Chemistry & Materials Science with the Amsterdam Modeling Suite 2019

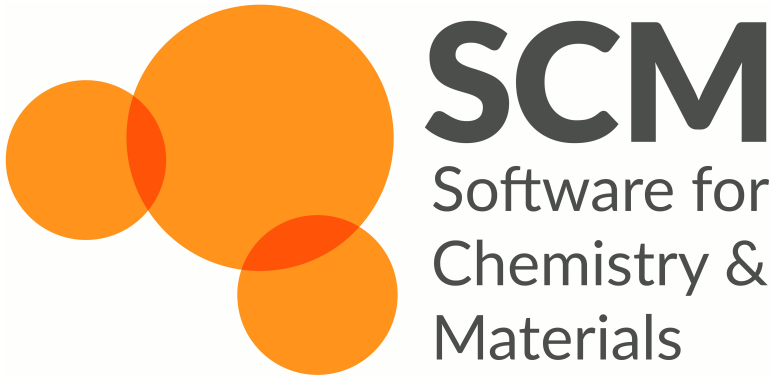
Hands-On Session Part 2 – Periodic Systems

Thomas M. Soini

Virtual Winter School
on Computational Chemistry
February 2020

Overview

- Handling periodic systems
- Hands-on exercises
  - **Electronic structure of PdS_2 slab models**
  - **Mechanical properties of polymers**: see example below & advanced tutorial "*Mechanical properties of epoxy polymers*"



Periodic Systems

- Import structures
- Prepare slab models

Crystal Structures

- Build from scratch in GUI:
  periodic atomistic editor activates when using a periodic engine
- Amsterdam Modeling Suite template library: Access via  in GUI
- Load structure file via **`File → Import Coordinates…`**
  - From previous input (.adf) or output (.rkf) files
  - AMS internal format (i.e. xyz-file + lattice vectors)
  - .cif files from external sources like:
    - *crystallography.net*
    - *materialsproject.org*
    - article supporting information etc.

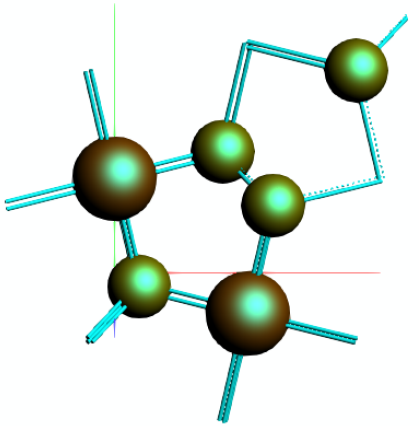
PdS_2 Crystal

- Preoptimized PdS_2 structure: import file

  `PdS2.cif` via **`File → Import Coordinates…`**
- Useful tools for periodic systems:

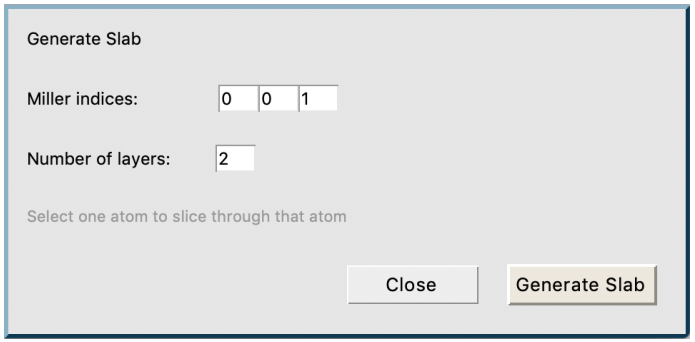
  `Edit → Crystal →`
  - E.g. map atoms back to unit cell:

    `Edit → Crystal → Map Atoms to (0..1)`

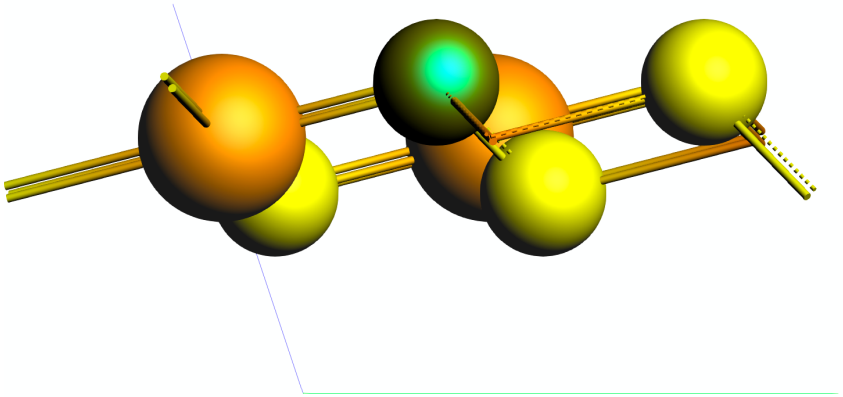


PdS_2 Slabs

- Create slab model:
  - **`Edit → Crystal → Generate Slab…`**



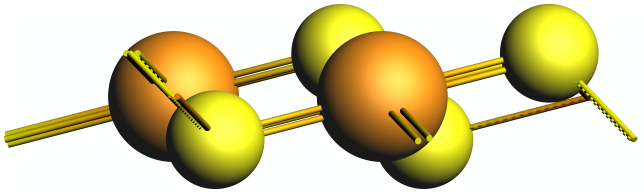
  - Press **`ctrl`** + **`1`** to change perspective and select atom **`S(4)`**

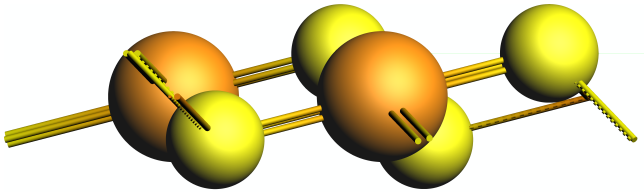


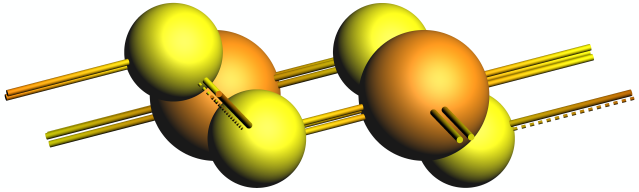
  - Click **`Generate Slab`**
  - **`File → Save as →`** Enter **`PdS2_Bilayer.adf`**

PdS_2 Slabs

- Repeat with **`Number of layers:  1`**







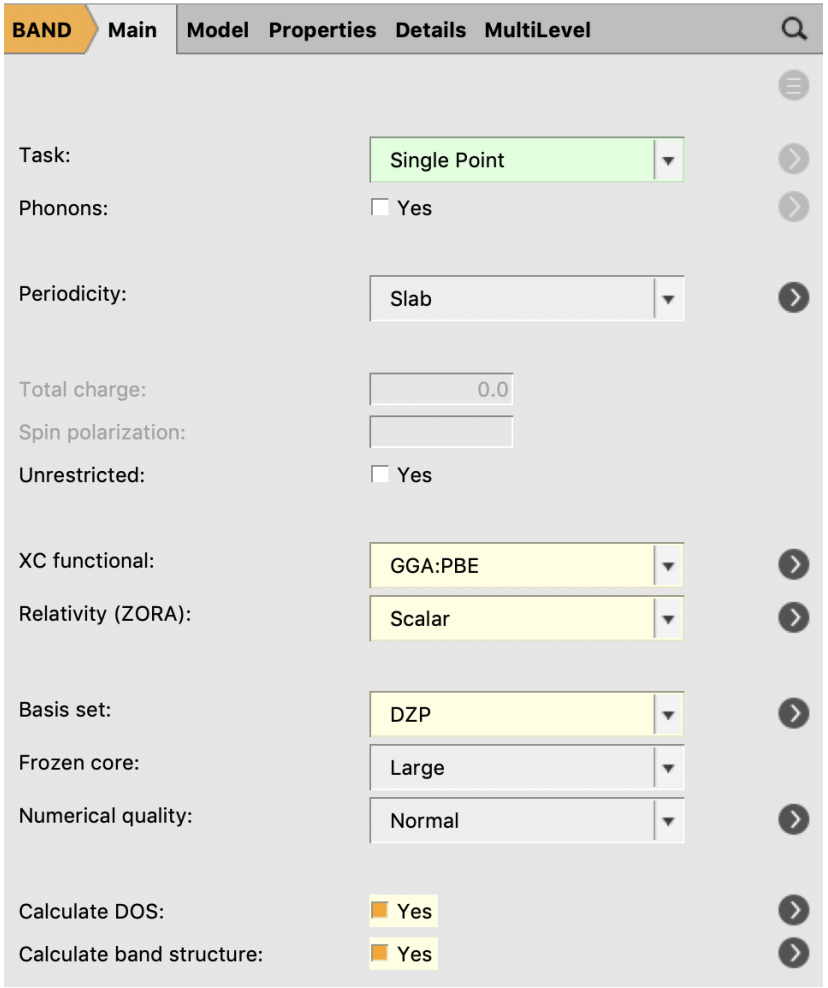
- **`File`** → **`Save as`** → Enter **`PdS2_Monolayer.adf`**

