PROGRESS REPORT 20230901

TEST CALCULATIONS OF SILICON

1. **(SCF) Initial scf input file: silicon.in**

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat silicon.in
&CONTROL
calculation = 'scf'
prefix = 'silicon'
outdir = './tmp/'
pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
verbosity = 'high'
/

&SYSTEM
ibrav = 2,
celldm(1) = 10.26,
nat = 2,
ntyp = 1,
ecutwfc = 30
nbnd = 8
/

&ELECTRONS
mixing_beta = 0.6
/

ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
Si 0.0 0.0 0.0
Si 0.25 0.25 0.25

K_POINTS (automatic)
6 6 6 0 0 0
```

---

2. **(CONVERGENCE TESTING) Convergence with cutoff energy using PWTK.**
   pwtk script file name for cutoff energy: si_scf_ecutoff.pwtk

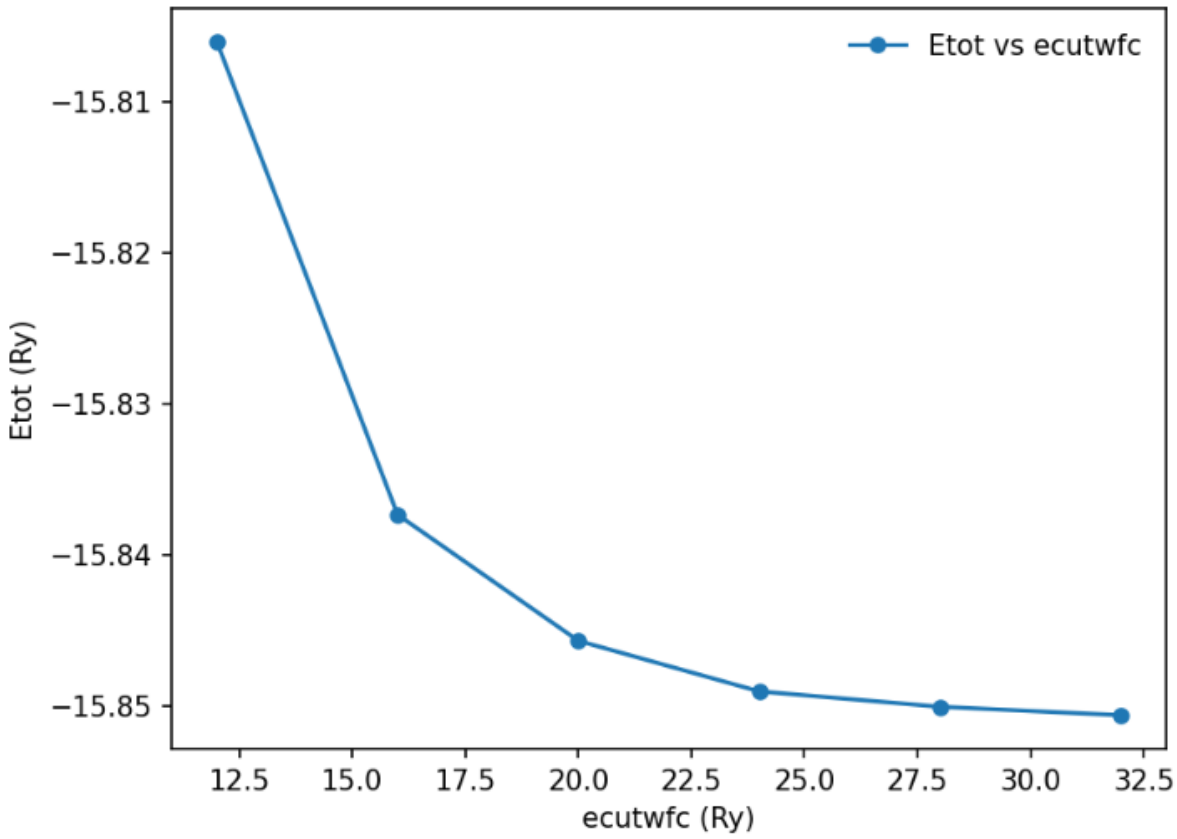
```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_scf_ecutoff.pwtk
# Load the pw.x input from file
load_fromPWI silicon.in

