Hands-on session: brief introduction into AITRANSS

by Alexei Bagrets

(alexej.bagrets@kit.edu)

Inst. of Nanotechnology (INT) & Steinbuch Centre for Computing (SCC)
@ Karlsruhe Institute of Technology (KIT), Germany

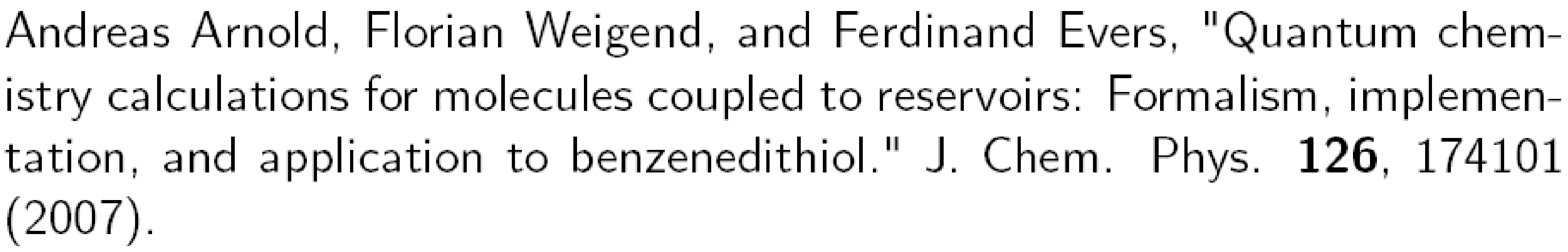
What is AITRANSS ?

- AITRANSS (*ab initio* transport simulations) : project under continuous development at Inst. of Nanotechnology (INT) of the Karlsruhe Inst. of Technology (KIT) since 2002

- when combined with FHI-aims, is a postprocessor module: enables calculation of electron transmission across molecular junctions based on a Landauer formalism

- advanced options to appear in future releases (spin-polarized transport, out-of-equilibrium response, etc.)

- when using AITRANSS, please cite the reference:



Andreas Arnold, Florian Weigend, and Ferdinand Evers, "Quantum chemistry calculations for molecules coupled to reservoirs: Formalism, implementation, and application to benzenedithiol." J. Chem. Phys. **126**, 174101 (2007).

Source of code & supporting materials

- Code is placed at the subdirectory `aitranss/` of the FHI-aims release

- `aitranss/` contains

  `source/`: with Fortran90 code & `Makefile`

  `tcontrol.script/`: contains a script to create a mandatory input file `tcontrol`

  `electrodes.library/`: library of fcc Au clusters modeling electrodes

  `examples/`: archetypical examples (benzene-dithiol, alkyl wire) with input & (partially) output files

- Documentation: look at README file and Chapter 5 of FHI-aims manual

Compiling the AITRANSS module

- Prerequisite: Fortran90/95 compiler, compiled LAPACK & BLAS libraries (e.g. Intel's MKL)
- Use an example `Makefile` found at `/source`, and adjust it accordingly
- Encouraged is to use compiler options aka "-openmp" to build a multithreaded code based on OpenMP directives
- For now (hands-on session) use precompiled binary file(s) and scripts: see next page for details

Links to binary files, scripts, visualization tool

- To be able to execute precomplied binary files of FHI-aims and aitranss, and related scripts, please, create symbolic links (or, alternatively, copy files)

```
mkdir bin ; cd bin
ln -s /home/bude/k0019299/bin/aims.scalapack.mpi.x .
ln -s /home/bude/k0019299/bin/aitranss.x .
ln -s /home/bude/k0019299/bin/tcontrol.aims.x .
ln -s /home/bude/k0019299/bin/aims2pdb .
```

- Edit your `.bashrc` file, add a line: `export PATH=~/bin:$PATH`

- To be able to use a simple structure visualization tool Rasmol (supporting a PDB format), create a symbolic link (or copy the binary file)

```
cd bin
ln -s /home/bude/k0019299/bin/rasmol .
```

Hands-on session: collection of examples

- Please, create a working directory and copy a collection of examples, e.g.

```
mkdir aitranss.tutorial
cd aitranss.tutorial
cp -r /home/bude/k0019299/aitranss.tutorial/examples .
```

