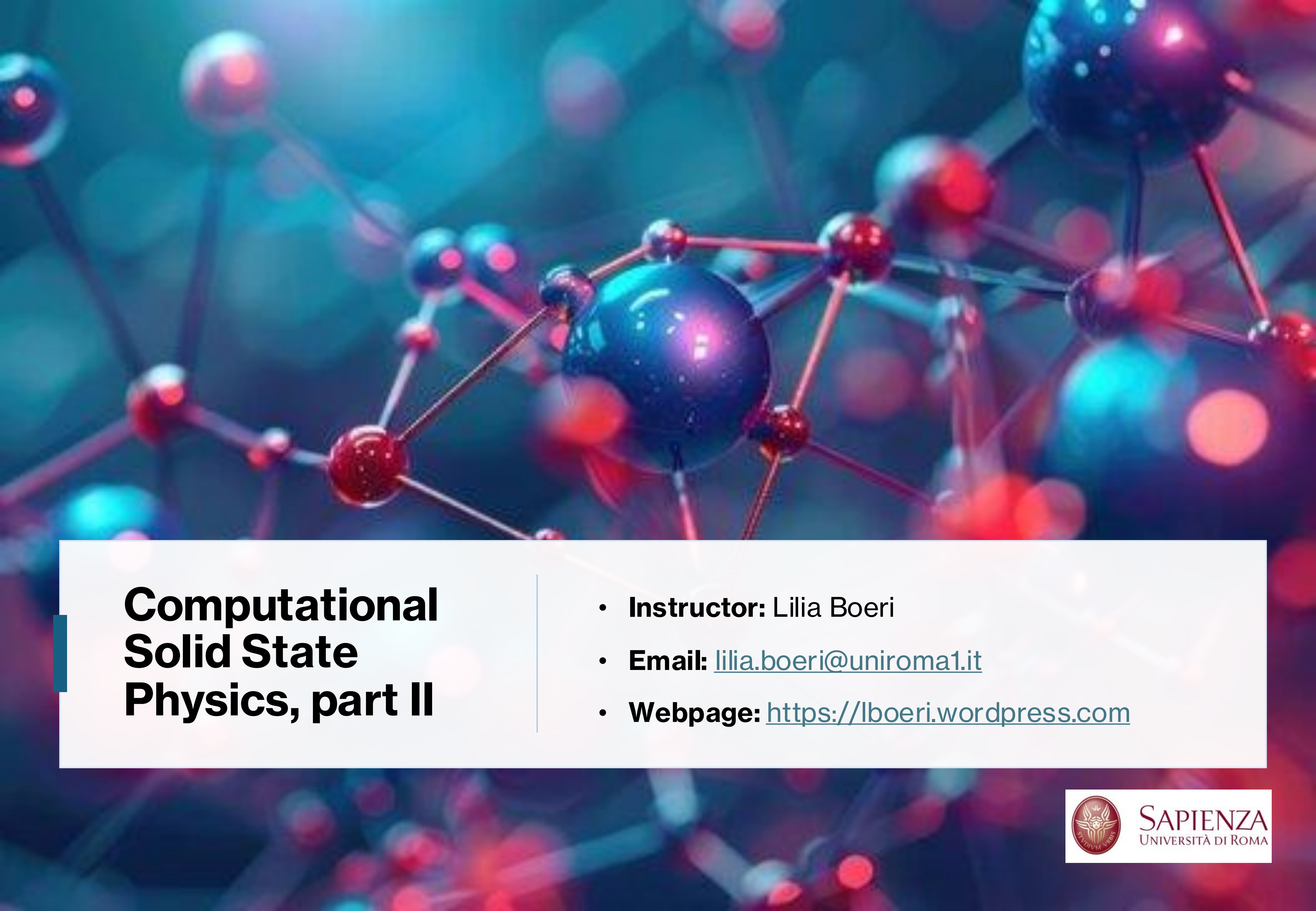


Computational Solid State Physics, part II

- **Instructor:** Lilia Boeri
- **Email:** lilia.boeri@uniroma1.it
- **Webpage:** https://lboeri.wordpress.com

Lab 4: Structural Relaxation

Recap of Important Concepts:

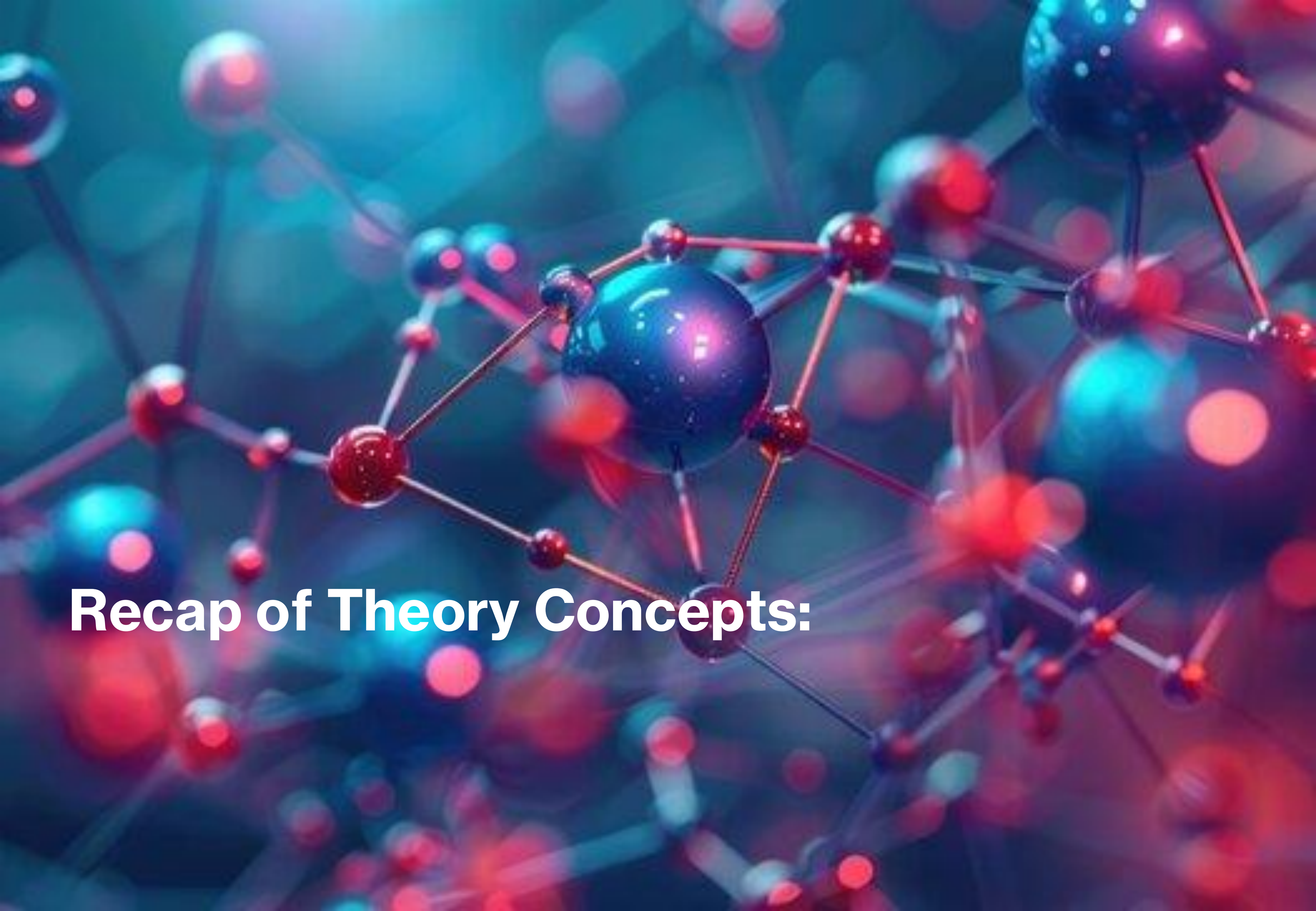
Forces and Relaxations in DFT, Potential Energy Surface, structural relaxation.

Hands-on:

- **Calculation of Forces:** Finite Differences.
- **Structural Optimization:** Internal Coordinates.
- **Pressure** and Variable Cell Optimization

Assignment:

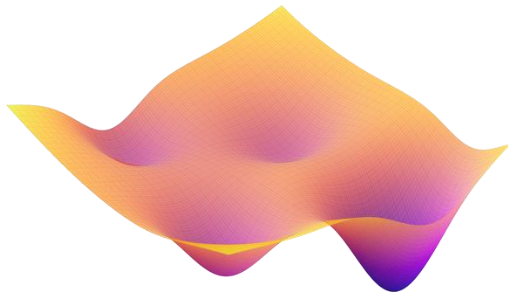
High-Pressure Silicon Phase Diagram.



Recap of Theory Concepts:

Structural Relaxation

Structural Relaxation: Optimize unit cell Parameters to find the local minima of the **Potential Energy Surface**.



What is needed?

- Defining the relation between the Potential Energy surface and Structural Optimization - **Born-Oppenheimer Approximation.**

- Computing reliable **derivatives** of the energy with respect to atomic positions (**forces**) and cell deformations (**stress**)- **Hellmann-Feynmann Theorem**.

- Exploiting robust **Minimization Algorithms** to locate the minima of the PES - **Steepest Descent, Conjugate Gradients, Quasi-Newton**.

Nuclear Hamiltonian:

Nuclear Hamiltonian (Born-Oppenheimer):

$$\boxed{\hat{H}_N = -\sum_{\alpha} \frac{\nabla_{\alpha}^2}{2M_{\alpha}} + U(\mathbf{R}_{\alpha}, ..., \mathbf{R}_{\nu})}$$

The **potential energy** of the nuclei is given by:

$$\boxed{U(\mathbf{R}_{\alpha}, ..., \mathbf{R}_{\nu}) = E_{el}(\mathbf{R}_{\alpha}, ..., \mathbf{R}_{\nu}) + \frac{1}{2}\sum_{\alpha,\beta} \frac{Z_{\alpha}Z_{\beta}}{|\mathbf{R}_{\alpha} - \mathbf{R}_{\beta}|}}$$

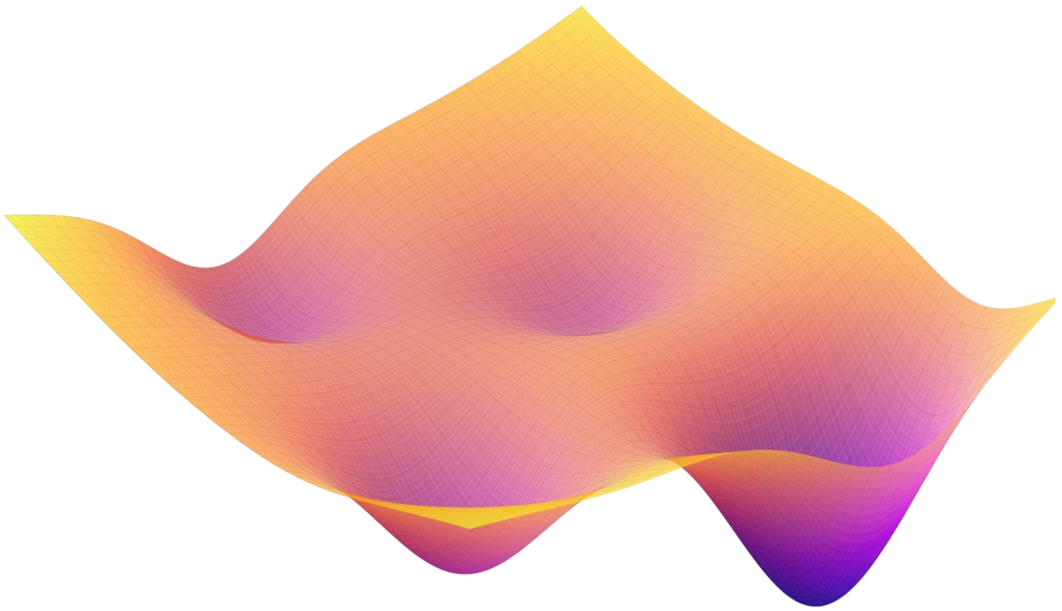
Except for very light nuclei (H,D), **classical mechanics** is sufficient to describe **nuclear dynamics:**

$$\boxed{H_N^{clas} = \sum_{\alpha} \frac{\mathbf{P}_{\alpha}^2}{2M_{\alpha}} + U(\mathbf{R}_{\alpha}, ..., \mathbf{R}_{\nu})}$$

Potential Energy Surface (B-O):

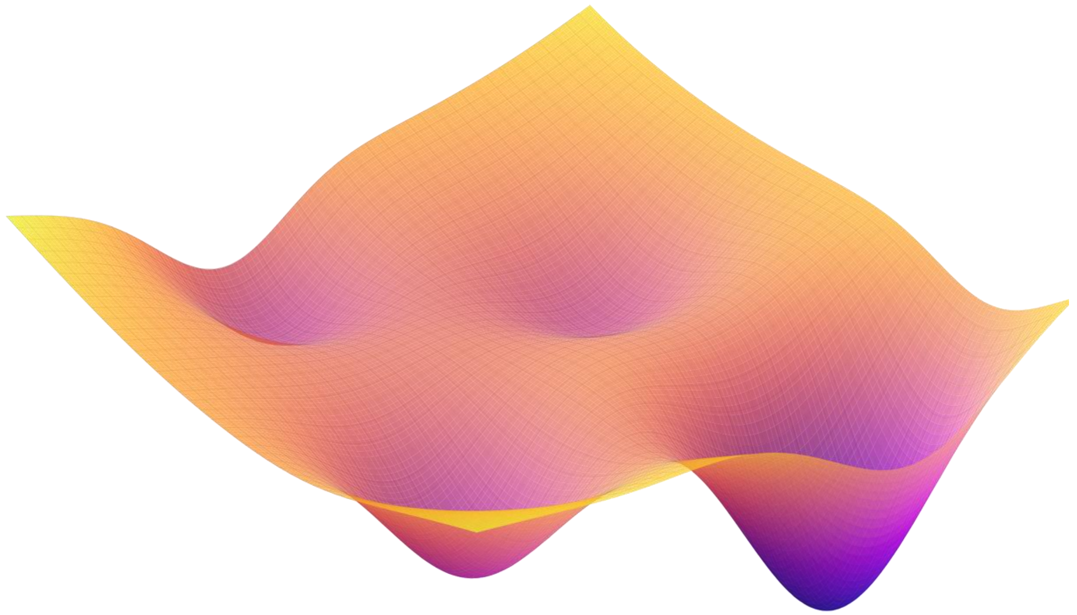
$$\boxed{U(\mathbf{R}_\alpha, ..., \mathbf{R}_\nu) = E_{el}(\mathbf{R}_\alpha, ..., \mathbf{R}_\nu) + \frac{1}{2}\sum_{\alpha,\beta}\frac{Z_\alpha Z_\beta}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|}}$$

Potential Energy Surface



The Bohr-Oppenheimer **Potential Energy** defines a multi-dimensional hypersurface in nuclear coordinate space. Local and global minima correspond to (meta)stable structures.

