

Transport in Graphene Nanoribbons

Tutorial

Version 12.2.0

Transport in Graphene Nanoribbons: 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
From graphite to graphene nanoribbons	1
Preparations	2
Prerequisites	2
2. Band structure of 2D graphene	3
Preface	3
Constructing graphene	3
Calculate the graphene band structure	4
The same thing, but with DFT	8
3. Band structure of an armchair ribbon	10
Building the armchair ribbon structure	10
Calculating the band structure	11
4. Transport properties of a zigzag nanoribbon	13
Transport properties of a perfect ribbon	13
Creating a distortion in the ribbon	15
Calculating the properties of the distorted ribbon	17

CHAPTER 1. INTRODUCTION

Research on fundamental aspects as well as device applications of graphene is a very active field. This tutorial illustrates how the capabilities of Atomistix ToolKit can be utilized to study various applications related to graphene nanoribbons, ranging from a simple, infinite sheet of graphene to more complex junctions.

The main purpose of the tutorial is to demonstrate the basic work-flow in VNL, how to build systems with nanoribbons and compute fundamental quantities such as band structures and transmission spectra.

FROM GRAPHITE TO GRAPHENE NANORIBBONS

The geometry of graphene is simple and regular, and the infinite, planar structure can easily be created either by hand or, as you will learn soon, by taking a single layer from the crystal structure of graphite.

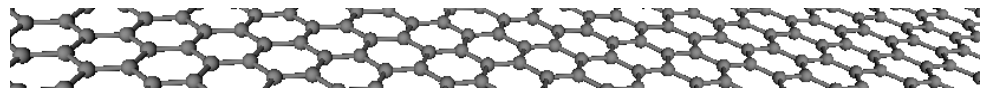


Figure 1.1: A graphene sheet.

However, to create a device-like structure, the infinite sheet must be cut into a suitable shape. A common such shape, at least for electronics applications, is a so-called graphene nanoribbon (GNR). These can be quite cumbersome to set up in an effective manner, not least considering the hydrogen passivation of the edges, which is needed in finite structures in order to satisfy all carbon valence electron bonds.

Virtual NanoLab has a specific graphene nanoribbon builder tool, and you will in this tutorial learn how to use it, and how to customize the structures it produces to make more advanced shapes of graphene. This will allow you to

- calculate the band structure of ideal, infinite graphene nanoribbons, of either armchair or zigzag type, as a function of the ribbon width;
- introduce doping or defects, on the edge or elsewhere, and study the influence on the transmission spectrum;
- use basic ribbons as building blocks for your own device ideas, such as e.g. Z-shaped nanoribbon junctions.

The tutorial will show examples of each type of system.

PREPARATIONS

Before commencing the tutorial, it is recommended to create a dedicated, new directory for storing all the input and output files for this tutorial. This directory will be referred to as the **working directory** of this tutorial. It is recommended to store it directly under the main ATK workspace directory.

PREREQUISITES



Note

This tutorial is based on operations in the graphical user interface VNL and involves almost no scripting at all. It is assumed that you are familiar with the basic operations in VNL, as introduced in the [VNL Tutorial](#).


Finally, you are ready to [begin this tutorial](#).

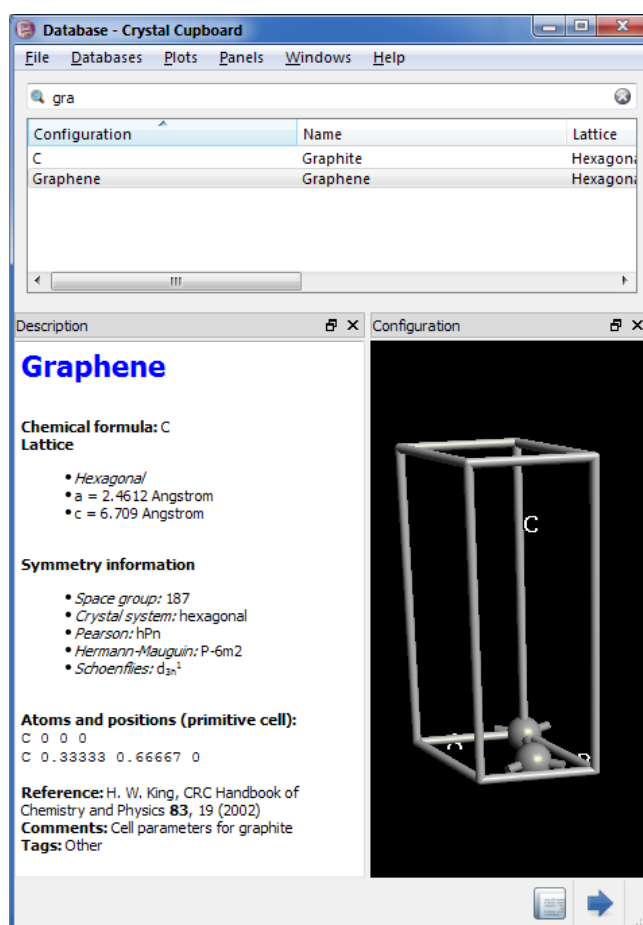
CHAPTER 2. BAND STRUCTURE OF 2D GRAPHENE


PREFACE

Before starting with the nanoribbons, you will do a quick introductory calculation of the band structure of an infinite 2D sheet of graphene. This will also serve as a way to refresh the basic concepts and work flow in the graphical user interface!

CONSTRUCTING GRAPHENE

Graphene, and a large amount of other crystal, molecular and fullerene structures, are available in the built-in database in VNL. Open the **Database** tool  via the menu **Tools>Database** in the main VNL window. Select **Crystal** from the **Databases** menu, if it is not already selected. Then, in the search field, type “graphene” (while you are typing, you will see that graphite is included in the database too).



The structure is ready for immediate use, so transfer it to the **Script Generator** by clicking the "Send To" icon  in the lower right-hand corner of the window, and select **Script Generator** (the default choice, highlighted in **bold**) from the pop-up menu.