Band Structures

- Settings
- Band structure program
- DOS

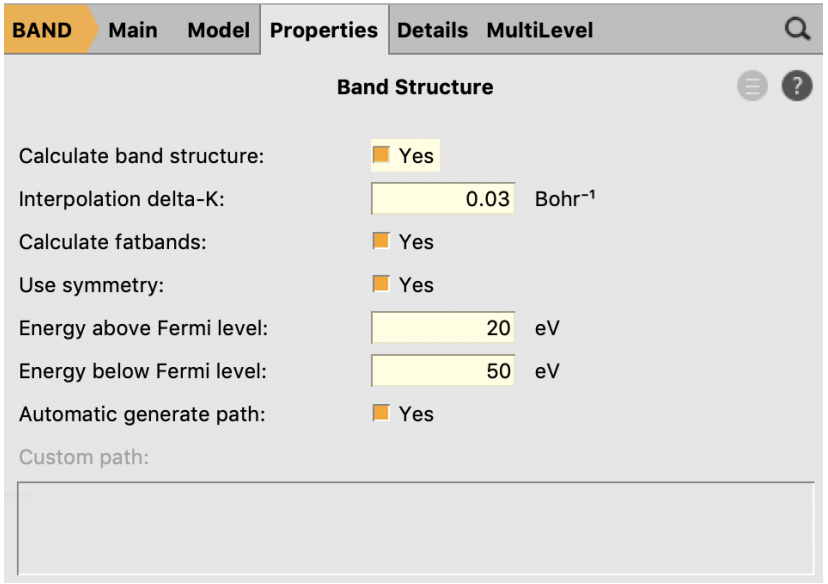
Band Structure Settings

- `Task:` → `Single Point`
- `XC functional` → `GGA` → `PBE`
- `Relativity (ZORA)` → `Scalar`
- `Basis Set:` → `DZP`
- Tick `Calculate DOS`
- Tick `Calculate band structure`
  - Click on ❯ next to it



