CP2K: An ab initio materials simulation code

Lianheng Tong

Physics on Boat Tutorial , Helsinki, Finland 2015-06-08

Faculty of Natural & Mathematical Sciences
Department of Physics

King's College London

Brief Overview

- Overview of CP2K
  - General information about the package
  - QUICKSTEP: DFT engine
- Practical example of using CP2K to generate simulated STM images
  - Terminal states in AGNR segments with 1H and 2H termination groups

What is CP2K?

Swiss army knife of molecular simulation

www.cp2k.org

- Geometry and cell optimisation
- Molecular dynamics (NVE, NVT, NPT, Langevin)
- STM Images
- Sampling energy surfaces (metadynamics)
- Finding transition states (Nudged Elastic Band)
- Path integral molecular dynamics
- Monte Carlo
- And many more…

Energy and Force Engine

- DFT (LDA, GGA, vdW, Hybrid)
- Quantum Chemistry (MP2, RPA)
- Semi-Empirical (DFTB)
- Classical Force Fields (FIST)
- Combinations (QM/MM)

Development

- Freely available, open source, GNU Public License
  - www.cp2k.org
- FORTRAN 95, > 1,000,000 lines of code, very active development (daily commits)
- Currently being developed and maintained by community of developers:
  - **Switzerland**: Paul Scherrer Institute Switzerland (PSI), Swiss Federal Institute of Technology in Zurich (ETHZ), Universität Zürich (UZH)
  - **USA**: IBM Research, Lawrence Livermore National Laboratory (LLNL), Pacific Northwest National Laboratory (PNL)
  - **UK**: Edinburgh Parallel Computing Centre (EPCC), King's College London (KCL), University College London (UCL)
  - **Germany**: Ruhr-University Bochum
  - **Others**: We welcome contributions from interested users, just send code to a developer. After passing quality check can be in SVN trunk within days.

User Base

- Large user base across the world
- Recent UK national HPC service report:
  - 2nd most used code on ARCHER
  - Number of users increasing
  - Preferred by users for larger simulations compared to traditional plane-wave codes
- User community through Google Group
  - https://groups.google.com/forum/#!forum/cp2k
- Tutorial pages:
  - http://www.cp2k.org/tutorials
  - Wiki format, users are encouraged to share their experiences with the community

Figure 3: Periodic electronic structure code usage across systems as a function of % core hours used.

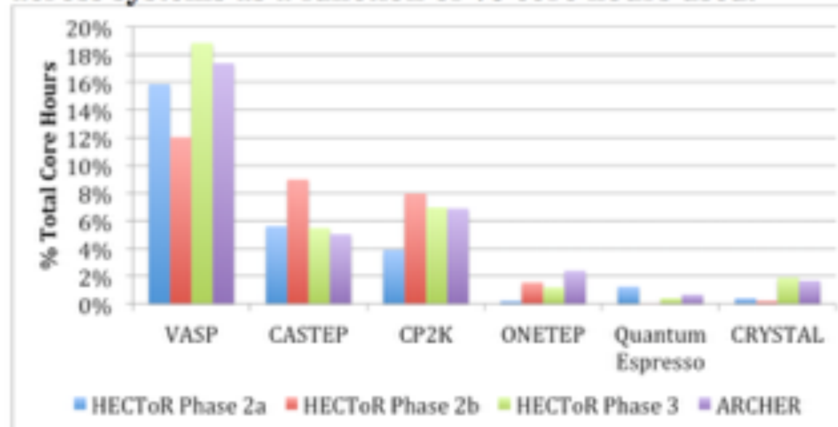


Table 5: Median job sizes (in cores) for periodic electronic structure codes on each of the systems.

