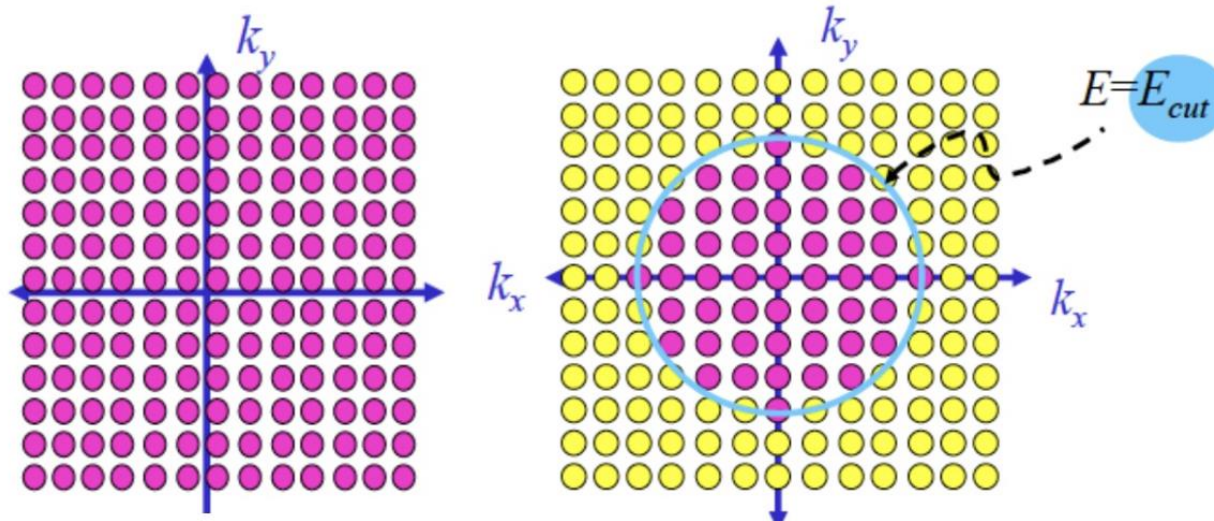
Convergence Tests: Basis set

1) How Many Plane Waves (Basis set functions)?

$$\sum_{\mathbf{G}'} H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) c_{n\mathbf{k}}(\mathbf{G}') = \epsilon_{n\mathbf{k}} c_{n\mathbf{k}}(\mathbf{G})$$

$$H_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) = \frac{1}{2} |\mathbf{k} + \mathbf{G}|^2 \delta_{\mathbf{G}\mathbf{G}'} + V_{\text{eff}}(\mathbf{G} - \mathbf{G}')$$

- The potential term couples \mathbf{G} , \mathbf{G}' ; For each value of \mathbf{k} we to diagonalize a $\mathbf{N}_{\text{pw}} \times \mathbf{N}_{\text{pw}}$ matrix.
- To decide the number of plane waves, most PW codes introduce a **kinetic energy cutoff**, i.e. we drop all components of the expansion for which:



N.B. The **number of planewaves** depends **critically** on the choice of the **pseudopotential**.

Bash Script (1):

To execute qe several times with different values of ecut it is convenient to use a simple bash script:

```
for ECUT in 20 30 40 50 60 80 100 120 160
do
cat > neon.scf.in <<EOF
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'neon'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
 ibrav = 2
  celldm(1) = 8.43
  nat = 1
  ntyp = 1
  ecutwfc = $ECUT
  occupations = 'fixed'
/
&ELECTRONS
  conv_thr = 1.0e-6
/
ATOMIC_SPECIES
  Ne 20.18 Ne.LDA.upf
ATOMIC_POSITIONS crystal
  Ne 0.00 0.00 0.00
K_POINTS automatic
  4 4 4 0 0 0
EOF
# Run the SCF calculation
pw.x < neon.scf.in > neon.$ECUT.scf.out
```

test_ecut.sh

Bash Script (2):

Adding a few lines we can automatically extract the total energy and create a two column file for plotting:

