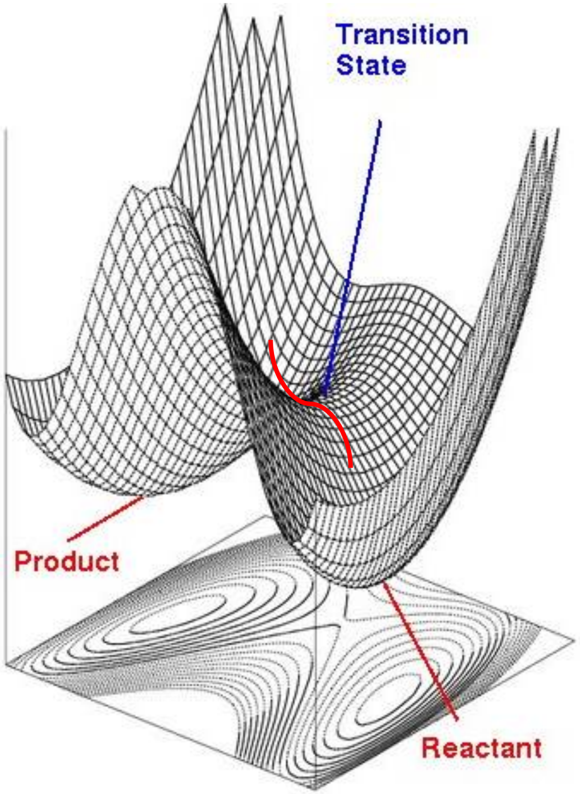
CIC nanoGUNE
nanoscience cooperative research center

Forces and Structure Relaxation

Emilio Artacho

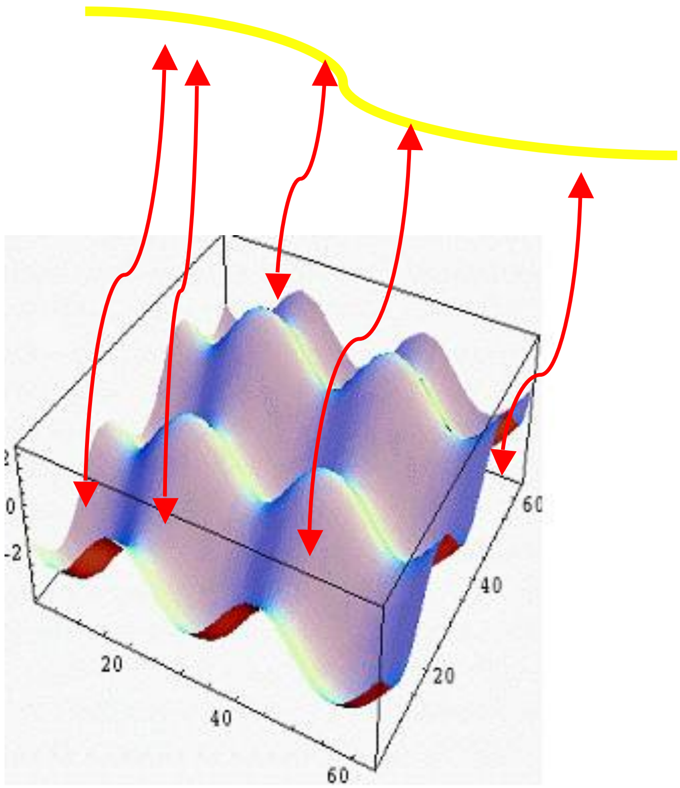
Nanogune, Ikerbasque & DIPC, San Sebastian, Spain
Cavendish Laboratory, University of Cambridge

Potential Energy Surface (PES)



Optimizations and Phonons:

- *We move on the PES*
- *Local vs global minima*
- *PES is harmonic close to minima*



MD

- *We move over the PES*
- *Good Sampling is required!!*

DERIVATIVES OF TOTAL ENERGIES

- Many properties depend more on the derivatives of the energies, than the total energies themselves
- We could get the derivatives by calculating the total energy at several points around each point, and do a numerical derivative
- This can be computationally costly, and is susceptible to numerical noise
- Another approach is to use perturbation theory – in the form of the Hellman-Feynman theorem

The Hellman-Feynman theorem

Forces on atoms: $\mathbf{F} = -\nabla_{\mathbf{R}} \langle E \rangle$ where

$\langle E \rangle = \langle \Psi | H | \Psi \rangle$, if $|\Psi\rangle$ are normalised

$$\frac{\partial E}{\partial \lambda} = \langle \frac{\partial \Psi}{\partial \lambda} | H | \Psi \rangle + \langle \Psi | \frac{\partial H}{\partial \lambda} | \Psi \rangle + \langle \Psi | H | \frac{\partial \Psi}{\partial \lambda} \rangle$$

$H|\Psi\rangle = E|\Psi\rangle$ and so

$$\frac{\partial E}{\partial \lambda} = \langle \Psi | \frac{\partial H}{\partial \lambda} | \Psi \rangle + E \frac{\partial}{\partial \lambda} \langle \Psi | \Psi \rangle$$

$$\boxed{\frac{\partial E}{\partial \lambda} = \langle \Psi | \frac{\partial H}{\partial \lambda} | \Psi \rangle}$$

Forces on the atoms

Hellman-Feynmann theorem:

$$F_I = -\frac{\partial \langle \psi | H | \psi \rangle}{\partial R_I} = -\cancel{\left\langle \frac{\partial \psi}{\partial R_I} \middle| H \middle| \psi \right\rangle} - \cancel{\left\langle \psi \middle| H \middle| \frac{\partial \psi}{\partial R_I} \right\rangle} - \left\langle \psi \middle| \frac{\partial H}{\partial R_I} \middle| \psi \right\rangle$$

