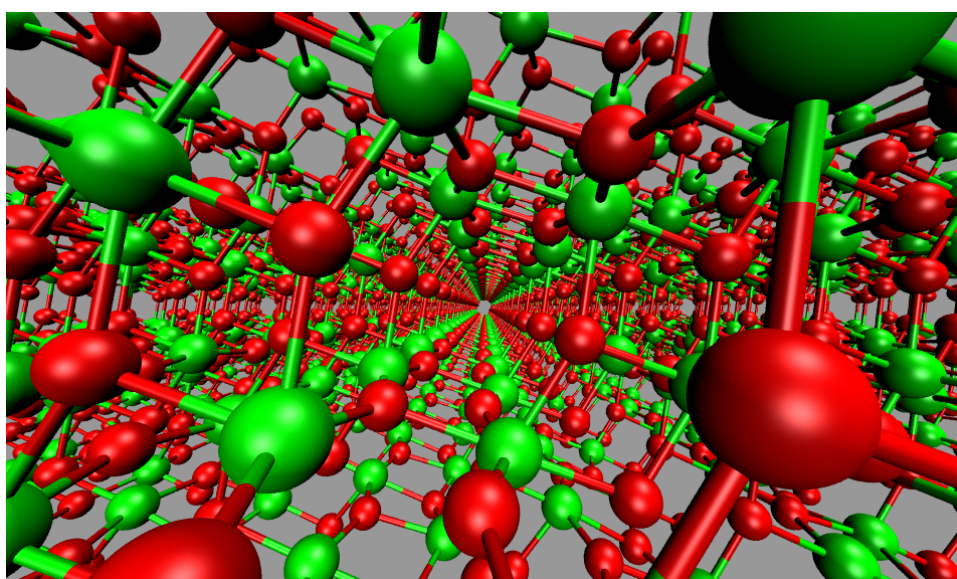

Special Course in Theoretical Physics PHYS-E0546: DFT for Practitioners

Aalto University, September 2022



Tutorial 3: Periodic systems Solid bulk and surfaces Manuscript for Exercise Problems

Prepared by Ondřej Krejčí, Orlando Silveira and Adolfo Fumega
Developed by Patrick Rinke, Milica Todorović,
Department of Applied Physics
Aalto University, 2022

Introduction

The aim of this tutorial is to familiarize you with the basic concepts of periodic density functional theory (DFT) calculations and the related settings necessary to run FHI-aims. Part I teaches you about the basic concepts of periodic systems. Part II introduces the basic concepts of surface calculations.

Part I: [Basic properties of solids and convergence tests](#)

 Problem I: [Basic calculations of bulk structures: cohesive energy curve](#)

 Problem II: [Energy convergence tests](#)

 Problem III: [The cohesive energy curves for bulk Si](#)

 Problem IV: [The equation of state for bulk Si](#)

 Problem V: [Unit cell relaxation](#)

 Problem VI: [Electronic band structure and density of states](#)

Part II: [Basic surface calculations](#)

 Problem VII: [Basic surface slab calculations](#)

 Problem VIII: [Relaxing surface structures](#)

Appendix I: [Structural information](#)

Appendix II: [High symmetry \$k\$ -points](#)

Appendix III: [Visualisation with VESTA](#)

For every exercise, we also provide solutions and sample input files. As always, they can be found in the `$TUTORIALS/` directory. However, we strongly recommend to use the provided input files only in case of time shortage. You will maximize your learning by trying to generate the input files on your own. In case you get stuck with a particular problem, do not hesitate to ask one of the tutors. An executable of FHI-aims, as well as species files and utility scripts, are provided in the `$COURSE/CODE/` directory.

Take your time to read the tasks carefully before proceeding to prepare the calculations. Each subtask starts with a short summary (gray box) and gives details and hints afterwards. Also, feel free to consult the supplementary information presented in the Appendices.

It is good practice to use a separate directory for every single run of FHI-aims in order to preserve the exact input files along with the output files.

Additional tools and programs

Visualization tools:

To visualize structures, vibrational modes, charge density plots, etc., several programs including molten, vmd, and jmol are installed on your workstations.

Utility scripts:

In this tutorial, additional scripts can be found in `$TUTORIALS/Tutorial_3/utilities`.

To visualize periodic structures, we recommend VESTA. A brief tutorial is provided in Appendix III. To open a `geometry.in` file in your current directory, type :

```
vesta `pwd`/geometry.in &
```

(Please note that the “```” character is a backtick). To get periodic images, choose “Objects” → “Boundary...” in the top panel. Then set the range of fractional coordinates that you want to see and press “Apply”. A range bigger than 1 results in a periodic image that is bigger than the specified unit cell. Note: if you see only the word "VESTA" on your screen, the file load didn't work and you should load the coordinates manually.

To visualize a structure given in `geometry.in` with Jmol (Fig. 2), please type:

```
jmol geometry.in &
```

Make sure the lattice vectors in your `geometry.in` are specified before your atomic positions to visualize the geometry properly with Jmol. To get periodic images, click with the right mouse button inside the Jmol drawing area and choose “Symmetry” → “Reload {444 666 1}”.

Part I: Basic properties of solids and convergence tests

In the first part of this tutorial, we will work on different structural phases of bulk silicon. The correct description of their pressure dependence by Yin and Cohen [1] is one of the early success stories of computational materials science. In this exercise, we show how to calculate basic properties of solids like lattice constants, cohesive energies, band structures, and density of states. Please use the basic settings given in Fig. 1 as default for part I of this tutorial.

```
# Physical settings
xc          pw-lda
spin        none
relativistic atomic_zora scalar

# SCF settings
sc_accuracy_rho    1E-4
sc_accuracy_eev    1E-2
sc_accuracy_etot   1E-5
sc_iter_limit      40

# k-grid settings (to be adjusted)
k_grid nkx nky nkz
```

Figure 1: Default physical and computational settings for `control.in` for part I

The Perdew-Wang LDA (`xc pw-lda`) exchange-correlation functional will be used for all calculations. Silicon is known to be non-magnetic, so no explicit spin treatment is needed. The “`relativistic atomic_zora scalar`” setting is not strictly necessary for silicon. The nuclear charge of silicon ($Z = 14$) is still small enough to allow for a non-relativistic treatment. But as the correction is computationally inexpensive, it does not hurt to use it (just be sure to never compare total energies from different relativistic settings). The `k_grid` setting will be discussed in the actual exercises. Please use the default “light” species settings for Si in: `$SPECIES/light/14_Si_default`.

Problem I: Basic calculations of bulk structures

Here, we learn how to set basic parameters in `control.in` for periodic calculations. Next, we construct periodic geometries in the FHI-aims geometry input format (`geometry.in`) and visualize them. Finally, we compare total energies of different Si bulk geometries.

Educational Objectives

- Learn how to set periodic boundary conditions (PBC) in calculations.
- Visualise and observe periodic unit cells.
- Learn to define the k-space sampling in a calculation.
- Compare energies of different lattice structures as a function of lattice constant.

The `control.in` for periodic calculations looks much the same as for the cluster calculations (molecules). There is one important difference: a *k*-grid for the Brillouin zone integrations must

be specified. The number of grid points determines the sampling density of the first Brillouin zone in 3D reciprocal space, needed to describe the variation of electronic bands $E(k)$ with wave vector k in periodic systems. The 3D Brillouin zone is defined by reciprocal lattice vectors computed from the real-space lattice vectors in the `geometry.in`. The order in which k-space grid settings are written is consistent with the ordering of the reciprocal lattice vectors, which follows from the order of real-space lattice vectors.

To set up a periodic structure in FHI-aims, all three lattice vectors as well as the atomic positions in the unit cell must be specified. The lattice vectors are denoted by the keyword `lattice_vector`. There are two ways to specify the atomic positions. You can specify absolute Cartesian positions with the keyword `atom`. Alternatively, you can specify the atomic positions in the basis of the lattice vectors, the **fractional** (or commonly, **direct**) coordinates, with the keyword `atom_frac`. The fractional coordinates s_i are dimensionless and the coefficients for the linear combination of the lattice vectors \mathbf{a}_i . Written out as a formula, this linear combination reads as follows

$$\mathbf{R} = s_1 \cdot \mathbf{a}_1 + s_2 \cdot \mathbf{a}_2 + s_3 \cdot \mathbf{a}_3, \quad (1)$$

where \mathbf{R} is the Cartesian position of the specified atom. For example, fcc Si with a lattice constant $a = 4 \text{ \AA}$ is defined by

```
lattice_vector    0.0    2.0    2.0
lattice_vector    2.0    0.0    2.0
lattice_vector    2.0    2.0    0.0
atom_frac         0.0    0.0    0.0    Si
```

The simplest way to check the `geometry.in` file is to visualize the corresponding geometry. This should **always** be done before any calculation, to verify that the structure is plausible (and that no atoms are extra or missing).

