

# Atomistix ToolKit Tutorial

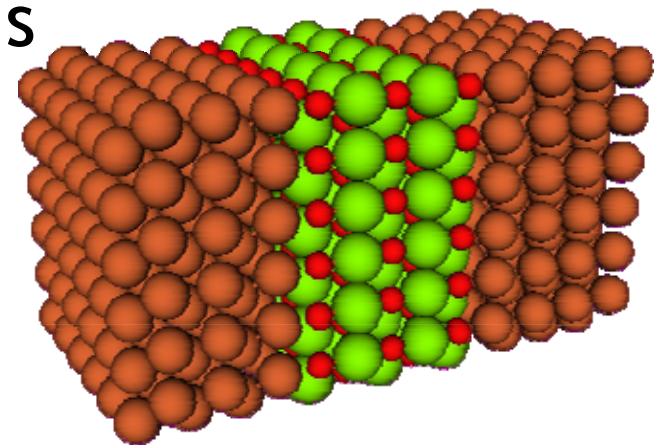
## Part 2: Analysis

Quantum  
Wise

## Outline

---

- ❖ Part 1: Fundamentals
- ❖ Part 2: Analysis
  - » Basic considerations
  - » Common analysis  
(bulk, molecules, two-probe)
  - » Advanced two-probe analysis
  - » Plotting





## What is “analysis”?

---

- ❖ Previous tutorial: converging the self-consistent field calculation
- ❖ That is just a (necessary) step on the way to the end results = some observable quantity
  - » Current
  - » Forces
  - » Total energy
  - » Spectrum, band structure
  - » Electron density
  - » etc...
- ❖ Analysis: computing these quantities
- ❖ Most analysis can be set up using VNL
  - » NanoLanguage scripting gives further, more advanced analysis possibilities, including customized plotting

## Two paths to the end results

---

- ❖ Input for analysis calculation is always a self-consistent field (SCF) calculation in the form of an “SCF object”, either
  - » Returned by  
`executeSelfConsistentCalculation(...)`
  - » Restored from NetCDF file by  
`restoreSelfConsistentCalculation(NetCDFFile)`
- ❖ Given these two options, the analysis statements can either be
  - » placed at the end of the main SCF script, or
  - » run as a separate script



Always make  
sure a NetCDF  
is created!



## Output from analysis functions

---

### ❖ Return from analysis functions

» Various kinds of **objects**: single value(s), array, grid

- Member functions (methods) like `.energies()` extract the data
- Use the `.toArray()` method to extract the data values from e.g. grids and arrays for custom manipulations in NanoLanguage



» Data is typically a **PhysicalQuantity** (or array thereof)

- Hence always with explicit units (same as for input values)
- Use the `.inUnitsOf()` method to extract the value

❖ All analysis quantities can and should be stored in a **VNL file** for keeping + plotting in Virtual NanoLab

❖ Function `nlPrint()`: nicely formatted printing to log file (stdout)

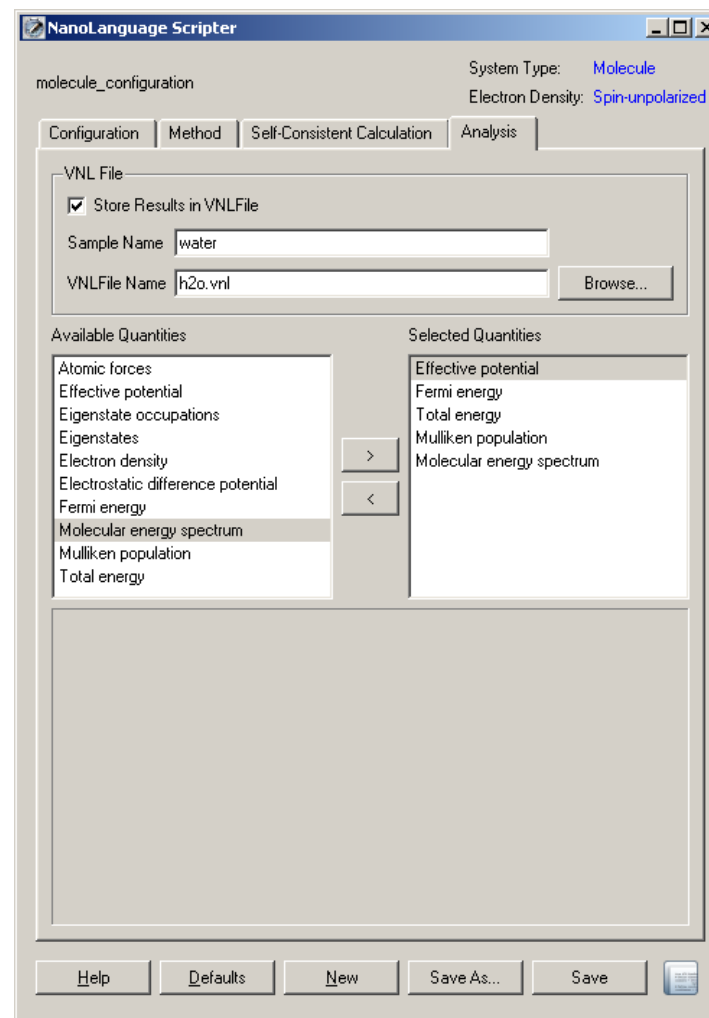
» Frequently used in VNL-produced scripts

## Setting up analysis from VNL

❖ **Analysis** tab in *NanoLanguage Scripter* gives access to all basic analysis options

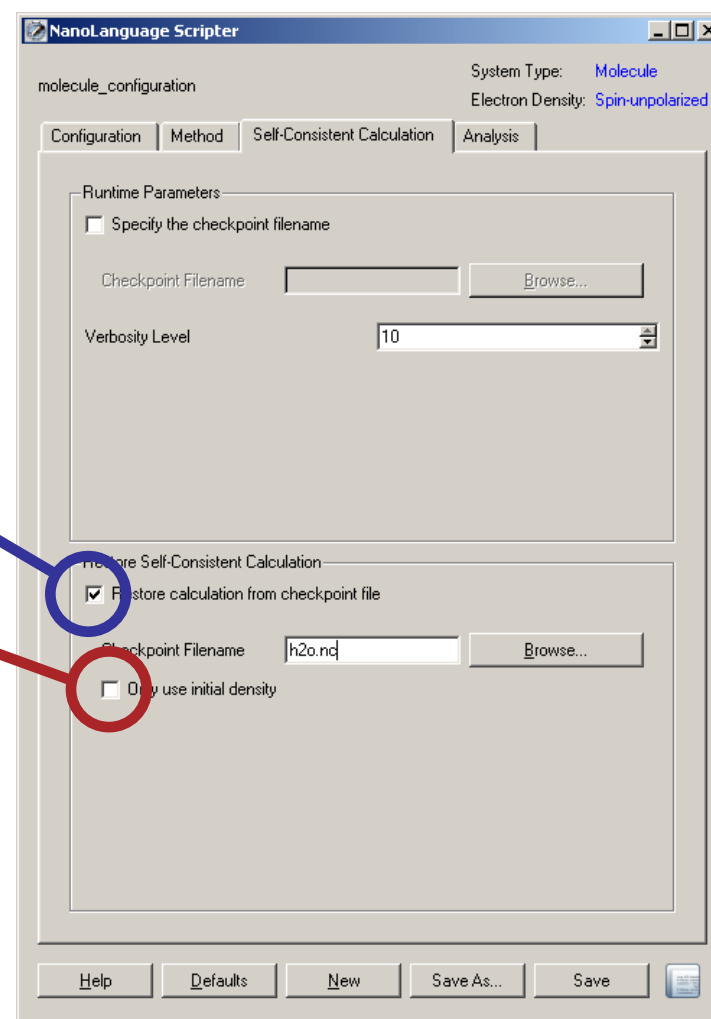
- » Simple clicking to add/remove analysis options
- » Use as a template generator for more advanced analysis

❖ Don't forget to store the results in a VNL file!



