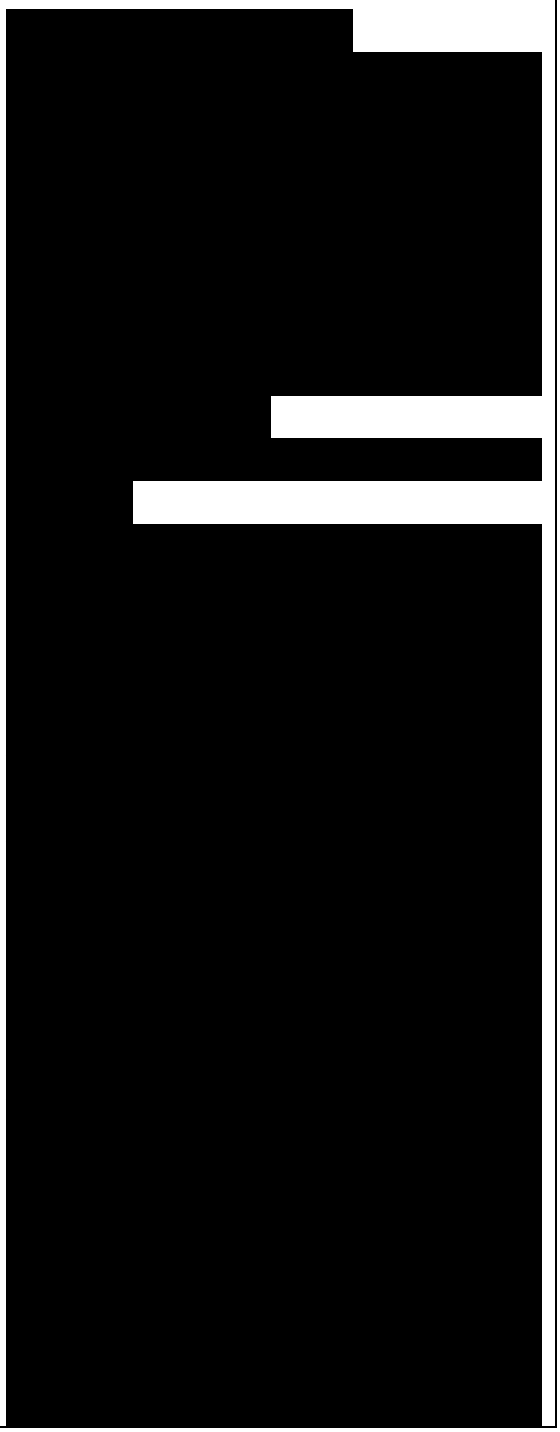
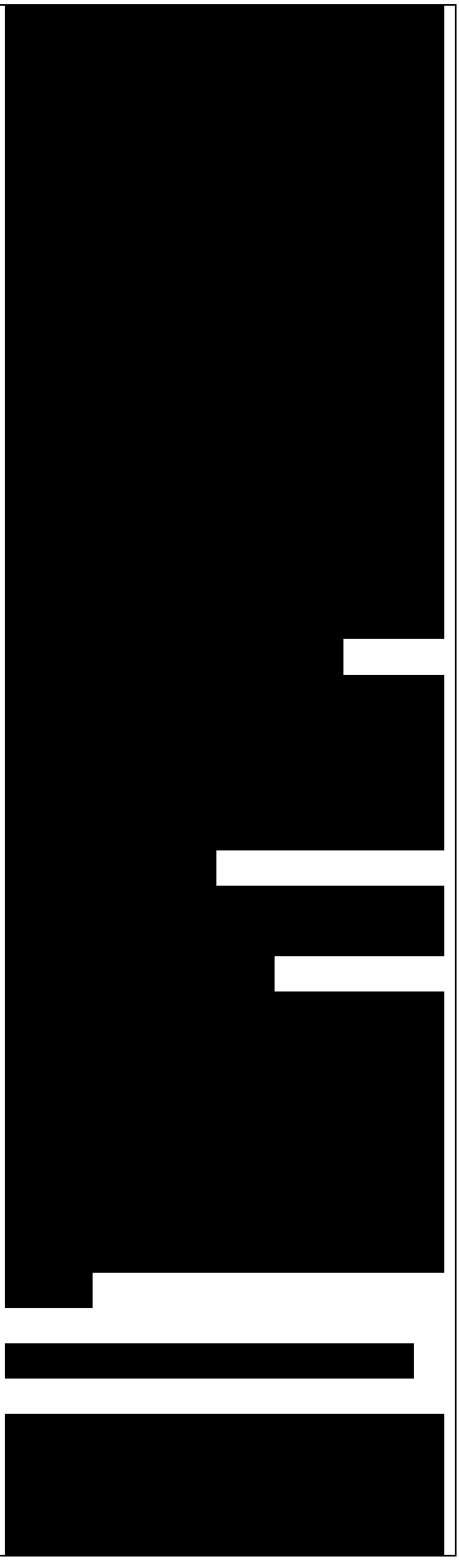
This tutorial was given in October 2014 at the CECAM Workshop on High Performance Models for Charge Transport in Large Scale Materials Systems at the Bremen Center for Computational Materials Science in Bremen, Germany. It demonstrates some of the possible uses of the DFTB+ code for calculating geometries, electronic structures and electron transport properties of	Loạt bài hướng dẫn này được trình bày vào Tháng Mười 2014 tại Hội thảo CECAM về Các Mô Hình Vận Chuyển Điện Tích Hiệu Quả Cao trong Các Hệ Vật Liệu Quy Mô Lớn tại Trung Tâm Khoa Học Vật Liệu Bremen ở Bremen, Đức. Loạt bài này đề cập đến việc dùng code DFTB+ để tính toán dạng hình học, cấu trúc điện tử và tính chất vận chuyển electron của các vật liệu

various 2D carbon materials. It is aimed at Master and starting PhD-students with background knowledge of the theory of electronic structure and electron transport. Minimal knowledge of Unix is also required. Knowledge of the DFTB+ code itself is not necessary, but familiarity with the basic ideas behind the Density Functional Tight Binding (DFTB) method could be helpful. 2.1 Perfect graphene 2.1.1 Geometry, density of state: Graphene has a hexagonal lattice with two C atoms in its primitive unit cell, which is specified in the supplied GEN-formatted geometry file. Open the file geo.gen in a text editor leafpad geo.gen &. You should see the following content: The format of this GEN file is the following: The first line contains the number of atoms (2) and the boundary condition type (S for solid). The second line lists all atomic elements present in the system separated by white space (C only in this example). Then a squence of lines follow, one for every atom in the system, each starting with a dummy integer (its sequential number in the structure), the type of the atom according to the list of elements in the second line of the file (1 for carbon in this example), and finally the cartesian coordinates of the atom in angstroms. Since the structure is periodic, appropriate information for this boundary condition must be provided after the atomic coordinates. For a GEN file of type S, this is the cartesian coordinates of the origin followed by the 3 cartesian lattice	carbon 2D khác nhau. Đối tượng là Thạc Sĩ và các bạn mới học nghiên cứu sinh nhằm cung cấp những kiến thức nền tảng về cấu trúc điện tử và sự vận chuyển điện tử. Độc giả cũng cần có kiến thức tối thiểu về Unix. Kiến thức về mã DFTB+ không cần thiết, nhưng phải biết những ý tưởng cơ bản về phương pháp Liên Kết Chặt Hàm Mật Độ. 

vectors (one per line). DFTB+ uses three dimensional periodic boundary conditions. In order to separate the graphene sheets from each other and to prevent interaction between them, the third lattice vector, which is orthogonal to the plane of graphene, has been chosen to have a length of 50 angstroms. Before running the code, you should check, whether the specified unit cell, when repeated along the lattice vectors, indeed results in a proper graphene structure. To repeat the geometry along the first and second lattice vectors a few times (the repeatgen script), convert it to XYZ-format (the gen2xyz script) and visualize it:

You should then see a graphene sheet displayed, similar to Figure 4x4x1 graphene supercell (page 4). Figure 2.1: 4x4x1 graphene supercell. You should see the following options within it:
First we include the GEN-formatted geometry file, geo.gen, using the inclusion operator (<< <):
Then we specify the ConjugateGradient dri ver to optimize the geometry and a lso the lattice vectors. Since neither the angle between the lattice vectors nor their relative lengths should change during optimization, we carry out an isotropic lattice optimization:

Then the details of the DFTB hamiltonian follow:
Within this block, we first specify the location of the parametrization files (the Slater-Koster files) and provide additional information about the



highest angular momentum for each element (this information is not yet stored in the Slater-Koster-files):

Please note, that the highest angular momentum is not a free parameter to be changed, but it must corre-spond to the value given in the documentation section of the correspoding homonuclear Slater-Koster-files (e.g. see the C-C.skf file for carbon). We use the self-consistent charge approach (SCC-DFTB), enabling charge transfer between the atoms: SCC = Yes
As graphene is metallic we smear the filling function to achieve better SCC-convergence:

For the Brillouin-zone sampling we set our k-points according to the 48 x 48 x 1 Monkhorst-Pack sampling scheme. This contains those k-points which would be folded onto the k-point (0.5, 0.5, 0.0) of an enlarged supercell consisting of the primitive unit cell repeated by (48, 0, 0), (0, 48, 0) and (0, 0, 1). This can be easily specified with the SupercellFolding option, where one defines those supercell vectors followed by the target k-point. We also want to do some additional analysis by evaluating the contributions of the s- and p-shells to the density of states (DOS). Accordingly, we instruct DFTB+ in the Analysis block to calculate the contribution of all C atoms to the DOS in a shell-wise manner (s and p) and store the shell-contributions in files starting with a prefix of pdos.C:
Analysing results: The very first thing you should check is whether your

calculation has converged at all to a relaxed geometry. The last line of the output file contains the appropriate message:
Geometry converged. This means that the program stopped because the forces on the atoms which are allowed to move (all of them in this example) were less than a given tolerance (specified in the option MaxForceComponent, which defaults to 1e-4 atomic units) and not instead because the maximal number of geometry optimization steps have been executed (option MaxSteps, default 200).You should visualize the resulting structure using Jmol (or any other molecular visualization tool). You should probably repeat the geometry again to get a better idea how it looks like, as we did for the starting structure above. The distance between the C atoms should be very similar to those in the initial structure.In order to visualize the density of states and the partial density of states, you should convert the corresponding human readable files (with prefix .out) to XY-format data. Please note the flag -w, which is mandatory when converting partial density of states data for plotting. You can obtain more information about various flags for dp_dos by issuing:

You can visualize the DOS and the PDOS for the s- and p-shells of carbon in one picture using the plotxy tool, which is a simple command line wrapper around the matplotlib python library (issue the command plotxy -h

for help):
You can use also any other program (gnuplot, xmgrace) which can visualize XY-data. You should see something similar to Figure DOS and PDOS of graphene (page 7).The position of the Fermi level (at -4.67 eV) can be read out from the detailed.out file, either directly or by using an appropriate grep command:

Figure 2.2: DOS and PDOS of graphene. As expected for graphene, the DOS vanishes at the Fermi-level. Around the Fermi-level, all states are composed of the p-orbitals of the carbons, the s-orbitals only contribute to energeticaly much lower and much higher states. Also, one can observe the van-Hove-singularties. The wiggles at around 0 eV and at higher energy are artifacts. Using more k-points for the Brillouin-zone sampling or using a slightly wider broadening function in dpdos would smooth them out.

2.1.2 Band structure : Band structure calculations in DFTB always consist of two steps:
1.Calculating an accurate ground state charge density by using a high quality k-point sampling.
2.Determining the eigenvalues at the desired k-points of the band structure, using the density obtained in the previous step. The density is not changed during this step of the band structure calculation.
Step 1 you just have executed, so you can copy the final geometry and the data file containing the converged charges from that calculation into

your current working directory:
Have a look on the dftb_in.hsd file for the band structure calculation. It differs from the previous one only in a few aspects:
We use the end geometry of the previous calculation as geometry:

We need static calculation only (no atoms should be moved), therefore, no driver block has been specified.The k-points are specified along specific high symmetry lines of the Brillouin-zone (K-Gamma-M-K):

We initialize the calculation with the charges stored during the previous run:
We do not want to change the charges during the calculation, therefore, we set the maximum number of SCC cycles to one: Let's run the code and convert the band structure output to XY-format:
The dp_bands tool extracts the band structure from the file band.out and stores it in the file band_tot.dat. For spin polarized systems, the name of the output file would be different. Use:

to get help information about the arguments and the possible options for dp_bands.In order to investigate the band structure we first look up the position of the Fermi level in the previous calculation performed with the accurate k-sampling which yields -4.67 eV, and then visualize the band structure by invoking. This results in the band structure as shown in Figure Band structure ofgraphene (page 8).
Figure 2.3: Band structure of graphene. You can see the linear

dispersion relations around the point K in the Brillouin-zone (k-points 0 and 51 in our circuit) which is a very typical characteristic of graphene.

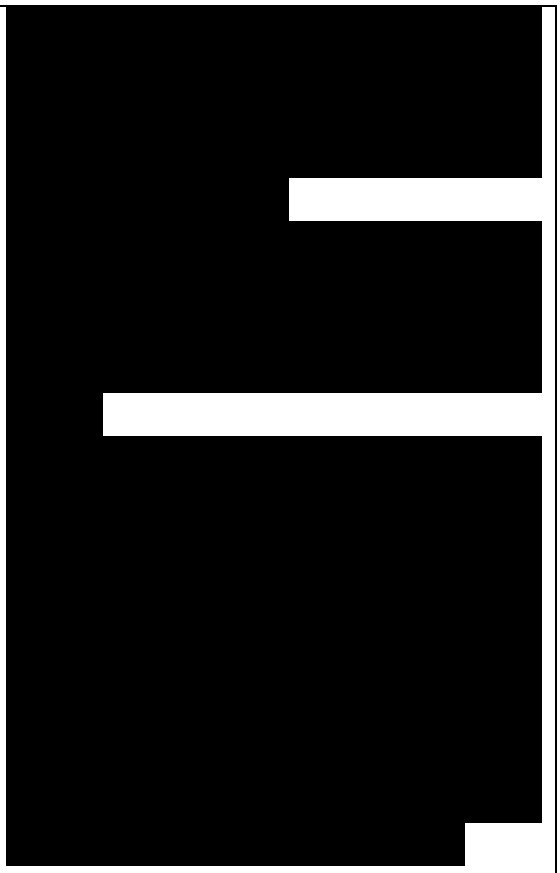
2.2 Zigzag nanoribbon: Next we will study some properties of a hydrogen saturated carbon zigzag nanoribbon.

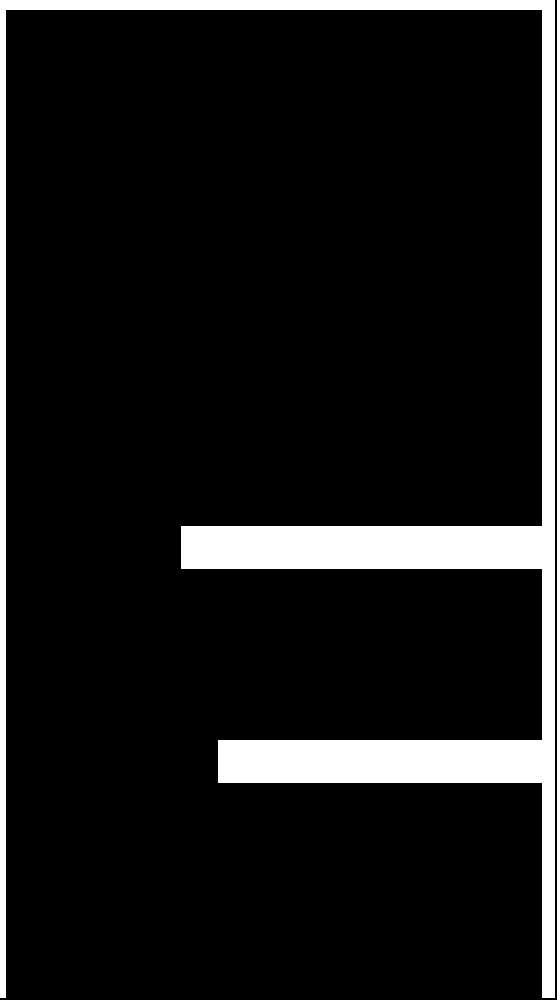
2.2.1 Calculting the density and DOS: The initial geometry for the zigzag nanoribbon contains one chain of the structure, repeated periodically along the z-direction. The lattice vectors orthogonal to the periodicity (along the x- and y- axis) are set to be long enough to avoid any interaction between the repeated images. First convert the GEN-file to XYZ-format and visualize it:

Similar to the case of perfect graphene, you should check first the initial geometry by repeating it along the periodic axis (the third lattice vector in this example) and visualize it. The necessary steps are collected in the file checkgeo.sh. Please have a look at its content to understand what will happen, and then issue to obtain the molecule shown in Figure Section of an H-saturated zigzag nanoribbon (page 9).