Potential Energy Surface:

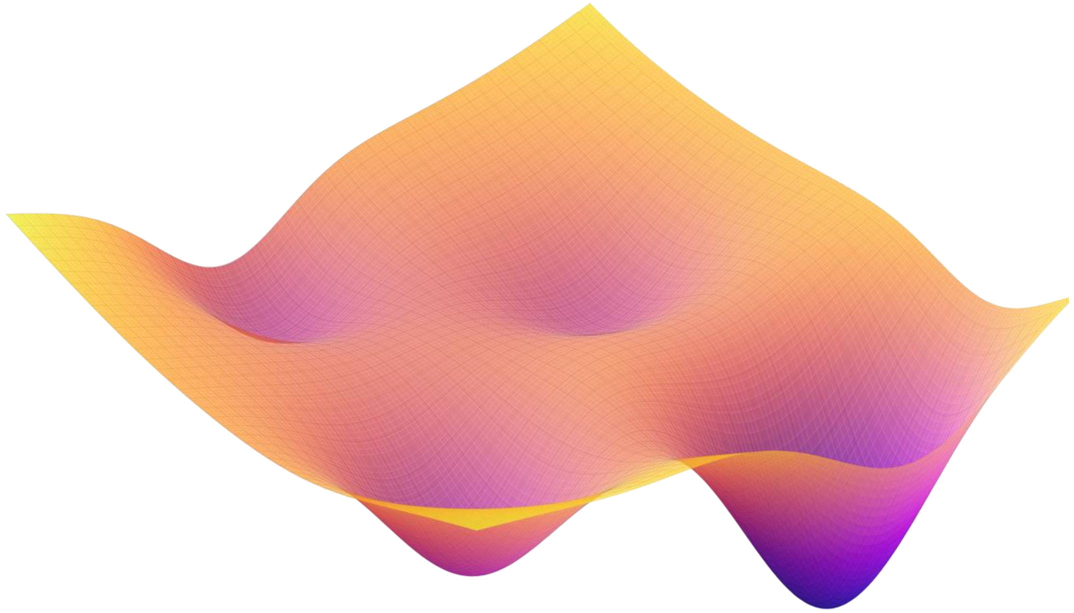


Relaxing a structure means starting from an arbitrary point of the PES and reach the relative **local minimum.**

$$\boxed{M_\alpha \frac{d^2\mathbf{R}_\alpha}{dt^2} = -\frac{\partial U}{\partial \mathbf{R}_\alpha} = 0, \forall \alpha}$$

A **structure** is at **equilibrium** when all forces acting on the nuclei vanish.

Potential Energy Surface:



Relaxing a structure means starting from an arbitrary point of the PES and reach the relative **local minimum.**

$$\boxed{M_\alpha \frac{d^2\mathbf{R}_\alpha}{dt^2} = -\frac{\partial U}{\partial \mathbf{R}_\alpha} = 0, \forall \alpha}$$

A **structure** is at **equilibrium** when all forces acting on the nuclei vanish.

How to compute forces?

- **Finite Differences:**

Simple, but **inefficient** (two additional scf calculations for each atomic displacement) . May work only for very simple (obvious) optimization.

- **Hellmann-Feynmann Theorem:**

$$\boxed{\frac{\partial E_0^\lambda}{\partial \lambda} = \langle \Psi_0^\lambda | \frac{\partial H_{KS}^\lambda}{\partial \lambda} | \Psi_0^\lambda \rangle}$$

Very efficient: Calculation of forces requires computing simple integrals of the ground-state density of the umperturbed structure, no additional scf cycle.

$$\boxed{\lambda \longrightarrow \mathbf{R}_\alpha}$$

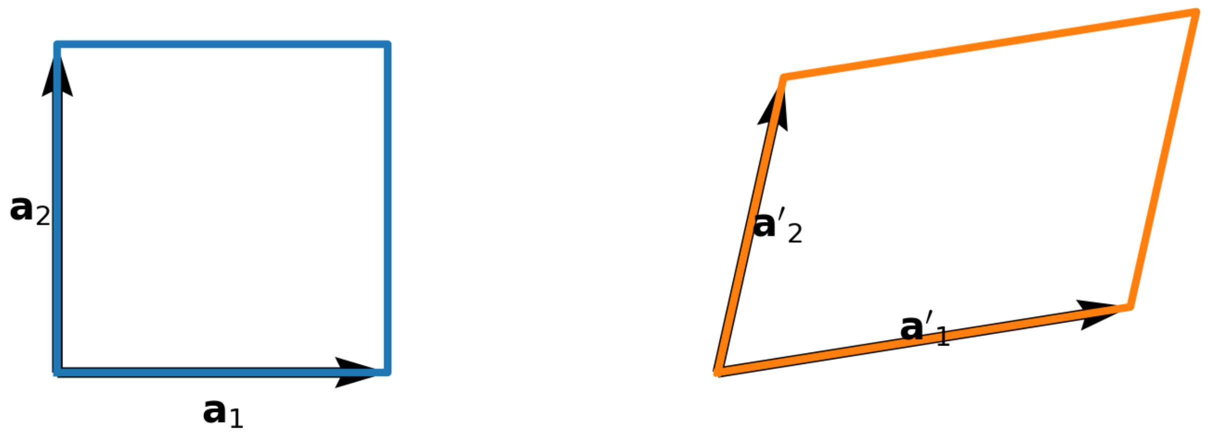
Positions of the atoms within the unit cell -> Forces

$$\boxed{\lambda \longrightarrow \mathbf{a}, \mathbf{b}, \mathbf{c}}$$

Deformation of the unit cell shape or volume -> Stress

Strain: Defintion

Deformations of the **unit cell** define a **strain tensor**.



reference cell

strained cell

Lattice deformation: Strain tensor

$$\mathbf{a}' = (\mathbf{I} + \boldsymbol{\varepsilon})\,\mathbf{a}$$

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix}$$

Stress Tensor:

Applying a strain to the lattice causes a response (**stress**).

Stress tensor:

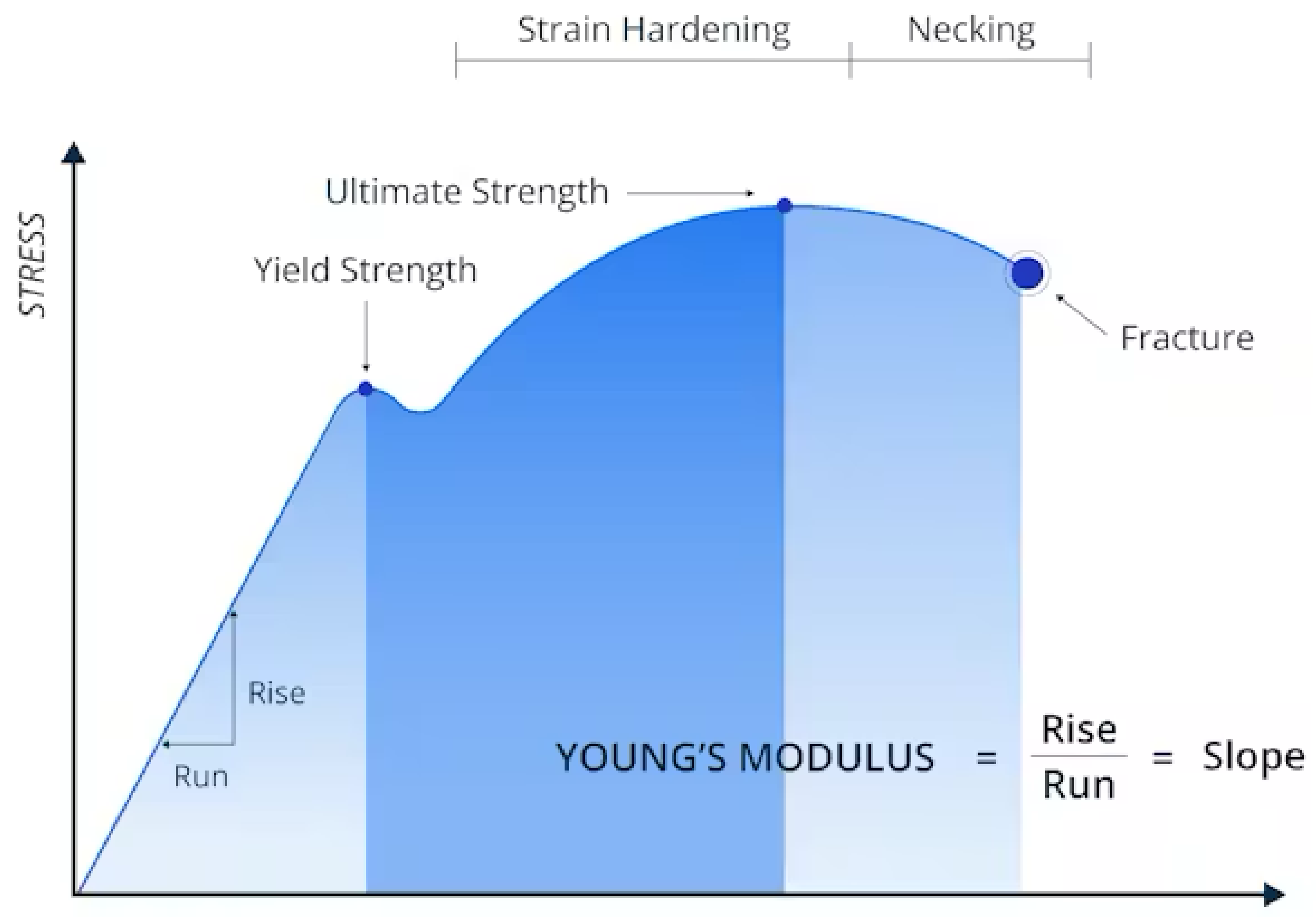
$$\boxed{\sigma_{\alpha\beta} = \frac{1}{\Omega}\frac{\partial E}{\partial \varepsilon_{\alpha\beta}}}$$

Diagonal components of the stress tensor are related to hydrostatic **pressure**:

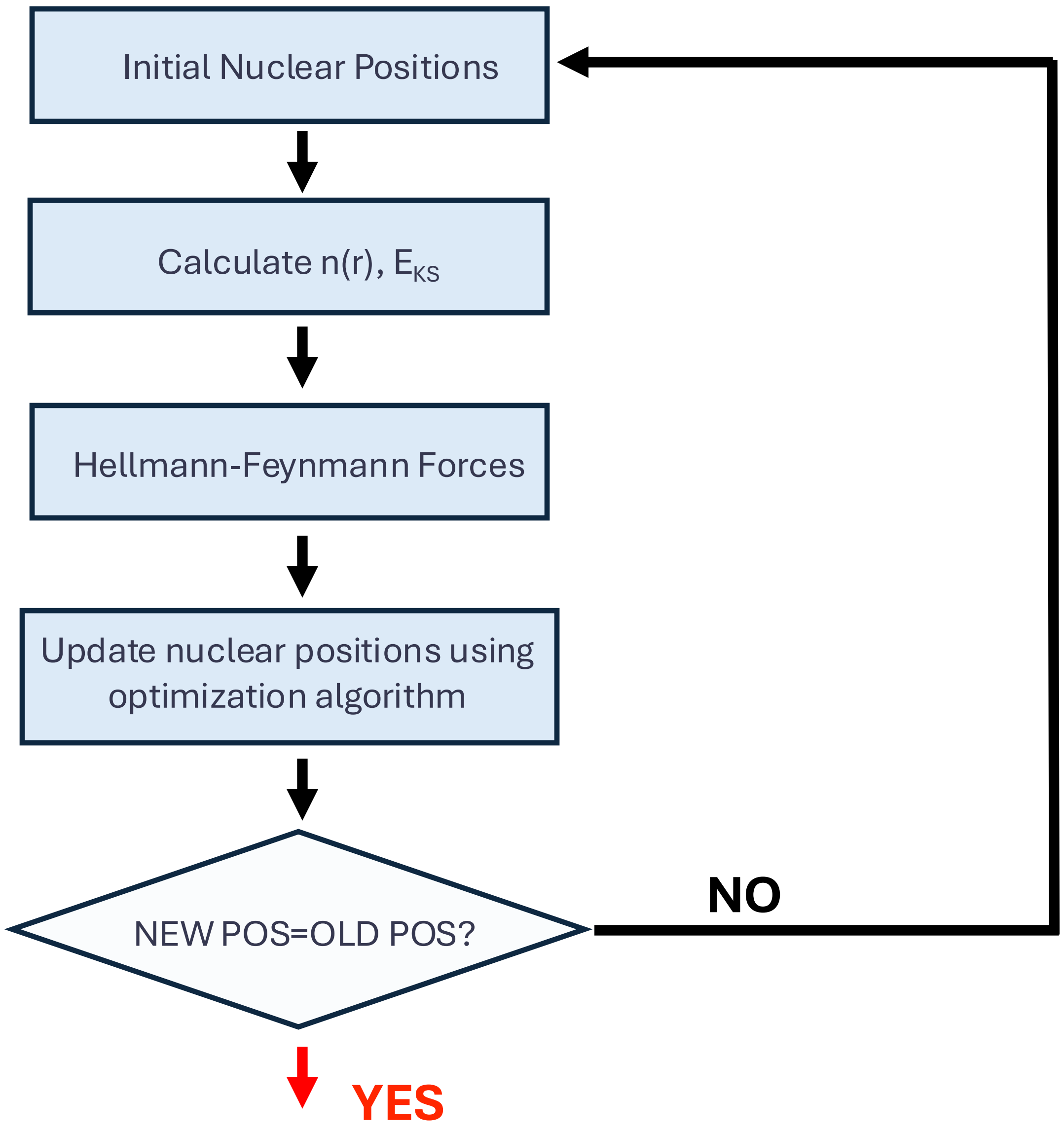
$$\boxed{P_{\text{hydro}} = -\frac{1}{3}\text{Tr}(\boldsymbol{\sigma}) = -\frac{1}{3}\left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}\right)}$$

Off-diagonal components indicate **shear stresses**, that tend to modify the cell shape (angles between lattice vectors).

