Quantum-ESPRESSO HANDS-ON TUTORIAL

Structural optimizations and NEB

Nicola Bonini (MIT)

Santa Barbara, 07/2009

Outline

Exercise 1 : geometry optimization of an “isolated” molecule

Exercise 2 : variable-cell relaxation for As bulk under pressure

Exercise 3 : NEB calculation on Si addimer on Si(001)

Exercise 4 : surface relaxation in Al(001)

BFGS: input variables

A detailed explanation of all the keywords can be found in the file Doc/INPUT_PW.

```
&CONTROL
  calculation = “relax”   <=  mandatory
  ...
  nstep                   <=  optional (50)
  ...
/
...
...
&IONS
  upscale                 <= optional (10.D0),
  bfbs_ndim               <= optional (1),
  trust_radius_ini        <= optional (0.5)
  trust_radius_min        <= optional (0.001)
  trust_radius_max        <= optional (0.8)
  w1                      <= optional (0.01),
  w2                      <= optional (0.5),
  pot_extrapolation       <= optional (atomic),
  wfc_extrapolation       <= optional (none),
/
```

Exercise 1: Structural optimization of a C_2H_2 molecule

We want to use pw.x to find the optimized geometry of an acetylene (C_2H_2) molecule.

Let us suppose we have the following guess for the structure (coordinates in atomic units):

```
C       0.000000000    0.000000000    0.000000000
C       2.100000000    0.000000000    0.000000000
H      -1.600000000    0.400000000    0.000000000
H       3.600000000   -0.400000000    0.000000000
```

pw.x adopts periodic boundary conditions.
We can put the molecule in a large box, whose size is a convergence parameter. We choose a cube of side length 12 a.u.

1) Visualize the structure with XcrySDen

```
prompt> xcrysden --pwi acetylene.scf.in
```

2) Run the SCF calculation and check the forces on the atoms.

```
prompt> pw.x < acetylene.scf.in > acetylene.scf.out
```

Exercise 1: Structural optimization of a C_2H_2 molecule

Input file:

```
 &CONTROL
                 calculation = 'scf' ,
                restart_mode = 'from_scratch' ,
                   tprnfor   = .true.
```

$\leftarrow$ Print forces

```
 /
 &SYSTEM
                       ibrav = 1,
```

$\leftarrow$ Simple cubic unit cell

```
                   celldm(1) = 12.D0,
                         nat = 4,
                        ntyp = 2,
                     ecutwfc = 25.0D0 ,
                     ecutrho = 160.0D0 ,
                       nosym = .true. ,
                 occupations = 'smearing' ,
```

$\leftarrow$ But it is not a metal !?

```
                     degauss = 0.005D0 ,
 /
 &ELECTRONS
                    conv_thr = 1.D-8 ,
                 mixing_beta = 0.5D0 ,
 /
ATOMIC_SPECIES
    H    1.00000  H.US_PBE.RRKJ3.UPF
    C   12.00000  C.US_PBE.RRKJ3.UPF
ATOMIC_POSITIONS bohr
    C     0.000000000   0.000000000   0.000000000   0  0  0
    C     2.100000000   0.000000000   0.000000000   1  1  1
    H    -1.600000000   0.400000000   0.000000000   1  1  1
    H     3.600000000  -0.400000000   0.000000000   1  1  1
K_POINTS gamma
```

Only the position of the first atom is fixed

$\leftarrow$ Gamma point only (sufficient for a molecule in a box)

Exercise 1: Structural optimization of a C_2H_2 molecule

From the output file:

```
Forces acting on atoms (Ry/au):

atom    1 type  2   force =     0.04228280   -0.07117323    0.00000000
atom    2 type  2   force =    -0.29088976    0.14204786    0.00000000
atom    3 type  1   force =    -0.50881449    0.10564514    0.00000000
atom    4 type  1   force =     0.75742145   -0.17651978    0.00000000

Total force =     0.989795     Total SCF correction =    0.000014
```

PAY ATTENTION !

This measures the error on forces given by a non perfect self-consistency. It has to be much smaller than the force itself.

Connected to `conv_thr`.

