

Bloch states in a graphene nanoribbon

Tutorial

Version 12.2.0

Bloch states in a graphene nanoribbon: 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
Prerequisites	1
2. Band structure of a zigzag nanoribbon	2
3. Bloch states	7
4. Introducing spin	11
5. Electron density and Mulliken populations	15
Bibliography	19

CHAPTER 1. INTRODUCTION

Graphene nanoribbons (GNRs) are very interesting systems for novel electronics applications. Depending on the shape of the ribbon edge, the ribbons can have metallic or semiconducting characteristics. As we will see in this tutorial (and as is well known from the literature [1]), spin also plays an important role for the transport properties of GNRs.

You will use the capabilities of Atomistix ToolKit to study the spin-dependent bandstructure of a zigzag GNR. These ribbons are metallic if the calculation is performed without accounting for spin, but when spin is included, a band gap opens up. By plotting conduction and valence band Bloch states for various k-points, You will see how the two spin-components are localized on opposite sides of the ribbon.


PREREQUISITES

A working installation of Virtual NanoLab is required, including a valid license for the software. A free demo license can be requested from our [web site](#).

This tutorial will involve calculations that take at most 5-10 minutes on a usual laptop, so there is no real need to use a separate, more powerful computer or run the jobs in parallel.

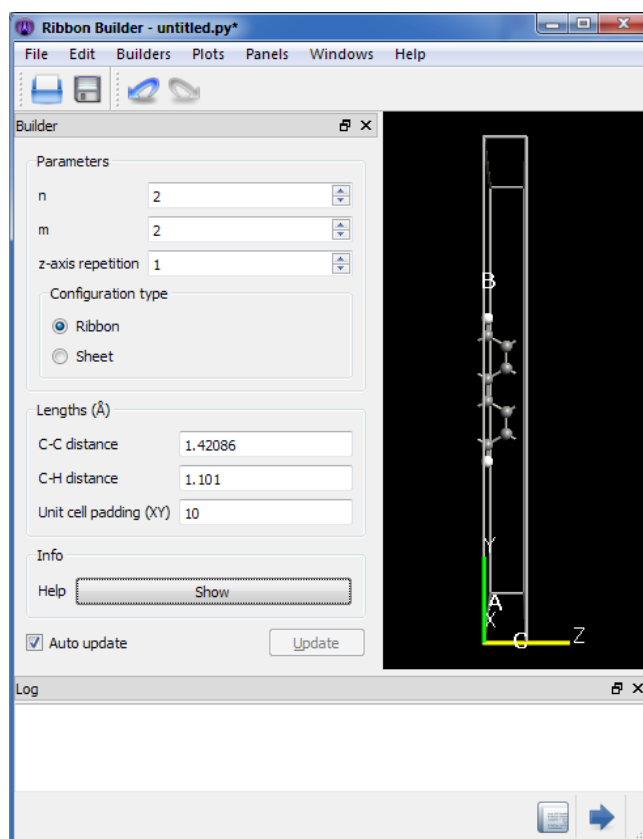
It will be assumed that you are familiar with the basic operations in the software corresponding to the level of experience gained from completing the [Virtual NanoLab Tutorial](#). No proficiency in programming will be necessary, although you will be editing a few lines in existing scripts.

CHAPTER 2. BAND STRUCTURE OF A ZIGZAG NANORIBBON

To define the geometry of the graphene ribbon, you will use the **Graphene Ribbon** builder in VNL. To do this, launch VNL and press the  icon on the VNL Toolbar. The Custom Builder tool then appears. From the menu bar, choose **Builders** → **Graphene Ribbon**. This launches the Graphene Ribbon builder tool.


Edit the following parameters to specify the ribbon;


- **n**: 2
- **m**: 2
- **z-axis repetition**: 1
- **Unit cell padding (XY)**: 10

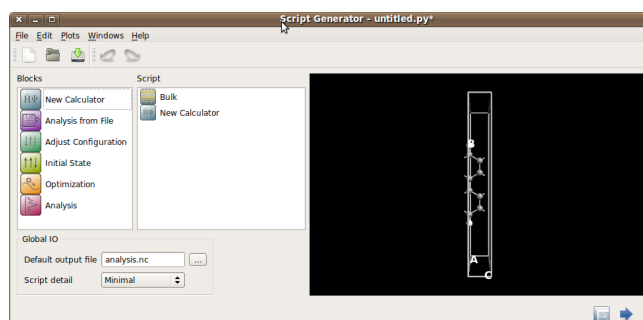


The repetition is set to the minimal value, this reduces calculation time and makes it easier to analyze the band structure.

There are periodic boundary conditions perpendicular to the ribbon, and the unit cell padding distance is set to 10 Å, to reduce any residual electrostatic interactions with the periodic images of the ribbon.

To set up a calculation of the band structure, press the  icon in the lower right corner of the Graphene Ribbon builder window and choose Script Generator. The **Script Generator** tool then appears.

To specify the calculator to be used, double-click the  icon. This adds a **New Calculator** block in the right panel of the Script Generator window.



Double-click the **New Calculator** block, and a dialog for specifying the calculator settings appears.

Most of the default parameters are good enough, but a few needs adjustment.