	HECToR Phase 2a	HECToR Phase 2b	HECToR Phase 3	ARCHER
VASP	240	456	480	240
CASTEP	252	720	512	360
CP2K	224	1320	608	672
ONETEP	104	504	416	864
Quantum Espresso	60	72	448	192
CRYSTAL	144	4032	3648	2808

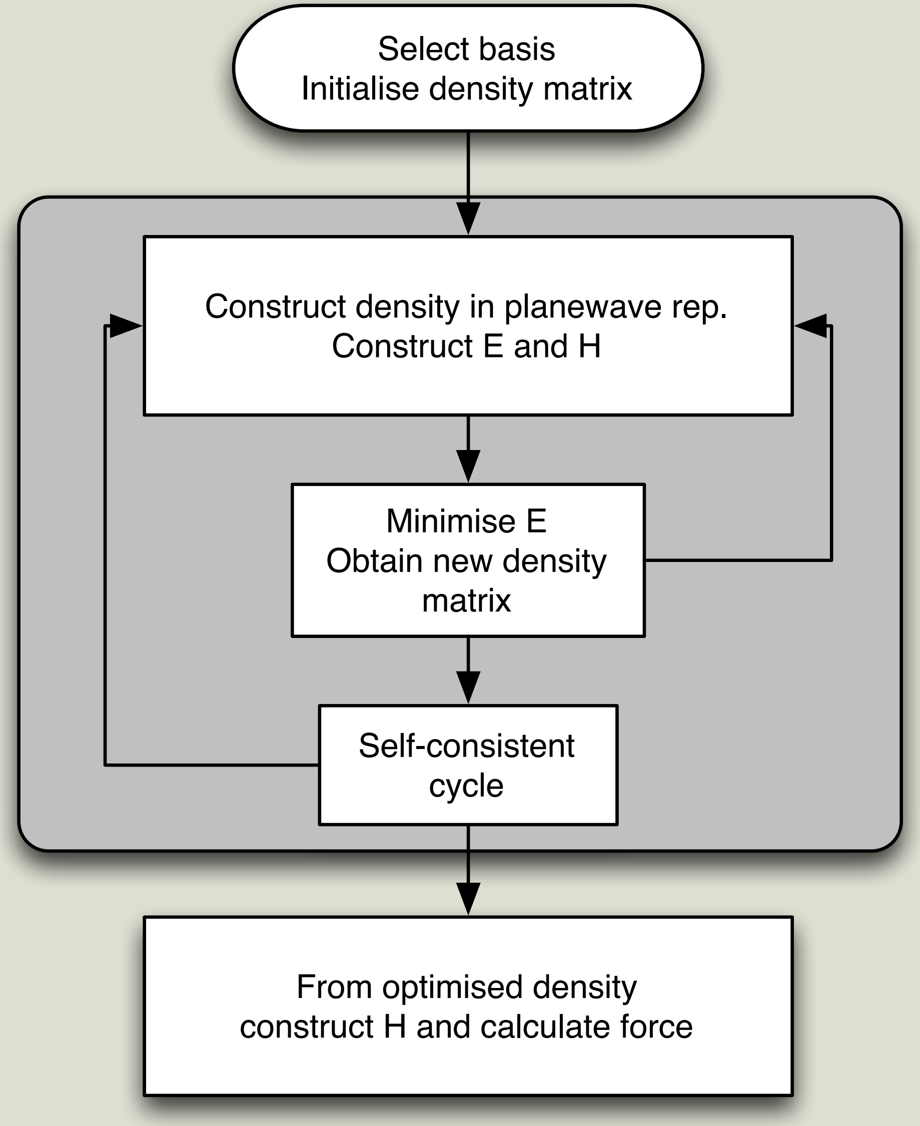
Turner, A, https://www.archer.ac.uk/documentation/white-papers/app-usage/UKParallelApplications.pdf

Information For Developers

- Parallelisation:
  - MPI + OpenMP
  - CUDA C (GPU): sparse matrix multiplication engine
- Quality Control on Development:
  - Over 2000 automatic regression tests and memory leak tests
- Own Libraries:
  - DBCSR (parallel sparse matrix format and multiplication)
  - libsmm (small matrix multiplication)
- External Libraries:
  - BLAS/LAPACK (MKL, ACML, ATLAS, Cray-LibSci, ...)
  - ScaLAPACK/BLACS, ELPA
  - FFTW
  - libint (HF exact exchange)
  - libxc (exchange-correlation functionals)

DFT Solver: QuickStep

- Basis Sets: GWP
  - Contracted Gaussians functions for matrices
  - Planewaves for density function: electrostatics using FFT
- Uses Pseudopotentials based on Gaussian functions (Goedecker-Teter-Hutter)
  - A library of pseudo potentials and basis functions for most elements in the periodic table comes with the package
- Energy Minimiser
  - Diagonalisation
  - Orbital Transform
- Self-consistent cycle
  - Pulay, Broyden