# open a file for writing resulting total energies
set fid [open etot_vs_ecutwfc.dat w]

# loop over different "ecut" values
foreach ecut { 12 16 20 24 28 32 } {

    # name of I/O files: $name.in & $name.out
    set name si_scf_ecutwfc-$ecut

    # set the pw.x "ecutwfc" variable
    SYSTEM "ecutwfc = $ecut"

    # run the pw.x calculation
    runPW $name.in

    # extract the "total energy" and write it to file
    set Etot [::pwtk::pwo::totene $name.out]
    puts $fid "$ecut $Etot"
}

close $fid
```

3. **run the script: pwtk si_scf_ecutoff.pwtk**
   we'll get: etot_vs_ecutwfc.dat

---

4. **plot the total energy with respect to ecutwfc using etot_vs_ecutwfc.dat**

```python
import matplotlib.pyplot as plt
from matplotlib import rcParamsDefault
import numpy as np
%matplotlib inline
plt.rcParams["figure.dpi"]=150
plt.rcParams["figure.facecolor"]="white"

x, y = np.loadtxt('C:/Users/User/QE_calculation/etot_vs_ecutwfc.dat', delimiter=' ', unpack=True)
plt.plot(x, y, "o-", markersize=5, label='Etot vs ecutwfc')
plt.xlabel('ecutwfc (Ry)')
plt.ylabel('Etot (Ry)')
plt.legend(frameon=False)
plt.show()
```



---

5. **Convergence test against the number of k-points**

   pwtk script file name for k-points: si_scf_kpoints.pwtk

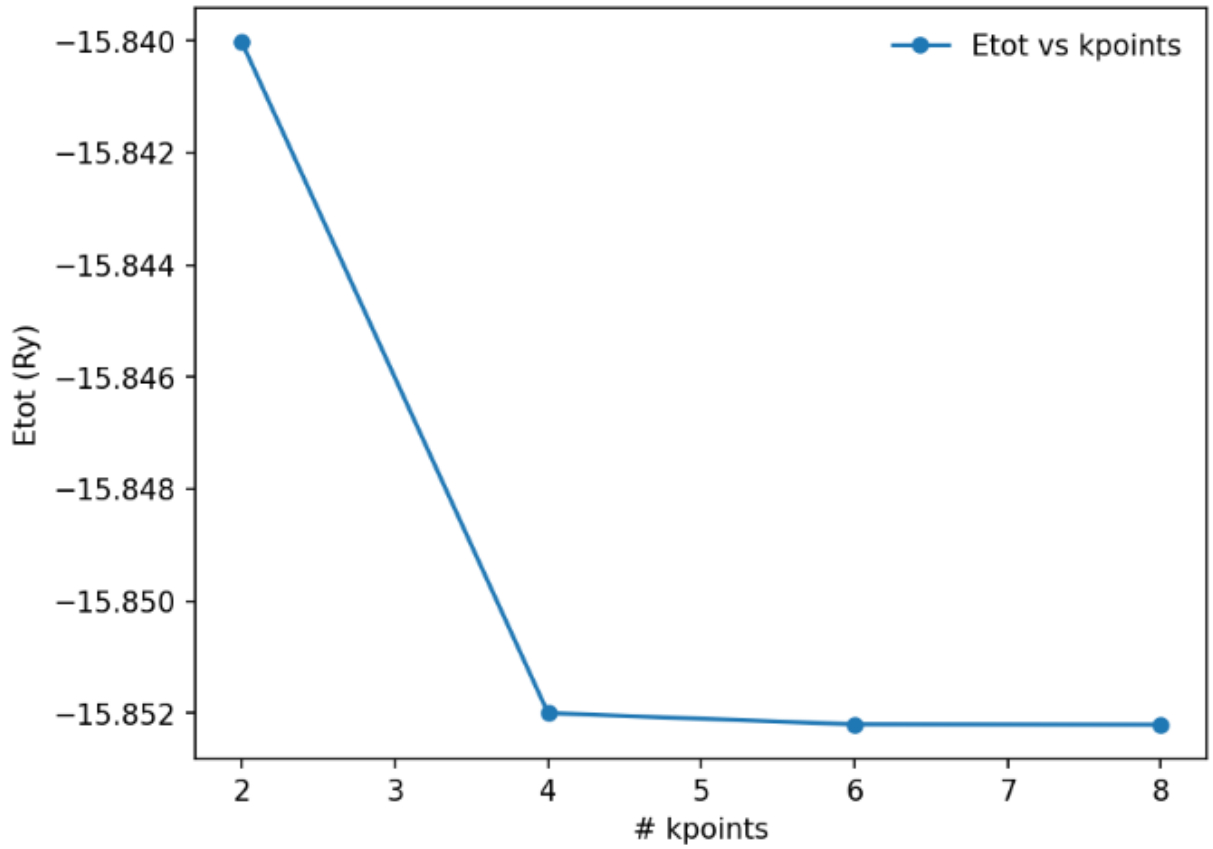
```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_scf_kpoints.pwtk
load_fromPWI silicon.in

