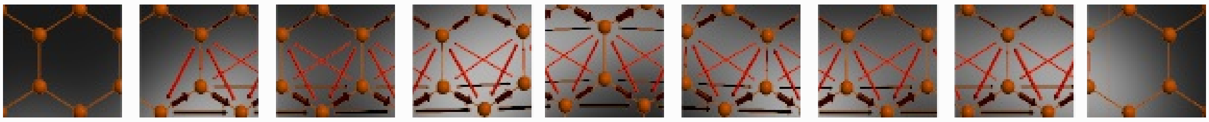
Table of Contents

Transport in a graphene nanoribbon with a distortion

Downloads & Links

- PDF version
- ATK Reference Manual
- Transport calculations with ATK

You will here use VNL and ATK to study the electron transport properties of a graphene nanoribbon with a distortion. You will be introduced to the device configuration and analysis tools that are particularly useful for investigating the properties of devices.

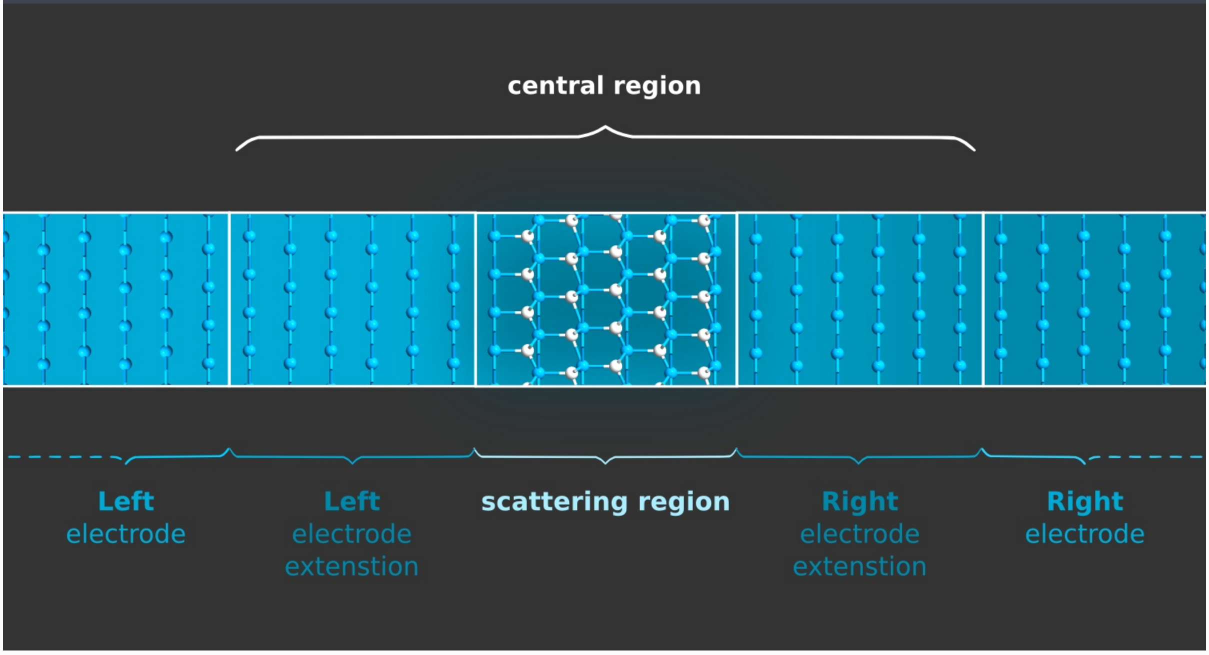


Device configuration

The nanoscale structure needed for ATK transport calculations is a semi-infinite *device configuration*. It consists of three main parts:

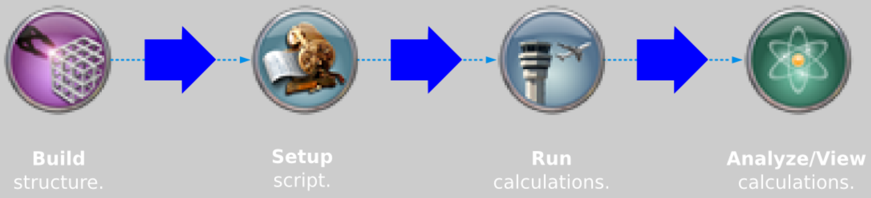
1. Left electrode.
2. Central region.
3. Right electrode.

DESCRIPTION OF THE SYSTEM



Current may flow between the two bulk *electrodes*, but has to pass through the*scattering region* in the middle of the device. Both ends of the central region, called the *electrode extensions*, must be exact replicas of the corresponding electrode.

WE FOLLOW THESE STEPS:



Build structure.

Setup script.

Run calculations.

