Features in SIESTA

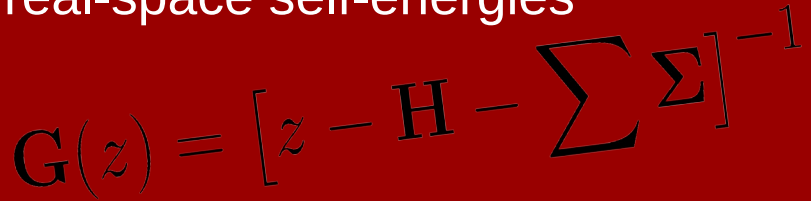
Nick Papior

Dept. of Applied Mathematics and Computer Science, Technical University of Denmark

MAX COE / ENCCS Siesta Webinar

TranSiesta
- $N \geq 1$ terminal NEGF
- real-space self-energies

Charged gates



Spin orbit coupling

Spin orbit coupling
- allows fully relativistic calculations
- 2x2 spin box → quadroupling of diagonalization memory
- spin-orbit contribution to the total energy is *small!*

$$\begin{bmatrix} \mathbf{H}_{\uparrow\uparrow} & \mathbf{H}_{\uparrow\downarrow} \\ \mathbf{H}_{\downarrow\uparrow} & \mathbf{H}_{\downarrow\downarrow} \end{bmatrix}$$

Quantities:
- calculating magnetic anisotropies (MCA)
- spin textures

An *off-site* method
- default, and should be used!

An *on-site* method
- non-default, may yield wrong results!
  see 10.1103/PhysRevB.104.195104

Requirements
- a well converged pseudo potential in the fully relativistic form
- a well converged basis set (remember talks from yesterday)
- a higher precision of the grid (`MeshCutoff`) is advised!
- SCF convergence may be more difficult (more DoF)
- *Be careful in your convergence studies!*

https://docs.siesta-project.org/projects/siesta/en/latest/index.html

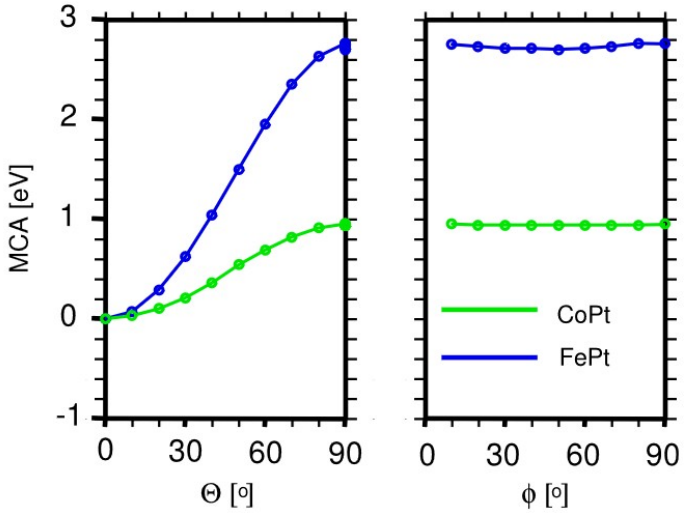
10.1088/0953-8984/24/8/086005 (off-site) | 10.1088/0953-8984/18/34/012 (on-site)

Spin orbit coupling

With spin orbit coupling the spin components are mixed.
There is no “spin up”/“spin down”.

Initialize spins to search for different spin configurations

```
%block DM.InitSpin
   5 -1.  90. 0.
%endblock DM.InitSpin
```



Searching MAE can be efficiently done by rotating density matrix elements and restarting (here sisl snippet):

```
DM = sisl.get_sile("RUN.fdf").read_density_matrix()
# Rotate 10^o around x, 20^o around y and then 30^o around z
DM = DM.spin_rotate([10, 20, 30])
DM.write(“new.DM”)
```

Mulliken/PDOS yields the different spin components.
- Total
- spin vector {x, y, z} components

https://zerothi.github.io/sisl

10.1088/0953-8984/24/8/086005 (off-site)

Custom k-point samplings

Siesta allows custom k-point samplings

- can be useful when converging properties where only a small portion of the Brillouin zone contributes to the Fermi-physics (e.g. k-point in graphene)

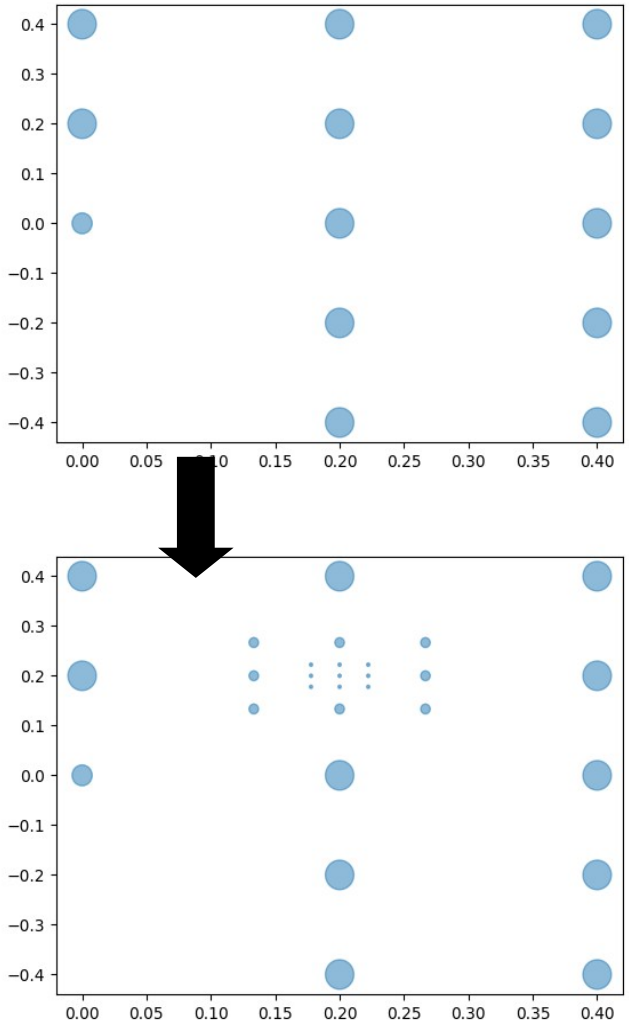
- spin-orbit requires a dense k-grid, often only Fermi-physics is important