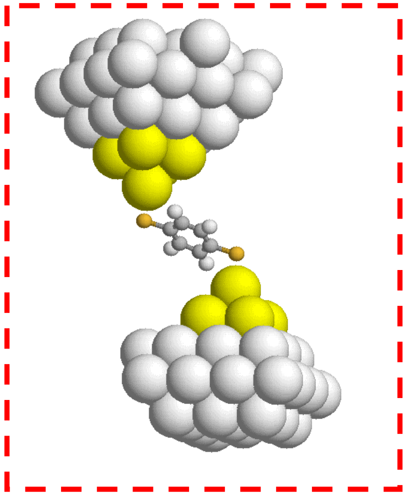
Collection contains three directories:

- Simplest example, benzene-dithiol (au-c6h6-au): takes ~ 25 min. wall time
- Further example, alkyl wire (au-alkyl_chain-au): takes ~ 50 min. wall time
- Exercises (exercise.alkyl.wires): 7 different molecular structures, to be tested during free time ☺

Algorithm behind AITRANSS

- Do DFT for eM with a quantum chemistry package (here FHI-aims) using local basis set $|\tilde{\mu}\rangle = |\mathbf{x}_n, l\rangle$
- Ask the package to provide the overlap matrix $S_{\mu\nu} = \langle\tilde{\mu}|\tilde{\nu}\rangle$
- Orthogonalize the basis set, $|\mu\rangle = \sum_\nu |\tilde{\nu}\rangle S^{-1/2}_{\nu\mu}$
- Use KS states $|p\rangle = \sum_\mu |\tilde{\mu}\rangle c_{\mu p}$ and energies ϵ_p to construct

$$\langle\mu|H_{eM}|\nu\rangle = \sum_{\mu'\nu' p} S^{1/2}_{\mu\mu'} c_{\mu' p}\, \epsilon_p\, c^T_{p\nu'} S^{1/2}_{\nu\nu'}$$



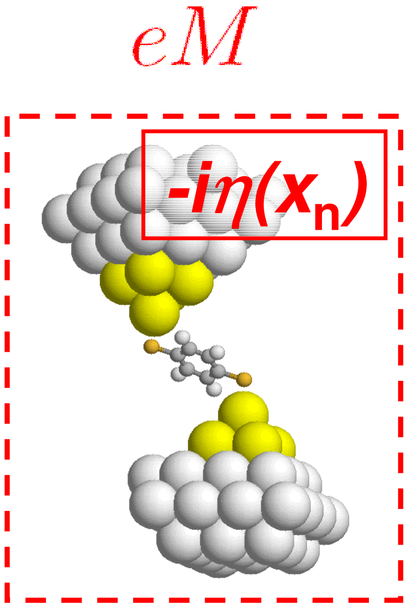
eM

see next page for further steps ...

Algorithm behind AITRANSS (continue)

- Make the choice for the left/right local leakage function $i\eta(\mathbf{x}_n)$ (look for remarks in the library files and use default set of parameters!), construct

$$\langle\mu|\Sigma_{L/R}|\nu\rangle = -i\eta(\mathbf{x}_n)\delta_{\mu\nu}$$



- Calculate transmission as

$$T(E) = \mathrm{Tr}\left[\Gamma_L G(E)\Gamma_R G^\dagger(E)\right]$$

where

$$\langle\mu|G(E)|\nu\rangle = \langle\mu|[E - H_{eM} - \Sigma_L - \Sigma_R]^{-1}|\nu\rangle$$

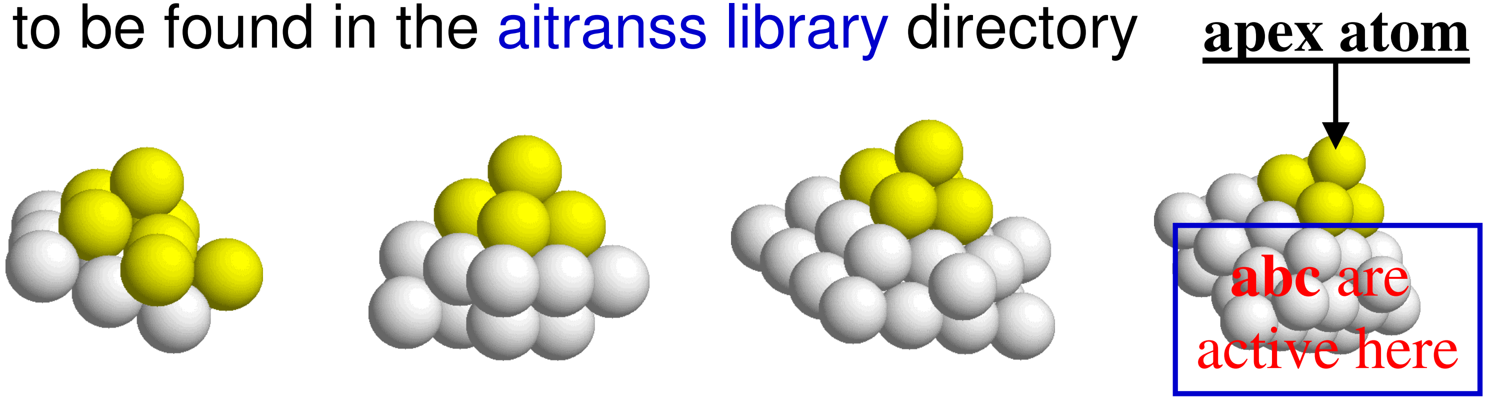
$$\Gamma_{L/R} = \frac{1}{2i}(\Sigma^\dagger_{L/R} - \Sigma_{L/R})$$