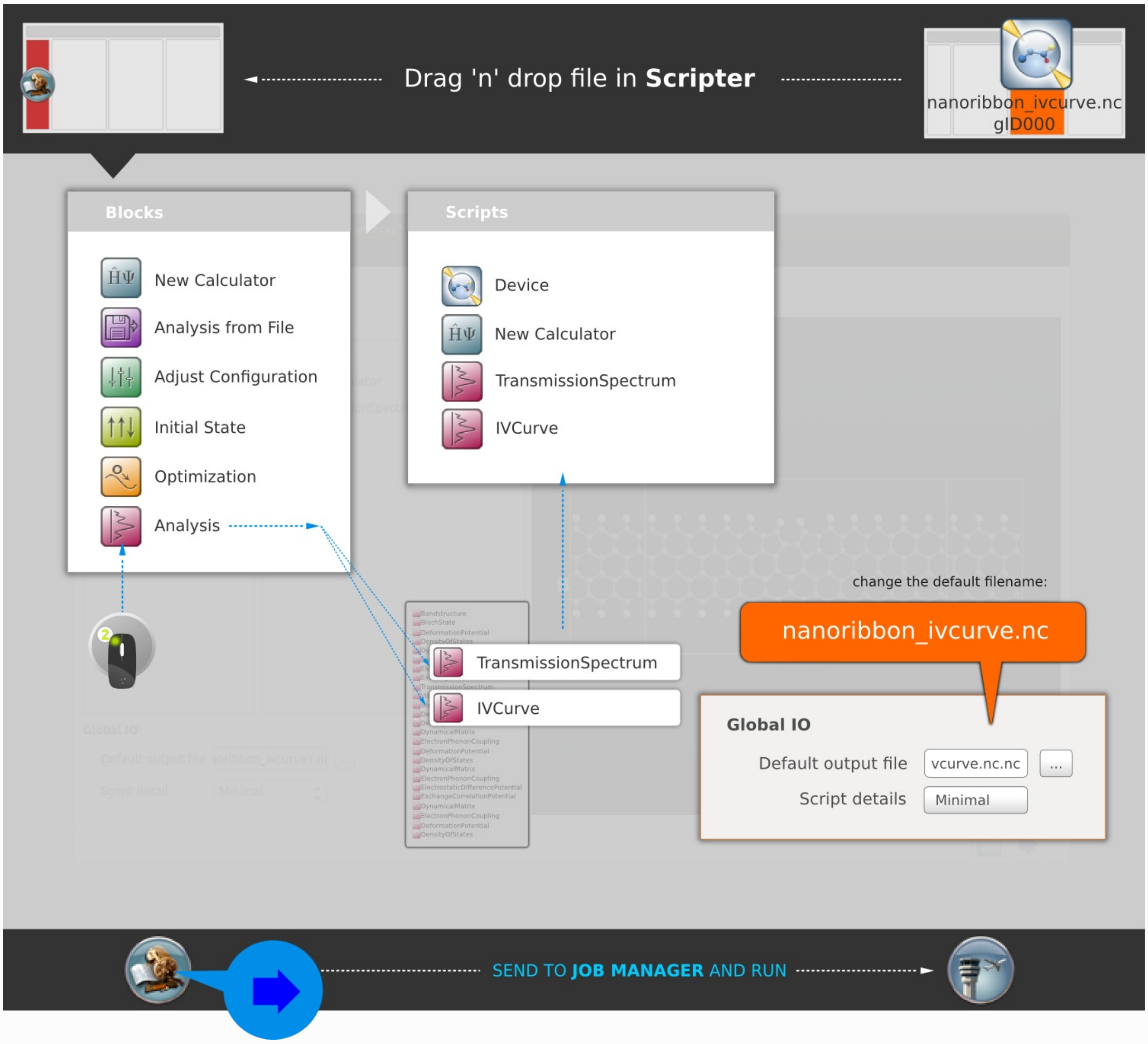
Analyze/View calculations.

Remember! Any time you can check LabFloor to see your files.

Graphene nanoribbon device

The tutorial section Building a graphene nanoribbon device teaches you how to build the graphene nanoribbon device used here. For now, however, simply use a device configuration that is already prepared for you: Start up VNL, open the project named “Example Project”, and locate the data file `nanoribbon_ivcurve.nc` in the Project Files list. Make sure the file is ticked and then follow these steps:

- On the LabFloor, select the **DeviceConfiguration** item saved in the NetCDF file.
- A calculation was already done for this device. Use the **General Info** tool to see some of the parameters used for the calculation.
- Finally, use the **Viewer** to visualize the graphene nanoribbon device. Note the defect in the middle of the central region, and that both electrodes consist of pristine (non-defected) nanoribbon unit cells.



❶ Important

The C-vector of the device configuration simulation cell is aligned with the Z-axis, which in ATK is the **transport direction**. The left/right electrodes may be different, but must both be periodic in the transport direction. The central (scattering) region is semi-infinite in the transport direction, where it couples to the electrode leads.

Transport analysis

Two types of transport analysis items have been pre-calculated and saved in the NetCDF data file; the I–V curve and the zero-bias device transmission spectrum:

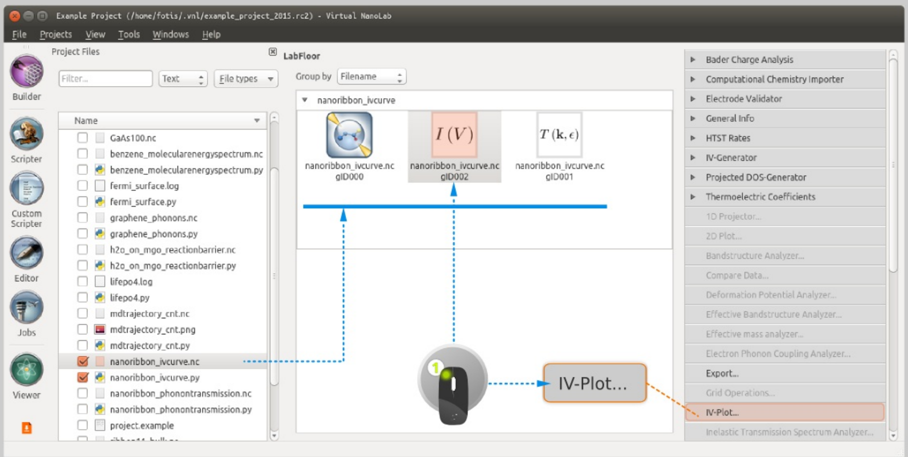
$I(V)$ IVCurve

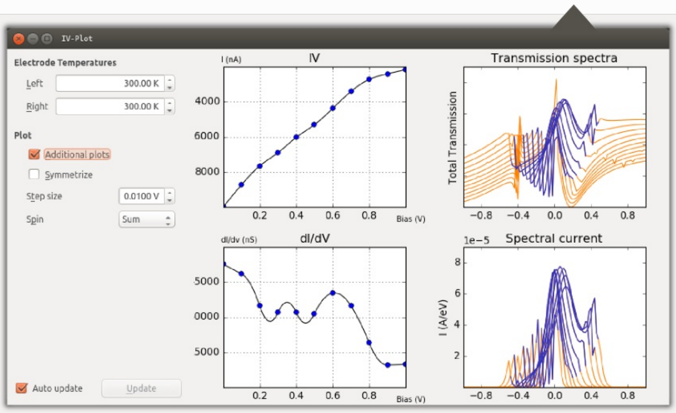
TransmissionSpectrum

I–V curve

The I–V curve is computed from finite-bias transmission spectra at several bias points. Select the **IVCurve** LabFLoor item and use the **IV-Plot** analysis tool to visualize it.

Example: **Study the I-V curve.**


Example Project (/home/fotis/.vnl/example_project_2015.rc2) - Virtual NanoLab
File Projects View Tools Windows Help
Builder
Scripter
Custom Scripter
Editor
Jobs
Viewer
Project Files
Filter...
Text
File types
Name
GaAs100.nc
benzene_molecularenergyspectrum.nc
benzene_molecularenergyspectrum.py
fermi_surface.log
fermi_surface.py
graphene_phonons.nc
graphene_phonons.py
h2o_on_mgo_reactionbarrier.nc
h2o_on_mgo_reactionbarrier.py
lifepo4.log
lifepo4.py
mdtrajectory_cnt.nc
mdtrajectory_cnt.png
mdtrajectory_cnt.py
nanoribbon_ivcurve.nc
nanoribbon_ivcurve.py
nanoribbon_phonontransmission.nc
nanoribbon_phonontransmission.py
project.example
LabFloor
Group by Filename
nanoribbon_ivcurve
nanoribbon_ivcurve.nc gID000
I (V)
nanoribbon_ivcurve.nc gID002
T (k, ε)
nanoribbon_ivcurve.nc gID001
IV-Plot...
Bader Charge Analysis
Computational Chemistry Importer
Electrode Validator
General Info
HTST Rates
IV-Generator
Projected DOS-Generator
Thermoelectric Coefficients
1D Projector...
2D Plot...
Bandstructure Analyzer...
Compare Data...
Deformation Potential Analyzer...
Effective Bandstructure Analyzer...
Effective mass analyzer...
Electron Phonon Coupling Analyzer...
Export...
Grid Operations...
IV-Plot...
Inelastic Transmission Spectrum Analyzer...



Left 300.00 K
Right 300.00 K
☑ Additional plots
□ Symmetrize
Step size 0.0100 V
Spin Sum
☑ Auto update

The rightmost window shows the transmission spectrum for each bias, where the blue part of each curve shows the bias window. For each transmission spectrum the electrical current is shown in the topmost middle plot.

Tick the option *Additional plots* to see not only the I–V curve, but also the differential current (dI/dV), and the transmission spectra and spectral currents for all sampled bias points.

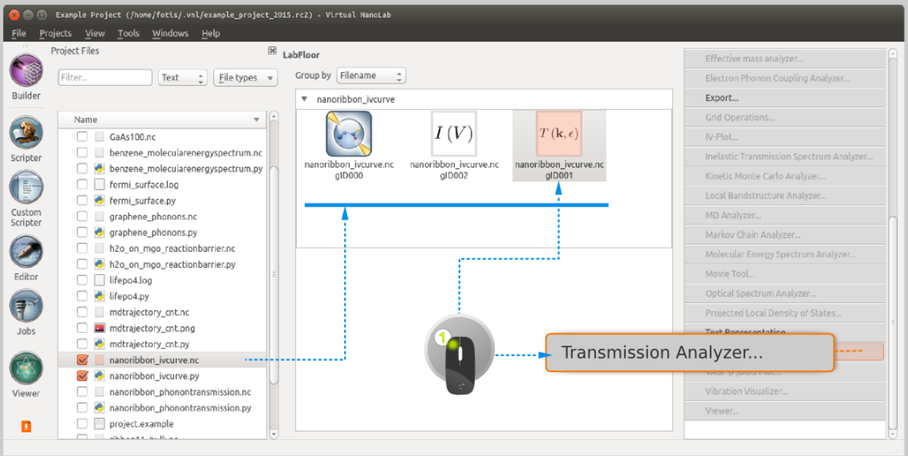
Tip

- Hover the mouse over an I–V point to highlight the corresponding transmission spectrum.
- The I–V points are connected by a spline interpolation, and the line connecting points in the dI/dV plot is obtained by differentiating this interpolation. The differentiation obviously becomes more accurate as the bias-step decreases.

