

The DFTB model in ATK-SE

Tutorial

Version 12.2.0

The DFTB model in ATK-SE: Tutorial

Version 12.2.0

Copyright © 2008–2012 QuantumWise A/S

Atomistix ToolKit Copyright Notice

All rights reserved.

This publication may be freely redistributed in its full, unmodified form. No part of this publication may be incorporated or used in other publications without prior written permission from the publisher.

TABLE OF CONTENTS

1. Introduction	1
2. Installing and testing new DFTB parameter sets	2
Installing DFTB parameters	2
Testing the installation	3
3. Performing a spin polarized calculation with DFTB	6
Calculating the spin polarization of a graphene ribbon	6
Calculate the properties of a series of configurations using scripting	9
4. NEB calculation for a reaction barrier using DFTB	12
Setting up the NEB object	12
Performing the NEB simulation	15
Analyzing the NEB simulation	17
A recipe for faster calculations	17
Bibliography	19

CHAPTER 1. INTRODUCTION

The purpose of this tutorial is to show how to install new parameter sets into the DFTB model [1], [2] in ATK.

ATK is shipped with Slater-Koster parameter files from the [CP2K](#) and [Hotbit](#) consortia, for use in the semi-empirical DFTB model implemented in ATK. This tutorial will show how to download additional parameters from [dftb.org](#) and install them such that they work as a native part of ATK.

Once the parameters are installed they can be used for modeling molecules, bulk and device systems. In this tutorial you will learn how to calculate properties of a spin polarized bulk systems and calculate the reaction barrier for a chemical reaction with a molecule.



Note

You will primarily use the graphical user interface Virtual NanoLab (VNL) for setting up and analyzing the results. To familiarize yourself with VNL, it is recommended to go through the [VNL Tutorial](#).

The underlying calculation engine for this tutorial is **ATK-SE**, the semi-empirical part of Atomistix ToolKit. A complete description of all the parameters, and in many cases a longer discussion about their physical relevance, can be found in the [ATK Reference Manual](#).

In order to run this tutorial, and in general use the semi-empirical models in ATK, you must have a license for ATK-SE. If you do not have one, you may obtain a time-limited demo license by contacting QuantumWise via [our website](#).

You are now ready to [begin the tutorial](#).

CHAPTER 2. INSTALLING AND TESTING NEW DFTB PARAMETER SETS

INSTALLING DFTB PARAMETERS

GETTING A DFTB LICENSE

The dftb.org website has a number of parameters for the DFTB method which can be used with ATK-SE. In order to download the parameters you will need to fill out their [registration form](#). When your registration has been processed, you will receive a username and password to their website.

DOWNLOADING THE PARAMETERS

On the [DFTB download page](#) you can get an overview of the different DFTB parameter sets. The primary ones are named [mio](#), [pbc](#), and [matsci](#); the "mio" set can further be extended by a number of specialized sets listed farther below. The page also lists the papers that must be referenced when using the parameter sets.

Log into dftb.org and download the parameter sets of your interest. The downloaded files will be in a compressed tar ball (`.tar.gz` files).

INSTALLING THE PARAMETERS

ATK has a special folder for each of the three DFTB parameters sets. For instance, to install the `mio` parameter set, unpack the downloaded tar ball and copy the `.skf` files to the directory

```
atkpython/share/tightbinding/dftb/mio/
```

under the directory where ATK is installed. In Linux you can use the following commands (to be issued in the ATK installation directory, where it is also assumed the downloaded files reside)

```
tar zxvf mio-1-1.tar.gz
cp mio-1-1/* atkpython/share/tightbinding/dftb/mio
```

Similarly, the `pbc` and `matsci` parameter files should be copied to the `dftb/pbc` and `dftb/matsci` directories. Note that these directories already exist, but are empty.

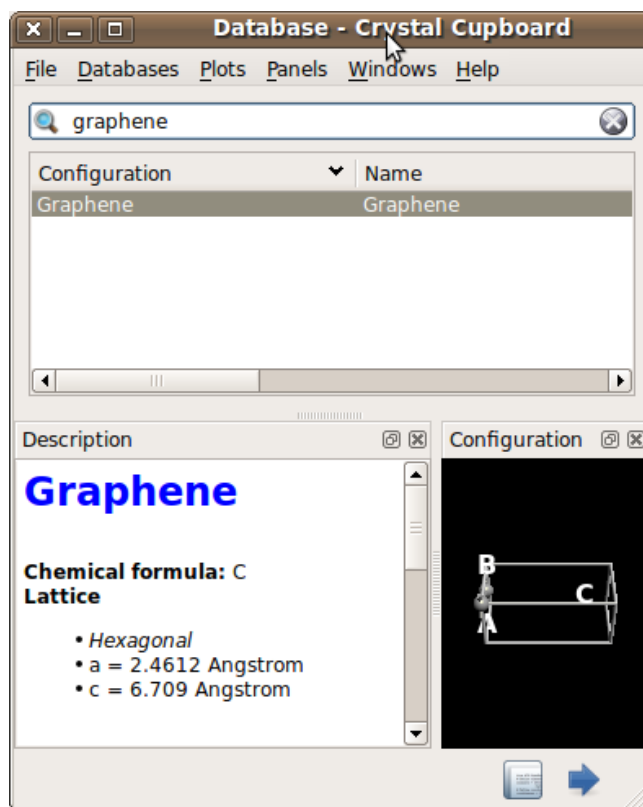
 **Note**


If you are using VNL on a laptop/workstation to set up the calculations, and then a separate installation of ATK on a cluster for running the calculations, then the parameter files must be installed on both computers separately.