Tip

The "Send To" menu can also be activated from the keyboard by pressing Alt-S. The default choice is invoked directly by pressing Ctrl-Alt-S.

CALCULATE THE GRAPHENE BAND STRUCTURE

You will now set up the band structure calculation. It is instructional for this particular system to compare the results of a DFT calculation with the semi-empirical method; as you will see, they are very similar.

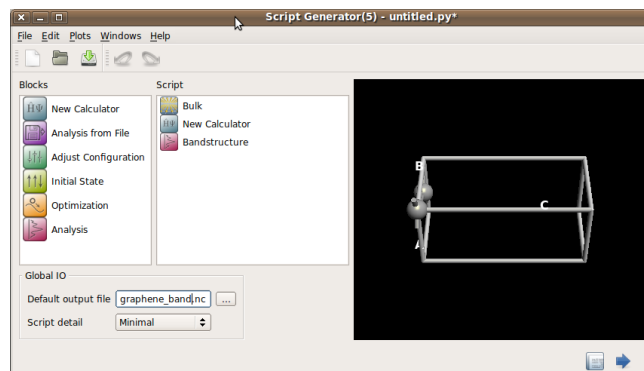
The first step is to use the **Extended Hückel** calculator to determine the required k-point sampling. Using the faster method saves time compared to doing the same investigation with DFT. The results should, for this aspect, be very similar.

To setup the calculation, insert a **New Calculator** and an **Analysis Band structure** block into the script.

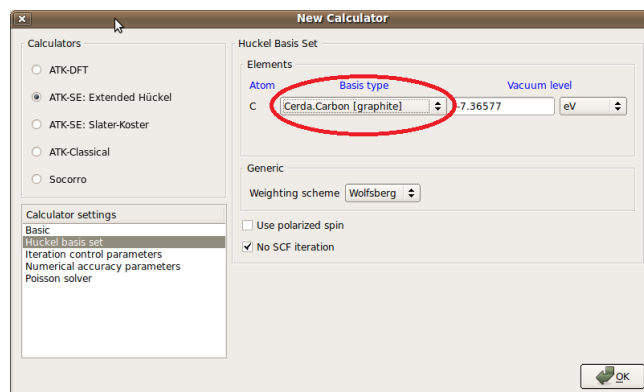
Define the **Default output file** by

- double-clicking the "... " icon to the right and navigate to the tutorial working directory
- and specify the filename `graphene_band.nc`.

The Scripser tool should now have the following settings

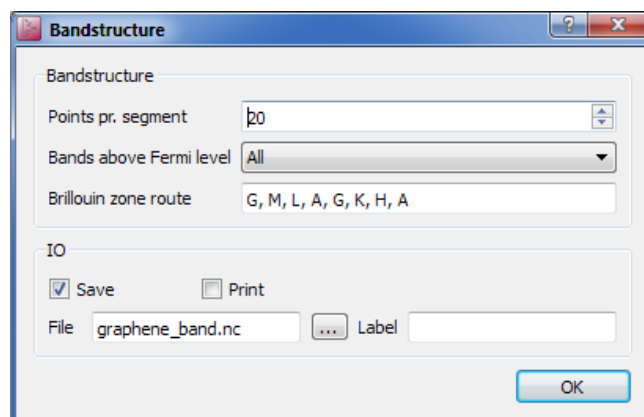


You will use default parameters, except the for the basis set parameters. Double-click the **New Calculator** block in the middle panel and change the basis type to the **Cerda graphite** parameter set under **Basis set**.




Make sure to use this basis set whenever studying graphene (or carbon nanotubes) with the extended Hückel method.

Now double-click **Bandstructure** in the middle-panel. A pre-defined route in the Brillouin zone is set up corresponding to a three-dimensional hexagonal Bravais lattice. Leave it like that for now; this point will be discussed soon.

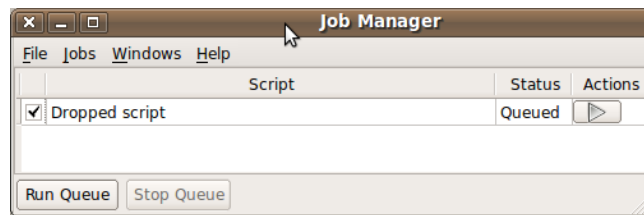


 **Note**

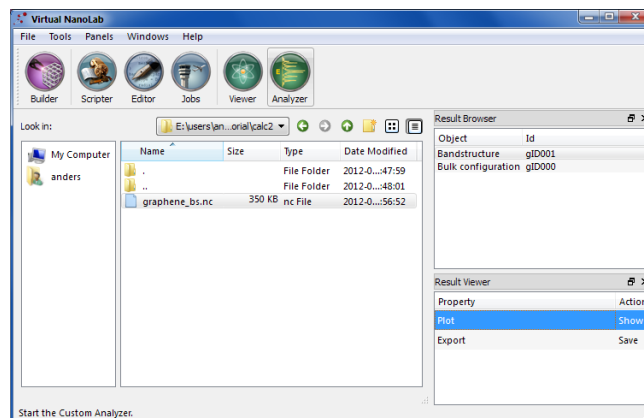
The different components of the script inherit the global output file name. It is possible to specify a unique file name for each quantity, but usually it makes most sense to collect all results in the same file.

Now, to run the calculation, click the "Send To" icon  and select Job_Manager from the pop-up menu.