Band Structure Settings

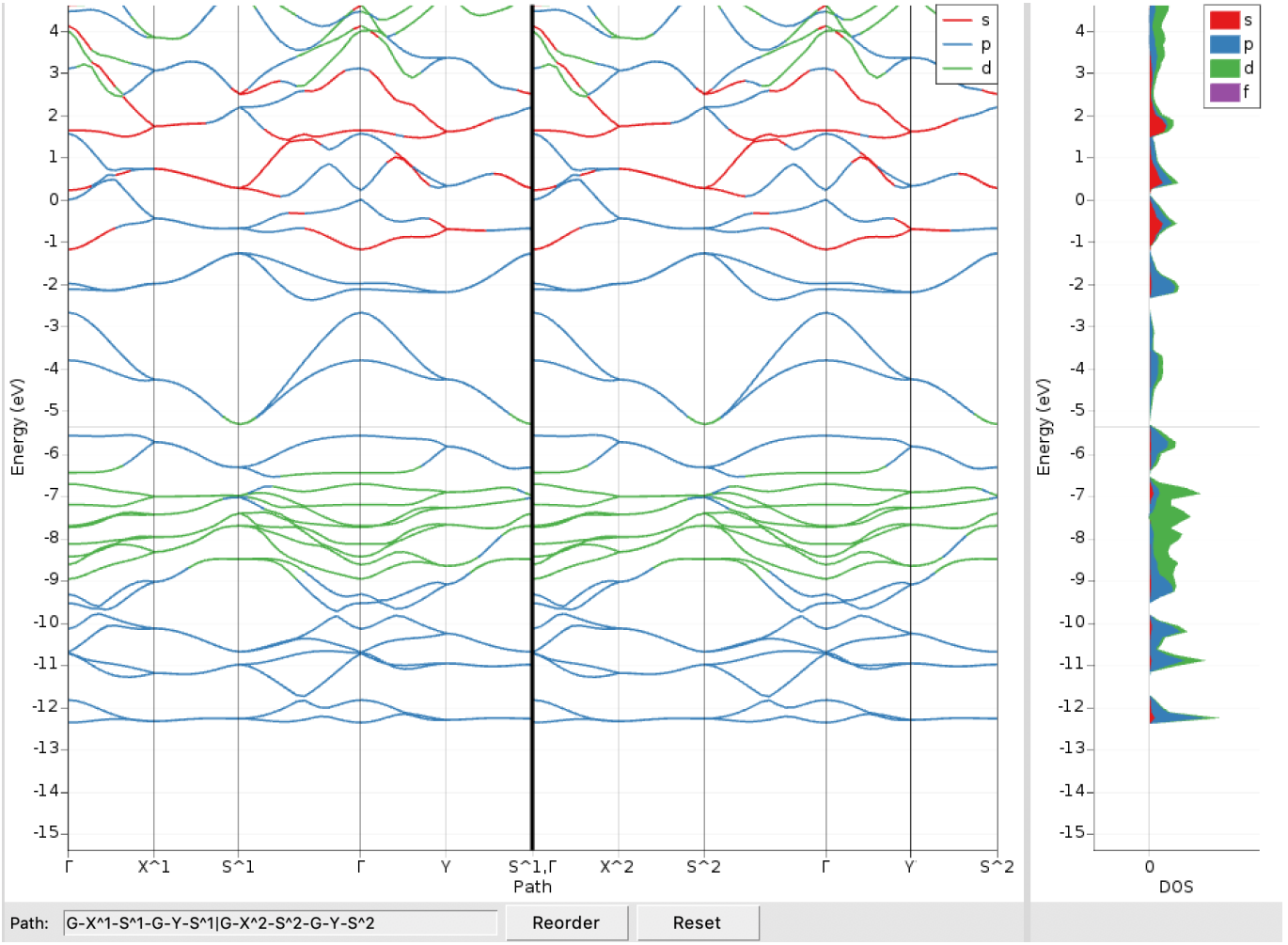
- `Interpolation delta-K: → 0.03`
- `Energy above Fermi level: 20 eV`
- `Energy below Fermi level: 50 eV`