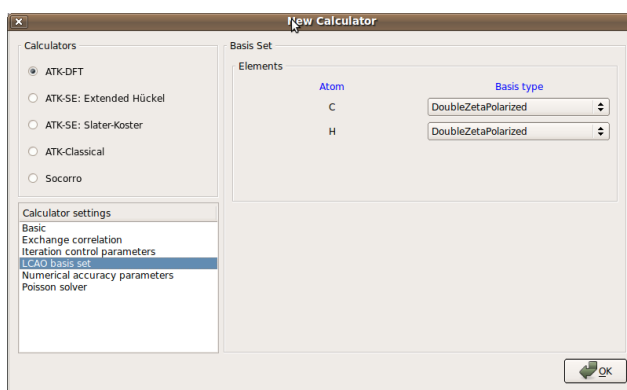
- You must carefully consider the k-point sampling. Since the unit cell along the ribbon is quite short, there must be a large number of k-points along the C-direction (along the ribbon).

Select 50 k-points in the C-direction.

This is a good trade-off between calculation time and accuracy. You only need a single point along the transverse A- and B-directions, since the system is not periodic in these directions.


- Next, under **Calculator settings**, choose **LCAO basis set** and reduce the basis set to **Single-ZetaPolarized** to gain some speed; for this particular system, this accuracy is sufficient.

The calculator dialog should now have the following settings



- Finally, click the **OK** button.

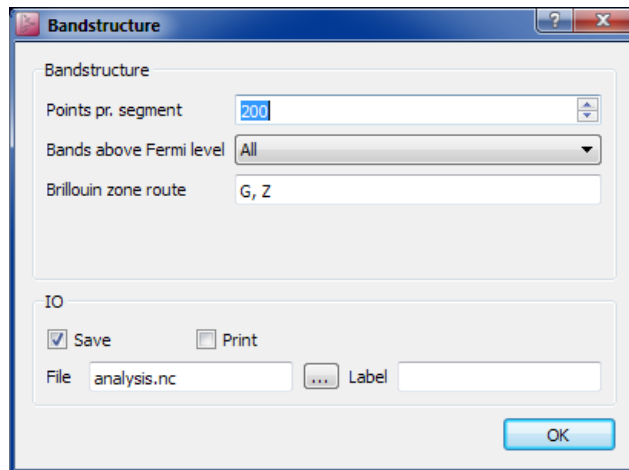
To specify the band structure calculation,

- double-click the  icon, and select **Bandstructure** from the menu.
- Then double-click the **Bandstructure** block.
- In the appearing dialog, 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 G (0,0,0) to Z (0,0,1/2)) is the appropriate one.



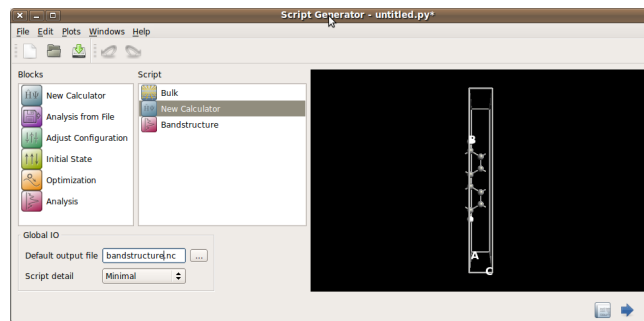
Note


The label "G" is used in ATK (and VNL) to designate the Γ point, when setting up the script. In the band structure plot, the proper Greek label will be shown (see below).



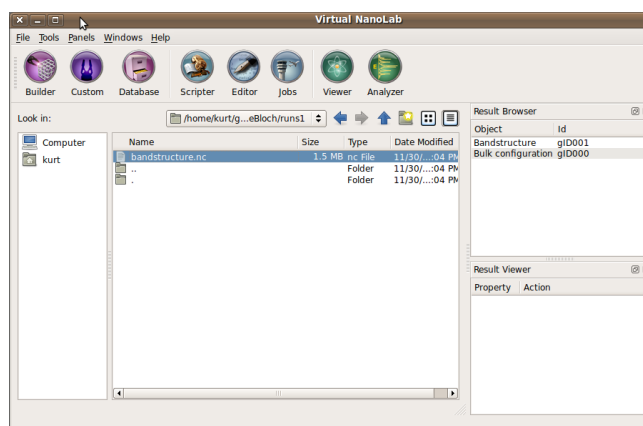
- Finally, click the **OK** button.

The final step is to set the name of the file where the output of the calculation is directed. To do this, change the **Default output file** from `analysis.nc` to `bandstructure.nc` in the **Script Generator** window. Later on, you will use this file for calculating the Bloch states associated with the bands of the bandstructure.



The entire calculation is now ready to be run. Perform the calculation by clicking the  icon in the lower right corner of the Script Generator window. From the menu, choose Job Manag-er. The **Job Manager** then appears. To start the job execution, press the **Process Queue** button. The job should take at most a few minutes.

Return to the main VNL window and select the file `bandstructure.nc`. The file contains two objects: the calculated band structure, and the **BulkConfiguration** itself. The latter object contains not only the geometry, but the full state of the calculation. We will use this in the [next chapter](#) to calculate the Bloch functions without having to rerun the self-consistent calculation. For this, you will need to know the **Id** of the configuration, which is listed in the Result Browser; it is **gID000** for our configuration.