Forces on atoms are quite large (you can visualize them with xcrysden).
We now optimize the structure in two ways:

A) **BFGS minimization**

B) **damped molecular dynamics**

Copy the input file for SCF to a new file, and open it with an editor:

```
prompt> cp acetylene.scf.in acetylene.bfgs.in
```

Exercise 1: Structural optimization of a C_2H_2 molecule

BFGS minimization: input file

Open `acetylene.bfgs.in` with an editor and:

1) Specify the following variable in the “Control” namelist:

```
calculation  = "relax",
```

2) Add “Ions” namelist:

```
&IONS
   ion_dynamics        = "bfgs",
   pot_extrapolation = "second_order",
   wfc_extrapolation = "second_order",
   upscale             = 100,
 /
```

3) Set a variable SCF threshold (100 times more accurate closer to the minimum):
   In “Electrons” namelist

```
         conv_thr = 1.D-6 ,
```

   In “Ions” namelist

```
         upscale = 100 ,
```

4) Run the calculation

```
prompt> pw.x < acetylene.bfgs.in > acetylene.bfgs.out
```

Exercise 1: Structural optimization of a C_2H_2 molecule

BFGS minimization: output file

1) Scroll the file:

   `less acetylene.bfgs.out` (press q to exit)

2) Extract key quantities:

   total energy: `grep ^! acetylene.bfgs.out | nl`

   total force: `grep "Total force" acetylene.bfgs.out | nl`

3) Use XCrySDen to visualize the dynamics:

   `xcrysden --pwo acetylene.bfgs.out`

   (remember to select “reduce dimension to 0D” and “Display All Coordinates as Animation”; type “f” to visualize the forces)

At convergence, forces are smaller than the specified (or default) threshold, and the algorithm stops:

```
Forces acting on atoms (Ry/au):

atom    1 type  2   force =     0.00006227   -0.00039094    0.00000000
atom    2 type  2   force =     0.00007045    0.00017243    0.00000000
atom    3 type  1   force =    -0.00011897   -0.00012610    0.00000000
atom    4 type  1   force =    -0.00001374    0.00034461    0.00000000

Total force =     0.000429     Total SCF correction =     0.000013

bfgs converged in  20 scf cycles and  19 bfgs steps
```

Exercise 1: Structural optimization of a C_2H_2 molecule

Accuracy of the electronic structure calculation

Try to perform a BFGS relaxation with a lousy calculation of the electronic structure.

1) Copy the input file for BFGS calculation to a new file, and open it with an editor:

```
cp acetylene.bfgs.in acetylene.bfgs-test.in
```

2) Set a SCF threshold of 1mRy (within “chemical accuracy”!)

   In “Electrons” namelist

```
        conv_thr = 1.D-3 ,
```

   In from “Ions” namelist

```
        upscale = 1 ,
```

3) Run the calculation

```
pw.x < acetylene.bfgs-test.in > acetylene.bfgs-test.out
```

Forces are so poorly described that the algorithm is unable to converge!

Exercise 1: Structural optimization of a C_2H_2 molecule

Damped molecular dynamics: input file

Copy the input file for SCF to a new file, and open it with an editor:

```
cp acetylene.scf.in acetylene.damp.in
```

1) Specify the following variables in the “Control” namelist:

```
calculation  = "relax",
dt           = 20.D0,
```

2) Set a strict SCF threshold (“Electrons” namelist):

```
conv_thr    = 1.D-8,
```

3) Add “Ions” namelist:

```
&IONS
  ion_dynamics      = "damp",
  pot_extrapolation = "second_order",
  wfc_extrapolation = "second_order",
/
```

← these extrapolations make the SCF loop shorter

4) Set equal masses (we are not interested in a real dynamics!)

```
H  1.0  H.US_PBE.RRKJ3.UPF
C  1.0  C.US_PBE.RRKJ3.UPF
```

5) Save and run:

```
pw.x < acetylene.damp.in > acetylene.damp.out
```

Exercise 1: Structural optimization of a C_2H_2 molecule

Damped molecular dynamics: output file

1) Scroll the file:

```
less acetylene.damp.out
```
(press q to exit)

2) Extract key quantities:

total energy: `grep ! acetylene.damp.out | nl`

total force: `grep "Total force" acetylene.damp.out | nl`

3) Use XCrySDen to visualize the dynamics:

```
xcrysden --pwo acetylene.damp.out
```

(remember to select “reduce dimension to 0D” and “Display All Coordinates as Animation”; type “f” to visualize the forces)

