

Forces and structural optimizations

Quantum-ESPRESSO tutorial

AIMS, July 2008

Hellmann-Feynman forces

In the Born-Oppenheimer approximation the total energy $E[R_i^{3N}, \rho(R_i^{3N})]$ is a function of ionic coordinates R_i^{3N} and defines a 3N-dimensional hyper-surface, called Potential Energy Surface (PES).

The forces acting on the ions are given by (minus) the gradient of the total energy. Using Hellmann-Feynman theorem:

$$\mathbf{F}_I = - \frac{\partial E(\mathbf{R})}{\partial \mathbf{R}_I} = - \left\langle \Psi(\mathbf{R}) \left| \frac{\partial H_{BO}(\mathbf{R})}{\partial \mathbf{R}_I} \right| \Psi(\mathbf{R}) \right\rangle$$

$$\mathbf{F}_I = - \int n_{\mathbf{R}}(\mathbf{r}) \frac{\partial V_{\mathbf{R}}(\mathbf{r})}{\partial \mathbf{R}_I} d\mathbf{r} - \frac{\partial E_N(\mathbf{R})}{\partial \mathbf{R}_I}$$

where the electron-nucleus interaction and the electrostatic ion-ion interaction,

$$V_{\mathbf{R}}(\mathbf{r}) = - \sum_I \frac{Z_I e^2}{|\mathbf{r} - \mathbf{R}_I|} \quad E_N(\mathbf{R}) = \frac{e^2}{2} \sum_{I \neq J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}$$

depend only on the ionic positions

Hellmann-Feynman forces

When using plane waves no corrections are needed to the previous formula:

$$\mathbf{F}_I = - \int n_{\mathbf{R}}(\mathbf{r}) \frac{\partial V_{\mathbf{R}}(\mathbf{r})}{\partial \mathbf{R}_I} d\mathbf{r} - \frac{\partial E_N(\mathbf{R})}{\partial \mathbf{R}_I}$$

The evaluation of forces is then a cheap byproduct of the electronic structure calculation.

The quality of forces depends on the quality of the electronic structure calculation!

From the forces:

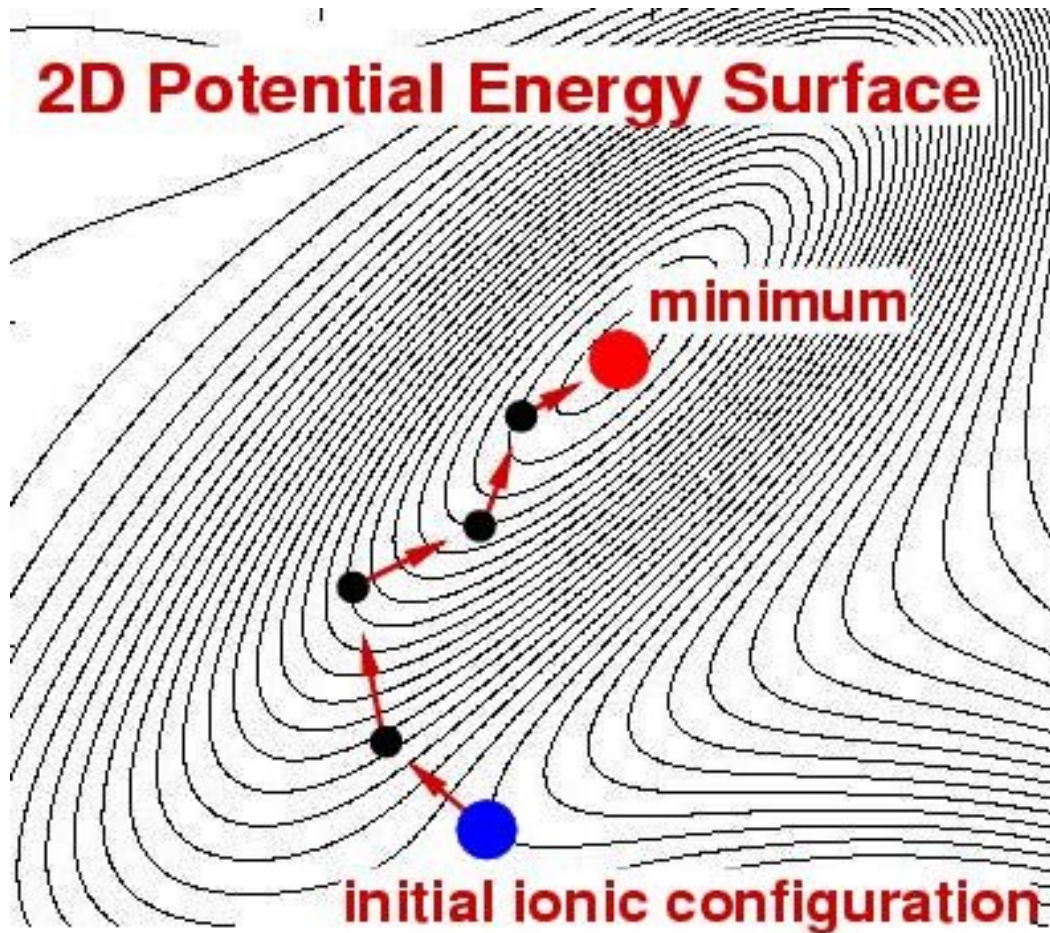
- structural optimization. Equilibrium condition given by