Now, to plot the band structure, select the **Bandstructure** object in the Result Browser and then click the **Show** button in the Result Viewer panel. The resulting plot (zoomed in), is shown in the figure below. Note how the conduction and valence bands stick together for high values of k_z . This is a characteristic feature of zigzag ribbons (in our model) and clearly makes our ribbon metallic.

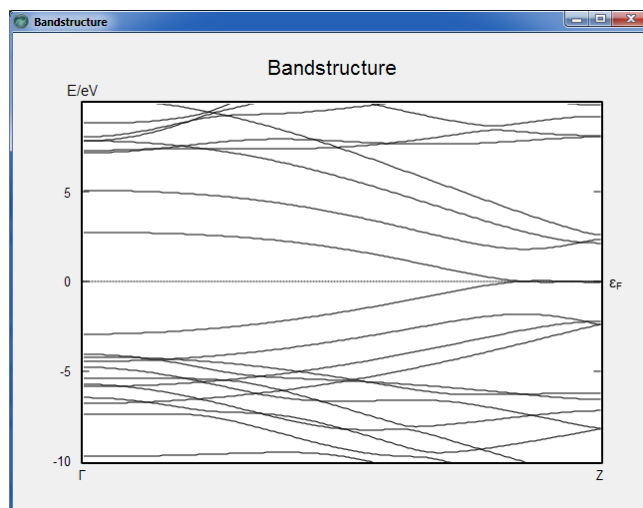




Figure 2.1: Band structure of a zigzag graphene nanoribbon, 8 atoms wide.

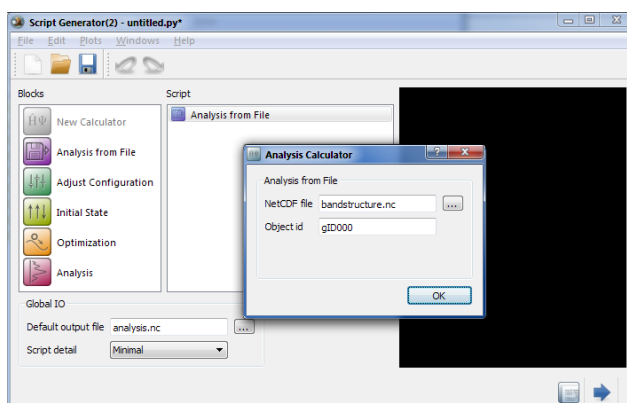
CHAPTER 3. BLOCH STATES

A very useful feature in ATK is its ability to calculate and plot Bloch states, which can be used to investigate the symmetry of certain bands and how this may relate to the transport properties. You will now use this capability to see how Bloch states close to the Γ point are delocalized across the ribbon, while states with higher k_z become increasingly localized towards the edges.

The ribbon consists of 8 carbon and 2 hydrogen atoms in each unit cell. A carbon atom contributes four valence electrons ($2s^2 2p^2$) and hydrogen one ($1s$), thus there are 34 electrons in the system. Each band is doubly degenerate (spin) and hence there will be 17 valence bands. Therefore, the band index of the highest valence band is 16 and the index of the lowest conduction band will be 17.


You will now set up a script that computes the Bloch functions of these two bands, in three different k -points. The self-consistent state of the ribbon was saved in the file `bandstructure.nc`, and you can just restore it for this analysis, instead of rerunning the self-consistent loop. This is a very convenient way to separate the actual self-consistent calculation, that you perhaps prefer to run on a cluster (at least for a bigger system), and the subsequent analysis that often requires less resources and can be run directly via the Job Manager in VNL.

- From the main VNL window, launch the **Script Generator** by clicking the  icon. Then double-click the  icon (it is the only available choice, when we start the Script Generator without dropping a configuration on it). The **Analysis from File** icon then appears in the middle panel of the Script Generator window.
- Double-click the icon and specify the file name `bandstructure.nc` in the appearing dialog.
- You should also specify the object ID of the configuration in this file you want to use for the analysis. As we found out in the [previous chapter](#), this is `gID000` for our case (it usually is, if the NetCDF file only contains one configuration).

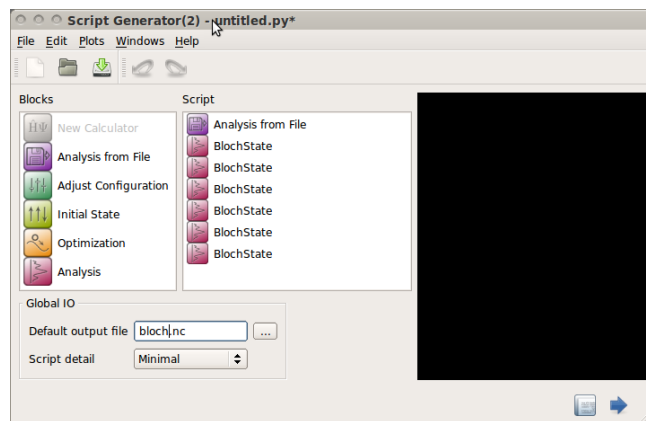


- Then click the **OK** button.