Using an atomic-like basis:

$$\psi(\mathbf{r}) = \sum_{\mu} c_{\mu} \phi_{\mu}(\mathbf{r})$$

$$F_I = -\frac{\partial \langle \psi | H | \psi \rangle}{\partial R_I} = -\frac{\partial}{\partial R_I} \sum_{\mu\nu} c_{\mu} c_{\nu} \langle \phi_{\mu} | H | \phi_{\nu} \rangle$$

$$= -\sum_{\mu\nu} \left[\cancel{\frac{\partial c_{\mu}}{\partial R_I} c_{\nu}} + \cancel{c_{\mu} \frac{\partial c_{\nu}}{\partial R_I}} \right] \langle \phi_{\mu} | H | \phi_{\nu} \rangle - \sum_{\mu\nu} c_{\mu} c_{\nu} \left\langle \phi_{\mu} \middle| \frac{\partial H}{\partial R_I} \middle| \phi_{\nu} \right\rangle -$$

$$-\sum_{\mu\nu} c_{\mu} c_{\nu} \left[\left\langle \frac{\partial \phi_{\mu}}{\partial R_I} \middle| H \middle| \phi_{\nu} \right\rangle + \left\langle \phi_{\mu} \middle| H \middle| \frac{\partial \phi_{\nu}}{\partial R_I} \right\rangle \right]$$

Pulay forces

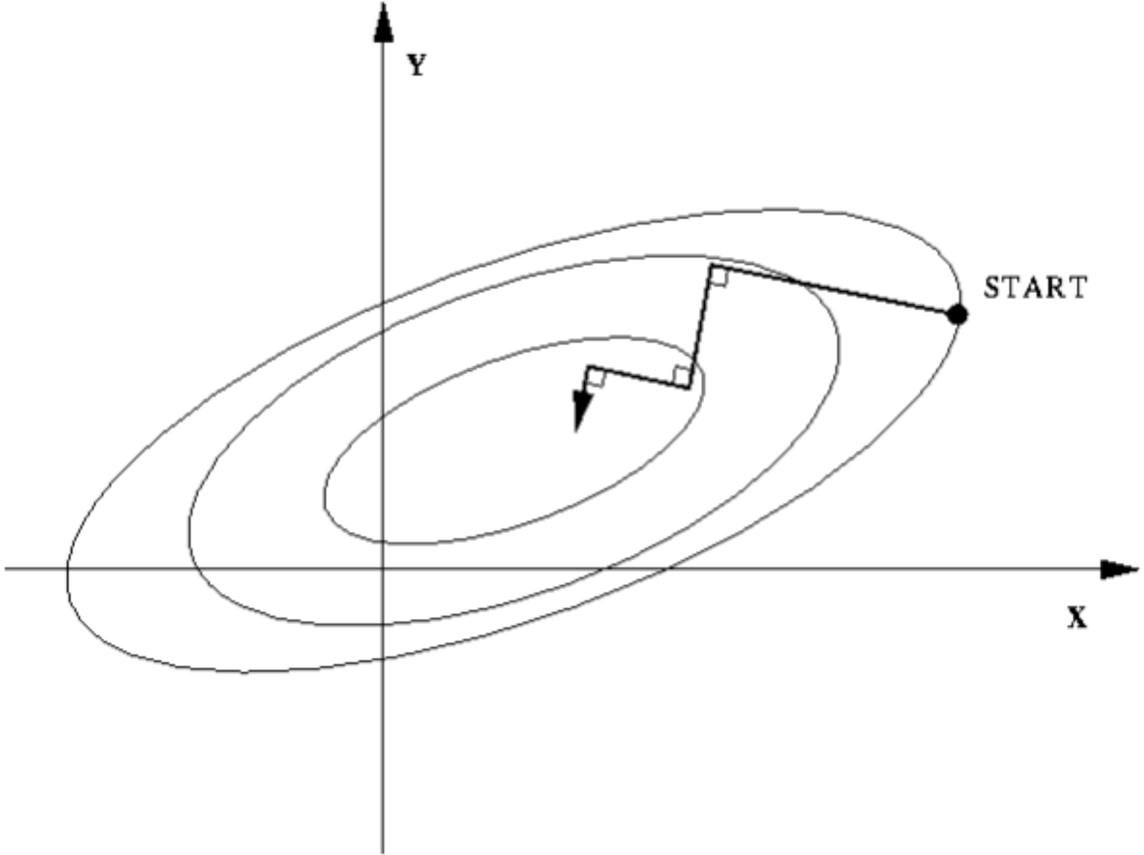
THE HELLMAN-FEYNMAN THEOREM AND DENSITY FUNCTIONAL THEORY

$$\boxed{H_{\mathbf{R}} = -\tfrac{1}{2}\nabla^2_{\mathbf{r}} + V_{\mathrm{e-e}}(\mathbf{r}) + V_{\mathrm{ion-e}}(\mathbf{r},\mathbf{R}) + V_{\mathrm{xc}}(\mathbf{r}) + V_{\mathrm{ion-ion}}(\mathbf{R})}$$

- The DFT Hamiltonian is parameterised in $\mathbf{R}$
- We get a contribution from the ion-electron (pseudo)potential
- We get a contribution from the ion-ion Couloumb interaction (from the Ewald sum)
- For the stresses, there are contributions from the kinetic energy and the Hartree terms
- While the total energy is correct to second order in the errors, the forces are only good to first order

Structural optimisation: Steepest descent

Move in the direction of maximum incline.



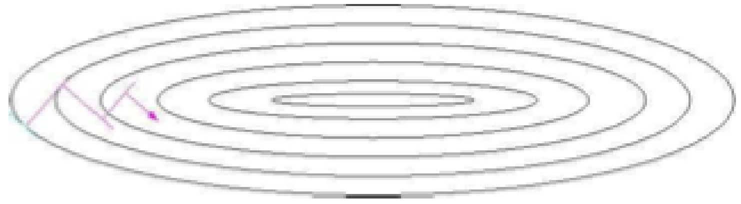
Repeats search directions

Structural optimisation:

minimum E vs atomic positions

Follow forces on atoms (gradient of function)

STEEPEST DESCENTS



- This is the simplest approach:
  - take a downhill step along the local steepest gradient, and a trial step length
  - use line minimisation to find the optimal step length

- Advantages:
  - simple to implement, and robust
  - reliable – will find the minima eventually