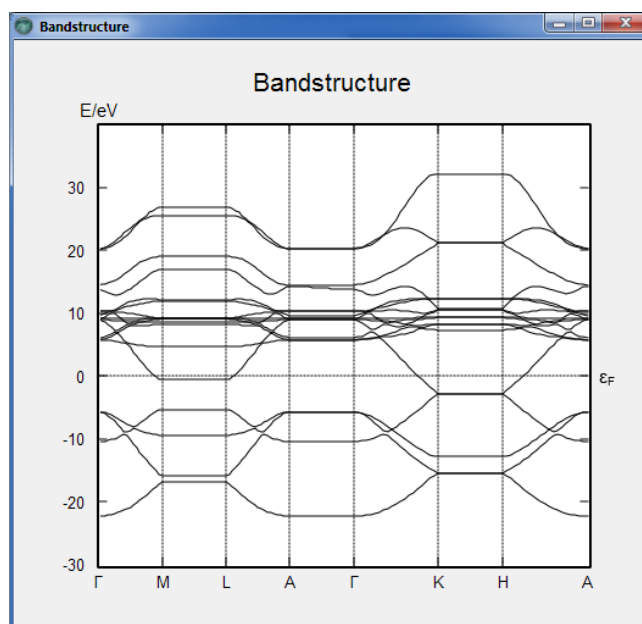
In the Job Manager, click **Start** to run the job; the calculation takes only seconds to complete.



Without closing the open windows, return to the main VNL window, and navigate to the tutorial working directory. Select the newly created NetCDF file, and note how the contents of the file are listed in the upper right-hand panel. Select the **Bandstructure** and then click the **Action** button **Show** next to the **Property** label **Plot** in the lower right panel.



On closer inspection, you can see that the band structure does not appear to be correct, since the K point is supposed to coincide with the Fermi level.

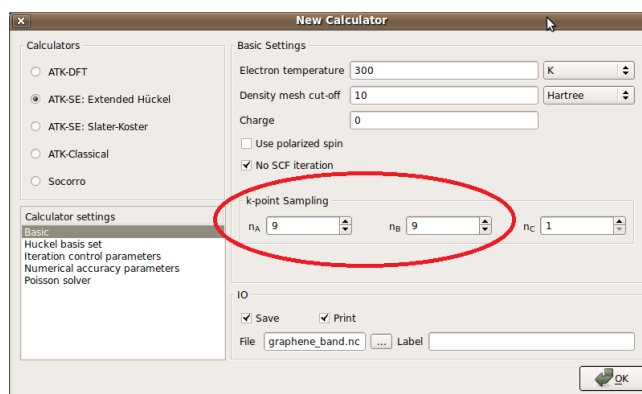


i Tip

By default, the quality of the 2D plot is reduced for quicker zooming etc. For better quality, right-click the plot and select Anti Aliasing from the context menu.

The cause of the poor band structure is insufficient k-point sampling; the default is to have just one point in each direction, and this fails to describe the graphene electronic structure properly.

Therefore, open the minimized **Script Generator** window again, double-click the **New calculator** block, and enter 9x9 k-points under "Basic settings".



You should also adjust the symmetry points used for the band structure calculation, to avoid the flat segments which corresponds to paths out of the planar graphene Brillouin zone. The only relevant k-points in a 2D hexagonal lattice are $G=(0,0,0)$, $K=(1/3,1/3,0)$, and $M=(0,1/2,0)$, in units of the reciprocal lattice vectors.

This can easily be done in the **Scripter** (double-click the **Bandstructure** block and edit the path), but to demonstrate how easy it is to use a script produced by the **Script Generator** as a template, you will now learn to do that by hand instead!

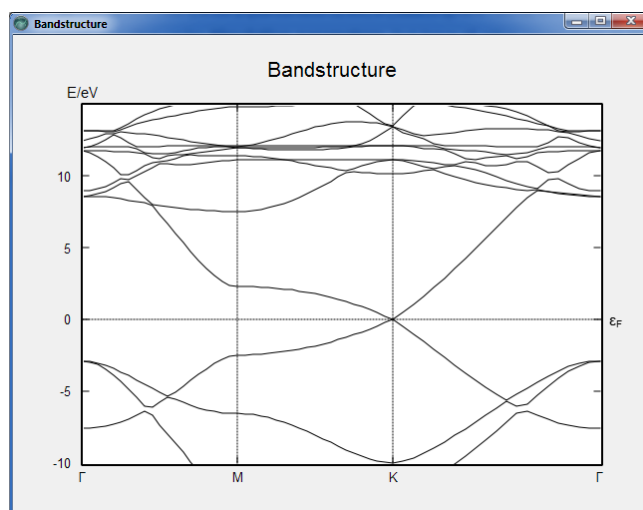
Send the script to the **Editor**, locate the line with the symmetry points, and edit it.

```

40
41 bulk_configuration.setCalculator(calculator)
42 nprint(bulk_configuration)
43 bulk_configuration.update()
44 nlsave('graphene_band.nc', bulk_configuration)
45
46 -----
47 # Bandstructure
48 -----
49 bandstructure = Bandstructure(
50     configuration=bulk_configuration,
51     route=['G', 'M', 'K', 'G'],
52     points_per_segment=20,
53     bands_above_fermi_level=All
54 )
55 nlsave('graphene_band.nc', bandstructure)

```


Run the calculation, by sending the script from the Editor to the **Job Manager**. When the calculation finishes, reselect the NetCDF file to access the new band structure. This time the K point is on the Fermi level, as can be seen from the bandstructure illustrated below. Note that the plot has been zoomed into the range [-10, 15] eV.



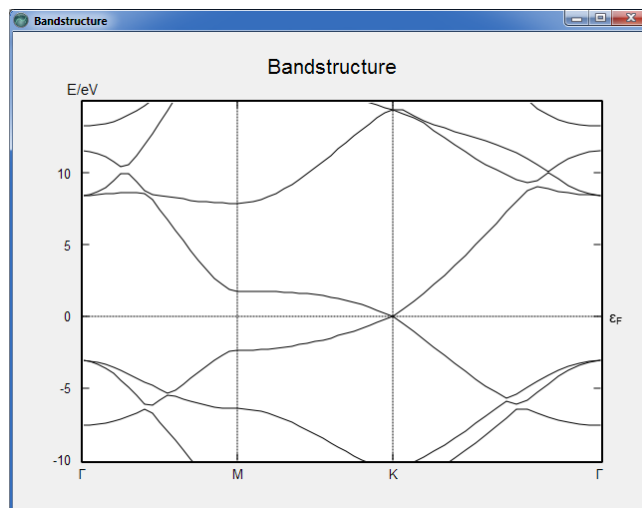
THE SAME THING, BUT WITH DFT

You now know the required k-point sampling, and you can use this also for the DFT calculation. To switch method, return to the **Script Generator** and select the **ATK-DFT** calculator. Set the k-point sampling under **Simulation parameters**, but leave all other parameters at their default.