## Restoring a NetCDF file from VNL

- ❖ Drop SCF script on the *NanoLanguage Scripter* to import geometry and method
  - » Needed to determine system type, spin-polarization, etc
- ❖ Tick “Restore calculation from checkpoint file”
- ❖ Untick “Only use initial density”
  - » Tick to initialize e.g. a finite bias calculation from a converged zero-bias density, or to restart a non-converged calculation
- ❖ Set up analysis on **Analysis** tab as on previous slide



## Basic analysis

---

- ❖ Available for all system types; no input parameters
  - » Mulliken populations
    - **Tip:** Use verbosity 10 for orbital-resolved populations
  - » Forces
    - Analytic (Hellmann-Feynman)
    - Can be used for customized relaxation algorithms
  - » Total energy
    - Relatively expensive, but parallelized
    - **Tip:** Use verbosity 10 to resolve individual contributions
  - » 3D grids:
    - Electron density
      - A few lines of code make it possible to visualize spin-polarization density  $n_{\uparrow}(r)-n_{\downarrow}(r)$
    - Effective potential
    - Electrostatic difference potential
      - Solution to the Poisson equation for the "difference density" = the total electron density minus the neutral atom electron density



The image displays three windows from the NanoLanguage Scripter software:

- Top Right (molecule\_configuration):** Shows the analysis setup. The "Analysis" tab is active. Under "Selected Quantities", several items are listed with colored circles: Atomic forces (yellow), Eigenstate occupations (red), Effective potential (blue), Electron density (orange), Electrostatic difference potential (green), Fermi energy (purple), Total energy (brown), Mulliken population (pink), and Molecular energy spectrum (magenta).
- Middle (Script Editor):** Shows the NanoLanguage script. Colored circles on the left of the script correspond to the quantities in the analysis window. The script includes functions like `calculateAtomicForces`, `calculateEigenstateOccupations`, `calculateEffectivePotential`, `calculateElectronDensity`, `calculateElectrostaticDifferencePotential`, `calculateFermiEnergy`, `calculateTotalEnergy`, `calculateMullikenPopulation`, and `calculateMolecularEnergySpectrum`. Each function call is followed by `nPrint()` and `addToSample()` to store results in the VNL file.
- Left (LogWindow: NanoLanguageScript [Finished]):** Shows the output log. Colored circles on the left of the log correspond to the quantities in the analysis window. The log contains numerical data for Atomic Forces, Fermi energy, Total Energy, Potential Energy, Ion Kinetic Energy, Charge, Band Structure Energy, Kinetic and NL Energies, ENA, Eions, Electron kinetic energy, Electrostatic energy, Exchange-corr. energy, Mulliken Population, and Energy Spectrum.

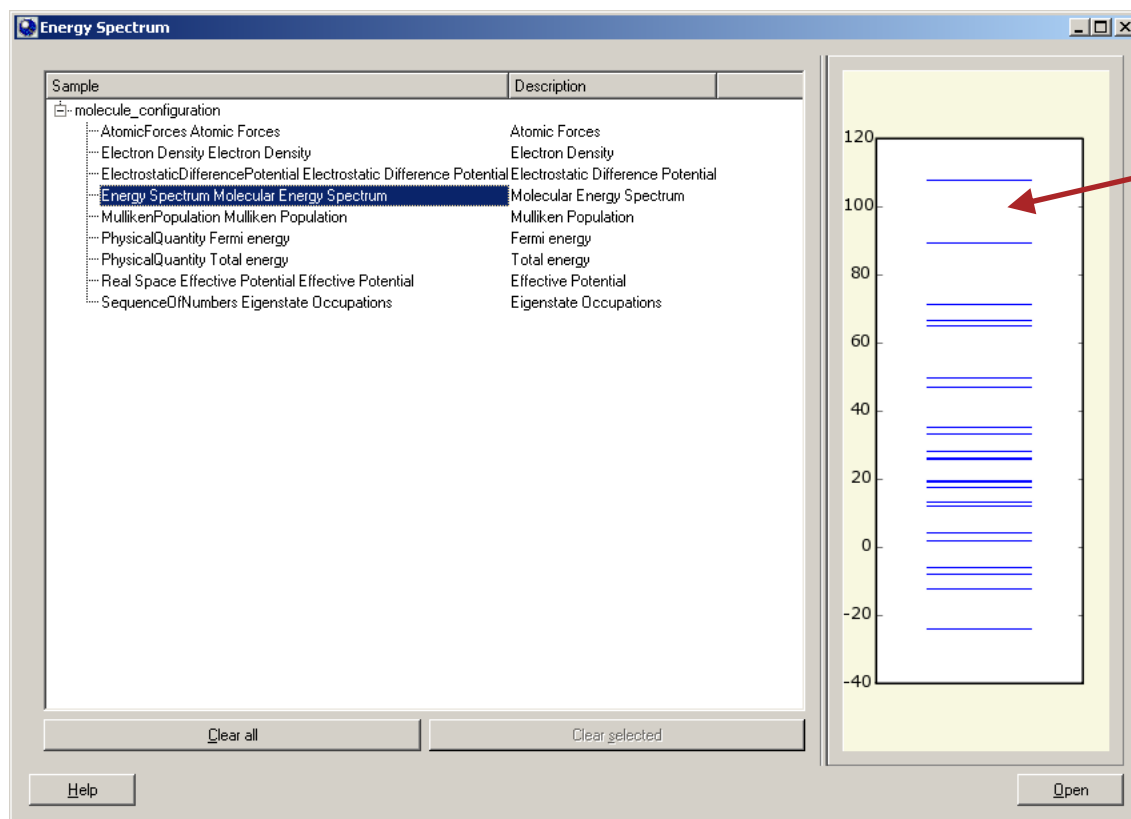
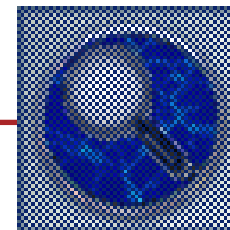
### Three representation of analysis:

1. **Up right:** Setting up the analysis in VNL
2. **Middle:** The resulting NanoLanguage script
3. **Left:** The output log

The colored circles mark the corresponding analysis quantities in input, script, and output.

Note especially the use of the `nPrint()` function to produce the output in the log file, and the way each calculated quantity is stored in the **VNL file**, with a label.