Direct Diagonalisation + DIIS

- Construct H and S, and solve eigenvalue problem directly — O(N^3)

$$\hat{H}\psi_n(\mathbf{r}) = E_n\psi_n(\mathbf{r})$$

$$\rho(\mathbf{r}) = \sum_n f_n(E_n, N_e)\psi_n(\mathbf{r})\psi_n(\mathbf{r})^*$$

- CP2K reorthogonalises the basis set by using transformation $\mathrm{S}^{-\frac{1}{2}}$
- Uses ScaLAPACK (or ELPA)
- DIIS (Pulay mixing): $\rho_{\mathrm{in}}^{\mathrm{opt}} = \sum_{i=m-N_P+1}^{m} \alpha_i \rho_{\mathrm{in}}^i \qquad \sum_{i=m-N_P+1}^{m} \alpha_i = 1$

$$\langle R[\rho_{\mathrm{in}}^m]|R[\rho_{\mathrm{in}}^m]\rangle \equiv \int \mathrm{d}^3\mathbf{r} R([\rho_{\mathrm{in}}^m],\mathbf{r})R([\rho_{\mathrm{in}}^m],\mathbf{r})$$

$$\alpha_i = \frac{\sum_{j=m-N_P+1}^{m} A_{ji}^{-1}}{\sum_{i,j=m-N_P+1}^{m} A_{ji}^{-1}}$$

$$\langle R[\rho_{\mathrm{in}}^{\mathrm{opt}}]|R[\rho_{\mathrm{in}}^{\mathrm{opt}}]\rangle \leq \langle R[\rho_{\mathrm{in}}]|R[\rho_{\mathrm{in}}]\rangle$$

$$A_{ij} \equiv \langle R[\rho_{\mathrm{in}}^i]|R[\rho_{\mathrm{in}}^j]\rangle$$

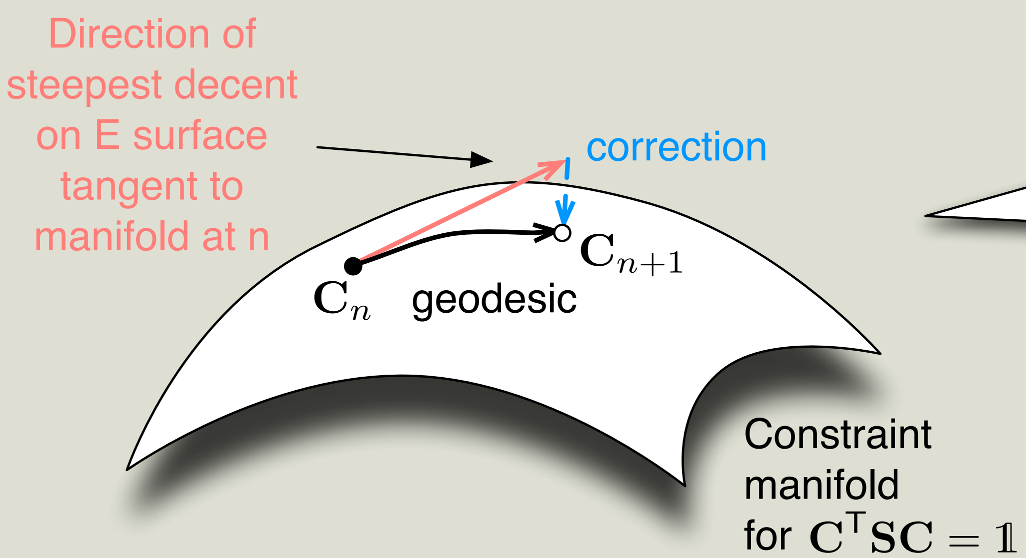
$$\rho_{\mathrm{in}}^{m+1} \equiv \rho_{\mathrm{in}}^{\mathrm{opt}} + AR[\rho_{\mathrm{in}}^{\mathrm{opt}}]$$