Stress Strain Curve



Structural Relaxation in DFT


Initial Nuclear Positions
Calculate n(r), E_{KS}
Hellmann-Feynmann Forces
Update nuclear positions using optimization algorithm
NEW POS=OLD POS?
NO
YES

Structural Relaxation Algorithms:

For this, different classes of methods are used:

- **Steepest Descent:** Atomic Positions are moved in the direction of the steepest decrease in energy (maximum gradient).
- **Conjugate Gradients:** Improves on Steepest Descent by choosing the descent direction in a more efficient way, which exploits information from previous steps.
- **Broyden–Fletcher–Goldfarb–Shanno (BFGS) Algorithm:** Quasi-Newton method that approximates the second derivatives of the energy (Hessian matrix) using gradients from previous steps.

Structural Relaxation Algorithms:

Steepest Descent:

$$\boxed{\mathbf{R}_{k+1} = \mathbf{R}_k - \alpha_k \nabla E(\mathbf{R}_k)}$$

Follow the **gradient**!

Conjugate Gradients:

$$\boxed{\mathbf{R}_{k+1} = \mathbf{R}_k + \alpha_k \mathbf{p}_k}$$

$$\mathbf{p}_{k+1} = -\nabla E(\mathbf{R}_{k+1}) + \beta_k \mathbf{p}_k$$

Update **Search Direction** at every step

Broyden–Fletcher–Goldfarb–Shanno (BFGS):

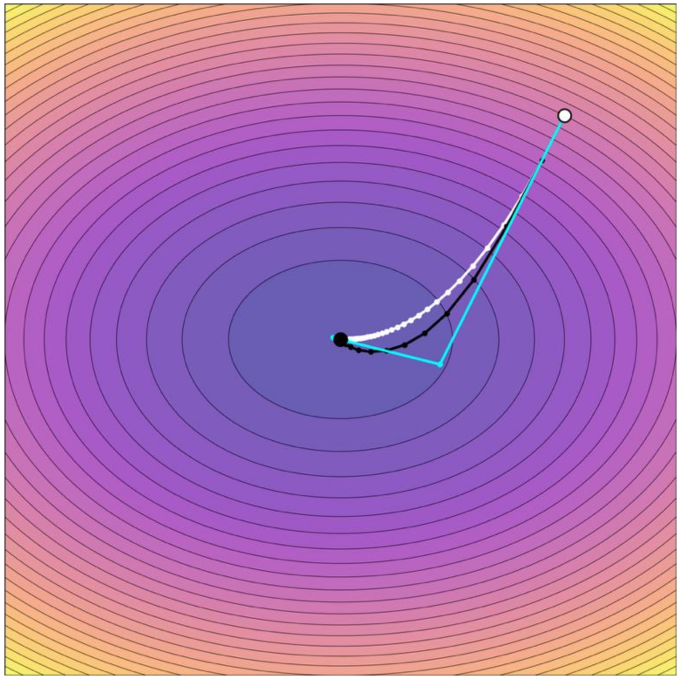
$$\boxed{\mathbf{R}_{k+1} = \mathbf{R}_k - \mathbf{B}_k \nabla E(\mathbf{R}_k)}$$

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{s}_k \mathbf{s}_k^\top}{\mathbf{s}_k^\top \mathbf{y}_k} - \frac{\mathbf{B}_k \mathbf{y}_k \mathbf{y}_k^\top \mathbf{B}_k}{\mathbf{y}_k^\top \mathbf{B}_k \mathbf{y}_k}$$

Approximate **Hessian** with finite-difference formulas based on gradients!

$$\mathbf{s}_k = \mathbf{R}_{k+1} - \mathbf{R}_k \qquad \mathbf{y}_k = \nabla E(\mathbf{R}_{k+1}) - \nabla E(\mathbf{R}_k)$$

Structural Relaxation Algorithms:



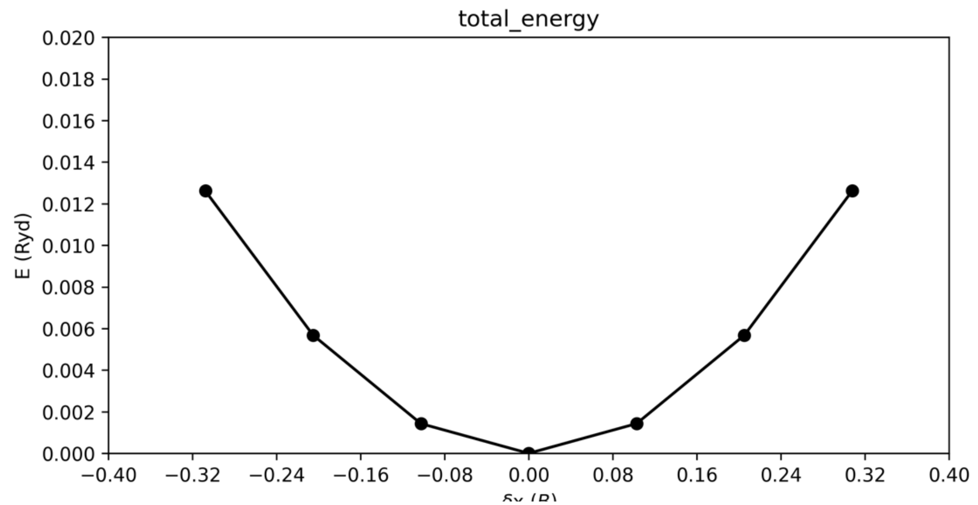
- **Steepest Descent:** Robust for initial steps if far from the minimum, but converges slowly near the minimum and can oscillate if the energy landscape has sharp features.
- **Conjugate Gradients:** Faster convergence compared to steepest descent, efficient for large systems, but requires recalculating conjugate directions periodically.
- **Broyden–Fletcher–Goldfarb–Shanno (BFGS) Algorithm:** Converges very fast near the minimum (Hessian), Efficient for small to medium-size systems, but is less robust for rough PES.

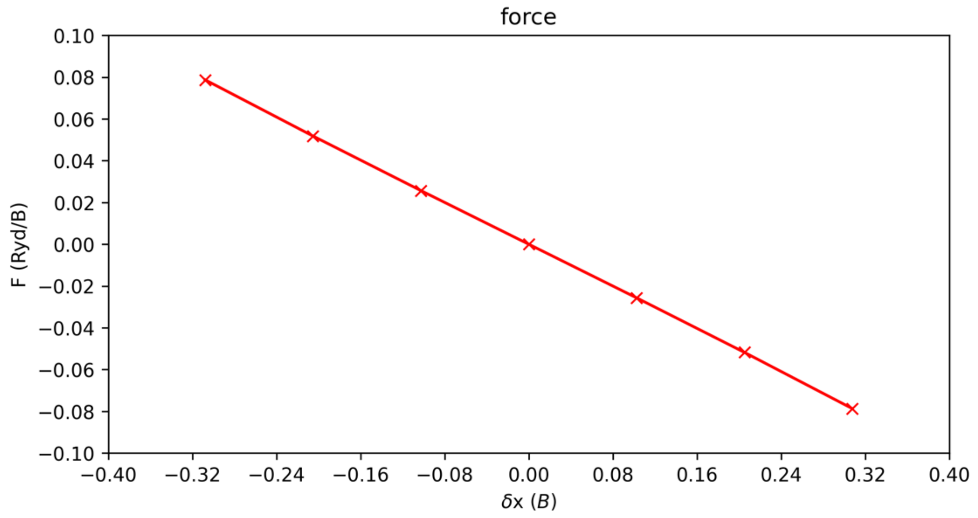
Hands-on:

1) Calculations of Forces

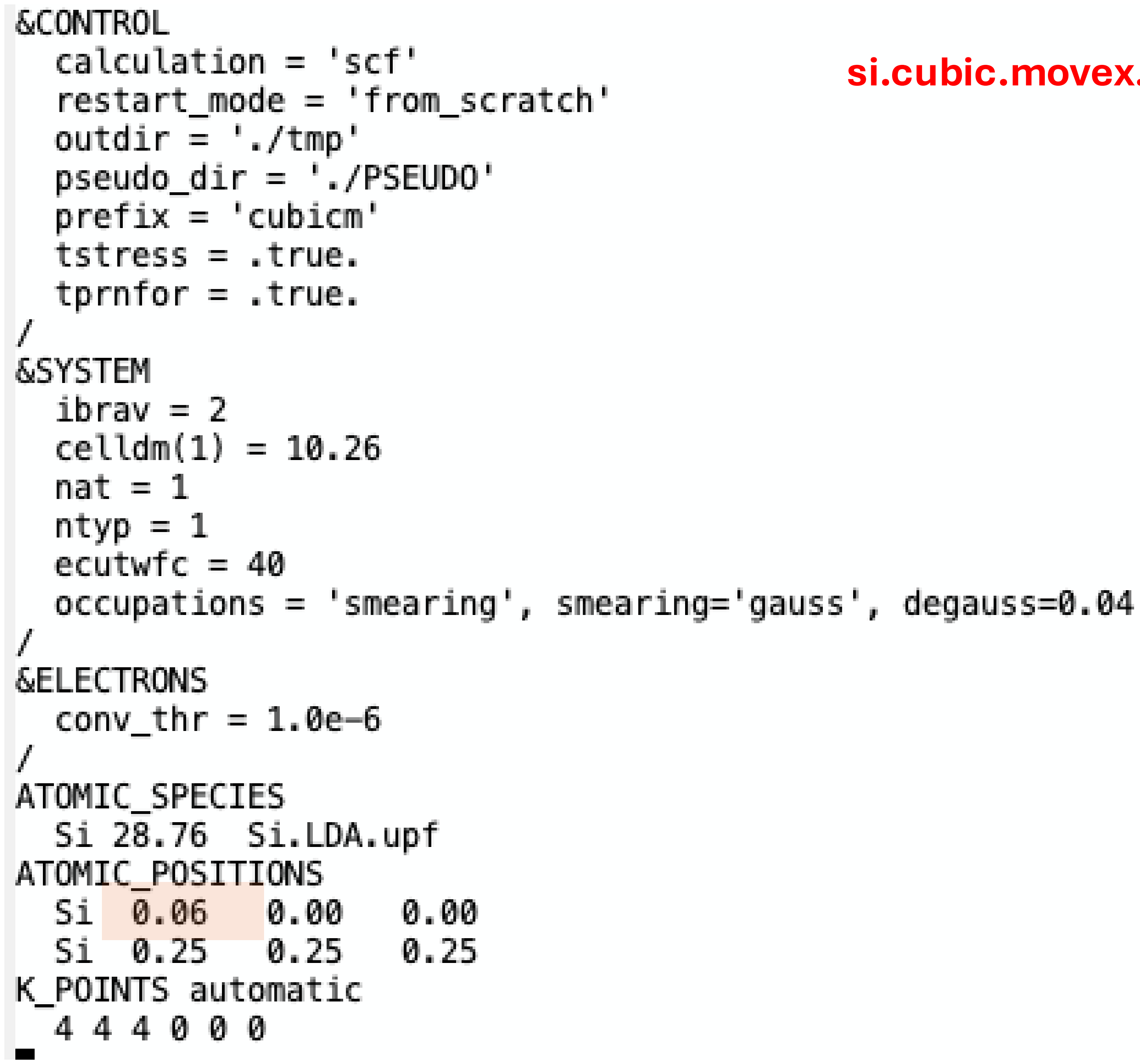
Forces by Finite Differences:

Forces can be computed directly by **shifting atoms** (by a small amount) and computing the resulting **energy difference .**



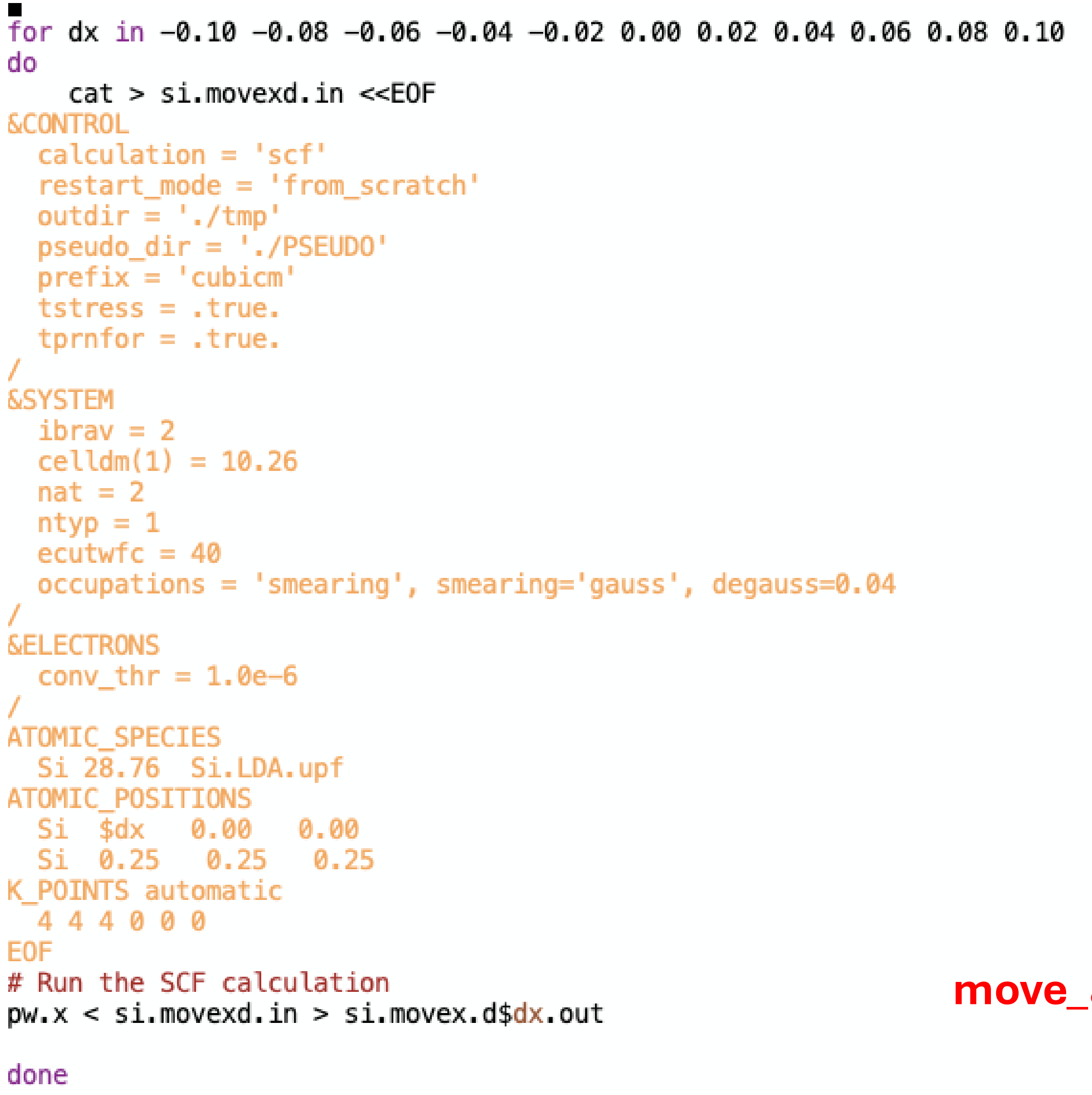


si.cubic.movex.in



```
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'cubicm'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
  ibrav = 2
  celldm(1) = 10.26
  nat = 1
  ntyp = 1
  ecutwfc = 40
  occupations = 'smearing', smearing='gauss', degauss=0.04
/
&ELECTRONS
  conv_thr = 1.0e-6
/
ATOMIC_SPECIES
  Si 28.76  Si.LDA.upf
ATOMIC_POSITIONS
  Si  0.06   0.00   0.00
  Si  0.25   0.25   0.25
K_POINTS automatic
  4 4 4 0 0 0
```