Tasks:

1. In the `control.in` file, set a $3 \times 3 \times 3$ k-space grid effectively making `nkx=nky=nxz=3`. You will see that for a small unit cell this k -point density is *never* enough, but we can try to find out what happens.
2. Construct `geometry.in` files for the Si *fcc*, *bcc*, and diamond structures. Use the approximate lattice constants a of 3.8 \AA for *fcc*, 3.1 \AA for *bcc*, and 5.4 \AA for the diamond structure. A full set of lattice vectors and atomic positions of primitive unit cells (the unit cell with the smallest volume, containing only one point of the crystal lattice) for *fcc*, *bcc*, and diamond can be found in Appendix I. Be sure to translate the given atomic positions (which are in Cartesian coordinates) to the right value for either the `atom` or the `atom_frac` keyword.

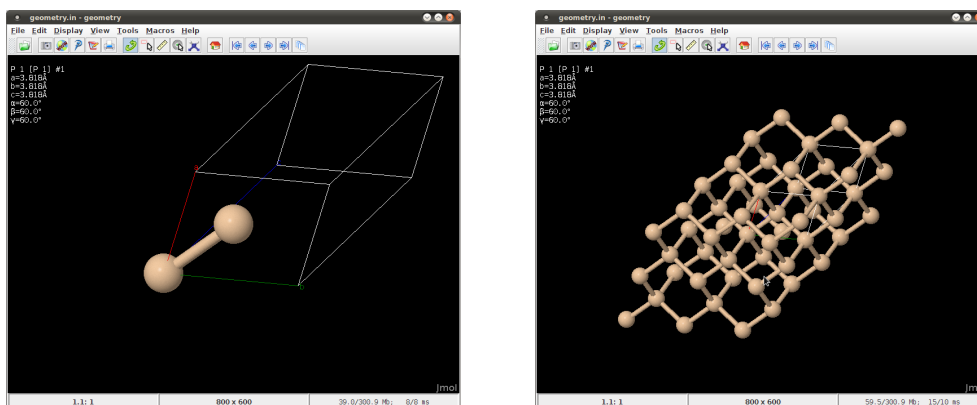


Figure 2: Single unit cell (left) and periodic images (right) of diamond Si in Jmol.

3. Visualize `geometry.in` files using VESTA (or Jmol). Can you spot how the primitive cell motif is repeated throughout space? And what about the number of nearest neighbors for each atom?
4. Make dedicated folders for each Si crystal structure and copy the input files there. Also copy over the provided `run_diamond.py` script to calculate total energies of the diamond Si phase as a function of lattice constant a .
5. Start with the diamond phase calculation and edit the script `run_diamond.py` to insert the range of unit cells to be computed: choose 7 different values of a in steps 0.1 \AA around the lattice constant given above. Use the command `python3 run_diamond.py` to start the calculation. It can take a while to finish.
6. Copy the `run_diamond.py` script to the remaining directories and carefully modify the geometry sections to adjust the lattice vectors for other primitive cells. Insert new unit cell ranges as appropriate and complete the calculations. Alternatively, you can use the scripts `run_fcc.py` and `run_bcc.py`, which have already been modified.
7. Use the `postprocess_diamond.py` script to retrieve the total energies which will be written to the file `energies.dat`, along with the lattice constants. For each phase of Si, modify the script to input the range of lattice constants you had tried and run the script. Alternatively, you can use the scripts `postprocess_fcc.py` and `postprocess_bcc.py`, which have already been modified.
8. Compute the total energy per atom, not per unit cell, so we can safely compare the energies between different structures. This is important for the diamond structure (in the case of *fcc* and *bcc*, each unit cell contains one atom).
9. Plot the total energy per atom for all structures as a function of the lattice constant. At which lattice parameters do your curves reach the minimum? How does this compare to the experimental lattice constants?
10. What is the most stable bulk phase of Si according to your results?

You should see that, with the current computational settings, the diamond Si phase is unfavorable compared to the other two phases by about 0.1 eV . However, the experimentally most stable phase is the diamond structure. We will show in the next two problems that the coarse $3 \times 3 \times 3$ k -grid is the reason for this disagreement.

Problem II: Energy convergence tests

The results of unconverged calculations can range from inaccurate to unphysical, as seen in the previous exercise. Now we explicitly check total energy convergence with respect to the k -grid and to the basis set. Each phase needs to be checked separately, but in the interests of time, choose one Si phase to proceed with.

Educational Objectives

- Develop a feeling for the k -point sensitivity.
- Deepen the understanding of convergence.

Subproblem 1: Convergence with k -grid

1. For your chosen phase, employ the input files from the previous exercise. Prepare `control.in` files with k -grids of $8\times 8\times 8$, $12\times 12\times 12$, and $16\times 16\times 16$.
2. For each k -grid setting, compute the total energy curve as before (repeat steps 5-8 from the previous exercise).
3. Plot the total energies versus the lattice constant for the different k -point settings. Add the $3\times 3\times 3$ results too.
4. Which k -grid should be used to achieve convergence within 10 meV?
5. How does the computation time increase with the number of k -points? Hint, you can find the total run time in the `aims.out` output file.
6. Which k -grid offers the best compromise between accuracy and runtime?

Subproblem 2: Convergence with basis set size

To change the basis size settings, take a look at the species-dependent settings in `control.in`. There, you will find a line starting with “# **First tier** - ...”. Each line after this defines a basis function which is added to the `minimal` basis. Right now, there is one additional function for each valence function (s and p) as well as a d function to allow the atoms to polarize. This is what we call `minimal+spd` in the context of this tutorial. In quantum chemistry and in particular the Gaussian community, this type of basis set is often called “double zeta (ζ) plus polarization” (DZP).

1. Use your `control.in` file from the previous problem with the $12\times 12\times 12$ k -point grid. Calculate the total energies for **your** phase of Si as a function of the lattice constant for the `minimal` and the `tier1` basis sets.
2. Plot the total energies curves obtained for the two different basis sets and write down the values of the lattice constants that minimize the energy. Also, consider the values of the lowest energy reached in each curve, and compare the total time of the calculations.
3. What do you observe? How do the curves change? How does this compare to the k -point convergence?

Optional subproblem 3: Effect of Gaussian broadening of the Kohn-Sham occupation numbers

FHI-aims always outputs three different energies. While these energies are all the same for systems with a band gap, they differ for metallic systems with finite Gaussian broadening. The “**Total energy uncorrected**” gives the value of the Kohn-Sham energy functional for the final self-consistent electronic structure. However, due to the Gaussian broadening the self-consistency procedure does not minimize this total energy but a “free energy” specified to the right of “**Electronic free energy**”. From these two numbers, FHI-aims extrapolates back to the total energy without broadening and writes the resulting number to the right of “**Total energy corrected**”. For true metals, it is generally best to make use of the correction. For finite systems and in particular for isolated atoms, however, the back-extrapolation is unphysical and should not be used.

Tasks:

1. Use the `control.in` file and scripts for the *fcc* Si system, with the $12\times 12\times 12$ k -point grid

setting and the `minimal+spd` basis. You can explicitly set the Gaussian broadening to $\sigma = 0.1$ eV by specifying

```
occupation_type gaussian 0.1
```

in `control.in`.

2. Prepare a plot with the corrected, uncorrected total energies, and the electronic “free energies” for a broadening of $\sigma = 0.1$ eV and the default value of $\sigma = 0.01$ eV from the previous calculations.
3. How does the broadening affect the results? Which broadening is safe to use?

In the calculations that follow, we will use the default value of $\sigma = 0.01$ eV. We will stick to the corrected total energy for the periodic systems in this part of the tutorial as it is the most accurate value for metals and makes no difference for semiconductors.

Problem III: The cohesive energy curves for bulk Si

Having established well-converged computational settings, we now revisit the phase stability of bulk silicon (in practice, one should always check convergence *first*). We will learn how to compute the basic cohesive properties and study the pressure dependence of phase stability. To proceed, we need to calculate the cohesive energy curve from the $E(\mathbf{a})$ curve computed in previous exercises.

Educational Objectives

- Learn to compute the cohesive properties of a solid.