Tip

In case you closed the Script Generator, you can reuse the geometry from the NetCDF file by dropping (one of) the **Bulk configuration** inside it on the **Scripter** icon  on the toolbar.

Again keep the same name for the output NetCDF file, and run the calculation. On inspection, you will find that the results are very similar to the Hückel band structure.



Thus, the Hückel method has good accuracy for this system and will be used for the further examples, mainly to save time; the interested reader is encouraged to verify that the results obtained with DFT are very similar.



Note


As it turns out, getting an accurate alignment of the K point to the Fermi level in graphene is not just a simple matter of having a lot of k-points. The results for 12×12 or 13×13 points are for instance considerably worse than those for 10×10 . There is a general trend that more k-points are better, but on the other hand 3×3 points are better than almost all other values up to 20×20 , except for 16×16 and - perhaps somewhat surprisingly - 17×17 . This behavior can be understood by studying how the k-points in the Monkhorst-Pack grids are distributed around $K = (1/3, 1/3, 0)$; the ideal is to have several points close to K (like in the cases 16×16 and 17×17) or on K (3×3 , 9×9 and 15×15 , for instance) although this is less favorable than having two points close by.

CHAPTER 3. BAND STRUCTURE OF AN ARMCHAIR RIBBON

RIBBON

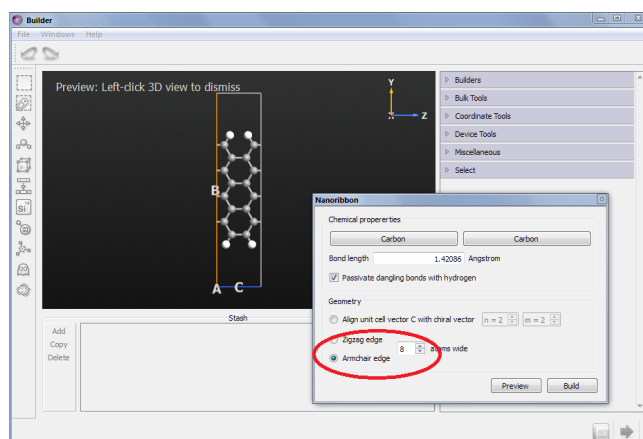
Having investigated the band structure of pure graphene, the focus is now on graphene nanoribbons (GNR), which are finite, narrow strips of graphene, cut out from the infinite 2D sheet. There are two primary ways to cut out such a ribbon, and these two structures are known as armchair or zigzag ribbons, a nomenclature that refers to the shape of the edges. Notably, an armchair ribbon is an unrolled zigzag nanotube! Armchair ribbons are predicted to be semi-conducting, and you will now compute the band structure of such a ribbon to confirm this.


BUILDING THE ARMCHAIR RIBBON STRUCTURE

Generating nanoribbons with VNL couldn't be easier – there is a ready template builder specifically designed for this task. To access it, launch the **Builder** by clicking its icon  on the main toolbar, and then click Add>Add Custom>Nanoribbon to the left of the **Stash** area.

In the widget that opens, you can design many different types of nanoribbons, not just graphene but also B-N, etc. By selecting a linear combination $n\mathbf{A}+m\mathbf{B}$ of the two unit cell vectors \mathbf{A} and \mathbf{B} in the graphene plane, ribbons of any chirality can be constructed (the "chirality" label comes from the fact that these structure correspond to an unrolled (n,m) nanotube). For simplicity, you can also directly choose to build armchair or zigzag ribbons, and specify their width (which must be even for zigzag ribbons).

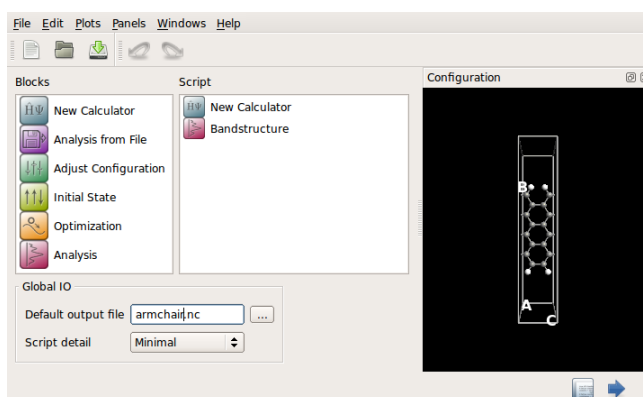
Select an armchair ribbon of width 8 (carbon atoms), and click **Build** to add it to the stash.



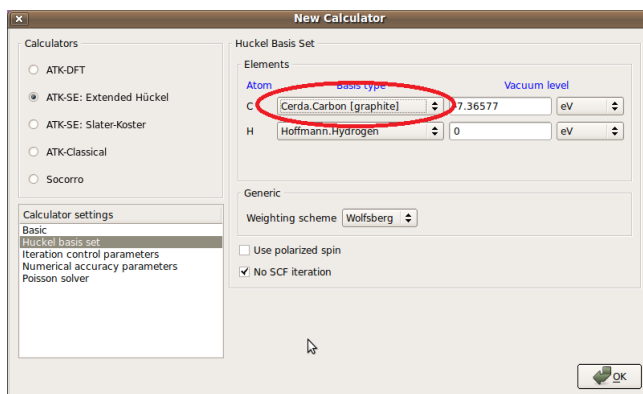
Send the structure to the **Script Generator** via the "Send To" icon  in the lower right-hand corner.