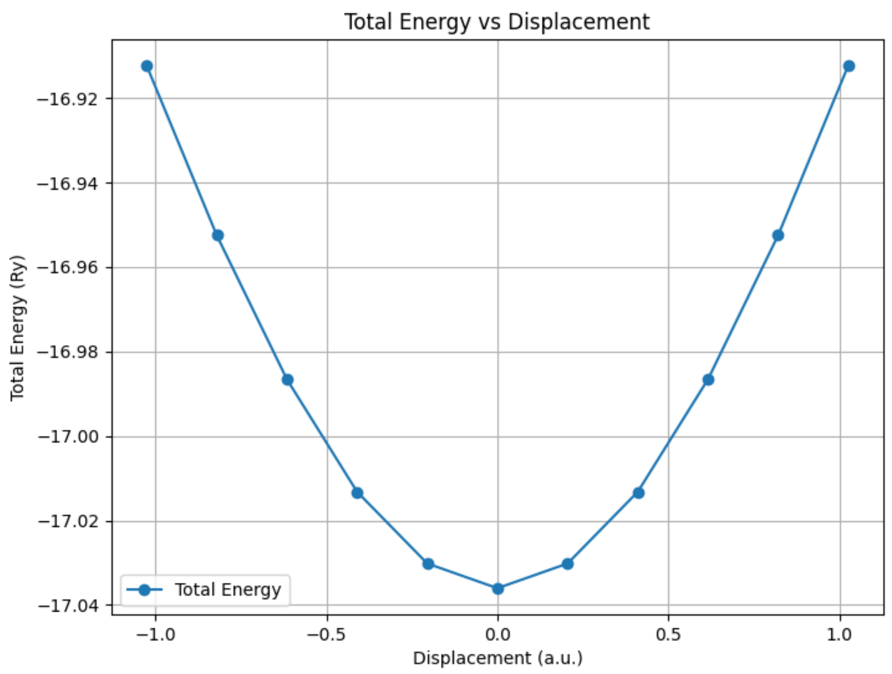
- **simple bash script** to compute the energy vs displacement ...

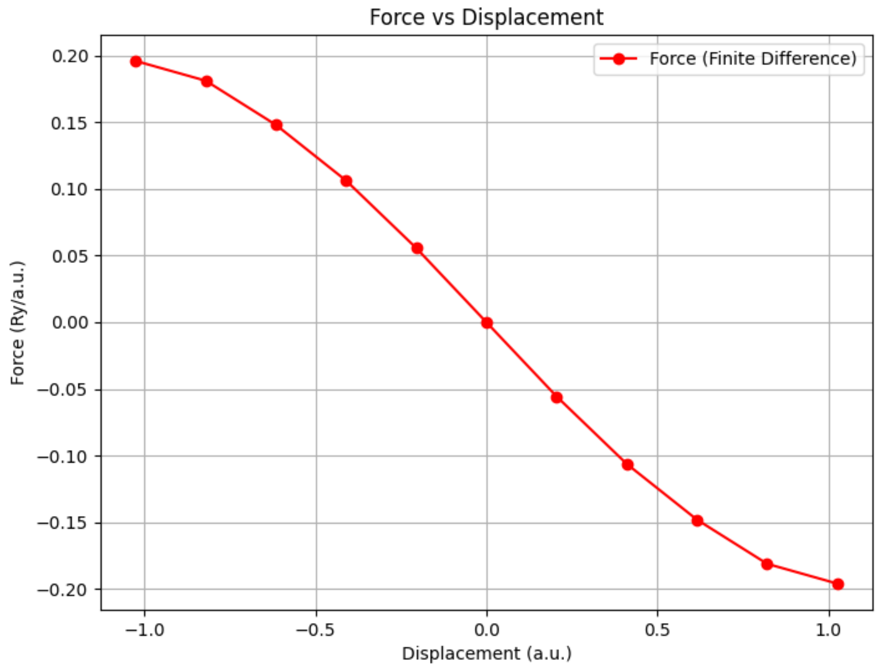


```
for dx in -0.10 -0.08 -0.06 -0.04 -0.02 0.00 0.02 0.04 0.06 0.08 0.10
do
    cat > si.movexd.in <<EOF
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'cubicm'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
  ibrav = 2
  celldm(1) = 10.26
  nat = 2
  ntyp = 1
  ecutwfc = 40
  occupations = 'smearing', smearing='gauss', degauss=0.04
/
&ELECTRONS
  conv_thr = 1.0e-6
/
ATOMIC_SPECIES
  Si 28.76  Si.LDA.upf
ATOMIC_POSITIONS
  Si  $dx    0.00   0.00
  Si  0.25   0.25   0.25
K_POINTS automatic
  4 4 4 0 0 0
EOF
# Run the SCF calculation
pw.x < si.movexd.in > si.movex.d$dx.out

done
```

move_atom.sh





Extract_energies.sh

Compute_forces.py

Hellmann-Feynmann Forces:

Quantum Espresso implements the calculation of forces using using **Hellman Feynmann's Theorem**:

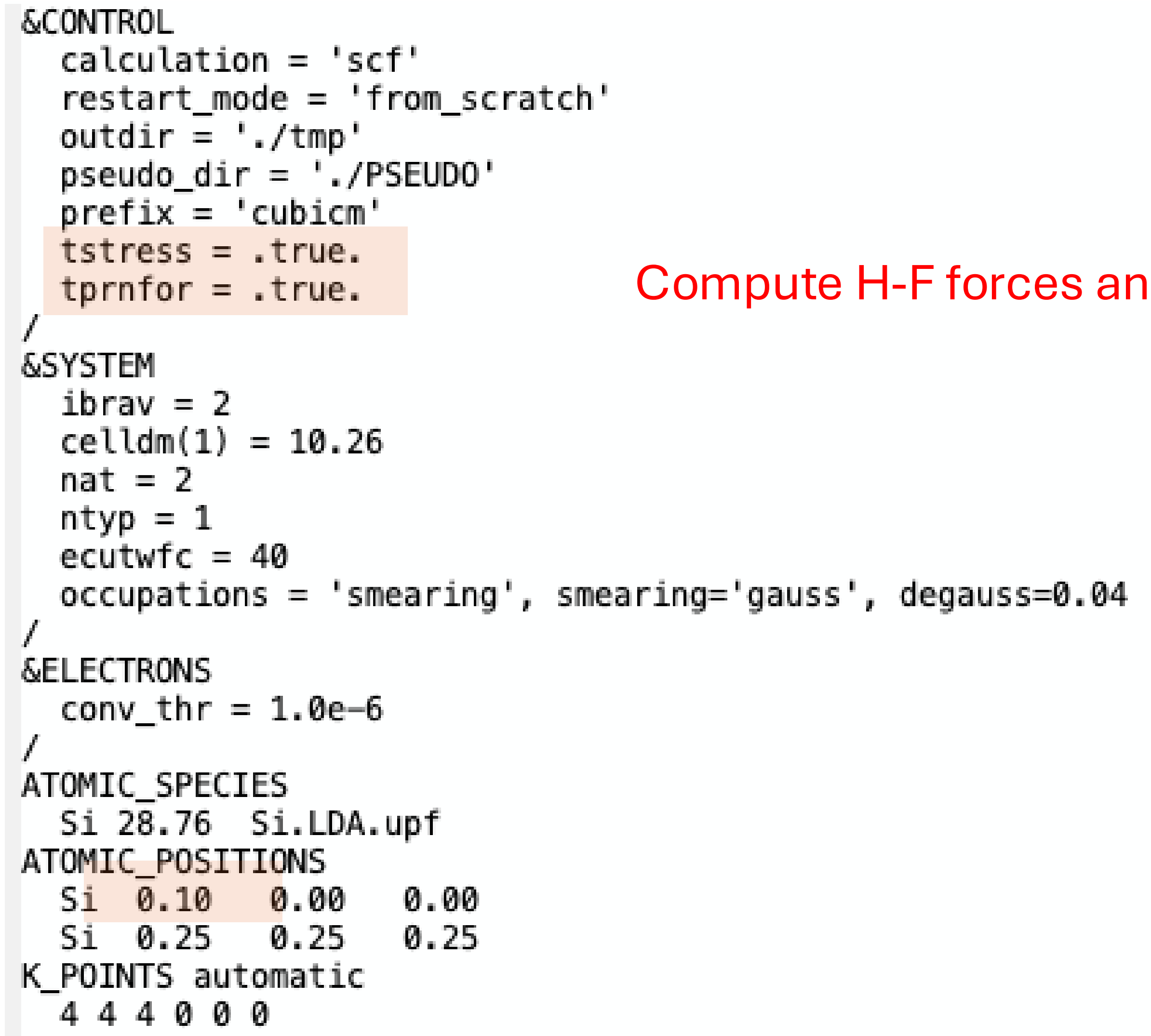
$$\boxed{\frac{\partial E}{\partial \lambda} = \left\langle \Psi_\lambda \middle| \frac{\partial H_\lambda}{\partial \lambda} \middle| \Psi_\lambda \right\rangle}$$

In particular, Forces are obtained directly from the GS average of the derivatives of the BO Hamiltonian:

$$\boxed{\mathbf{F}_\alpha = \frac{\partial E(\mathbf{R})}{\partial \mathbf{R}_\alpha} = \left\langle \Psi(\mathbf{R}) \middle| \frac{\partial H_{BO}(\mathbf{R})}{\partial \mathbf{R}_\alpha} \middle| \Psi(\mathbf{R}) \right\rangle}$$

The value of the Hellmann-Feynmann's force is written on the pwscf output file!

si.movex.d0.10.in

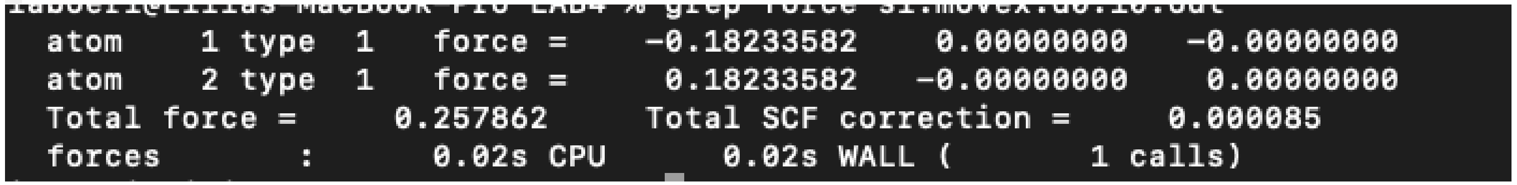


```
&CONTROL
  calculation = 'scf'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'cubicm'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
  ibrav = 2
  celldm(1) = 10.26
  nat = 2
  ntyp = 1
  ecutwfc = 40
  occupations = 'smearing', smearing='gauss', degauss=0.04
/
&ELECTRONS
  conv_thr = 1.0e-6
/
ATOMIC_SPECIES
  Si 28.76  Si.LDA.upf
ATOMIC_POSITIONS
  Si  0.10   0.00   0.00
  Si  0.25   0.25   0.25
K_POINTS automatic
  4 4 4 0 0 0
```

INPUT

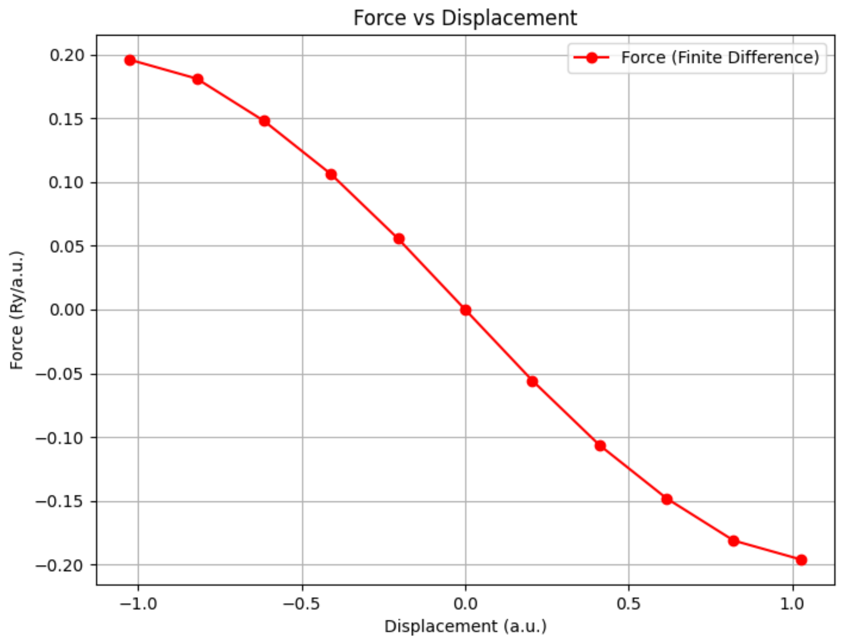
Compute H-F forces and stresses

grep force si.movex.d0.10.out yields:

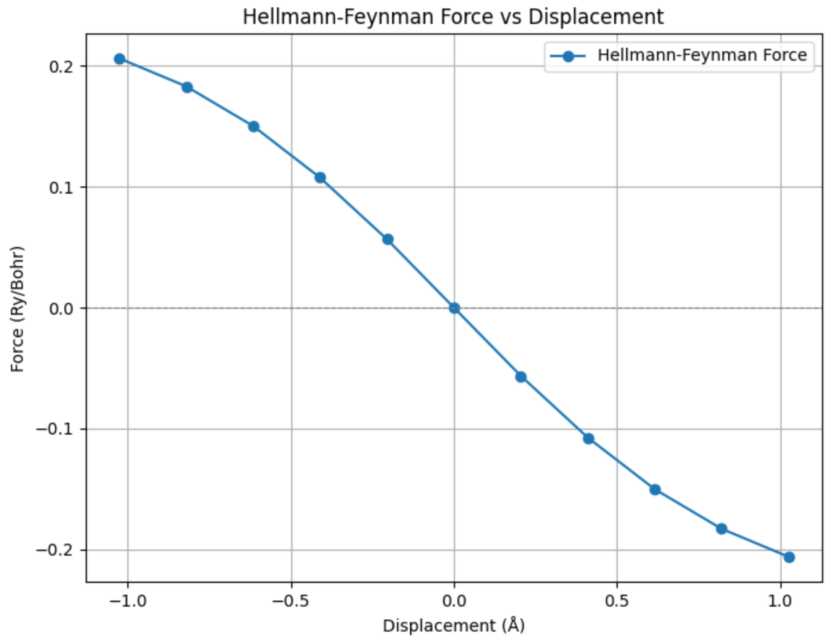


```
labocri@Lilias-MacBook-Pro LAB4 % grep force si.movex.d0.10.out
  atom    1 type  1   force =    -0.18233582    0.00000000   -0.00000000
  atom    2 type  1   force =     0.18233582   -0.00000000    0.00000000
  Total force =     0.257862     Total SCF correction =     0.000085
  forces        :      0.02s CPU      0.02s WALL (       1 calls)
```

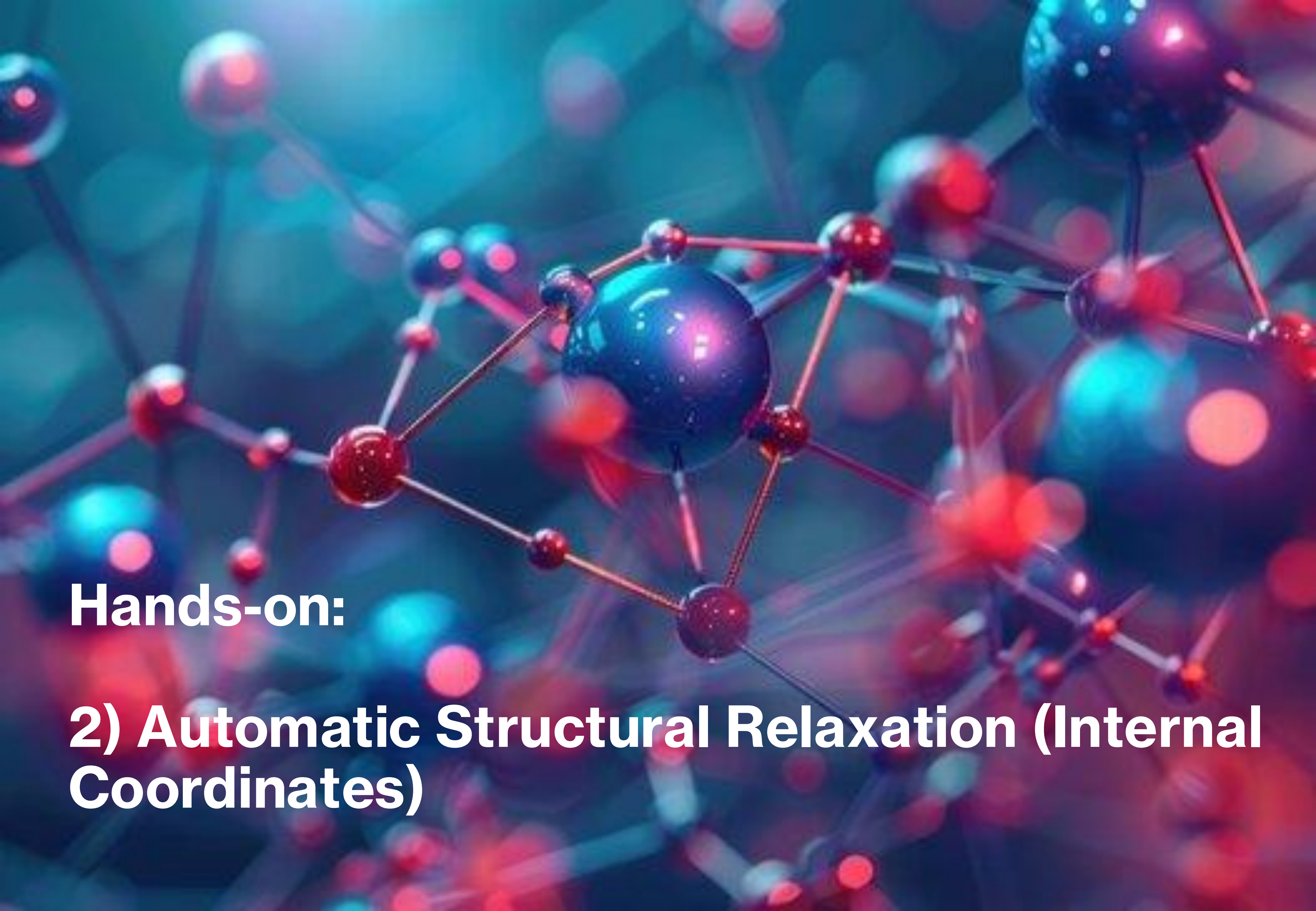
OUTPUT



Finite Differences



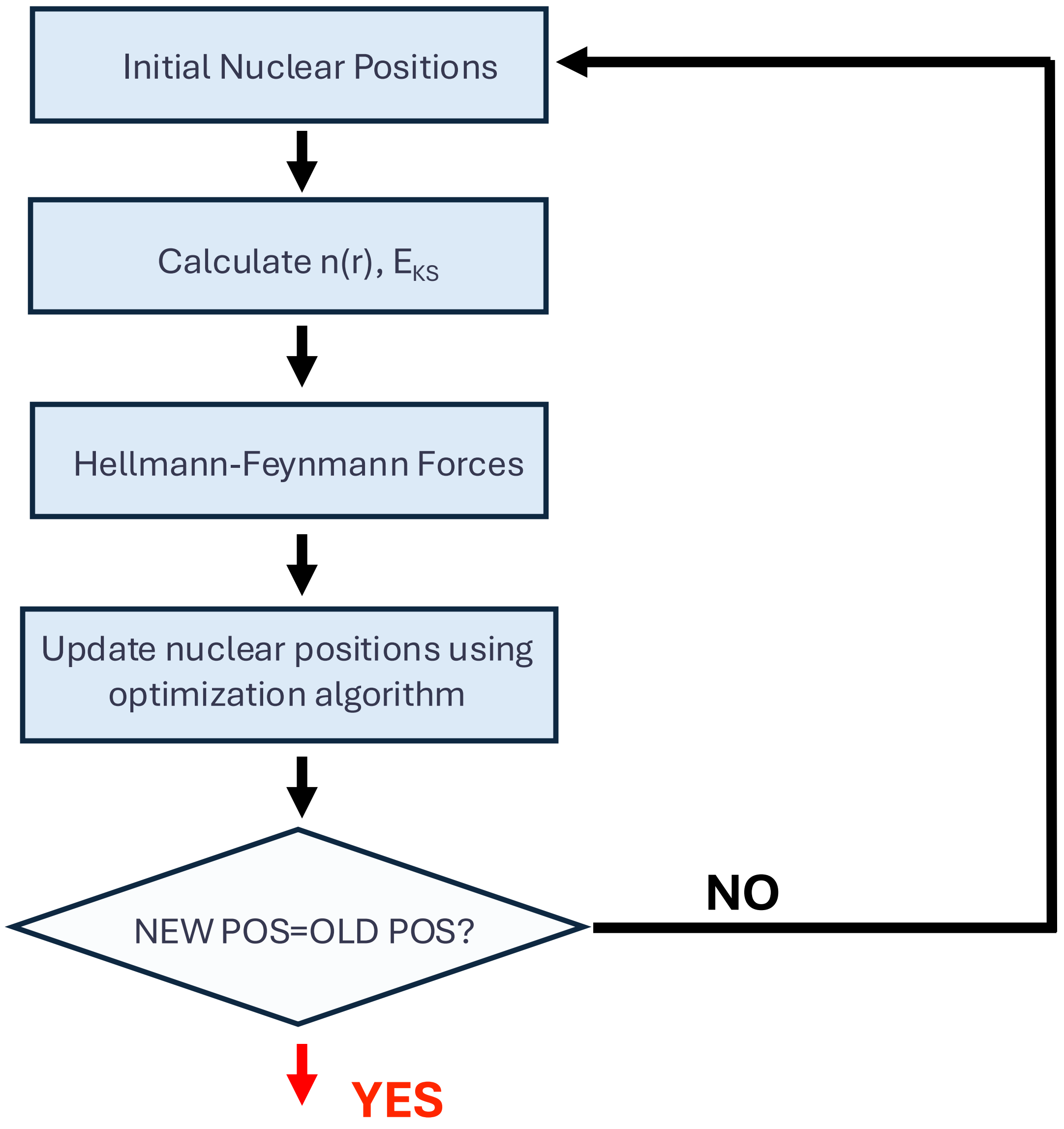
Hellmann Feynmann



Hands-on:

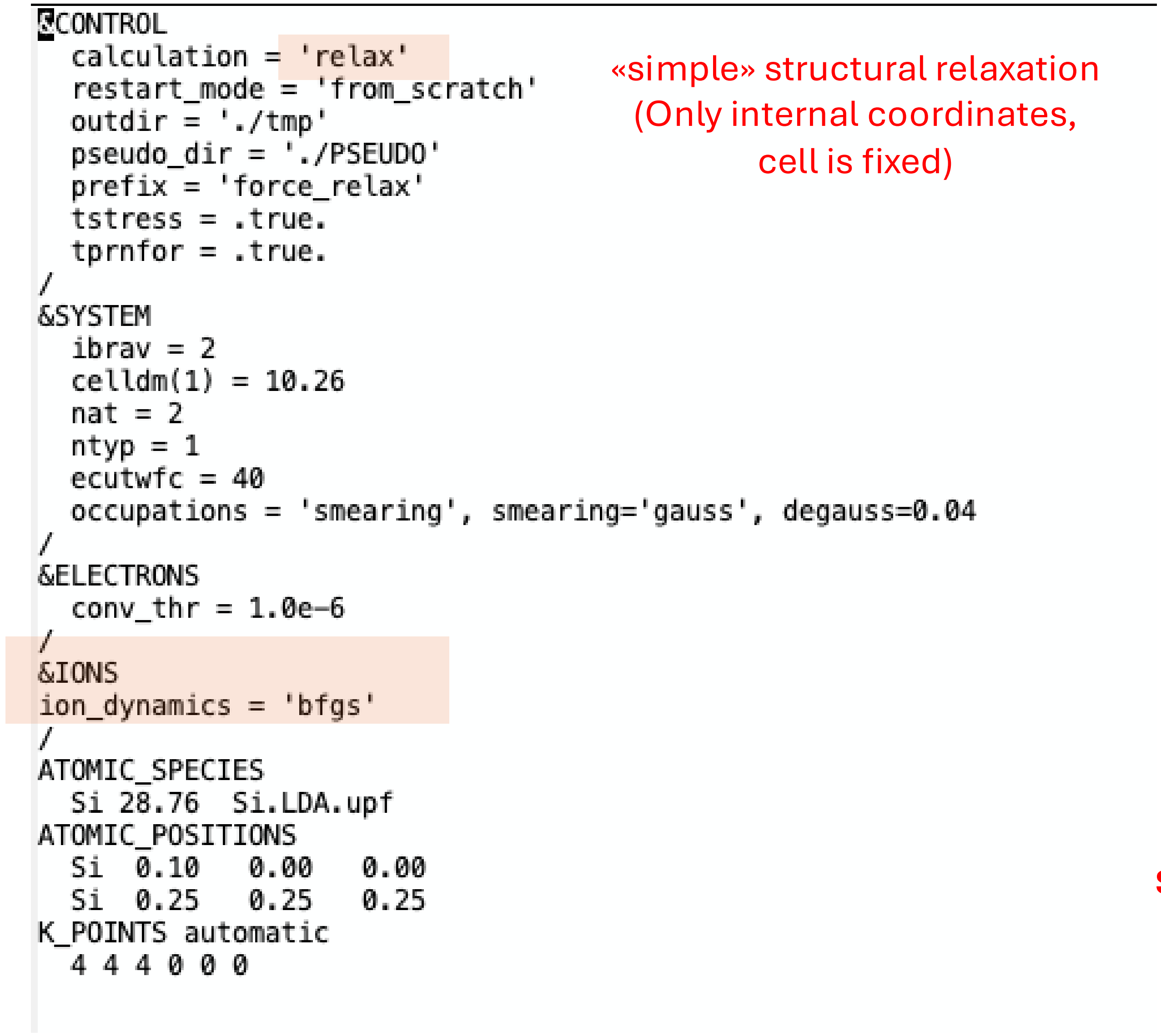
2) Automatic Structural Relaxation (Internal Coordinates)

Structural Relaxation in DFT:


Initial Nuclear Positions
Calculate $n(r)$, E_{KS}
Hellmann-Feynmann Forces
Update nuclear positions using optimization algorithm
NEW POS=OLD POS?
NO
YES