$$\mathbf{F}_I \equiv - \frac{\partial E(\mathbf{R})}{\partial \mathbf{R}_I} = 0$$

- molecular dynamics
- higher-order derivatives (phonons, ...)

Structural optimizations

Several algorithms for searching an equilibrium configuration, close to the initial ionic configuration (a local minimum of the PES). For example:



Steepest Descent optimisation

$$\dot{X}_I^{3N} = F(X_I^{3N})$$

discretisation

$$X_{k+1} = X_k - \lambda \frac{g_k}{|g_k|}$$

gradient unitary vector

Quasi-Newton ionic relaxation

the Broyden Fletcher Goldfarb Shanno algorithm

Taylor expansion of the energy around a stationary point (X_{k+1}):

$$s_k = X_{k+1} - X_k$$

Hessian matrix

$$E(X_{k+1}) - E(X_k) = g(X_k) \cdot s_k + \frac{1}{2} s_k^T H_k s_k$$

gradient vector

Quasi-Newton ionic relaxation

the Broyden Fletcher Goldfarb Shanno algorithm

Equivalently, for the gradient vector we have:

$$g(X_{k+1}) - g(X_k) = H_k s_k$$

Stationary condition:

$$\|g(X_{k+1})\|_{\infty} = 0$$



Newton-Raphson step:

$$s_k^{NR} = -H_k^{-1} g_k$$

Quasi-Newton ionic relaxation

the Broyden Fletcher Goldfarb Shanno algorithm

$$X_{k+1} = X_k + \overset{\text{trust radius}}{T_k^L} \frac{s_k^{NR}}{|s_k^{NR}|} \quad \leftarrow s_k^{NR} = -H_k^{-1} g_k$$

The inverse Hessian matrix is updated using the BFGS scheme:

$$H_{k+1}^{-1} = H_k^{-1} + \left(1 + \frac{\gamma_k^T H_k^{-1} \gamma_k}{s_k^T \gamma_k} \right) \frac{s_k s_k^T}{s_k^T \gamma_k} - \left(\frac{s_k \gamma_k^T H_k^{-1} + H_k^{-1} \gamma_k s_k^T}{s_k^T \gamma_k} \right)$$

$$\gamma_k = g_{k+1} - g_k$$

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)
  w_1              <=  optional (0.01),
  w_2              <=  optional (0.5),
  pot_extrapolation <=  optional (atomic),
  wfc_extrapolation <=  optional (none),
/
```

Acetylene molecule (C₂H₂)

We want to use `pw.x` to find the optimized geometry of acetylene.

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

```
C      0.0000000000    0.0000000000    0.0000000000
C      2.1000000000    0.0000000000    0.0000000000
H     -1.6000000000    0.4000000000    0.0000000000
H      3.6000000000   -0.4000000000    0.0000000000
```

`pw.x` adopts periodic boundary conditions.

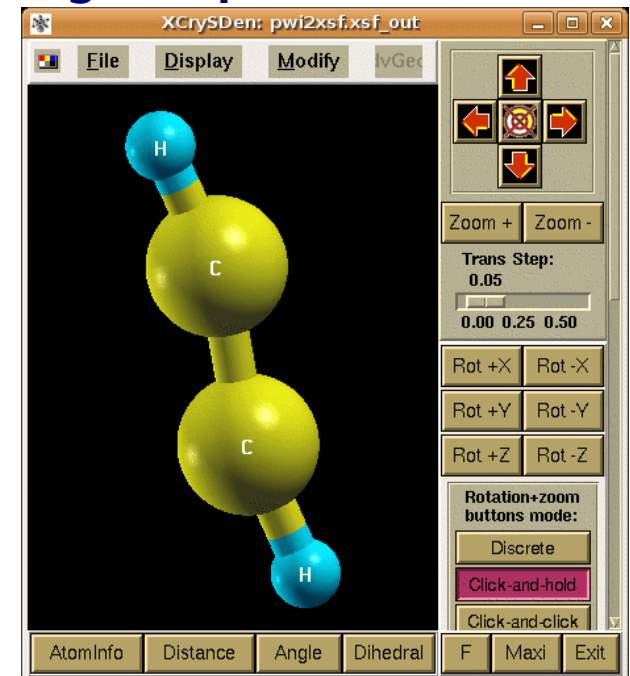
We can put the molecule in a large box, whose size is a convergence parameter.

We choose a cube of side length 12 a.u. for quickness.

1) Prepare the input file for the SCF calculation
(see next slide for other details), use PWgui if you prefer

2) Visualize the structure with XcrySden
`xcrysdn --pwi acetylene.scf.in`

3) Run the SCF calculation and look at forces on atoms.
`pw.x < acetylene.scf.in > acetylene.scf.out`



C₂H₂ SCF: the input

&CONTROL