The cohesive energy (E_{coh}) of a crystal is the energy per atom needed to separate it into its constituent neutral atoms. It is computed per simple cubic unit cell defined by the lattice constant a to compare cohesion of different phases on an equal footing. E_{coh} is defined as

$$E_{\text{coh}} = -\frac{E_{\text{bulk}} - NE_{\text{atom}}}{N} = -\left[\frac{E_{\text{bulk}}}{N} - E_{\text{atom}}\right], \quad (2)$$

where E_{bulk} is the bulk total energy **per unit cell** and N the number of atoms in the unit cell. E_{atom} is the energy of the isolated atom. The number of atoms in the simple cubic unit cell defined by the lattice constant a is different for each phase of Si, as illustrated in Fig. 6, Appendix I.

Tasks:

1. Use the converged settings (k -grid $12 \times 12 \times 12$, `minimal+spd` basis, and $\sigma = 0.01$ eV occupation broadening) to compute the total energy of *fcc*, *bcc*, and diamond Si as a function of lattice constant a . Which phase of Si is the most stable?
2. Compute the energy of a single Si atom following what you learned in Tutorial 1 (make sure you use different settings in the input file, and note the correct energy in the output file).
3. Determine the number of atoms N_{sc} per cubic unit cell defined by the lattice constant a for *fcc*, *bcc*, and diamond Si crystal phases. Do not confuse N_{sc} with the number of atoms N in the unit cell of your calculations.
4. For each phase of Si, use the script `convert-coh.py` to compute the cohesive energies `cohesive.dat` from `energies.dat`:

`python3 convert-coh.py --Natoms .. --Eatom .. -i energies.dat -o cohesive.dat`
 where the “..” need to be replaced by the corresponding values of N_{sc} and E_{atom} , respectively. Note that the result the script provides is instead the energy per atom ($E = -E_{\text{coh}}$), since this will be the quantity required in the next problem.

5. Plot the cohesive energies of all three phases on one graph with the atomic volume (not lattice constant a) on the x axis.
6. What do you observe?

Problem IV: The equation of state for bulk Si

Educational Objectives

- Learn to analyze the equation of state for a solid.

The equilibrium lattice constant a_0 is an important quantity which we can calculate from our data. In principle, this can be done with a quadratic ansatz for $E(a)$ or $E(V)$. Here, we will discuss and use a thermodynamically motivated and more accurate fitting function, the Birch-Murnaghan equation of state [2, 3]. The energy per atom ($E = -E_{\text{coh}}$) is expressed as a function of the atomic volume ($V = V_{\text{atom}}$)

$$E(V) = E_0 + \frac{B_0 V}{B'_0} \left[\frac{(V_0/V)^{B'_0}}{B'_0 - 1} + 1 \right] - \frac{B_0 V_0}{B'_0 - 1}. \quad (3)$$

The fitting parameters V_0 and E_0 are the equilibrium atomic volume and atomic energy, respectively, B_0 the bulk modulus and B'_0 its derivative with respect to pressure. Equation (3) can be derived by assuming a constant pressure derivative B'_0 .

The fitting script `murn.py` is included in the utilities folder for this tutorial. If you have prepared the files `cohesive.dat` from the previous problem, you can simply use this script with

```
python3 murn.py -o fit.dat -p cohesive.dat
```

The program then outputs the parameters V_0 , E_0 , B_0 , and B'_0 for the given data set as output. As a quick plausibility check of the fit, you can use the option `-p` to see a plot. The script performs no unit conversions, so the bulk modulus B_0 is given in units of $\text{eV}/\text{\AA}^3$ because the cohesive energies and atomic volumes were provided in eV and \AA^3 , respectively ($1 \text{ eV}/\text{\AA}^3$ is 160.21765 GPa). The optimal lattice constant can be calculated from the equilibrium atomic volume V_0 by $a_0 = \sqrt[3]{N_{\text{sc}} V_0}$ with N_{sc} the number of atoms in the cubic unit cell.

Tasks:

1. For each phase of Si, apply the script `murn.py` to perform the Birch-Murnaghan fit of the cohesive energy curves `cohesive.dat`.
2. Take note of the output parameters from the script, and use them to compute the equilibrium lattice constant a_0 in \AA , the bulk modulus B_0 in GPa , and the cohesive energy E_{coh} in eV .
3. Compare the calculated quantities for the diamond phase with the experimental values of $a = 5.430 \text{ \AA}$, $B_0 = 98.8 \text{ GPa}$, and $E_{\text{coh}} = 4.63 \text{ eV}$ [4]. Exact agreement with experimental data is not expected, since xc functionals are approximated. Based on your observations, what can you tell about the tendencies of the `pw-lda` functional - does it overbind or underbind?
4. Plot the resulting fitted curves saved in `fit.dat` into one figure. You should get something similar to Fig. 1 in the paper by Yin and Cohen [1].

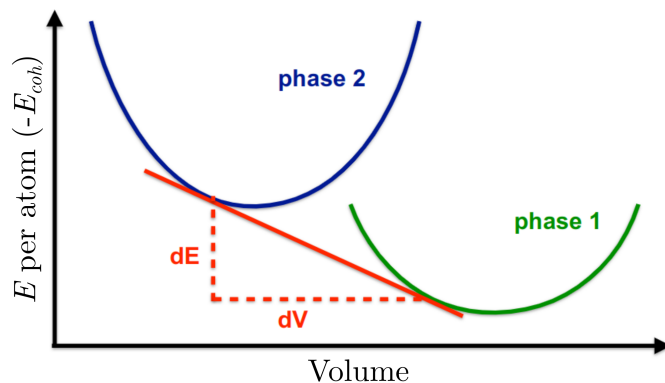


Figure 3: Maxwell construction

5. *Bonus:* Given this data, can you estimate at what pressure a phase transition would occur? Recall the Maxwell construction (see below).

By exposing the crystal to the right pressure, one can enforce many different atomic volumes smaller than the equilibrium ones. In principle, the most stable phase for a given atomic volume V can simply be deduced from the curve with the lowest $E(V)$. The corresponding pressure can be calculated from the slope of the curve by the simple thermodynamic relation $p = -\partial E / \partial V$.

However, in the regime of about $13 \text{ \AA}^3 < V_{\text{atom}} < 18 \text{ \AA}^3$ coexistence of a diamond phase at $\sim 18 \text{ \AA}^3$ and a *bcc* phase at $\sim 13 \text{ \AA}^3$ is favorable (see Fig. 3). The fraction of atoms in the two phases then determines the average atomic volume. The resulting average atomic energy is a straight line between the corresponding points which is below both the lines of diamond and *bcc* Si. This is called the Maxwell construction. From the slope of this line (a common tangent), we can deduce a transition pressure of about 14 GPa at which diamond and *bcc* Si could coexist according to our calculations. This is more than 10^5 times the ambient pressure of about 100 kPa. Note that there are additional phases for silicon which have not been calculated here. In fact, the Si β -tin phase is more stable than the *bcc* phase at high pressures. For a more thorough discussion, refer to [1].

Problem V: Unit cell relaxation

In the previous tutorial you learned how to optimize the positions of atoms in molecules. To obtain optimized periodic structures, not only the atomic positions must be optimized but also the lattice vectors. Here, we perform a structure relaxation from *bcc* to diamond Si.

When performing a *local* structure optimization, we do not go uphill in the energy landscape or cross energy barriers. The path from the starting structure to the desired local minimum has no barriers. This means that we have no assurance that we have found the global minimum, which may lie over an energy barrier from our starting geometry (to cross energy barriers, one can do, for example, variable cell-shape molecular dynamics, but this is out of the scope of this tutorial).

Since we want to end up in a diamond structure, we have to start with at least 2 atoms: it is crucial that the number of atoms in the initial and final unit cells is the same. Our starting unit cell is a cubic *bcc* structure consisting of 2 atoms. Of course, we cannot start with the *ideal* cubic *bcc* unit cell because we will be stuck in the local minimum of this high symmetry structure. Even if we distort the starting geometry significantly (atomic positions & lattice vectors), we can easily end up in an unwanted local minimum. Here we provide a suitable starting geometry, feel free to try other ones when you are finished with this one.

Tasks:

1. Prepare input files from the previous exercise (k-grid $12 \times 12 \times 12$, minimum+spd basis set) for unit cell relaxation by adding the following keywords to `control.in`:

```
relax_geometry trm 1E-2
relax_unit_cell full
```

The latter keyword enables full optimization of the lattice vectors (their lengths and the angles between them). The convergence criterion will be the same as the one specified for the atoms.