Figure 2.4: Section of an H-saturated zigzag nanoribbon. The control file dftb_in.hsd is similar to the previous examples, with a few differences only:

We use the 1 x 1 x 24 Monkhorst-Pack k-point set to sample the Brillouin-zone, since the ribbon is only periodic along the direction of the third lattice vector. The two other





lattice vectors have been choosen to be long enough to avoid interaction between the artificially repeated ribons.:

In order to analyze, which atoms contribute to the states around the Fermi-level, we create four projection regions containing the saturating H-atoms, the C atoms in the outermost layer of the ribbon, the C atoms in the second outermost layer and finally the C atoms in the thirds outermost layer, respectively. Since the ribbon is mirror symmetric, we include the corresponding atoms on both sides in each projection region:

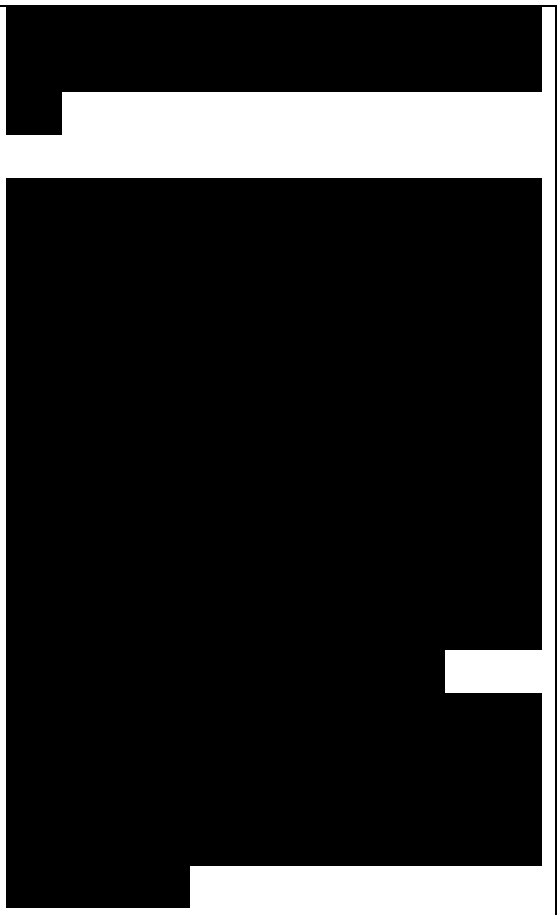
When the program has finished, look up the Fermi-level and visualize the DOS and PDOS contributions. The necessary commands are collected in showdos.sh:

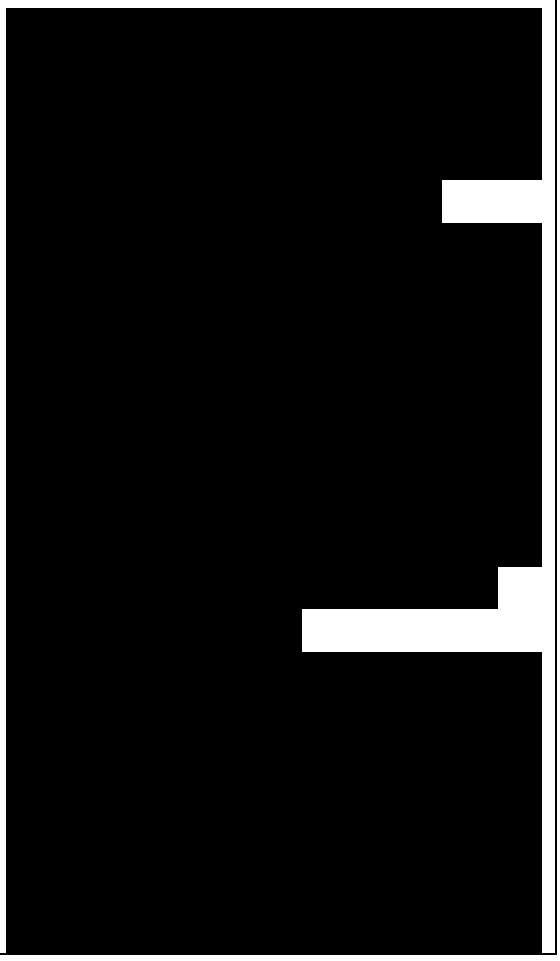
When you zoom into the area around the Fermi level (-4.57 eV), you should obtain something like Figure DOS of the zigzag nanoribbon around the Fermi energy (page 10).

Figure 2.5: DOS of the zigzag nanoribbon around the Fermi energy. states around the Fermi-level are composed of the orbitals of the C atoms in the outermost and the third outermost layer of the ribbon. There is no contribution from the C atom in the layer in between or from the H atoms to the Fermi level.

2.2.2 Band structure

Now let's calculate the band structure of the zigzag nanoribbon. The commands are in the script run.sh, so just issue: ./run.sh. You will see DFTB+ finishing with an error message ERROR! SCC is NOT converged, maximal SCC iterations

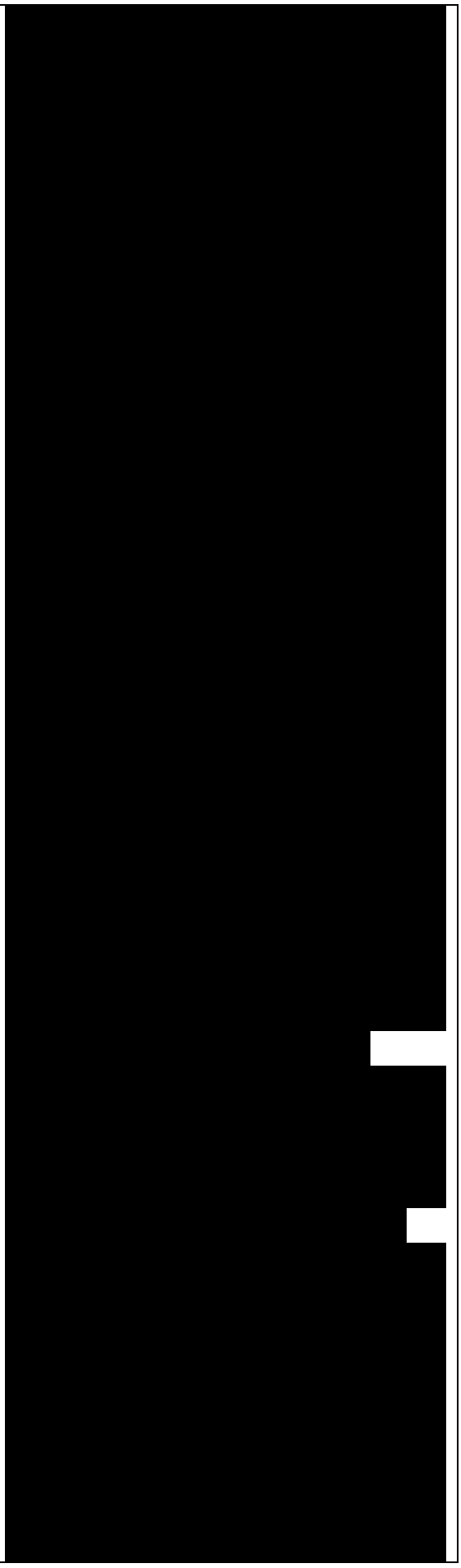




exceeded. Normally, it would mean that DFTB+ did not manage to find a self consistent charge distribution for its last geometry. In our case, however, it is not an error, but the desired behaviour. We have specified in dftb_in.hsd the options ReadInitialCharges = Yes MaxSCCIterations = 1 requiring the program to stop after one SCC iteration. The charges are at this point not self consistent with respect to the k-point set used for sampling the band structure calculation. However, k-points along high symmetry lines of the Brillouin-zone, as used to obtain the band structures, usually represent a poor sampling. Therefore the a converged density obtained with an accurate k-sampling should be used to obtain the eigenlevels, and no self consistency is needed. To look up the Fermi-level and plot the band structure use the commands in showbands.sh: /showbands.sh. You should obtain a band structure similar to Figure Band structure of the zigzag nanoribbon (page 11).

Figure 2.6: Band structure of the zigzag nanoribbon. Again, one can see, that there are states around the Fermi-energy, so the nanoribbon is metallic.

2.3 Armchair nanoribbon with defects: We now investigate a hydrogen saturated armchair carbon nanoribbon, examining both the perfect ribbon and two defective structures, each with a vacancy at a different position in the ribbon. In order to keep the tutorial short, we will not relax the vacancies, but will



only remove one atom from the perfect structure.

2.3.1 Perfect armchair nanoribbon Total energy and density of state: The steps to calculate the DOS of the perfect H-saturated armchair nanoribbon are the same as for the zigzag case. First check the geometry with the help of repeated supercells: ./checkgeo.sh. You will see a repeated image of the perfect armchair nanoribbon unit cell (Figure Perfect armchair nanoribbon unit cell (page 12)).Figure 2.7: Perfect armchair nanoribbon unit cell. The edge of the ribbon is visually different from the zigzag case. As it turns out, this also has some physical consequences. Let's calculate the electronic density and extract the density of states: ./run.sh. If you look up the calculated Fermi-level and then visualize the DOS. you can immediately see (Figure DOS of the perfect armchair nanoribbon (page 13)) that there are no states around the Fermi-energy (-4.4 eV), i.e. the investigated armchair nanoribbon is non-metallic.

Band structure.Let’s have a quick look at the band structure of the armchair H-saturated ribbon. The steps are the same as for the zigzag case, so just issue:

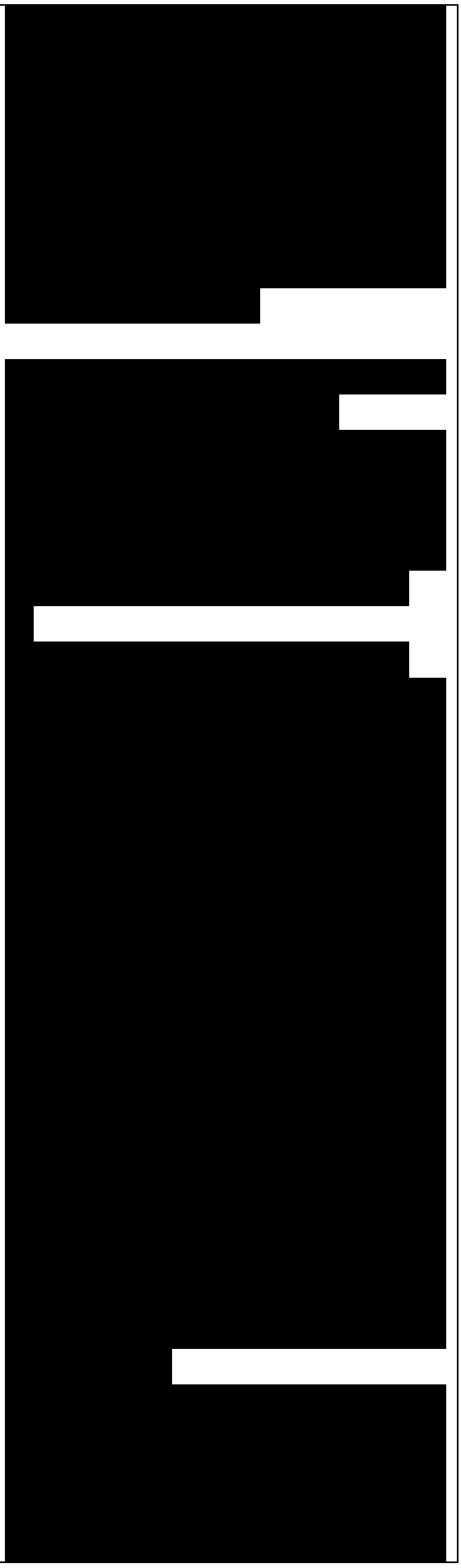
Figure 2.8: DOS of the perfect armchair nanoribbon. You should obtain a band structure like in Figure The band structure of the perfect hydrogen passivated armchair nanoribbon. The Fermi energy is at -4.4 eV (page 14). You can read off the position of the band edges, when

you zoom into the energy region around the gap: The valence band edge and the conduction band edge are in the Gamma point at -4.7 and -4.2 eV, respectively. You can also easily extract this information from the band.out file, when you look where to occupation goes from nearly 2.0 to nearly 0.0 in the first k-point (the Gamma point).

2.3.2 Armchair nanoribbon with vacancy Density and DOS

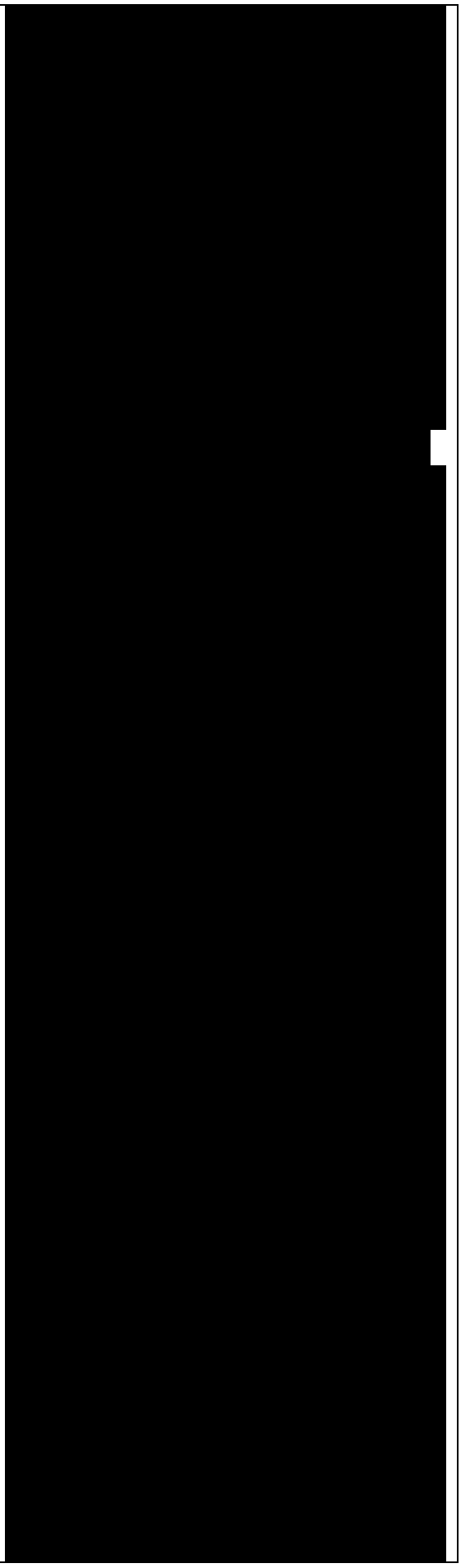
As next, we should investigate two armchair nanoribbons with a vacancy in each. The inputs can be found in the subdirectories elect/armchair/vacancy1_density and elect/armchair/vacancy2_density and you can visualize both with the command ./showgeom_v12.sh. As you can see on Figures Armchair nanoribbon with vacancy (structure 1) (page 14) and Armchair nanoribbon with vacancy (structure 2) (page 14), the vacancy is in the two cases on different sublattices. The two vacancies (structures 1 and 2) are located on different sublattices. Since the geometries are periodic along the z-direction, the defects are also repeated. As we would like to calculate a single vacancy, we have to make our unit cell for the defect calculation large enough to avoid significant defect-defect interactions. In this case, the defective cells contain twelve unit cells.In order to calculate the electron density of both vacancies, issue:

This will take slightly longer than the previous calculations, since each system contains more than four hundred atoms. Figure 2.9: The band structure of the perfect hydrogen



passivated armchair nanoribbon. The Fermi energy is at -4.4 eV. Figure 2.10: Armchair nanoribbon with vacancy (structure 1). Figure 2.11: Armchair nanoribbon with vacancy (structure 2). We want to analyse the density of states of the two different vacancies, together with that of the defect-free system. The commands necessary to extract the DOS of all three configurations and show them in one figure have been stored in the script showdos_perf_v12.sh. Execute it to obtain a figure like Figure The DOS of the perfect nanoribbon is indicated by solid blue line, the DOS of the nanoribbons with vacancies with green and red lines, respectively. (page 15).