Structural Relaxation in QE:

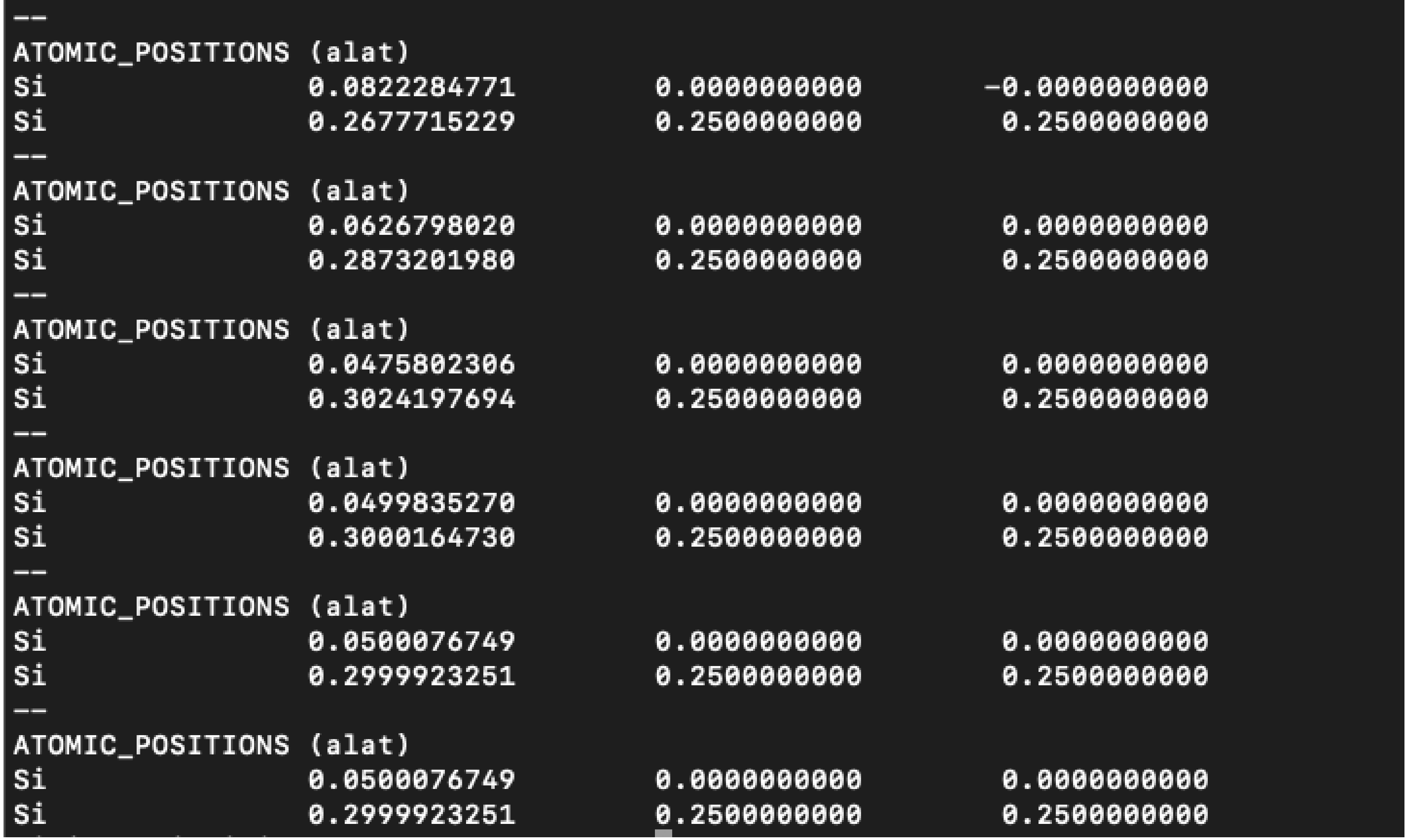
Quantum Espresso implements several algorithms for automatic structural relaxation.



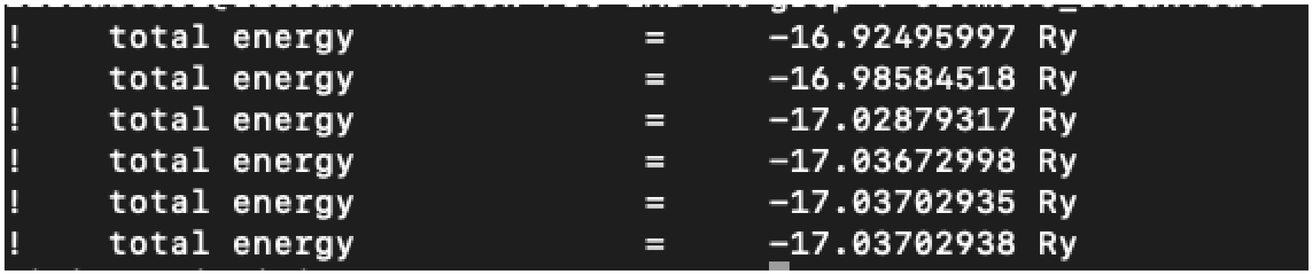
```
&CONTROL
  calculation = 'relax'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'force_relax'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
  ibrav = 2
  celldm(1) = 10.26
  nat = 2
  ntyp = 1
  ecutwfc = 40
  occupations = 'smearing', smearing='gauss', degauss=0.04
/
&ELECTRONS
  conv_thr = 1.0e-6
/
&IONS
ion_dynamics = 'bfgs'
/
ATOMIC_SPECIES
  Si 28.76  Si.LDA.upf
ATOMIC_POSITIONS
  Si  0.10   0.00   0.00
  Si  0.25   0.25   0.25
K_POINTS automatic
  4 4 4 0 0 0
```

«simple» structural relaxation
(Only internal coordinates,
cell is fixed)

si.move_relax.in



```
--
ATOMIC_POSITIONS (alat)
Si            0.0822284771       0.0000000000      -0.0000000000
Si            0.2677715229       0.2500000000       0.2500000000
--
ATOMIC_POSITIONS (alat)
Si            0.0626798020       0.0000000000       0.0000000000
Si            0.2873201980       0.2500000000       0.2500000000
--
ATOMIC_POSITIONS (alat)
Si            0.0475802306       0.0000000000       0.0000000000
Si            0.3024197694       0.2500000000       0.2500000000
--
ATOMIC_POSITIONS (alat)
Si            0.0499835270       0.0000000000       0.0000000000
Si            0.3000164730       0.2500000000       0.2500000000
--
ATOMIC_POSITIONS (alat)
Si            0.0500076749       0.0000000000       0.0000000000
Si            0.2999923251       0.2500000000       0.2500000000
--
ATOMIC_POSITIONS (alat)
Si            0.0500076749       0.0000000000       0.0000000000
Si            0.2999923251       0.2500000000       0.2500000000
```



```
!    total energy              =     -16.92495997 Ry
!    total energy              =     -16.98584518 Ry
!    total energy              =     -17.02879317 Ry
!    total energy              =     -17.03672998 Ry
!    total energy              =     -17.03702935 Ry
!    total energy              =     -17.03702938 Ry
```

Output

Practical Tips for Relaxation:

1. Starting Structure:

- Provide a reasonable starting structure to avoid excessive iterations.
- Use accurate computational settings (e.g., energy cutoffs, k-point sampling) to ensure reliability. Convergence tests must be performed on forces and not just total energies.

2. Monitor Convergence:

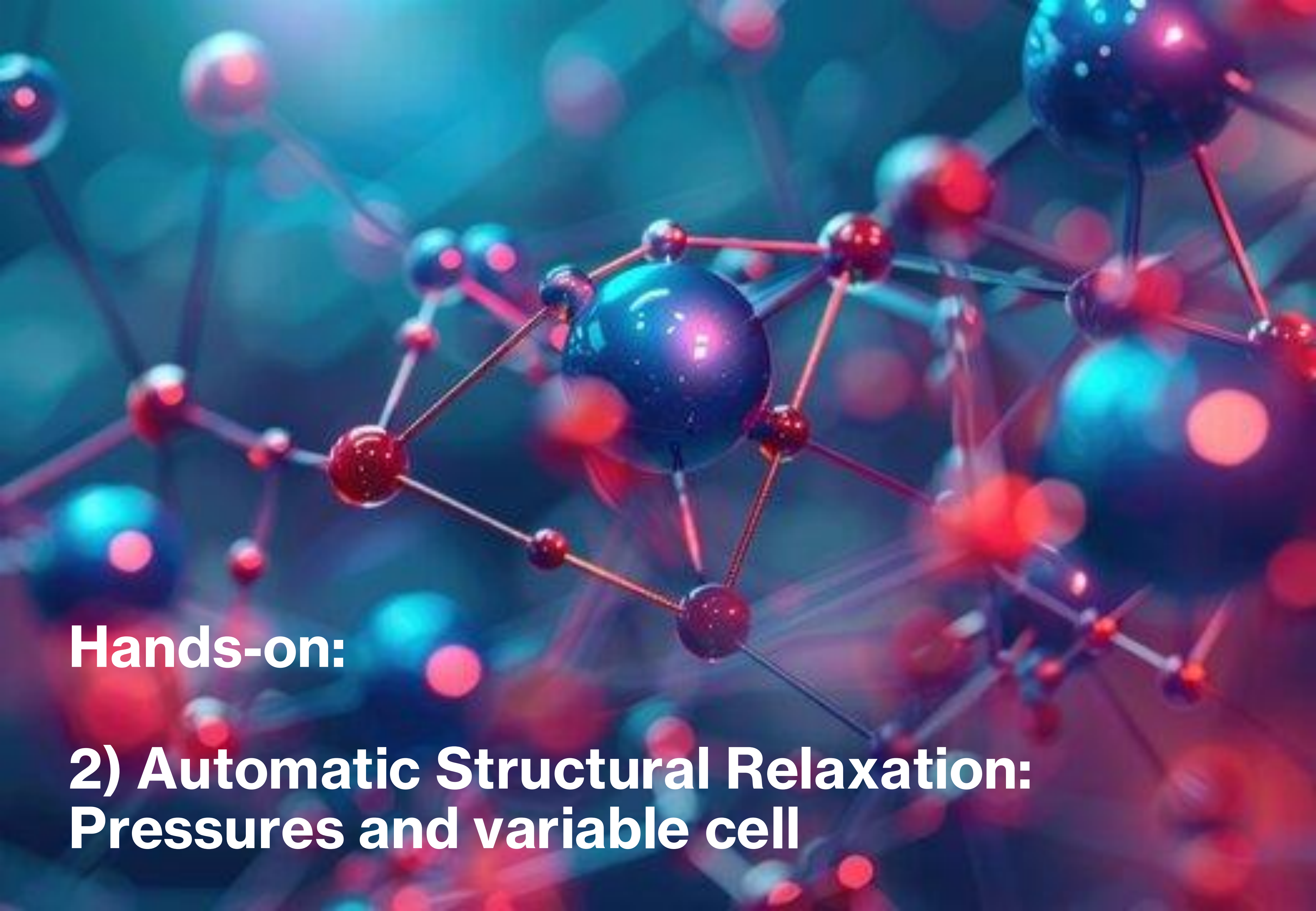
- Check forces and stress in the output after each step.
- Watch for oscillations or divergences.

3. Relaxation Parameters:

- Adjust mixing parameters and force thresholds to improve convergence.

4. Advanced Troubleshooting:

- Switch algorithms if stuck (e.g., CG to BFGS).
- Use constrained relaxation techniques to fix certain degrees of freedom (for example cell shape, volume, etc).

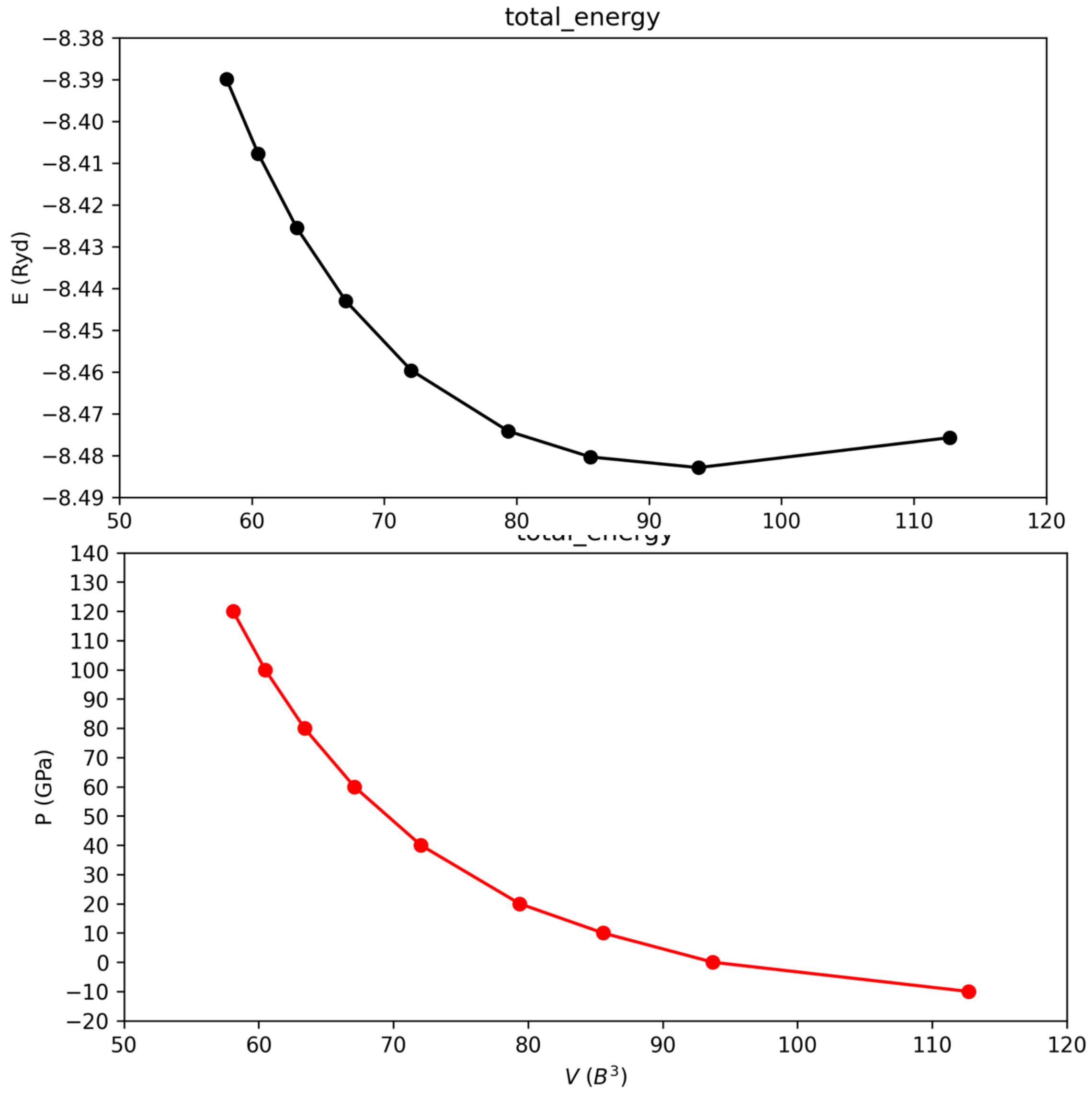


Hands-on:

2) Automatic Structural Relaxation: Pressures and variable cell

Equation of State:

The **P-V equation of state** of any solid can be computed from its **E(V) curve**:



$$\boxed{P(V) = -\frac{\partial E}{\partial V}}$$

Hellmann-Feynmann Pressure:

Using Hellman-Feynmann's theorem, QE automatically computes the **theoretical pressure** of any **crystal structure** from the volume (strain) derivatives of the total energy.

$$\boxed{\sigma_{ij} = \frac{1}{V}\frac{\partial E}{\partial \epsilon_{ij}}}$$

The Theoretical Pressure is obtained from the diagonal components of the stress tensor.

$$\boxed{P_{\text{hydro}} = -\frac{1}{3}\text{Tr}(\boldsymbol{\sigma}) = -\frac{1}{3}\left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}\right)}$$

Variable Cell Relaxation:

Once the stress tensor is known, the same structural relaxation routines used for atomic coordinates can be used to **relax the cell** parameters of a given crystal structure to reach **a target pressure.**

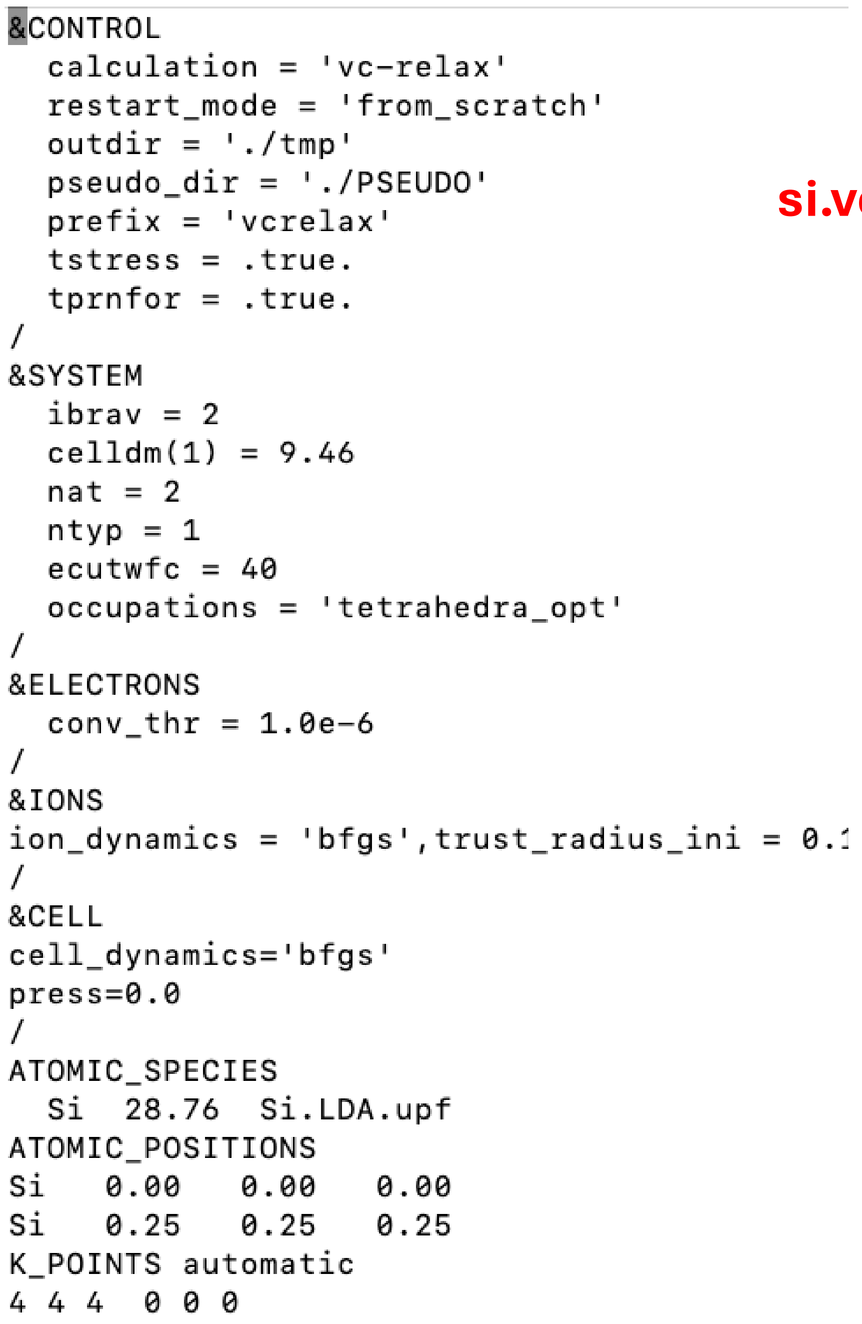
$$\boxed{\begin{cases} \mathbf{a}_i^{(k+1)} = \mathbf{a}_i^{(k)} + \Delta\mathbf{a}_i \\ \Delta\mathbf{a}_i \propto \underbrace{(\sigma_{ij} - P_{\text{target}}\delta_{ij})}_{\Delta P}\mathbf{a}_j \end{cases}}$$

The difference between the predicted and target pressure acts as an effective force at every step.

$$\boxed{\sigma_{ij} = \frac{1}{V}\frac{\partial E}{\partial \epsilon_{ij}}}$$

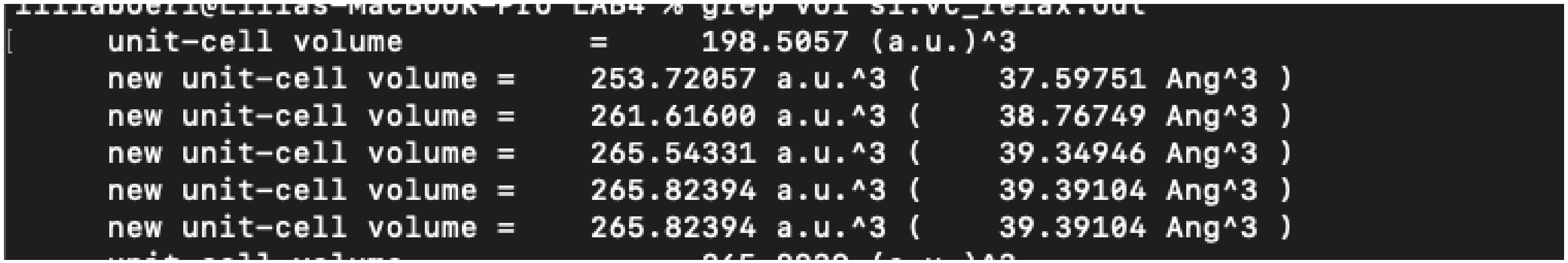
$$\boxed{P_{\text{hydro}} = -\frac{1}{3}\mathrm{Tr}(\boldsymbol{\sigma})}$$

si.vc_relax.in



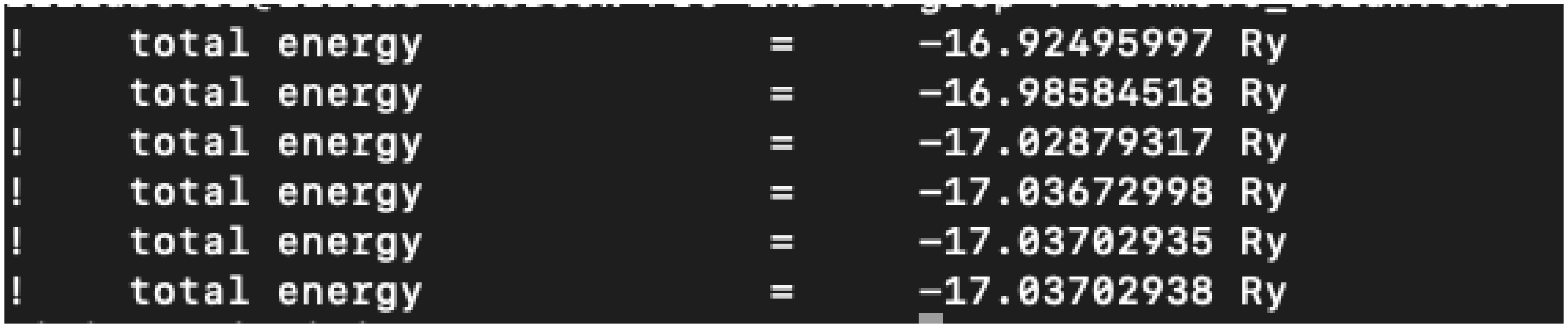
```
&CONTROL
  calculation = 'vc-relax'
  restart_mode = 'from_scratch'
  outdir = './tmp'
  pseudo_dir = './PSEUDO'
  prefix = 'vcrelax'
  tstress = .true.
  tprnfor = .true.
/
&SYSTEM
  ibrav = 2
  celldm(1) = 9.46
  nat = 2
  ntyp = 1
  ecutwfc = 40
  occupations = 'tetrahedra_opt'
/
&ELECTRONS
  conv_thr = 1.0e-6
/
&IONS
ion_dynamics = 'bfgs',trust_radius_ini = 0.
/
&CELL
cell_dynamics='bfgs'
press=0.0
/
ATOMIC_SPECIES
  Si   28.76   Si.LDA.upf
ATOMIC_POSITIONS
Si    0.00    0.00    0.00
Si    0.25    0.25    0.25
K_POINTS automatic
4 4 4  0 0 0
```