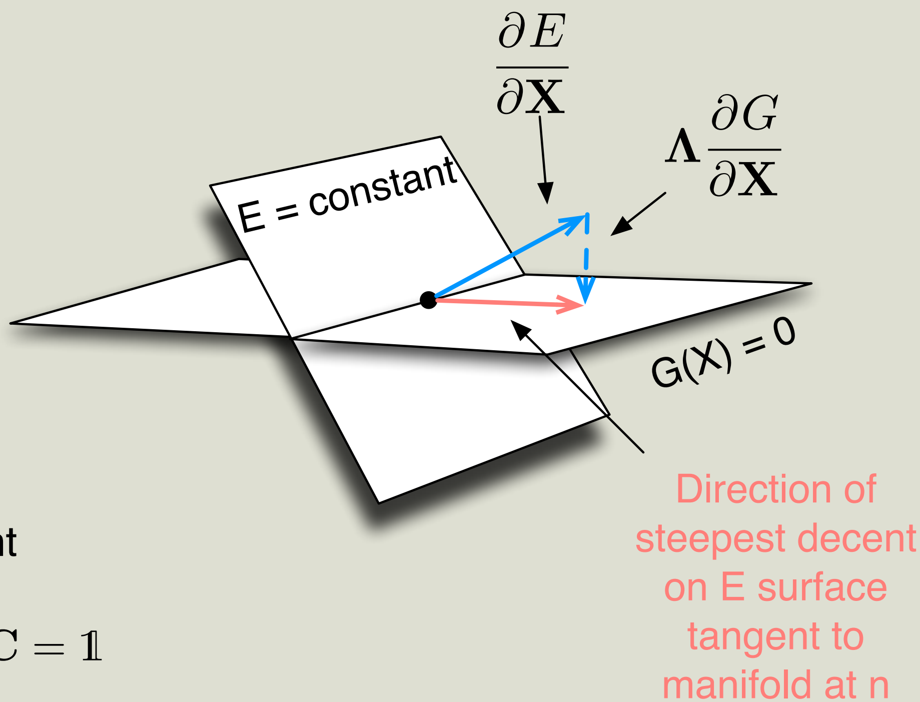
Orbital Transform Method

- Total KS energy is a functional of the electron density, which is a functional of the wavefunctions.
- Density Functional Theory: the minimal of the KS functional gives the ground state density and energy
- Orbital Transform method: find the ground state density (wavefunctions) by direct minimisation of the KS energy as a function of the wavefunction coefficients (in the Gaussian basis representation)
- Advantages:
  - Fast: does not involve expensive diagonalisation
  - If preconditioned correctly, method guaranteed to find minimum
- Disadvantages:
  - Sensitive to preconditioning. A good preconditioner can be expensive
  - No smearing, or advanced SCF mixing possible: poor convergence for metalic systems

Orbital Transform

- We need to minimise energy with respect to the constraint that wavefunctions always remain orthonormal: $$\sum_{ii'\mathbf{l}\mathbf{l}'} C_n^{i\mathbf{l}*} \langle \phi_{i\mathbf{l}} | \phi_{i'\mathbf{l}'} \rangle C_m^{i'\mathbf{l}'} = \delta_{nm}$$





Orbital Transform

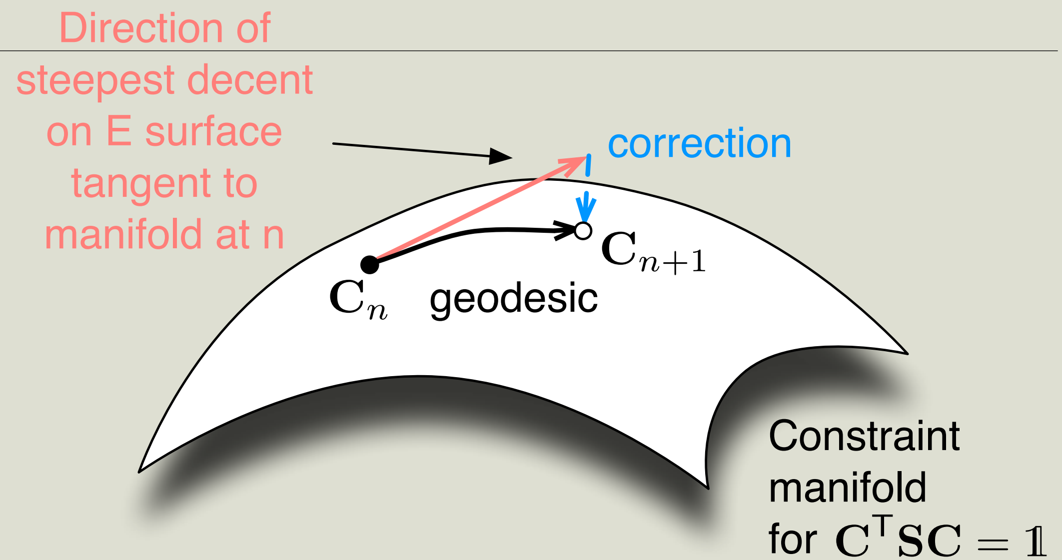
- Instead of minimising E with respect to wavefunction coefficients, make a transformation of variable to:

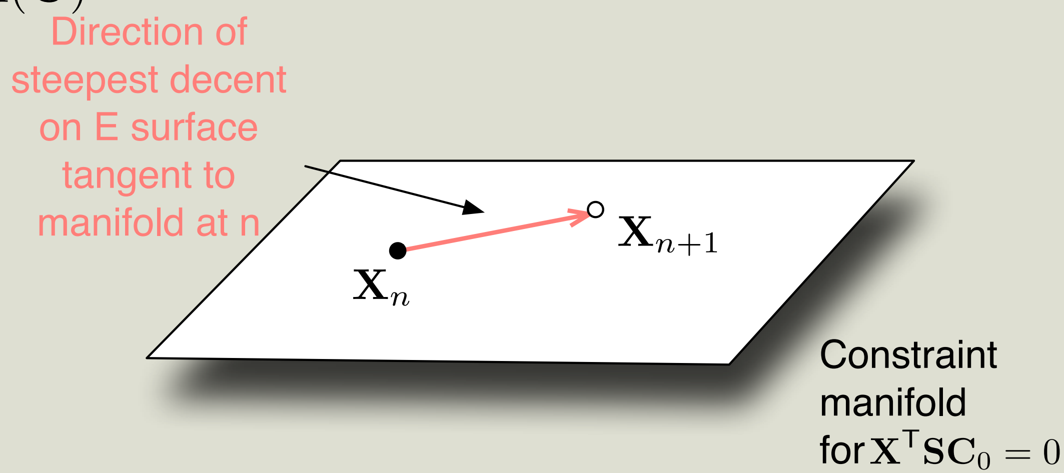
$$\mathbf{C}(\mathbf{X}) = \mathbf{C}_0 \cos(\mathbf{U}) + \mathbf{X}\mathbf{U}^{-1} \sin(\mathbf{U})$$

$$\mathbf{U} = (\mathbf{X}^{\mathsf{T}}\mathbf{S}\mathbf{X})^{\frac{1}{2}}$$

- With the constraint:

$$\mathbf{X}^{\mathsf{T}}\mathbf{S}\mathbf{C}_0 = 0$$





Orbital Transform

- Need to avoid diagonalisation
- Cosine and Sine functions are expanded in Taylor series up to order K (K = 2, 3 already give machine precision)
- Calculate inverse U as part of Taylor expansion

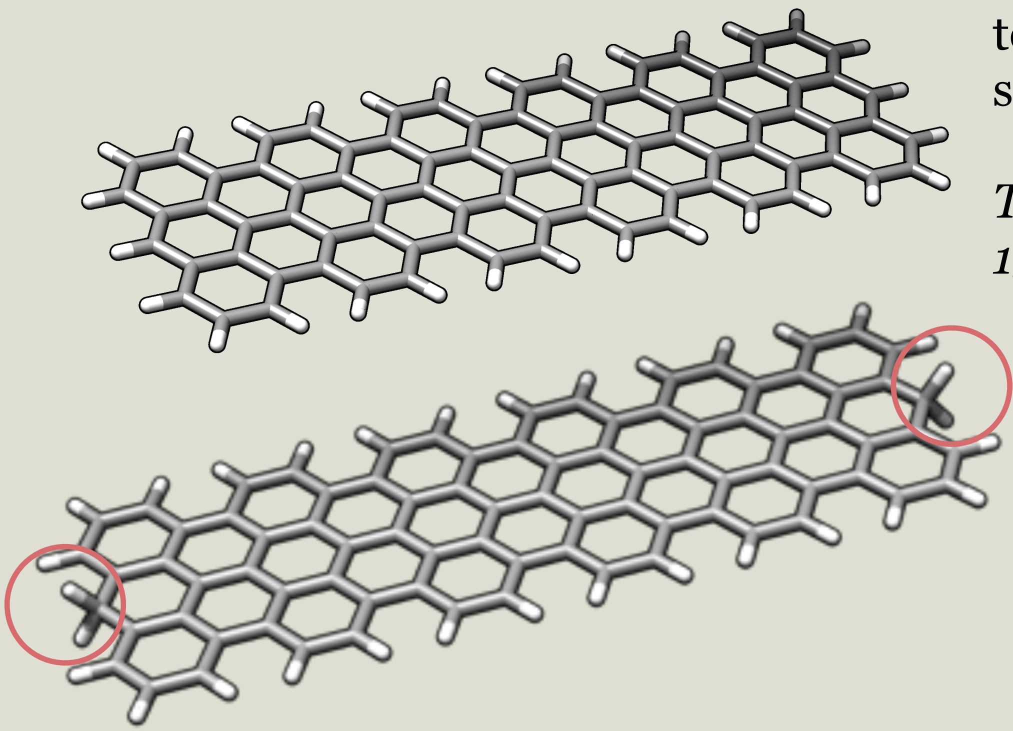
$$\cos(\mathbf{U}) = \sum_{i=0}^{K} \frac{(-1)}{(2i)!} (\mathbf{X}^{\mathsf{T}}\mathbf{S}\mathbf{X})^{i}$$