Transmission spectrum

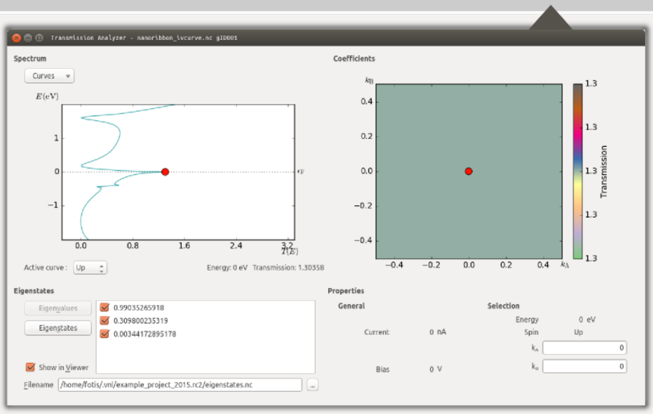
Select the **TransmissionSpectrum** item on the LabFLoor, and click the **Transmission Analyzer** plugin to visualize it. The left-hand plot shows the transmission spectrum, while the right-hand plot shows an interpolated contour plot of the transmission coefficients in reciprocal space.

Example: **Study the transmission spectrum.**


Example Project (/home/fotis/.vnl/example_project_2015.rc2) - Virtual NanoLab
File Projects View Tools Windows Help
Project Files
Builder
Scripter
Custom Scripter
Editor
Jobs
Viewer
Filter...
Text
File types
Name
GaAs100.nc
benzene_molecularenergyspectrum.nc
benzene_molecularenergyspectrum.py
fermi_surface.log
fermi_surface.py
graphene_phonons.nc
graphene_phonons.py
h2o_on_mgo_reactionbarrier.nc
h2o_on_mgo_reactionbarrier.py
lifepo4.log
lifepo4.py
mdtrajectory_cnt.nc
mdtrajectory_cnt.png
mdtrajectory_cnt.py
nanoribbon_ivcurve.nc
nanoribbon_ivcurve.py
nanoribbon_phonontransmission.nc
nanoribbon_phonontransmission.py
project.example
LabFloor
Group by
Filename
nanoribbon_ivcurve
nanoribbon_ivcurve.nc gID000
$I(V)$
nanoribbon_ivcurve.nc gID002
$T(\mathbf{k},\epsilon)$
nanoribbon_ivcurve.nc gID001
1
Transmission Analyzer...
Effective mass analyzer...
Electron Phonon Coupling Analyzer...
Export...
Grid Operations...
IV-Plot...
Inelastic Transmission Spectrum Analyzer...
Kinetic Monte Carlo Analyzer...
Local Bandstructure Analyzer...
MD Analyzer...
Markov Chain Analyzer...
Molecular Energy Spectrum Analyzer...
Movie Tool...
Optical Spectrum Analyzer...
Projected Local Density of States...
Text Representation...
Vibration Visualizer...
Viewer...

The left plot in the transmission analyzer shows the transmission spectrum. In the **Curves** menu, you can select which spin components are shown. For non-polarized calculations, the default is to show the transmission spectrum of the sum of the up and down channel.

With the mouse you can select a point (E,T(E)) in the active curve. This k-dependent transmission for this energy is then illustrated in the right plot.