To set up the calculation of the Bloch states,

- double-click the  icon, and select **BlochState** from the menu. Do this until six Bloch state icons are present.
- Then in the **Script Generator** window, change **Default output file** from `analysis.nc` to `bloch.nc`.

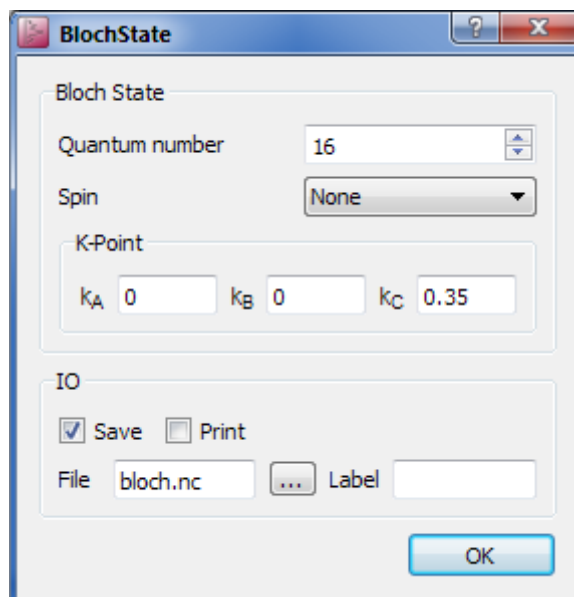
The Script Generator tool now looks as follows:



Double-click each of the first three Bloch state icons, enter 16 as the **Quantum number** (i.e. the band index), but give each one different k-points:

- (0,0,0)
- (0,0,0.35)
- (0,0,0.5)

In the second case, the dialog Bloch state dialog should look as



Repeat this procedure for the three remaining Bloch state icons, but this time enter 17 as the value for the **Quantum number**.

Finally, like before, send the job to the Job Manager and start the execution of the job.

 **Note**

The calculation is very fast, but the generated NetCDF file will be quite large (about 120 Mb), which is why the data is saved in a separate file, not to bloat the original file containing the self-consistent calculation (making it slower to read, if you desire to do some other analysis).

Once it finishes, analyze the NetCDF file from the VNL main window and plot each Bloch function as an isosurface.

To superimpose the geometry of the ribbon on the plot, drag the Bulk configuration from the file `bandstructure.nc` onto the open Viewer window.

To adjust the plot properties, right-click the Viewer and choose Properties... from the appearing context menu.

- Hide the **Axes** by removing the tick from Visible under the Axes entry
- Open **Bulk** Unit cell and hide the unit cell.
- Under **Isosurface**, set the **isovalue** to 0.1, the **Grid sampling** to 1, and select the HSV color map.

Below the visualization of the 6 different bloch states has been cut and pasted into the same figure.

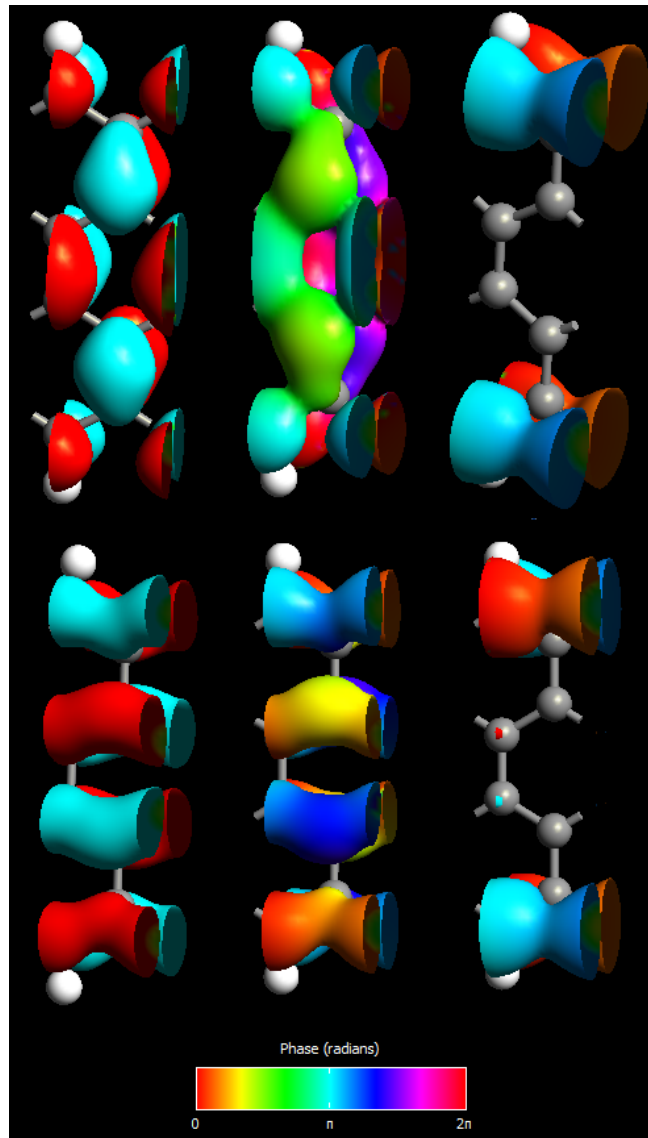


Figure 3.1: From left to right: The Γ point, $k=0.35$, and the Z point. The top and bottom rows display the conduction and valence band, respectively. (You may obtain a phase shift of π in your figures compared to this image; this has no physical relevance, of course.)