For that task, eigenstate representation of the complex-valued operator $(H_{eM} + \Sigma_L + \Sigma_R)$ is used.

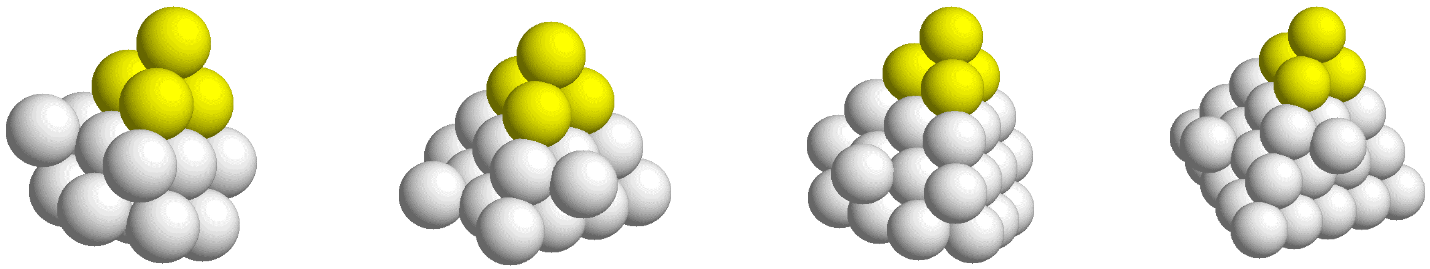
First step: prepare your molecular structure

- use your favorite visualization and modeling tool and build up an “extended molecule” by linking your molecule via anchoring groups to the apex atoms of Au clusters
- “chaotic” clusters are to be found in the aitranss library directory

Examples of fcc(001) clusters:



Examples of fcc(111) clusters:



- For your convenience, all examples considered in this tutorial already contain extended structures (not isolated molecules)

Simple visualization tool: Rasmol

- To visualize your structure, you may use an old-fasion but simple tool called Rasmol (http://www.rasmol.org/software/RasMol_2.7.5/)
- To convert aims-coordinate file to the PDB format, type

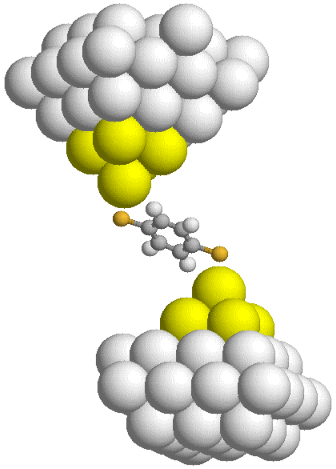
  `aims2pdb geometry.in` (you‘ll get file called `coord.pdb`)
- To obtain a picture shown below, (i) type

  `rasmol coord.pdb` (two windows will pop up) ;

  (ii) Choose in the graphical window Display | Ball & Stick ;

  (iii) Switch to the control window and type in the command line:

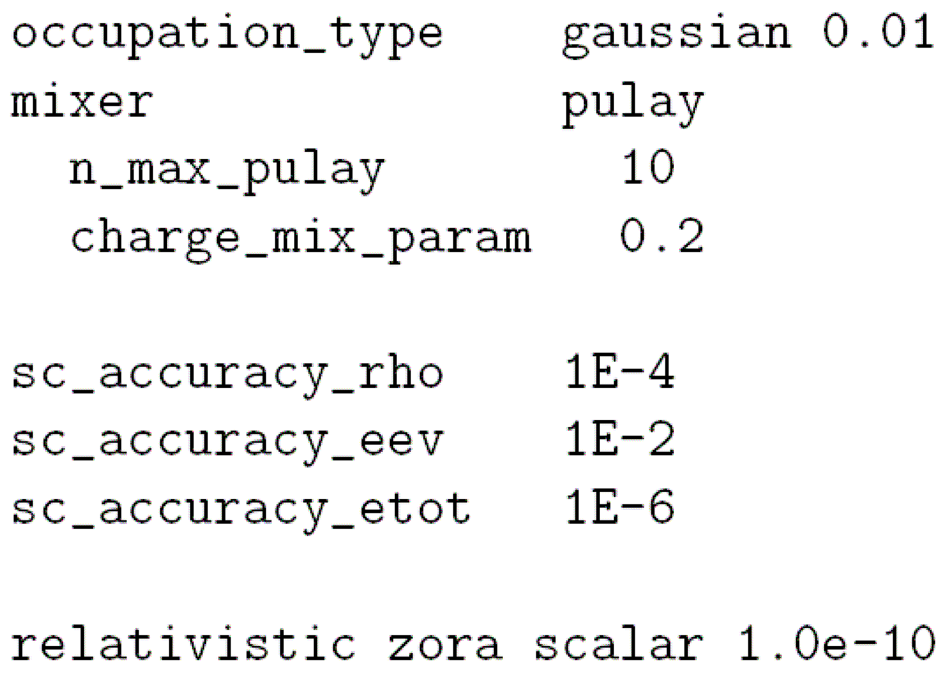
```
background white
select 1, 2
spacefill 380
select 1
color yellow
select 2
color white
```



Hint: by clicking on atom, you find its index (look at the control window)

Second step: FHI-aims run (input)

- FHI-aims should be invoked with a cluster type (non-periodic) settings
- Include a line `output aitranss` into `control.in` file
- Following parameters are recommended



```
occupation_type    gaussian 0.01
mixer              pulay
  n_max_pulay        10
  charge_mix_param   0.2

sc_accuracy_rho    1E-4
sc_accuracy_eev    1E-2
sc_accuracy_etot   1E-6

relativistic zora scalar 1.0e-10
```

Second step: FHI-aims run (output)

- Invoke FHI-aims, e.g.:

```
nohup mpirun –np 4 aims.scalapack.mpi.x > aims.dft.out&
```

- After FHI-aims completed its job, following ASCII files will be generated:

`basis-indices.out` : indexing of basis functions

`omat.aims` : contains overlap integrals

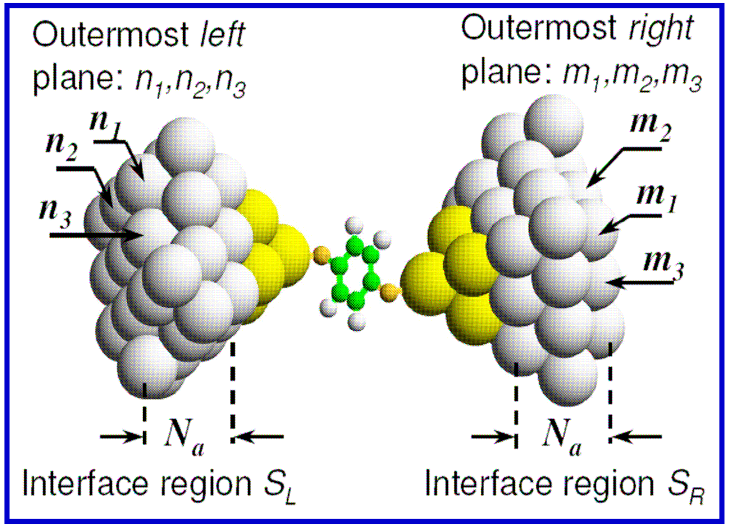
`mos.aims` : contains Kohn-Sham molecular orbitals & energies of the “extended molecule”

- AITRANSS needs also a mandatory file `tcontrol`, which has to be created with help of a script `tcontrol.aims.x`

What is needed to run AITRANSS job?

- Files `geometry.in, basis-indices.out, omat.aims, mos.aims`
- Create a file `tcontrol`, launch a script `tcontrol.aims.x` with arguments (consult also README file for parameters):

tcontrol.aims.x -lsurc n_1 -lsurx n_2 -lsury n_3 -rsurc m_1 -rsurx m_2 -rsury m_3 -nlayers N_a -ener E_1 -estep dE -eend E_2



here n_1, n_2, n_3 are indices of atoms defining the left outermost atomic plane ;

m_1, m_2, m_3 are indices of atoms defining the right outermost plane ;

N_a is number of atomic layers where absorbing boundary conditions are active (value of N_a is to be taken from the header of aitranss library files comprising structures of fcc Au clusters!) ;

E_1, E_2 & dE define a window [E_1, E_2] and energy step (in Ha units!) required to output the transmission function $T(E)$