- Disadvantages:
  - very slow to converge
  - can get stuck in a local minima

Theory for (local) geometry optimization

$$E(x+\delta x) = E(x) + \underset{\underset{\textit{Gradients}}{\uparrow}}{G(x)}\delta x + \frac{1}{2}\underset{\underset{\textit{Hessian}}{\uparrow}}{H(x)}\delta x^2$$

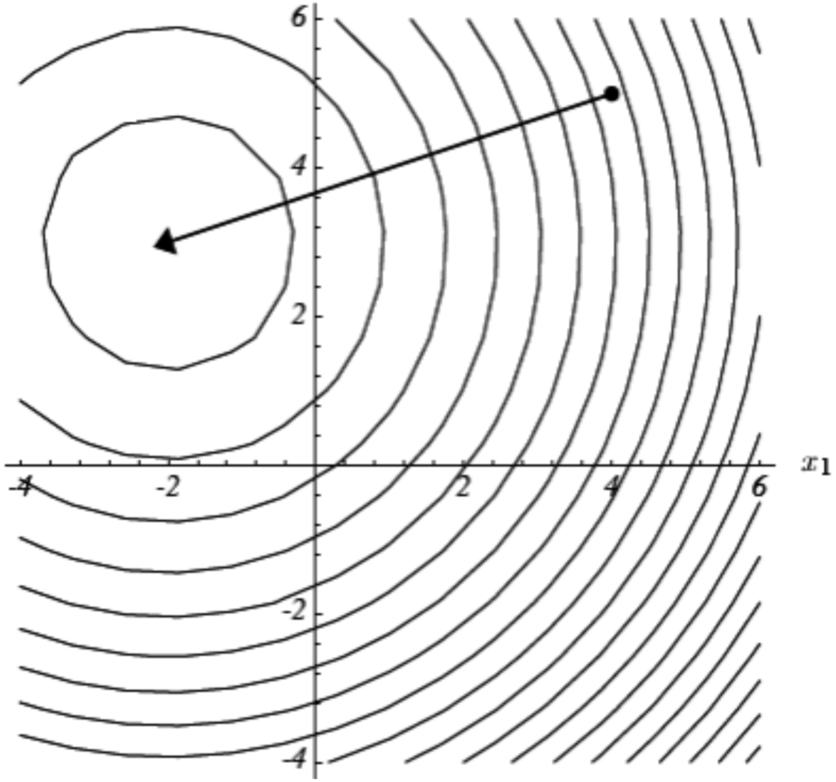
$$\delta x = -\underset{\underset{\alpha\ \textit{= 1 for quadratic region}}{\uparrow}}{\alpha} H(x)^{-1} G(x)$$

Energy minimisation within a basin

Conditioning

Given eigenvalues of Hessian $H\vec{\chi}_i = \omega_i \vec{\chi}_i$

Condition number: $\kappa \propto \frac{\omega_i^{max}}{\omega_i^{min}}$ *Determines convergence*



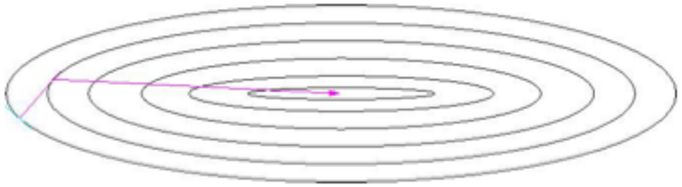
It converges on the first iteration If all ω are equal

Structural optimisation:

minimum E vs atomic positions

Follow forces on atoms (gradient of function)

CONJUGATE GRADIENTS



- This improves on steepest descents:
  - the gradient is constructed to be conjugate to all previous directions
  - does not undo previous minimisation
  - a line minimisation is performed

- Advantages:
  - rapid convergence – in a quadratic energy landscape, one dof per iteration
  - low storage requirements

- Disadvantages:
  - more complex to implement than SD
  - Hessian not explicitly calculated
  - can get stuck in a local minima

Structural optimisation:

minimum E vs atomic positions

Follow forces on atoms (gradient of function)

BROYDEN-FLETCHER-GOLDFARD-SHANNO – BFGS

$$\mathbf{A} = \begin{pmatrix} \frac{\partial^2 E}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 E}{\partial x_N \partial x_N} \end{pmatrix}$$

