Self-Consistent Field (SCF) Calculations
Hands-on with Quantum ESPRESSO

Emmanuel Oppong

Supervisor: Elliot Sarpong Menkah (Dr.)

Quantum ESPRESSO Practical Tutorial

October 28, 2025

A condensed, practical workshop: SCF, convergence, EOS, bands, DOS, metals, magnetism, relaxations, and graphene examples.

Outline

Overview: Day 1 Hands-on Session

What We'll Cover

Before diving into quantum calculations, we need to master the tools for:

- **Plotting data** — XMGrace for 2D graphs
- **Visualizing structures** — XCrySDen and VESTA for crystal structures

Important Notes

- File names are shown in magenta
- Commands to type are shown in `blue`
- Input parameters are shown in dark green

Useful Linux Commands

`vi/gedit/emacs` — open files `mkdir/mv/rm` — create/move/delete
`cat/less` — view files `grep` — search text

What is XMGrace?

XMGrace Overview

XMGrace is a plotting package for generating 2D graphs

- Export results in scalar and vector formats (publication-ready!)
- Simple function fitting capabilities
- Most tutorials use XMGrace or gnuplot

Getting Started

Launch XMGrace from terminal:

```
xmgrace
```

Exercise 1: Making a Graph from Data

Dataset: Volume-Energy Calculations

File: example1.dat contains Quantum ESPRESSO results

- Column 1: Volume
- Column 2: Total energy for FCC structure
- Column 3: Total energy for BCC structure

Step 1: View the data

```
vim example1.dat
```

Step 2: Launch XMGrace

```
xmgrace
```

Importing Data into XMGrace

1. In XMGrace menu: **Data → Import → ASCII**
2. Navigate to `Day1/xmgrace` directory
3. Click ONCE on example1.dat
4. In 'Load As' box, select **Block Data** and click OK
5. Select X data from **column 1**, Y data from **column 2**
6. Click **Apply** and close

Why Block Data?

Block Data allows us to choose specific columns to plot (we only want columns 1 and 2 first)

Customizing Axis Range

Step 1: Open Axis Properties

Plot → Axis Properties

For X Axis:

- Select **Edit: X Axis**
- Change X range: 20 to 45
- Click **Apply**

For Y Axis:

- Select **Edit: Y Axis**
- Change Y range: -220 to 0
- Click **Apply**

Adding Labels and Legends

X Axis Label:

- In Label String: `Volume (\c\E\C\S3\N)`
- This creates: Volume (Å^3)
- Set Major Spacing: 4, Minor Ticks: 0

Y Axis Label:

- Label String: `Energy (Hartree)`

Add Legend:

- Double-click on plotted data
- In Legend box: type "FCC"
- Click **Apply**

Add Title:

- **Plot → Graph Appearance**, enter "FCC structure"

Improving Plot Appearance

Make axis labels clearer:

- **Plot → Axis Properties**
- Click **Axis label & bar** tab
- Font: Helvetica-Bold, Char size: 120
- Bar Width: 2

Adjust tick labels:

- Click **Tick labels** tab, Char size: 120
- Click **Tick marks** tab, Line width: 2

Add symbols and change line style:

- Double-click on curve
- Symbol Properties → Type: Circle
- Line Properties → Change Style and Color

Saving and Exporting

Save as XMGrace file (.agr):

- **File → Save As**
- Filename: example1.agr

Export as image (e.g., PNG):

1. **File → Print Setup**
2. Device: select `PNG`
3. **File → Print** (this creates example1.png)
4. **File → Exit**

Verify files created:

```
ls
```

XMGrace Tasks

Task 1

Practice with plotting options to reproduce a professional-looking graph

Task 2

Plot **both** Column 2 and Column 3 as functions of Column 1

- This will show FCC and BCC structures on the same plot
- Use different colors/symbols for each dataset
- Add appropriate legends

Hint

Import the data twice, selecting different Y columns each time

Exercise 2: Fitting Data

Objectives

- Fit dataset to nth order polynomial
- Fit dataset to custom equation

Fitting to nth Order:

1. Import column 1 (X) and column 2 (Y)
2. **Data → Transformations → Regression**
3. Select fit type (e.g., Quadratic)
4. Click **Accept**
5. A pop-up shows the equation fit

Task

Find a 2nd order fit to columns 1 and 3. Display raw data as symbols and fit as a line.

Visualizing Crystal Structures

Why Visualize?