TESTING THE INSTALLATION



SETTING UP A BANDSTRUCTURE CALCULATION

To test that the parameters are correctly installed you can perform the following bandstructure calculation for graphene. Open the database tool in VNL from the menu **Tools>Database**, and type graphene in the search field

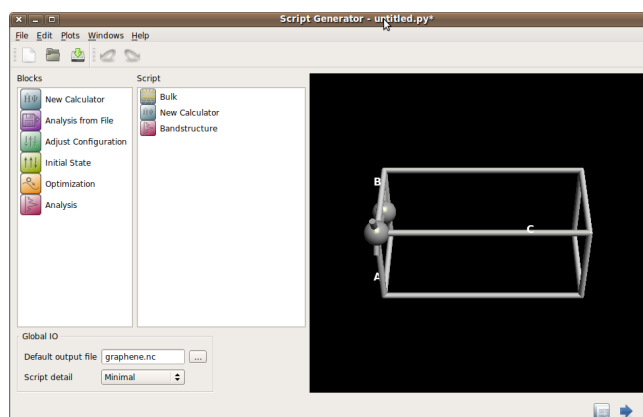


Send the structure to the **Script Generator** using the "Send To" button .

In the Script Generator, add a

-  **New Calculator** block;
-  **Bandstructure** analysis block.
- Change the output filename to `graphene.nc`

The Script Generator should now have the following settings



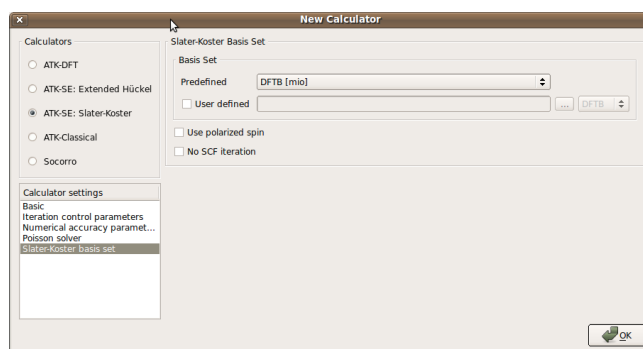
Now open the calculator block and

- select the "ATK-SE: Slater-Koster" calculator,
- change the k-point sampling to (5,5,1).
- Go to the "Slater-Koster basis set" and check that the installed basis sets are in the basis set list. Select the "DFTB [mio]" basis set.

Note

If you do not see the "DFTB [mio]" basis set, something went wrong in the installation process. Check that the file `atkpython/share/tightbinding/dftb/mio/C-C.skf` exists in your ATK installation directory.


- Uncheck the **No SCF iteration** box, to perform a self-consistent DFTB calculation.



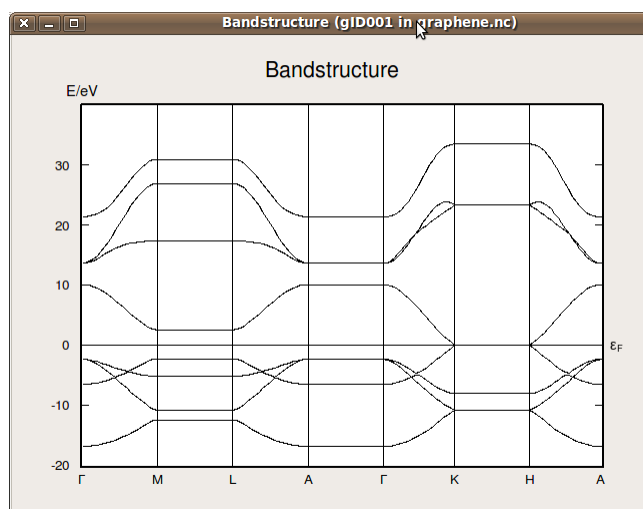
Note

The default assumption is that a tight-binding model is non-self-consistent. It is the responsibility of the user to uncheck the **No SCF iteration** box for models which are self-consistent. Most DFTB models are self-consistent.

RUNNING THE TEST CALCULATION

Transfer the script to the **Job Manager** using the "send-to"-button  and start the calculation.

When the job has finished, locate the file `graphene.nc` in the file browser in the main VNL window and plot the band structure. You should get the result shown below



If the test went well then the DFTB parameters are properly installed. You will now be able to use DFTB parameters instead of DFT for some of the other tutorials available on [the Quantum-Wise website](#) (including quantum transport calculations), in order to save some time. Obviously this only works if the elements used are covered by the DFTB basis sets.

