GEOMETRY OPTIMIZATION

In this tutorial, you will learn how to perform a geometry optimization with `FHI-vibes` for fcc Silicon. At the same time, you will also learn a little bit about the internal settings in `FHI-aims`. Note that you can find the full manual for `FHI-aims` as PDF in `/home/triqs/FHI_local/FHI-aims/`.

Define Inputs

FHI-vibes workflows typically rely on two input files:

1) a **geometry file**, which specifies the atomic positions and cell vectors.

2) a **control file**, which contains the calculator section and defines the computational tasks and all the numerical parameters.

To kick off our first relaxation, we'll use the `geometry.in` file for bulk Silicon in the primitive cell.

The calculator section is contained in the file `calculator.in`. Let's firstly walk through these settings once:

- [files]
  - `geometry: geometry.in`: read the input geometry from `geometry.in`.
- [calculator]
  - `name: aims` means that `FHI-aims` will be used.
  - `socketio: true` means that the socketIO communication will be used. This will speed up the computation by reusing some information from one evaluation to the next.
- [calculator.parameters]
  - `xc: pw-lda` means that the pw-LDA exchange-correlation functional will be used.
  - `compute_forces: true` means that forces will be computed.

- [calculator.kpoints]
  - `density: 2` use a k-point density of at least 2 per $\mathring{\text{A}}^{-1}$
- [calculator.basissets]
  - `default: light` : use light default basis sets and numerical settings for silicon. By default, `FHI-aims` provides a series of different pre-configured numerical sets and basis sets for all elements of the periodic table. The here chosen `light` settings are ideal to obtain qualitatively correct results in a short time. Fully converged and more accurate results can be obtained by using `intermediate` , `tight` , or even `really_tight` settings. In addition, it is always possible to customize the basis sets for the actual system of interest to better balance speed and accuracy.

We will use these settings defined in `calculator.in` for most of the rest of these tutorials. Let's first make a copy for the relaxation step:

```
cp calculator.in relaxation.in
```

Next, use the command line interface (CLI) of `FHI-vibes` to obtain default settings for performing the relaxation and appending them to the input file:

```
vibes template relaxation >> relaxation.in
```

This has now added an additional section for the relaxation settings to your **control file**. Accordingly, the updated input file `relaxation.in` should look like this:

```
[files]
geometry:                       geometry.in

[calculator]
name:                           aims
socketio:                       True

[calculator.parameters]
xc:                             pw-lda
compute_forces:                 true

[calculator.kpoints]
density:                        2

[calculator.basissets]
default:                        light

[relaxation]
driver:                         BFGS
fmax:                           0.001
unit_cell:                      True
fix_symmetry:                   False
hydrostatic_strain:             False
constant_volume:                False
scalar_pressure:                0.0
decimals:                       12
symprec:                        1e-05
workdir:                        relaxation

[relaxation.kwargs]
maxstep:                        0.2
logfile:                        relaxation.log
restart:                        bfgs.restart
```

The settings file template you just generated contains all the necessary settings to set up and run a geometry optimization with `FHI-vibes` using `FHI-aims` as the force/stress calculator. `FHI-vibes` will perform a BFGS optimization of the structure as implemented in Atomic Simulation Environment (ASE). As the keywords suggest, a full unit cell relaxation will be performed, no symmetry constraints will be applied, and the relaxation will stop once the maximum force acting on the atoms is smaller than 0.001 eV/Å. More details on these keywords can be found in the documentation.