- DFT input files: lattice parameters + atomic coordinates (numbers)
- Visualization: actual spatial arrangement (pictures!)
- Much better insight into structure

Two programs:

- **XCrySDen**: Directly reads Quantum ESPRESSO files
- **VESTA**: Better for high-quality figures (needs file conversion)

XCrySDen Basics

Launch XCrySDen:

```
xcrysden
```

Load structure:

- File menu → Open Day1/calculations/PbTiO3/pto.scf.in
- Format: `pw.x` input format

Explore:

- Buttons at bottom: atom info, distances, angles
- **Exercise**: What is Ti coordination? Find Ti-O distances. Are they equal?

Export:

- **Tools → k-path** (for band structure calculations)
- File menu → Export to `xsf` format (for VESTA)

VESTA: High-Quality Visualization

Launch VESTA:

```
VESTA
```

Basic operations:

- Load xsf file from XCrySDen export
- Check distances/angles: buttons on left side
- Add bonds/polyhedra: **Edit → Bonds**
- Modify boundaries: **Objects → Boundaries**
- View unit cell info: **Edit → Edit data → Unit cell**
- Export image: **File → Export raster image** (PNG)

Visualizing Electron Density

Charge Density from Quantum ESPRESSO

Load cube file: Day1/calculations/Si/si.rho.cube

Customize visualization:

- Modify isosurface: **Objects → Properties → Isosurfaces**
- 2D plot: **Utilities → 2D Data Display**

Exercise

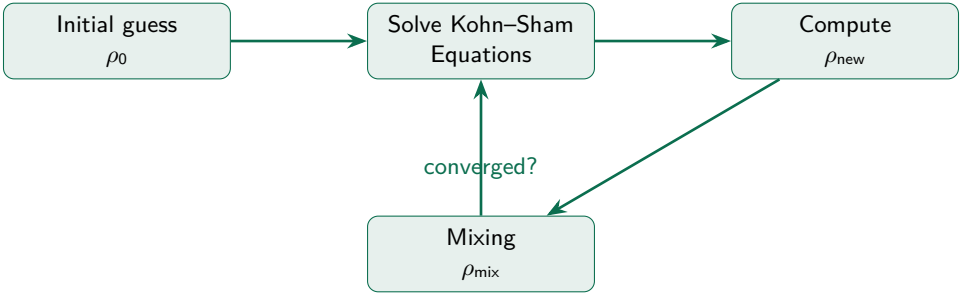
Pick a crystal structure from VESTA's library and analyze:

- What elements?
- Coordination polyhedra?
- How are they connected?

What is an SCF run? (High level)

- SCF = **Self-Consistent Field**. Iterative method solving the Kohn–Sham DFT equations.
- Goal: obtain ground-state electron density $\rho(\mathbf{r})$ and total energy E_{tot}.
- Loop: guess $\rho_0 \rightarrow$ build $V_{\text{KS}}[\rho] \rightarrow$ solve for $\{\psi_i\} \rightarrow$ get $\rho_1 \rightarrow$ mix $\rightarrow$ check convergence.

SCF loop (diagram)



Stop when $\|\rho_{n+1} - \rho_n\| < \varepsilon$ (or $\Delta E <$ tol).

Understanding the Convergence Criterion

Mathematical Expression

$$\|\rho_{n+1} - \rho_n\| < \varepsilon$$

- $\|\cdot\|$: Norm (magnitude/distance) — usually Euclidean distance
- ρ_n: Electron density at iteration n
- ρ_{n+1}: Electron density at next iteration $n+1$
- ε **(epsilon)**: Small tolerance value (e.g., 10^{-3}, 10^{-6})

Translation

Stop when the change between successive iterations is smaller than a threshold

Why SCF matters

- Provides the self-consistent electronic potential and eigenvalues — input to bands, DOS, response properties.
- SCF convergence quality affects all derived properties (energies, forces, phonons).
- Use conservative thresholds for production runs: `conv_thr` $\leq 10^{-8}$ (example).

Pseudopotentials: motivation and types

- **Core vs valence** — core electrons are chemically inert; valence electrons control bonding.
- **Pseudopotentials** (UPF files) replace nucleus + core electrons with a smooth effective potential.
- Types: Norm-conserving (NC), Ultrasoft (USPP), PAW. Choose according to accuracy/performance trade-offs.
- Example: `Si.pz-vbc.UPF` (LDA-type pseudo used in many examples).

Plane-Waves & Cutoffs: The Simple Story

The Core Idea (In Plain English)

Problem: We need to describe where electrons are in a crystal.
Solution: Use **waves** (like sound waves or ocean waves) to build electron shapes.