At convergence, forces are smaller than the specified (or default) threshold, and the algorithm stops:

```
Forces acting on atoms (Ry/au):

atom    1 type  2   force =    -0.00040916   -0.00032040    0.00000000
atom    2 type  2   force =     0.00036937    0.00021942    0.00000000
atom    3 type  1   force =    -0.00003031   -0.00019599    0.00000000
atom    4 type  1   force =     0.00007010    0.00029698    0.00000000

Total force =     0.000563     Total SCF correction =     0.000036
```

Exercise 2: Variable-cell relaxation

As bulk input file: As0.bfgs.in

```
 &CONTROL
    calculation = "vc-relax",
    etot_conv_thr = 1.0E-4, forc_conv_thr = 1.0D-3,
    nstep = 50,
  /
  &SYSTEM
    ibrav = 0,
    A = 3.85,
    nat= 2, ntyp= 1, nbnd = 9, nelec = 10,
    occupations = 'smearing', smearing = 'mp', degauss = 0.005,
    ecutwfc = 30.0, /
  &ELECTRONS
    conv_thr = 1.0d-7,
  /
  &IONS
  /
  &CELL
    Press = 0.0,
  /
CELL_PARAMETERS hexagonal
   0.58012956  0.00000000  0.81452422
  -0.29006459  0.50240689  0.81452422
  -0.29006459 -0.50240689  0.81452422
ATOMIC_SPECIES
As   74.90000  As.pz-bhs.UPF
ATOMIC_POSITIONS crystal
As     0.2750        0.2750        0.2750
As    -0.2750       -0.2750       -0.2750
K_POINTS automatic
  4 4 4   1 1 1
```

ibrav = 0 ← "Free" lattice

A = 3.85 ← Crystallographic constant "a"

Press = 0.0 ← Target pressure (KBar = 0.1GPa)
[**cell_dynamics = 'bfgs'** (default)]

CELL_PARAMETERS ← Lattice vectors

ATOMIC_POSITIONS ← Atomic positions

Exercise 2: Variable-cell relaxation

As bulk input file: As0.damp.in

```
 &CONTROL
    calculation = "vc-relax",
    etot_conv_thr = 1.0E-4, forc_conv_thr = 1.0D-3,
    nstep = 50, dt = 100,
  /
  &SYSTEM
    ibrav = 0,
    A = 3.85,
    nat= 2, ntyp= 1, nbnd = 9, nelec = 10,
    occupations = 'smearing', smearing = 'mp', degauss = 0.005,
    ecutwfc = 30.0, /
  &ELECTRONS
    conv_thr = 1.0d-7, /
  &IONS /
  &CELL
!    cell_dynamics = 'damp-w' , wmass =  0.00150000  ,
    press = 0.0, /
CELL_PARAMETERS hexagonal
    0.58012956  0.00000000  0.81452422
   -0.29006459  0.50240689  0.81452422
   -0.29006459 -0.50240689  0.81452422
ATOMIC_SPECIES
As   74.90000  As.pz-bhs.UPF
ATOMIC_POSITIONS crystal
As    0.2750        0.2750        0.2750
As   -0.2750       -0.2750       -0.2750
K_POINTS automatic
  4 4 4   1 1 1
```

Damped dynamics & fictitious cell mass

Exercise 2: Variable-cell relaxation

1) Run the vc-relax calculation and check the output. What is the final lattice constant?

```
prompt> pw.x < As0.bfgs.in > As0.bfgs.out
```

In the output file after each iteration the code writes:

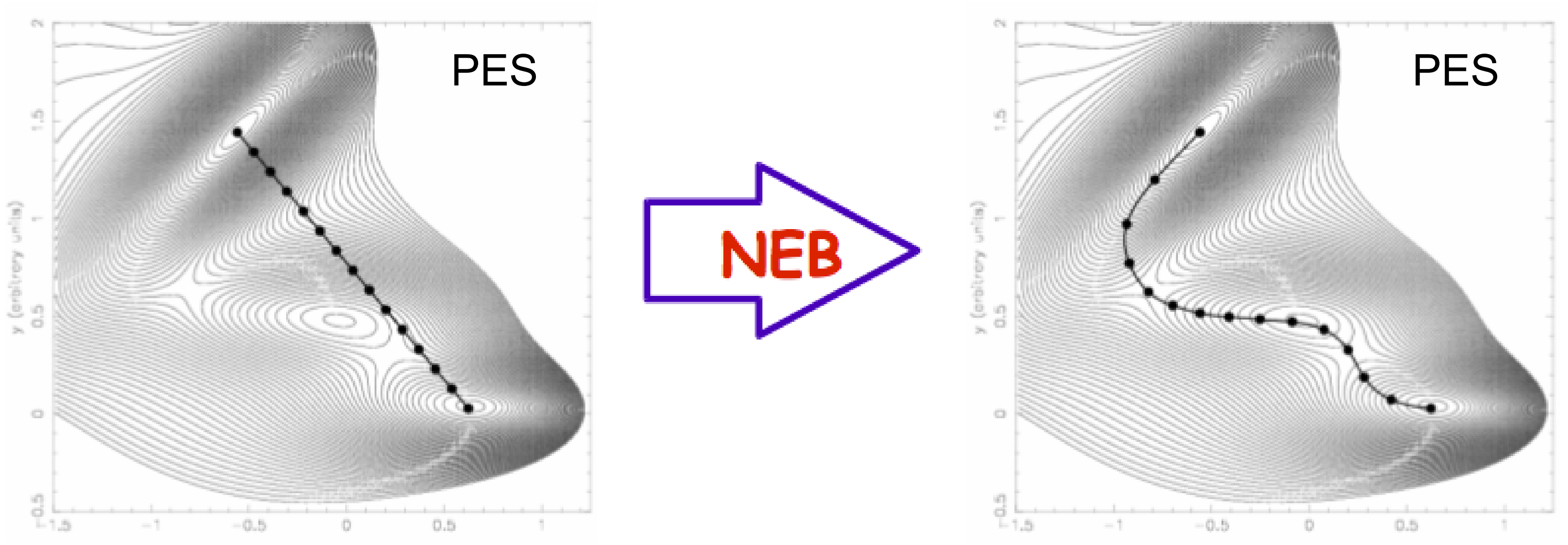
Pressure

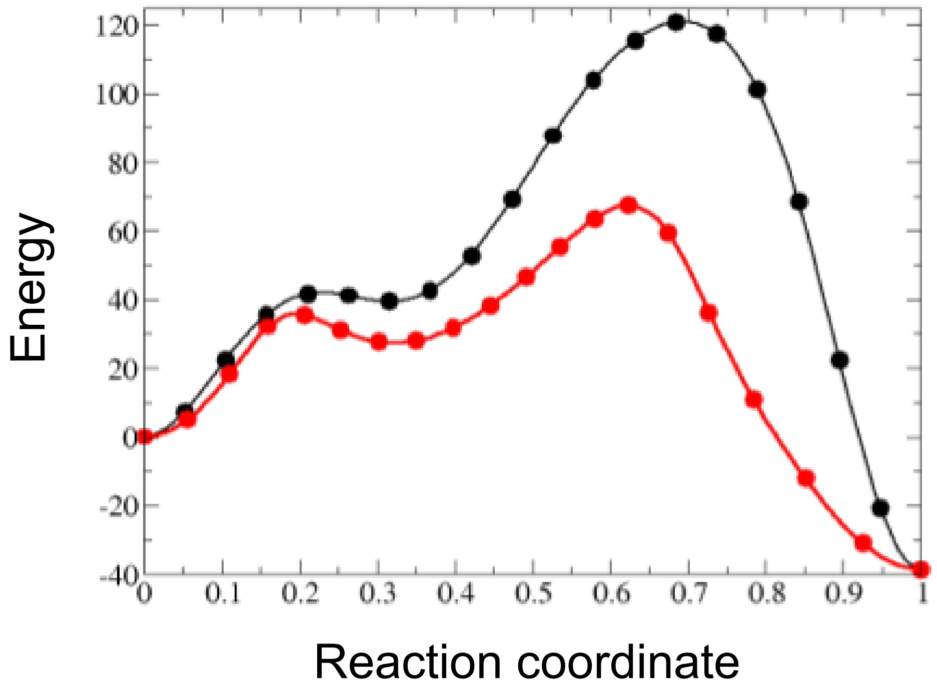
```
...
       total   stress  (Ry/bohr**3)                   (kbar)     P=     1.29
0.00000443   0.00000000   0.00000001            0.65       0.00       0.00
0.00000000   0.00000445   0.00000000            0.00       0.65       0.00
0.00000001   0.00000000   0.00001734            0.00       0.00       2.55

...

CELL_PARAMETERS (alat)
   0.571326124   0.000000000   0.836077307
  -0.285663594   0.494782125   0.836077799
  -0.285663594  -0.494782125   0.836077799

ATOMIC_POSITIONS (crystal)
As        0.271892705   0.271893060   0.271893060
As       -0.271892705  -0.271893060  -0.271893060
```