Figure 2.12: The DOS of the perfect nanoribbon is indicated by solid blue line, the DOS of the nanoribbons with vacancies with green and red lines, respectively. As you can see, in contrast to the zigzag nanoribbon, the perfect armchair nanoribbon is insulating as it has no states around the Fermi-energy (-4.45 eV). The structures with vacancies, on the other hand, introduce dangling (unsaturated) bonds, leading to unoccupied states around the Fermi-energy. We can also see, that the defects affect the band edges, which are shifted with respect to their position in the perfect structure. It also seems that the valence band edge is more affected than the conduction band edge, and in the case of vacancy 2 (red line) the effect is significantly larger than for vacancy 1 (green line).



Vacancy formation energy: You should also be able to calculate the formation energies of the two vacancies. The formation energy £form of the vacancy in our case can be calculated as Eform =(Evac + EC) -12 x Eperf where £^ac is the total energy of the nanoribbon with the vacancy present, EC is the energy of a C-atom in its standard phase and £perf is the energy of the perfect nanoribbon. Since the defective nanoribbons contain 12 unit cell of the perfect one, the energy of the perfect ribbon unit cell has to be multiplied by twelve. As a standard phase of carbon, we will take perfect graphene for simplicity. The energy of the c-atom in its standard phase is then obtained by dividing the total energy of the perfect graphene primitive unit cell by two. (Look up this energy from detailed.out in the directory elect/graphene/density.) By calculating the appropriate quantities you should obtain ~8.5 eV for the formation energy of both vacancies. This is quite a high value, but you should recall that the vacancies have not been structurally optimised, and their formation energies are therefore, significantly higher than for the relaxed configurations.

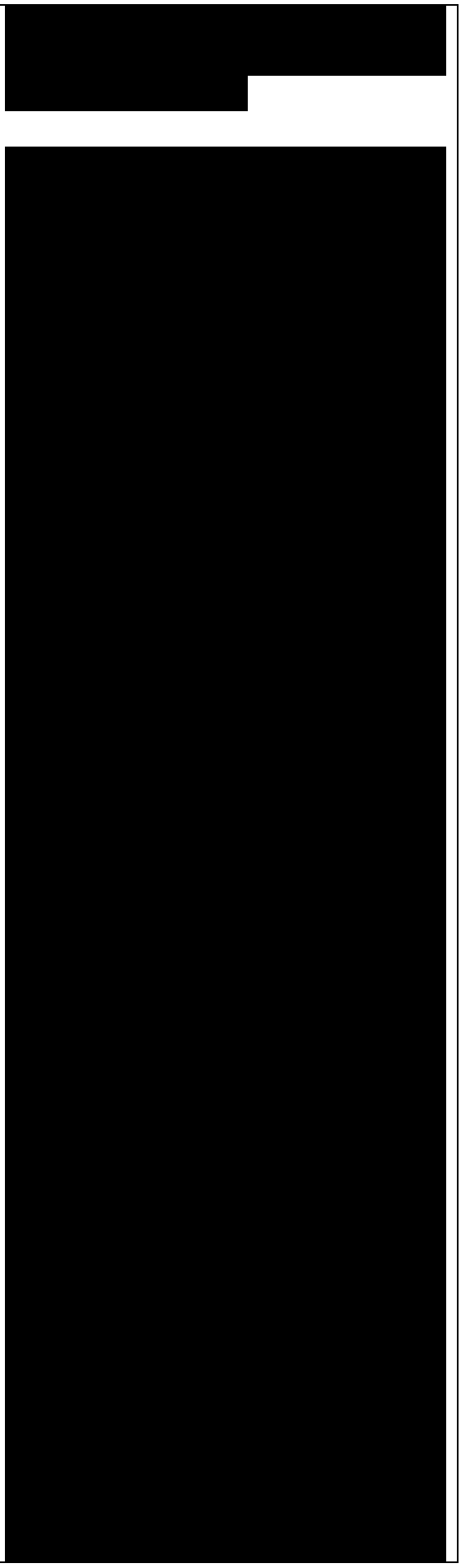
Defect levels: Finally we should identify the localised defect levels for vacancy 2 and plot the corresponding one-electron wave- functions. The vacancy was created by removing one C-atom, which had three first neighbors. Therefore, three sp2 type

dangling bonds remain in the lattice, which will then form some linear combinations to produce three defect levels, which may or may not be in the band gap. The DOS you have plotted before, indicates there are indeed defect levels in the gap, but due to the smearing it is hard to say how many they are. We want to investigate the defect levels at the Gamma point, as this is where the perfect nanoribbon has its band edges. We will therefore do a quick Gamma-point only calculation for vacancy structure 2 using the density we obtained before. We will set up the input to write out also the eigenvectors (and some additional information) so that we can plot the defect levels with waveplot later. This needs the following additional settings in dftb_in.hsd:

To just run the calculation ./run.sh and open the band.out file. You will see, that you have three levels (levels 742, 743 and 744 at energies of -4.51, -4.45 and -4.45 eV, respectively) which are between the energies of the band edge states of the perfect ribbon. We will visualize those three levels by using the waveplot tool. Waveplot reads the eigenvectors produced by DFTB+ and plots real space wave functions and densities. The input file waveplot_in.hsd can be used to control which levels and which region waveplot should visualize, and on what kind of grid. In the current example, we will project the real part

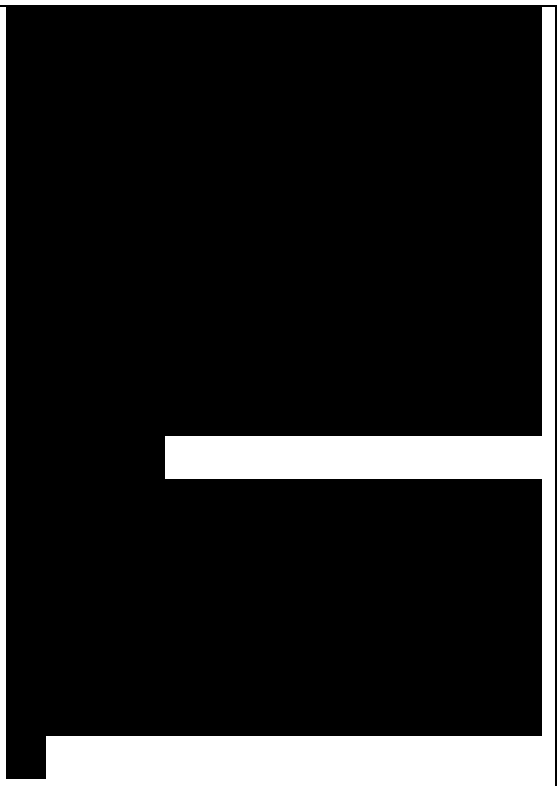
of the wave functions for the levels 742, 743 and 744. In order to run Waveplot, enter: waveplot | tee output.waveplot

The calculation could again take a few minutes. At the end, you should see three files with the .cube prefix, containing the volumetric information for the three selected one-electron wavefunctions. We will use Jmol to visualize the various wave function components. Unfortunately, the visualization of iso-surfaces in Jmol needs some scripting. You can find the necessary commands in the files show*.js. You can either type in these commands in the Jmol console (which should be opened via the menu File l Console...) or pass it to Jmol using the -s option at start-up. For the case latter you will find prepared command to visualize the various orbitals in the files. Looking at the defect levels, you can see that the defect level lowest in energy (742) has a significant contribution on the atoms around the defect, but also a non-negligible delocalized part smeared over almost all atoms in the system. Apparently a localized defect level has hybridized with the delocalized valence band edge state, resulting in a mixture between localized and non-localized state. The other two defect levels, on the other hand, have wavefunctions which are well localized on the atoms around the vacancy site. Note that in accordance with the overall symmetry of the system, the defect levels are either symmetric or antisymmetric with respect to the mirror plane in the middle of the ribbon. Figure 2.13: Wave function of the lowest defect

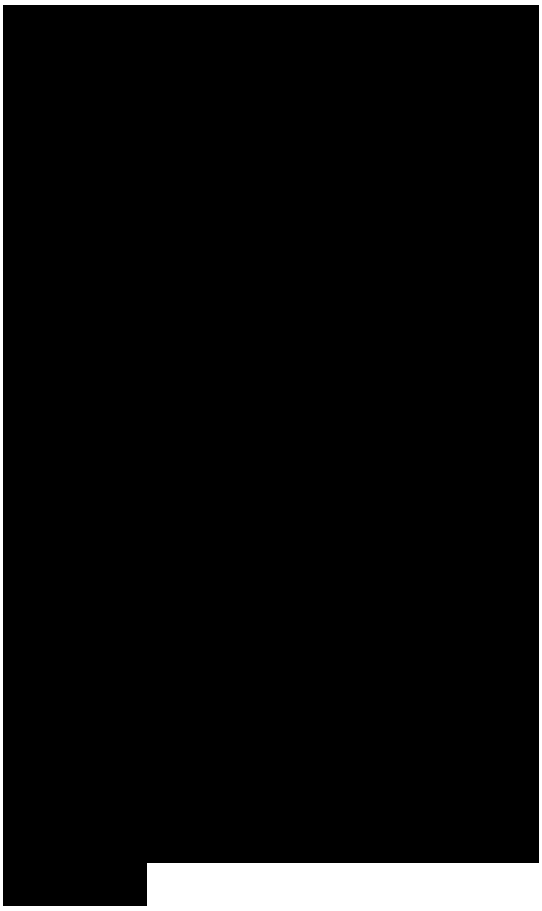


level of the hydrogen saturated armchair nanoribbon with a vacancy. Blue and red surfaces show indicate isosurfaces at +0.02 and -0.02 atomic units, respectively.Figure 2.14: Wave function of the second lowest defect level of the hydrogen saturated armchair nanoribbon with a vacancy. Blue and red surfaces show indicate isosurfaces at +0.02 and -0.02 atomic units, respectively.

Figure 2.15: Wave function of the highest defect level of the hydrogen saturated armchair nanoribbon with a vacancy. Blue and red surfaces show indicate isosurfaces at +0.02 and -0.02 atomic units, respectively.

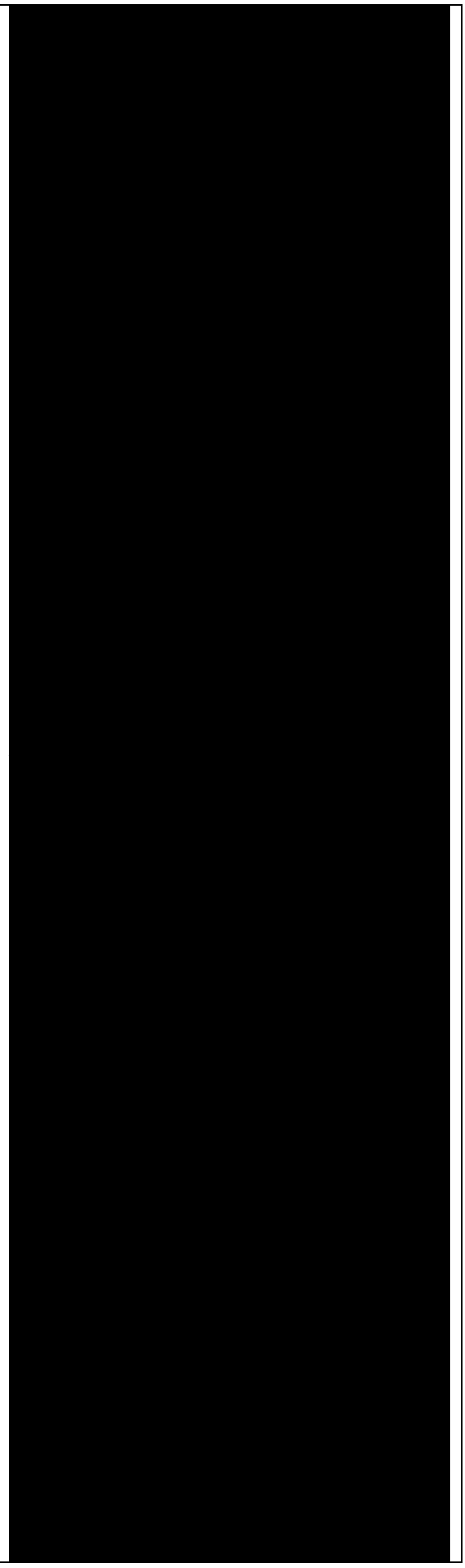


Electron transport calculations in armchair nanoribbons: In this sections of the tutorial we will learn how to set up self consistent and non self consistent simulations with open boundary conditions by using the mpi/negf version of dftb+. We will then calculate the density of states and transmission coefficients and then analyse the results in comparison with the previous periodic calculations. If you have not done so yet, please download the file tutorial_cecamhp.zip and decompress it by issuing:N unzip tutorial_cecamhp.zip in your HOME directory. Then enter the directory transport/. All directories given in this part of the tutorial are sub-directories of the transport/ directory.



3.1Non-SCC Pristine armchair nanoribbon. 3.1.1Preparing the

structure: When we run a transport calculation with open boundary conditions, the geometric structure specified in the input needs to obey some rules. The system must consist of an extended device (or molecule) region, and two or more semi-infinite bulk contacts. The bulk contacts are described by providing two principal layers for each contact. A Principal Layer (PL) is defined as a contiguous group of atoms that have finite interaction only with atoms belonging to adjacent PLs. In a sense, a PL is a generalisation of the idea of nearest neighbour atoms to the idea of nearest neighbour blocks. The PL partitioning in the electrodes is used by the code to retrieve a description of the bulk system. PLs may be defined, as we will see, in the extended device region to take advantage of the iterative Green's function solver algorithm. Additional information about the definition of PLs, contacts and extended device region can be found in the manual and in the on-line recipes. In the case of an ideal one-dimensional system, all the PLs are identical. The system we start with is an infinite armchair graphene nanoribbon (AGR), therefore the partitioning into device and contact regions is somewhat arbitrary. We will therefore start from a structure file containing a single PL (2cell_7.gen), which has been previously relaxed. The PL can be converted to XYZ format by using the gen2xyz script and visualised with Jmol. The structure is shown in Figure Armchair nanoribbon principal layer (PL) (page 20). As you may notice, we did not take a



single unit cell length as a PL, but rather two unit cells. This choice is dictated by the definition of the PL itself, as we want to avoid non-zero interactions between second-neighbour PLs. This is better explained by referring to Figure Layer definition (page 20). The red carbon atoms represent the closest atoms which would belong to non-nearest neighbour PLs, and these have a separation of 0.568 nm, as shown in Figure Layer definition (page 20). The carbon-carbon interaction is non zero up to a distance of 6 a.u., therefore the interaction between the two red atoms would be small, but non zero. Hence this is too small a separation for a one unit cell long section of nanoribbon to be used as the PL.