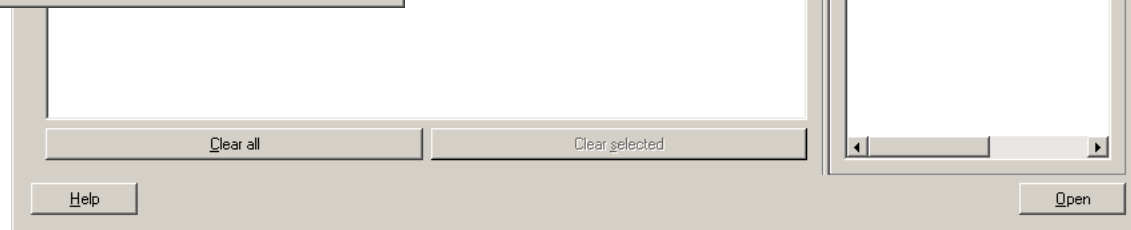
## VNL file imported to the Result Browser



Right-click to Export/Print

Right-click to Mark/Copy

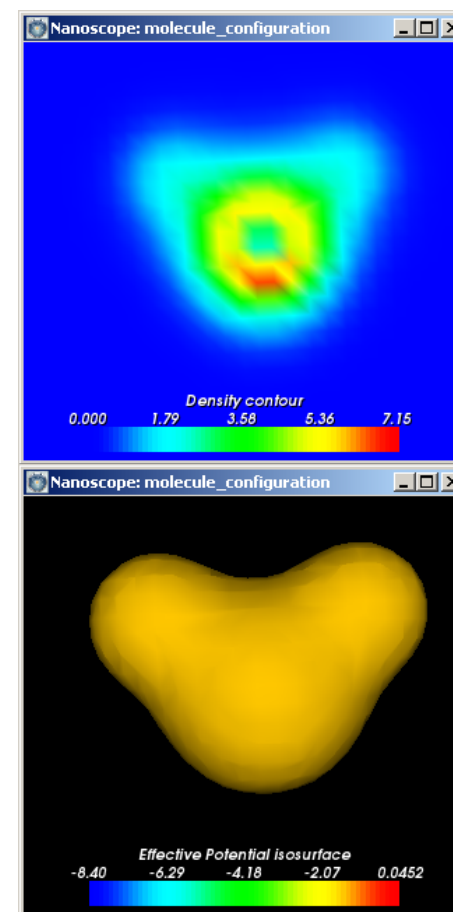
Drop file (or drag from Result Browser) on Nanoscope to visualize the 3D grids!



## 3D plotting in the Nanoscope

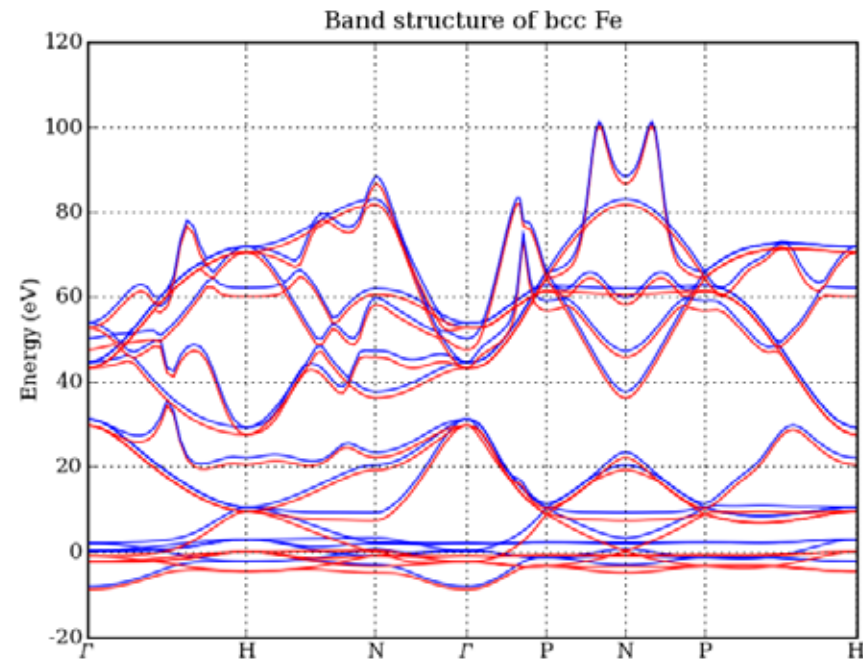


- ❖ VNL: 3 types of plots
  - » Contour plot
  - » Isosurface
  - » Volume plot
- ❖ A variety of color & appearance options
  - » Colorbar, different maps, “zoom” in color
- ❖ Export
  - » Variety of bitmap formats (BMP, PNG, EPS, etc)
  - » Print directly
- ❖ Custom plotting/data manipulation:
  - » Extract values in NanoLanguage via `object.toArray()`
  - » Grid dimensions can also be extracted; contact QuantumWise for more information



## Band structure

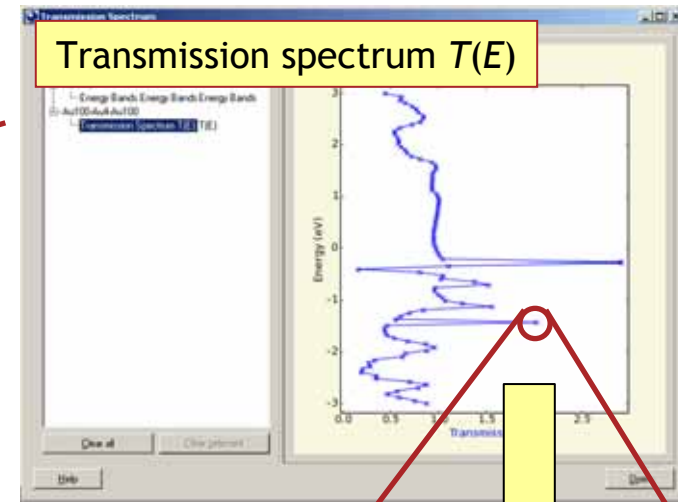
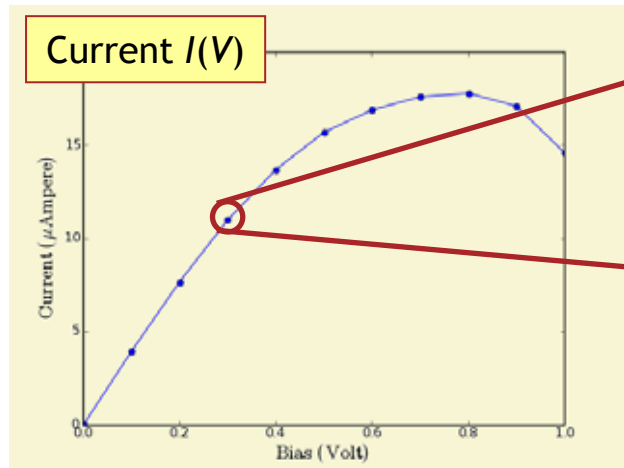
- ❖ ATK function to calculate “bandlines” is a low-level work-horse routine
  - » A higher-level band structure calculator is available on the Forum
  - » Becoming part of the standard product in 2009
- ❖ For density of states, see Forum!
- ❖ Compute effective masses (and more...!) in NanoLanguage