Looking at the respective Bloch functions, first note that the wave functions at Γ and Z are real (as expected), and second that there is a distinct difference between valence and conduction band Bloch functions. The occupied valence band states appear to be “connected” in the ribbon direction, while the (unoccupied) conduction band states are more oriented across the ribbon. At Z, the states become localized towards the edges of the ribbon.

In the next section, you will see how spin splits the degeneracy around Z, and how the two spin states become localized on opposite edges.

CHAPTER 4. INTRODUCING SPIN

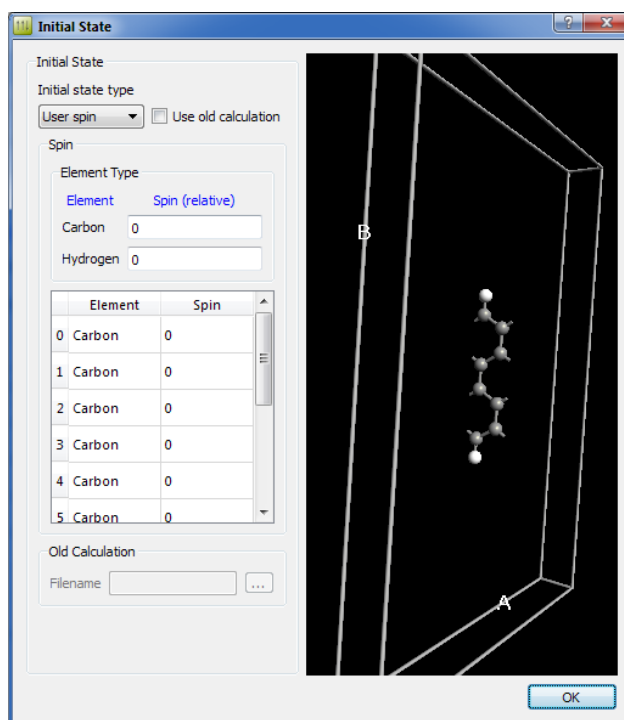
You will now redo the entire calculation with spin-polarization included. The work-flow is very similar, and each step is only outlined.

1. Repeat the steps from the [bandstructure](#) calculation including the step where you set up the **New Calculator**. Here you set the same parameters as before (SingleZetaPolarized basis set and (1,1,50) k-points).
2. In addition, in the **New Calculator** dialog, under **Basic** settings, choose the **LSDA** exchange-correlation functional.
3. The default in ATK is to start a spin-polarized calculation with a symmetric initial spin density. However, this will result in a local minimum, corresponding to a symmetric (ferromagnetic) state. To get the real anti-ferromagnetic ground state of the GNR we have to set up the initial spin configuration with opposite polarizations on the two nanoribbon edges (the middle carbon atoms and the hydrogen can be left “unpolarized”).

To set up the initial spin state of the system,

- double-click the **Initial State** icon .
- Then double-click the inserted **Initial State** block in the middle panel.
- Set the **Initial state type** to "User spin", and then under **Spin**, set the default spin of both carbon and hydrogen to 0.

The dialog should now look as follows



Then in the 3D view, select (by using Ctrl+left-mouse click) the two upper carbon atoms (number 0,7 in the list) and set their spin values in the table to 1.0. Repeat this procedure for the two lower carbon atoms (number 1,2 in the list) setting their initial values to -1.0. The hydrogen atoms and the four middle carbon atoms can be left with spin values equal to 0.0. The Initial State dialog now look as (don't be confused by the fact that the carbon atoms are not ordered by Y coordinate)

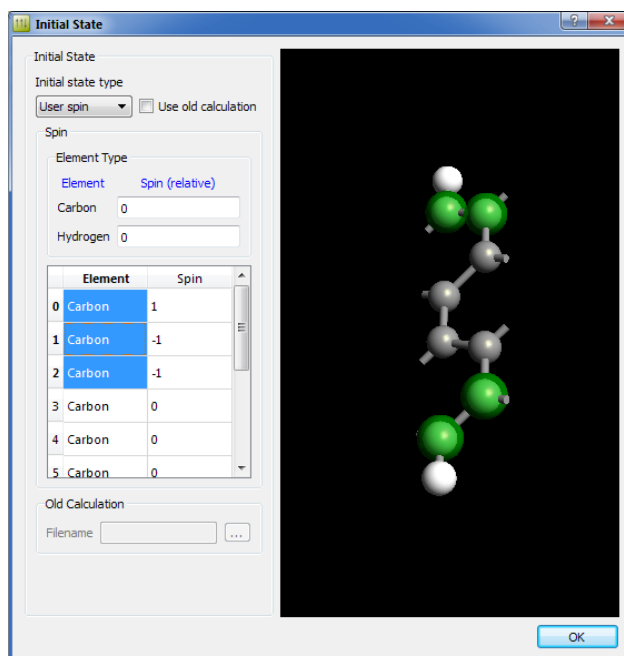


Figure 4.1: Setting the spin in the Initial State dialog. All carbon atoms with user-defined initial spin are selected.