Active curve : Up

Energy: 0 eV Transmission: 1.30358

☑ 0.99035265918
☑ 0.309800235319
☑ 0.00344172895178

☑ Show in Viewer

Filename /home/fotis/.vnl/example_project_2015.rc2/eigenstates.nc

Current 0 nA

Bias 0 V

Energy 0 eV

Spin Up

k_A 0

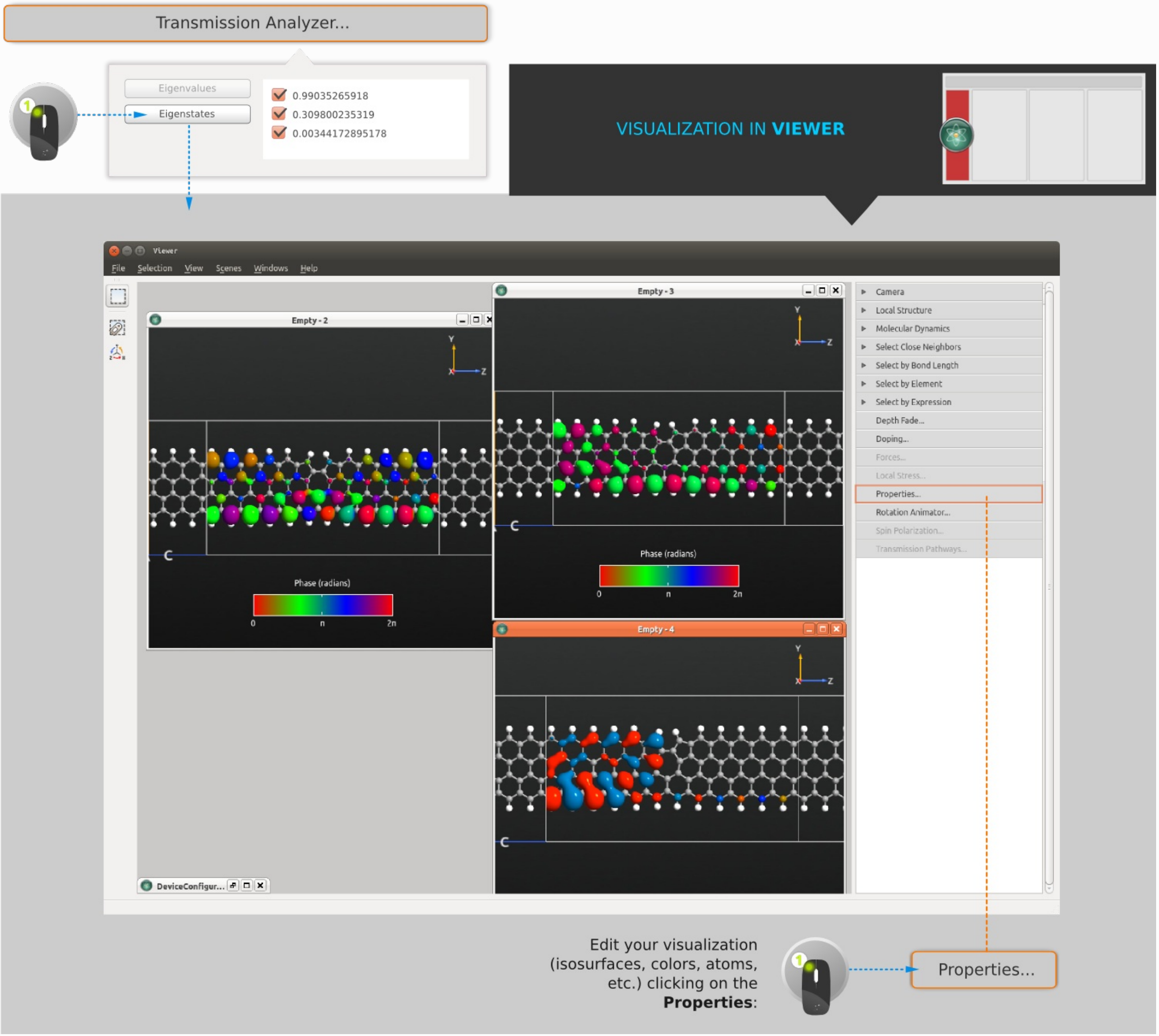
k_B 0

Transmission eigenstates

For a given energy and k-point, the electron transport can be described in terms of **transmission eigenstates**. The transmission eigenvalues and corresponding eigenstates are conveniently calculated from the Transmission Analyzer, see the image above.

Select E=0 in the T(E) plot, and the (0,0) point in the plot of transmission coefficients. Then click the **Eigenvalues** button, and the plugin will report the corresponding eigenvalue(s). Tick all three eigenvalues.

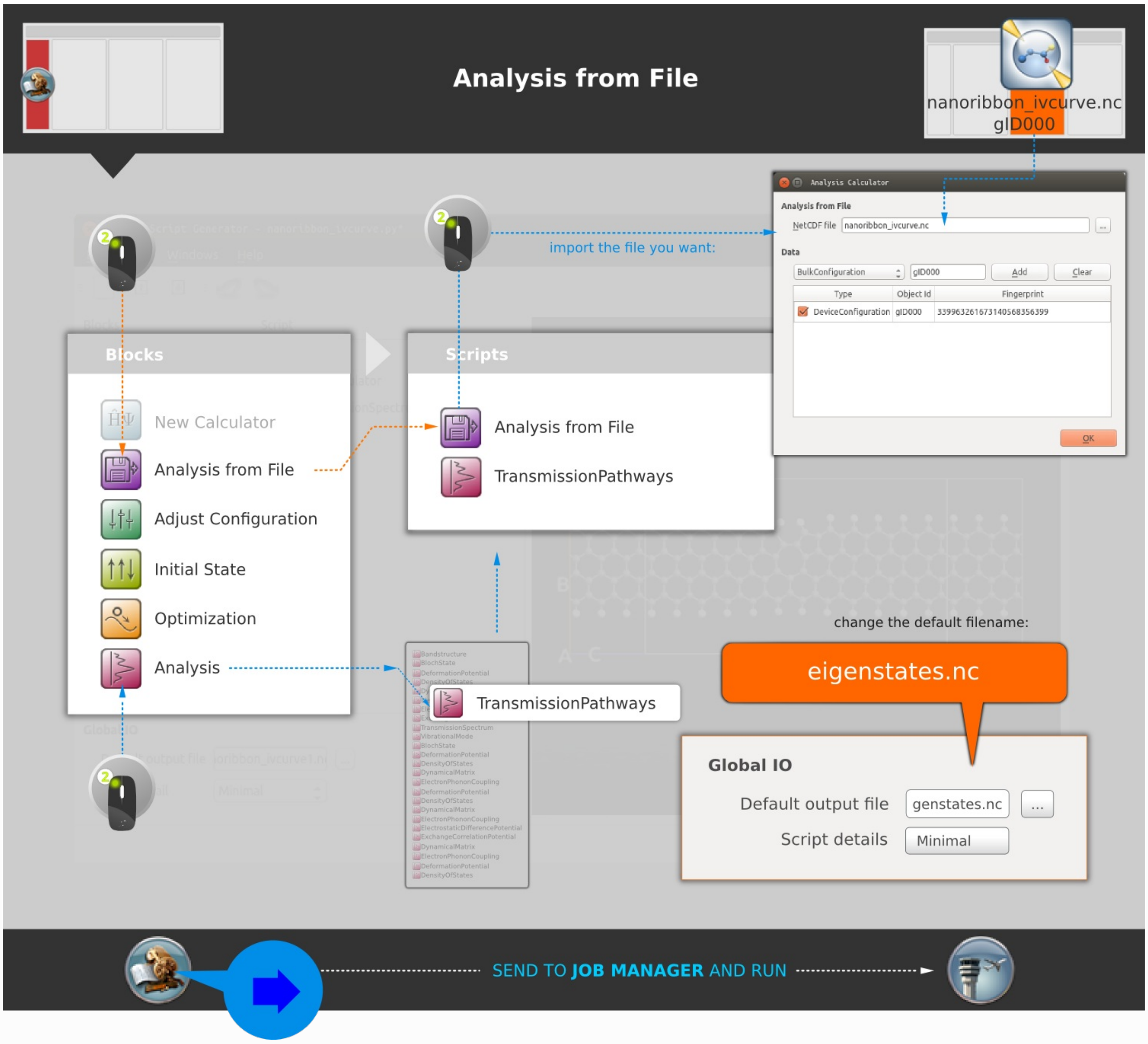
Next, click the **Eigenstates** button to calculate the eigenstates for the ticked eigenvalues. A dialog then asks how you want to visualize the eigenstates – select *isosurface*. In the **Viewer** window that pops up, open the *Properties* menu, and set the isovalue to 0.1. You should then see the following visualizations of the transmission eigenstates through the graphene nanoribbon device.