Think of it like building with LEGO blocks:

- **Plane waves** = Your LEGO blocks (simple building units)
- **Electron wavefunction** = The structure you build from those blocks
- **More blocks** = More detailed structures (but takes longer to build!)

The Key Question

How many LEGO blocks (plane waves) do we need?

$\rightarrow$ **This is what the "cutoff" decides!**

Plane-wave basis & energy cutoffs

- Wavefunctions expanded in plane waves up to kinetic energy cutoff `ecutwfc`.
- Density cutoff `ecutrho` typically = 4–12× `ecutwfc` (USPP needs high `ecutrho`).
- Variational principle ⇒ energy decreases monotonically with increasing basis size (cutoff).

$$E_{\text{cut}} = \frac{\hbar^2 G_{\text{max}}^2}{2m_e}$$

Understanding Plane-Wave Basis Sets

Why Plane Waves?

- **Plane waves**: $e^{i\mathbf{G}\cdot\mathbf{r}}$ where $\mathbf{G}$ are reciprocal lattice vectors
- Orthogonal, complete basis set — ideal for periodic systems (crystals)
- Wavefunctions: $\psi(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{G}} e^{i\mathbf{G}\cdot\mathbf{r}}$

The Truncation Problem

Cannot include infinite plane waves — must truncate at some maximum $|\mathbf{G}|$

Energy Cutoff Explained

Cutoff Formula

$$E_{\text{cut}} = \frac{\hbar^2 G_{\text{max}}^2}{2m_e}$$

Breaking it down:

- $\hbar$: Reduced Planck constant ($h/2\pi$) — quantum scale
- G_{max}: Maximum reciprocal vector magnitude — highest frequency wave included
- m_e: Electron mass
- E_{cut}: Maximum kinetic energy of plane waves in the basis set

Physical Meaning

Only include plane waves with kinetic energy $\leq E_{\text{cut}}$. Higher cutoff = more waves = better accuracy (but slower!)

Why Two Different Cutoffs? (Simple Explanation)

The Photography Analogy

Think of taking a photo and then zooming in on details:

`ecutwfc`

The Original Photo

- Describes *one electron*
- Like a photo with certain resolution
- Set this value in input file

Example: `ecutwfc = 40`

`ecutrho`

The Zoomed Detail

- Describes *electron density* (many electrons)
- Needs **finer detail** (higher resolution)
- Usually 4–12× bigger!

Example: `ecutrho = 160`

Simple Rule

Density needs more detail than wavefunctions!

Because: density = wavefunction × wavefunction (multiplying creates finer details)

Two Different Cutoffs: `ecutwfc` vs `ecutrho`

`ecutwfc`

Wavefunction cutoff

- Controls expansion of $\psi(\mathbf{r})$
- Direct representation of electronic states
- Typical values: 30–100 Ry

`ecutrho`

Density cutoff

- Controls charge density $\rho = |\psi|^2$
- Product of two wavefunctions
- Needs higher frequencies!

Why `ecutrho` > `ecutwfc`?

- $\rho(\mathbf{r}) \propto \psi^*(\mathbf{r})\psi(\mathbf{r})$ — product doubles the frequency content
- **Norm-conserving PP**: `ecutrho` $= 4\times$ `ecutwfc` (automatic)
- **Ultrasoft PP (USPP)**: needs `ecutrho` $= 8$–$12\times$ `ecutwfc` (set manually!)

Variational Principle & Convergence

Variational Principle in Action

- Ground state energy is the **minimum** of the energy functional
- Larger basis set = more freedom to minimize energy
- $E_{\text{cut}}^{(1)} < E_{\text{cut}}^{(2)} \Rightarrow E_{\text{tot}}^{(1)} \geq E_{\text{tot}}^{(2)}$

Practical convergence strategy:

1. Start with low cutoff (e.g., 20 Ry)
2. Increase systematically: 30, 40, 50, 60 Ry...
3. Stop when $|\Delta E| <$ tolerance (e.g., 1 meV/atom)
4. Energy **decreases** (or stays same) — never increases!

Convergence Check

Plot E_{tot} vs `ecutwfc` — should see asymptotic approach to converged value

What are K-Points? (Simple Explanation)

The Big Picture

Problem: Crystals repeat infinitely — we can't calculate every atom!
Solution: Use **k-points** to sample the crystal's "recipe" instead of the whole crystal.

Think of it like taste-testing soup:

- You don't drink the whole pot to know if it tastes good
- You take a few **sample spoonfuls** from different locations
- More samples = better idea of the overall taste
- K-points are those "sample spoonfuls" in momentum space!