- create your k-points using your favourite program and write a file

- input using:

```
kgrid.File filename.RKP
```

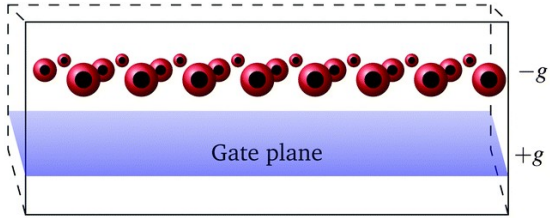
```
4
1 0.0 0.0 0.0 0.25
2 0.5 0.5 0.5 0.25
3 0.2 0.2 0.2 0.25
4 0.3 0.3 0.3 0.25
```



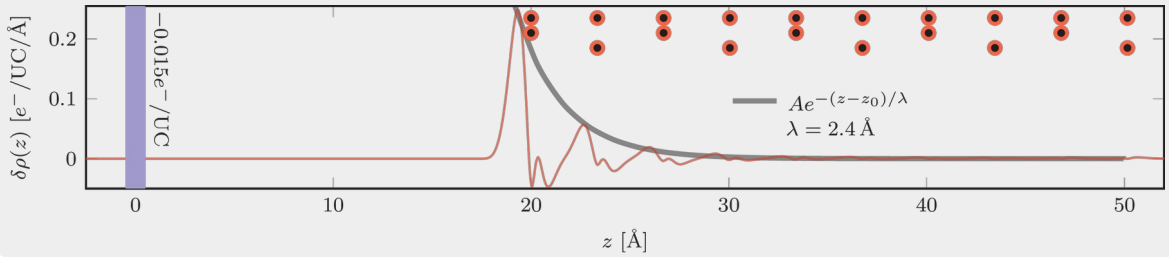
Charged gates

Siesta allows for complex gate/charge configurations

- adding spheres in *any* position in the simulation box
- infinite planes
- boxes of charges
- any combination and count of the above



Density decay length for graphite (capacitor)

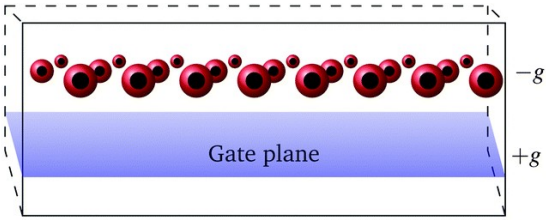


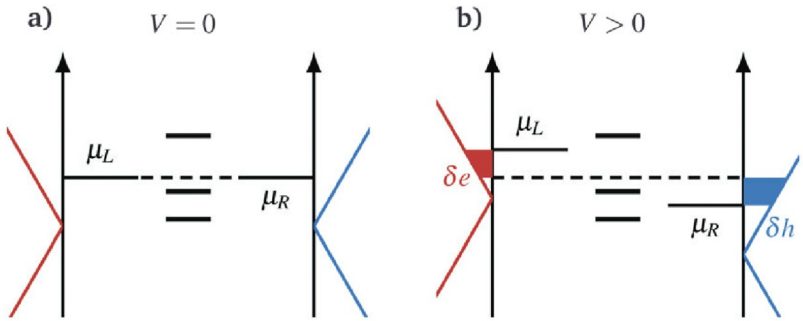
10.1039/C5CP04613K

Charged gates – capacitor calculations

Can be combined with TranSiesta to investigate gate constructs.

- requires care for correct gate handling around electrodes
  - a charge gate under an electrode should be reflected in the electrode calculation
- note that it is not a *gate potential*, rather a *gate charge*!





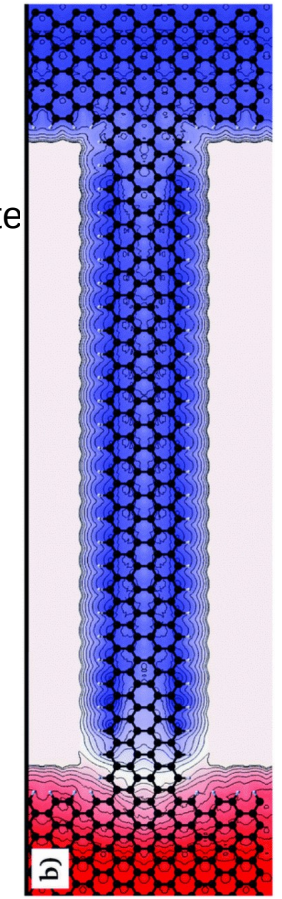
Graphene constriction with gate (capacitor)

- Applied bias of 0.5 V

- Potential drop

- Pinned to one electrode

- Not found without gate!



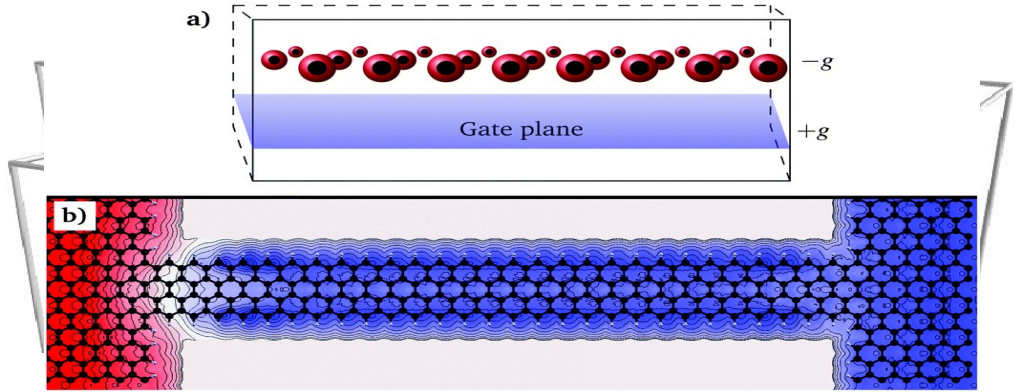
10.1039/C5CP04613K | 10.1016/j.cpc.2016.09.022