Edit your visualization (isosurfaces, colors, atoms, etc.) clicking on the **Properties**:

Transmission pathways

Finally, you should calculate the transmission pathways at the Fermi level. This allows you to visualize the pathways for electron transmission from the left to the right electrode. Use **Analysis from File** to load the DeviceConfiguration stored in the `nanoribbon_ivcurve.nc` data file.



Analysis Calculator

Analysis from File

NetCDF file: nanoribbon_ivcurve.nc

Data

BulkConfiguration | gID000 | Add | Clear

Type	Object Id	Fingerprint
☑ DeviceConfiguration	gID000	33996326167314056835639 9

OK

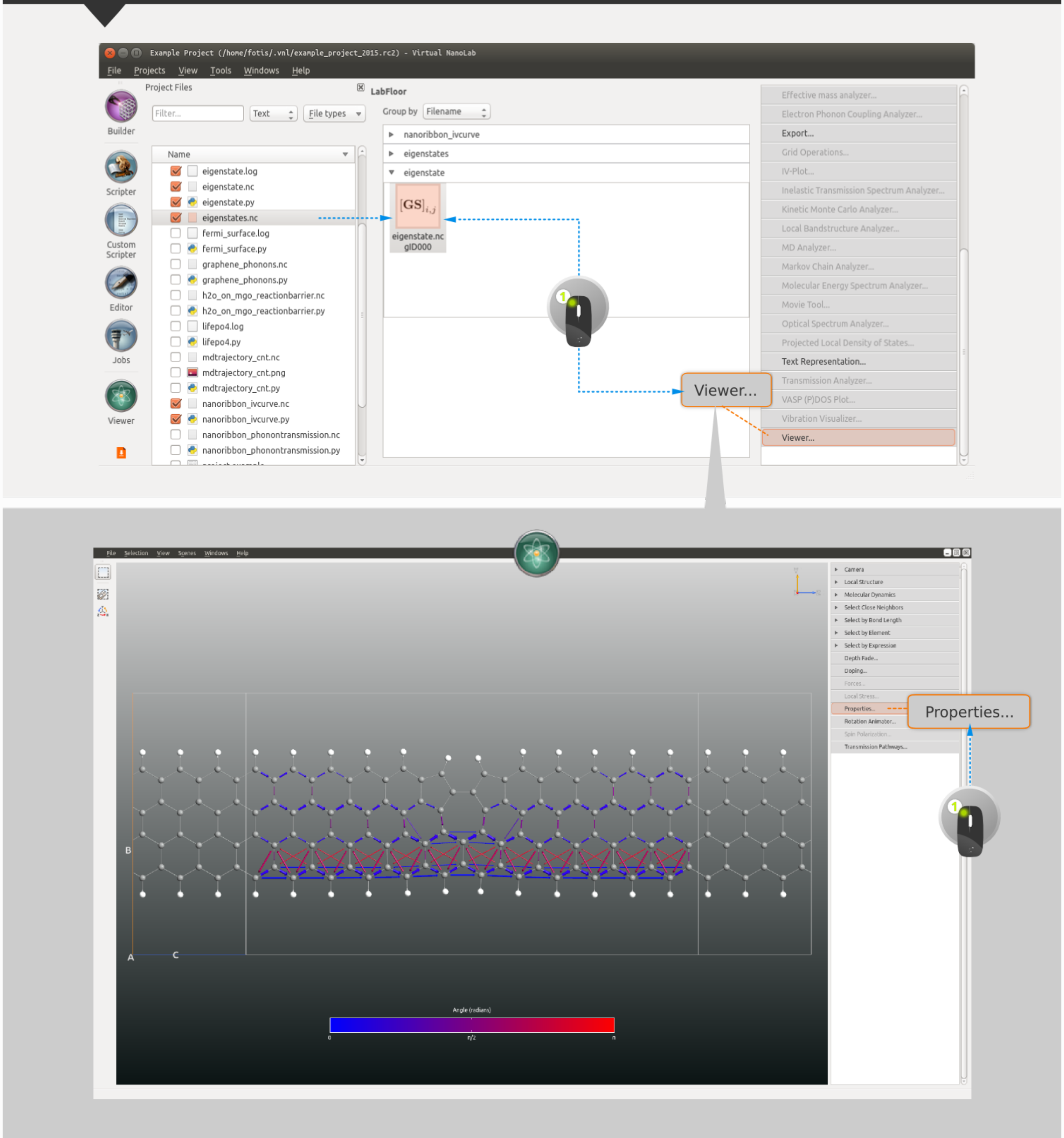
Global IO

Default output file: genstates.nc

Script details: Minimal

The **TransmissionsPathways** data item, $[GS]_{i,j}$, should now be available on the LabFloor. Visualize it using the **Viewer**, as illustrated below. The volume of each arrow indicates the magnitude of the local transmission between each pair of atoms, while the arrow and color indicate the direction of the electron flow.