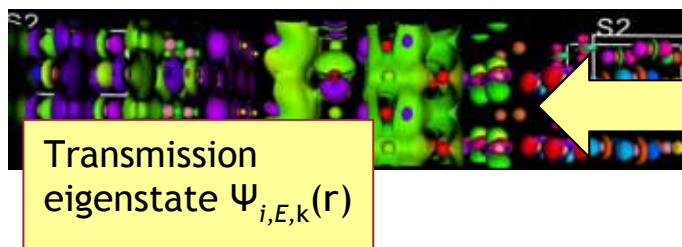
*E=0 corresponds to the Fermi level.*

# Decomposition and analysis of the current

$$I = \frac{2e}{h} \int_{-\infty}^{\infty} T(E) [n_F(E - \mu_L) - n_F(E - \mu_R)] dE$$



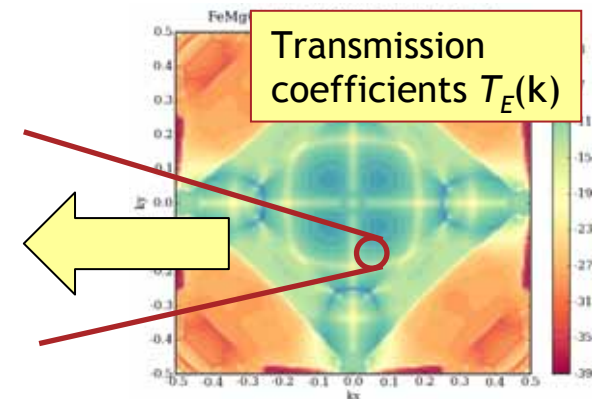
$$T(E) = \sum_{\mathbf{k}} T_E(\mathbf{k}) w_{\mathbf{k}}$$



Transmission eigenvalues  $\tau_i$

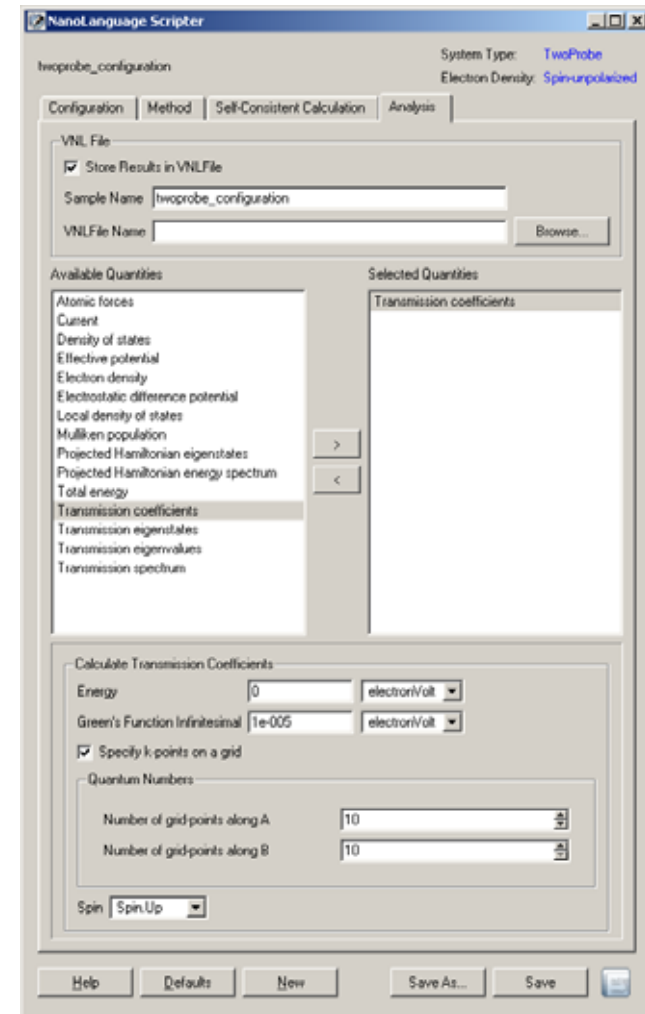
Eigenvalue	0	0.17252
Eigenvalue	1	0.00267

$$T_E(\mathbf{k}) = \sum_i \tau_i(E, \mathbf{k})$$



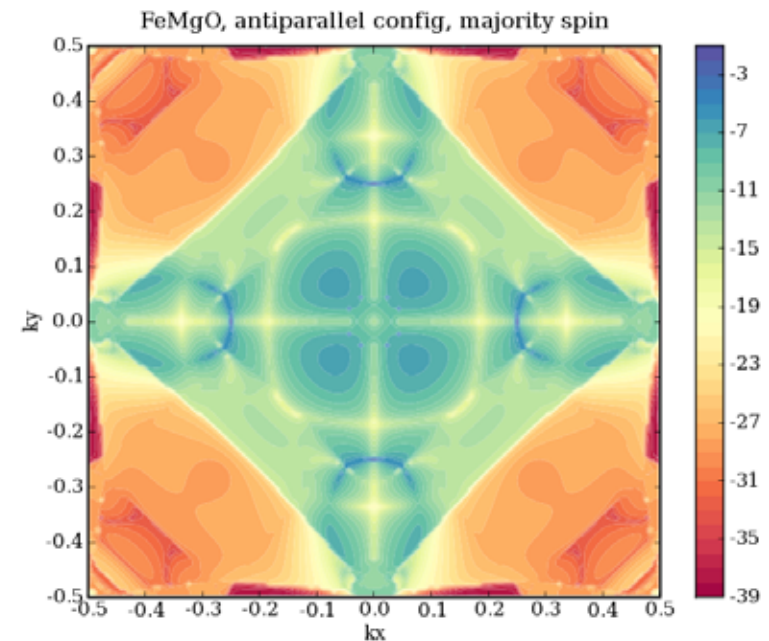
## Transmission coefficients

- ❖ Input parameters:
  - » energy  $E$
  - » quantum numbers [ $k_1, k_2, k_3, \dots$ ]
  - » (Spin.Up | Spin.Down)
- ❖ In VNL, k-points can be set up as
  - » individual points
  - » regular grid (not Monkhorst-Pack)
- ❖ k-points are always given in fractional coordinates ( $-0.5 \leq k_{A,B} \leq 0.5$ )
- ❖  $T_E(k) > 1$  if several channels contribute to the transport at the same energy
- ❖ Parallelized over the k-points
  - » Linear scaling