TranSiesta

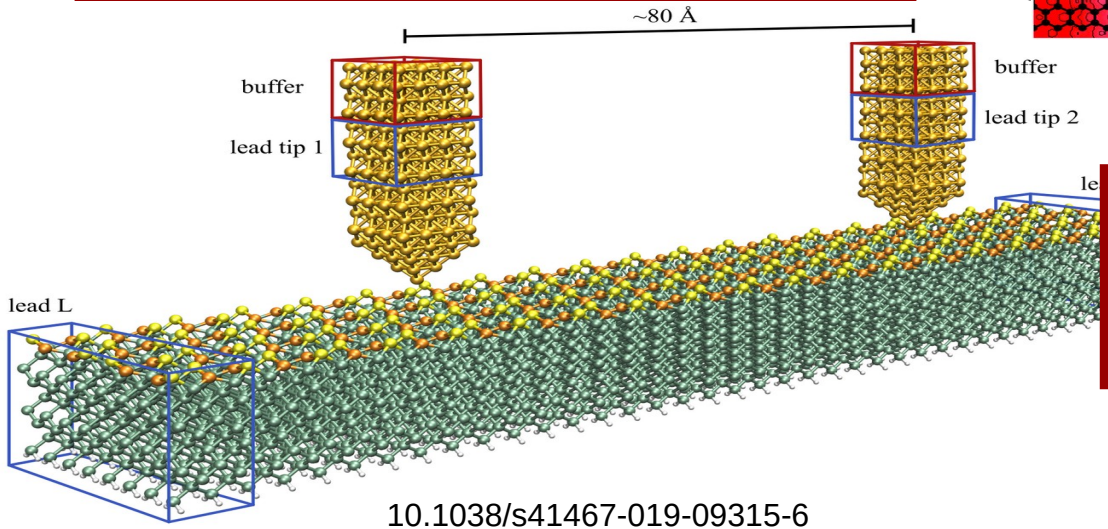
10.1063/1.4974895

- N ≥ 1 electrodes
  - 1 electrode is a surface!
  - surrounding electrodes

Any number of electrodes
and with bias!

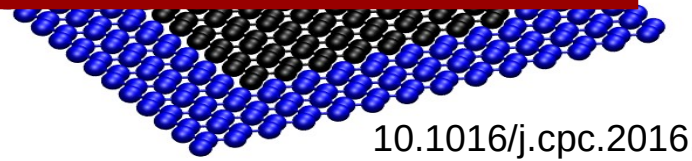


10.1039/C5CP04613K
10.1103/PhysRevB.100.195417



10.1038/s41467-019-09315-6

Gates with complex shapes



10.1016/j.cpc.2016.09.022

TranSiesta -- NEGF

Quantities:
- can calculate systems *out-of-equilibrium* (potential difference between electrodes)
- calculate energy + k-point resolved transmission functions
- (projected) density of states
- calculate current between electrodes
- orbital transmissions (bond currents) to detect current paths
- … many more ...

Requirements
- same requirements as regular Siesta calculations (pseudo, basis set, etc.), additionally:
- bulk and *metallic* electrodes
- electrode regions in the device should be screened from the *defect*
- special contour parameters needs care to ensure lowest lying states are captured
- SCF convergence may be more difficult, typically lower mixing weights are needed
- 1 calculation == 1 bias point

- a 2 step process
  1) converge non-equilibrium density (for a single bias point) [TranSiesta]
  2) calculate transmission + others [TBtrans]
- TBtrans requires subsequent convergence of k-points (generally much higher grids are needed)

TranSiesta + TBtrans + sisl tutorials: https://github.com/zerothi/ts-tbt-sisl-tutorial

10.1016/j.cpc.2016.09.022

Matrix inversions

The NEGF algorithm is limited by the matrix inversion:

$$\mathbf{G}(z) = \left[z - \mathbf{H} - \sum \mathbf{\Sigma}\right]^{-1}$$

Performance limited by inversion

- Dense matrix inv

- Sparse inversions scale depending on sparsity patterns (basis set)
  - PEXSI (10.1088/0953-8984/26/30/305503)
  - MUMPS (10.1093/gji/ggx106)
  - others..

- Block tri diagonal inversion (recursive Green function)

Block tri diagonal inversion

Basic principle for narrow matrices

more narrow -> smaller blocks -> faster the algorithm

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{C}_2 & 0 & & \cdots \\ \mathbf{B}_1 & \mathbf{A}_2 & \mathbf{C}_3 & 0 & \cdots \\ 0 & \mathbf{B}_2 & \ddots & \ddots & \\ & 0 & \ddots & & \mathbf{C}_p \\ \vdots & \vdots & & \mathbf{B}_{p-1} & \mathbf{A}_p \end{pmatrix}$$

$$\mathbf{X}_n = \mathbf{C}_{n+1}\left[\mathbf{A}_{n+1} - \mathbf{X}_{n+1}\right]^{-1}\mathbf{B}_n \quad \mathbf{X}_p \equiv 0$$
$$\mathbf{Y}_n = \mathbf{B}_{n-1}\left[\mathbf{A}_{n-1} - \mathbf{Y}_{n-1}\right]^{-1}\mathbf{C}_n \quad \mathbf{Y}_1 \equiv 0$$