CALCULATING THE BAND STRUCTURE

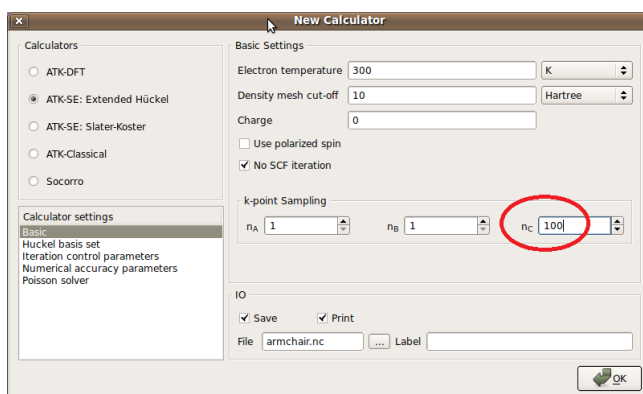
You will use the **Extended Hückel** method for the calculations. Don't forget to set an appropriate name for the NetCDF file, and make sure it is stored in the tutorial working directory!



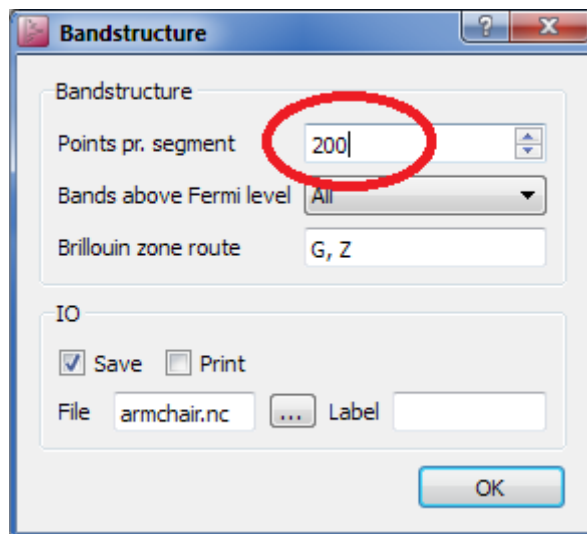
Again select the **Cerda graphite** basis set for carbon, while the default **Hoffmann** set appears to work fine for hydrogen.



In this case you need only 1x1 k-points in the directions perpendicular to the direction of the ribbon, since the system are finite in these direction. The system is periodic in the C direction,. A large number of points are needed in this direction, since the unit cell is quite short, and graphene has a rather peculiar point-like Fermi "surface" as we saw above; 100 points should be sufficient (the reader is encouraged to verify this!).

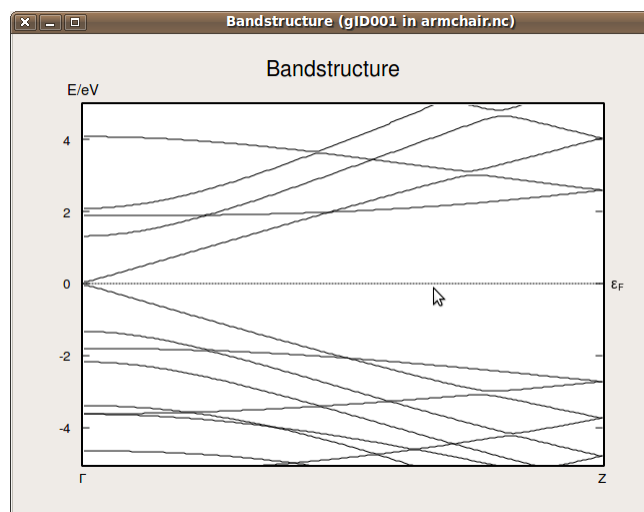


The default suggested route in the Brillouin zone from $G=(0,0,0)$ to $Z=(0,0,1/2)$ for the band structure is indeed the appropriate one, so you just need to increase the number of points to 200 to obtain smooth curves.



Run the calculation by sending the script to the **Job Manager**.

The computed band structure is shown in the figure below (zoomed in and with anti-aliasing turned on). By zooming in a bit more you can see that there is indeed a band gap at the Γ point.



As an extension of this exercise, you are recommended to redo the calculations for different types of ribbons (armchair/zigzag) and for different widths. The band gap is strongly dependent on both factors!

In the [next section](#), you will investigate a zigzag ribbon, which is metallic instead.

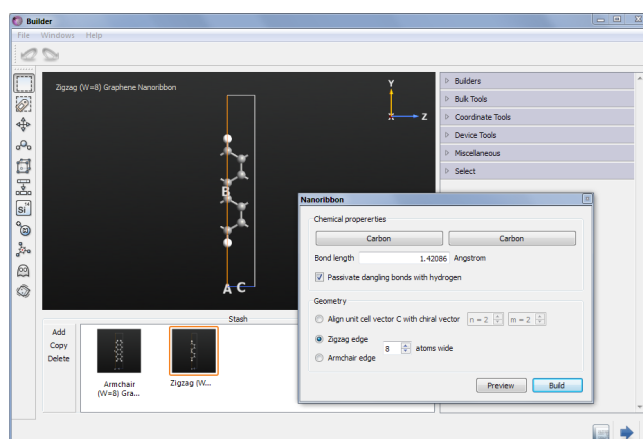
CHAPTER 4. TRANSPORT PROPERTIES OF A ZIGZAG NANORIBBON


In this chapter you will do some transport calculations. One of the most basic transport system that can be made with graphene is a perfect, infinite zigzag nanoribbon. Such ribbons are metallic and display no elastic scattering, i.e. perfect conductivity. In the second part you will investigate what happens when scattering is introduced by distorting the edges of the ribbon.