Tip

If you have no idea about the spin state of the system, you may select Random spin. It sets a random spin on each atom, and dependent on the random values, the system may converge to different spin states of the system. The spin state with lowest total energy is the ground state.

4. Then set the name of the output file for the calculation in the **Script Generator** window, by changing the **Default output file** from `analysis.nc` to `spin.nc`.
5. Add a bandstructure calculation for the spin polarized calculation and again set **Points pr. segment** to 200.
6. You may this time just as well include the Bloch states from the beginning, to get all done in one script. For the following analysis, it is sufficient to only calculate the Bloch functions in the Z point. Add four Bloch state icons all defined at the Z point (0,0,1/2). Let two of these have **Quantum number** 16 with spin **Up** and **Down** respectively. Do the same thing for the remaining two, but this time for **Quantum number** 17.
7. Run the calculation!

By inspecting the calculation log you will see that there is no total spin-polarization in this system (the total spin up and down will both be 17), which is very reasonable; carbon is after all not magnetic.

You can also observe that there is no spin-dependence in the band structure; the up and down states are completely degenerate (otherwise you would see both blue and red curves in the band structure plot).

The inclusion of spin does however still have a profound effect on the bandstructure. The earlier degeneracy of the conduction and valence band close to $k_z=0.5$ is broken, and a band gap has opened up.

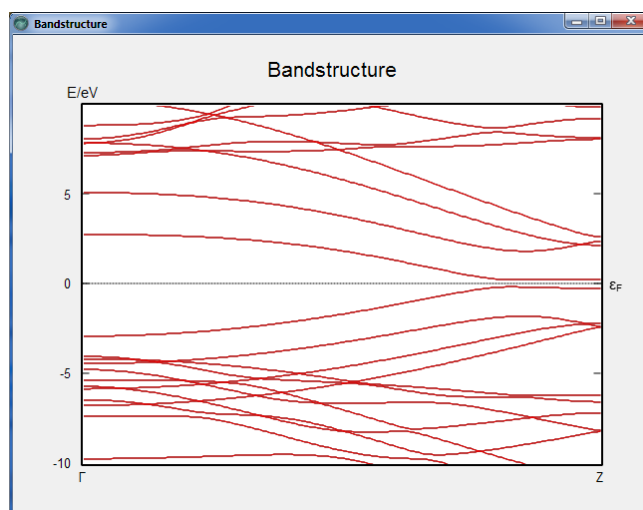


Figure 4.2: The smallest band gap occurs in the middle of the zone, as expected [1]. Actually, the band gap appears to be indirect, but the difference in k_z between the valence band maximum and the conduction band minimum is very small and may be a numerical artifact that would disappear with more accurate parameters.

If you plot the corresponding Bloch states at the Z point and adjust the view properties as in the previous chapter, you clearly see how the two spin valence states are localized on opposite

edges of the ribbon. The conduction band states behave completely analogously, but are localized on opposite edges compared to the valence band states.