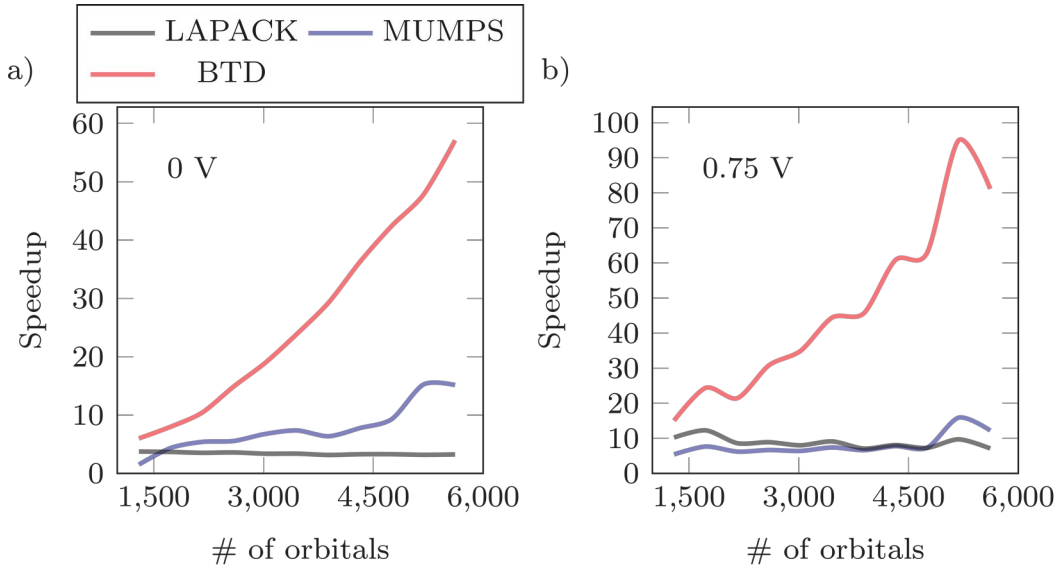
$$\mathbf{M}^{-1}_{n,n} = \left[\mathbf{A}_n - \mathbf{X}_n - \mathbf{Y}_n\right]^{-1}$$
$$\mathbf{M}^{-1}_{m,n} = -\left[\mathbf{A}_m - \mathbf{X}_m\right]^{-1}\mathbf{B}_{m-1}\mathbf{M}^{-1}_{m-1,n} \quad , \text{for } m > n$$
$$\mathbf{M}^{-1}_{m,n} = -\left[\mathbf{A}_m - \mathbf{Y}_m\right]^{-1}\mathbf{C}_{m+1}\mathbf{M}^{-1}_{m+1,n} \quad , \text{for } m < n$$
$$\mathbf{M}^{-1}_{m,n} = -\mathbf{C}^{-1}_m\mathbf{X}_{m-1}\mathbf{M}^{-1}_{m-1,n} \quad , \text{for } m > n$$
$$\mathbf{M}^{-1}_{m,n} = -\mathbf{B}^{-1}_m\mathbf{Y}_{m+1}\mathbf{M}^{-1}_{m+1,n} \quad , \text{for } m < n$$

Much – much faster
Does NOT require hermiticity!

10.1088/0953-8984/3/40/005

Performance of matrix inversions (comparison with <= 4.0)

TranSiesta implements 3 different inversion algorithms
- dense matrices (LAPACK)
- sparse inversion (MUMPS)
- BTD



doubling threads
->
halves memory

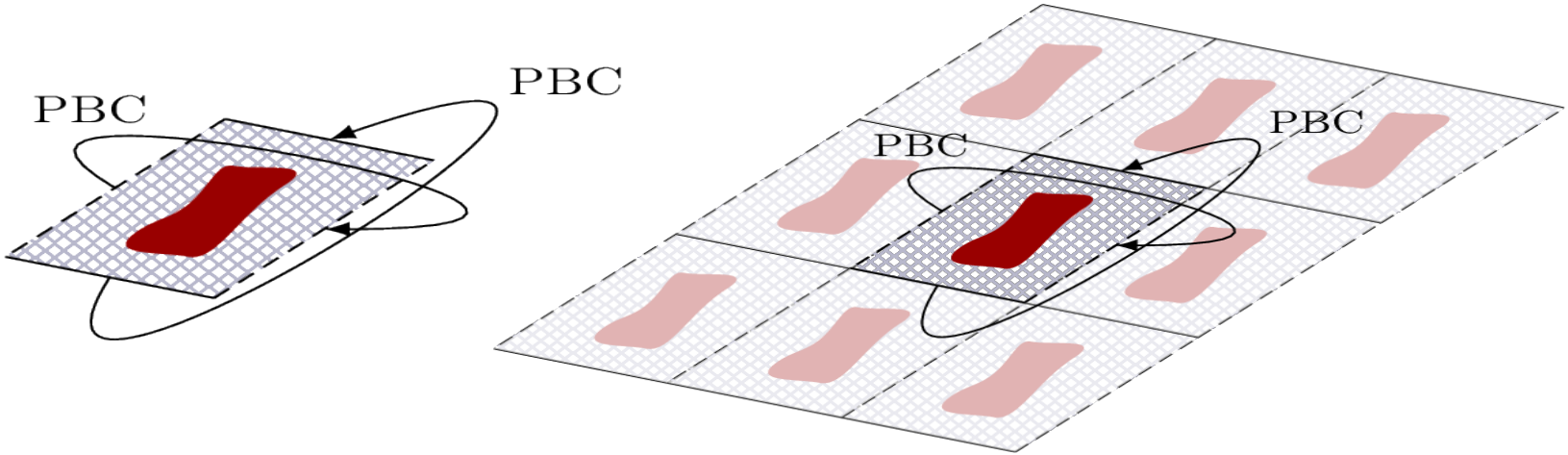
A factor ~100!

10.1016/j.cpc.2016.09.022

Removing all periodic boundaries

– a NEGF approach

DFT calculations for defects are problematic due to the periodic boundary conditions. One have to perform large supercell calculations to damp interference effects in order to retrieve the defect ground state.