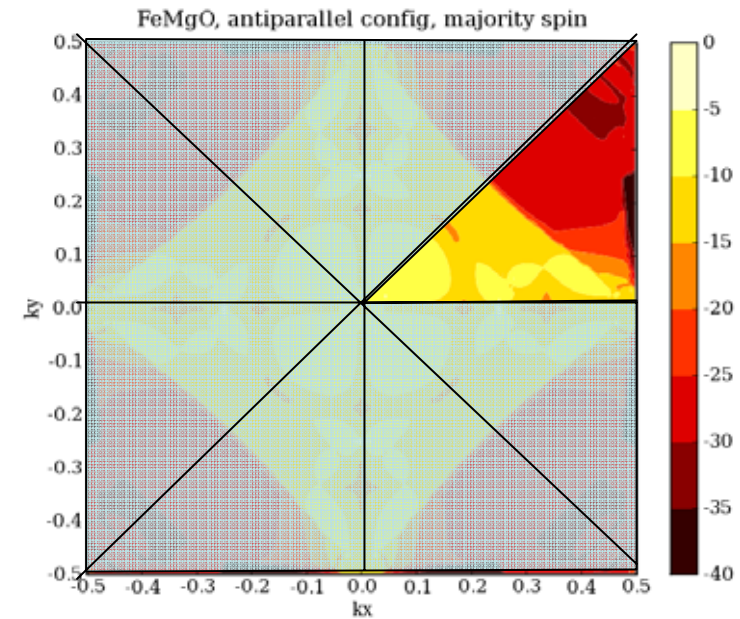
## Plotting transmission coefficients

- ❖ Coefficients computed on a regular grid can easily be visualized using ATK
  - » ATK has an built-in plotting engine!
    - Matplotlib, to be specific <http://matplotlib.sourceforge.net>
    - Fully customizable 2D plots, Matlab style
- ❖ Some simple scripting needed (5-10 lines)
  - » Script available from the Forum



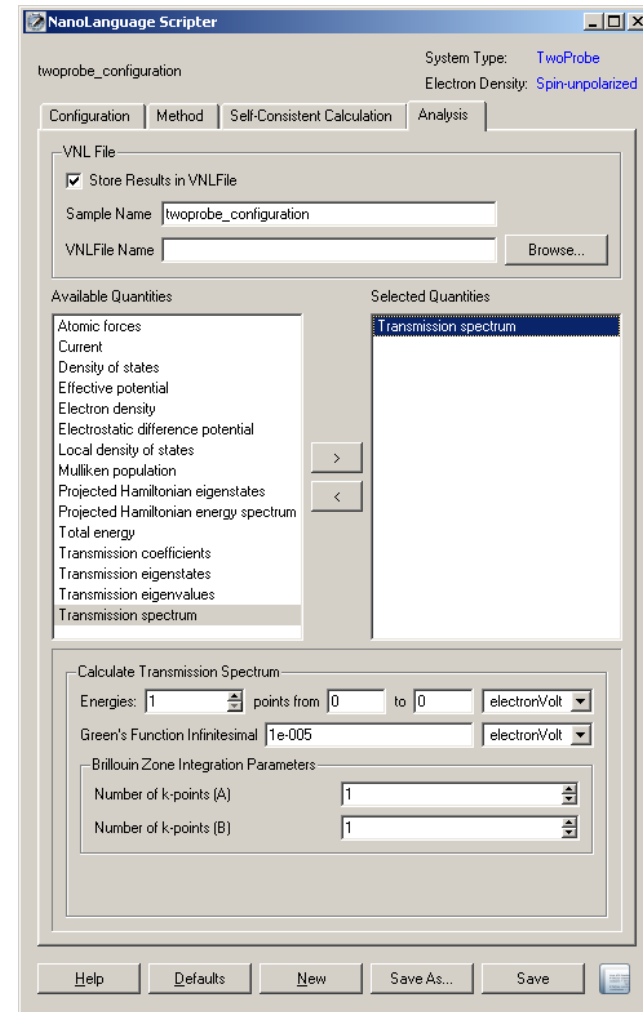
## Symmetries reduce calculation time

- ❖ Time-reversal symmetry  $T_E(\mathbf{k})=T_E(-\mathbf{k})$  is always used in ATK to reduce calculation time
- ❖ In some cases, the transmission coefficients have additional symmetries
  - » Example: Fe/MgO/Fe, bcc [100]; unique values only in first octant
  - » Calculation time for current, etc, can be reduced by a factor of 4 compared to using standard ATK
  - » ... or a factor 8, if you want to plot the transmission coefficients, compared to a brute force script!
- ❖ Custom script solution required
  - » Script available from the Forum



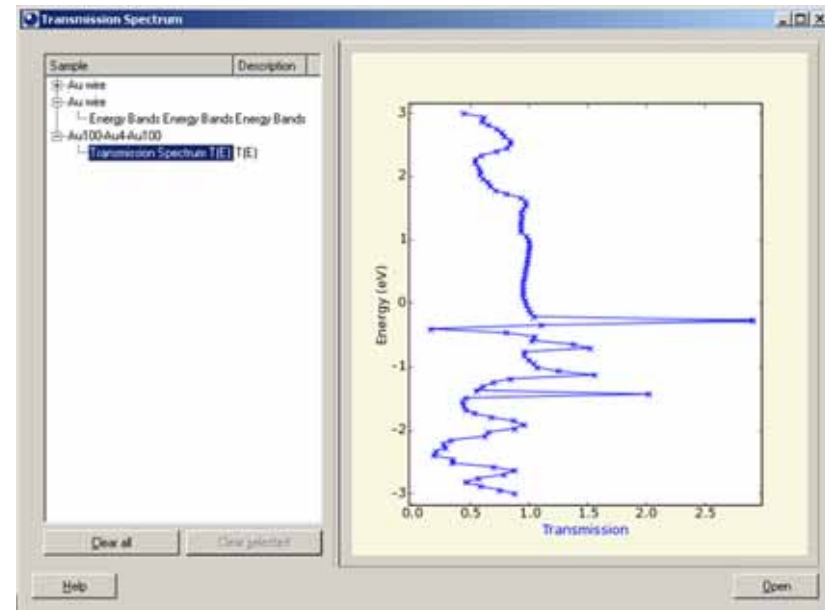
## Transmission spectrum

- ❖ Input parameters:
  - »  $E_{\min}$ ,  $E_{\max}$ , number of points in between
  - » Monkhorst-Pack grid (NAXNB)
- ❖ Required number of k-points depends varies a lot
  - » Always 1x1 for 1D systems
  - » Roughly 10x10 for “typical” 3D systems - **but must be checked for convergence!**
  - » 400x400 for minority spin in Fe/MgO/Fe!!!
  - » Unrelated to the number of k-points used for the SCF loop; depends on the central region, surface states, and the nature of the transport mechanism
- ❖ Parallelized over k-points
  - » Linear scaling
- ❖ **Density of states:** completely analogous input and output