TRANSPORT PROPERTIES OF A PERFECT RIBBON

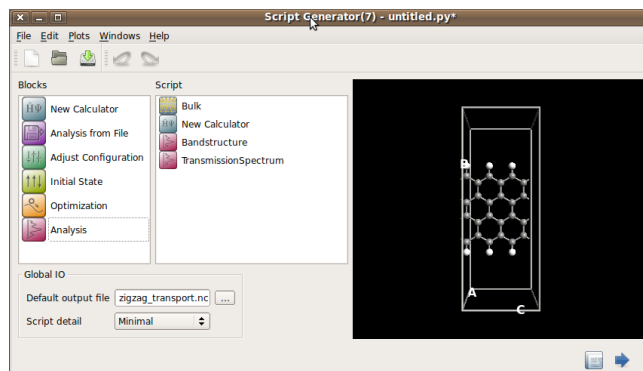
In a perfect device geometry there is no scattering from the central region and as illustrated in the following, you can obtain the transmission spectrum from the properties of the electrode.

Return to the Builder and again open the Nanoribbon widget from **Add>Add Custom>Nanoribbon**. Build a zigzag nanoribbon with width=8.



Use "Send To"  to transfer the structure to the **Script Generator**


- Set **Default output file** to zigzag_transport.nc
- Add a  **New Calculator** Block.
- Add a  **Bandstructure** Block
- Add a  **TransmissionSpectrum** Block

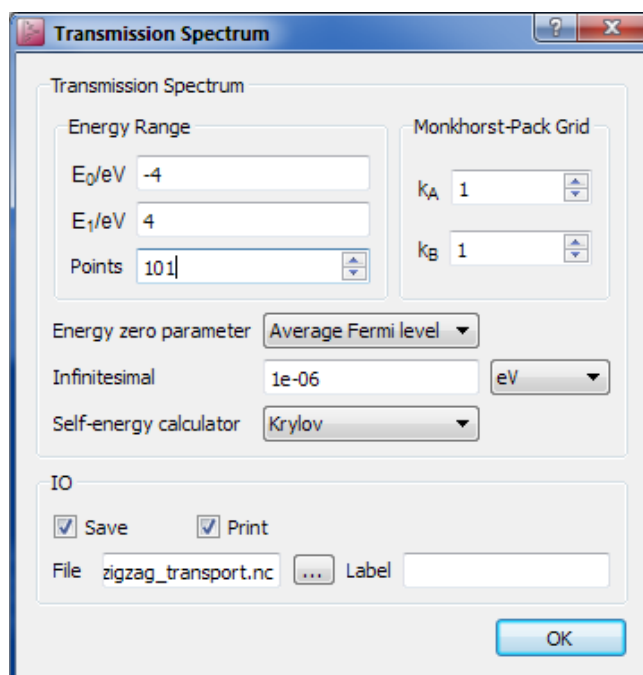


Open (double-click) the  **New calculator** block and make the same setting as in the previous chapter, i.e.

- Select the **Extended Hückel** calculator
- Set the k-point sampling to 1, 1, 100.
- Change the Hückel basis set to **Cerda.Carbon [graphite]** for C.


Next open the the  **Bandstructure** block and set points per segment to 200.

Finally, open the  **TransmissionSpectrum** block and set the energy range from -4 eV to $+4$ eV. Also, change the self-energy calculator method to **Krylov**. This method is considerably faster than the default Direct method, which however is a bit more robust. For the current system the Krylov method works fine, however.

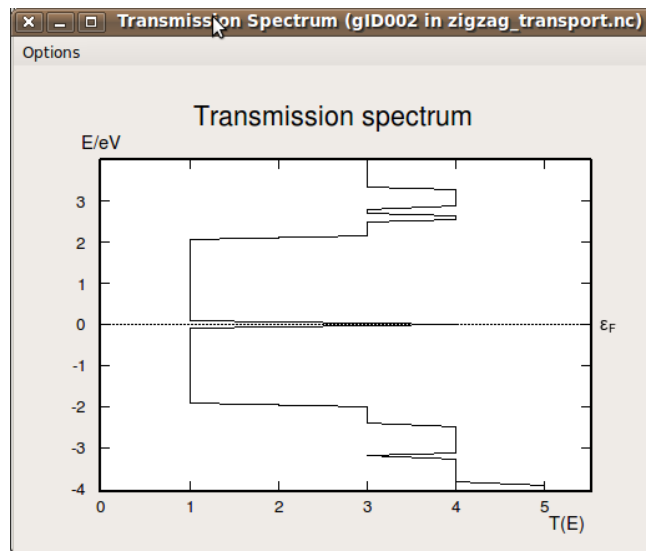


Note

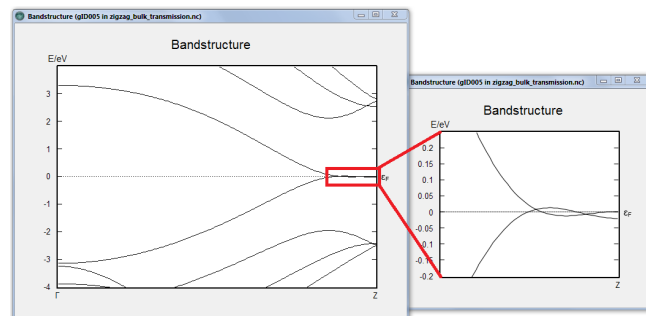
The odd number of energy points ensures that there will be an energy point at the Fermi level.

Send  the script to the **Job Manager** and **Start** the job. While it's a bit more time-consuming than the previous calculations, it should still only take a few minutes.

Once the calculation finishes, select the NetCDF file to view the transmission spectrum. As expected for a perfect 1D system, it exhibits a sequence of steps with integer transmission.