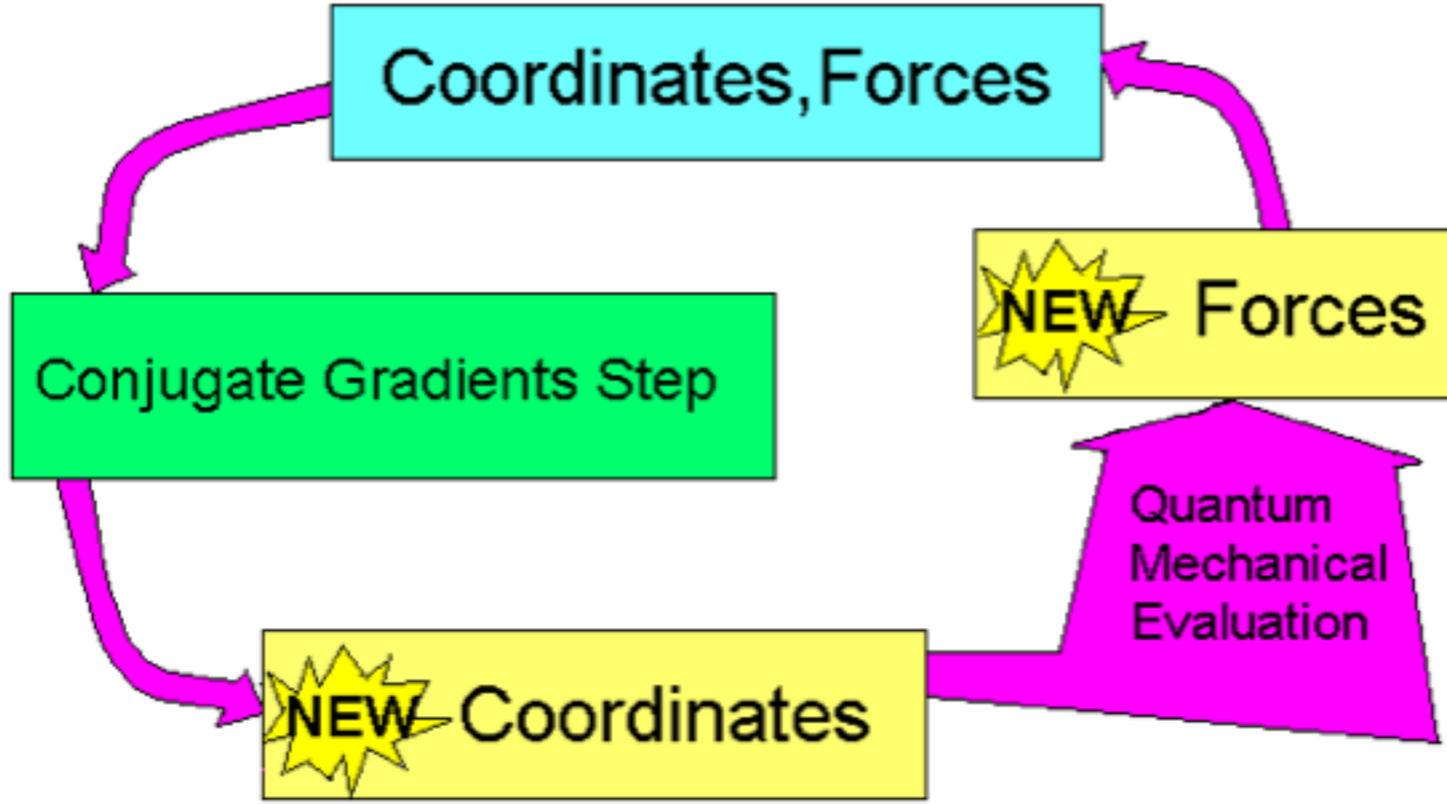
$$\delta E = \frac{1}{2}(\mathbf{x} - \mathbf{x}_{\text{min}})^T \cdot \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}_{\text{min}})$$

- If we know the Hessian $\mathbf{A}$ we can move from nearby the minimum straight to it
  - we don't know it, so we build up a guess using the BFGS algorithm

- Advantages:
  - convergence similar or better than CG
  - extra physical information is contained in the Hessian

- Disadvantages:
  - complex to code
  - Hessian must be stored (# dof^2)
  - can get stuck in a local minima

Optimization (and MD) general basic Step



Optimization in SIESTA(1)

- *Set runtype to conjugate gradients:*

  MD.TypeOfRun CG, Broyden

- *Set maximum number of iterative steps:*

  MD.NumCGsteps 100

- *Optionally set force tolerance:*

  MD.MaxForceTol 0.04 eV/Ang

- *Optionally set maximum displacement:*

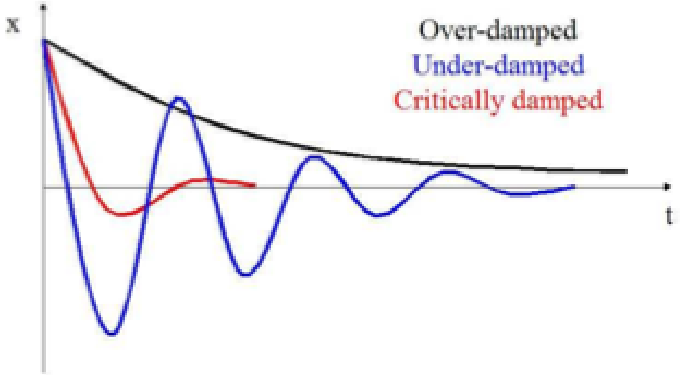
  MD.MaxCGDispl 0.2 Bohr

Structural optimisation:

minimum E vs atomic positions

Follow forces on atoms (gradient of function)

DAMPED MOLECULAR DYNAMICS



- This improves on steepest descents:
  - use velocities as well as forces
  - start with $\mathbf{v} = \mathbf{0}$ and add damping term to forces $-\gamma\mathbf{v}$
  - adjust γ and time step to obtain optimal convergence

- Advantages:
  - simple to implement, robust and more efficient than SD
  - can use wavefunction extrapolation (or Car-Parinello)

- Disadvantages:
  - convergence rate depends on damping factor γ
  - can get stuck in a local minima

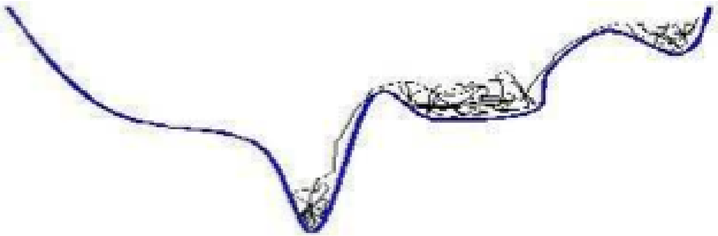
Structural optimisation:

global minimum E vs atomic positions

Follow forces on atoms (gradient of function)

SIMULATED ANNEALING

MD



- This is a stochastic method:
  - always accept steps that lower the energy, and sometimes accept upward steps, using a Boltzman distribution
  - slowly reduce temperature, and iterate to the goundstate

- Advantages:
  - very robust and reliable
  - reasonably immune to getting stuck in local minima

- Disadvantages:
  - incredibly slow convergence
  - the cooling rate must be carefully adjusted to avoid quenching into local minima
  - no guarantee that the true global minima will be found

Structural optimisation:

Variable cell

STRESS AND STRAIN

- The concept of forces is straightforward, but you can also take derivatives with respect to the crystal unit cell

- The application of a *strain* changes the shape of the unit cell

- If we write the three unit cell vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ as columns of a matrix $\mathbf{h}$ the shape change is described by:

$$\mathbf{h}' = (\mathbf{I} + \boldsymbol{\epsilon})\mathbf{h}$$

- The *stress* tensor $\boldsymbol{\sigma}$ is related to the strain tensor $\boldsymbol{\epsilon}$:

$$\boldsymbol{\sigma}_{\alpha\beta} = \frac{1}{\Omega}\frac{\partial E}{\partial \boldsymbol{\epsilon}_{\alpha\beta}}$$

where $\boldsymbol{\Omega} = \mathbf{a} \cdot \mathbf{b} \times \mathbf{c}$ is the volume of the unit cell