## Plotting transmission spectrum

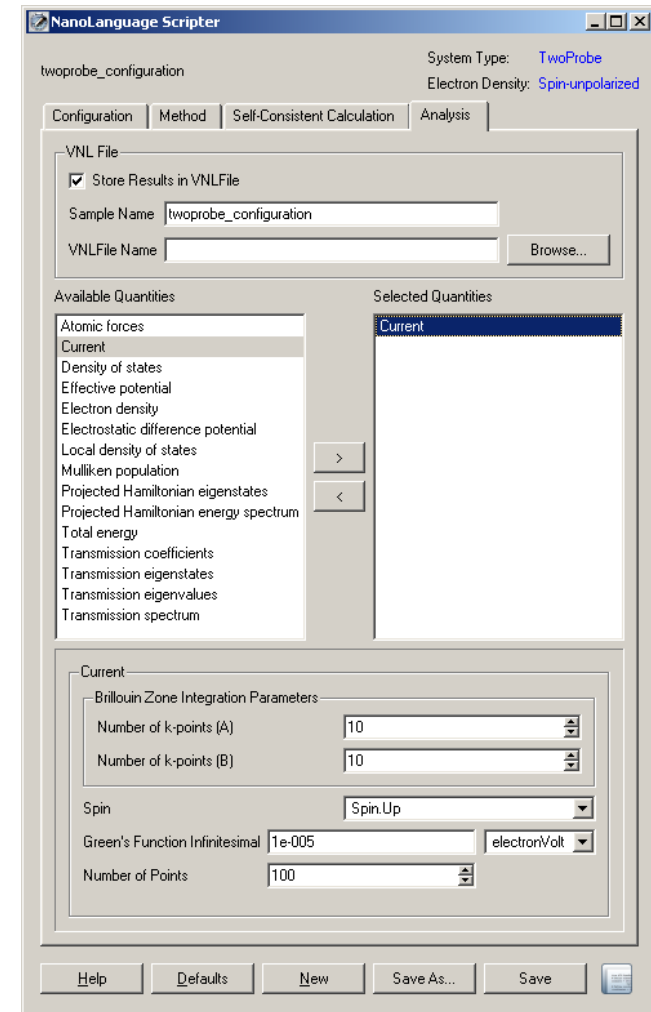
- ❖ Calculator function returns  $T(E)$  for both spins (if spin-polarized)
- ❖ Store in VNL file and plot in **Result Browser**
  - » Open/drop VNL file, view spectrum in right panel
  - » Can be zoomed, exported, printed
- ❖ Extract values, plot yourself in GNUPlot, Excel, ...
  - » `nlPrint(T)`, or
  - » `T.energies()` & `T.coefficients()`
- ❖  $T_E > 1$  if several channels contribute to the transport at the same energy



*Transmission spectrum for a 4-atom Au monowire between [100] Au surfaces*

## Current

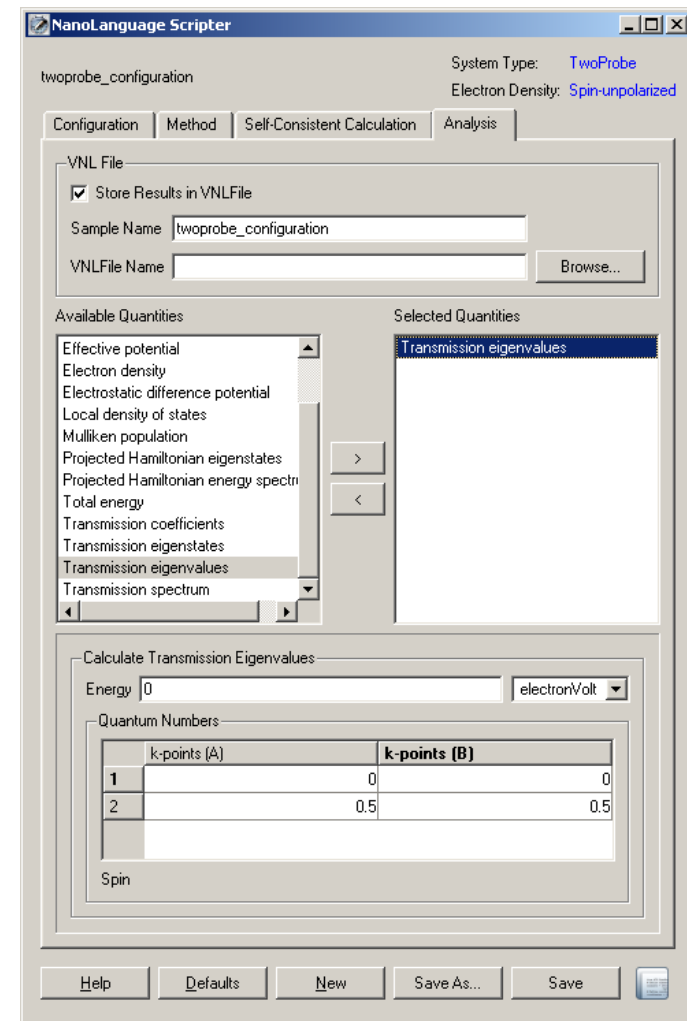
- ❖ Input parameters:
  - » Monkhorst-Pack grid
  - » Number of points
  - » Spin component
- ❖ Integration window = bias  $\pm 5kT_{\max}$
- ❖ Required number of points depends on smoothness of transmission spectrum in the bias window
  - » Possible to develop an adaptive integration routine to automatically handle this
  - » Would also provide error estimate
- ❖ Sign convention: positive current = flow of *charge* from left to right (i.e. electrons flow right to left)
  - » Obtained when the bias on the left electrode is *higher* than on the right one
- ❖ Parallelized over k-points
  - » Linear scaling



## Transmission eigenvalues

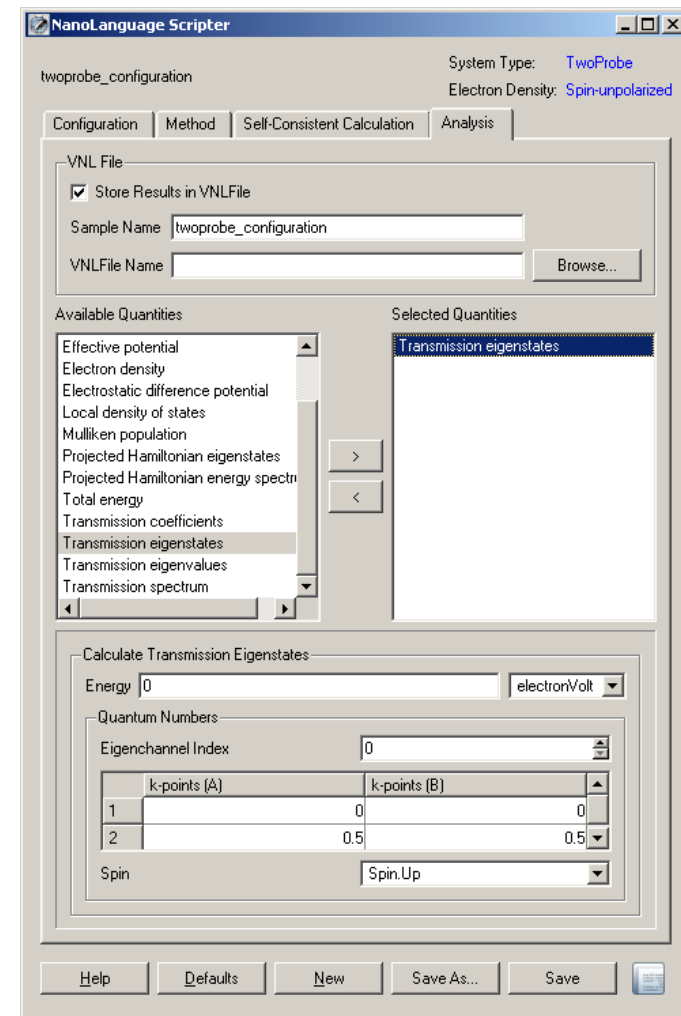
- ❖ The **transmission eigenvalues** are found by diagonalizing  $t^\dagger t$  where  $t$  is the transmission matrix  $t_E(k)$
- ❖ Each eigenvalue  $\tau_n$  ( $0 \leq \tau_n \leq 1$ ) represents a transmission eigenchannel and measures its contribution to the total transmission coefficient  