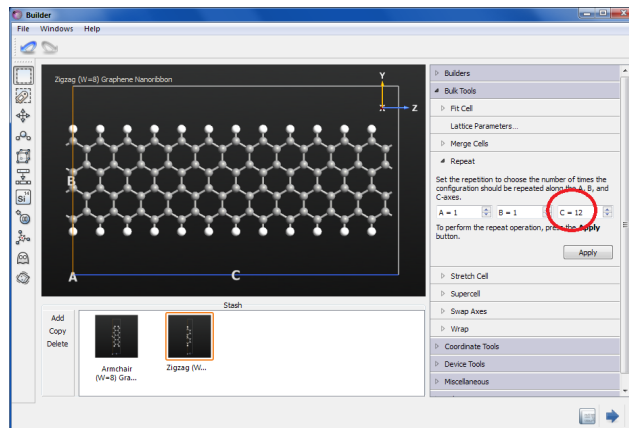
The most prominent feature of the plot is the enhanced transmission around the Fermi level. This is due to a peculiarity in the band structure of the zigzag ribbon (see bandstructure below), and interestingly the enhancement is retained even if the ribbon is distorted, at least in the way that will be investigated in the next example.



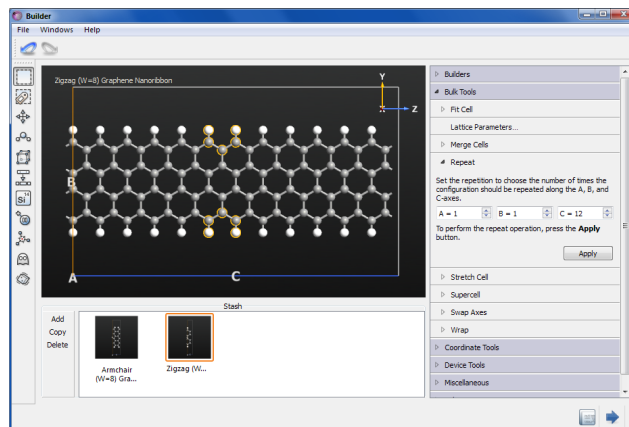
CREATING A DISTORTION IN THE RIBBON


So far you have basically just confirmed that armchair ribbons are semiconducting and zigzag ribbons are metallic (at least as long as spin is not considered). It is now time to do something a little bit more interesting. The next system you will study is a distorted ribbon, more specifically you will make a constriction in the perfect zigzag ribbon, and see how this influences the transport properties.

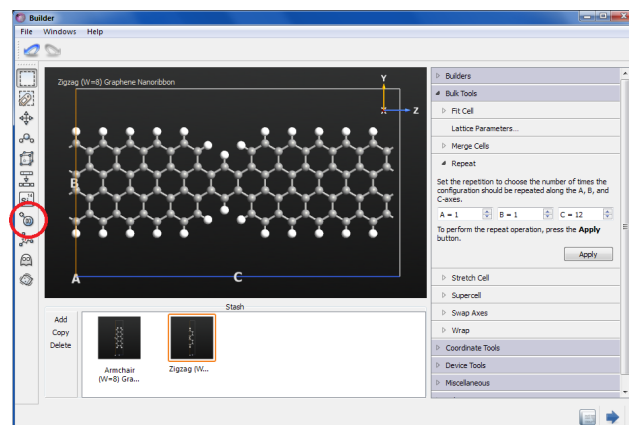
- Return to the Builder and open the **Repeat** tool (available from **Bulk Tools** in the list of plugins in the right-hand side panel). Set $C=12$ and press Apply, then press Ctrl+R to reset the 3D view.



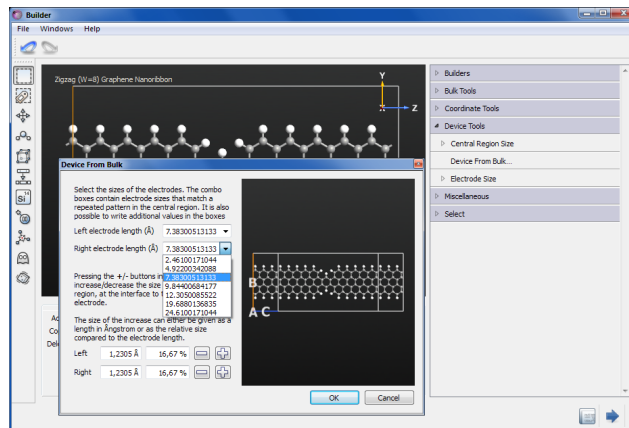
- Select the atoms indicated in the figure below by drawing a rectangle with the mouse, while holding down Ctrl.



- **Delete** the atoms by pressing the Delete key.
- The dangling bonds that have now been created should be saturated by adding hydrogen atoms. To do this, push the **Passivate** button  on the toolbar on the left-hand side.



- Finally, convert the periodic structure to a device geometry by using **Device From Bulk**, under **Device Tools**.



The builder will detect any pattern of periodicity close to the edges of the structure, and suggest these as electrodes. You may see these suggestions in the dropdown combo boxes, as indicated in the figure above.

For the current structure several periodicity patterns can be found, corresponding to a certain number of repetitions of the zigzag unit cell. The suggested value (7.383 Å), corresponds to 3 repetitions, and this is a suitable length, thus no further adjustments are needed. Click **OK**, and you now have the ready device geometry for the distorted zigzag nanoribbon.

CALCULATING THE PROPERTIES OF THE DISTORTED RIBBON



In the following you will learn how to perform the calculation of the zero bias conductance of the distorted ribbon at 3 different level of theory, non-self consistent, semi-self consistent using periodic boundary conditions, fully self-consistently using open boundary conditions.

Note

Calculations at finite bias must always be performed fully self-consistently using open boundary conditions, since the non-self-consistent models do not produce a correct voltage drop.

NON-SELF-CONSISTENT CALCULATION

Send the device geometry to the **Script Generator** using send to .