2. In the convergence section of the input file, add the line:

```
sc_accuracy_forces 1E-3
```

The computed forces will change during the SCF cycle as the electronic structure evolves, so the convergence of forces within the SCF cycle also needs to be considered for accurate relaxation calculations. Tolerance of 10^{-3} eV/Å is sufficient given the desired relaxation accuracy above.

3. Prepare the initial geometry for structure relaxation as follows:

```
lattice_vector  3.1  0.4  0.4
lattice_vector  0.45 3.1  0.4
lattice_vector  0.45 0.45 3.1
atom_frac       0.0  0.0  0.0  Si
atom_frac       0.3  0.3  0.3  Si
```

4. Run the calculation (`mpirun -np aims.x | tee output`), it might take a while, and use the `get_relaxation_info.pl` script on the output file to check the calculation has finished correctly. Take a closer look at the output file to see that not only the atomic forces are calculated but also a quantity called stress tensor. The stress tensor is a measure of the forces acting on the unit cell itself.
5. To visualise the geometry optimisation, apply the `create_geometry_zip.pl` to your output file. Unzip the resulting file `geometries.zip` to see the individual geometries or use Jmol by typing the command “`jmol -s geometries.spt`”.
6. Analyze the final structure (angles between lattice vectors, their length, and atomic positions) and compare it to the equilibrium values of diamond Si obtained from the Birch-Murnaghan fit in the previous problem.

Problem VI: Electronic band structure and density of states

The electronic band structure describes the dispersion relation of electrons inside a solid. The band structure gives information about ranges of energy that electrons are allowed to have or not. Permitted ranges are called bands, forbidden ones band gap. Many properties of a solid can be deduced from its band structure, e.g., if it is a metal, semiconductor, or insulator.

The density of states (DOS) is one of the basic concepts in solid state physics. Particularly, the DOS around the Fermi level is of interest as it will give you information about the properties of your system, for example, whether it is conducting, semiconducting, or insulating. Many material properties depend on the DOS, for instance the conductivity. The number of states n within a given energy interval $(\epsilon_0 - \Delta\epsilon) < \epsilon < (\epsilon_0 + \Delta\epsilon)$ per unit volume V_{cell} is given by

$$n = \int_{\epsilon_0 - \Delta\epsilon}^{\epsilon_0 + \Delta\epsilon} d\epsilon g(\epsilon) \quad (4)$$

where $g(\epsilon)$ is the density of states (DOS). In a free atom or an isolated molecule, the DOS consists of a series of discrete energy levels (δ peaks) and can be written as

$$g(\epsilon) = \sum_i \delta(\epsilon_i - \epsilon). \quad (5)$$

In a periodic system, the single particle energies become k dependent and the DOS continuous. The number of states per energy is averaged over k

$$g(\epsilon) = \frac{1}{V_{\text{BZ}}} \sum_i \int_{\text{BZ}} d^3k \delta(\epsilon_{i,k} - \epsilon). \quad (6)$$

In order to calculate the density of states numerically, we have to replace the integral over the Brillouin zone (BZ) in Eq. (6) by a sum over k -points. In the case of infinite k -points, this replacement is exact. To compensate the deficiency of a finite grid, we broaden the $\delta(\epsilon_{k,i} - \epsilon)$ distribution by a Gaussian function with an broadening factor σ ,

$$g(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{n_k} \sum_i \sum_k \exp \left[-\frac{1}{2} \left(\frac{\epsilon - \epsilon_{k,i}}{\sigma} \right)^2 \right]. \quad (7)$$

Tasks:

1. Use your `control.in` and `geometry.in` for the equilibrium lattice parameter for diamond Si from Problem IV (Murnaghan fit) to prepare input files for electronic structure calculation.
2. In the `control.in` file, add the keywords:

```
output dos -18. 0. 200 0.1
```

The first two values define the energy window of interest, the first value is the lower energy bound and the second value is the upper energy bound. The third value is an integer specifying at how many energy data points the DOS is evaluated, and the last value gives the Gaussian broadening σ . All energies (bounds and broadening) are given in eV.

3. For a well-resolved DOS function, a dense k -grid to determine the sum over k -points in Eq. (7) is necessary. After self-convergence is reached, the DOS can be computed using an auxiliary k -grid that is made denser by factors n_1 , n_2 , n_3 , respectively. To switch on this function, specify factors n_1 to n_3 in the `control.in` with the keyword:

```
dos_kgrid_factors 5 5 5
```

The main calculation will be executed using the $k_1 \times k_2 \times k_3$, but the density of states is calculated on a denser grid after the SCF cycle. The dimensions of the new k -point grid are $k_1 \times n_1$, $k_2 \times n_2$, $k_3 \times n_3$.

4. To calculate the band structure, specify the path along high-symmetry points in reciprocal space: $L \rightarrow \Gamma \rightarrow X \rightarrow W \rightarrow K$ in `control.in` file:

```
# diamond band structure:
output band 0.5 0.5 0.5 0.0 0.0 0.0 21 L Gamma
output band 0.0 0.0 0.0 0.0 0.5 0.5 21 Gamma X
output band ...
```

Please refer to Appendix II for the location of the other high symmetry points in the Brillouin zone. In each line, the first three numbers are the coordinates of the starting point in units of the reciprocal lattice vectors. The next three numbers specify the end point. The band structure is then calculated along the path connecting these two points.

5. Run the calculation. Densities of states will be output into separate files: `KS_DOS_total_raw.dat`, where the zero of energy has not been altered, and `KS_DOS_total.dat`, where the zero of energy has been shifted to the computed Fermi level. Plot the last file to visualize the density of states of silicon in the diamond phase. How does the LDA bandgap compare with the experimental band gap of silicon of 1.17 eV [4]?
6. To visualise the computed band structure, use the script `aimsplo.py` without any arguments in the calculation directory. Note that the DOS plotted along the band structure reflects the density of the electronic bands at any energy.

Part II: Basic surface calculations

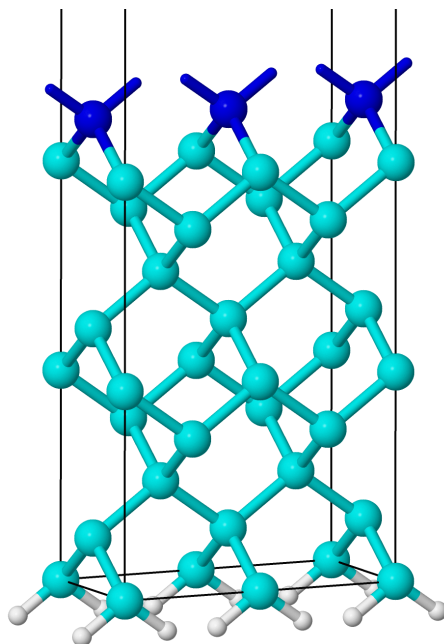


Figure 4: The hydrogen-saturated ideal 2×1 Si(001) surface. The cyan (light) atoms correspond to the bulk Si atoms, the blue (dark) atoms are the top Si atoms. The white atoms on the bottom layer are hydrogen atoms.

In this exercise we focus on the clean Si(001) surface, technologically the most relevant Si surface. This surface terminates by unsaturated (dangling) silicon bonds, so the surface atoms re-arrange into a more energetically favourable *surface reconstruction*. The nature of the surface reconstruction was unclear for many years. After long discussions and different models used to understand the differences between various experimental (LEED, STM) and theoretical approaches, it was shown by direct evidence in STM by Wolkow [5] that the main surface feature at low temperatures is the asymmetric Si dimer in a 2×1 reconstructed surface unit cell.

We will continue with the same level of accuracy, so you can take your `control.in` and only adjust the k -grid settings as shown below. For "saturating" the bottom layer of the slab, we additionally need the *light* species defaults for hydrogen, so add this to the `control.in` file. As shown in Fig. 4, each silicon atom has to be saturated by two hydrogen atoms.

```
# Physical settings
xc                pw-lda
spin              none
relativistic      atomic_zora scalar

# SCF settings
sc_accuracy_rho   1E-4
sc_accuracy_eev   1E-2
sc_accuracy_etot   1E-5
sc_iter_limit     40

# k-grid settings
k_grid            12 24 1
```

Figure 5: Physical and computational settings for `control.in` for part III.

Problem VII: Basic surface slab calculations

Educational Objectives

- Learn to build and visualize a surface structure.
- Compute and analyse surface band structures.