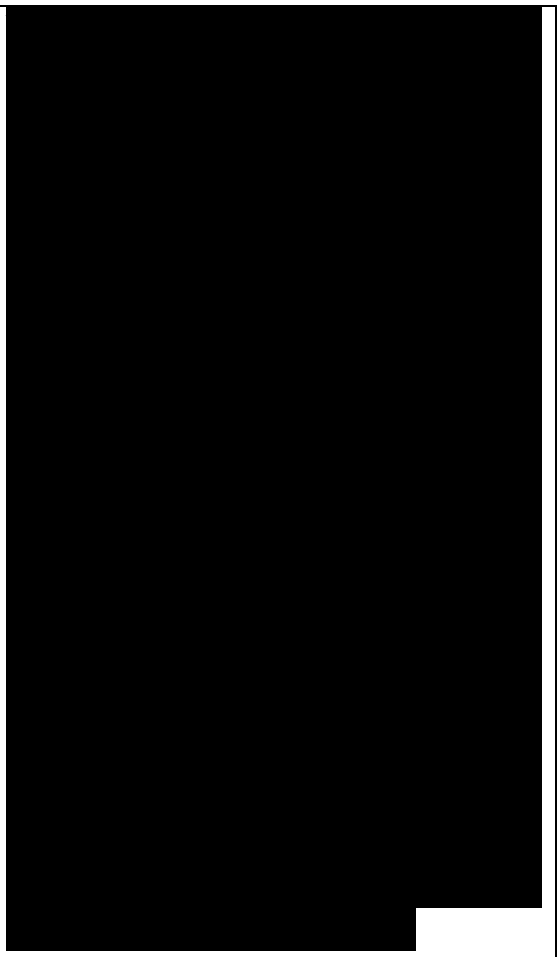


Figure 3.1: Armchair nanoribbon principal layer (PL)

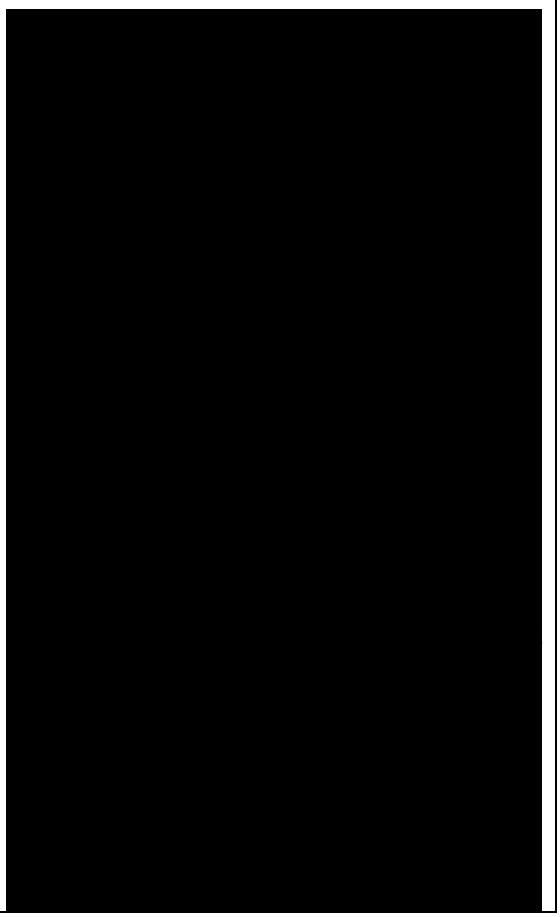
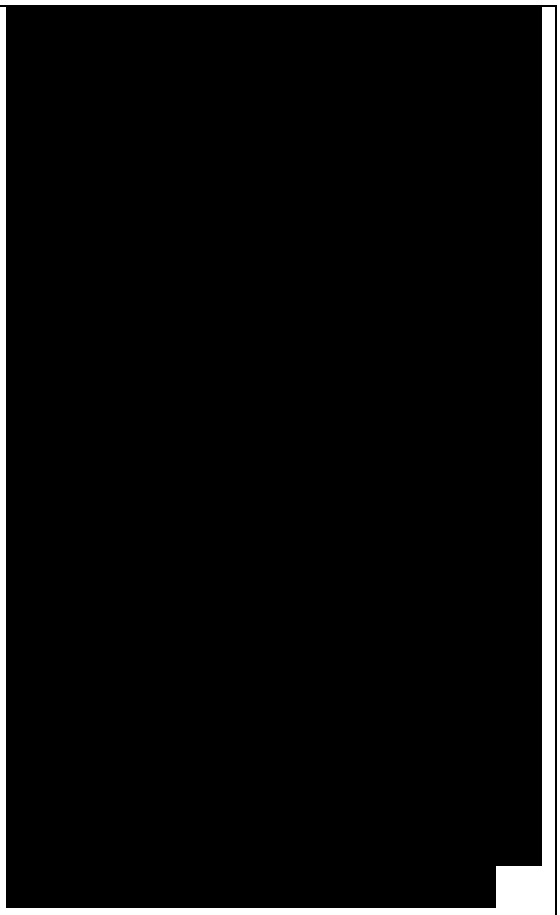


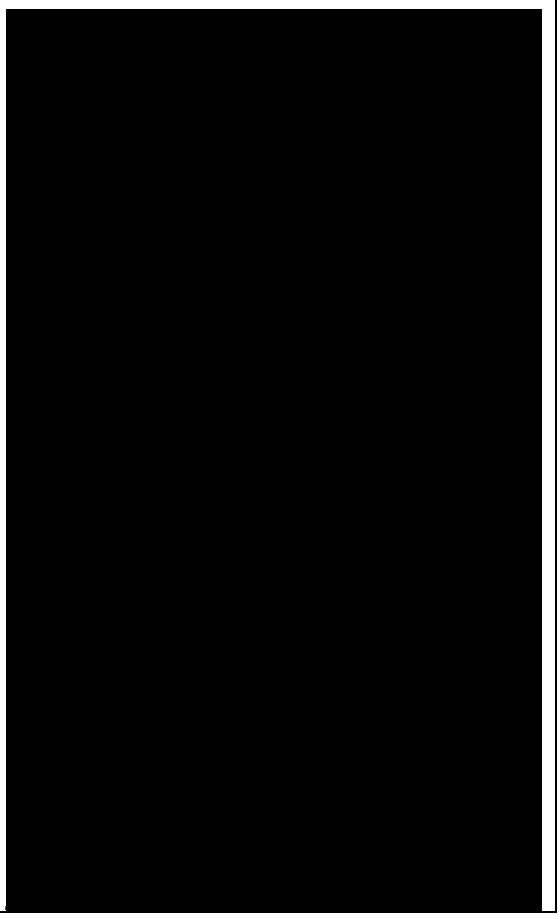
Figure 3.2: Layer definition. As currently there is no way to damp out small interactions, the PL must contain two unit cells in this case, as shown in figure Layer definition (page 20). It follows that the correct definition of a PL depends both on the geometry of the system and the interaction cut-off distance. After having defined a proper PL, we then build a structure consisting of a device region with 2 PLs and contacts at each end of this region, each with 2 PLs. Note: For the pristine system, the equilibrium results should not depend on the length of the device region, as the represented system is an infinite ideal nanoribbon with discrete translational symmetry along the ribbon. The input atomic structure must be defined according to a

specific ordering: the device atoms come first, then each contact is specified, starting with the PL closer to the device region. For an ideal system defined by repetition of identical PLs, the tool buildwire (distributed with the code) can be used to build a 1D geometry with the right ordering. When you type: buildwire 2cell_7.gen you will be asked to type the name of the file containing the input supercell (2cell_7.gen) and the number of principal layers in the device region (we will set this to be 2). buildwire will always give its output as a supercell structure, but in some cases we will need to manually modify this structure file so that it corresponds to the ordering explained in the previous paragraph. The following output will be visualised:



…….

The indexes iatc and PLs define the atoms belonging to the contacts, to the device region and to the PLs of the device region, and will be useful when we will write the input files. You should take a note of them (the iterative algorithm used in solving the Green's function requires these values). A file Ordered_2cell_7.gen will have been created, and defined as a supercell format GEN file (S), which I will rename device_7.gen for the following. We can better understand the ordering of the atomic indexes if we convert this structure to XYZ, open it with jmol and then change the colours of specific ranges of atoms by using the following syntax in the jmol console (for example, we select here the first contact and split it into two sub-



ranges containing its first and second PLs):

Check xong

In Figure The PLs of contact 1 (page 22) a Jmol export of the structure is shown.The yellow and red atoms represent the first and second PLs of the first contact. When you build a structure yourself, it is always a good idea to use a visualiser and verify that the atomic indices are consistent with the transport setup definitions.The last step is to change the supercell definition in the gen structure file. From the point of view of an open boundary condition calculation, Supercell (S) and cluster (C) have a slightly different meaning with respect to a canonical dftb calculation. By Supercell we mean any structure which is periodic in any direction transverse to the transport direction, while for cluster we mean any structure not periodic in any direction transverse to transport. It follows that purely 1D systems, like nanowires and nanoribbons, should be regarded as clusters (C). Therefore we edit the structure file device_7.gen, changing in the first line the S (supercell) to be C (cluster) and remove the last four lines, which would normally only be defined for periodic systems.This is the file we get after running buildwire:

Note that the numbering of atoms at the start of each line, as output by buildwire are sequential according to the numbering of the initial structure, not its global position in the output

file.The corrected definition for the 1D ribbon with open boundary conditions is then:

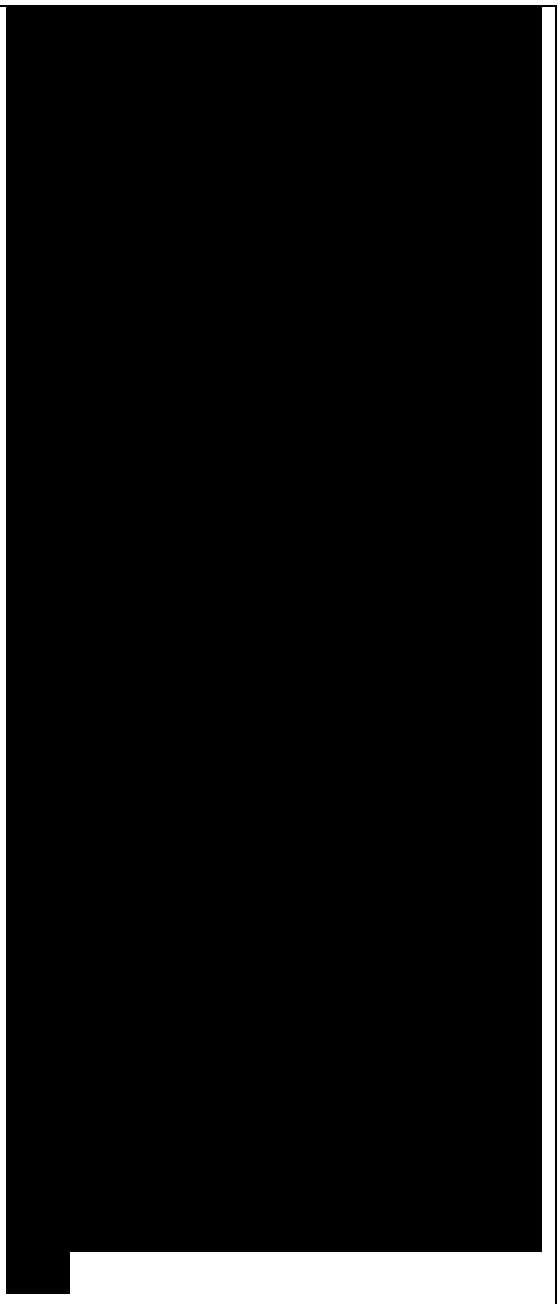
Now the file device_7.gen contains the correct structure, defined as a cluster and with the proper atom ordering. Next, we set up the input file for a tunnelling calculation.

3.1.2 Transmission and density of states

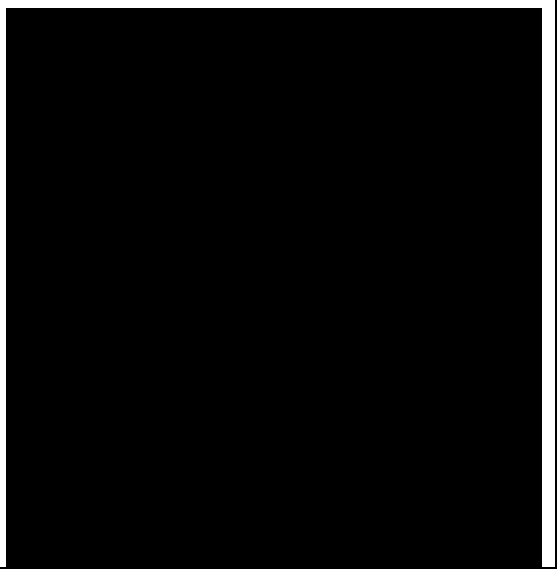
In the DFTB+ input format, settings related to a transport calculation may be required to appear in separate sections of the dftb_in.hsd file, depending on the functionality they invoke. In the following we will set up the simplest open boundary condition calculation: a calculation of transmission coefficients according to the Landauer-caroli formula, assuming a non-Scc DFTB hamiltonian. We will analyse and comment the different sections contained in the file dftb_in.hsd. First, we have the specification of the geometry:This follows the same rule as in a regular DFTB+ calculation, except for the fact that the structure should follow the specific partitioning structure explained in the previous section. Whenever an open boundary system is defined, we have to specify a block named Transport which contains information on the system partitioning and additional information about the contacts to the device:

Here we have used the indexes printed by buildwire. Device contains two fields: AtomRange specifies which atoms belong to the extended device region (1 to 136) and

FirstLayerAtoms specify the starting index of the PLs in the device region. This field is optional, but if not specified the iterative algorithm will not be applied and the calculation will be slower, even though the result will be still correct. Then we have the definitions of the contacts. In this example we define a two terminal system, but in general N contacts are allowed. A contact is defined by an Id (mandatory), the range of atoms belonging to the contact specified in AtomRange (mandatory) and a FermiLevel (mandatory). The potential is set by default to 0.0, therefore need not be specified in this example. Note that according to equilibrium Green's function theory, the Fermi level and the contact potential are not necessary to calculate the Transmission curve, but are required to calculate the current via the Landauer formula, as they would determine the occupation distribution in the contacts.Then we have the Hamiltonian block, which describes how the initial Hamiltonian and the Scc component, if any, will be calculated:



In this example we will calculate the transmission according to caroli (referred by some authors as Fisher Lee) formula in a non-SCC approximation, i.e. the Hamiltonian is directly assembled from the Slater-Koster files and used as is to build the contact self energies and the extended device Green's function. The definition of an eigensolver is not meaningful in an open boundary setup, as the system is instead solved by the Green's function technique.