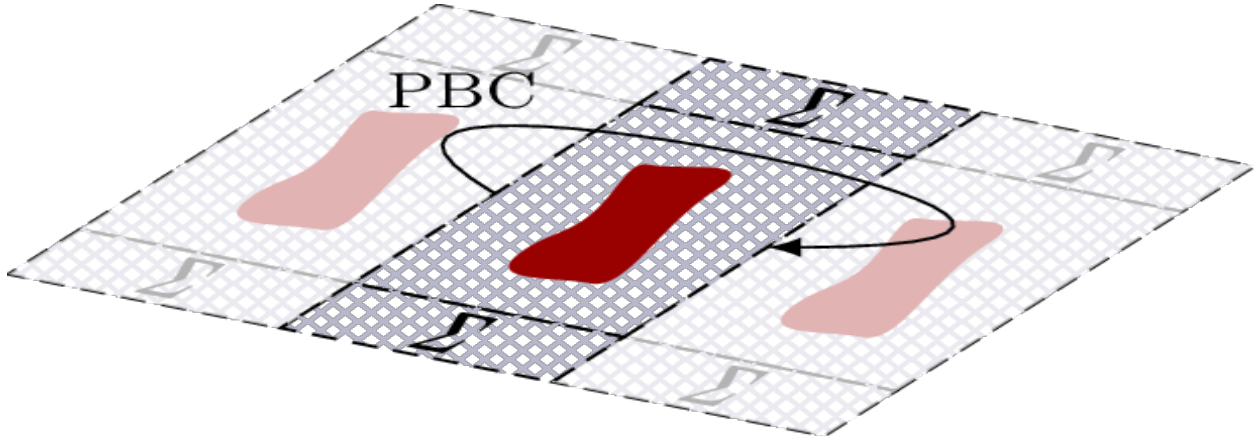
Removing **all** periodic boundaries

– a NEGF approach

Non equilibrium Green functions *also* relies on the periodic supercell approach, this induces interference effects

[see Thygesen 10.1103/PhysRevB.72.033401]

It however limits PBC to transverse directions by the use of *self-energies*

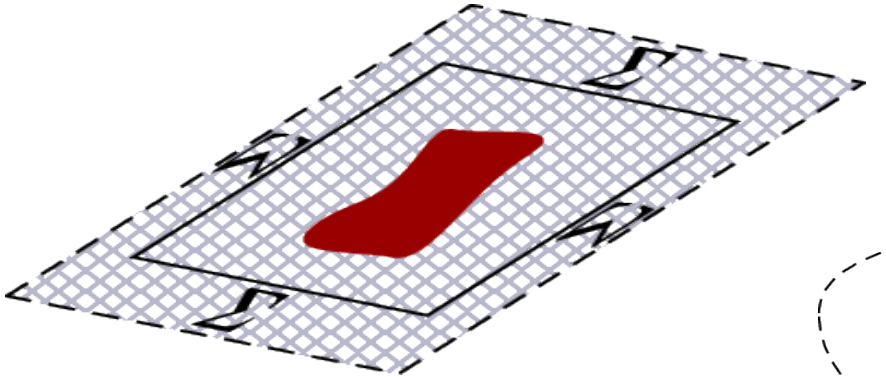


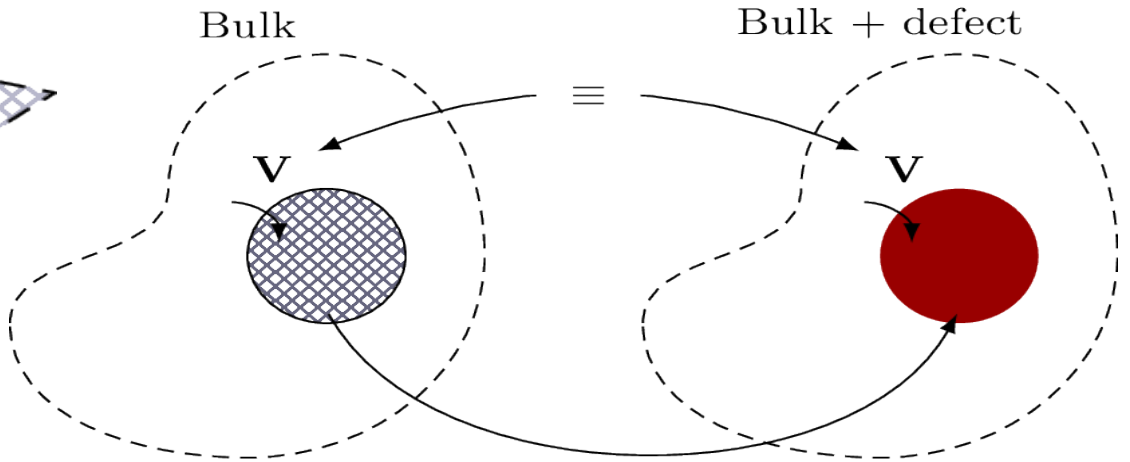
10.1103/PhysRevB.100.195417

Removing **all** periodic boundaries

– a NEGF approach

Completely remove image coupling





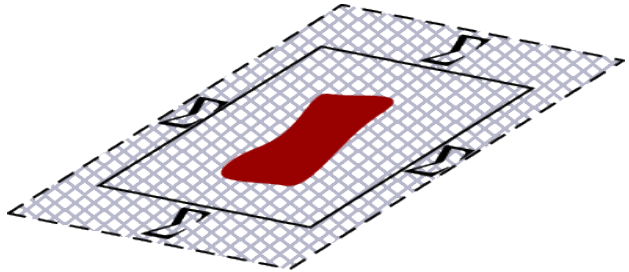
$$\mathbf{S}_{\otimes} z - \mathbf{H}_{\otimes} - [\mathbf{G}^{\mathcal{R}}_{\otimes}]^{-1} = \mathbf{\Sigma}^{\mathcal{R}}_{\otimes} \equiv \mathbf{\Sigma}^{\mathcal{R}}_{\bullet}$$

$$\mathbf{G}_{\mathcal{R}} = \int_{\mathrm{BZ}} \mathbf{G}_{\mathbf{k}}$$

Calculating the real space self-energy

The problem is calculating the real space Green function:

$$\mathbf{G}_{\mathcal{R}} = \int_{\mathrm{BZ}} \mathbf{G}_{\mathbf{k}}$$



- The Green function calculation scales N^3
- Size of Green function depends on “surrounding part”