Example: **Transmission Pathways.**

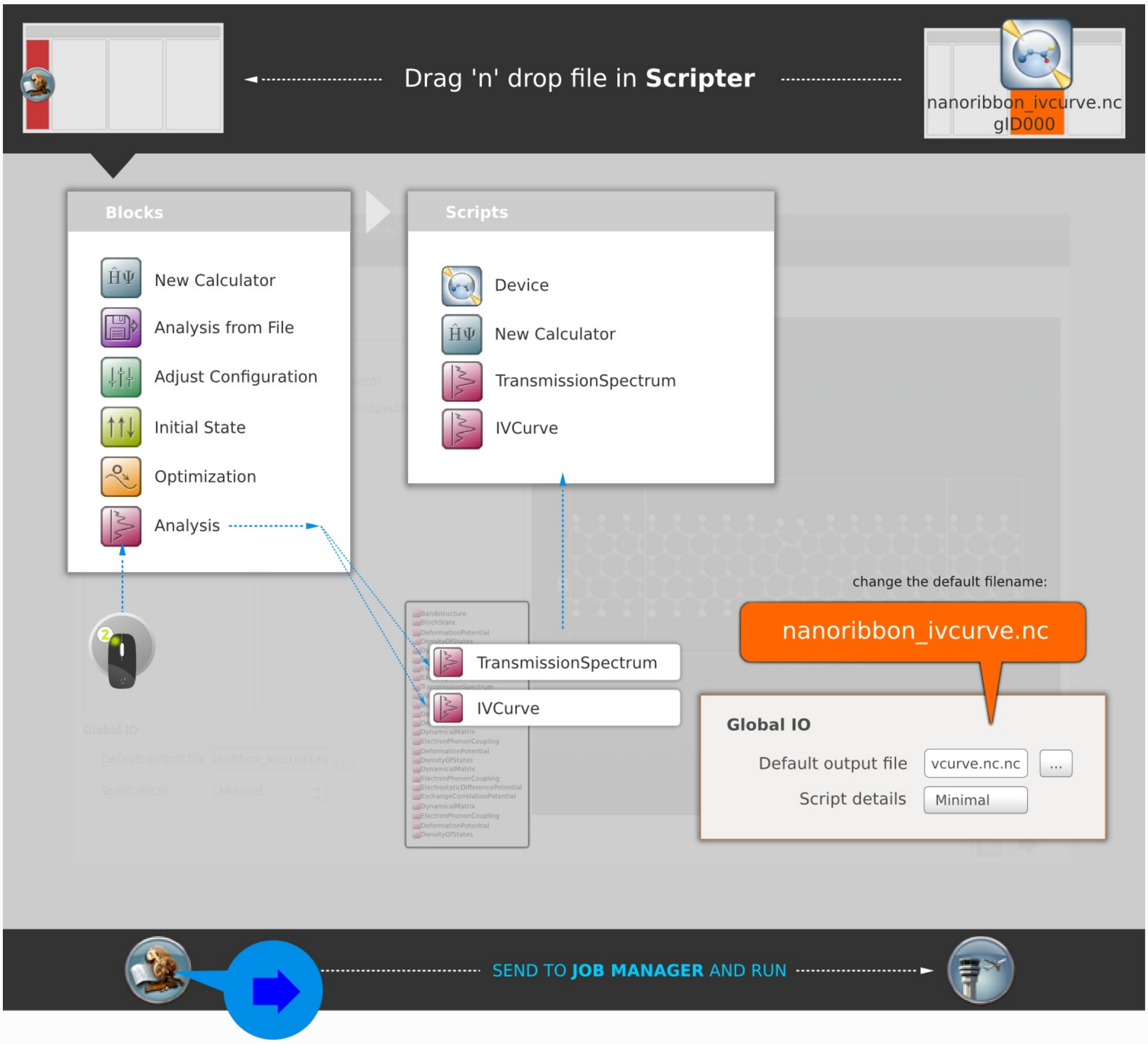


It is not hard to deduce the positions of the atoms, but you can also add them explicitly by dropping the DeviceConfiguration onto the Viewer canvas. However, the bonds hide the arrows, so you may want to adjust the radius of the atoms. Use the *Properties* menu for this.

Performing the device calculations

The transport analysis in the previous section was done using pre-calculated data. However, it is fairly simple for you to redo the device calculation and compute the post-SCF analysis data on your own. As an example, you will here set up an ATK Python script that first runs a semi-empirical Hückel calculation for the graphene nanoribbon, and then computes the transmission spectrum and I–V curve.