- Repeat for both systems and run calculations with **`File → Run`**

Band Structure Analysis

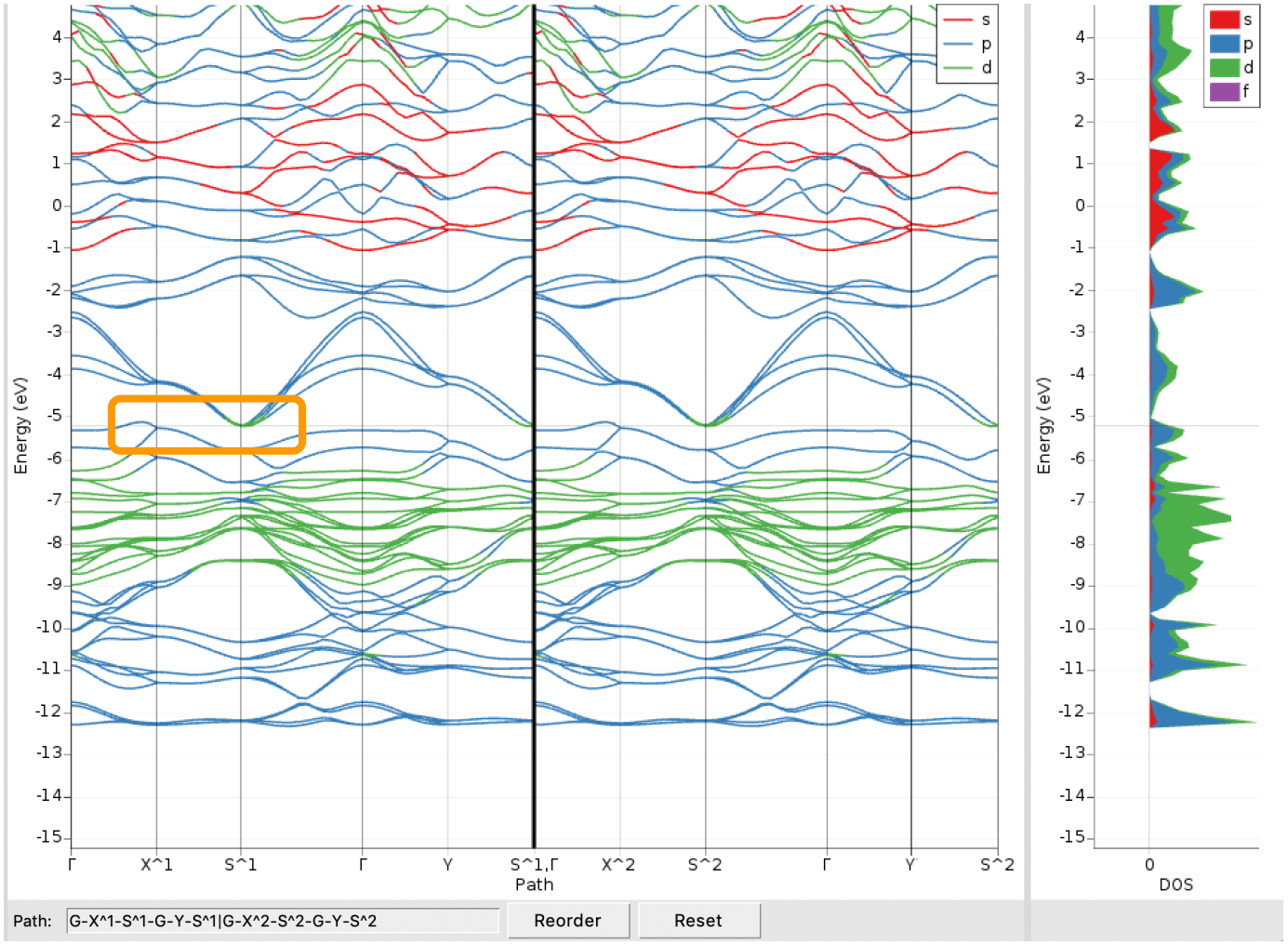
- **`ADFjobs`** opens
- After calculations finish
  - SCM → **`Band Structure`**