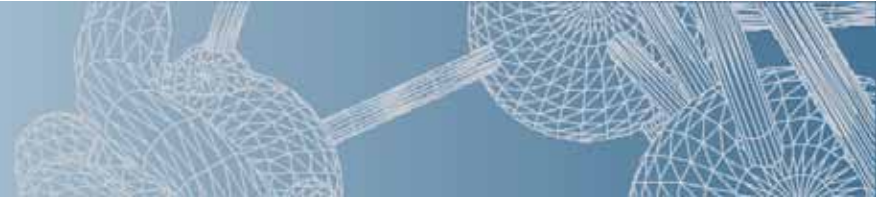
$$T_E(k) = \text{Tr}[t^\dagger t] = \sum_n \tau_n$$
- ❖ The **transmission matrix**  $t_E(k)$  is computed using scattering states (not Green's functions)
- ❖ K-points are given in reciprocal coordinates ( $-0.5 \leq k \leq 0.5$ )



## Transmission eigenstates

- ❖ Project transmission eigenchannel on the real-space orbitals
  - » “Eigenchannel Index” corresponds to the transmission eigenvalues
  - » K-points are given in reciprocal coordinates ( $-0.5 \leq k \leq 0.5$ )
- ❖ 3D grid, use VNL to visualize
- ❖ Illustrates, in a sense, the “pathway” of the electrons tunneling through the central region





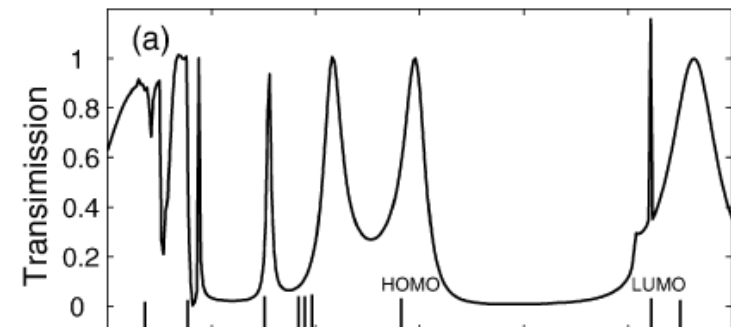
## Additional two-probe analysis

---

- ❖ Surface density of states (DOS)
- ❖ Spatially resolved DOS
  - » DOS projected on the real-space orbitals
- ❖ Conductance (do-it-yourself)
  - »  $I/V$
  - »  $dI/dV$
  - » Zero bias:  $G_0 T(E=\mu)$ , where  $G_0 = 2e^2/h = 77.48 \mu\text{S}$   
Exact in the limit of zero temperature
- ❖ Prototype: Linear response current
  - » Quick method to get an I-V curve
  - » Assumes transmission spectrum for small voltages is independent of bias
  - » `import ATK.Future`

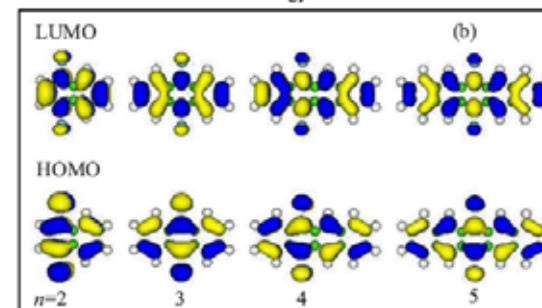
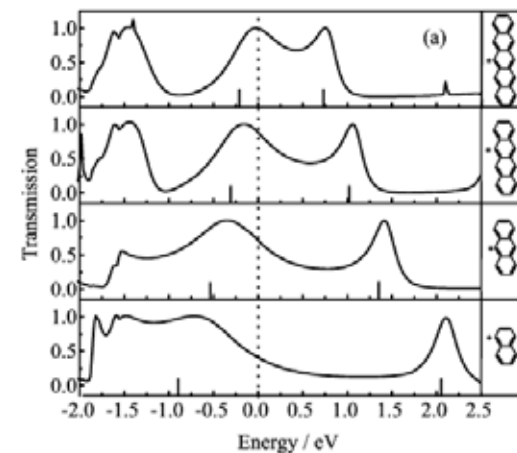
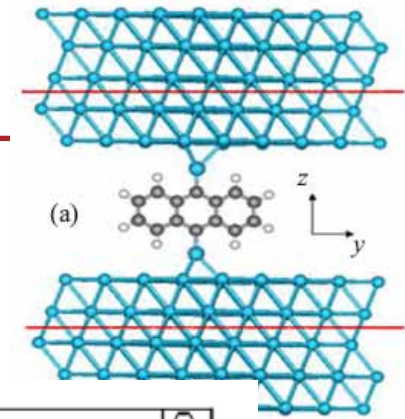
## MPSH analysis - eigenvalues

- ❖ Molecular Projected Self-Consistent Hamiltonian
- ❖ Molecular calculation for all atoms in central region minus specified “surface atoms” (truncated Hamiltonian)
  - » Primary result is a molecular eigenvalue spectrum
- ❖ Accounts for the coupling to the electrodes, but not the self-energy coupling
  - » Real part of self-energy coupling shifts transmission peaks relative to MPSH levels
  - » Imaginary part of self-energy broadens transmission peaks (hybridization with continuum)
- ❖ The MPSH peaks can often be correlated to transmission peaks
  - » Or dips, in case of localized orbitals!
- ❖ Makes most sense when a “molecule” can naturally be defined in the central region
  - » Can also be a stump of a nanotube or graphene ribbon, e.g. segment with dopant
  - » Compare to HOMO/LUMO states in the isolated molecule



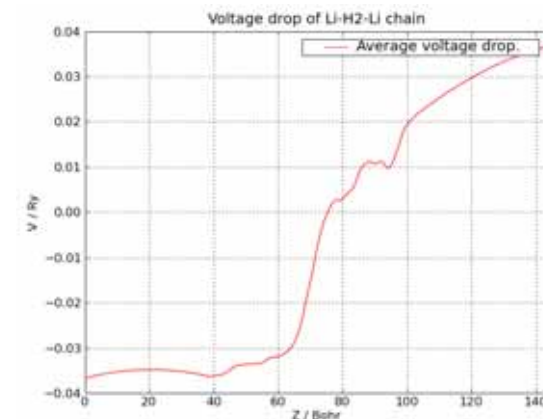
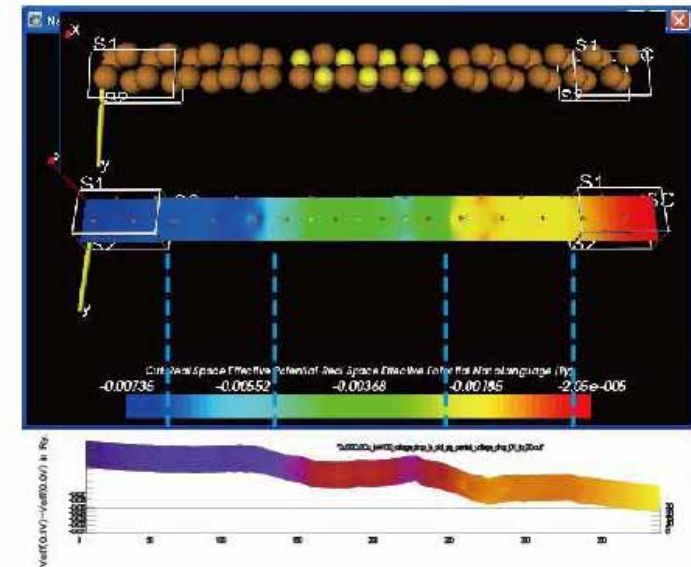
## MPSH analysis - eigenstates