set fid [open etot-vs-kpoint.dat w]

foreach k { 2 4 6 8 } {

    set name si_scf_kpoints-$k

    K_POINTS automatic "$k $k $k 1 1 1"
    runPW $name.in

    set Etot [::pwtk::pwo::totene $name.out]
    puts $fid "$k $Etot"
}

close $fid
```

6. **run the script: pwtk si_scf_kpoints.pwtk**
   we'll get: etot-vs-kpoint.dat

---

7. **plot the total energy with respect to ecutwfc using etot_vs_kpoints.dat**

```
x, y = np.loadtxt('C:/Users/User/QE_calculation/etot-vs-kpoint.dat', delimiter=' ', unpack=True)
plt.plot(x, y, "o-", markersize=5, label='Etot vs kpoints')
plt.xlabel('# kpoints')
plt.ylabel('Etot (Ry)')
plt.legend(frameon=False)
plt.show()
```



---

8. **Convergence against lattice constant**
   pwtk script file name for lattice constant:si_scf_alat.pwtk

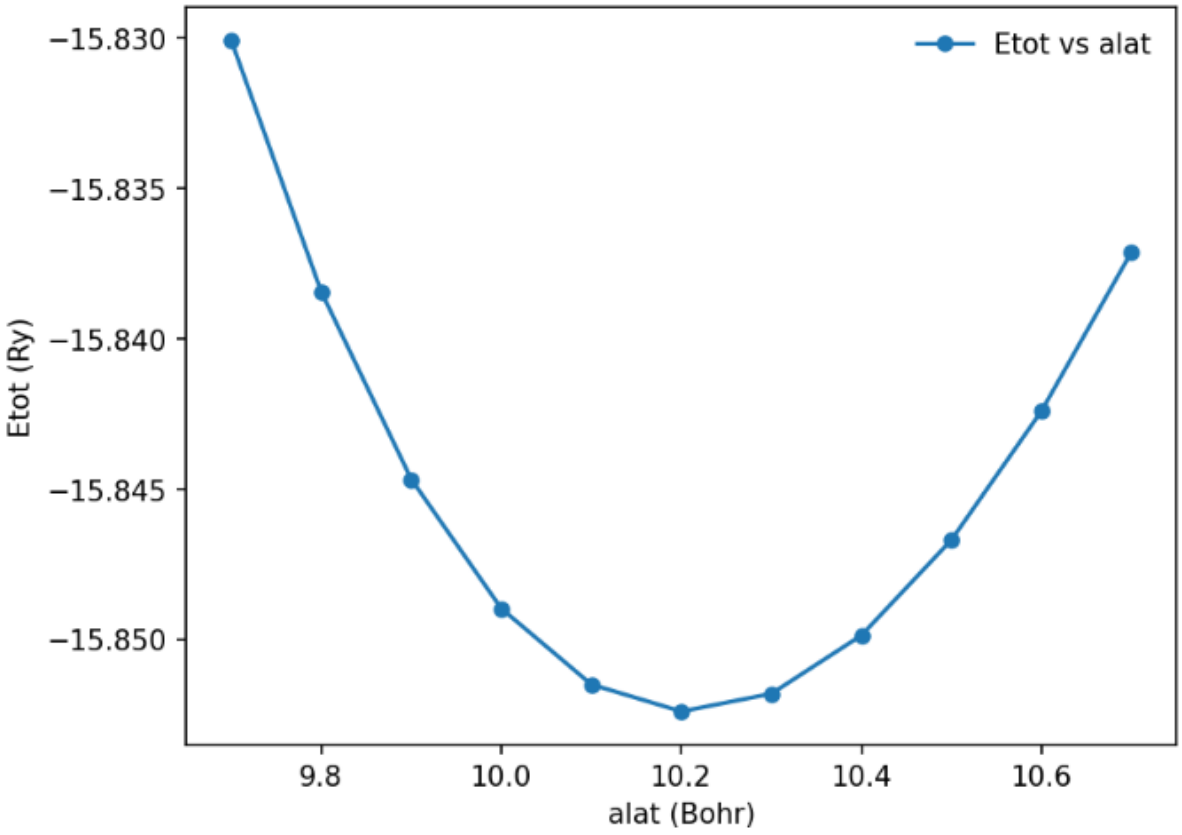
```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_scf_kpoints.pwtk
load_fromPWI silicon.in