grep vol si.vc_relax.out

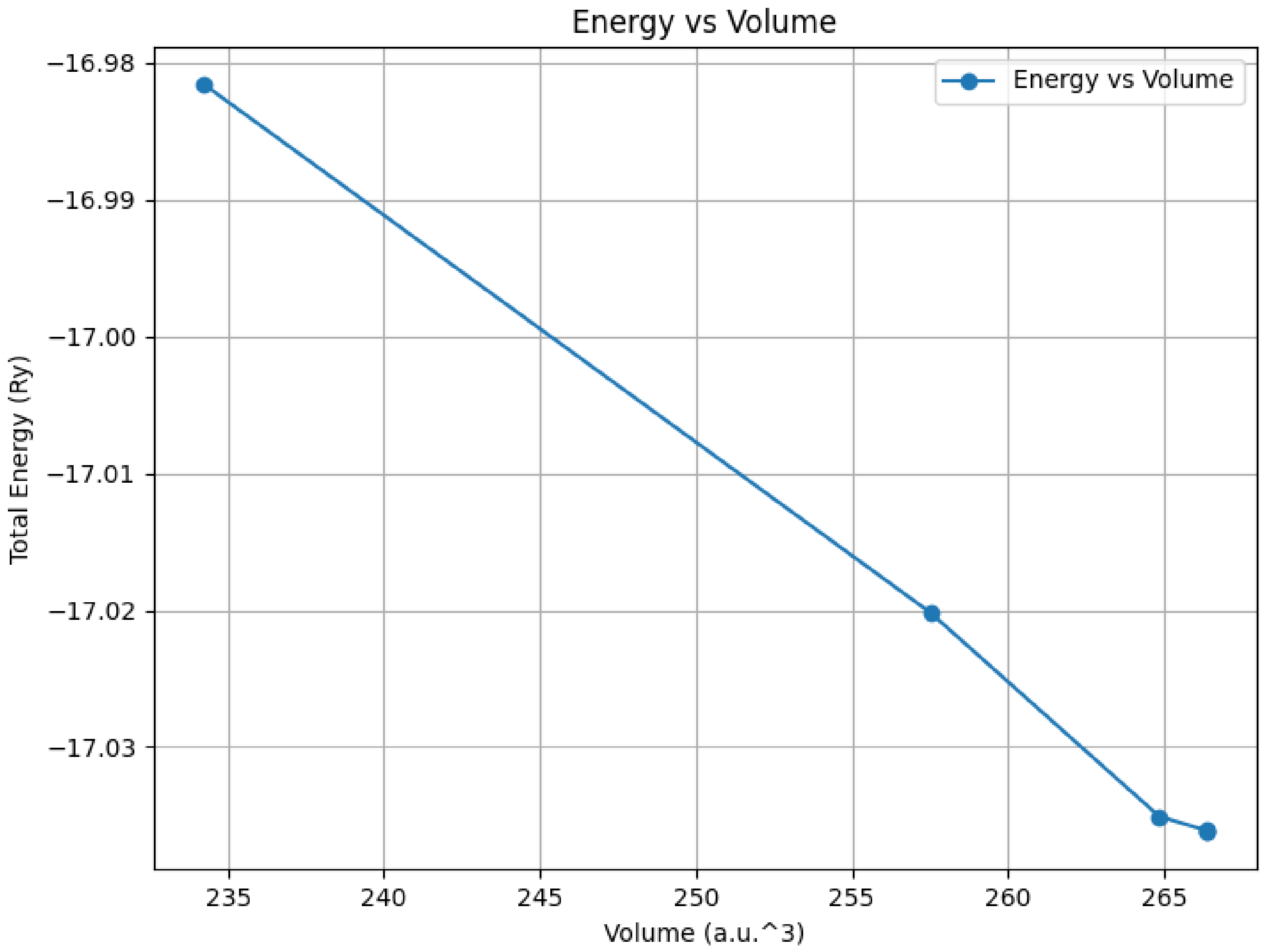


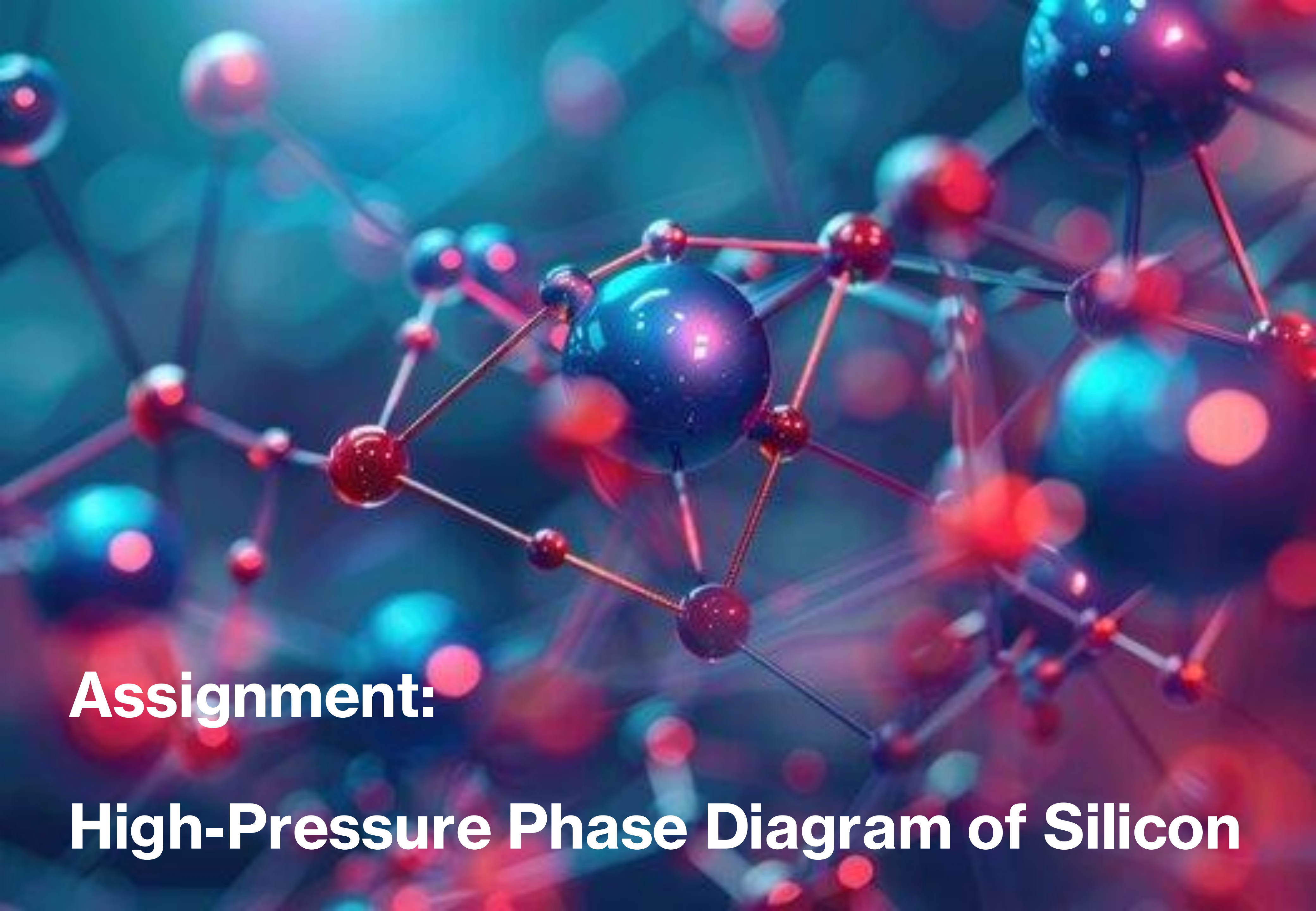
```
      unit-cell volume          =      198.5057 (a.u.)^3
      new unit-cell volume =    253.72057 a.u.^3 (    37.59751 Ang^3 )
      new unit-cell volume =    261.61600 a.u.^3 (    38.76749 Ang^3 )
      new unit-cell volume =    265.54331 a.u.^3 (    39.34946 Ang^3 )
      new unit-cell volume =    265.82394 a.u.^3 (    39.39104 Ang^3 )
      new unit-cell volume =    265.82394 a.u.^3 (    39.39104 Ang^3 )
```

grep ! si.vc_relax.out



```
!    total energy              =     -16.92495997 Ry
!    total energy              =     -16.98584518 Ry
!    total energy              =     -17.02879317 Ry
!    total energy              =     -17.03672998 Ry
!    total energy              =     -17.03702935 Ry
!    total energy              =     -17.03702938 Ry
```

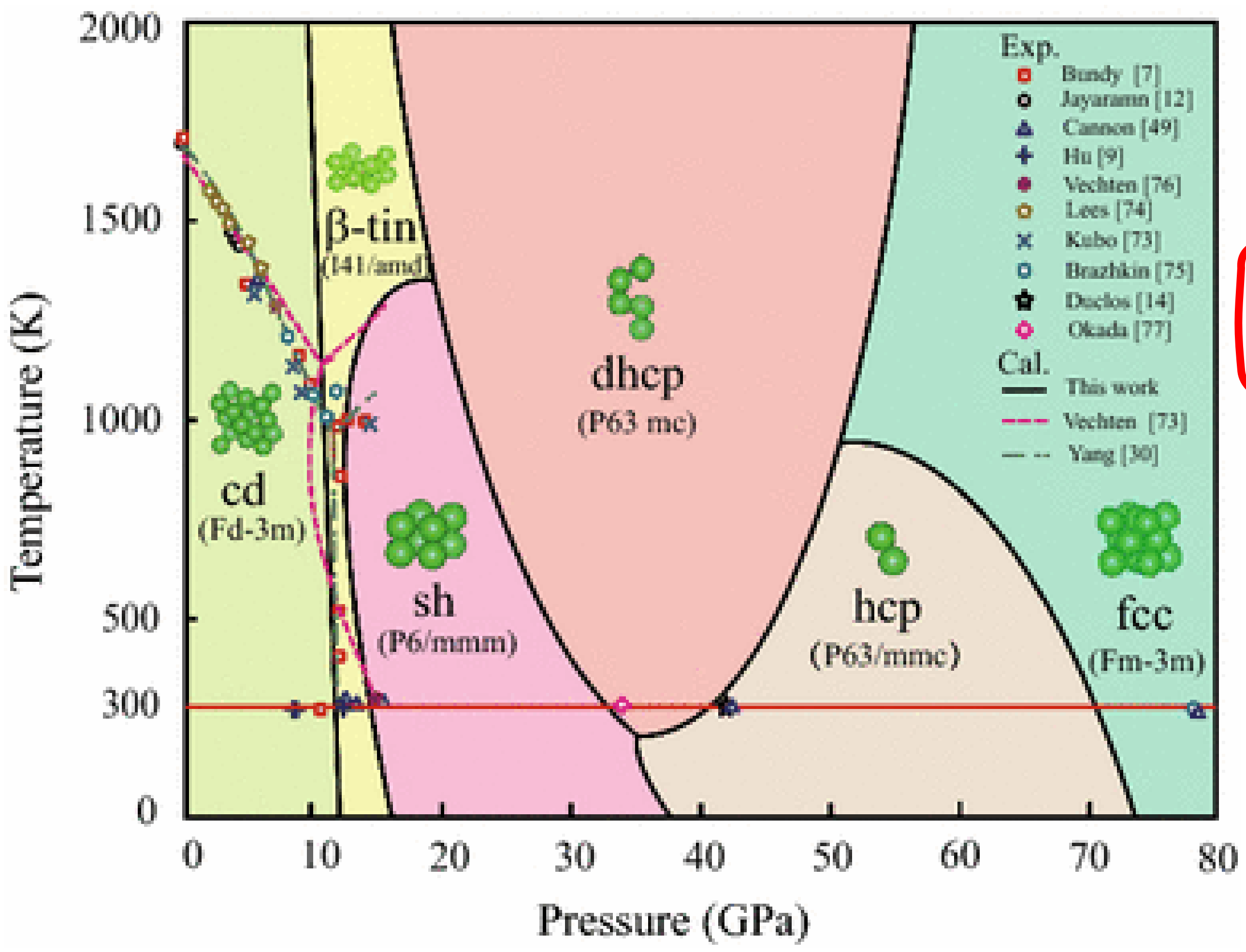

Energy vs Volume
Energy vs Volume
Total Energy (Ry)
Volume (a.u.^3)
−16.98
−16.99
−17.00
−17.01
−17.02
−17.03
235
240
245
250
255
260
265



Assignment:

High-Pressure Phase Diagram of Silicon

Calculation of Solid-Solid Phase Transitions:



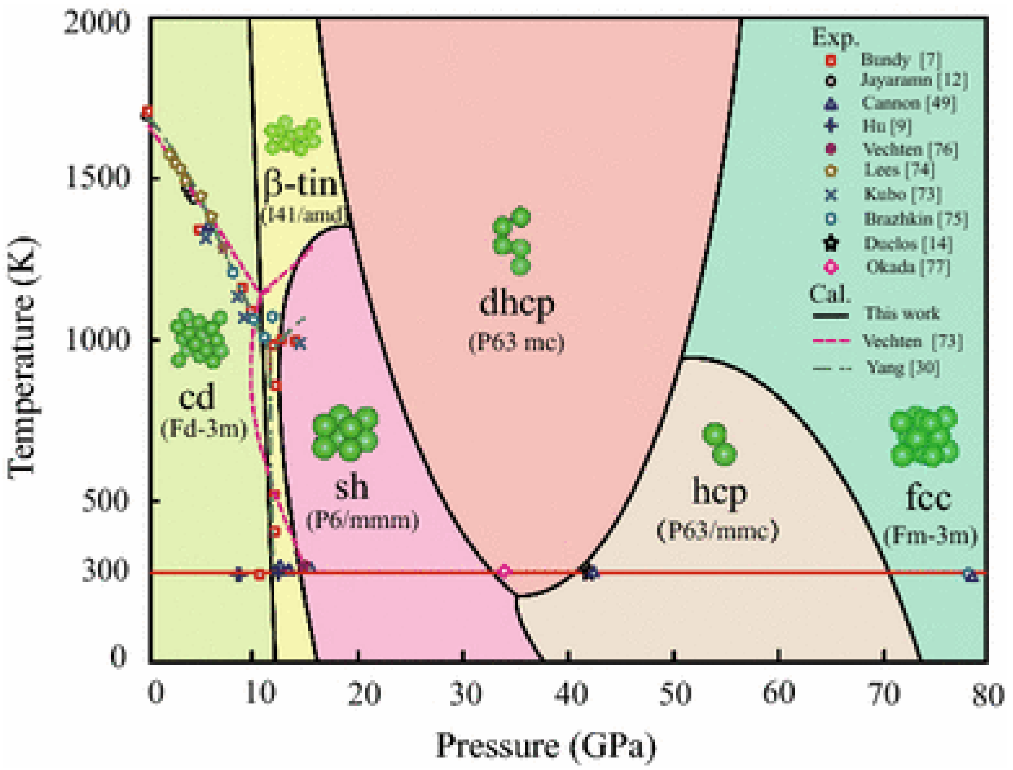
Enthalpy:

$$\boxed{H(P) = E + PV}$$

Transitions between solid phases as a function of pressure can be predicted **comparing the DFT enthalpies (/f.u.) of the two phases at different pressures**. The phase with the lowest enthalpy is the most stable one at a given pressure.

High-Pressure phase diagram of Silicon

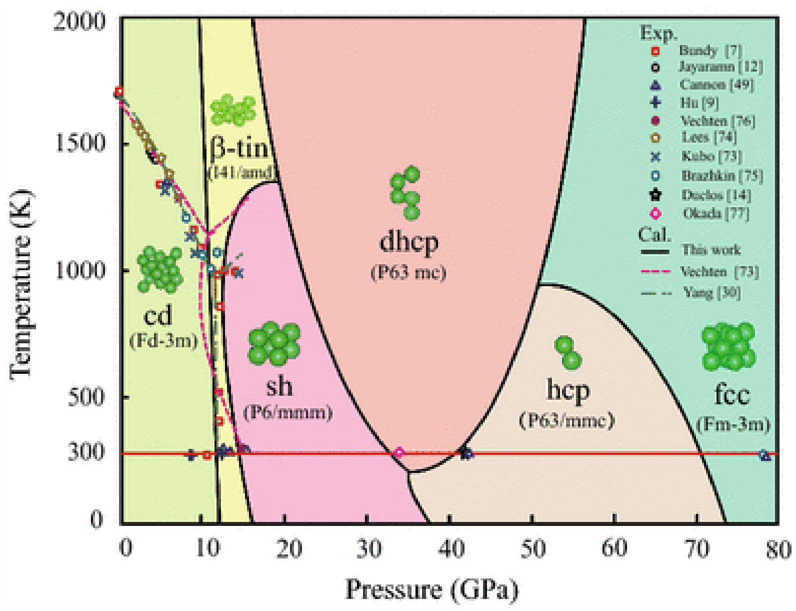
The High-pressure phase diagram of silicon is relatively complex (see below). We will consider only the two extrema, i,e. the cd (cubic diamond structure) and the simple fcc phase.
Relaxing the two structures at intermediate pressure, estimate the pressure for the transition cd -> fcc. Are the two phases metallic or insulating at ambient (0) and High (100 Gpa) pressure?



High-Pressure phase diagram of Silicon

Assigment:

The High-pressure phase diagram of silicon is relatively complex (see below). We will consider only the two extrema, i,e. the cd (cubic diamond structure) and the simple fcc phase.
Relaxing the two structures at intermediate pressure, estimate the pressure for the transition cd -> fcc. Are the two phases metallic or insulating at ambient (0) and High (100 Gpa) pressure?



In practice:

1) Setup input files for silicon in the fcc and cubic diamond (CD) Structure.

NB: Since we don't know whether the result is a metal or a semiconductor/insulator, we will treat it as a metal!

2) Converge parameters (Ecut, # kpts for both structures).

3) Relax both structures (fcc, CD) at zero and 100 GPa.

NB: Since 100 Gpa is a huge structure, relax to an intermediate pressure first!

4) Compute Energies/Enthalpies for both structures.

5) Run Band Structure/DOS calculations to answer the last question.