Therefore we just use a keyword TransportOnly to indicate that we do not want to solve an Eigenvalue problem. The other fields are filled up in the same way as for a regular DFTB calculation. In general, in DFTB+ an Eigensolver is regarded as a calculator which can provide charge density in the SCC cycle, therefore we will define a Green's function based eigensolver later, but only for Scc calculations. Note that as C-H bonds are present in the system, charge transfer should occur, hence the result will not be accurate at the non-SCC level. It is not a-priori trivial to predict whether this affects qualitatively or quantitatively the transmission. We will therefore later compare these results with an SCC calculation - at the moment we will stay at the level of a non-SCC calculation, because it is faster to execute and also allows us to use the simplest input file possible. Finally, the implementation of the Landauer-Caroli formula is regarded as a post-processing operation and specified by the block TunnelingAndDos inside Analysis:

TunnelingAndDos allows for the calculation of Transmission coefficient, Local Density of States (LDOS) and current. A transmission is always calculated using the energy interval and energy step specified here. The LDOS is only calculated when sub-blocks Region are defined. Region can be used to select some

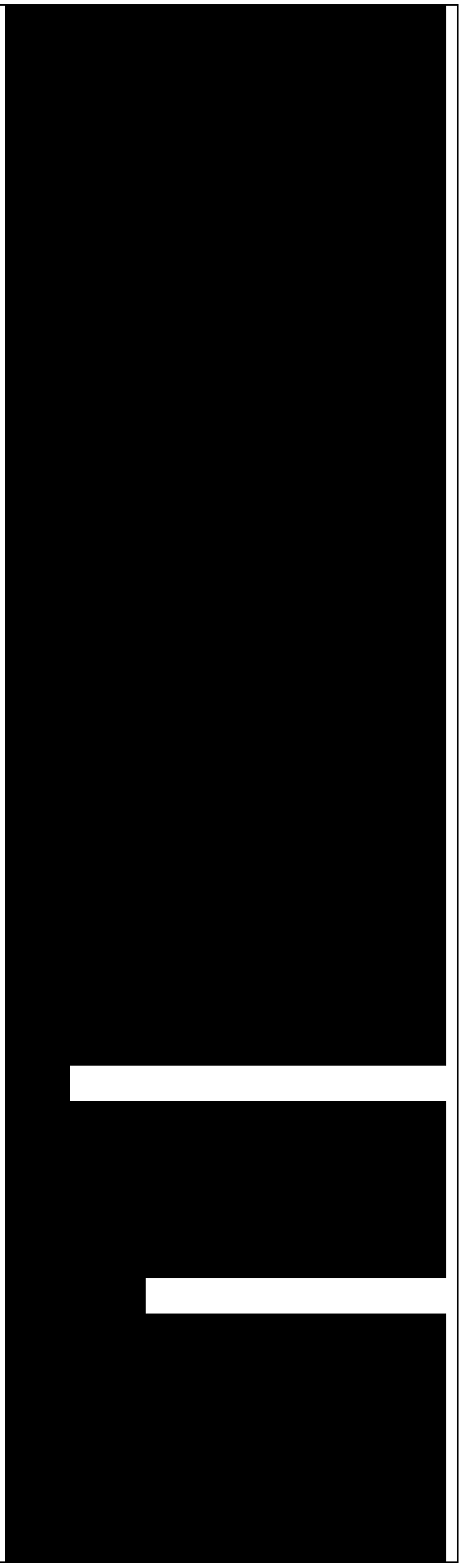
specific subsets of atoms or orbitals, according to the syntax explained in the manual. In this example, we are specifying the whole extended device region (atoms 1 to 136). Note that the energy range of interest is not known a priori. Either you have a reference band structure calculation, therefore you know where the first sub-bands are (this is the correct way to do this), or you can run a quick calculation with a large energy step and on the basis of the transmission curve then refine the range of interest.Now that the input file is complete, we have to complete one last step. During a transport run, DFTB+ will look for two directories named GS and contacts. We have to create these directories in advance:…We can then start the calculation: dftb+ dftb_in.hsd | tee output.log. We can take advantage of parallelisation over the energy points in the calculation by running the code with mpirun: mpirun -n 4 dftb+ dftb_in.hsd | tee output.log. where 4 should be substituted by the number of available nodes.

~~Note that NEGF is parallelised over energy points,~~ therefore a number of nodes larger than the energy grid will not improve performances and secondly that the memory consumption is proportional to the number of nodes used - this may be critical in shared memory systems with a small amount of memory per node. When the calculation has finished, the transmission and density

of states are saved to both the detailed.out file and to two separate tunneling.dat and localDOS.dat files. These additional files both contain the energy points in the first column and the desired quantities as additional columns.We can plot the transmission by using the plotxy script: plotxy —xlabel 'Energy [eV]' —ylabel 'Transmission' -L tunneling.dat. The plot is shown in Figure Non-SCC transmission through a pristine AGR (page 25):Figure 3.4: Non-SCC transmission through a pristine AGR. The ribbon is semiconducting, therefore we can see a zero transmission at energies corresponding to the band gap. As the system is ideal, outside of the band gap we can observe the characteristic conductance steps where the value of the transmission is 1.0 for every band which crosses a given energy. This is a normal signature of ideal 1D systems with translational invariance. Similarly, we can visualise the density of states by typing (the x and y axis limits are chosen to focus on the first few sub-bands):

The result is shown in Figure Non-SCC density of states for a pristine AGR (page 26):You can plot the transmission or the density of states on a semi-logarithmic scale:

If you do so, you will obtain the plot shown in Figure Non-SCC density of states on logarithmic scale (page 26).The density of states in the band-gap is not zero, but decreases by several orders of magnitude. This is a natural consequence of the quasi-



particle nature of the Green's function formalism: every state in the system has a finite broadening in energy. Figure 3.5: Non-SCC density of states for a pristine AGR. Figure 3.6: Non-SCC density of states on logarithmic scale.

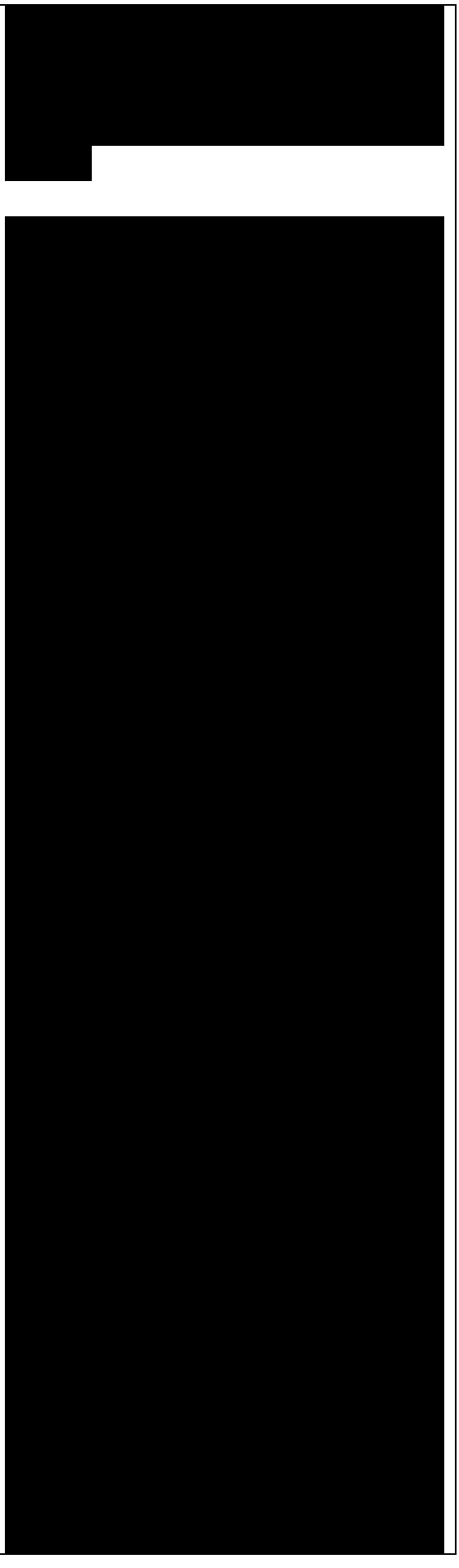
3.2 Non-SCC armchair nanoribbon with vacancy (A).

3.2.1 Transmission and Density of States: Now that we have a calculation of the reference pristine system, we will introduce a scattering centre by producing a vacancy in the system. In order to do so, we directly modify the structure file device_7.gen and the input file dftb_in.hsd. We remove atom number 48 from the structure file. Note that DFTB+ ignores the indexes in the first column of the .gen file, therefore we do not need to adjust them. We have, however, to remember to change the total number of atoms in the first line from 408 to 407:The resulting structure should look like this:

We then also adjust the dftb_in.hsd file accordingly. As we have removed an atom, all the indexes in the transport block need to be adjusted properly. Note that we removed an atom in the first PL of the extended device, therefore we also need to adjust the values of FirstLayerAtoms. The Transport block now reads:

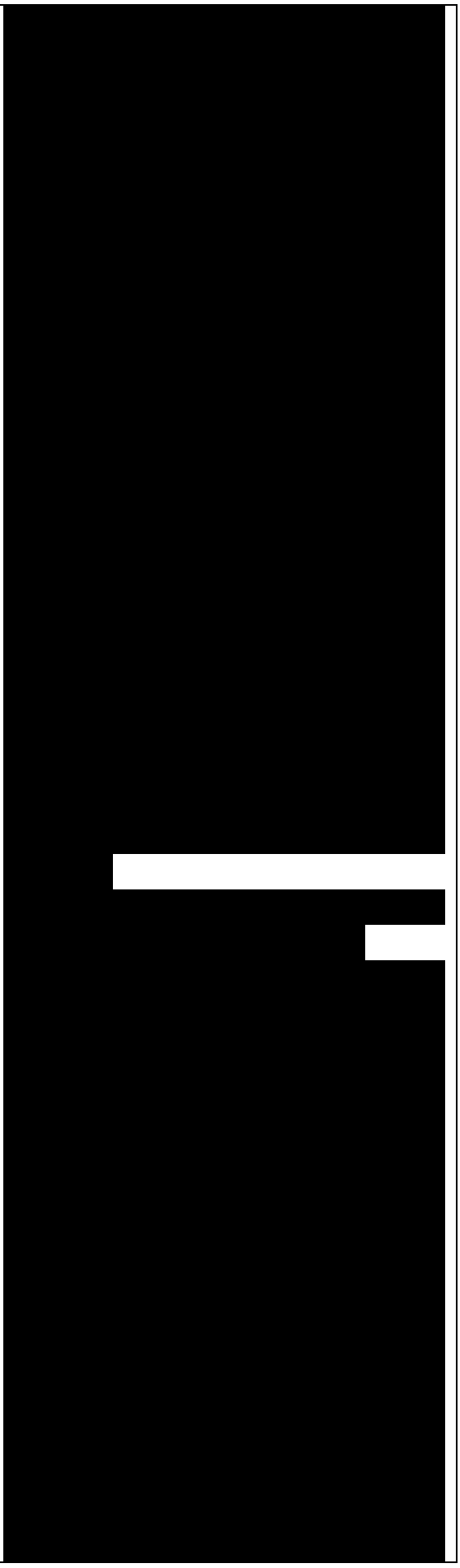
Compared to the pristine system, we have modified AtomRange in all the blocks and the values of FirstLayerAtoms. After running the

calculation, we can compare the transmission curve for this structure with a single vacancy and the pristine ribbon by using plotxy:



Energy [eV].Figure 3.8: Non-SCC Transmission in pristine (green) and single vacancy (blue) ribbons. Clearly, the presence of a vacancy introduces some finite scattering which reduce the transmission with respect to the ideal ribbon. In particular, the effect is quite small in the first conductance band while it is more visible in the first valence band and in higher bands. The reflection amplitude is increased near the band edges. This is expected in 1D systems, as near the band edges the density of states diverges (Van Hove singularities), hence the group velocity is lower, and it is known from semi-classical transport theory that the scattering probability is, when short range disorder is present, inversely proportional to the group velocity. The absence of resonant features in the transmission may point to the fact that the vacancy does not induce additional states in the conduction or valence bands. This can be verified by visualising the density of states, as in Figure Non-SCC DOS for single vacancy in sublattice A (linear scale) (page 29).The same density of states can be visualised on logarithmic scale as well, as in Figure Non-SCC DOS for single vacancy on sublattice A (semilog scale) (page 29). The vacancy is adding some close energy levels in the gap, as verified already from the DFTB calculation in the first

part of the tutorial. The Van Hove singularities are partially suppressed as the system no longer possesses translational symmetry along the transport direction. Even in a simple non-SCC approximation, the qualitative picture is consistent with the previous SCC periodic calculation. We will now consider a vacancy sitting on the other sublattice (B) and try to understand whether the relative position of the vacancy is relevant or not by calculating once more the non-SCC transmission and density of states. Figure 3.9: Non-SCC DOS for single vacancy in sublattice A (linear scale). Figure 3.10: Non-SCC DOS for single vacancy on sublattice A (semilog scale)

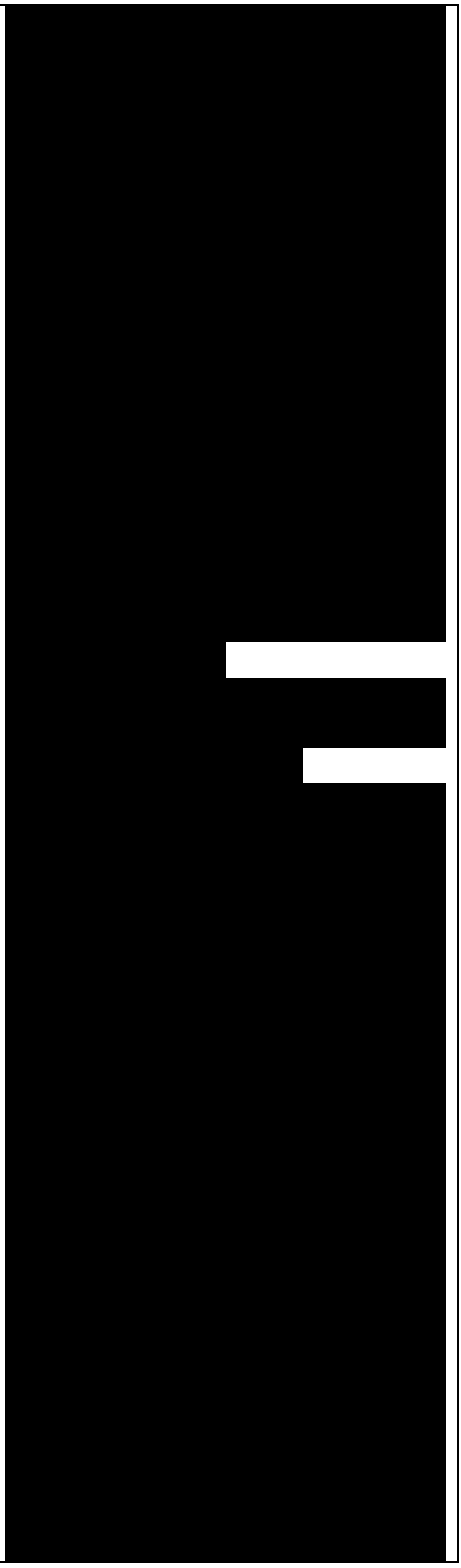


3.3 Non-SCC armchair nanoribbon with vacancy (B).

3.3.1 Transmission and Density of States: We will now consider a vacancy sitting on the other sublattice (B), i.e. we can take the structure file we used for the ideal ribbon and delete the atom number 47. The structure file is:….: Figure 3.11: Geometry with vacancy on sublattice B. Also in this case we remove an atom from the first PL of the extended device region, therefore the rest of the dftbjn.hsd input file is identical to the one we used for the vacancy on sublattice A. We can therefore just copy it and run the dftb calculation. The transmission is shown in Figure Non-SCC