set fid [open etot-vs-kpoint.dat w]

foreach k { 2 4 6 8 } {

    set name si_scf_kpoints-$k

    K_POINTS automatic "$k $k $k 1 1 1"
    runPW $name.in

    set Etot [::pwtk::pwo::totene $name.out]
    puts $fid "$k $Etot"
}

close $fid
```

9. **run the script: pwtk si_scf_alat.pwtk**
   we'll get: etot-vs-alat.dat

---

10. **plot the total energy with respect to ecutwfc using etot_vs_alat.dat**

```
x, y = np.loadtxt('C:/Users/User/QE_calculation/etot-vs-alat.dat', delimiter=' ', unpack=True)
plt.plot(x, y, "o-", markersize=5, label='Etot vs alat')
plt.xlabel('alat (Bohr)')
plt.ylabel('Etot (Ry)')
plt.legend(frameon=False)
plt.show()
```



---

11. **(STRUCTURE OPTIMIZATION) vc-relax**
    input file name: si_relax.in

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_relax.in
&CONTROL
calculation = 'vc-relax'
prefix = 'silicon'
outdir = './tmp/'
pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
etot_conv_thr = 1e-5
forc_conv_thr = 1e-4
/

&SYSTEM
ibrav=2,
celldm(1) =10.2,
nat=2,
ntyp=1,
ecutwfc=30
/

&ELECTRONS
conv_thr=1e-8
/

&IONS
/

&CELL
cell_dofree='ibrav'
/

ATOMIC_SPECIES
Si 28.0855 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
Si 0.0 0.0 0.0
Si 0.25 0.25 0.25

K_POINTS (automatic)
6 6 6 1 1 1
```

---

12. **Perform the plane wave calculation using pw.x -inp si_relax.in > si_relax.out.**
we'll get: si_relax.out

```
     Final enthalpy =     -15.8523802650 Ry
Begin final coordinates
     new unit-cell volume =    265.42133 a.u.^3 (    39.33138 Ang^3 )
     density =      2.37149 g/cm^3

CELL_PARAMETERS (alat= 10.20000000)
  -0.500074956   0.000000000   0.500074956
   0.000000000   0.500074956   0.500074956
  -0.500074956   0.500074956   0.000000000

ATOMIC_POSITIONS (alat)
Si       -0.000000000  -0.000000000   0.000000000
Si        0.250037478   0.250037478   0.250037478
End final coordinates
```

Lattice constant is: ((10.2)(0.500074956))/0.5 = 10.2015291024 Bohr.