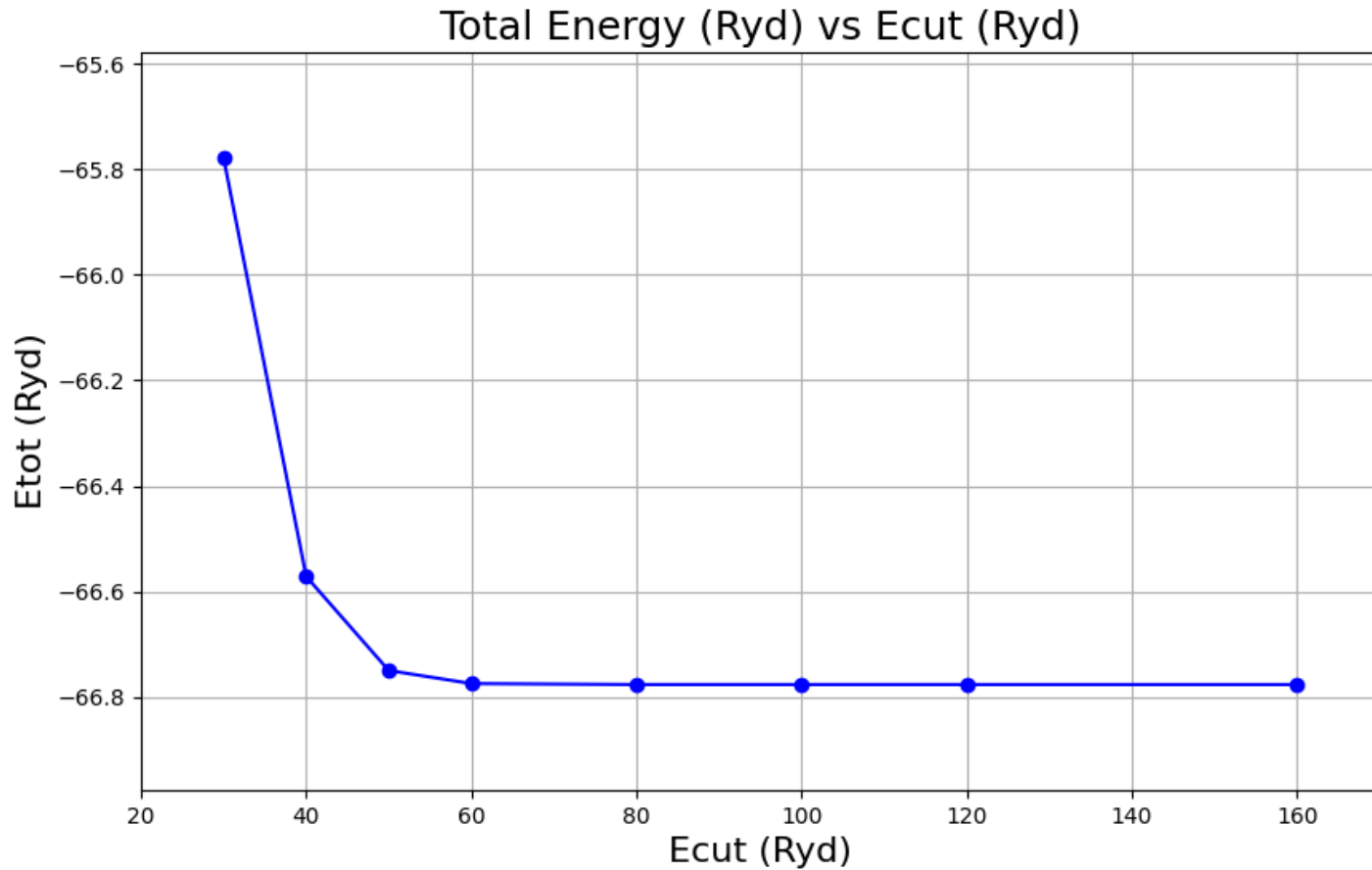
```
# Check if energy_summary.dat exists and delete it if so  
if [ -f energy_summary.dat ]; then  
    rm energy_ecut.dat  
fi
```

test_ecut_full.sh

```
# Extract the total energy and append to the summary file  
ETOT=$(grep '!' neon.$ECUT.scf.out | cut -d'=' -f2 | cut -d'R' -f1)  
echo "$ECUT $ETOT" >> energy_ecut.dat
```