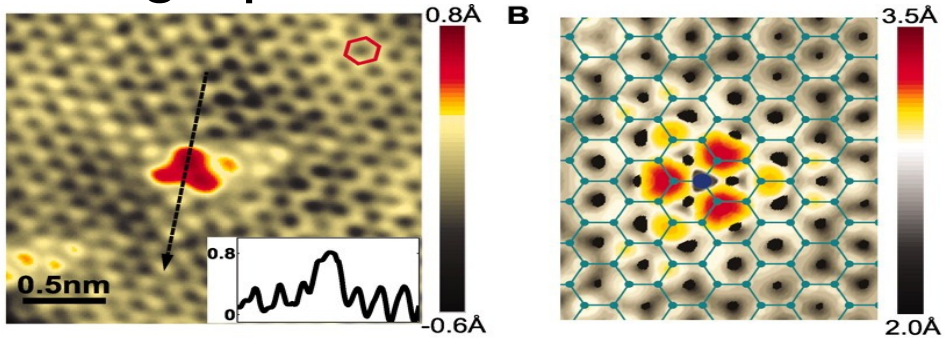
Using full matrices, one cannot do real space Green function calculations on systems with more than a few hundred orbitals!

10.1103/PhysRevB.100.195417

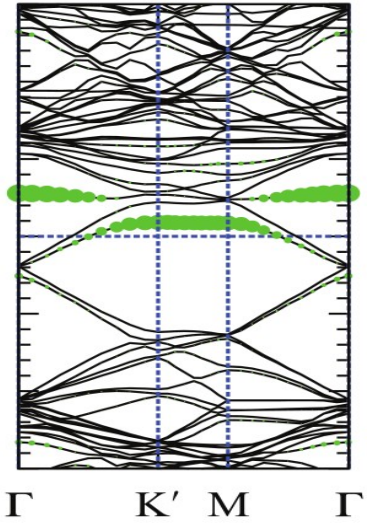
Case study – Nitrogen defect in graphene

Science **333**, 6045, 999 (2011)

Experiment and DFT-STM (7x7)



DFT study of graphene with Nitrogen defect using supercell approach



[Hou: 10.1103/PhysRevB.87.165401]

Band structure of graphene with N defect

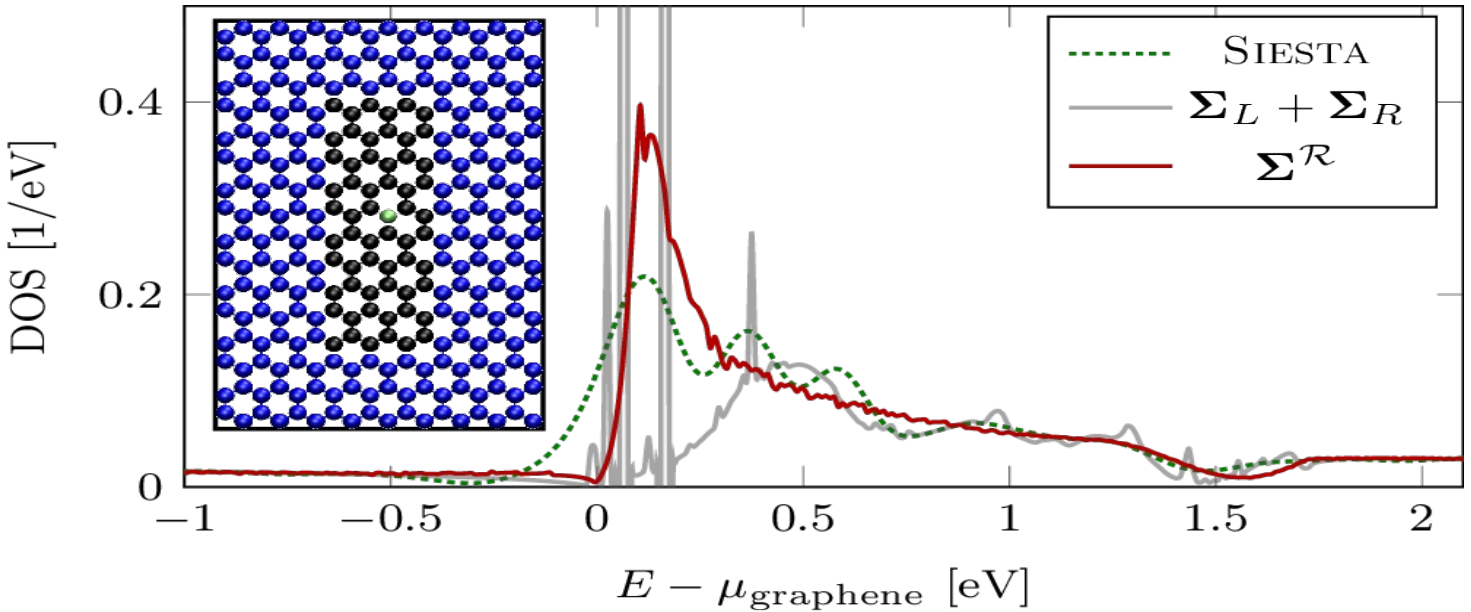
Green highlight N defect

Graphene 9x9 system (162 atoms)

Dispersion resulting in 2-peak structure in DOS!