K-Point Grid Example

4×4×4 grid = 64 sample points in 3D space
More k-points = more accurate, but slower calculation

Brillouin Zone: Your Crystal's "Music Sheet"

Simple Analogy: Think of Music

Real space (crystal) = Time (when notes are played)
Brillouin Zone = Sheet music (which notes/frequencies exist)

What is it?

- A "map" of all electron wave patterns
- Like a box containing all possible wavelengths
- Every k-point = one specific wave pattern

Why k-points?

- Can't check every wave—too many!
- Sample a few representative ones
- Grid = evenly spaced samples

The Trade-off

Few k-points ($2\times2\times2$): Fast but rough $\rightarrow$ Quick tests
Many k-points ($12\times12\times12$): Slow but accurate $\rightarrow$ Final calculations

K-points and Brillouin-zone sampling

- Integrals over the BZ are replaced by sums over discrete **k-points**.
- Monkhorst–Pack grids: `K_POINTS automatic`
  `nk1 nk2 nk3 k1 k2 k3`.
- Dense grids required for metals; convergence is not necessarily monotonic.

Convergence testing: recommended workflow

1. Fix pseudopotential and k-grid; sweep `ecutwfc` (16,20,24,28,32 Ry).
2. Fix cutoff; sweep k-point density ($2\times2\times2$,$4\times4\times4$,...).
3. For metals, test smearing type and `degauss`.
4. For USPP, test `ecutrho` ratios: dual = `ecutrho/ecutwfc` = 4,6,8,12.

Present results as tables and E vs parameter plots for clarity.

Silicon — quick facts

- Diamond structure (fcc with 2 atoms per cell).
- Use `ibrav=2`, `celldm(1)` (in a.u.), `nat=2`, `ntyp=1`.
- Typical test cutoffs: start 12–20 Ry; k-grid $6\times6\times6$ etc.

Silicon: SCF input (si.scf.in)

```
&CONTROL
  calculation = 'scf',
  prefix = 'silicon',
  outdir = '../tmp/',
  pseudo_dir = '../pseudo/',
/
&SYSTEM
  ibrav = 2,
  celldm(1) = 10.26,
  nat = 2,
  ntyp = 1,
  ecutwfc = 12.0,
/
&ELECTRONS
  conv_thr = 1.0d-8,
  mixing_beta = 0.7,
/
ATOMIC_SPECIES
Si   28.086   Si.pz-vbc.UPF

ATOMIC_POSITIONS (crystal)
Si   0.00   0.00   0.00
Si   0.25   0.25   0.25

K_POINTS automatic
6 6 6 1 1 1
```

Running Si SCF and checking output

- Run: `pw.x -in si.scf.in > si.scf.out`
- Scratch results: `outdir/prefix.save` contains wavefunctions and data.
- Inspect: `grep -e '!  total energy' -e 'estimated scf accuracy' si.scf.out`
- Example convergence from the tutorial shows monotonic energy lowering with increasing `ecutwfc`.

Si: Equation of State (EOS) workflow

1. Choose converged `ecutwfc`, k-grid (e.g., 20 Ry, $4\times4\times4$).
2. Run SCF for a set of `celldm(1)` values (e.g., 9.8–10.7 a.u. in 0.1 steps).
3. Collect $E(a)$ in a file, fit with `ev.x` to extract a_0 and B_0.

Note: structural properties and energy differences converge faster than total energies.

Band structure: workflow

1. SCF calculation at equilibrium lattice parameter (a_0).
2. Non-SCF `calculation='bands'` with k-path along high-symmetry lines.
3. Use `bands.x` and `plotband.x` for plotting and symmetry analysis.

Keep `prefix` and `outdir` identical between SCF and bands.

DOS & PDOS: workflow

1. Run a dense `nscf` calculation (often with tetrahedron method).
2. Compute DOS: `dos.x -in prefix.dos.in`.
3. For projected DOS: `projwfc.x -in prefix.pdos.in`.
4. Plot spin-up and spin-down DOS for magnetic systems.

Why smearing?

- Metals have a discontinuous occupation at E_F — numerical integration becomes unstable.
- Smearing smooths the occupation near the Fermi level: Gaussian, Methfessel–Paxton (mp), Marzari–Vanderbilt (mv).
- Key variables: `occupations='smearing'`, `smearing='mp'`, `degauss`=0.01–0.05 Ry.