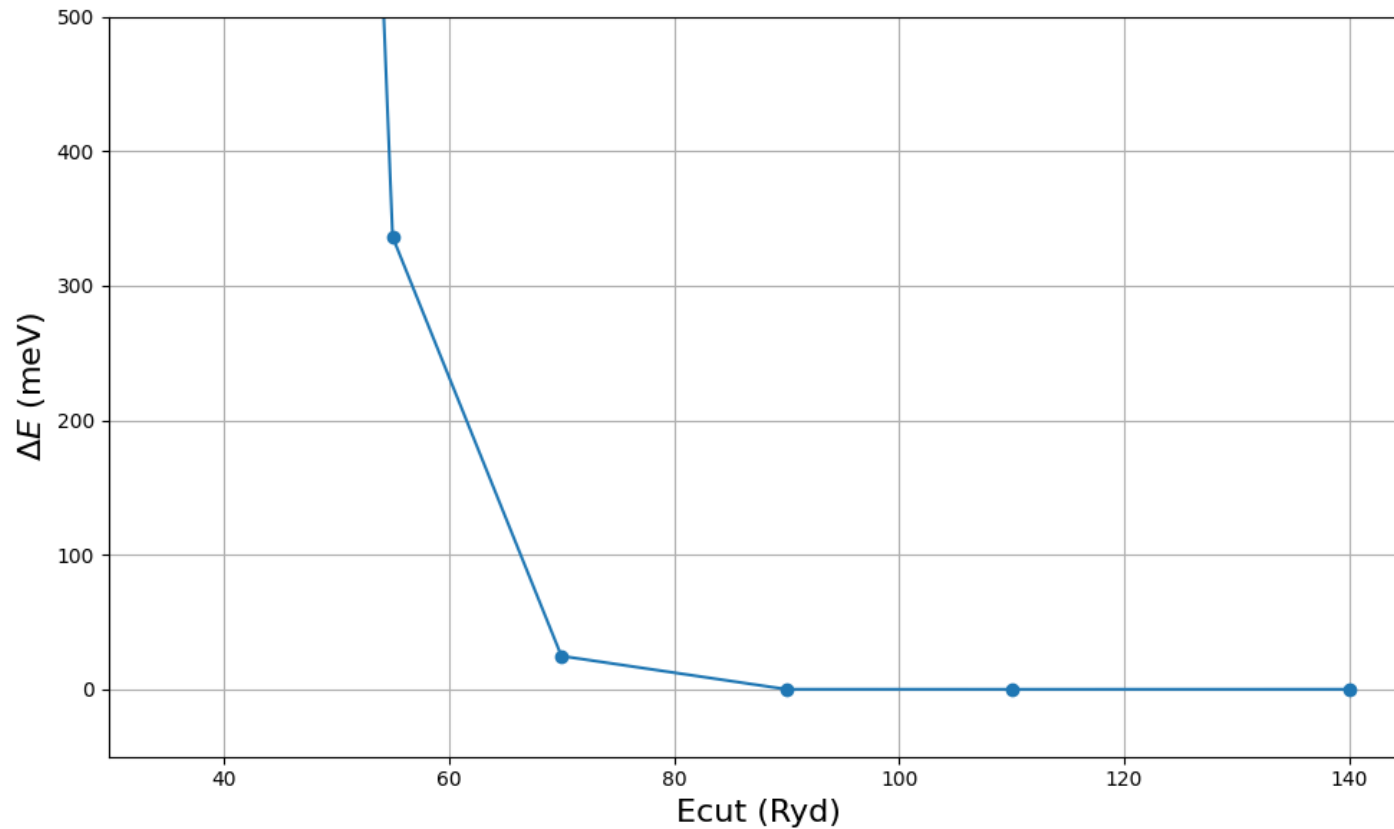
You can then plot the resulting file with any 2d plot program (for example, you can use the python script plot_ecut.py).

Convergence Tests: Basis set



- Neon has a «hard» pseudopotential (large ecut); first and second-row elements are «hard» (many G component are required to converge).

Convergence Tests: Basis set



To visualize convergence it's usually convenient to monitor the error: $E(i+1)-E(i)$.