$$\mathbf{U}^{-1}\sin(\mathbf{U}) = \sum_{i=0}^{K} \frac{(-1)^{i}}{(2i+1)!} (\mathbf{X}^{\mathsf{T}}\mathbf{S}\mathbf{X})^{i}$$

- 10 to 20 times faster than Diagonalisation + DIIS

Practical Example: ANGR Terminal States

- Consider two polyantryl molecules (armchair graphene nano ribbon segments):



It has been observed that different termination leads to a change of state structure completely:

Talirz et al. J. Am. Chem. Soc., 2013, 135 (6), pp 2060–2063

The associated files for this tutorial is available for download at:

http://www.cp2k.org/howto:stm

Prepare CP2K Input Files

- To run a CP2K simulation, you need the minimum of following files:
  - Input parameters file
  - Atomic coordinate file — if not include in the input file
  - Pseudopotential data file
  - Gaussian basis set data file
- Pseudopotential and basis set can simply be copied from cp2k/data/ directory included in the package
- Atomic coordinate files can be most of the common formats such as .xyz or .pdb etc.
- A web-based GUI for generating the input parameter file is under development, expect to be released in the near future!

CP2K Input Parameters File

```
&section_name
  keyword values
  ...
  &subsection_name
    keyword values
    ...
  &END subsection_name
  ...
&END section_name
&another_section
  ...
&END another_section
```

CP2K input consists of nested sections

Job Admin

Controls overal verbosity

Job name

Job type

```
&GLOBAL
  PRINT_LEVEL LOW
  PROJECT GNR_A7_L11_2H
  RUN_TYPE GEO_OPT
&END GLOBAL
```

MD

Parameters for Job type

```
&MOTION
  &GEO_OPT
    ...
  &END GEO_OPT
  &CONSTRAINT
    ...
  &END CONSTRAINT
  &PRINT
    ...
  &END PRINT
&END MOTION
```

Constraints

Controls level of output during MD

Force Engine

Model type

Atomic coordinates and kinds

Model parameters

Controls level o output during Force eval

```
&FORCE_EVAL
  METHOD Quickstep
  &SUBSYS
    ...
  &END SUBSYS
  &DFT
    ...
  &END DFT
  &PRINT
    ...
  &END PRINT
&END FORCE_EVAL
```

First Relax The System

```
&GLOBAL
  PRINT_LEVEL LOW
  PROJECT GNR_A7_L11_1H
  RUN_TYPE GEO_OPT
&END GLOBAL
```

It is essential to first find you have enough grid points for the real space and FFT grid (i.e. large enough planewave basis set). A tutorial on how to do this can be found on http://www.cp2k.org/

```
&MOTION
  &GEO_OPT
    TYPE MINIMIZATION
    MAX_DR     1.0E-03
    MAX_FORCE 1.0E-03
    RMS_DR     1.0E-03
    RMS_FORCE 1.0E-03
    MAX_ITER 200
    OPTIMIZER BFGS
  &END GEO_OPT
  &CONSTRAINT
    &FIXED_ATOMS
      COMPONENTS_TO_FIX XYZ
      LIST 1
    &END FIXED_ATOMS
  &END CONSTRAINT
&END MOTION
```