Stress tensor and pressure

$$r_\alpha \rightarrow \sum_\beta \varepsilon_{\alpha\beta} r_\beta \qquad \alpha,\beta = \{x,y,z\}$$

$$u_{i\alpha} \rightarrow \sum_\beta \varepsilon_{\alpha\beta} u_{i\beta}$$

Strain tensor

$$\sigma_{\alpha\beta} \equiv \frac{1}{\Omega} \frac{\partial E}{\partial \varepsilon_{\alpha\beta}}$$

Stress tensor (Siesta)

$$P = -\frac{1}{3} \sum_\alpha \sigma_{\alpha\alpha}$$

Hydrostatic pressure

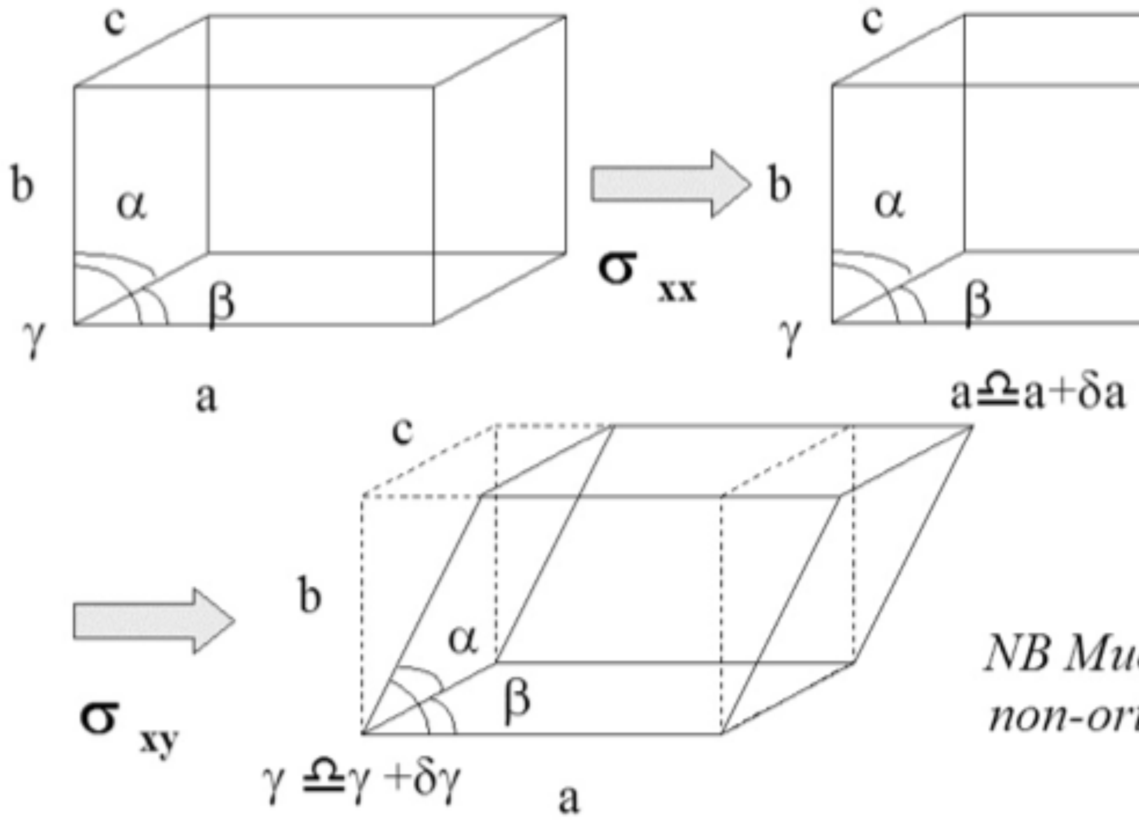
$$P_{mol} = P - \frac{1}{\Omega} \sum_I \mathbf{R}_I \mathbf{F}_I$$

Corrected pressure

Structural optimisation:

Variable cell

STRESS AND STRAIN



NB Much messier if non-orthogonal cell

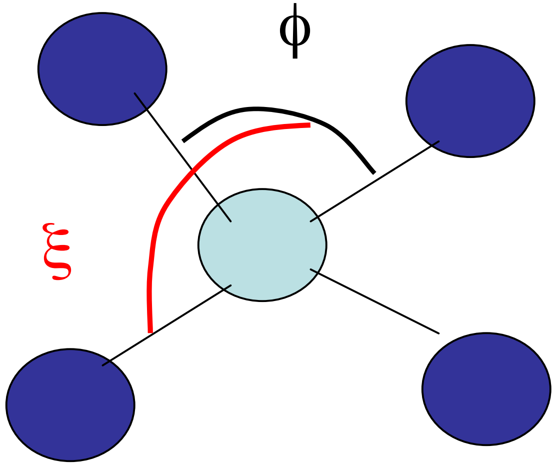
Optimizations in SIESTA(2)

By default optimisations are for a fixed cell

- *To allow unit cell to vary:*
  MD.VariableCell `true`
- *Optionally set stress tolerance:*
  MD.MaxStressTol `0.1 Gpa`
- *Optionally set cell preconditioning:*
  MD.PreconditionVariableCell `5.0 Ang`
- *Set an applied pressure:*
  MD.TargetPressure `5.0 GPa`