Case study – Nitrogen defect in graphene

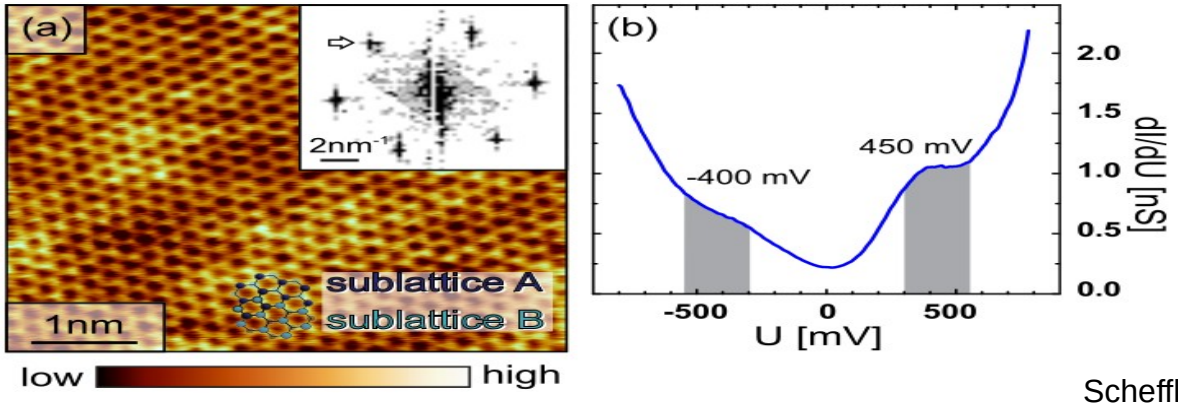
- 2D PBC (Siesta)
- 1D PBC (TranSiesta)
  2 electrodes
- No PBC (TranSiesta)
  1 electrode



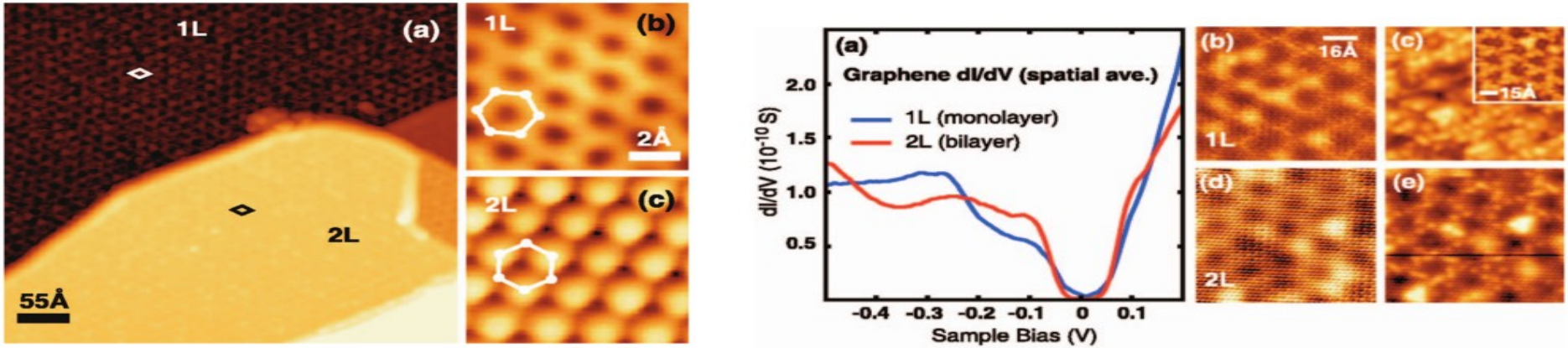
Multiple peaks reduces to one, consistent with large scale tight binding calculations PRB **86** 045448 (2012)

10.1103/PhysRevB.100.195417

Case study – STM tip on graphene



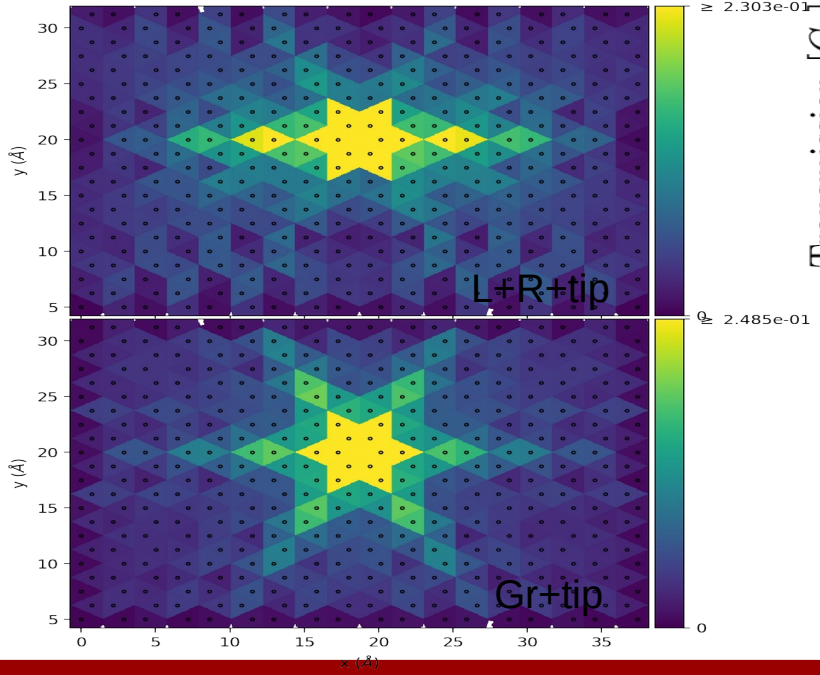
Scheffler, ACS Nano **6** 12 10590 (2012)