Building surface slab models is not difficult, but one must proceed with care. The geometry of four layers of Si(001)-(1×1) in diamond structure is provided in the Appendix I. By convention the surface is rotated with respect to the bulk structure by 45° around the z axis, so that the z axis is perpendicular to the surface plane. Adding the next four layers in thickness would involve a repetition of the first four layers, with coordinates shifted by a in z direction. To expand this 1×1 unit cell into the (2×1) reconstructed surface model, one must add a repetition of all the atomic coordinates, shifted by $a/\sqrt{2}$ in x direction and double the corresponding lattice vector. This is handily done in a spreadsheet or with scripts: here we provide script `write-geom.py` for your convenience.

To determine a sufficient amount of vacuum between slabs and reduce interactions between periodic images, you would need to run several calculations with different vacuum thickness (typically, at least 10 Å of vacuum is needed). In FHI-aims vacuum is cheap, you can easily use a vacuum of $L_{\text{vac}} = 30 \text{ Å}$ or more without noticeable performance impact.

For periodic calculations with a slab, the k-point grid needs to be modified to request only one k-point in the z direction (there should be no interaction between different periodic images of the slab). Also, in a 2×1 unit cell, the lattice vector a_x is twice as large as a_y , so the reciprocal vector r_x is twice smaller than r_y and needs only half the number of k-points.

Tasks

1. Use the python scripts `write-geom_4L.py` and `write-geom_8L.py` (from utilities directory) to build the 2x1 ideal, hydrogen-saturated, 4 and 8 layer surface slabs respectively. Edit the scripts to insert the optimized lattice constant obtained from the Murnaghan fit and look at the difference between the scripts. Leave the value $L_{\text{vac}} = 30 \text{ Å}$ for vacuum thickness. The scripts need no input file: `python3 write-geom_4L.py > geometry.in`
2. Visualise the 2x1 slabs with Jmol or VESTA, and note its orientation with respect to the coordinate axes. How many atoms are there in each layer of the unit cell?
3. In the `control.in` files, adjust the k-point settings for the surface using the precomputed converged setting of $12 \times 24 \times 1$.
4. Switch on the DOS calculation by adding the following lines:

```
output dos -18. 0. 200 0.1
dos_kgrid_factors 5 5 1
```
5. For band structure calculations in the 2D reciprocal space of the surface (see Appendix II) following the path $\bar{\Gamma} \rightarrow \bar{J} \rightarrow \bar{K}$, introduce the following settings:

```
# Si 2x1 surface band structure:
output band 0.0 0.0 0.0 0.5 0.0 0.0 21 Gamma J
output band 0.5 0.0 0.0 0.5 0.5 0.0 21 J K
```
6. Perform single-point calculations for the four layer and eight layer surface slabs. Use the `aimsplot.py` script to visualise and compare the band structures between the two slabs.

When plotting the band structures as before, note the differences between the four- and the eight-layer slabs. While the bands within the fundamental band gap (which arise from the surface atoms) hardly change with the number of layers, the bands in the valence and the conduction bands get about twice as dense. These bands are bulk-like; the more layers you add, the more of these bands appear.

Problem VIII: Relaxing surface structures

Educational Objectives

- Familiarize yourself with atomic relaxation at surfaces.

Relaxing surface slabs is conceptually different than relaxing bulk structures. To mimic realistic surfaces, lower layers of the slab are assigned bulk-like character and kept in fixed position, while the upper layers are relaxed. In FHI-aims, particular atoms can be constrained by inserting the following line right under the atom coordinates.

```
constrain_relaxation .true.
```

An example excerpt from `geometry.in` reads like this:

```
atom -1.2063524529754976  0.0000000000000000  -0.8530200000000001  H
  constrain_relaxation .true.
atom  1.2063524529754976  0.0000000000000000  -0.8530200000000001  H
  constrain_relaxation .true.
atom ...
```

Tasks

1. Start with the input files from a previous calculation, using the four layer Si slab structure. Fix the bottom two layers (one H and one Si layer) and leave the top three (all Si) layers to relax.
2. Adjust the `control.in` file for structural relaxation by inserting the following keywords:

```
sc_accuracy_forces 1E-3
relax_geometry trm 1E-2
```

The settings above will relax the atoms inside the unit cell without changing unit cell parameters. These are fixed to allow lower slab layers to mimic ideal bulk structure.

3. In structural relaxations, it is a useful trick to depart from perfectly symmetric coordinates in the initial geometry. Introduce a slight perturbation (about 0.5 Å) in one or more coordinates of the two Si atoms in the surface layer of `geometry.in`. This ensures that the initial geometry is different from the symmetry-induced saddle point of the ideal geometry. A possible choice for the top silicon atoms is:

```
atom 2.4148451634531707  0.0000000000000000  3.262  Si
atom 5.2445354903595121  0.0000000000000000  4.062  Si
```

Here, the two top Si atoms have been moved in the x-direction by ± 0.5 Å, and one Si atom has been pushed down by 0.8 Å.

4. Perform the calculation (it should take about 5min on 4CPUs). Check that it completed correctly with the `get_relaxation_info.pl` script. To visualise the geometry optimisation, apply the `create_geometry_zip.pl` to your output file. Unzip the resulting file

`geometries.zip` to see the individual geometries or use Jmol by typing the command `jmol -s geometries.spt`. Has the surface changed its morphology?

5. Apply scripts to compute the band structure of the relaxed surface structure. Compare the bands with the data from the previous, ideal surface calculation.
6. What do you observe?

Appendix I: Structural information

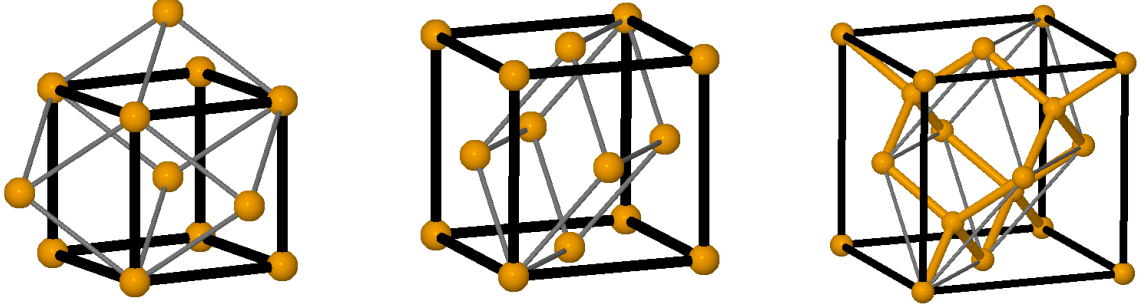


Figure 6: Cubic (black) and primitive (grey) unit cell for *bcc*, *fcc* and diamond Si (from left to right).

Crystal structures	atomic coordinates				lattice vectors		
<i>fcc</i>	0 0 0				0	$a/2$	$a/2$
					$a/2$	0	$a/2$
					$a/2$	$a/2$	0
diamond	0	0	0		0	$a/2$	$a/2$
	$a/4$	$a/4$	$a/4$		$a/2$	0	$a/2$
					$a/2$	$a/2$	0
<i>bcc</i>	0 0 0				$-a/2$	$a/2$	$a/2$
					$a/2$	$-a/2$	$a/2$
					$a/2$	$a/2$	$-a/2$
zincblende with atom species A and B	A	0	0	0	0	$a/2$	$a/2$
	B	$a/4$	$a/4$	$a/4$	$a/2$	0	$a/2$
					$a/2$	$a/2$	0

Table 1: Solids: Atomic coordinates and lattice vectors for different crystal structures.

diamond(001)						
atomic coordinates				lattice vectors		
0	0	0		$a/\sqrt{2}$	0	0
$a/2\sqrt{2}$	0	$a/4$		0	$a/\sqrt{2}$	0
$a/2\sqrt{2}$	$a/2\sqrt{2}$	$a/2$		0	0	L
0	$a/2\sqrt{2}$	$3a/4$				

Table 2: Surfaces: The atomic coordinates of an ideal diamond(001) surface. *Note:* a is the bulk lattice constant and L is the total slab thickness ($L = a + L_{\text{vac}}$ with the vacuum size L_{vac}).