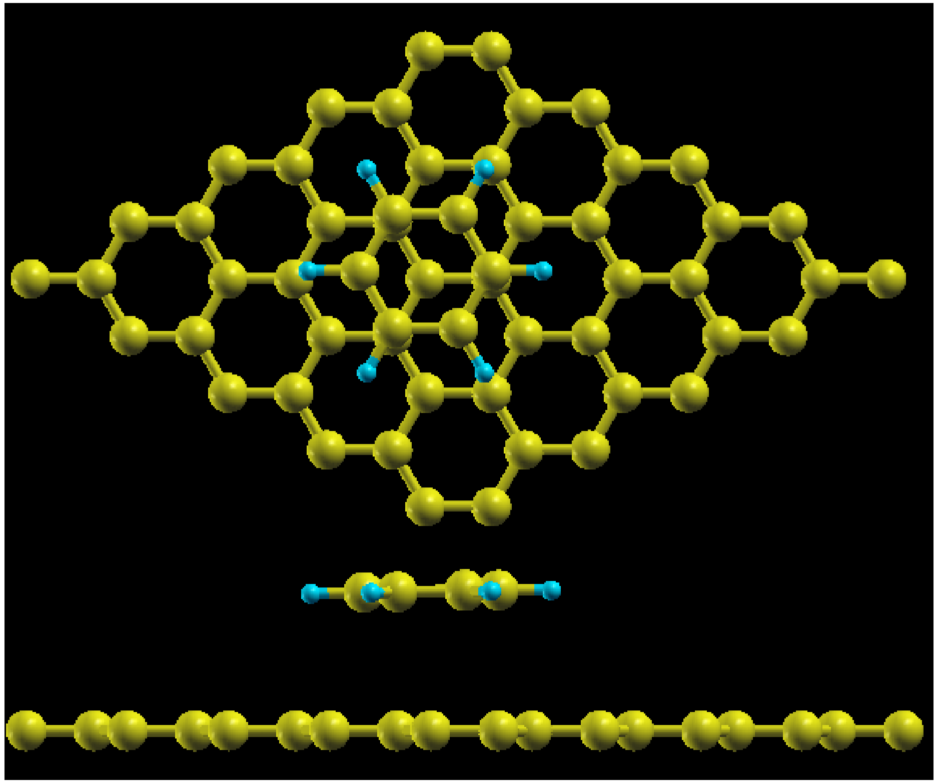
Z-Matrix coordinate format

- *Internal coordinates: Molecules represented by :*
  - *Bond lengths* χ_i
  - *Bending angles* ϕ_i
  - *Dihedral angles* ξ_i



Z-Matrix

- *Allows for mixing of generalised and Cartesian coordinates: Useful for constrained relaxations*

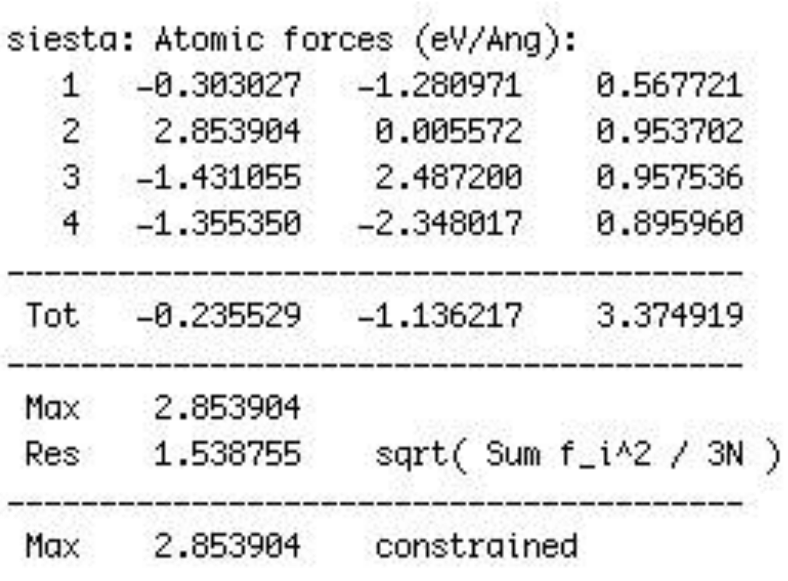


Explore the PES by using
A relevant coordinate:
Useful for estimating
barriers

Advice on optimizations in SIESTA(I)

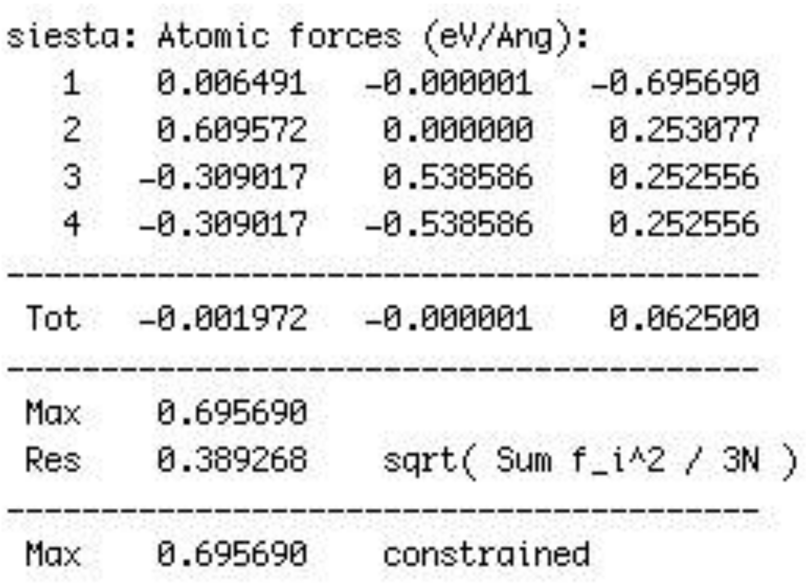
Make sure that your MeshCutoff is high enough:

- *Mesh leads to space rippling*
- *If oscillations are large convergence is slow*
- *May get trapped in wrong local minimum*



```
siesta: Atomic forces (eV/Ang):
     1   -0.303027   -1.280971    0.567721
     2    2.853904    0.005572    0.953702
     3   -1.431055    2.487200    0.957536
     4   -1.355350   -2.348017    0.895960
----------------------------------------
   Tot   -0.235529   -1.136217    3.374919
----------------------------------------
   Max    2.853904
   Res    1.538755    sqrt( Sum f_i^2 / 3N )
----------------------------------------
   Max    2.853904    constrained
```