The next chapters illustrate how to perform [spin polarized calculations](#) and [calculations of chemical reactions](#) with the DFTB method.

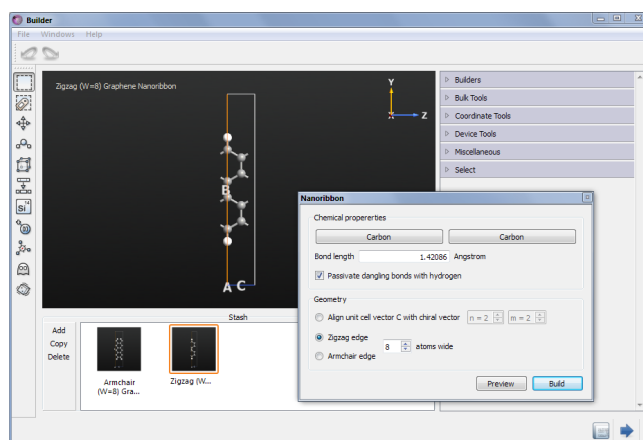
CHAPTER 3. PERFORMING A SPIN POLARIZED CALCULATION WITH DFTB


CALCULATING THE SPIN POLARIZATION OF A GRAPHENE RIBBON

In the following you will set up a graphene nanoribbon and perform a spin-polarized calculation using the DFTB [mio] model.




Open the **Builder** , and click **Add>Add Custom>Nanoribbon**.

Build a zigzag ribbon of width 8 atoms.

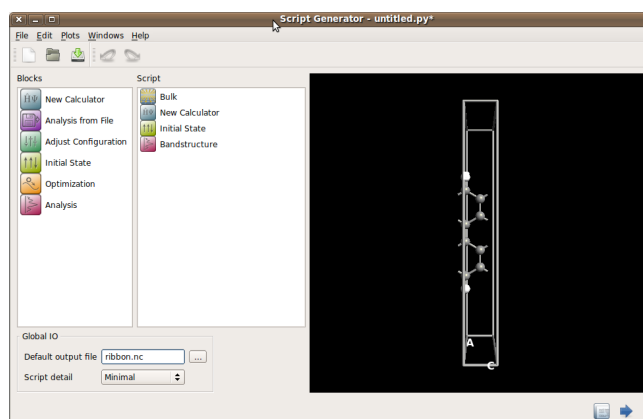


To set up a calculation of the band structure, send the structure to the Script Generator using the "Send To" icon  in the lower right-hand corner of the builder window.

In the Script Generator, add the following blocks to the script:

-  **New Calculator**
-  **Initial State**
-  **Analysis/Bandstructure**

Finally, change the output filename to ribbon.nc

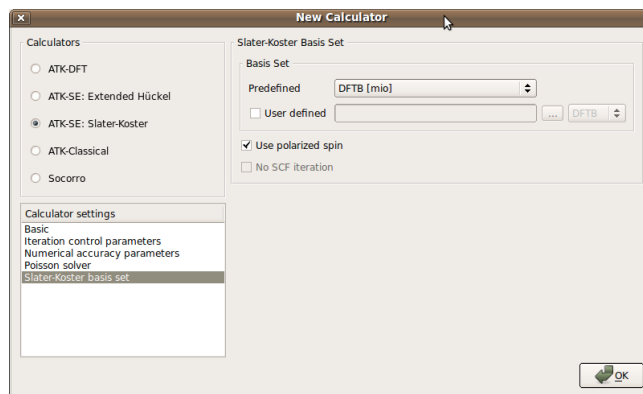


Open the  **New Calculator** block.

- Select the "ATK-SE: Slater-Koster" calculator.
- Change the k-point sampling to (1,1,11).
- Go to the "Slater-Koster basis set" and select the "DFTB [mio]" basis set.
- Check the **Use polarized spin** box, to perform a spin polarized DFTB calculation.

 **Note**

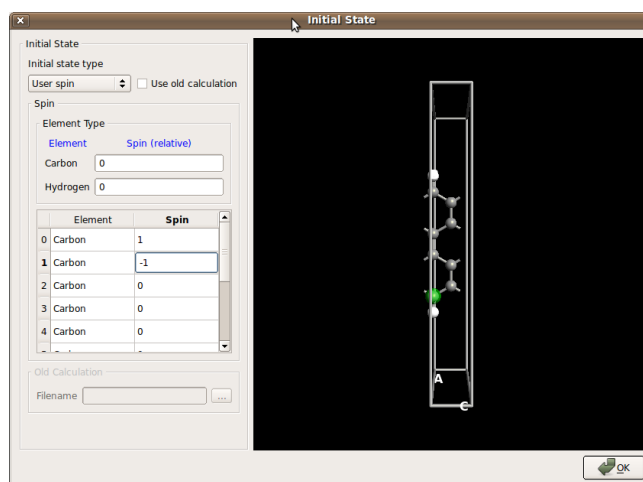
The DFTB parameter set does not include the spin polarization parameters. Instead, the spin polarization parameters are obtained from the [ATK_W](#) database.