Final step: AITRANSS calculation and output

- Invoke AITRANSS (without MPI!) :

```
nohup aitranss.x > aitranss.out &
```

- On output, you get two ASCII files

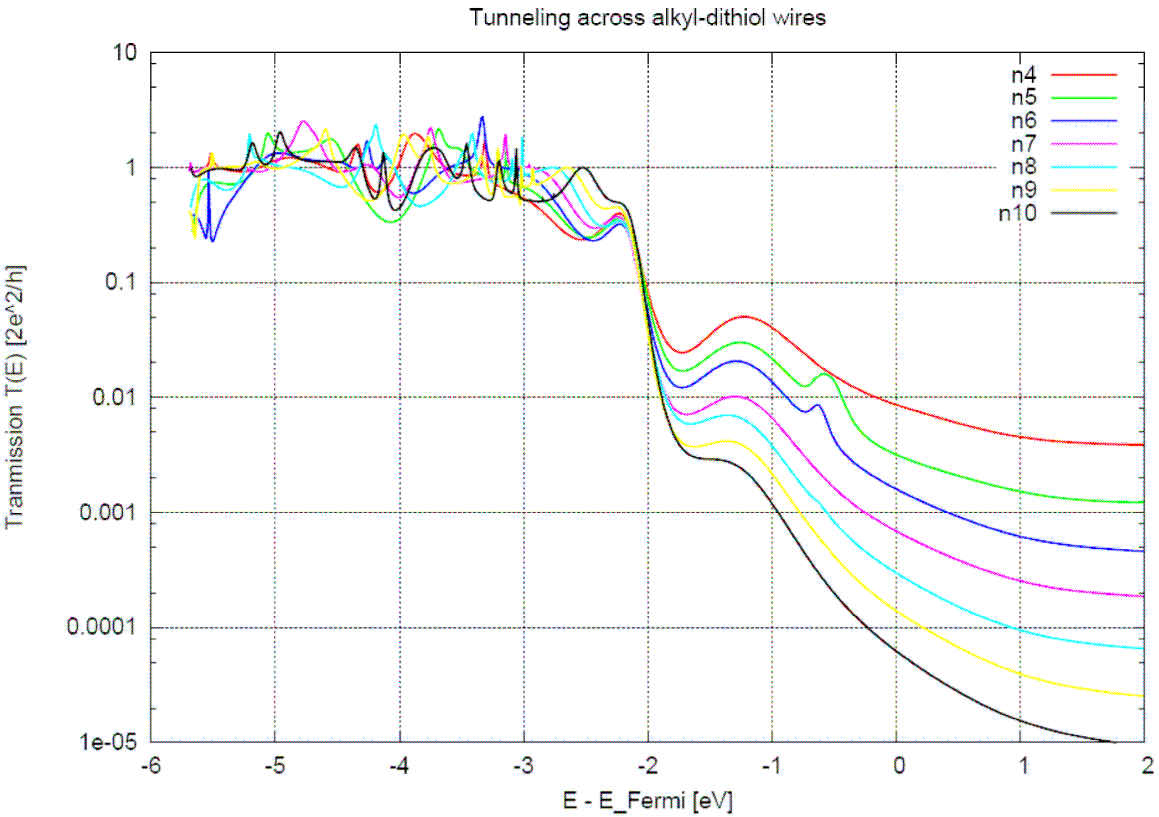
`self.energy.in` : contains information on the ***model self-energy*** ;
data are arranged in 7 columns: atom index, xyz-coordinates ($\mathbf{x}_n$), atom symbol, atom target (left/right/empty) and the local leakage rate $\eta(\mathbf{x}_n)$ [in Ha units]

`TE.dat` : contains ballistic transmission function $T(E)$;
data are arranged in 3 columns: E in Ha units, E-E_F in eV units, and $T(E)$
results may be visualized with gnuplot, e.g. `plot 'TE.dat' u 2:3 w l`

Advanced exercise

- Look into directory .../exercise.alkyl.wires: it contains 7 molecular structures: organic molecular wires (alkyl chains) of different length linked to Au electrodes
- Exercise: compute tunneling conductance *G* through all molecules; estimate a decay exponent of *G* as function of CH2 units in the wire; compare your results with those published by C. Li *et al.*, J. Am. Chem. Soc. **130**, 318 (2008)

Figure on the right show expected results


Tunneling across alkyl-dithiol wires
Tranmission T(E) [2e^2/h]
1e-05
E - E_Fermi [eV]
n4
n5
n6
n7
n8
n9
n10