We fix the first atom

```
&FORCE_EVAL
  METHOD Quickstep
  &SUBSYS
    &CELL
      ABC [angstrom] 60 30 20
      MULTIPLE_UNIT_CELL 1 1 1
    &END
    &TOPOLOGY
      COORD_FILE_NAME ./GNR_A7_L11_1H.xyz
      COORDINATE xyz
      MULTIPLE_UNIT_CELL 1 1 1
    &END
    &KIND C
      BASIS_SET DZVP-MOLOPT-GTH
      POTENTIAL GTH-PBE-q4
    &END KIND
    &KIND H
      BASIS_SET DZVP-MOLOPT-GTH
      POTENTIAL GTH-PBE-q1
    &END KIND
  &END SUBSYS
  &DFT
    BASIS_SET_FILE_NAME ./BASIS_MOLOPT
    POTENTIAL_FILE_NAME ./GTH_POTENTIALS
    &MGRID
      CUTOFF 350
      NGRIDS 5
    &END
    &SCF
      MAX_SCF 100
      SCF_GUESS ATOMIC
      EPS_SCF 1.0E-6
      &OT
        PRECONDITIONER FULL_KINETIC
        ENERGY_GAP 0.01
      &END
      &OUTER_SCF
        MAX_SCF 30
        EPS_SCF 1.0E-6
      &END
    &END SCF
    &XC
      &XC_FUNCTIONAL PBE
      &END XC_FUNCTIONAL
    &END XC
  &END DFT
&END FORCE_EVAL
```

atomic coordinat

which basis and PP to use

Basis and Pseudopotential files

Planewave cutoff

Functional to use

First Relax The System: SCF output

```
 SCF WAVEFUNCTION OPTIMIZATION
  ----------------------------------- OT ---------------------------------------
  Allowing for rotations:  F
  Optimizing orbital energies:  F
  Minimizer      : CG                    : conjugate gradient
  Preconditioner : FULL_KINETIC          : inversion of T + eS
  Precond_solver : DEFAULT
  Line search    : 2PNT                  : 2 energies, one gradient
  stepsize       :    0.15000000
  energy_gap     :    0.01000000
  eps_taylor     :   0.10000E-15
  max_taylor     :             4
  mixed_precision    : F
  ----------------------------------- OT ---------------------------------------
  Step     Update method      Time    Convergence         Total energy    Change
  ------------------------------------------------------------------------------
     1 OT CG       0.15E+00   10.1     0.02378927      -453.3605987256 -4.53E+02
     2 OT LS       0.26E+00    6.9                     -467.1228126140
     3 OT CG       0.26E+00   12.8     0.01988127      -470.7669071970 -1.74E+01
     4 OT LS       0.19E+00    6.9                     -478.9120765273
     5 OT CG       0.19E+00   12.9     0.01840474      -479.5432087931 -8.78E+00
     6 OT LS       0.13E+00    6.9                     -483.6122036385
     7 OT CG       0.13E+00   12.8     0.01235887      -484.5052898176 -4.96E+00
     8 OT LS       0.25E+00    6.9                     -487.8838131986
     9 OT CG       0.25E+00   12.8     0.00866152      -489.0337576747 -4.53E+00
    10 OT LS       0.29E+00    6.8                     -491.5792427348
```

First Relax The System: GEO_OPT output

```
--------  Informations at step =     3 ------------
  Optimization Method        =                 BFGS
  Total Energy               =      -495.8615666787
  Real energy change         =        -0.0042200259
  Predicted change in energy =        -0.0029921326
  Scaling factor             =         0.0000000000
  Step size                  =         0.0742559544
  Trust radius               =         0.4724315332
  Decrease in energy         =                  YES
  Used time                  =              400.750

  Convergence check :
  Max. step size             =         0.0742559544
  Conv. limit for step size  =         0.0010000000
  Convergence in step size   =                   NO
  RMS step size              =         0.0187665678
  Conv. limit for RMS step   =         0.0010000000
  Convergence in RMS step    =                   NO
  Max. gradient              =         0.0078329898
  Conv. limit for gradients  =         0.0010000000
  Conv. for gradients        =                   NO
  RMS gradient               =         0.0019762953
  Conv. limit for RMS grad.  =         0.0010000000
  Conv. for gradients        =                   NO
 ---------------------------------------------------
```