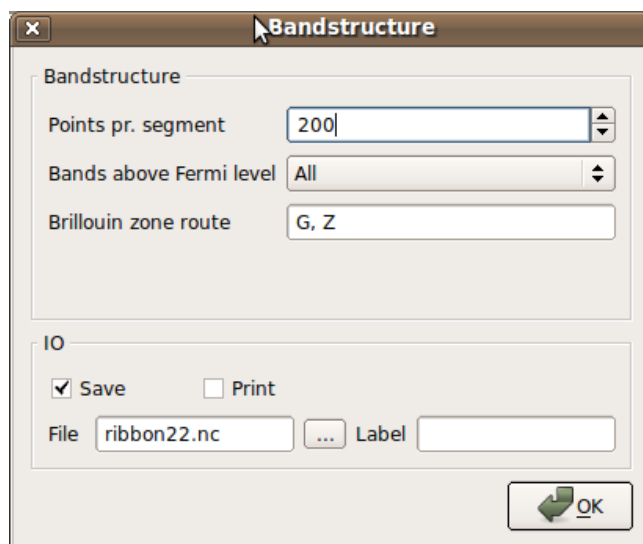
To set up the initial spin state of the system, open the **Initial State** block.

Set the **Initial state type** to "User spin", and then under **Spin**, set the default spin of both carbon and hydrogen to 0.

Then, in the 3D view, click the upper carbon atom (index 0 in the list of atoms) and set its initial spin value to 1.0. Repeat this procedure for the lower carbon atom (number 1 in the list) setting its initial values to -1.0. The hydrogen atoms and the other carbon atoms can be left with spin values equal to 0.0. The dialog now look as (don't be confused by the fact that the carbon atoms are not ordered by Y coordinate)

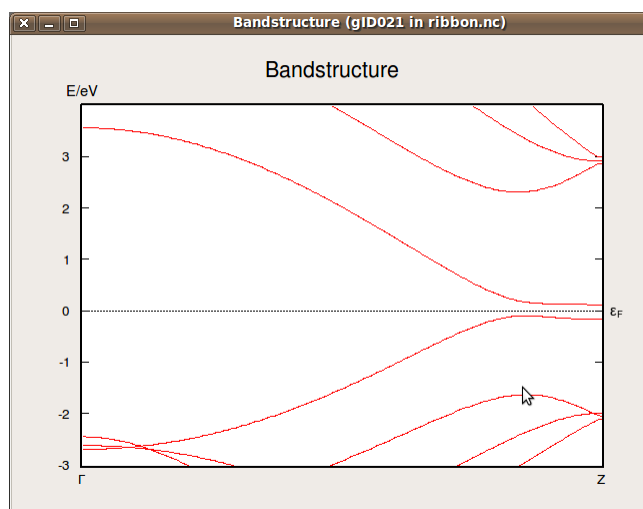


Finally open the **Bandstructure** block and change **Points pr. segment** to 200. This implies, that each band will be calculated with a resolution of 200 points. Also notice that the default suggested route in the Brillouin zone (from the Γ point $G=(0,0,0)$ to $Z=(0,0,1/2)$ in units of the reciprocal lattice vectors) is the appropriate one.



Send the script to the Job Manager and run it.

Return to the main VNL window and select the file `ribbon.nc` and plot the bandstructure. By zooming in, into the plot, you should be able to obtain the following figure.



CALCULATE THE PROPERTIES OF A SERIES OF CONFIGURATIONS USING SCRIPTING

If you want to calculate the bandstructure of a number of nanoribbons it is very convenient to use Python scripting. The script below performs a calculation of the bandstructure of nanoribbons with chiral index (n,n) , where n runs from 1 to 10.

Note

The "chiral index" notation for nanoribbons in ATK is based on which linear combination of the planar graphene unit vectors that are used to form the unit cell. Thus, (n,n) have zigzag edges, and actually correspond to an unrolled armchair nanotube with the same chiral indices. For armchair edge nanoribbons, the chiral indices are $(n,0)$.

```
# Setup the DFTB Calculator
basis_set = DFTBDirectory("dftb/mio/")
pair_potentials = DFTBDirectory("dftb/mio/")

numerical_accuracy_parameters = NumericalAccuracyParameters(
    k_point_sampling=(1, 1, 51),
)

iteration_control_parameters = IterationControlParameters()

calculator = SlaterKosterCalculator(
    basis_set=basis_set,
    pair_potentials=pair_potentials,
    numerical_accuracy_parameters=numerical_accuracy_parameters,
    iteration_control_parameters=iteration_control_parameters,
    spin_polarization=True,
)

#loop over different NanoRibbons
for n in range(1,10):
    #generate the nanoribbon
    bulk_configuration = NanoRibbon(n,n)

    #Determine the initial spin of the configuration
    elements = bulk_configuration.elements()
    coords = bulk_configuration.fractionalCoordinates()
    scaled_spins = numpy.zeros(len(elements))

    #find the index of the two edge Carbon atoms,
    ymin = 0.5
```

```

ymax = 0.5
imin=0
imax=0
for i in range(len(elements)):
    if coords[i][1] < ymin and elements[i] == Carbon:
        ymin = coords[i][1]
        imin = i
    if coords[i][1] > ymax and elements[i] == Carbon:
        ymax = coords[i][1]
        imax = i

# set opposite spins on the edge Carbon atoms
scaled_spins[imin] = 1
scaled_spins[imax] = -1

#attach calculator, and set the initial spin
bulk_configuration.setCalculator(
    calculator(),
    initial_spin=InitialSpin(scaled_spins=scaled_spins),
)
bulk_configuration.update()

# Calculate Bandstructure
bandstructure = Bandstructure(
    configuration=bulk_configuration,
    route=['G', 'Z'],
    points_per_segment=200,
    bands_above_fermi_level=All
)
nlsave('bandgap.nc',bandstructure)