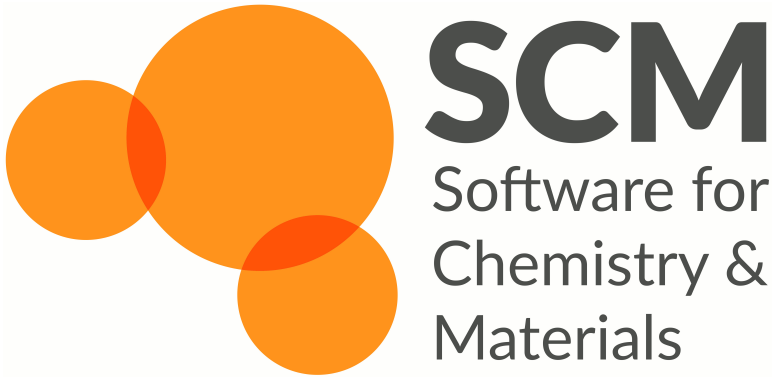


Band Structure Analysis

- Comparing both band structures:
  - Semiconducting monolayer
  - Conducting bilayer

"A Single-Material Logical Junction Based on 2D Crystal PdS2"
M. Ghorbani-Asl, A. Kuc, P. Miró, T. Heine, Adv. Materials 28 (2016)



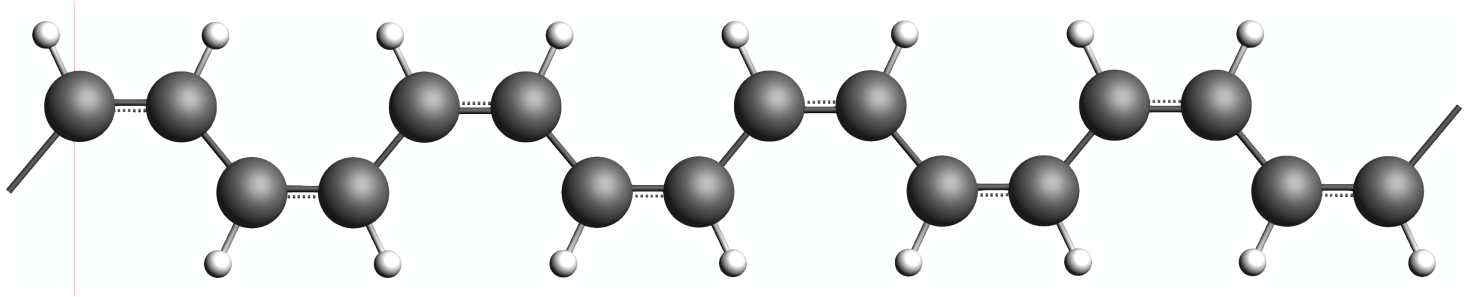


Mechanical Properties

- Small polymer toy system
- Stress-strain analysis
- Online version here
- Full-blown realistic tutorial