- ❖ Each eigenstate can be projected on real space in 3D
  - » Similar to usual molecular eigenstates (i.e. wavefunctions)
- ❖ Visualize in VNL
- ❖ Caution: uses a lot of memory...
  - » Don't put too many in the same VNL file



## Voltage drop

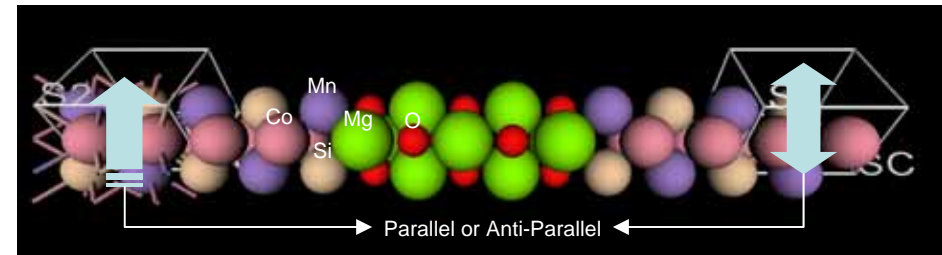
- ❖ Visualize where the quantum resistance appears
- ❖ Must use electrode constraint **DensityMatrix** (introduced in 2008.10) for reliable results
- ❖ Visualize as 3D in VNL or use NanoLanguage to average over the XY plane



## Collinear spin-torque transfer (STT) and TMR

- ❖ `import ATK.Consortia`
- ❖ Heusler alloys suggested as new materials for MTJs  
A. Thomas *et al.*, APL 89, 012502 (2006)
- ❖ Example:  $\text{Co}_2\text{MnSi}$   
(semi-metallic; majority spin metallic, minority semiconducting)

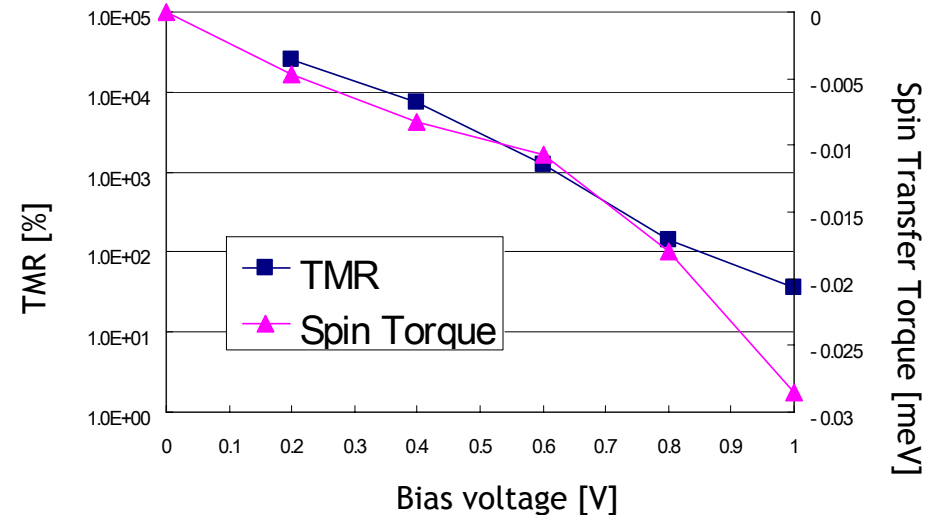
Results courtesy of Cybernet Systems



$$\text{TMR} \equiv \frac{J_{\uparrow\uparrow}^{\text{total}} - J_{\uparrow\downarrow}^{\text{total}}}{J_{\uparrow\downarrow}^{\text{total}}} \times 100$$

$$T_{\parallel} \equiv \frac{\hbar}{400e} (J_{\uparrow\downarrow}^{\text{total}} P_{\uparrow\downarrow} - J_{\uparrow\uparrow}^{\text{total}} P_{\uparrow\uparrow})$$

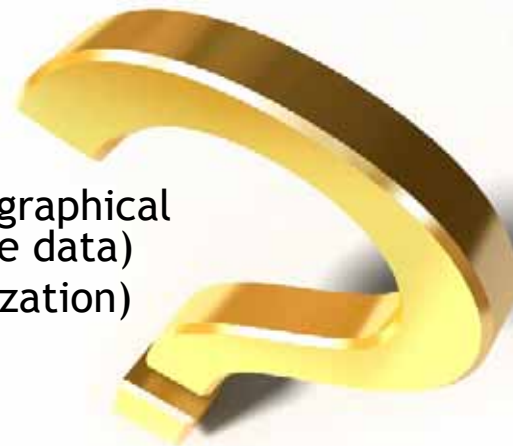
$$P_{\uparrow(\downarrow)} \equiv \frac{J_{\uparrow(\downarrow)}^{\text{majority}} - J_{\uparrow(\downarrow)}^{\text{minority}}}{J_{\uparrow(\downarrow)}^{\text{majority}} + J_{\uparrow(\downarrow)}^{\text{minority}}} \times 100$$



## Help us improve ATK and VNL!

---

- ❖ The analysis output are the true results of a calculation with ATK, and therefore of fundamental importance for the successful use of the product
- ❖ QuantumWise is therefore particularly interested in getting input from users about **analysis**
  - » New quantities that can be computed
  - » Data handling
    - Storage
    - Export
  - » Plotting
    - Data management
    - Plot types (different graphical representations of the data)
    - Plot options (customization)
    - Export options



Let us know what *you* need!

## New features delivered via the Forum

- ❖ Analysis extensions
  - » Bulk DOS
  - » Projected DOS (PDOS)
  - » Symmetry reduction of k-point sampling for transmission & DOS
  - » Effective mass and band gap for semiconductors
  - » I-V curves
  - » Atomization energy
  - » Current density
- ❖ Geometry setup and manipulation
  - » Graphene nanoribbons
  - » Nanotubes
  - » Relaxation under bias
- ❖ Plotting tools
  - » K-point resolved transmission
  - » Band structure
  - » Voltage drop
- ❖ Miscellaneous
  - » XYZ file manipulations
  - » Extracting data from VNL files
  - » Converting old ATK files to NanoLanguage