On the LabFloor, select the DeviceConfiguration previously saved in the `nanoribbon_ivcurve.nc` data file. Drop it on the **Scripter**, and notice that both the device configuration and saved calculator are added to the script. Then add analysis blocks to the script and set the default output filename.



change the default filename: nanoribbon_ivcurve.nc

Global IO

Default output file: vcurve.nc.nc

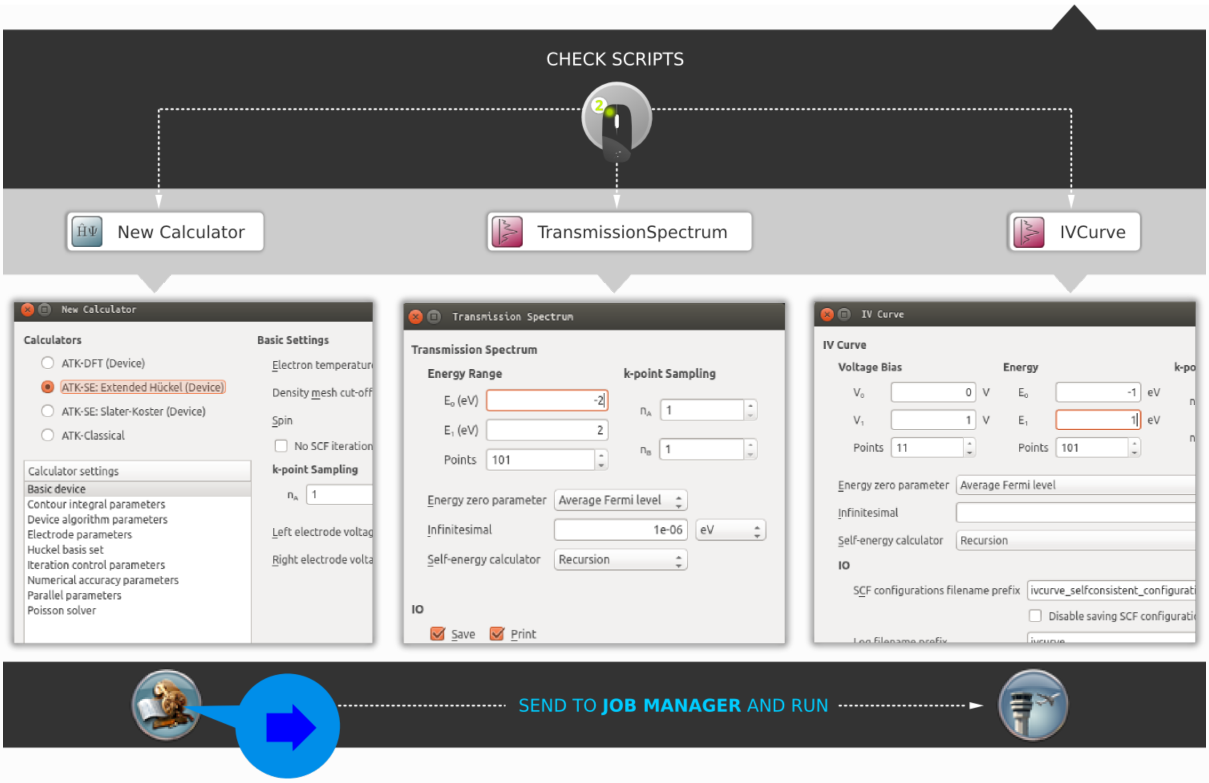
Script details: Minimal

Tip

You can use the mouse to reorder the inserted script blocks if needed. You can also delete blocks using Delete on your keyboard.

Next, you should make sure that each script block is set up properly:

- Double-click the **New Calculator** block to open it and check the calculator parameters. In particular, make sure to choose the **ATK-SE** engine and the **Extended Hückel** method.
- No options should be changed in the TransmissionSpectrum block, but you should open it to see the available options.
- In the IVCurve block, setup a voltage **bias window** of 0 to 1 V, and make the **energy window** span from -1 to +1 eV.
- Remember to save the script.



New Calculator

Calculators

○ ATK-DFT (Device)
● ATK-SE: Extended Hückel (Device)
○ ATK-SE: Slater-Koster (Device)
○ ATK-Classical

Calculator settings
Basic device
Contour integral parameters
Device algorithm parameters
Electrode parameters
Huckel basis set
Iteration control parameters
Numerical accuracy parameters
Parallel parameters
Poisson solver

Basic Settings

Electron temperatur
Density mesh cut-off
Spin
□ No SCF iteration

k-point Sampling

n_A 1

Left electrode voltag
Right electrode volta

Transmission Spectrum

Transmission Spectrum

Energy Range

E_0 (eV) -2
E_1 (eV) 2
Points 101

k-point Sampling

n_A 1
n_B 1

Energy zero parameter: Average Fermi level
Infinitesimal: 1e-06 eV
Self-energy calculator: Recursion

IO

☑ Save ☑ Print

IV Curve

IV Curve

Voltage Bias

V_0 0 V
V_1 1 V
Points 11

Energy

E_0 -1 eV
E_1 1 eV
Points 101

k-po

Energy zero parameter: Average Fermi level
Infinitesimal:
Self-energy calculator: Recursion

IO

SCF configurations filename prefix: ivcurve_selfconsistent_configurati
□ Disable saving SCF configurati
Log filename prefix: ivcurve

You are now ready to run the calculations – you can use the **Job Manager** for this. *Please note that the job takes several hours on a standard desktop computer.* You may bring this down to perhaps one hour if you run the job in parallel on multiple CPUs, either on your local machine or on a remote computing cluster.

Please proceed to the next chapter:

Example 3: Building a graphene nanoribbon device.

Previous

Next