Run calculation

You can start an interactive calculation with `vibes run relaxation`. We suggest to

`pipe` the output, e.g., like this:

```
vibes run relaxation | tee log.relaxation
```

`FHI-vibes` will create a working directory with the default name `relaxation` and will handle running the `FHI-aims` calculations necessary to perform the geometry optimization. The log file `relaxation.log` should read like that:

```
[vibes.run]    run relaxation workflow with settings from relaxation.in

  [relaxation]   ** /draco/u/christia/Codes/vibes_v2/tutorials/GR/relaxa
  [calculator]   Update aims k_grid with kpt density of 3 to [8, 8, 8]
  [calculator]   .. add `sc_accuracy_rho: 1e-06` to parameters (default)
  [calculator]   .. add `relativistic: atomic_zora scalar` to parameters
  [calculator]   .. add `compensate_multipole_errors: False` to paramete
  [calculator]   .. add `output_level: MD_light` to parameters (default)
  [calculator]   Add basisset `light` for atom `Si` to basissets folder.
  [calculator]   Calculator: aims
  [calculator]   settings:
  [calculator]     xc: pw-lda
  [calculator]     compute_forces: True
  [calculator]     k_grid: [8, 8, 8]
  [calculator]     sc_accuracy_rho: 1e-06
  [calculator]     relativistic: atomic_zora scalar
  [calculator]     compensate_multipole_errors: False
  [calculator]     output_level: MD_light
  [calculator]     compute_analytical_stress: True
  [calculator]     use_pimd_wrapper: ('localhost', 10011)
  [calculator]     aims_command: /u/christia/Codes/vibes_v2/run_aims.sh
  [calculator]     species_dir: /draco/u/christia/Codes/vibes_v2/tutorial
  [relaxation]   filter settings:
  [relaxation]     hydrostatic_strain: False
  [relaxation]     constant_volume: False
  [relaxation]     scalar_pressure: 0.0
  [relaxation]   driver: BFGS
  [relaxation]   settings:
  [relaxation]     type: optimization
  [relaxation]     optimizer: BFGS
  [relaxation]     maxstep: 0.2
  [socketio]     Use SocketIO with host localhost and port 10011
  [relaxation]   filter settings:
  [relaxation]     hydrostatic_strain: False
  [relaxation]     constant_volume: False
  [relaxation]     scalar_pressure: 0.0
  [relaxation]   Start step 0
  [relaxation]   Step 0 finished.
  [relaxation]   .. residual force:  0.000 meV/AA
  [relaxation]   .. residual stress: 289.641 meV/AA**3
```

```
[vibes]        .. Space group:     Fd-3m (227)
[relaxation]   clean atoms before logging
[relaxation]   .. log
[relaxation]   Step 1 finished.
[relaxation]   .. residual force:  0.000 meV/AA
[relaxation]   .. residual stress: 3.463 meV/AA**3
[vibes]        .. Space group:     Fd-3m (227)
[relaxation]   clean atoms before logging
[relaxation]   .. log
[relaxation]   Step 2 finished.
[relaxation]   .. residual force:  0.000 meV/AA
[relaxation]   .. residual stress: 0.039 meV/AA**3
[vibes]        .. Space group:     Fd-3m (227)
[relaxation]   clean atoms before logging
[relaxation]   .. log
[relaxation]   Relaxation converged.
[relaxation]   done.
```

You will find the `FHI-aims` in- and output in `relaxation/calculation/`, the final converged structure in `relaxation/geometry.in.next_step`, and a summary of the relaxtion path in `relaxation/relaxation.log`.

For a detailed summary of the relaxation path, you may run

```
vibes info relaxation relaxation/trajectory.son
```

and obtain an output like:

```
Relaxation info for relaxation/trajectory.son:
fmax:              1.000e+00 meV/AA
# Step |   Free energy   |   F-F(1)   | max. force |  max. stress |  Volu
#      |       [eV]      |    [meV]   |  [meV/AA]  |  [meV/AA^3]  |  [AA

    1    -15748.26920828     -3.914683       0.0000         5.0815     3
    2    -15748.26921491     -3.921313       0.0000         0.1132     3
--> converged.
```

It shows how energy, forces, and stresses have decreased during the relaxation, until convergence was reached.

OPTIONAL: You can also look at the relaxed structure in `relaxation/geometry.in.next_step` in an editor. Do you notice something? Try removing the old relaxation via

```
mv relaxation relaxation.old
```

and restarting the relaxation with the setting

`fix_symmetry: True`

What changed?