---

13. **MAKE NEW SCF FILE WITH OPTIMIZED VALUES: ecutoff = 30; celldm(1)= 10.2; k-points: 6 6 6 1 1 1**

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_scf.in
&CONTROL
calculation = 'scf',
prefix = 'silicon',
outdir = './tmp/'
pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
verbosity = 'high'
/

&SYSTEM
ibrav =  2,
celldm(1) = 10.2015,
nat =  2,
ntyp = 1,
ecutwfc = 30,
nbnd = 8
/

&ELECTRONS
/

ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
Si -0.000000000 -0.000000000 0.000000000
Si 0.250037478 0.250037478 0.250037478

K_POINTS (automatic)
6 6 6 1 1 1
```

---

14. **(DENSITY OF STATES CALCULATION) Fixed-ion self consistent field calculation**

create a new input file: si_scf_dos.in (same with scf input file but with increased ecutwfc and kpoints)

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_scf_dos.in
&CONTROL
calculation = 'scf',
restart_mode = 'from_scratch',
prefix = 'silicon',
outdir = './tmp/'
pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
verbosity = 'high'
/

&SYSTEM
ibrav = 2,
celldm(1) = 10.2015,
nat = 2,
ntyp = 1,
ecutwfc = 50,
nbnd = 8
/

&ELECTRONS
conv_thr = 1e-8,
/

ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
Si -0.000000000 -0.000000000 0.000000000
Si 0.250037478 0.250037478 0.250037478

K_POINTS (automatic)
8 8 8 0 0 0
```

15. **Run the calculation using pw.x < si_scf_dos.in > si_scf_dos.out.**
we'll get: si_scf_dos.out

---

16. **Non-self consistent field (nscf) calculation with denser k-point grid**
create a new input file: si_nscf_dos.in (we added occupations in the &system card as tetrahedra (appropriate for DOS calculation and increased the number of k-points to 12 x 12 x 12).

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_nscf_dos.in
&CONTROL
calculation = 'nscf',
restart_mode = 'from_scratch',
prefix = 'silicon',
outdir = './tmp/'
pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
verbosity = 'high'
/

&SYSTEM
ibrav = 2,
celldm(1) = 10.2015,
nat = 2,
ntyp = 1,
ecutwfc = 50,
nbnd = 8,
occupations='tetrahedra'
nosym = .TRUE.
/

&ELECTRONS
conv_thr = 1e-8,
/

ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
Si -0.000000000 -0.000000000 0.000000000
Si 0.250037478 0.250037478 0.250037478

K_POINTS (automatic)
12 12 12 0 0 0
```

17. **Run the calculation using pw.x < si_nscf_dos.in > si_nscf_dos.out.**
we'll get: si_nscf_dos.out

```
the Fermi energy is     6.6638 ev
```

---

18. **Make a DOS input file si_dos.in**

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_dos.in
&DOS
prefix='silicon',
outdir='./tmp/',
fildos='si_dos.dat',
emin=-9.0,
emax=16.0
/
```

19. **Run: dos.x < si_dos.in > si_dos.out**
    we get: si_dos.out and si_dos.dat