```

Execute the script by downloading it and dropping it on the Job Manager. This will generate the file `bandgap.nc`.

To analyze the data, drop the following script on the Job Manager

```

w_list = []
gap0_list = []
gap1_list = []
# read a list with the bandstructure data
bandstructure = nlsread('bandgap.nc', Bandstructure)

for i in range(len(bandstructure)):
    # get all the bandlines
    energies = bandstructure[i].evaluate().inUnitsOf(eV)
    #calculate the number of occupied bands
    occupied_bands = numpy.sum(numpy.array(energies[0]) < 0)
    #calculate the minimum direct band gap
    gap0 = numpy.min(energies[:,occupied_bands]- energies[:,occupied_bands-1])
    #calculate the band gap at the zone boundary
    gap1 = energies[-1,occupied_bands]- energies[-1,occupied_bands-1]
    #determine the coordinates in the y direction
    y_values = NanoRibbon(i+1,i+1).cartesianCoordinates().inUnitsOf(Ang)[: ,1]
    # calculate the width, as distance between furthest carbon atoms
    # i.e. calculate the H-H distance and subtracting the C-H distance
    w = numpy.max(y_values)-numpy.min(y_values)-2.2
    #append calculated values to the list
    w_list.append(w)
    gap0_list.append(gap0)
    gap1_list.append(gap1)

#print the data
import pylab
pylab.figure()
pylab.plot(w_list,gap0_list, 'k-o')
pylab.plot(w_list,gap1_list, 'k-o', markerfacecolor='white')
pylab.xlabel(r"$w_z$ ($\AA$)", size=16)

```

```
pylab.ylabel("\Delta_z$ (eV)", size=16)
pylab.show()
```

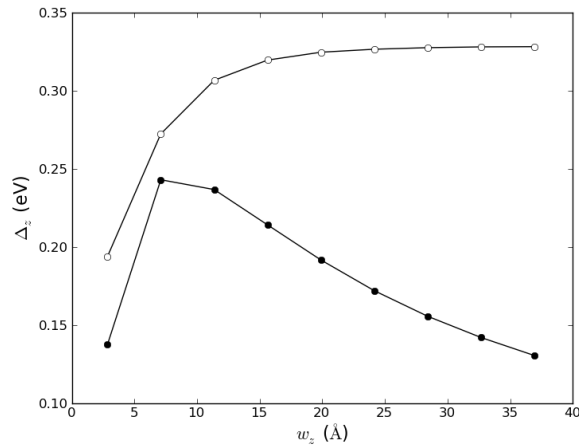


Figure 3.1: Direct band gap (solid circles) and the band gap at the zone boundary as function of the width of the nanoribbon.



Note

Except for the smallest ribbon, the plot shows the same qualitative behavior as Fig. 4c in Ref. [3]. The discrepancy in the results are related to the accuracy of the DFTB method compared with a DFT-LDA calculation. To test this, you may modify the script and change the definition of the variable **calculator**:

```
calculator = LCAOcalculator(  
    numerical_accuracy_parameters=numerical_accuracy_parameters,  
    exchange_correlation = LSDA.PZ  
)
```

Remember to also modify the name of the output file to separate the new data from the old data set and then execute the script.