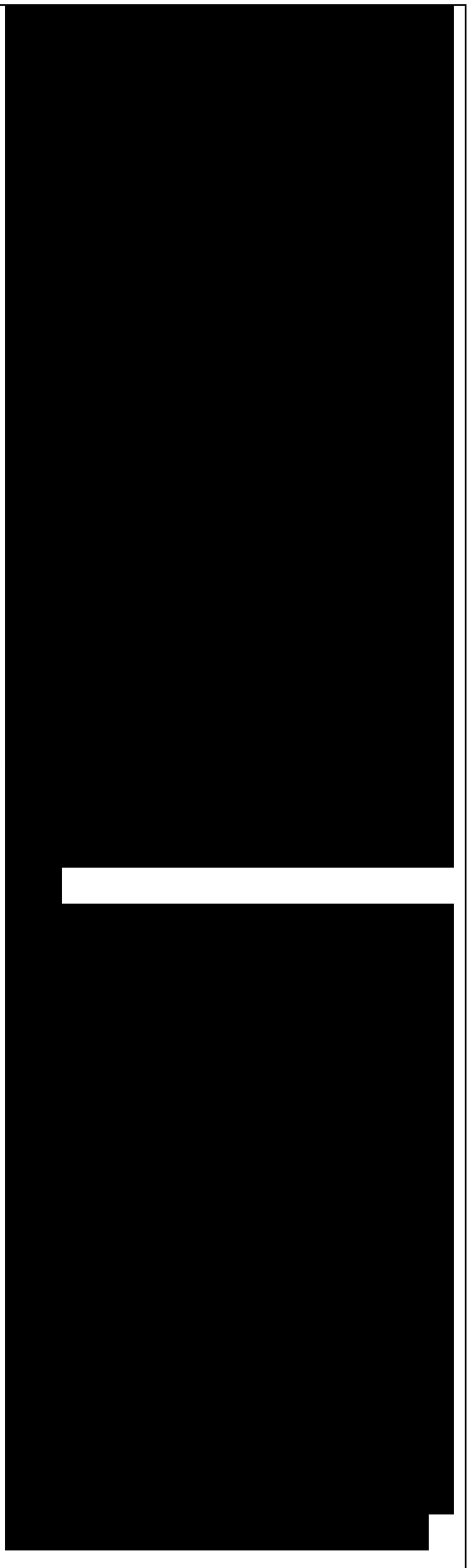
Transmission for vacancy B (blue), pristine (green) and vacancy A (green) (page 31) (Transmission for vacancy on sublattice B in blue, transmission for vacancy on sublattice A in green and pristine system in green): We can see a very strong suppression of transmission in the first sub-bands, especially in the first valence band. Again, the absence of resonances may be due by gap states. In fact, we can verify it by plotting the density of states, as shown in Figure Non-SCC DOS for vacancy in sublattice B (page 31). We can clearly see that the vacancy induces some nearly degenerate gap states, and that the density of states at higher energies is largely unaffected. It is known that the relative position of a scattering centre in a graphene nanoribbon with respect to different sub-lattices strongly affects its scattering properties, as is shown in these non-SCC calculation. Qualitatively, the picture is also consistent with periodic DFTB calculations, with the difference that we obtain directly information on the effect on transport properties via transmission function. This also ensures that we do not have to worry about choosing the right supercell or k-point sampling as the open boundary conditions represent exactly the infinite system with a single scattering centre. As already pointed out earlier, there is no	[redacted]

warranty that a non-SCC calculation give the proper result in a system if relevant charge transfer is occurring, and in general it will not. Therefore in the next section we will repeat the same calculation by solving the SCC problem. Energy [eV]: Figure 3.12: Non-SCC Transmission for vacancy B (blue), pristine (green) and vacancy A (green). Figure 3.13: Non-SCC DOS for vacancy in sublattice B



3.4 SCC Pristine armchair nanoribbon: A DFTB Hamiltonian is in general given by two terms:
Where the component Hshhis the self-consistent (SCC) correction. The SCC correction is in general needed whenever there is a finite charge transfer between atoms, i.e. whenever there are bonds between atoms with different chemical species or with different coordination numbers. In our case, we can expect a finite charge transfer between the C and H atoms at the edges, and an SCC component may be relevant. While in the previous sections, we have only considered the non-SCC component H0, in the next sections we will compute the same calculation by including the correction given by the shifts Hshifts. Note that the equilibrium SCC problem can be tackled in two ways: we could apply the Landauer-Caroli to an SCC Hamiltonian taken, for example, from

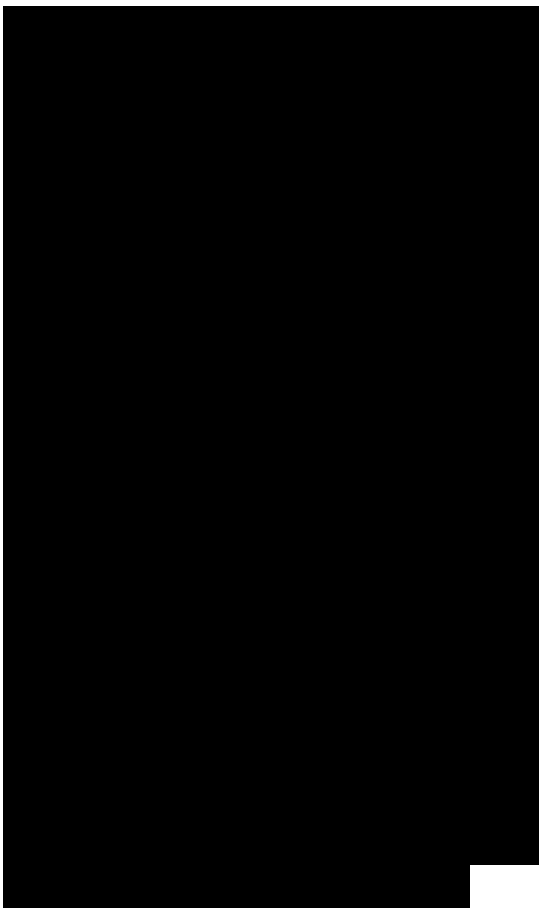
a periodic calculation (i.e. uploading the SCC component), or we can solve the problem as a full NEGF setup with 0 bias. The code flow is currently such that this second procedure has to be used (however, the first technique will be available in future release). Therefore we will need to learn to set the input related to two other components of the NEGF machinery: the real space Poisson solver and the Green's function density matrix.In this way we will introduce a first complete input file. It is important, from a didactic point of view, to be clear that as long as the applied bias is zero and we are interested in equilibrium properties, the two approaches are equivalent and the results are only valid in the limit of linear response.



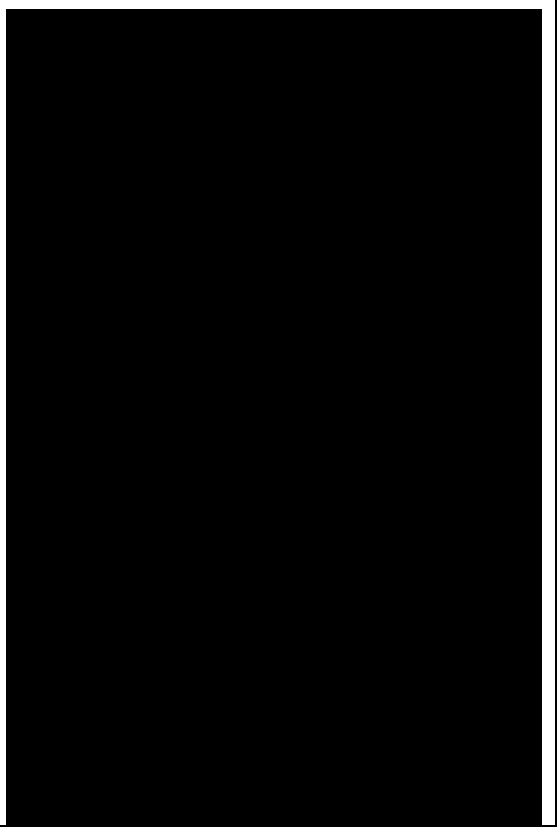
3.4.1 Contact calculation: In order to run an SCC transport calculation, the code needs some additional knowledge about the contact PLs. In particular, the SCC shifts and Mulliken charges have to be saved somewhere to enable consistency between the calculation of the self-energy and the calculation of the Poisson potential. To this end, we have to introduce an additional step in the procedure: the contact calculation. The contact calculation is simply a DFTB periodic calculation for the contact PL. As such, not all the field defined in the transport are meaningful and the input file will of course look different. The Geometry block is identical:

While the Transport block needs to be modified as follows:

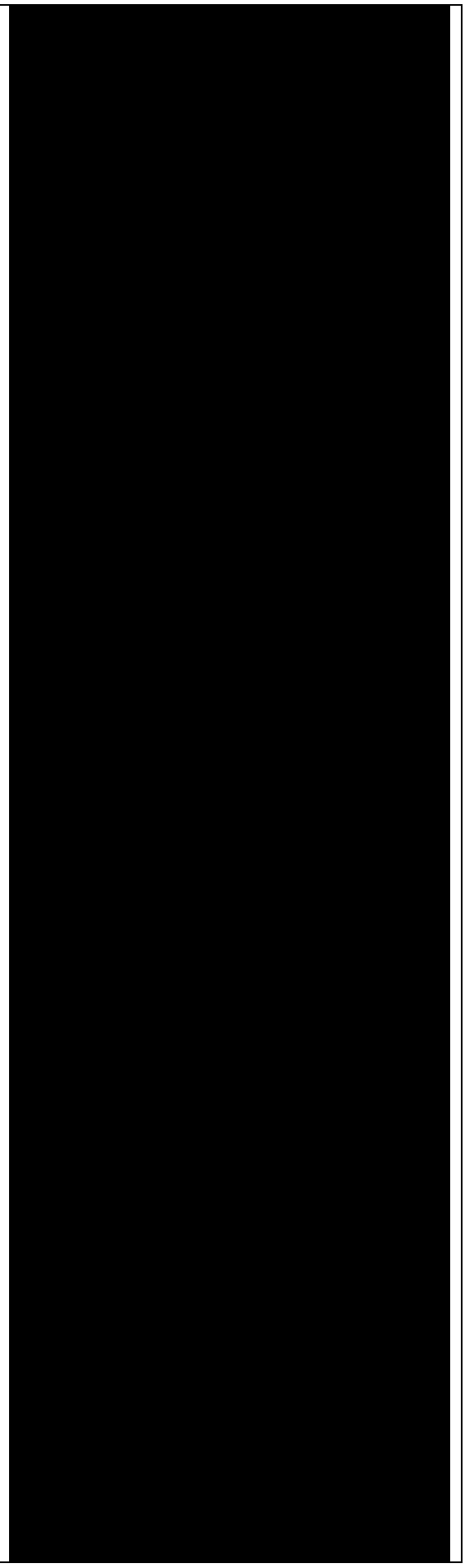
We first notice the addition of an option Task =ContactHamiltonian{...}, which was previously absent. This block specifies that we intend to calculate the bulk contact SCC properties, and the field ContactId specifies which contact we want to calculate. The field FirstLayerAtoms in the Device block is absent (it does not make sense in a contact calculation) and so are the fields FermiLevel and Potential in the two Contact sections, as they are not meaningful during this step. In general, the philosophy of a DFTB input file is that if input fields that would be useless or contradictory are present, the code will halt with an error message. The Hamiltonian block shows some differences, too:



The flags SCC=Yes and SCCTolerance=1e-6 enable the SCC calculation. A small tolerance in the contact calcu-lation, and in general in transport calculation, helps to avoid artificial mismatches at device/contact boundaries. The parameter EwaldParameter needs to sometimes be set when using parallel calculations to reduce the size of the neighbour list. Typically, the code may complain about a too small parameter: in that case, setting a value of 0.1 is considered to be good practice. The other parameters are the usual ones, except for the KPointsAndWeight, which deserves special attention. The bulk contact is of course a periodic structure, hence



we need to specify a proper k-point sampling, as we would do in a regular periodic DFTB calculation. However, you should be careful about the way the lattice vector is internally defined. In the input system is a cluster (C), i.e. it has no periodicity in direction transverse to the transport directions, the lattice vector of the contact is internally reconstructed and assigned to be the first lattice vector, regardless the spatial orientation of the structure. This means that the KPointsAndWeights for a cluster system are always defined as above: a finite number of k-points along the first reciprocal vector (according to a 1D Monkhorst-Pack scheme) and a Gamma point sampling along the other two directions. The reason for this choice is that we do not want to assign a specific direction to the structures, i.e. at this level we do not assume in any way that the structure must be oriented along x,y or z direction. Note also that as the contact information is used in the transport calculation, it is a good idea to use a dense k point sampling and a low SCC tolerance, in order to get a very well converged solution. The contact calculation will be usually much faster than the transport calculation, so this does not usually present a problem. On the other hand, this rule regarding k-points does not apply to periodic transport calculations, as the periodicity along the transverse directions must also be preserved (refer to the following section for a periodic system example). We can run the calculation by typing: dftb+ dftb_in.hsd After running the calculation, we notice

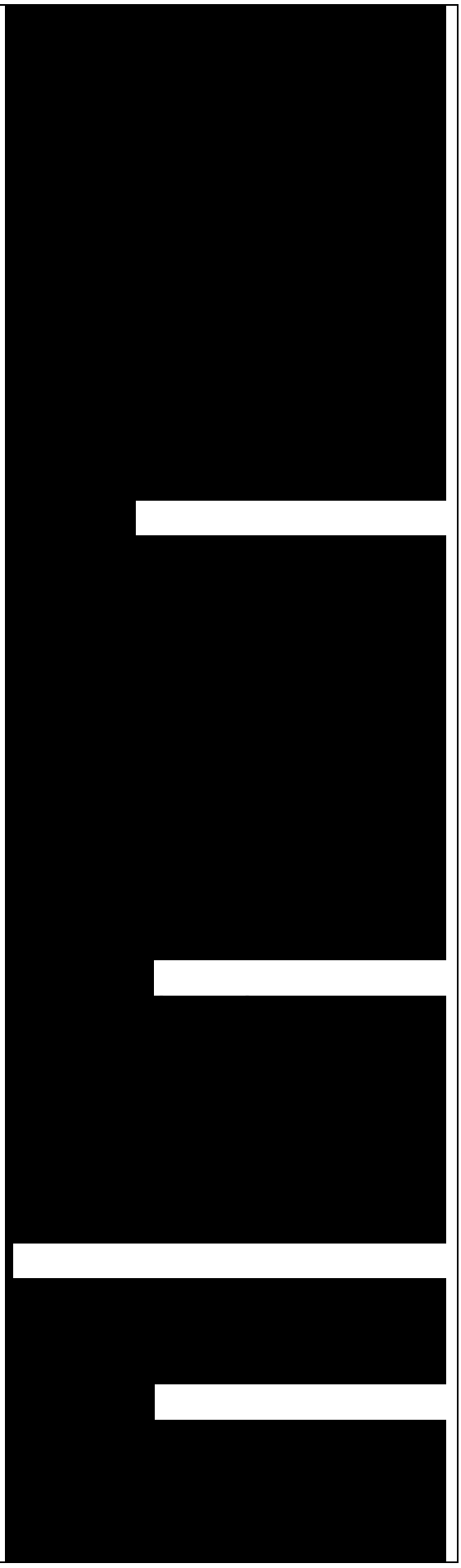


that a file shiftcont_source.dat is generated. This file contains the infor-mation useful for the transport calculation (shifts and charges of a bulk contact). It is suggested you also keep a copy of the detailed.out for later reference. We can obtain the value of the Fermi energy, which we will later need, from detailed.out as -4.7103 eV. We can now run the same calculation for the drain contact by just modifying the Task block:

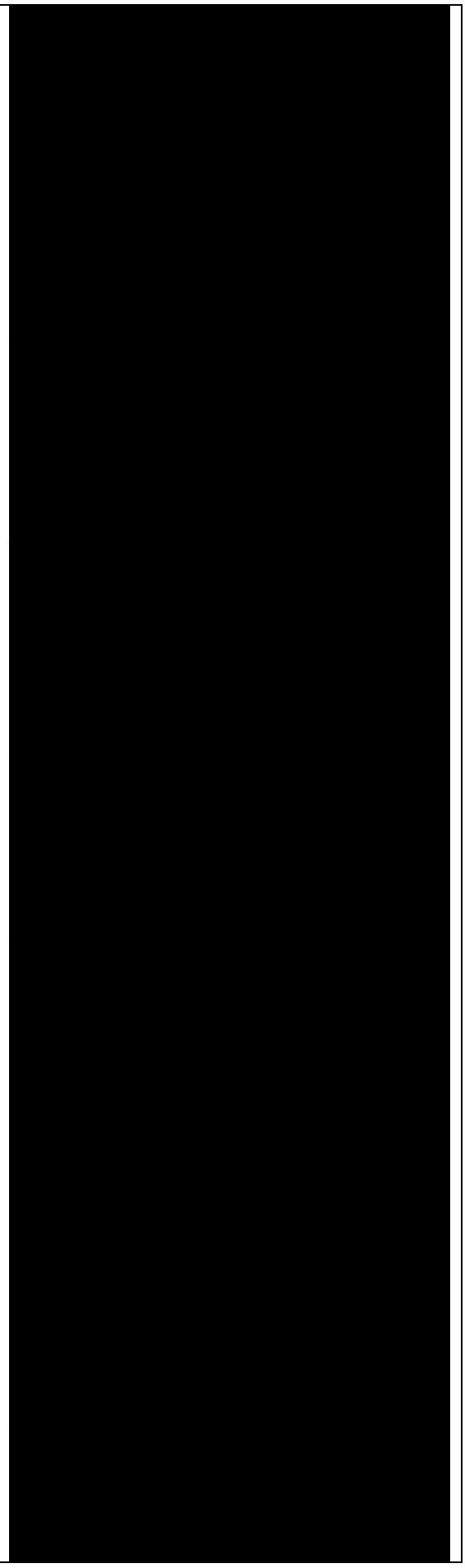
The contact are identical, therefore we expect the same results, also with the same Fermi energy. We now have a file shiftcont_drain.out, which is equivalent to shiftcont_drain.dat apart from small numerical error. In fact, we could have simply copied the previous contact results into this file.Now that the contact calculation is available, we can set up the transport calculation.