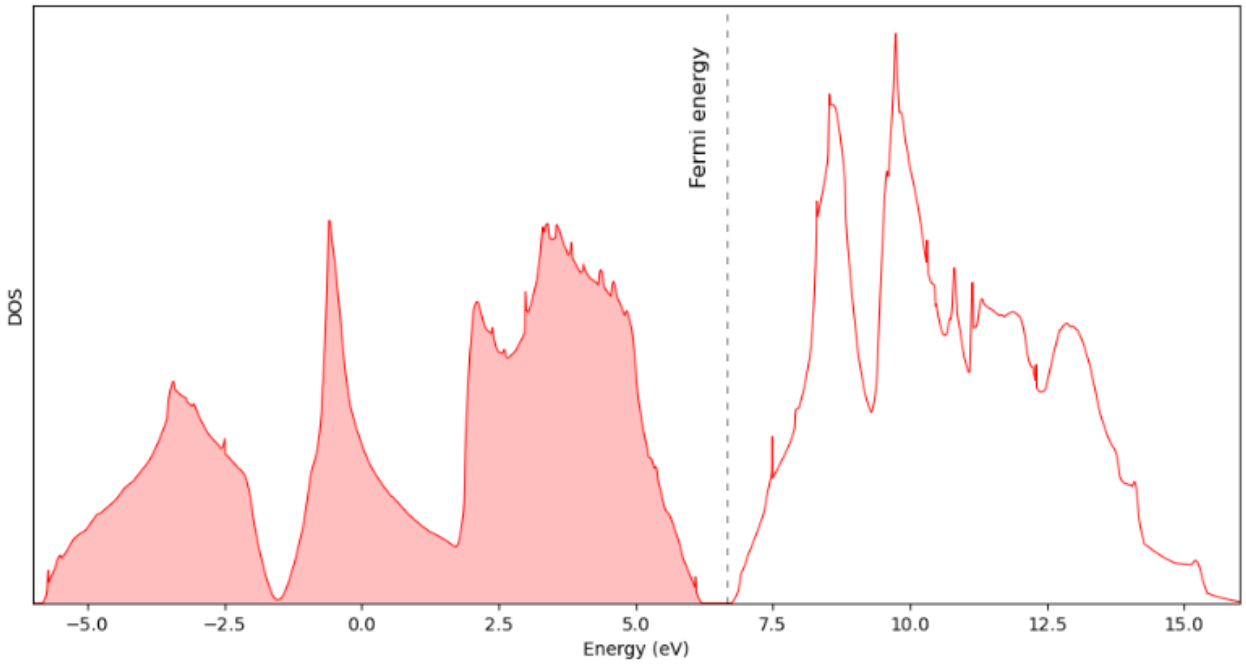
```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_dos.dat
#  E (eV)   dos(E)     Int dos(E) EFermi =    6.664 eV
```

(Check Fermi energy either in nscf output or here)

---

20. **plot the DOS using si_dos.dat**

```python
import matplotlib.pyplot as plt
from matplotlib import rcParamsDefault
import numpy as np
%matplotlib inline

# load data
energy, dos, idos = np.loadtxt('C:/Users/User/QE_calculation/si_dos.dat', unpack=True)

# make plot
plt.figure(figsize = (12, 6))
plt.plot(energy, dos, linewidth=0.75, color='red')
plt.yticks([])
plt.xlabel('Energy (eV)')
plt.ylabel('DOS')
plt.axvline(x=6.6638, linewidth=0.5, color='k', linestyle=(0, (8, 10)))
plt.xlim(-6, 16)
plt.ylim(0, )
plt.fill_between(energy, 0, dos, where=(energy < 6.66638), facecolor='red', alpha=0.25)
plt.text(6, 1.7, 'Fermi energy', fontsize= 12, rotation=90)
plt.show()
```



---

21. **(BANDSTRUCTURE CALCULATION)** Single-point self consistent field calculation

make a new input file (scf file but with new parameters): si_bands_scf.in

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_bands_scf.in
&CONTROL
  calculation = 'scf',
  restart_mode = 'from_scratch',
  prefix = 'silicon',
  outdir = './tmp/'
  pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
  verbosity = 'high'
/

&SYSTEM
  ibrav =  2,
  celldm(1) = 10.2015,
  nat =  2,
  ntyp = 1,
  ecutwfc = 50,
  ecutrho = 400,
  nbnd = 8,
!  occupations = 'smearing',
!  smearing = 'gaussian',
!  degauss = 0.005
/

&ELECTRONS
  conv_thr = 1e-8,
/

ATOMIC_SPECIES
  Si 28.086 Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
  Si -0.000000000 -0.000000000 0.000000000
  Si 0.250037478 0.250037478 0.250037478