CHAPTER 4. NEB CALCULATION FOR A REACTION BARRIER USING DFTB

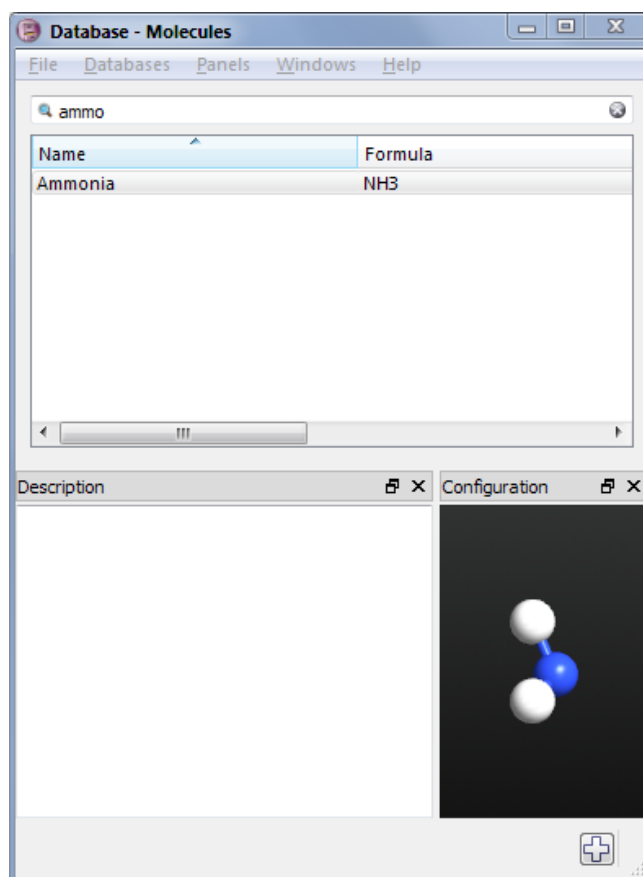
The following example illustrates how to use the DFTB model to calculate the reaction barrier for inverting an ammonia molecule. ATK uses the Nudged Elastic Bands (NEB) method [4] to find the reaction pathway and the barrier.

SETTING UP THE NEB OBJECT

Open the **Builder** .


Click the Add button in the lower left-hand panel and select **Add From Database...**

The Database tool will open. From the menu, select Databases → Molecules. Type NH3 in the search field, and you should see the following.

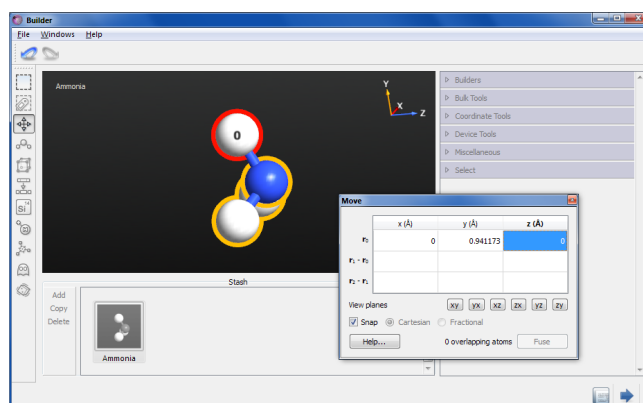


Double-click the "Ammonia" entry (or the ) icon), and a NH₃ molecule will be added to the stash.

When running NEB calculations it is important to restrict at least a few degrees freedom. What primarily happens in the ammonia reaction considered here is that the nitrogen atom moves through the hydrogen plane. It is therefore a good idea to keep the hydrogen atom plane fixed when defining the initial reaction path. This can be achieved rather easily.

Activate the **Move Tool** . Select one of the hydrogen atoms as an anchor atom (marked by a red halo); this is the atom you will operate on. Since no atoms were selected when the Move tool was opened, all atoms will simultaneously be selected (marked by a yellow halo); these atoms will be affected by any operation carried out in the Move tool.

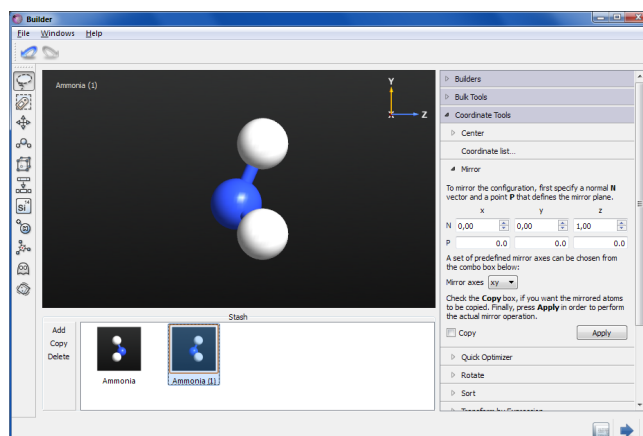
In the **Move** widget, enter the value 0 for the z coordinate of the selected atom, and press **Enter**.



Close the Move tool by closing the widget or clicking the select tool.

Now ensure that the Ammonia molecule is selected on the stash by left-clicking it, then click the **Copy** button. This adds an additional, identical ammonia molecule to the stash.

Open the **Mirror** tool from the **Coordinate Tools** in the right tool bar. Ensure that the mirror plane has a normal parallel to the Z axis and contains the origin (this is the default), and press **Apply** to invert the molecule.

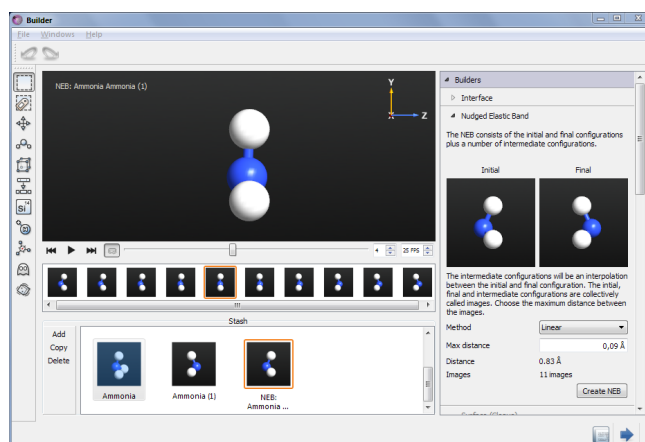


You have now defined the initial and final states of the reaction path, and the next step is to set up the NEB configuration by inserting a number of images in-between these states.