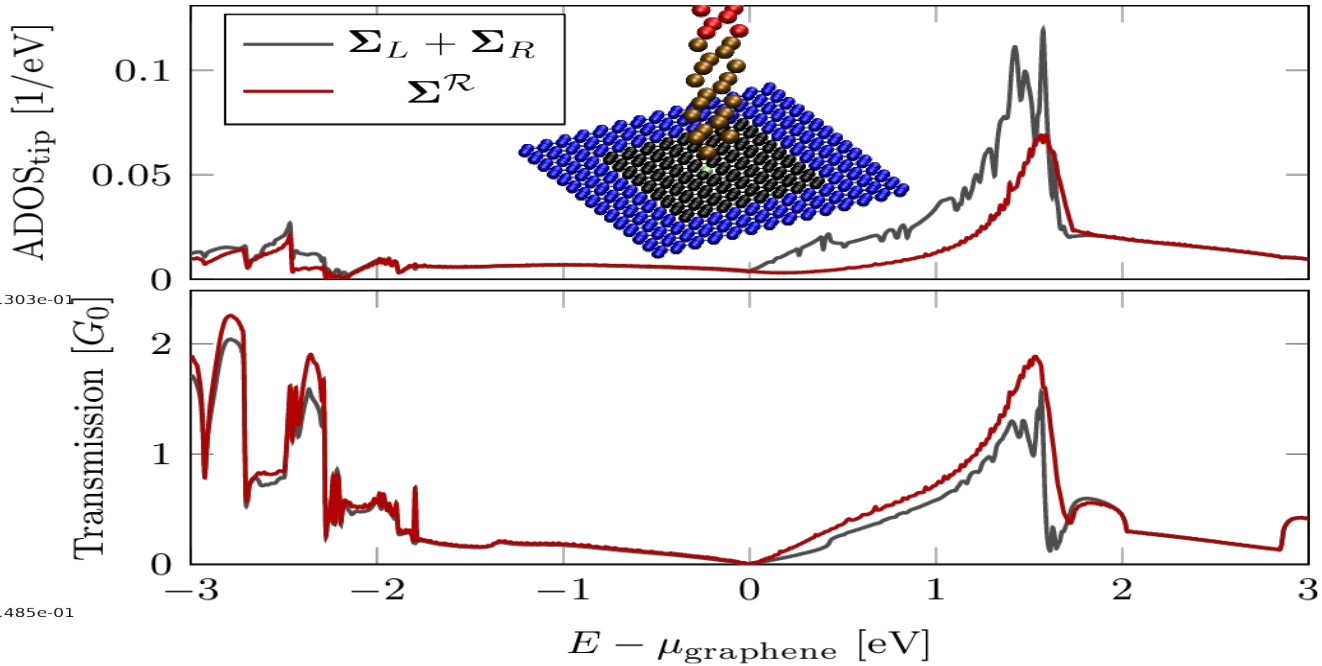


Brar, Appl. Phys. Lett. **91** 122102 (2007)

Case study – STM on graphene @ V = 0.5 V

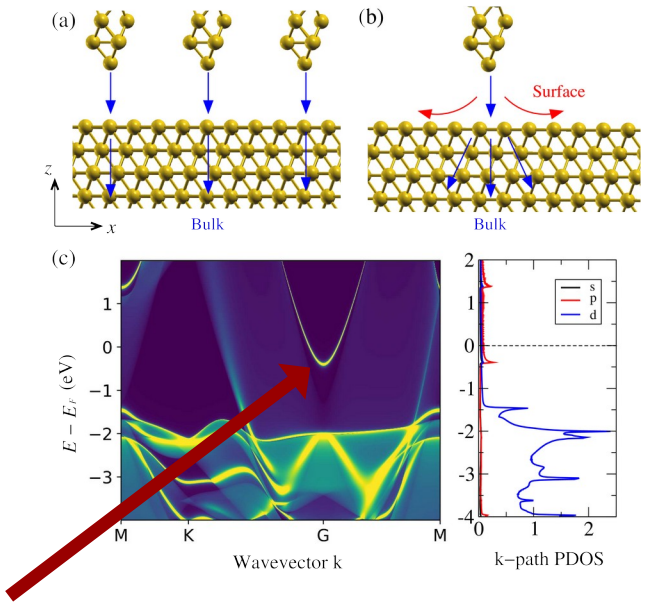
Interference effects:





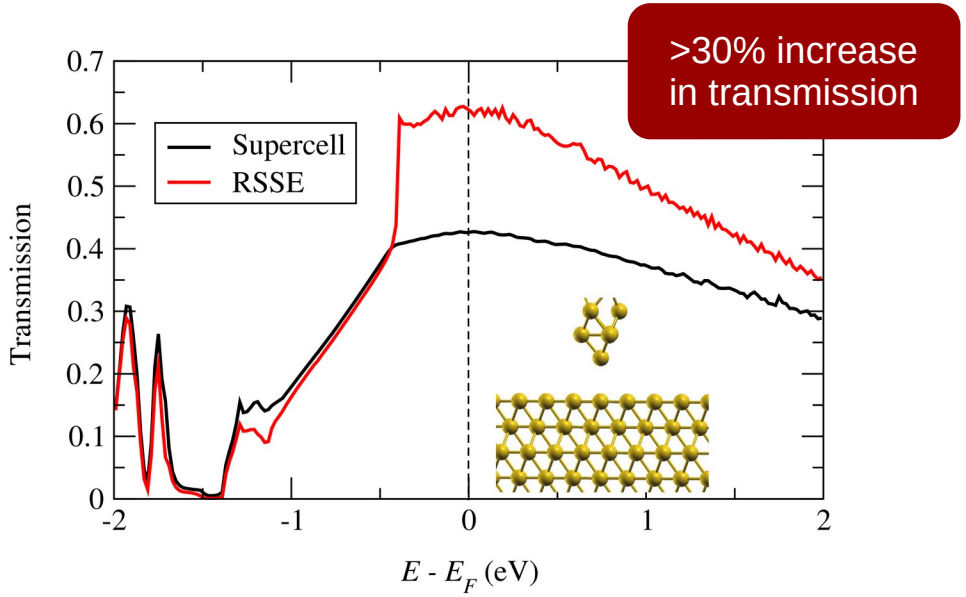
10.1103/PhysRevB.100.195417

Case study – Surface states on Au(111)



Surface state with *p* character!

Problem of PBC



10.1103/PhysRevResearch.3.033017

Lua – scripting interface

Siesta can be *controlled*/monitored through a Lua scripting language

```
function siesta_comm()
   if siesta.state == siesta.INITIALIZE then
     -- startup phase of siesta
     -- get SystemLabel and initial coordinates/species
   elseif siesta.state == siesta.SCF_LOOP then
     -- control mixing parameters, extract SCF variables
     -- dDmax, dHmax, Ef, etc.
   elseif siesta.state == siesta.MOVE then
     -- control MD in Lua scripting
...
```

Change mixing parameters on the fly

Prototype MD algorithms

Monitor quantities on the fly

See Siesta sources:

- Tests/Dependency_Tests
- Tests/Interface_Tests

Or the tutorials:

https://docs.siesta-project.org/projects/siesta/en/latest/index.html