✗



```
siesta: Atomic forces (eV/Ang):
     1    0.006491   -0.000001   -0.695690
     2    0.609572    0.000000    0.253077
     3   -0.309017    0.538586    0.252556
     4   -0.309017   -0.538586    0.252556
----------------------------------------
   Tot   -0.001972   -0.000001    0.062500
----------------------------------------
   Max    0.695690
   Res    0.389268    sqrt( Sum f_i^2 / 3N )
----------------------------------------
   Max    0.695690    constrained
```

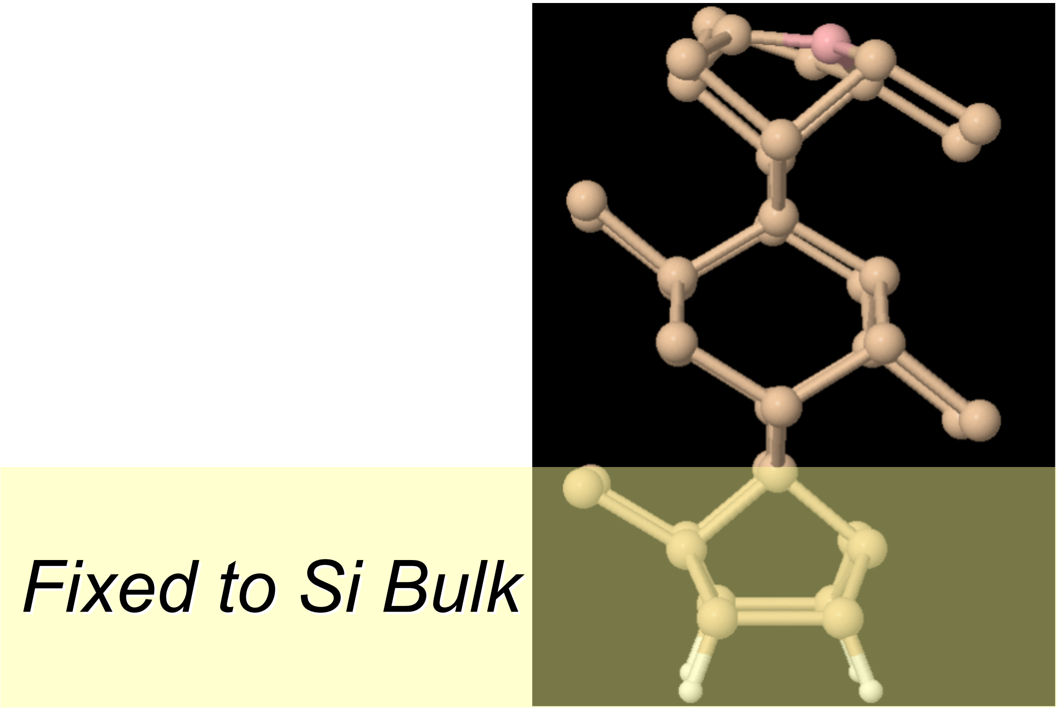
✓

Eggbox ….

Advice on Optimizations in SIESTA(II)

Ill-conditioned systems (soft modes) can slow down optimizations, very sensitive to mesh cutoff.

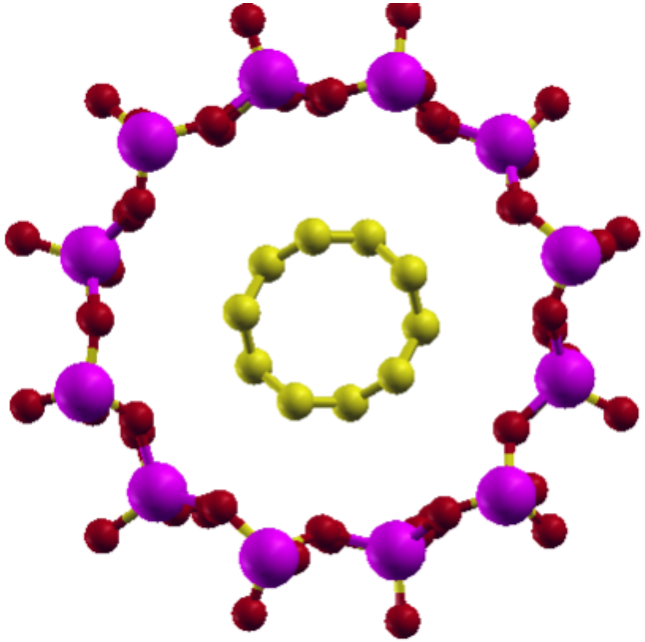
Use constraints when relevant.



Advice on Optimizations in SIESTA(III)

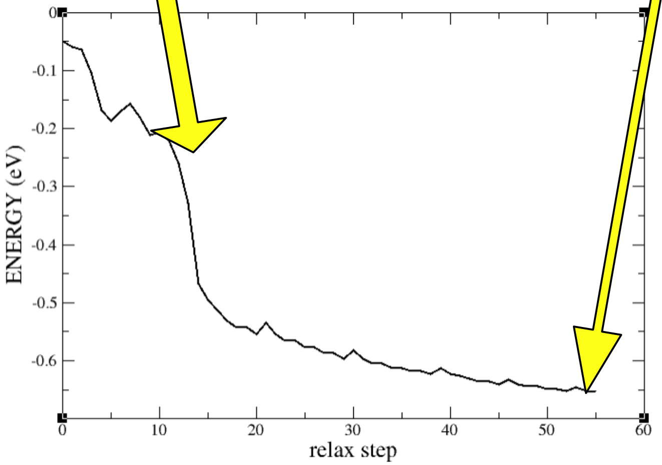
- *Decouple Degrees of freedom (relax separately different parts of the system).*