```
calculation = 'scf' ,  
restart_mode = 'from_scratch' ,  
outdir = 'your directory for temporary files/' ,
```

/

&SYSTEM

```
ibrav = 1,  
celldm(1) = 12.D0,  
nat = 4,  
ntyp = 2,  
ecutwfc = 25.0D0 ,  
ecutrho = 160.0D0 ,  
nosym = .true. ,  
occupations = 'smearing' ,  
degauss = 0.005D0 ,
```

Change this!

Why do we treat acetylene as a metal?

/

&ELECTRONS

```
conv_thr = 1.D-7 ,  
mixing_beta = 0.5D0 ,
```

/

ATOMIC_SPECIES

```
H 1.00000 H.US_PBE.RRKJ3.UPF  
C 12.00000 C.US_PBE.RRKJ3.UPF
```

Do you have these pseudopotentials?

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
```

This atom is locked

K_POINTS gamma

Gamma point only is sufficient for the molecule-in-a-box approach

C₂H₂ SCF: the output

From the output file:

Forces acting on atoms (Ry/au):

```
atom  1 type  2   force =    0.04225946   -0.07108502   0.00000000
atom  2 type  2   force =   -0.29082443    0.14196405   0.00000000
atom  3 type  1   force =   -0.50847780    0.10558208   0.00000000
atom  4 type  1   force =    0.75704277   -0.17646111   0.00000000

Total force =      0.989284   Total SCF correction =      0.000255
```

Forces on atoms are quite large.

We now optimize the structure by 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:

```
cp acetylene.scf.in acetylene.bfgs.in
```

PAY ATTENTION !

This number 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`.

C₂H₂ BFGS minimization

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

```
cp acetylene.scf.in acetylene.bfgs.in  
vi acetylene.bfgs.in
```

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

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

C₂H₂ BFGS minimization

Examine the output:

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:

```
xcrysdn --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
```

C₂H₂ BFGS & SCF threshold

The quality of forces, is very important. Here we had:

Total force = 0.000429 Total SCF correction = 0.000013

Exercise:

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
vi acetylene.bfgs-test.in
```

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

In “Electrons” namelist

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

In from “Ions” namelist

```
upscale = 1 ,
```

4) 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!

C₂H₂ damped dynamics

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

```
cp acetylene.scf.in acetylene.bfgs.in
```

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

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

2) Set a stricter 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
```

C₂H₂ damped dynamics

Examine the output:

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:

```
xcrysdn --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 =  -0.00052921  -0.00029862   0.00000046
atom  2 type  2  force =   0.00029473   0.00021132  -0.00000059
atom  3 type  1  force =   0.00006651  -0.00020371  -0.00000019
atom  4 type  1  force =   0.00016798   0.00029100   0.00000032
```

```
Total force =      0.000539      Total SCF correction =      0.000021
```

Damped Dynamics: convergence achieved in 17 steps

Structural optimization of a slab

We want now to optimize the geometry of an Al(001) slab.

- 1) Take a look at the provided input file for SCF calculation, `al001.scf.in`.
we have five atomic layers at the bulk-truncated positions (a_{lat} units):

Al	0.000000000	0.000000000	2.828426000	1	1	1
Al	0.500000000	0.500000000	2.121320000	1	1	1
Al	0.000000000	0.000000000	1.414213000	1	1	1
Al	0.500000000	0.500000000	0.707107000	0	0	0
Al	0.000000000	0.000000000	0.000000000	0	0	0

(The bottom two layers are fixed)

Visualize the structure with XCrySDen (show multiple unit cells, press `shift-n`)

- 2) Copy to a new file `cp al001.scf.in al001.bfgs.in`
and edit it by adding the variables for the structural optimization, as seen before.

- 4) Run the calculation `pw.x < al001.bfgs.in > al001.bfgs.out`

- 5) Look at the optimized geometries:

Al	-0.000000019	-0.000000008	2.757233372
Al	0.499999997	0.499999999	2.089582133
Al	-0.000000002	-0.000000001	1.397022977
Al	0.500000000	0.500000000	0.707107000
Al	0.000000000	0.000000000	0.000000000

- 6) Compare with experiments (LEED), but before check all the convergency parameters, which in this exercise are not accurate enough!