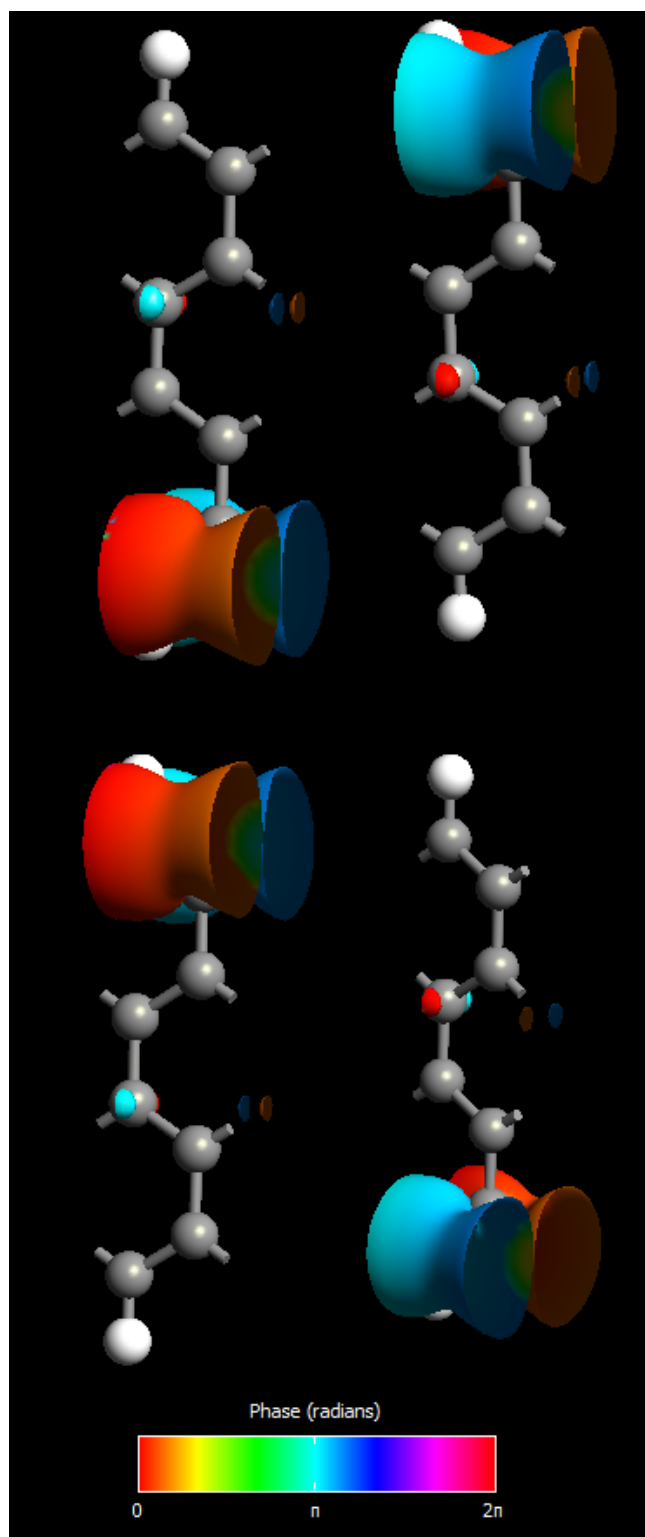



Figure 4.3: Spin up (top) and down (bottom) states at the Z point for the highest valence band (left) and lowest conduction band (right).


CHAPTER 5. ELECTRON DENSITY AND MULLIKEN POPULATIONS

Since the up and down Bloch states are localized on opposite edges, there must be a difference in the total spin up and down electron density.

You will therefore now compute and plot the spin-polarization, that is, the normalized difference between the spin up and down density:

$$r = (n_{\text{up}} - n_{\text{down}}) / (n_{\text{up}} + n_{\text{down}})$$

From the main VNL window, launch the Script Generator. Then double-click the Analysis from File icon . As before, open the **Analysis from File** dialog and specify the file name `spin.nc`.

To calculate the electron density, double-click the  icon, and select `ElectronDensity` from the menu. Do this twice; in the dialog boxes for each of these, set the **Spin** to **Up** and **Down**, respectively.

Also, add a **MullikenPopulation** block from the Analysis block. Then set the **Default output file** to `density.nc`.

Run the calculation; it will finish almost immediately.

To calculate the difference in the total spin up and down electron density, you need to write a small script. Open the Editor tool and insert the following lines of code

```
(n_u,n_d) = nload('density.nc', ElectronDensity)
r = (n_u - n_d)/(n_u + n_d)
nlsave('diff.nc', r, object_id="Spin Polarization Density")
```

As before, send the script to the Job Manager and run it. When it completes, go to the main VNL window and locate the created file `diff.nc`; plot the "Spin Polarization Density" contained in it.

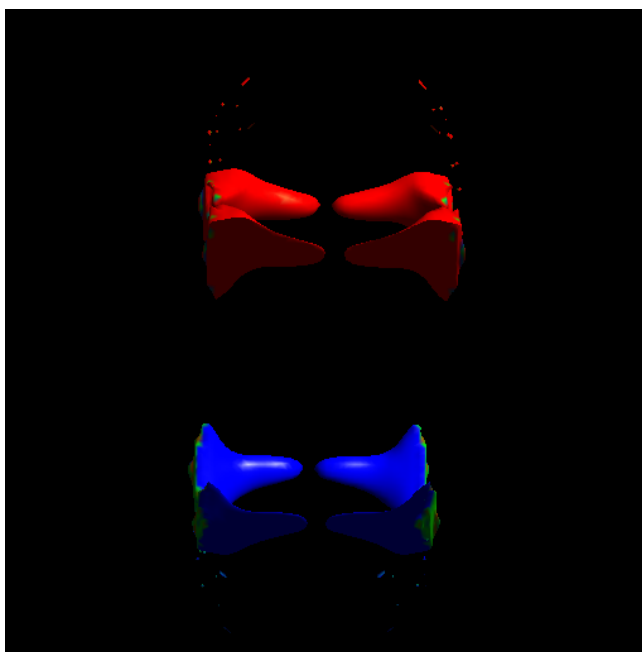


Figure 5.1: The spin polarization density, plotted as an isosurface plot. The two colors indicate the areas where there is a surplus of spin up/down density, respectively. The strongest polarization occurs around the two edge atoms, as expected [1].

The Mulliken populations is most conveniently viewed in the Log Window, where you find that the two edge carbon atoms (numbers 0 and 1) have a surplus population of 0.25 of either spin-up or spin-down.

You can also average the spin polarization along the x-axis (perpendicular to the graphene sheet). The following script will generate the plot below (cf. Fig. 2b in [2]), using the **matplotlib** module which is integrated in ATK.