- Set **Default output file** to zigzag_transport.nc (i.e. the same name as used for the [bulk transmission calculation above](#), so that we can compare the results later on).
- Add  **New Calculator** Block.
- Add  **TransmissionSpectrum** Block

Open the  **TransmissionSpectrum** block.

- Set the energy range from -4 eV to +4 eV, and again select the faster **Krylov** method for the self-energy calculator.

Open the  **New calculator** block

- Select the **Extended Hückel (Device)** calculator
- Change the Hückel basis set to **Cerda.Carbon[graphite]** for C.

Send  the script to the **Job Manager** and **Start** the job. It takes a minute or so, but while it is running you can already proceed and prepare the [next step](#).

EQUIVALENT BULK CALCULATION

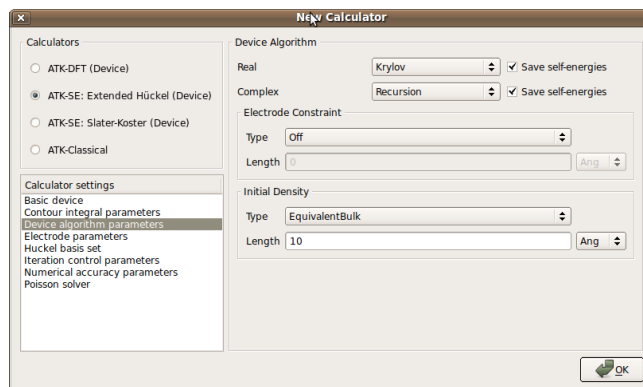
In the previous section both the electrode and central regions were treated non-self-consistently. Next, you will try a slightly more accurate method, where the electrodes are treated self-consistently and the central region is treated self-consistently using periodic boundary conditions in the transport direction. The latter is called an "equivalent bulk approximation".


To set up the calculation, you will modify the non-self-consistent calculation you have setup in the Script Generator.

Reopen the Script Generator (it is accessible from the Windows menu in all VNL instruments).

Open the  **New Calculator** Block

Select **Equivalent Bulk** for the initial density type in the Device algorithm parameters calculator settings.



Again send  the script to the **Job Manager** and **Start** the job. It takes a bit longer, perhaps 3-4 minutes.


OPEN BOUNDARY SELF-CONSISTENT CALCULATION

Finally, you will perform a fully self-consistent calculation with open boundary conditions.

Reopen the Script Generator.

Open the  **New Calculator** Block

Remove the tick mark from **No SCF iteration** in the Iteration control parameters calculator settings.

Send  the script to the **Job Manager** and **Start** the job. This time the job will take somewhat longer (perhaps 10 minutes), since a full self-consistent calculation with open boundary conditions will be performed.

COMPARING THE TRANSMISSION SPECTRA

To compare the transmission spectra for the perfect ribbon and the different level of theory for the distorted one, a simple "Custom Analyzer" is provided with this tutorial. It's very simple (see code below), and is designed to show all the transmission spectra in a NetCDF file in the same plot.

```
import numpy

def analyzer(filename, **args):
    if filename == None:
        return

    data_list = nread(filename, TransmissionSpectrum)

    plot = Plot2D()
    plot.setXLabel("Energy / eV")
    plot.setYLabel("Transmission")
    plot.setTitle("Transmission spectrum")

    # Loop through the data sets
    for data in data_list:
        t = data.evaluate()
        e = data.energies().inUnitsOf(eV)
        plot.addData([e,t])

    return plot


#-----

# Initialize builder
builder = Builder()
builder.title('Compare data')

# Set the configuration generator
builder.setAnalyzerGenerator(analyzer)

# Set up a Builder widget interface
builder.newGroup('Parameters')
builder.filename('filename', label='NetCDF file')

builder.plot('plot0', label='Data plot')
```

Download and save this script in the tutorial working directory, and then drop it on the **Analyzer** icon 

Then drop the NetCDF file with the ribbon transmission spectra onto the drop-zone **Drop file(s) here**.

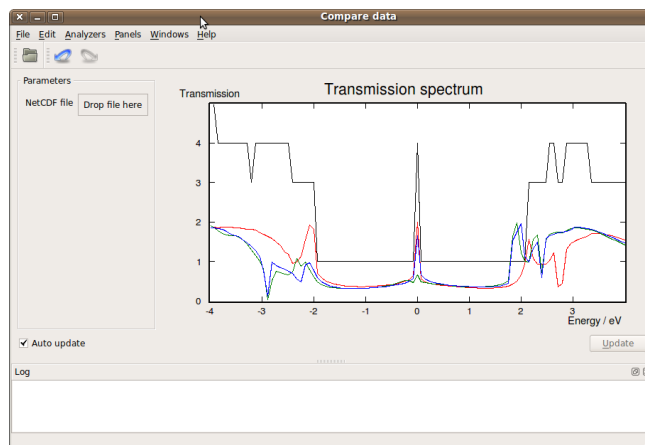


Figure 4.1: Comparison of the transmission coefficient of a perfect ribbon (black) and the distorted ribbon calculated non-self consistent (red), using equivalent bulk (green), and full self-consistent (blue).

Comparing the results you see that the strong transmission at the Fermi level is also present in the distorted ribbon, but otherwise the transmission spectrum is strongly suppressed due to scattering by the distortion.

It is also interesting to compare the different level of approximations for the transmission spectrum calculations. The qualitative trends are the same for the 3 models. In particular, away from the Fermi energy the equivalent bulk calculation (green curve) is very similar to the full self-consistent calculation (blue curve). The equivalent bulk calculation does not reproduce the kink at the Fermi energy, most likely because the Fermi-level band in the equivalent bulk calculation is slightly displaced and this result in additional scattering.

This concludes the tutorial. You have learned how to use the database and the built-in custom builders to set up simple calculations, i.e. band structures and transmission spectra, of graphene structures. You created a device configuration based on a distorted ribbon and tried different approximations for the calculation of the Transmission spectrum. You should now be familiar with the basic work flow related to creating geometries and setting up and running calculations.