K_POINTS (automatic)
  8 8 8 0 0 0
```

22. **Run the calculation using pw.x < si_bands_scf.in > si_bands_scf.out.**
we'll get: si_bands_scf.out

---

23. **Band calculation (non-self consistent field) calculation**
make a new input file: si_bands.in

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_bands.in
&control
  calculation = 'bands',
  restart_mode = 'from_scratch',
  prefix = 'silicon',
  outdir = './tmp/'
  pseudo_dir = '/home/meyn/q-e-qe-7.1/pseudo/'
  verbosity = 'high'
/

&system
  ibrav =  2,
  celldm(1) = 10.2015,
  nat =  2,
  ntyp = 1,
  ecutwfc = 50,
  ecutrho = 400,
  nbnd = 8
 /

&electrons
  conv_thr = 1e-8,
 /

ATOMIC_SPECIES
  Si 28.086  Si.pz-vbc.UPF

ATOMIC_POSITIONS (alat)
  Si -0.000000000 -0.000000000 -0.000000000
  Si 0.250037478 0.250037478 0.250037478

K_POINTS {crystal_b}
5
  0.0000 0.5000 0.0000 20  !L
  0.0000 0.0000 0.0000 30  !G
  -0.500 0.0000 -0.500 10  !X
  -0.375 0.2500 -0.375 30  !U
  0.0000 0.0000 0.0000 20  !G
```

24. **Run pw.x with bands calculation input file using pw.x < si_bands.in > si_bands.out**

we'll get: si_bands.out

---

25. **Postprocessing using band.x: si_bands_pp.in**

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ cat si_bands_pp.in
&BANDS
  prefix = 'silicon'
  outdir = './tmp/'
  filband = 'si_bands.dat'
/
```

26. **Run bands.x from post processing (PP) module: bands.x < si_bands_pp.in > si_bands_pp.out**

we'll get: si_bands_pp.out

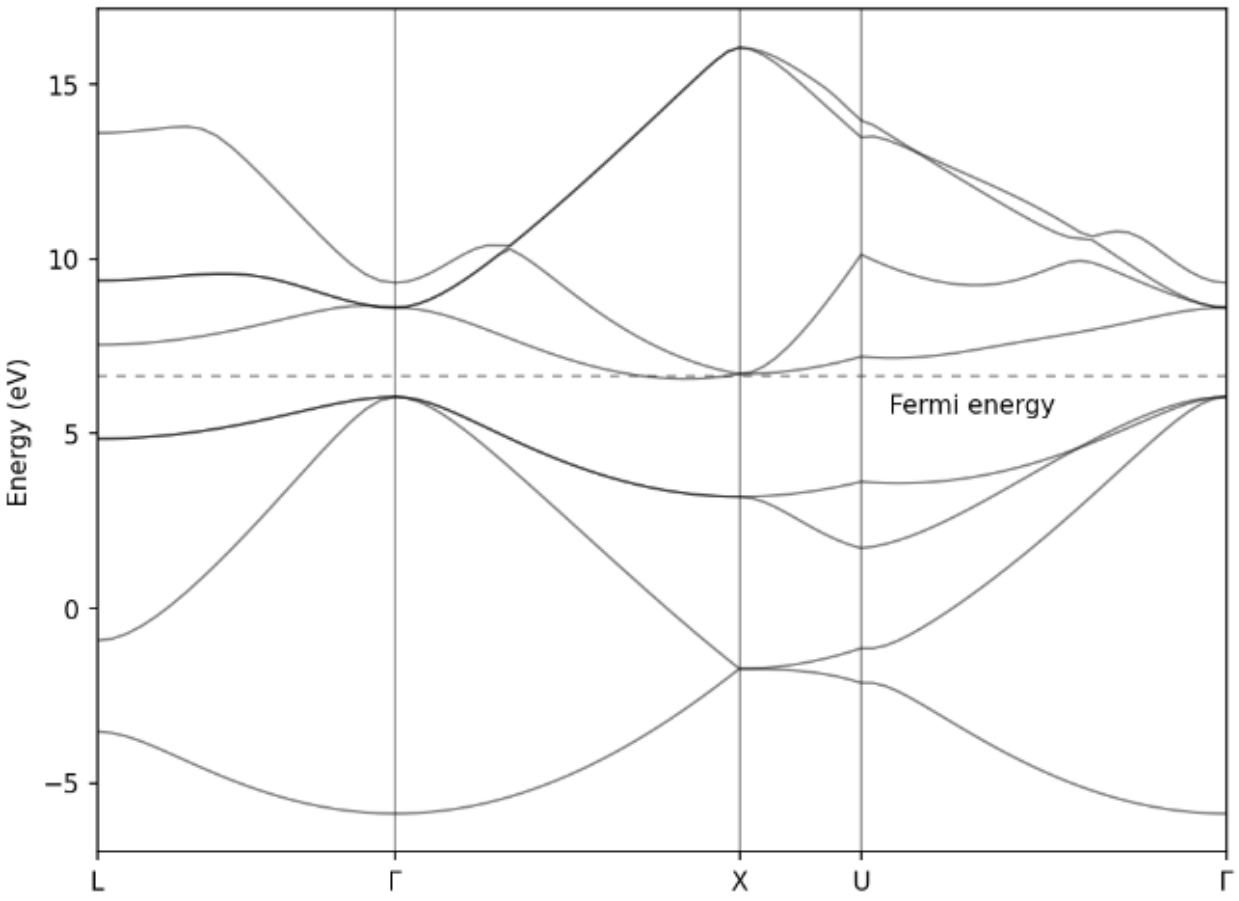
27. **Run plotband.x to visualize bandstructure**