$\leftarrow$ New lattice parameters [in units of “a”]

2) Change the target pressure to 40 Gpa (400 Kbar) and run the vc-relax calculation. What is the final structure?

Exercise 3: NEB calculation on Si addimer on Si(001)





Exercise 3: NEB calculation on Si addimer on Si(001)

A detailed explanation of all the keywords can be found in the file Doc/INPUT_PW.

```
&CONTROL
  calculation = “neb”  <=  mandatory
  ...
  nstep                <=  optional (0)
  ...
/
...
...
&IONS
  num_of_images        <=  mandatory
  CI_scheme            <=  optional (no-CI) ⟵ Climbing image
  opt_scheme           <=  optional (quick-min)
  ds                   <=  optional (1.5)
  first_last_opt       <=  optional (.FALSE.)
  k_max                <=  optional (0.1)
  k_min                <=  optional (0.1)  ⟵ Variable elastic constants
  path_thr             <=  optional (0.05)
  ...
/
```

Exercise 3: NEB calculation on Si addimer on Si(001)

A detailed explanation of all the keywords can be found in the file Doc/INPUT_PW.

```
first_image                                                <=  mandatory
   X 0.0  0.0  0.0  { if_pos(1) if_pos(2) if_pos(3) }
   Y 0.5  0.0  0.0  { if_pos(1) if_pos(2) if_pos(3) }
   Z 0.0  0.2  0.2  { if_pos(1) if_pos(2) if_pos(3) }
intermediate_image 1                                       <=  optional
   X 0.0  0.0  0.0
   Y 0.9  0.0  0.0
   Z 0.0  0.2  0.2
intermediate_image ...                                     <=  optional
   X 0.0  0.0  0.0
   Y 0.9  0.0  0.0
   Z 0.0  0.2  0.2
last_image                                                 <=  mandatory
   X 0.0  0.0  0.0
   Y 0.7  0.0  0.0
   Z 0.0  0.5  0.2
```

Exercise 3: NEB calculation on Si addimer on Si(001)

Output files (files in the working directory):

`prefix.path`	<=	file containing data required to restart a NEB calculation
`prefix.axsf`	<=	file containing the path in xcrysden format
`prefix.xyz`	<=	file containing the path in xyz format
`prefix.dat`	<=	file containing the reaction coordinate, the energy and the error of each image
`prefix.int`	<=	file containing a cubic interpolation for the energy profile

Exercise 3: NEB calculation on Si addimer on Si(001)

Use xcrysden to visualize the initial guess for the reaction path:

```
    prompt> xcrsden –pwi Si-addimer.neb.in
```

Run the calculation:

```
    prompt> pw.x < Si-addimer.neb.in > Si-addimer.neb.out
```

Check the reaction barrier:

```
    prompt> grep “activation energy \(->\)” Si_addimer.neb.out
```

Plot the current energy profile (files .dat and .int):

```
    prompt> gnuplot plot_path.gnu
```
(it creates an eps figure)