3.4.2 Transmission and Density of States: In order to calculate the transmission for the SCC system, we have to copy the files shiftcont_drain.dat and shiftcont_source.dat into the current directory: cp ../contacts/shiftcont* .

Then, we have to specify some additional blocks with respect to a non-SCC calculation. We first look at the Transport block.:
The atom indices are of course the same, as the geometry of the system is not changed. This time though, we explicitly specified a Task block



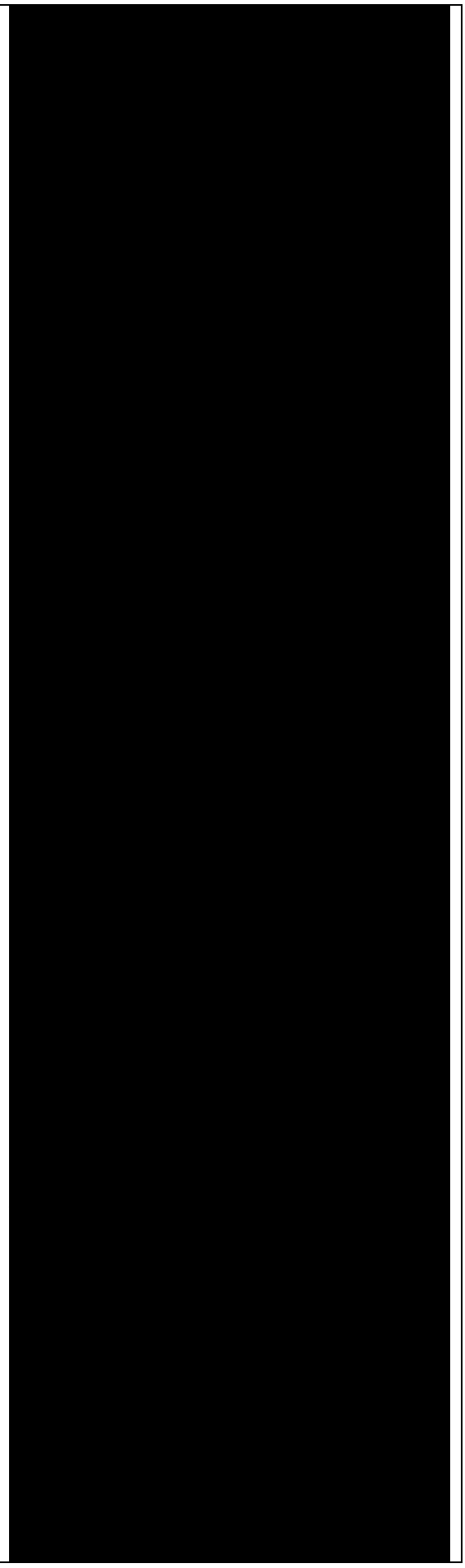
named UploadContacts, which declares that we are now running a full transport calculation. Task=UploadContacts{} is the default and does not take any additional parameters, therefore you can safely omit it. Now that we are solving the full SCC scheme, we will allow for charge transfer between the open leads and the extended device region, therefore it is important to set a well-defined Fermi energy. While this does not make any difference in a non-SCC transmission calculation, it is crucial for the SCC calculation. A wrong or unphysical Fermi energy will lead to unphysical charge accumulation or depletion in the system. To this end, you will have to pay some attention to the definition of the Fermi energy. As we are calculating a semiconductor system, the Fermi level should be in the energy gap. By calculating a band structure or by inspection of the eigenvalues in the file detailed.out you can verify that the value -4.7103 is on the edge of the conduction band. This can be explained as numerically the Fermi level is defined by filling the single particle states till the reference density is reached, therefore its position inside the gap of a semiconductor is arbitrary. Therefore, while in metallic system we may ensure consistency and use a well calculated Fermi level at some specific temperature during all our transport calculation, in the case of a semiconductor system we can manually set the Fermi level in the middle of the energy gap (for this system, roughly at -4.45 eV) and freely vary the temperature as long as



the gap is larger than several times the value of kT. We will see in the following that there are some ways to verify that the Fermi level is defined consistently, as this is often source of confusion. Note also that, differently from other codes, dftb+ allows for different Fermi levels in different contacts, which can be useful when heterogeneous contacts are defined (for example, in a PN junction). In that case a built-in potential is internally added to ensure no current flow at equilibrium. In the Hamiltonian block now an SCC calculation has to be specified:

MaxAngularMomentum and SlaterKosterFiles are not modified. But differently from the non-SCC calculation, we now need to specify a way to solve the Hartree potential and the charge density self-consistently. In a NEGF calculation, we use a real-space Poisson solver to calculate the potential, and a Green's function integration method to calculate the density matrix:

The Poisson section contains the definition of the real space grid parameters. Note that differently from a normal dftb+ calculation, simulating regions of vacuum is not for free, as the simulation domain must be spanned by the real space grid. The grid is always oriented along the orthogonal cartesian coordinate system. Poissonbox specifies the lateral length of the grid. The length along the transport direction is ignored as it is automatically determined by the code

(in this case, z=30.0). The length along the transverse direction are relevant and should be carefully set. In order not to force unphysical boundary conditions, you may extend the grid at least 1 nm away. If a strong charge transfer is present, you may go for a larger grid according to your available computational resources. A poorly defined grid can lead to no convergence at all, to a very strange (and slow) convergence path or to unphysical results. MinimalGrid specifies the minimum step size for the multigrid algorithm. Values between 0.2 and 0.5 are usually good, where a lower value stands for higher precision. SavePotential = Yes will return a file containing the potential and charge density profile, for later reference. These files can be quite large, therefore the default is No. The Eigensolver is now specified as GreensFunction. With this definition, we instruct the code not to solve an eigenvalue problem but rather to calculate the density matrix by integration of the Keldysh Green's function. This block provides the SCC charge density with or without applied bias. The options define the integration path. Usually the default options are good enough in most cases and advanced users may refer to the manual and references therein. The Mixer options is present in dftb calculations as well. Convergence is known to be critical in NEGF schemes. In that case, a lower MixingParameter value will help to avoid strong oscillation in the SCC iterations. The last block is Analysis This block is identical to the non-scc	

calculation as the same task is performed: calculation of Transmission, current and DOS by using the Landauer-Caroli formula. The Transmission will be of course be different due to the fact that the ground state charge density is now solution of the SCC Hamiltonian and we have slightly changed the energy range as the SCC component introduce a shift of the band-structure (try to compare the SCC and non-SCC transmission results when you are done). We can now run the calculation (after defining the directories GS and contacts):

[redacted]

Where -n 4 should be adapted to the number of available nodes. As transport calculations in dftb+ are parallelised on energy points, a quantity larger than 40 (the default number of integration points at equilibrium) will not speed up the calculation of the density matrix. An inspection of the file detailed.out reveals that we have additional information with respect to the non-SCC calculation, including a list of atomic charges and orbital population, as now the SCC density matrix has been calculated. The transmission is also saved as separate file, and is shown in Figure SCC transmission in pristine AGR (page 36). Figure 3.14: SCC transmission in pristine AGR. As you'd expect, it still step-like as in the non-SCC calculation. This is correct, as we're calculating an ideal 1D

[redacted]

system. The bandwidth (i.e., the steps width) may differ due to SCC contribution and the overall transmission is shifted. Note that while the non-SCC calculation is very robust, meaning that you will always get step-like transmission for a 1D system, in the SCC calculation a poor definition of the boundary conditions, of the bulk contact properties or of the additional GreensFunction and Poisson blocks may induce numerical artifacts and scattering barriers which should not be there. As a result, the transmission will not appear step-like but rather visibly smoothed out. You can also verify the quality of the calculation by inspection of the potential and charge density profiles. In a pristine periodic system we would expect a periodic potential, without discontinuities at the boundary between extended device and electrodes. The information needed to construct the real space potential and charge density are contained in 5 files: box3d.dat, Xvector.dat, Yvector,dat, Zvector.dat, potential.dat and charge_density.dat. The first 4 files contain the grid information, and the last two ones the list of potential and charge density values (following a row major order). Those information can be converted to any useful with some simple scripting, we provide an utility called makecube which can be used to convert them to Gaussian cube format or a more flexible vtk format. There's plenty of software to visualise vtk or cube files, but unluckily at present current choices of software which are effective at visualising real space grid data are weak at visualising atomistic

structures, and vice versa. In the following we will use paraview and work with the vtk format. Paraview is freely available and is supplied with many gnu/linux distributions as a compiled package. The vtk file can be obtained by simply running:

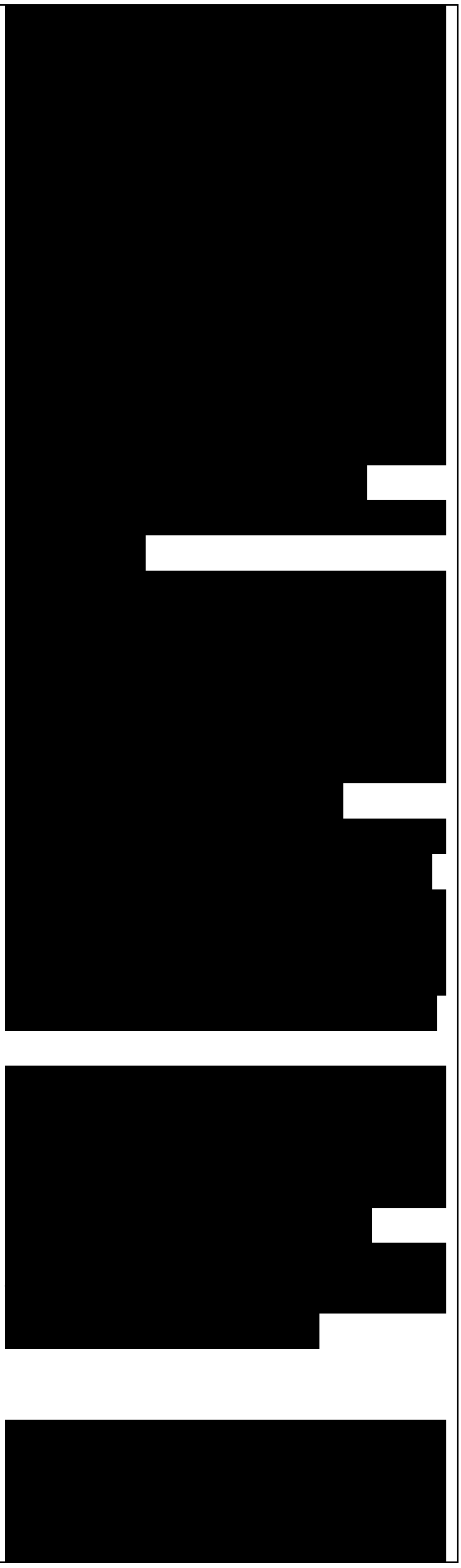


Figure 3.15: Potential profile along the nanoribbon

An extensive explanation of paraview features is beyond the scope of this tutorial. Following some easy steps, you can produce the potential map shown in Figure Potential profile along the nanoribbon (page 37).

1.Open paraview and import the file pot.vtk from File->Open

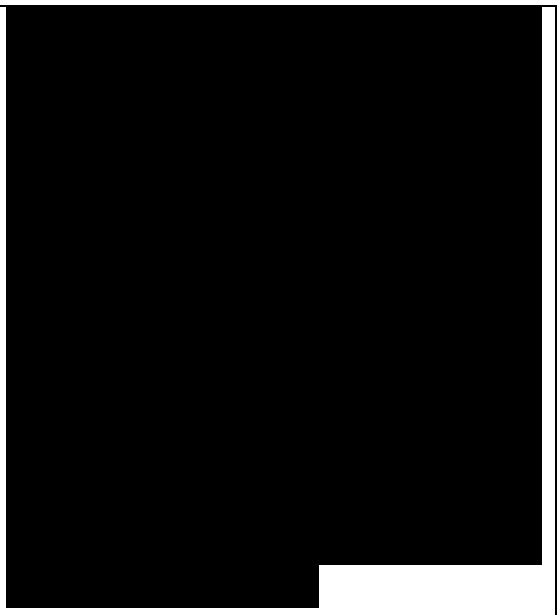
2.Click on Properties->Apply (Properties are usually visualised on the left side of the screen) and you should see the bounding box in the visualisation windows.

3.In the Pipeline browser select the file pot.vtk by clicking once on it, and then select the Clip filter from Filters->Alphabetical (or from the filter toolbar).

4.In Properties, click on 'Y Normal' to produce a clip along the nanoribbon.