Example

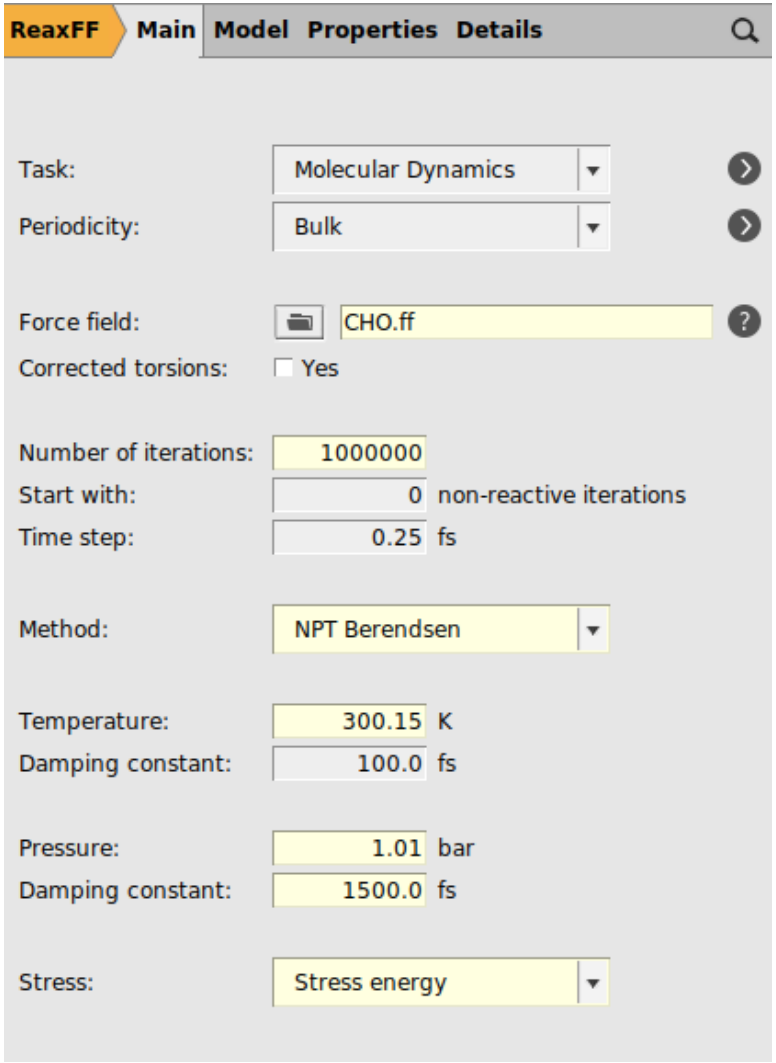
- **cis-Polyacetylene**
  - 3D periodic model in 19×19×19 Å (required by ReaxFF)
  - **`File → Import Coordinates…`**



  - Alternatively: Build from scratch (demo)
- Strain in z-direction: induces transition to trans-Polyacetylene
- Even more strain: polymer chain snaps

Setup MD Calculation

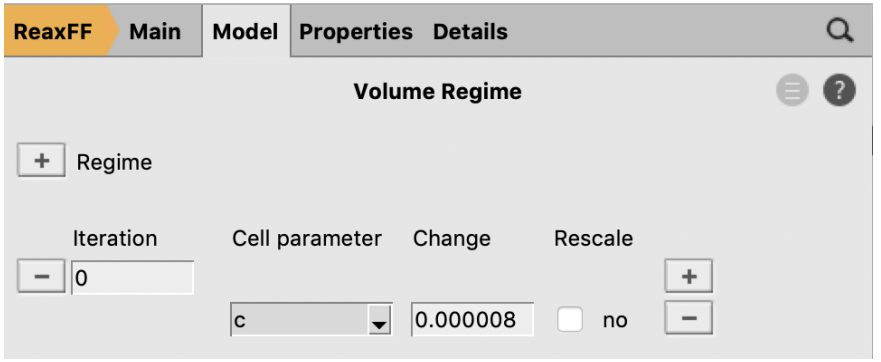
- SCM → `New Input`
- `File → Import Coordinates…`
- Pick `cis_PA.xyz`
- Select `ReaxFF`
  - `Force Field: CHO.ff`
  - `Number of iterations: 1000000`
  - `Method: NPT Berendsen`
  - `Temperature: 300.15 K`
  - `Pressure: 0.101 MPa`
  - `Damping constant: 1500.0 fs`
  - `Stress: Stress energy`



Strain Rate

- **`Details → Molecular Dynamics`**
  - **`Fix cell parameters (NPT only): c`**
  - Output frequency: **`KF result file: 2000`** (saves disk space)

- **`Model → Volume Regime`**
  - Click on `+`
  - **`Cell parameter:`** select **`c`**
  - **`Change: 0.000008`**



- See Advanced Tutorial on www.scm.com for calculation of strain rate

Run & Evaluation

- **`File → Save`** & **`File → Run`**
- Use **`ADFmovie`** to see polymer change under strain
- After calculation is completed
  - Open console (via **`Help → Terminal`** or **`Command Line`**)
  - **`startpython stress_strain_curve.py JOBNAME.rxkf`**
- Plot **`stress_zz`** component against **`strain_z`**, e.g.:
  - Type **`gnuplot`**
  - **`plot "stress-strain-curve.csv" using 3:6`**