Appendix II: High symmetry k -points

fcc	x_1	x_2	x_3
L	0.5	0.5	0.5
Γ	0	0	0
X	0	0.5	0.5
W	0.25	0.5	0.75
K	0.375	0.375	0.75

$fcc(001)$	x_1	x_2	x_3
$\bar{\Gamma}$	0.0	0.0	0.0
\bar{J}	0.5	0.0	0.0
\bar{K}	0.5	0.5	0.0

Table 3: High symmetry points for fcc /diamond bulk and (001)surface structures given in units of the three reciprocal lattice vectors ($\vec{k} = x_1\vec{b}_1 + x_2\vec{b}_2 + x_3\vec{b}_3$). In the case of fcc /diamond bulk, the reciprocal lattice vectors form a bcc structure and the corresponding lattice vectors can be found in Tab. 1.

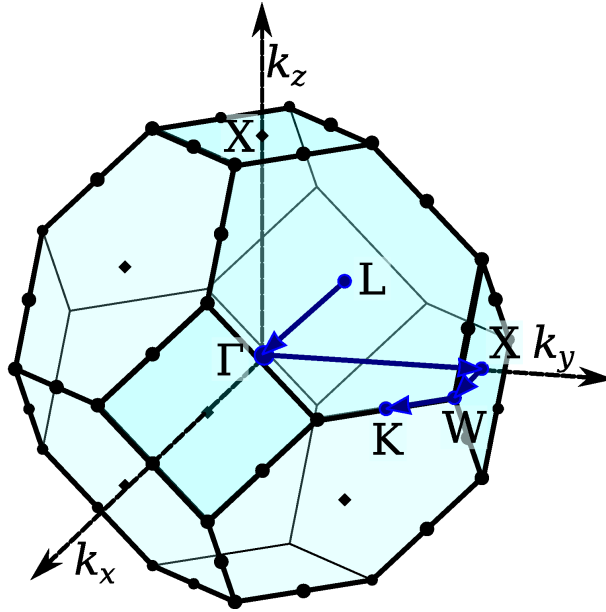


Figure 7: Brillouin zone and high symmetry points for fcc /diamond structure. The three coordinate axes (\vec{k}_x , \vec{k}_y , and \vec{k}_z) form a Cartesian coordinate system – not to be confused with the reciprocal lattice vectors of the fcc /diamond structure.

Appendix III: Visualisation with VESTA

Visualizing your results from density functional calculations is useful and important for a couple of reasons. First, you can visualize the structure **as it was read by FHI-aims** and check its accuracy. This is a fool-proof way to check that the geometry handled by the code is actually the one you want. Second, you can view ‘volumetric data’ superimposed on the structure of whatever system you are dealing with. Volumetric data can be any kind of data written to a real-space grid including charge densities, wave functions, potentials, etc. Visualization of volumetric data can help you extract meaningful physics from your calculation. This document is a quick-start guide to visualizing output from FHI-aims with VESTA.

VESTA is a public, easy-to-use application for visualization of atomic structures and volumetric data. You can find it here: <http://jp-minerals.org/vesta/en/>

The complete manual for VESTA is here:

http://www.geocities.jp/kmo_mma/crystal/download/VESTA_Manual.pdf

Visualizing crystal structures

It is difficult to understand what a bulk, three-dimensional crystal structure looks like just by reading atomic coordinates. Luckily, VESTA will read the `geometry.in` file needed for FHI-aims and visualize the crystal for you. Type `vesta` to launch.

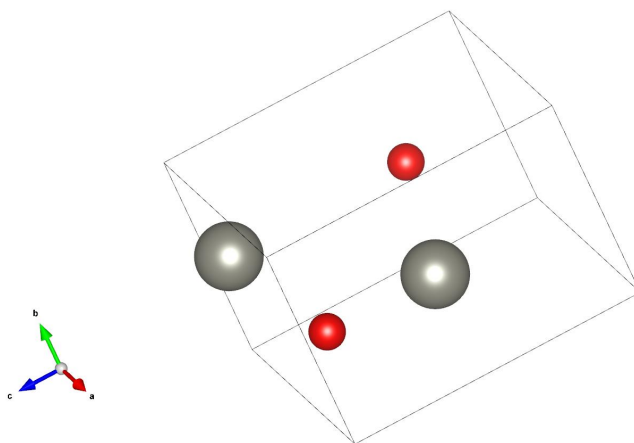


Figure 8

Now navigate to **File** and open your `geometry.in` file for FHI-aims. It should show the crystal structure you are interested in. An example `geometry.in` file for wurtzite ZnO and its appearance in VESTA are shown below and in Fig. 8.

```
lattice_vector 3.27534473 -0.07200997 -0.00749503
lattice_vector -1.57091511 2.87405204 -0.05082762
lattice_vector 0.01006313 0.10435241 5.29125719
atom 0.04290507 1.88121444 -0.06093020 Zn
atom 1.66090983 0.94898484 2.59987036 Zn
atom 0.04188438 1.91975639 1.94486746 O
atom 1.66718882 0.99019168 4.60509159 O
```

You can rotate the structure just by clicking and dragging the mouse. If you lose track of what direction you are looking at, click the ‘a,’ ‘b,’ or ‘c’ buttons on the top left. These will automatically orient the camera with the first, second, or third crystallographic direction. Notice the three

crystallographic directions are given by the axes in the bottom left corner. If you click on ‘Tools’ at the top left of Fig. 12, you can select the ‘Animation’ rotation mode. With this selected, the crystal will slowly rotate in the direction you choose.

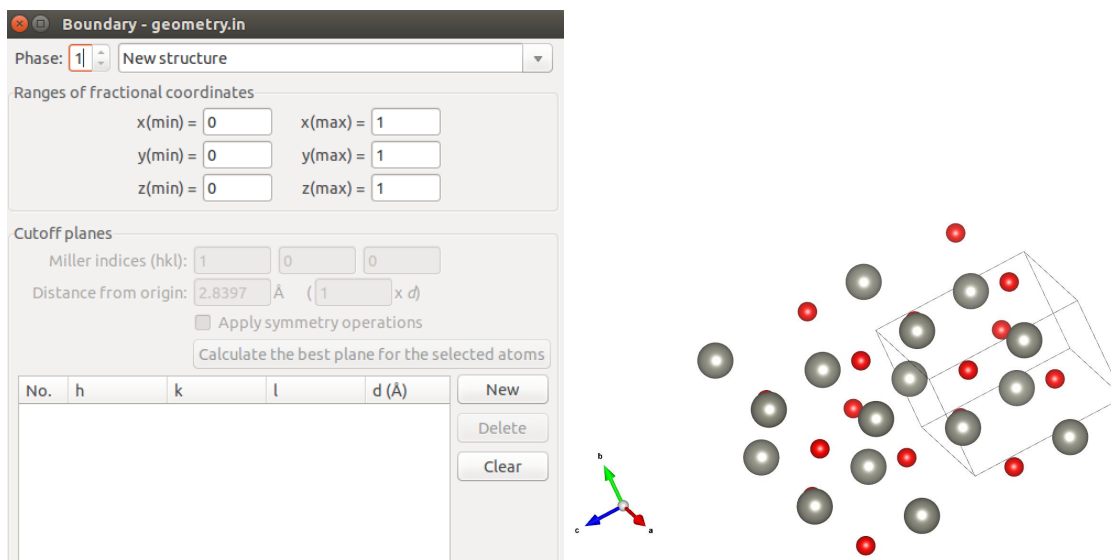


Figure 9

Fig. 8 looks ok, but it can be improved a lot. When looking at a periodic structure, it is helpful to increase the number of unit cells displayed so that you can see how different cells meet at their boundary. Click on **Objects** → **Boundary** to get the menu shown in Fig. 9. The boundaries of the displayed structure are set by ‘xmin’ and ‘xmax’ in reduced coordinates (and likewise for *y* and *z*). Setting the boundaries to 0 and 1 shows one cell. Try setting all of the maximum boundaries to 2.