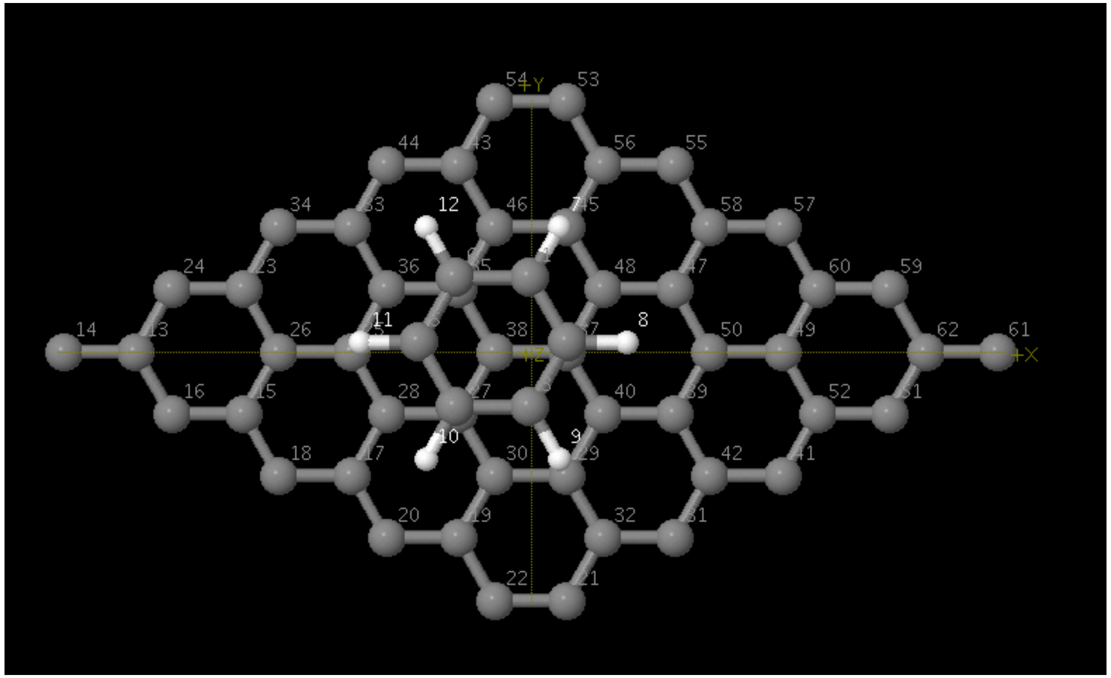
Look at the evolution of relevant physics quantities (band structure, Ef).

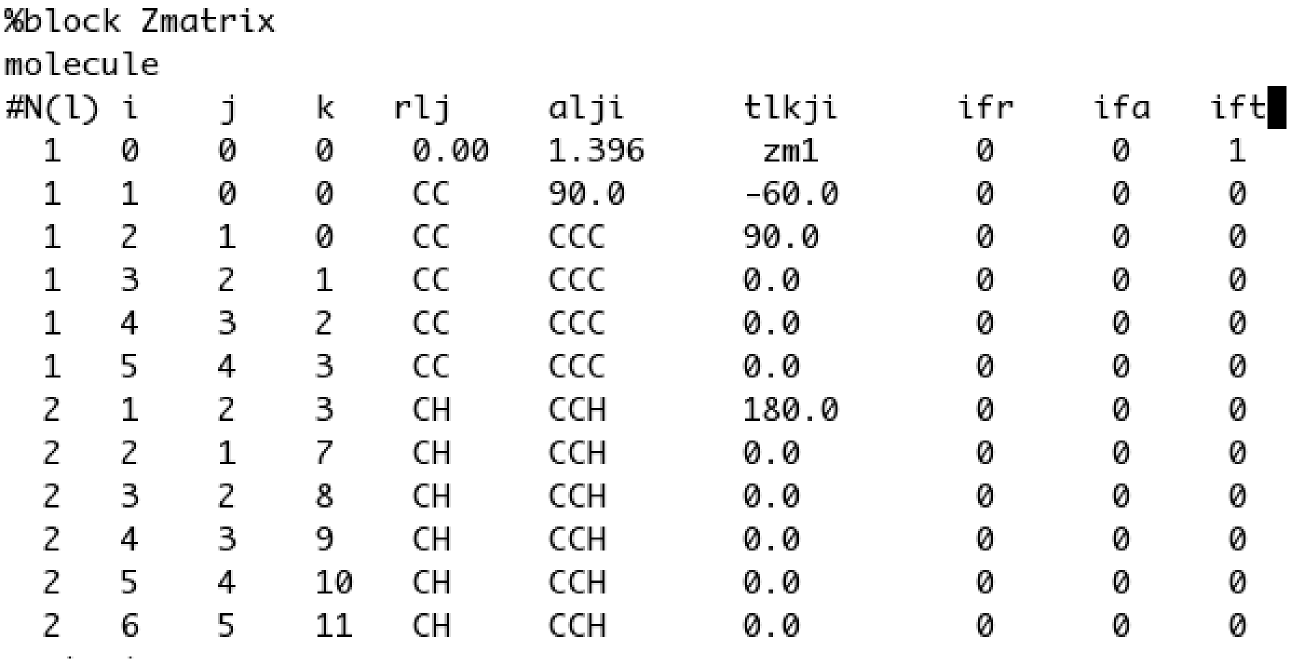


No constraints

Fix the Zeolite,
Its relaxation is no
Longer relevant.
F_{tube}<0.04 eV/A
F_{zeol}>0.1 eV/A







```
%block Zmatrix
molecule
#N(l) i     j     k    rlj      alji       tlkji      ifr    ifa    ift
  1   0     0     0     0.00    1.396       zm1       0      0      1
  1   1     0     0     CC      90.0       -60.0      0      0      0
  1   2     1     0     CC      CCC        90.0       0      0      0
  1   3     2     1     CC      CCC        0.0        0      0      0
  1   4     3     2     CC      CCC        0.0        0      0      0
  1   5     4     3     CC      CCC        0.0        0      0      0
  2   1     2     3     CH      CCH        180.0      0      0      0
  2   2     1     7     CH      CCH        0.0        0      0      0
  2   3     2     8     CH      CCH        0.0        0      0      0
  2   4     3     9     CH      CCH        0.0        0      0      0
  2   5     4     10    CH      CCH        0.0        0      0      0
  2   6     5     11    CH      CCH        0.0        0      0      0
```