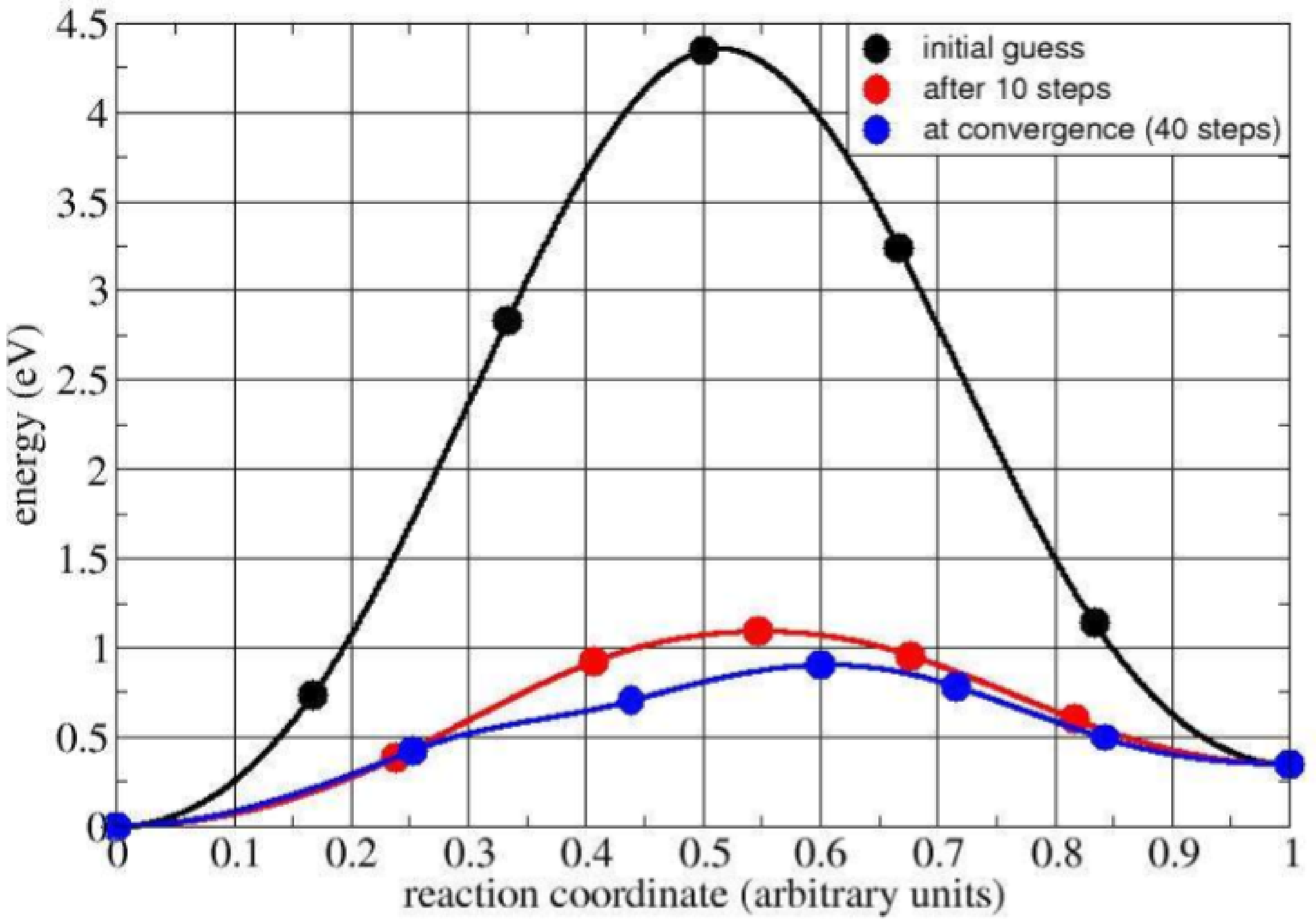
Stop the run after 10 iterations by creating in the working directory the empty file `Si_addimer.EXIT`:

```
    prompt> touch Si-addimer.EXIT
```

Modify the file `Si-addimer.neb.in` by activating a climbing image and by adding restart_mode=”restart” in the &CONTROL namelist, and rerun the job:

```
    prompt> pw.x < Si-addimer.neb.in >> Si-addimer.neb.out
```

Exercise 3: NEB calculation on Si addimer on Si(001)



Exercise 4: Al(001) surface relaxation

- ✔ Use the `al001-scf.in` input file to run a self-consistent calculation for the *Al (001)* slab:

```
prompt> pw.x < al001-scf.in > al001-scf.out
```

- ✔ Scroll to the end of the output file and analyse the forces: what do you notice? Are the atomic forces pointing outward or inward?

- ✔ Modify the input file so that to perform a structural relaxation with the BFGS method and save the new input as `al001-BFGS.in`.

- ✔ Run the structural relaxation:

```
prompt> pw.x < al001-BFGS.in > al001-BFGS.out
```

- ✔ Visualize the output of the relaxation using xcrysden:

```
prompt> xcrysden --pwo al001-BFGS.out
```