Aluminum: example settings

```
&SYSTEM
  ibrav = 2,
  celldm(1) = 7.5,
  nat = 1,
  ntyp = 1,
  ecutwfc = 16.0,
  occupations = 'smearing',
  smearing = 'mp',
  degauss = 0.03,
/
K_POINTS automatic
12 12 12 1 1 1
```

Smearing convergence remarks

- Test different smearings and `degauss` values; evaluate sensitivity of energies and forces.
- For Aluminum, `mp` or `mv` with `degauss` $\approx$ 0.01–0.05 Ry and a $12\times12\times12$ grid often good.
- Avoid too-small `degauss` which removes the smoothing advantage.

Spin-polarized calculations

- Set `nspin = 2` to treat spin-up and spin-down densities separately.
- Use `starting_magnetization(i)` per species to break symmetry.
- Output includes `total magnetization` and `absolute magnetization`.

Iron (Fe) — SCF input highlights

```
&SYSTEM
  ibrav = 3,
  celldm(1) = 5.42,
  nat = 1,
  ntyp = 1,
  ecutwfc = 25.0,
  ecutrho = 250.0,
  nspin = 2,
  starting_magnetization(1) = 0.5,
  occupations = 'smearing',
  smearing = 'mp',
  degauss = 0.02,
/
```

USPP and dual

- Ultrasoft pseudopotentials require larger `ecutrho`. Typical choice: `ecutrho` = 8–12 × `ecutwfc`.
- Convergence must be checked vs both cutoffs and k-points.
- Magnetic systems double k-point sampling (spin channels) internally in outputs.

Relaxation (ionic) runs

- `calculation = 'relax'`: minimize total energy wrt atomic positions.
- &IONS namelist controls ionic dynamics; common optimizers: BFGS (default).
- Flow: SCF $\rightarrow$ compute forces $\rightarrow$ move atoms $\rightarrow$ repeat until forces $<$ tol.

Supercells: purpose and behavior

- Supercells (e.g., 3×3) model defects, adsorbates, surfaces in periodic codes.
- Reciprocal lattice shrinks accordingly; check equivalence of k-grids between cell and supercell.
- Energies scale: $E_{\mathrm{SC}} \approx N\, E_{\mathrm{UC}}$ for large supercells with equivalent sampling.

Graphene: cell and k-grid

- Hexagonal 2-atom cell: `ibrav=4`, `celldm(1)=4.6542890`, `celldm(3)=3.0` (vacuum).
- Use dense in-plane grid, e.g., $9\times9\times1$.
- Band path: Γ–K–M–Γ reveals Dirac cones at K.

Graphene SCF input (snippet)

```
&SYSTEM
  ibrav = 4,
  celldm(1) = 4.6542890,
  celldm(3) = 3.0,
  nat = 2,
  ntyp = 1,
  ecutwfc = 40.0,
/
K_POINTS automatic
9 9 1 0 0 0
```

Graphane (hydrogenated graphene) — relaxation

- Graphane requires ionic relaxation: `calculation='relax'` + `&IONS`.
- Typical cell: 4 atoms (2 C + 2 H), `ntyp=2`, atomic positions provided in `ATOMIC_POSITIONS`.
- Expect buckling and symmetric C–H bonds above/below plane.

Graphene Oxide (example)

- Model epoxy by adding O to a 3×3 graphene supercell (19 atoms total for one epoxy).
- Relaxation required to find local geometry; analyze formation energy and distortions.

Practical execution tips

- Keep consistent `prefix` and `outdir` across SCF $\rightarrow$ bands/nscf.
- Never run two jobs writing to the same `outdir/prefix.save` concurrently.
- For large runs, use MPI/parallel pw.x and per-job unique scratch directories.
- Document pseudopotential source and all input parameters for reproducibility.

Output checks and common grep commands

```
grep -e "! total energy" si.scf.out
grep -e "the Fermi energy" al.scf.out
grep -e "total magnetization" fe.scf.out
```

Summary (takeaways)

- SCF is the foundation of DFT work in Quantum ESPRESSO.
- Converge `ecutwfc`, `K_POINTS`, and smearings carefully.
- Use appropriate pseudopotentials (NC/USPP/PAW) per element and problem.
- Metals need smearing; magnetic materials need spin-polarized runs; relaxations need ionic steps.
- Supercells let you model defects and surfaces within periodic boundary conditions.

Further resources

- Official QE manuals and tutorials: `https://www.quantum-espresso.org`
- The tutorial slides used here: `handson_pwscf.pdf`
- Example scripts: `run_si_eos`, `run_al_test`, etc.

Questions? `Emmanuel Oppong`