```
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ plotband.x
     Input file > si_bands.ps
STOP Error reading file header
meyn@LAPTOP-C0A2DPIS:~/q-e-qe-7.1/test_calculation$ plotband.x
     Input file > si_bands.dat
Reading    8 bands at     91 k-points
Range:    -5.8300   16.3420eV  Emin, Emax > -6, 16
high-symmetry point:  0.5000 0.5000 0.5000   x coordinate   0.0000
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   0.8660
high-symmetry point:  1.0000 0.0000 0.0000   x coordinate   1.8660
high-symmetry point:  1.0000 0.2500 0.2500   x coordinate   2.2196
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   3.2802
output file (gnuplot/xmgr) > si_bands.gnuplot
bands in gnuplot/xmgr format written to file si_bands.gnuplot

output file (ps) > si_bands.ps
Efermi > 6.6416
deltaE, reference E (for tics) 4, 0
bands in PostScript format written to file si_bands.ps
```

OR

plot using python:

```python
import matplotlib.pyplot as plt
from matplotlib import rcParamsDefault
import numpy as np
%matplotlib inline

plt.rcParams["figure.dpi"]=150
plt.rcParams["figure.facecolor"]="white"
plt.rcParams["figure.figsize"]=(8, 6)

# Load data
data = np.loadtxt('C:/Users/User/QE_calculation/si_bands.dat.gnu')

k = np.unique(data[:, 0])
bands = np.reshape(data[:, 1], (-1, len(k)))

for band in range(len(bands)):
    plt.plot(k, bands[band, :], linewidth=1, alpha=0.5, color='k')
plt.xlim(min(k), max(k))

# Fermi energy
plt.axhline(6.6638, linestyle=(0, (5, 5)), linewidth=0.75, color='k', alpha=0.5)
# High symmetry k-points (check bands_pp.out)
plt.axvline(0.8660, linewidth=0.75, color='k', alpha=0.5)
plt.axvline(1.8660, linewidth=0.75, color='k', alpha=0.5)
plt.axvline(2.2196, linewidth=0.75, color='k', alpha=0.5)
# text labels
plt.xticks(ticks= [0, 0.8660, 1.8660, 2.2196, 3.2802], \
           labels=['L', '$\Gamma$', 'X', 'U', '$\Gamma$'])
plt.ylabel("Energy (eV)")
plt.text(2.3, 5.6, 'Fermi energy', fontsize= 10)
plt.show()
```


15
10
5
0
−5
Energy (eV)
Fermi energy
L
Γ
X
U
Γ