Charge Density Plots:

- To obtain a charge Density Plot we need to post-process the **binary output files** of quantum espresso, using the package **pp.x**.

```
&INPUTPP
prefix='neon'
plot_num=0
outdir='./tmp'
filplot='neon.rho.out'
&end

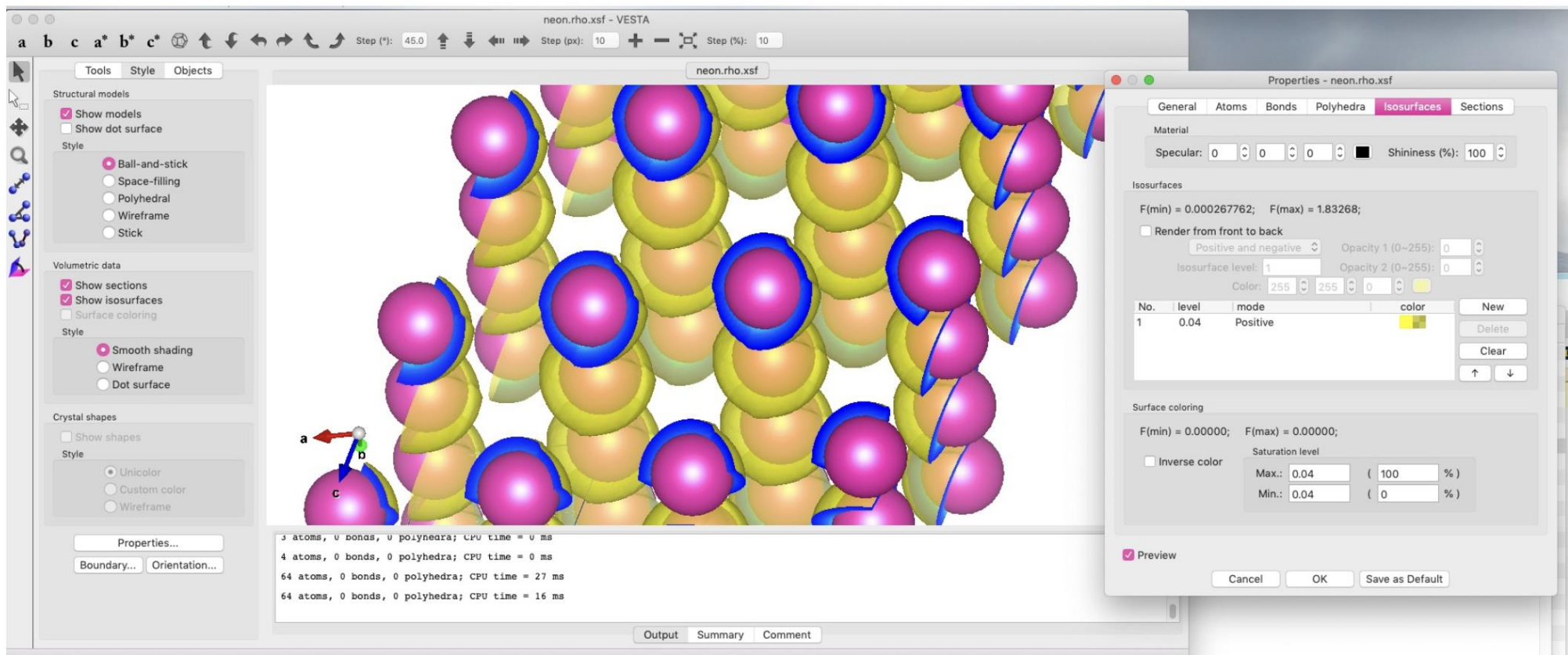
&PLOT
nfile=1
weight(1)=1.0
iflag=3
output_format=3
fileout='neon.rho.xsf',
nx=32,ny=32,nz=32
&end
```

rho.pp.in

- 1) Run pp as: **pp.x < rho.pp.in > rho.pp.out**

Full manual at: https://www.quantum-espresso.org/Doc/INPUT_PP.html

2) The file neon.rho.xsf can be visualized with VESTA/Xcrysden:





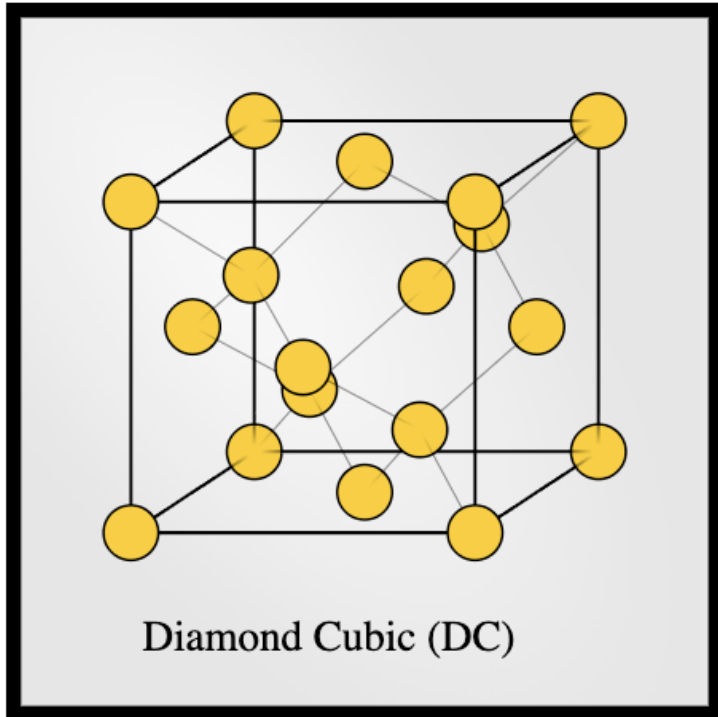
TODO: Exercise

Assignment:

1. Download the attached Silicon LDA pseudopotential (Si.LDA.upf).
2. Setup and converge calculations for cubic (diamond) silicon and hexagonal silicon, repeating what we did for fcc neon.
3. Plot the charge density for the two structures: which difference do you observe in the charge density distribution compared to neon? Why?

Cubic Silicon (*diamond* structure)

<https://msestudent.com/diamond-cubic-unit-cell/>



Lattice (Conventional)

a	5.44 Å
b	5.44 Å
c	5.44 Å
α	90.00 °
β	90.00 °
γ	90.00 °
Volume	161.32 Å ³

<https://next-gen.materialsproject.org/materials/mp-149>

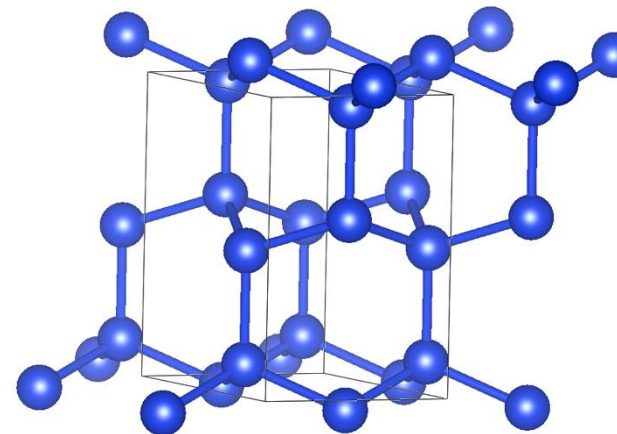
- The **Unit cell** is fcc, with a two-atom basis (space group 227, Wyckof positions **8a**):

$$\left\{ \begin{array}{l} \mathbf{R}_1 = \frac{a}{2}(0, 1, 1) \\ \mathbf{R}_2 = \frac{a}{2}(1, 0, 1) \\ \mathbf{R}_3 = \frac{a}{2}(1, 1, 0) \end{array} \right.$$

$$\left\{ \begin{array}{l} \boldsymbol{\tau}_1 = (0, 0, 0) \\ \boldsymbol{\tau}_2 = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) \end{array} \right.$$

Hexagonal Silicon (*lonsdaleite* structure):

<https://materialsproject.org/materials/mp-165>



Lattice (Conventional)

a	3.83 Å
b	3.83 Å
c	6.33 Å
α	90.00 °
β	90.00 °
γ	120.00 °
Volume	80.50 Å ³

Atomic Positions

Wyckoff	Element	x	y	z
4f	Si	1/3	2/3	0.06301

Space group 194, hexagonal unit cell, 4 atoms, 4f Wyckoff positions.