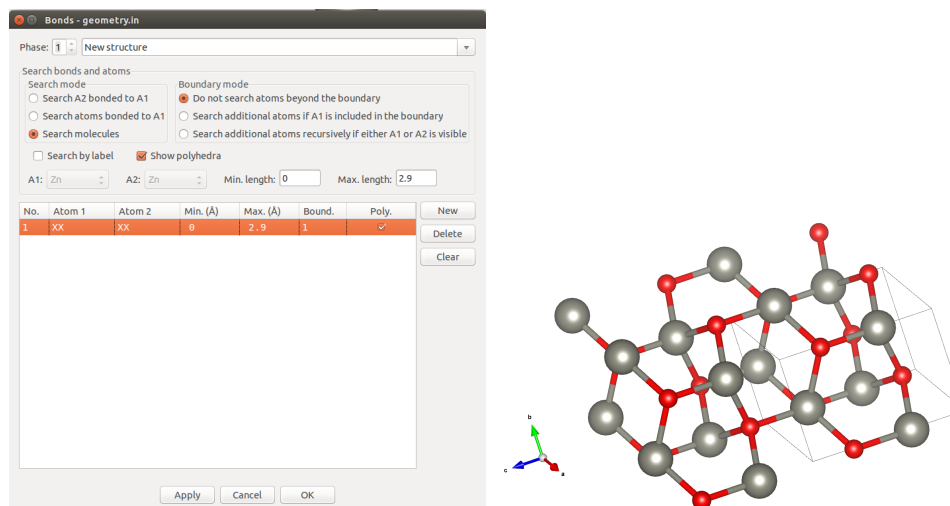


Figure 10

That looks better, but it is still difficult to see how the material is bonded. To fix that, click on **Edit** → **Bonds**. You should see the menu shown in Fig. 10. You have a few choices on how VESTA will search for bonds. The simplest, catch-all method is to click on ‘New’, choose ‘Search molecules,’ check ‘Do not search atoms beyond the boundary,’ and then set a maximum bond length. This choice will draw a bond for any two atoms that are less than the maximum bond length away from each other. You need to determine a reasonable maximum bond length for your structure. By searching atoms bonded only to ‘A1’ and/or ‘A2,’ you can search for bonds only

involving those one or two chemical species. Now that the bonds are drawn, it is much easier to visualize the crystal and how the atoms are arranged when you rotate the crystal. For reference, a single unit cell is still outlined in black by VESTA.

Visualizing cube files

Navigate to **File** and open your **.cube** file from FHI-aims. It should show the atomic structure and some colored surfaces, something like Fig. 11. The yellow surfaces are called **isosurfaces** – surfaces of constant value. In the case of plotting ρ or ρ_i , they are surfaces of constant charge density. These isosurfaces are analogous to reading a topographic map in 2D. Instead of lines at constant value in the 2D case, we now visualize surfaces. Try clicking the different options under ‘Style’ on the left to change the appearance of the structure (Fig. 12). You can turn isosurfaces on or off by clicking the ‘Show isosurfaces’ box. And you can change the isosurface style by choosing a different option under ‘Style’ (the second ‘Style’ panel). The ‘Show sections’ box is probably also checked. Sections are cuts through the charge density on the boundary of your plotting box. They appear as contour plots on the sides of the plotting box. These can be useful for periodic systems, but for molecular systems, the density near the edges of your plotting box is so low, there is not much meaningful information there.

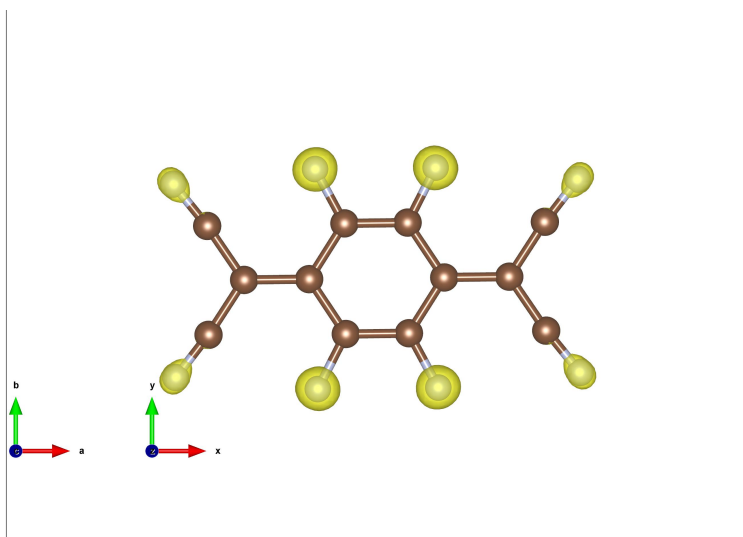


Figure 11

The tools on the left side of the screen (very left side of Fig. 12) allow you to measure bond lengths, angles, etc. with your mouse. For example, click the one with the little arrow between two atoms, then click on two atoms in the structure. In the output dialogue on the bottom, VESTA should tell you the distance between those two atoms. The next tool down the list will allow you to measure bond angles. Click it, then click on three atoms in the structure. The angle formed by these three should be printed in the output dialogue.

If you navigate to **Objects** → **Properties** → **Isosurfaces**, you should see the ‘Properties’ menu shown in Fig. 12. Here, you can change the isosurface value by clicking on the number underneath the ‘level’ label. You can also change the color or add multiple isosurfaces by selecting ‘New.’ Try selecting a reasonable isosurface value that allows you to see the character of the charge density. Selecting isosurface values is subjective and can be tricky, as different values can show very different shapes. One possibility is to select isosurface values that are a certain percentage of the maximum value on the grid, which is conveniently reported by VESTA. Always selecting the same percentage of the maximum offers some plotting consistency between different molecules.

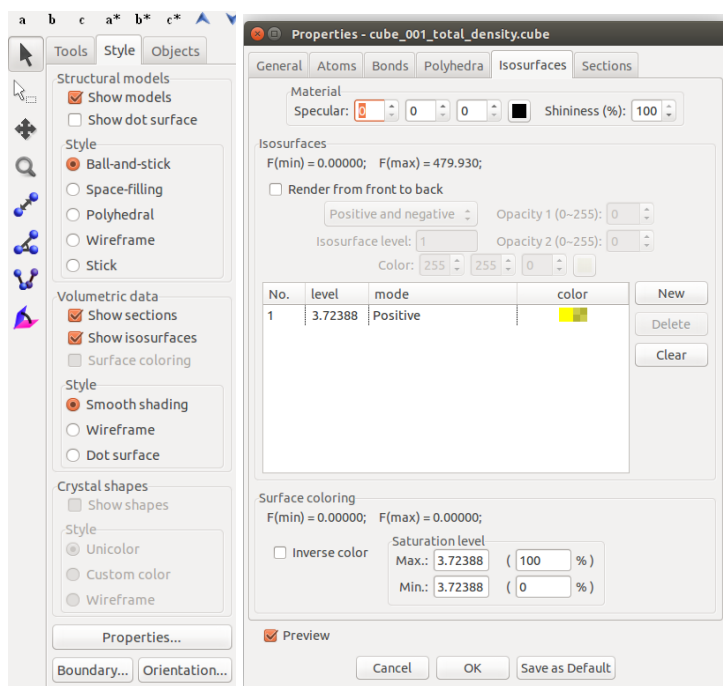


Figure 12

Under the ‘General’ tab (Fig. 13), you can change the overall appearance of the viewer. You can turn on/off the outline of the plotting box, remove the coordinate axes, or adjust the material appearance. Under ‘Atoms,’ you can change the color of each species, the size of each atom, etc. There are lots of customizable aspects to create exactly the look you want. Explore all the options!

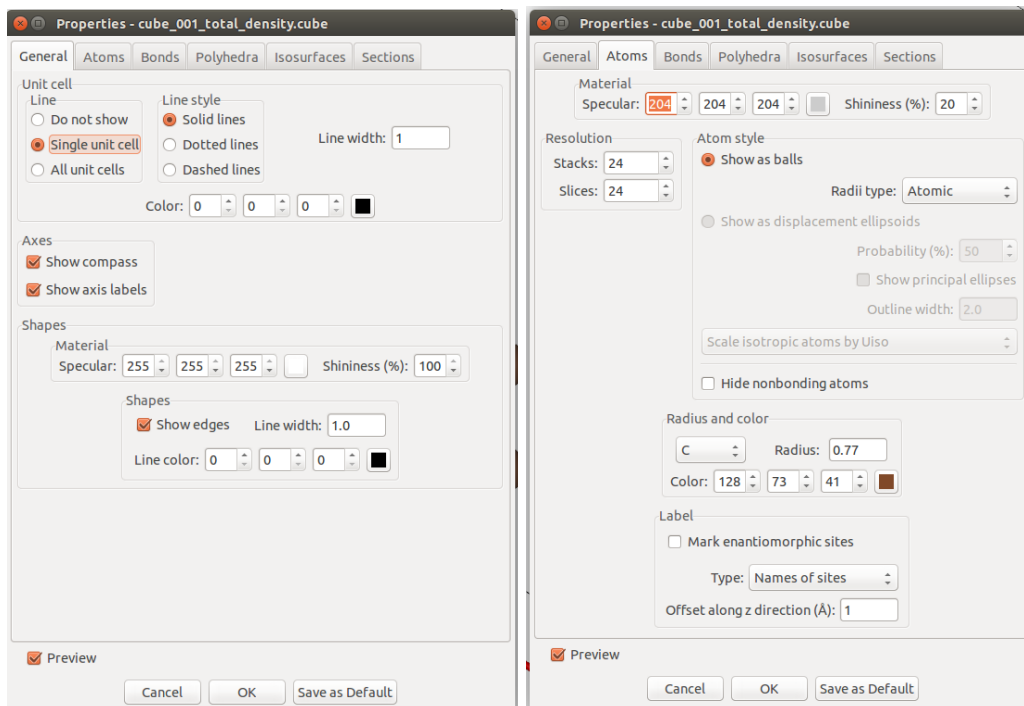


Figure 13

Images can be exported by going to File → Export Raster Image or File → Export Vector

Image. Either of these will save a snapshot of the viewing window. Be sure to zoom in and orient the molecule correctly for the best quality image. That's it! Now you can visualize your results from FHI-aims.