To this end, open the **Create Nudged Elastic Band** tool under **Builders** in the right-hand panel, and drag the two configurations into the "Initial" and "Final" slots.

Set the maximum distance between images to 0.09 Å to have an odd number of images, which is very important since the reaction path is symmetric. Choose the **Linear** method and press **Create NEB**.



The NEB object will now be built and added to the stash. Push the play button to see the different configurations that will be used to as starting configurations when optimizing the reaction pathway.

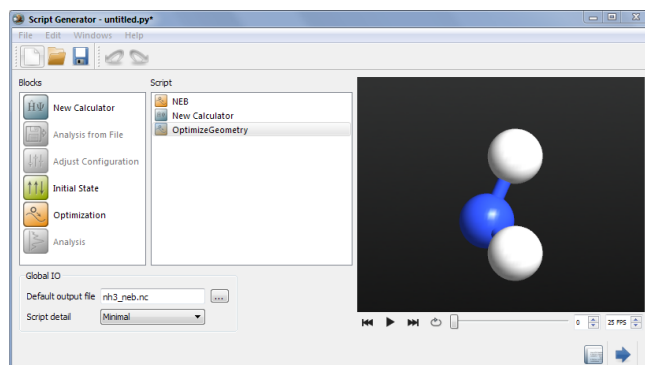


PERFORMING THE NEB SIMULATION

To set up the NEB calculation, drag the NEB configuration from the stash area and drop it onto the **Script Generator** .

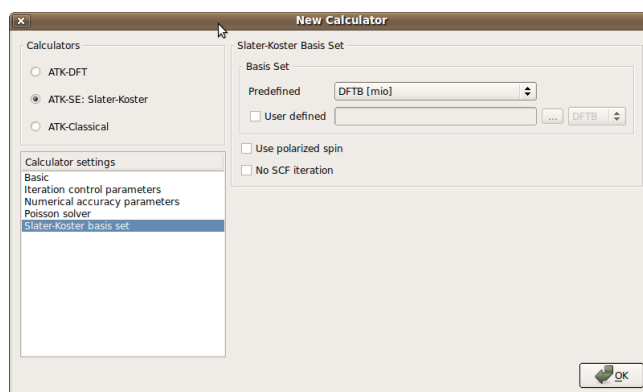
In the Script Generator,

- Add a  **New Calculator** block.
- Add an  **OptimizeGeometry** block.
- Change the default output name to `nh3_neb.nc`

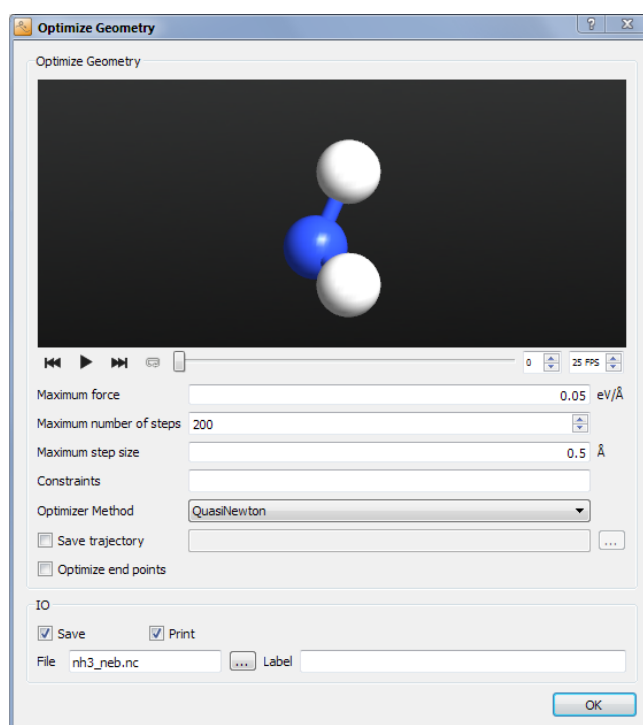


Now open the  **New Calculator** block.

- Select the "ATK-SE: Slater-Koster" calculator.
- Go to the "Slater-Koster basis set" and check that the "DFTB [mio]" basis set is selected.
- Uncheck the **No SCF iteration** box.



Open the  **OptimizeGeometry** block.



You are now ready to perform the NEB optimization. Send the script to the [Job Manager](#) and execute it.

The optimization will take 20-30 minutes depending on your computer.

 **Note**

While most other implementations of the DFTB model use a pointwise electrostatic pair potential interaction model, ATK-SE uses a Poisson solver for calculating the electrostatic interactions. This approach is significantly slower for small systems, like the current NH₃ molecule, but faster for large systems where it scales as $O(N)$ instead of a typical $O(N^2)$ scaling.