The plot shown in Figure Potential profile along the nanoribbon (page 37) above is the self-consistent potential along the nanoribbon. We

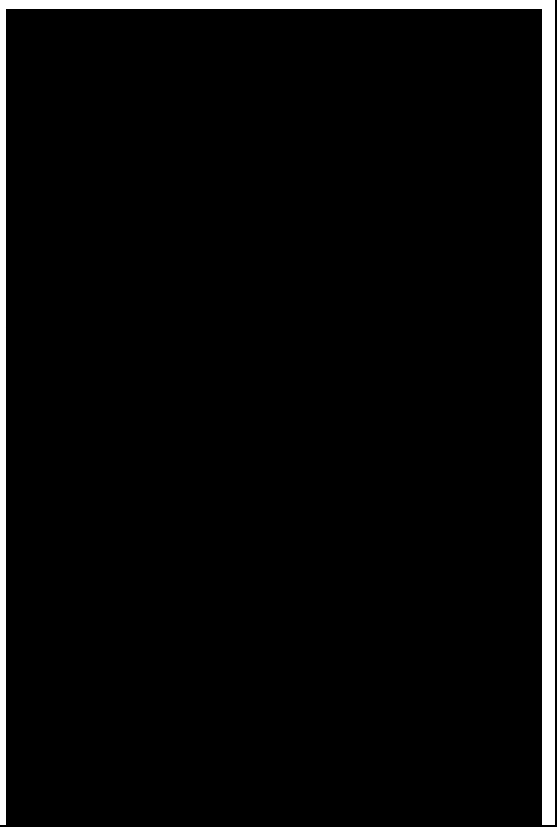
can see that the charge transfer between carbon and hydrogen at the edges results in a non-flat potential. At a first glance, the potential looks quite homogeneous, meaning that there are no clear discontinuities at the box boundary. This is important: being it a homogeneous ribbon, the potential should have the same periodicity as the lattice. We can verify this with a closer inspection by plotting a cut along a line. We apply the following steps:



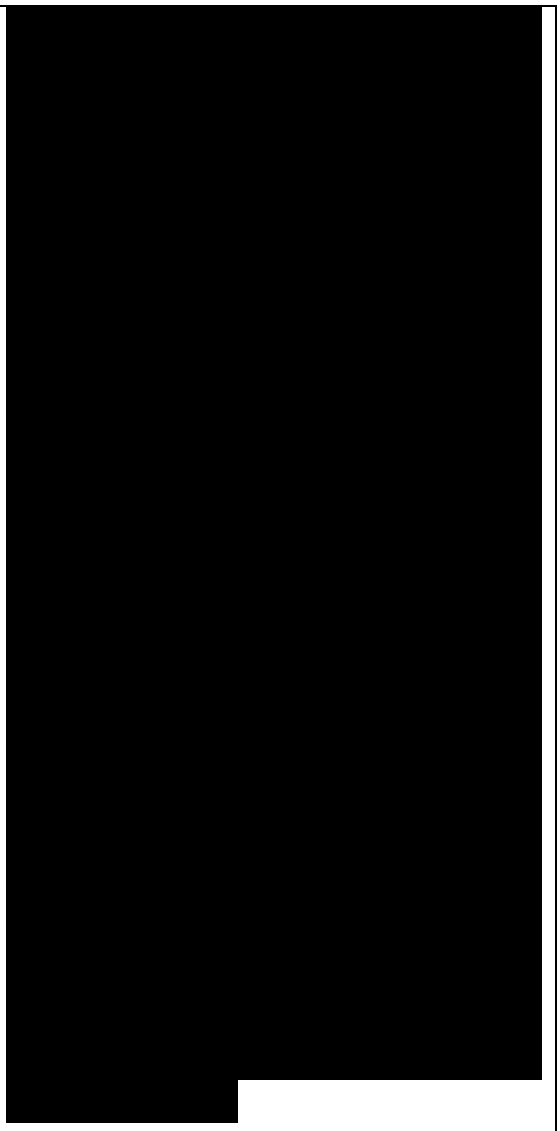
1.We select pot.vtk in the Pipeline Browser and Filters->Alphabetical->Plot Over Line

2. From the Properties window, we select ‘Z Axis’ and click on ‘Apply’ By following this procedure we obtain Figure Potential profile along the nanoribbon (page 38).

Figure 3.16: Potential profile along the nanoribbon. As you can notice, there is a discontinuity at the interface. However, it is quite small (~ 12 meV). Defining a ‘perfect’ interface between the bulk semi-infinite contacts and the device region is very difficult, especially in a semiconductor where no free charge can contribute to screen such an interface potential. A smaller tolerance in the self-consistent charge during the contact and the device calculation, a finer calculation of the Fermi level (in metallic systems) and a finer Poisson grid can decrease the discontinuity: you should be able to reach about 1 meV, but it is difficult to go below this value. However, as

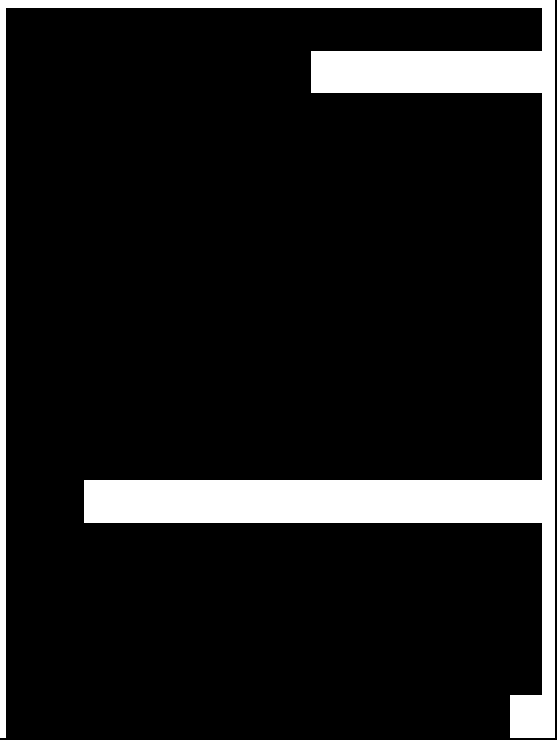


you can see in the transmission plot, as long as the discontinuity is this small, it hardly affects the transmission. However, it is important for you to verify that the behaviour at the boundaries is reasonable. Otherwise, the extended region may be too small to allow to the relevant physical quantities (charge, potential) to relax to bulk values. Be aware that numerical errors are unavoidable, therefore it is important to understand their relevance and the impact on the results. In the transmission calculation we do not notice anything different because the energy step is close to the mismatch at the boundaries. After running the calculation for the pristine system, we will introduce vacancies as we did in the non-SCC calculation. The results should be now directly comparable to the bulk periodic SCC dftb calculation.



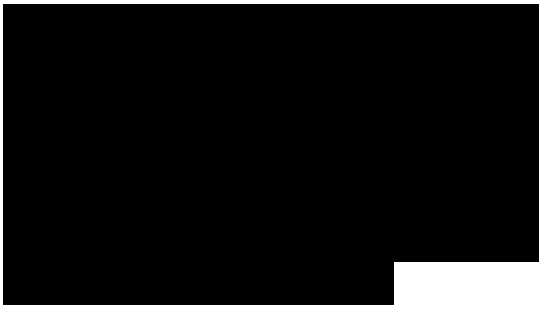
3.5 SCC armchair nanoribbon with vacancy (A):

We will now calculate the SCC transmission for the nanoribbon with a vacancy on the sublattice A, using the same input structure set up for the non-SCC calculation. The contacts are identical to the pristine case, therefore in the following we will only modify the extended device calculation.



3.5.1 Transmission and Density of States: As previously done, the transport section must be modified in order to account for the different number of atoms in the extended

device region:
We use the same Fermi level and the files shiftcont_source.dat and shiftcont_drain.dat as in the pristine system calculation, as the contacts are not modified. The Hamiltonian block is also not modified, except for an additional finite temperature:



A finite temperature is used to provide a finite temperature broadening, useful if the vacancy induces partially filled gap states. In general, temperature broadening may improve convergence and dump oscillations in the SCC iterations. The Analysis block is also similar, we add the DOS calculation to verify if we can identify a vacancy state:

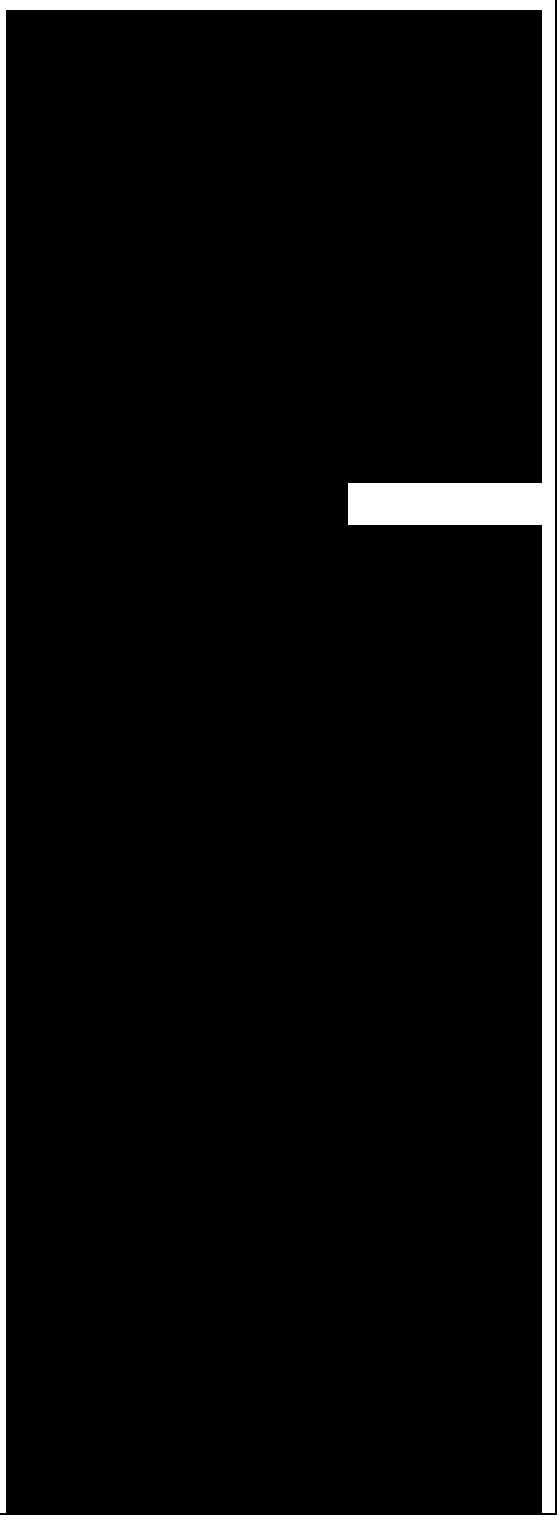
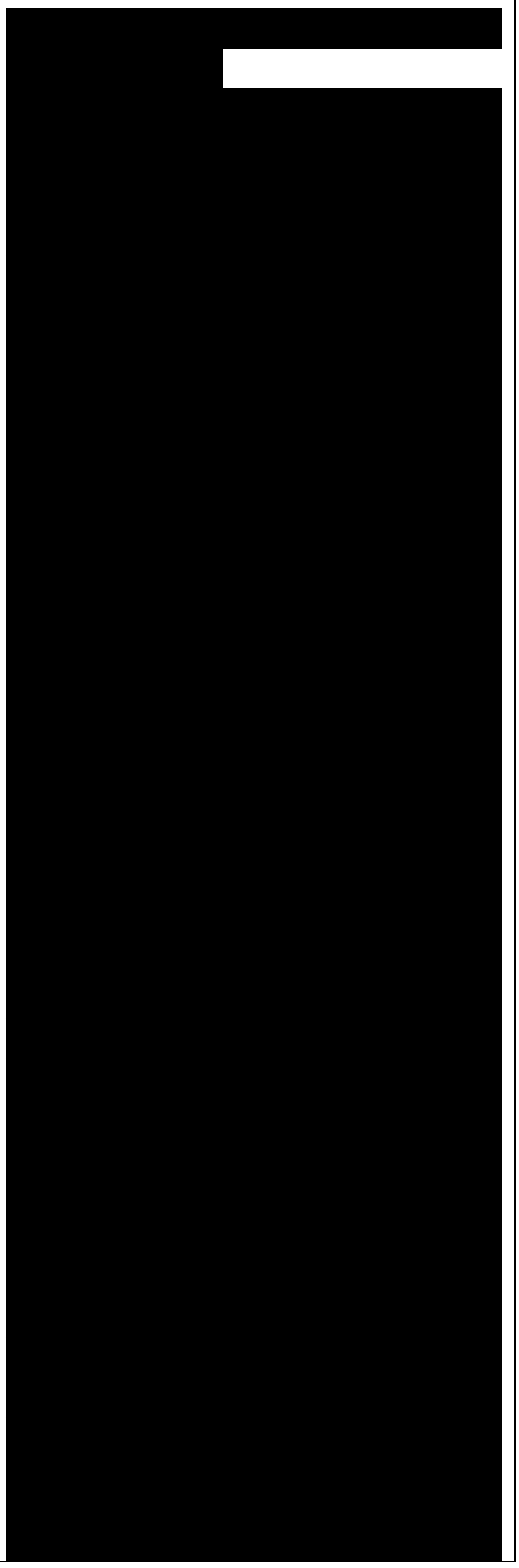
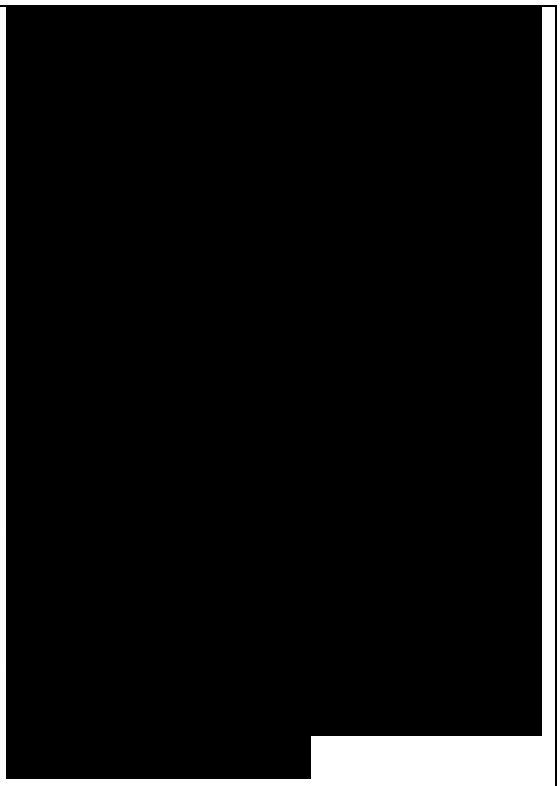


Figure 3.17: Density of states for vacancy (A). The vacancy states are located in the energy gap, consistently with the periodic calculation, and that the tunneling curve is qualitative similar to the non-scc calculation. The first conduction and valence band are weakly affected by the vacancy which does not act as a strong scatterer. There is no signature of resonances, as the additional levels are located in the gap.Note also that we previously recommended the use of large extended regions and to verify that the potential and charge density are smooth at interfaces. As you can see in Figure Potential profile for vacancy (A) (page 41), the impurity is very close to the boundaries, resulting to a potential profile which varies significantly close in to the boundary. It is left to the reader to

verify that the overall transmission does not change significantly if a longer extended region is considered.

3.6 SCC armchair nanoribbon with vacancy (B): We will now run the same calculation, but with the vacancy on the sublattice B. As in the non-SCC case, the only difference with the previous calculation is the location of the vacancy, therefore the input file is absolutely identi-cal. The contacts are the same, therefore all we have to do is copy the shiftcont_source.dat and shiftcont_drain.dat files into the current directory and run the calculation. The resulting transmission and density of states are shown in Figures Density of states for vacancy (B) (page 42) and Transmission for vacancy (B) (page 42). We immediately notice that the Van Hove singularities are strongly suppressed and that the valence band is almost completely suppressed. Consistently with the picture obtained by periodic calculation, a quasi-bounded vacancy level hybridise with the valence band edge causing a strong back-scattering. A comparison between all the three cases (see Figure Transmission for pristine system (blue), vacancy (A) (green) and vacancy (B) (red) (page 43)) shows that the scattering probability is deeply affected by the exact position of the vacancy. Figure 3.18: Transmission for vacancy (A), Figure 3.19: Potential profile for vacancy (A), Figure 3.20: Density of states for vacancy (B), Figure 3.21: Transmission for vacancy (B) .This



is, in graphene nanoribbon, generally true for other kinds of short range scattering centres such as substitutional impurities. We can also notice that, in this particular case, the non-scc approximation is qualitatively consistent for two reasons: the vacancy level are not populated and the charge transfer at the edges is not critical as the edges contribute poorly to the transmission in an armchair ribbon. Figure 3.22: Transmission for pristine system (blue), vacancy (A) (green) and vacancy (B) (red).	
CHAPTER 4: This tutorial is licensed under the Creative Common Attribution 4.0 International license.You are free to: Share - copy and redistribute the material in any medium or format. Adapt - remix, transform, and build upon the material for any purpose, even commercially.The licensor cannot revoke these freedoms as long as you follow the license terms. Under the following terms: Attribution You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.