VESTA also has a 2D display mode. Sometimes, making a 2D contour plot on a slice through a certain plane looks better than an isosurface in 3D. To start the 2D mode, click on **Utilities** → **2D Data Display**. You should see the menu shown in Fig. 14.

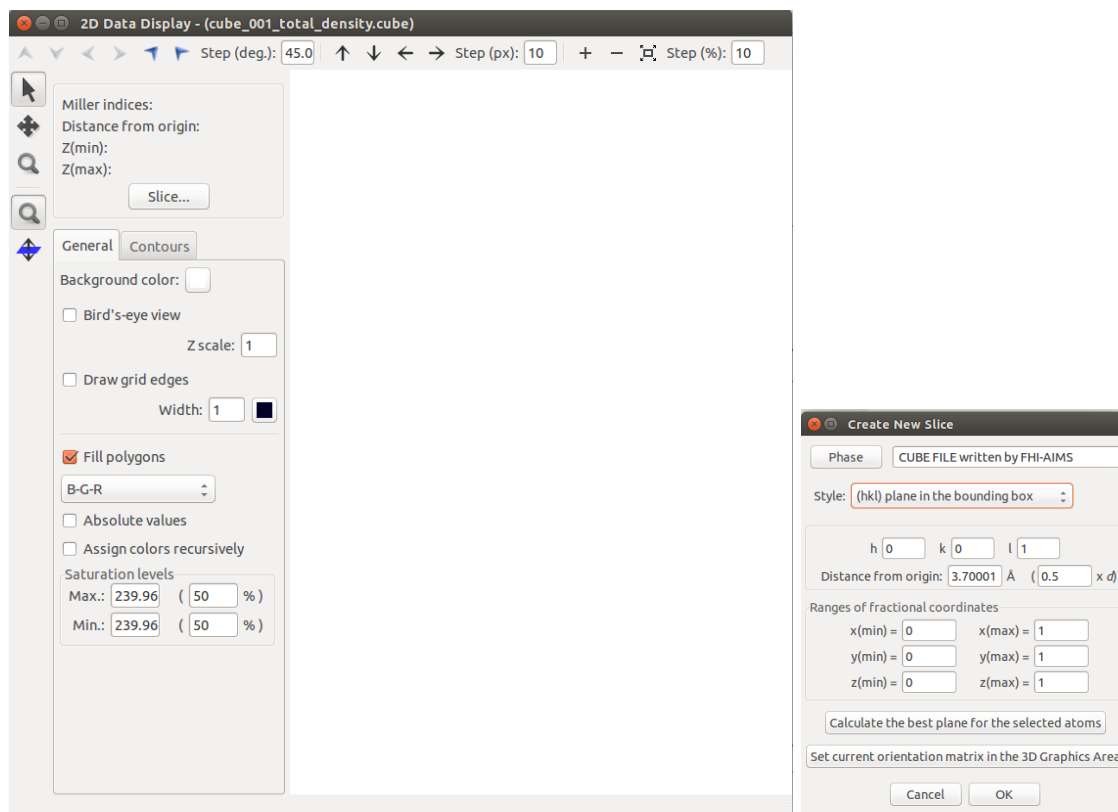


Figure 14

First, you must select which plane you want to make your 2D slice through. Click on ‘Slice’ to get the second menu shown in Fig. 14. Set a plane based on its (hkl) index. Don’t worry about setting the distance from the origin - you will be able to visualize data in *all* planes parallel to the one set by (hkl) . The default setting is $(hkl) = (001)$. If your molecule is planar and oriented in the xy -plane, this is a good choice. Click ‘Ok.’

Now you need to set the saturation levels for the color scale. The maximum value will be the grid value that appears red, and the minimum value will appear blue. These do *not* have to be 100% and 0%. Usually, the maximum and minimum values on the grid are very far apart, with most of the grid points taking a value much closer to the minimum than the maximum. If you simply set the maximum and minimum to 100% and 0%, you probably won’t be able to see much. The majority of the data points will be in a very narrow color band, and you won’t be able to see much detail at all. Next to ‘Max,’ set the saturation level to something around 3%. Then set the ‘Min’ level to 0%. This will probably take some tuning to get a nice image — you can reset these later. Now click on the blue plane with an arrow going through it and scroll with the wheel on the mouse. You should see a new image with every wheel click as you scroll through different planes parallel to the one you set, something like Fig. 15. Each plot is a slice through the density parallel to the (hkl) plane that you set. Try setting different saturation levels and a different (hkl) plane. Again, you can export an image by navigating to **File** → **Export Raster Image**.

Now check the ‘Bird’s-eye view’ box. As long as the blue plane with the arrow going through

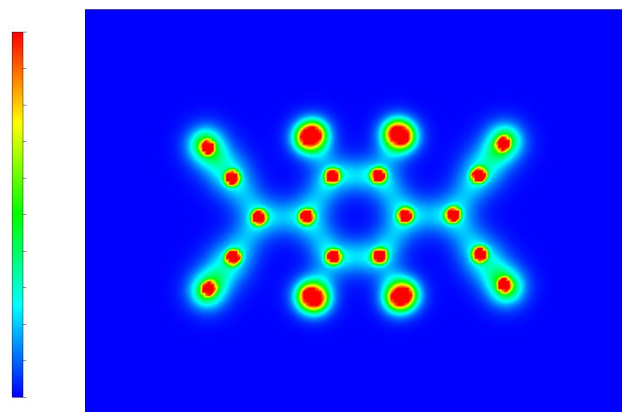


Figure 15

it is checked, you can still scroll through different plane levels. Now, though, click and drag the plotting area. You should be able to rotate the plotting plane, and the plot now has depth. The colors correspond to the height of the features out of the plotting plane. To see more dramatic features, you might want to increase the maximum saturation level. Now you can generate a surface corresponding to the density through a single plane. 2D displays can be easier to look at than 3D isosurfaces, especially for planar molecules.

References

- [1] M. T. Yin and Marvin L. Cohen. “Microscopic Theory of the Phase Transformation and Lattice Dynamics of Si”. In: *Phys. Rev. Lett.* 45 (12 Sept. 1980), pp. 1004–1007. DOI: [10.1103/PhysRevLett.45.1004](https://doi.org/10.1103/PhysRevLett.45.1004). URL: <http://link.aps.org/doi/10.1103/PhysRevLett.45.1004>.
- [2] F. D. Murnaghan. “The compressibility of media under extreme pressures”. In: *Proc. Natl. Acad. Sci.* 30 (9 Sept. 1944), pp. 244–247. DOI: [10.1073/pnas.30.9.244](https://doi.org/10.1073/pnas.30.9.244). URL: <http://www.pnas.org/content/30/9/244.short>.
- [3] Francis Birch. “Finite Elastic Strain of Cubic Crystals”. In: *Phys. Rev.* 71 (11 June 1947), pp. 809–824. DOI: [10.1103/PhysRev.71.809](https://doi.org/10.1103/PhysRev.71.809). URL: <http://link.aps.org/doi/10.1103/PhysRev.71.809>.
- [4] Charles Kittel. *Introduction to Solid State Physics*. 6th. John Wiley & Sons Inc, Dec. 1986. ISBN: 0471874744.
- [5] Robert A. Wolkow. “Direct observation of an increase in buckled dimers on Si(001) at low temperature”. In: *Phys. Rev. Lett.* 68 (17 Apr. 1992), pp. 2636–2639. DOI: [10.1103/PhysRevLett.68.2636](https://doi.org/10.1103/PhysRevLett.68.2636). URL: <http://link.aps.org/doi/10.1103/PhysRevLett.68.2636>.