The use of a Poisson solver ensures that the DFTB model is equivalent to the other calculators in ATK, and thus can be used for modeling device geometries and allows for using implicit solvent models.

ANALYZING THE NEB SIMULATION

Return to the main VNL window and select the file `nh3_neb.nc`. Select the last NEB configuration in the file (the one with the highest ID number). Press the Show configuration and you will see the calculated energy barrier and the corresponding configurations in the reaction path. The reaction path can also be illustrated as an animation by pressing the play button.

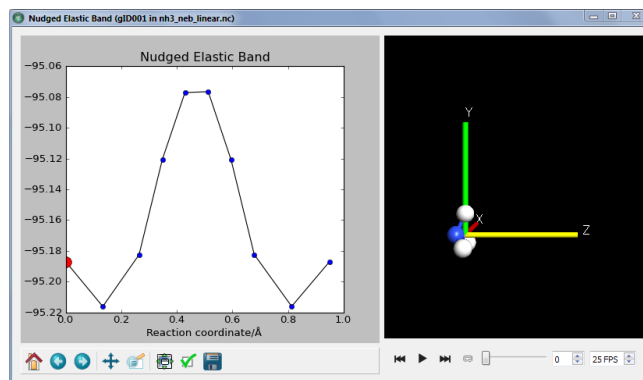


Figure 4.1: Reaction path of the ammonia inversion, computed with the DFTB [mio] method. The end-points were not optimized.

Note

The obtained reaction barrier is not fully accurate since the end points were not optimized. This can not least be seen from the fact that the second image has a lower energy than the end-points. For real work with the NEB method the end-points should always be optimized first. If you do this for the ammonia molecule, you will obtain the barrier shown in the figure below.



Figure 4.2: Reaction path of the ammonia inversion, computed with the DFTB [mio] method. In this case the end-points were first optimized, and the image distance was also adjusted (0.07 Å) to have the same number of images as above.

Tip

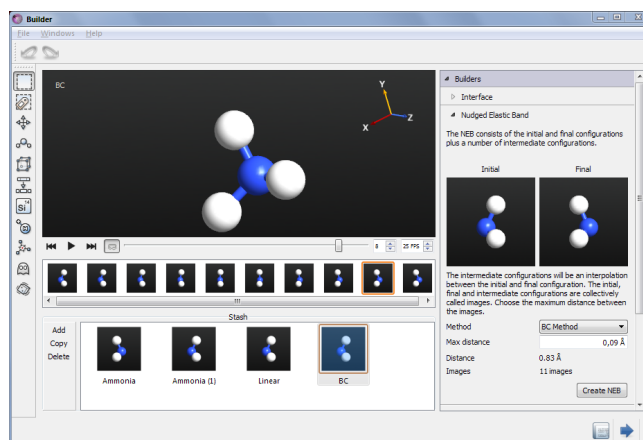
You can inspect an individual optimized configurations in the reaction path closer by clicking the dots in the reaction paths plot (and stopping the movie).

A RECIPE FOR FASTER CALCULATIONS

In the calculation above you used a linear interpolation between the end points to set up the initial guess for the NEB path. In many cases the linear path is however very far from the optimized one. Comparing the movie of the converged result above to the initial NEB path you set up, it can be seen that that the hydrogen atoms need to move out to make room for the nitrogen atom as it passes through the center of the molecule, and then they move back in. If this behavior somehow could already be part of the initial path, one would save some steps in the ionic dynamics, and hence time. However, since the initial and final positions of the hydrogen atoms are identical, this cannot be captured by the linear interpolation.

There are however other methods for generating the initial path, and these are available in ATK. Go back to the Builder, and the Nudged Elastic Band plugin. Keep the image distance but

change the Method to **BC Method** and click Create NEB. If you now look at the initial path, in the Builder, you can see that it already contains this behavior of the hydrogen atoms.



Follow the same procedure as above to run the NEB calculation via the Scriptor, and you will find that it runs about 30% faster this time, but the results are identical.



Note

The BC (bond conservation) method is similar to the Halgren-Lipscomb method [5] which is also available in ATK (but not suitable for this particular system).

BIBLIOGRAPHY

- [1] M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haug, T. Frauenheim, S. Suhai, and G. Seifert, *Phys. Rev. B*, **58**, 07260, 1998.
- [2] C. Kohler, T. Frauenheim, B. Hourahine, G. Seifert, and M. Sternberg, *J. Phys. Chem. A*, **111**, 5622, 2007.
- [3] Y.W. Son, M.L. Cohen, and S.G. Louie, *Phys. Rev. Lett.*, **97**, 216803, 2006.
- [4] G. Henkelman, B. P. Uberuaga, and H. Jonsson, *J. Chem. Phys.*, **113**, 9901, 2000.
- [5] T. A. Halgren and W. N. Lipscomb, *Chem. Phys. Lett.*, **49**, 225, 1977.