Drop the script on the Job Manager, and it will generate the plot. The plot will be saved in the file `av.png`.

```
import pylab

# Read Y and Z coordinates
coords = nload('bandstructure.nc', BulkConfiguration)[0]
coords = coords.cartesianCoordinates().inUnitsOf(Angstrom)[: ,1:3]

# Calculate the polarization density from the up/down densities
(n_u, n_d) = nload('density.nc', ElectronDensity)
n_up = n_u[:, :, :].inUnitsOf(n_u.unit())
n_dn = n_d[:, :, :].inUnitsOf(n_d.unit())
n = (n_up - n_dn)/(n_up + n_dn + 1e-19)
av = numpy.array(n[:, :, :].sum(axis=0))

# Set the horizontal/vertical axes by scaling the data indices with the unit cell
y = numpy.array(range(av.shape[0]))/float(av.shape[0])*n_u.unitCell()[1][1].inUnitsOf(Ang)
z = numpy.array(range(av.shape[1]))/float(av.shape[1])*n_u.unitCell()[2][2].inUnitsOf(Ang)

# A 'spin-dependent colorbar'
cdict = {
    'red': ((0.0, 0.0, 0.0),
            (0.5, 0.0, 1.0),
            (1.0, 1.0, 1.0)),
    'green': ((0.0, 1.0, 1.0),
             (0.5, 0.0, 0.0),
             (1.0, 1.0, 1.0)),
```

```

    'blue': ((0.0, 0.0, 0.0),
            (0.5, 1.0, 0.0),
            (1.0, 0.0, 0.0))
    }
spin_cmap = pylab.matplotlib.colors.LinearSegmentedColormap('my_colormap',cdict,256)

# Build up the plot
pylab.figure(figsize=(5,y[-1]/z[-1]))
pylab.xlabel('z / Angstrom')
pylab.ylabel('y / Angstrom')
pylab.contourf(z,y,av,40,colors='k')
pylab.contourf(z,y,av,40,cmap=spin_cmap)
pylab.plot(coords[:,1],coords[:,0], 'ko',ms=15.0)
axis = pylab.axis('image')
# Limit the Z axis to the ribbon (default is the entire unit cell)
# The H-atom coordinates are z=10 and 19.3 Angstrom
v = [axis[0],axis[1],9.0,20.3]
pylab.axis(v)
pylab.colorbar()
pylab.savefig('av.png')
pylab.show()

```

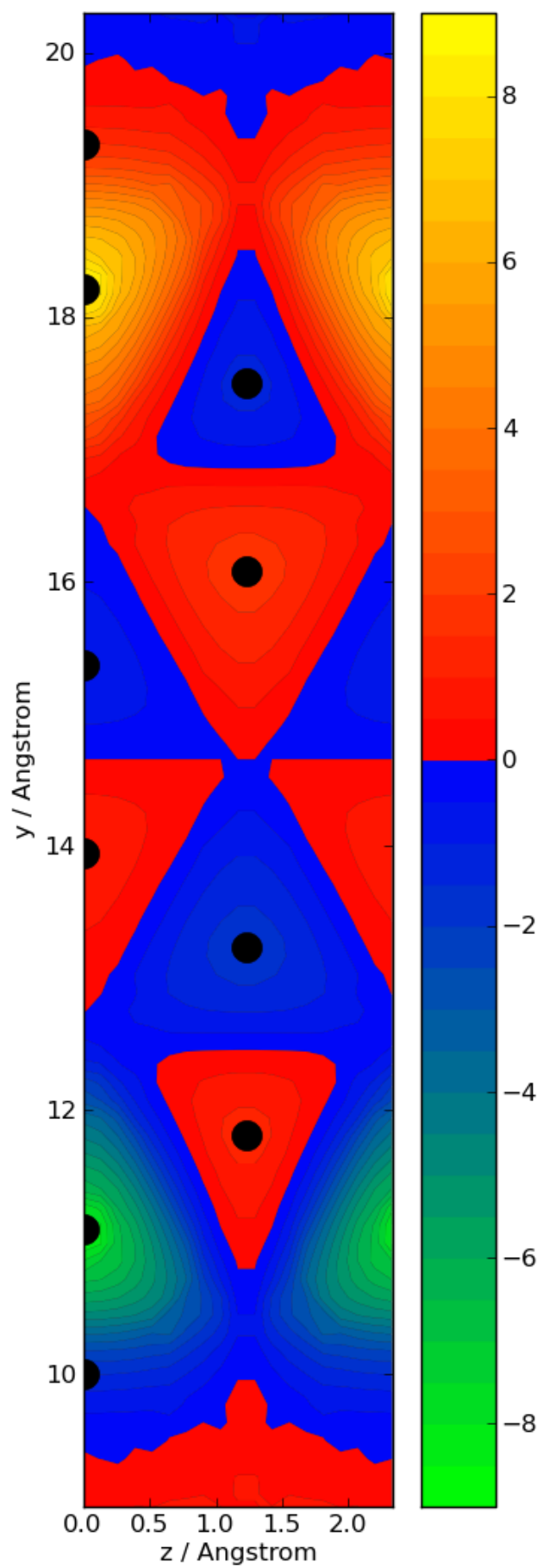


Figure 5.2: Averaged spin polarization density. The filled circles indicate the atom positions.

BIBLIOGRAPHY

- [1] Y.W. Son, M.L. Cohen, and S.G. Louie, *Phys. Rev. Lett.*, **97**, 216803, 2006.
- [2] Y.W. Son, M.L. Cohen, S.G. Louie, *Nature*, **444**, 347, 2006.