Not converged

```
--------  Informations at step =    97 ------------
  Optimization Method        =                 BFGS
  Total Energy               =      -495.8856430646
  Real energy change         =        -0.0000002756
  Predicted change in energy =        -0.0000001832
  Scaling factor             =         0.0000000000
  Step size                  =         0.0007308991
  Trust radius               =         0.4724315332
  Decrease in energy         =                  YES
  Used time                  =              123.284

  Convergence check :
  Max. step size             =         0.0007308991
  Conv. limit for step size  =         0.0010000000
  Convergence in step size   =                  YES
  RMS step size              =         0.0001960784
  Conv. limit for RMS step   =         0.0010000000
  Convergence in RMS step    =                  YES
  Max. gradient              =         0.0000373960
  Conv. limit for gradients  =         0.0010000000
  Conv. in gradients         =                  YES
  RMS gradient               =         0.0000115708
  Conv. limit for RMS grad.  =         0.0010000000
  Conv. in RMS gradients     =                  YES
 ---------------------------------------------------
```

Converged

Simulated STM Using Tersoff-Hamann approximation

- Tersoff-Hamann approximation:
  - Assume spherical (s-wave-function) tip, with low bias applied

$$I = eV \sum_{E_n = E_F + eV}^{E_F} \|\Psi_n(\mathbf{r})\|^2$$

  - Tunnelling current through the tip at $\mathbf{r}$ is proportional to the partial electron density in the energy window between E_F and $E_F + eV$ of the sample at $\mathbf{r}$
- $+V$ probes the conduction band, $-V$ probes the valence band
- From SCF energy calculation, obtain both occupied and unoccupied orbitals that lie within the energy window, sum up and obtains volume data of the tunnelling current at every point in the simulation cell.
- From the volume data of tunnelling current, we can obtain both constant current or constant height images.

STM Input

```
&DFT
  ...
  &PRINT
    &STM
      BIAS -2.0 -1.0 1.0 2.0
      TH_TORB S
      NLUMO 200
    &END STM
    ...
  &END PRINT
&END DFT
```

Biases in V

Type of tip symmetry

Number of unoccupied orbitals to include in the calculation

Tip type

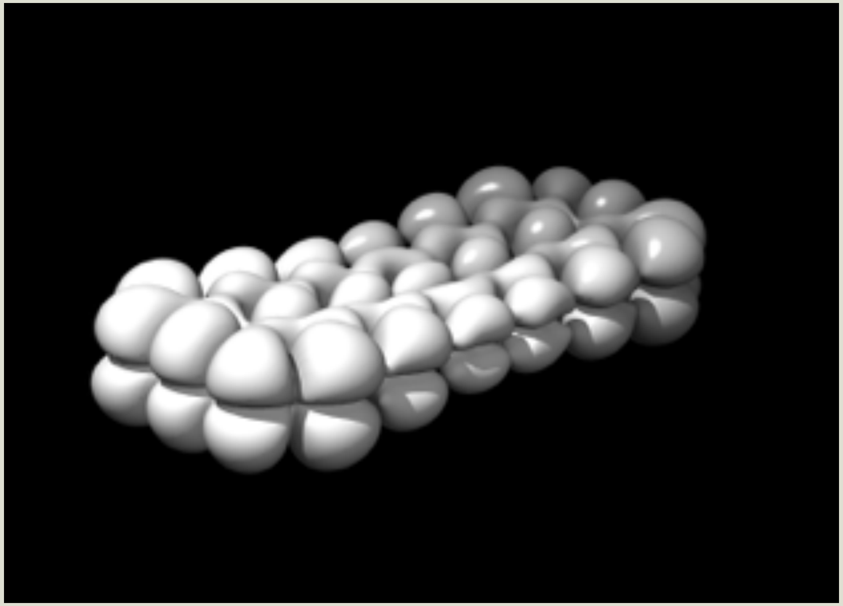
Outputs CUBE files: `GNR_A7_L11_1H_STM-STM_00_00002-1_0.cube`

Job name

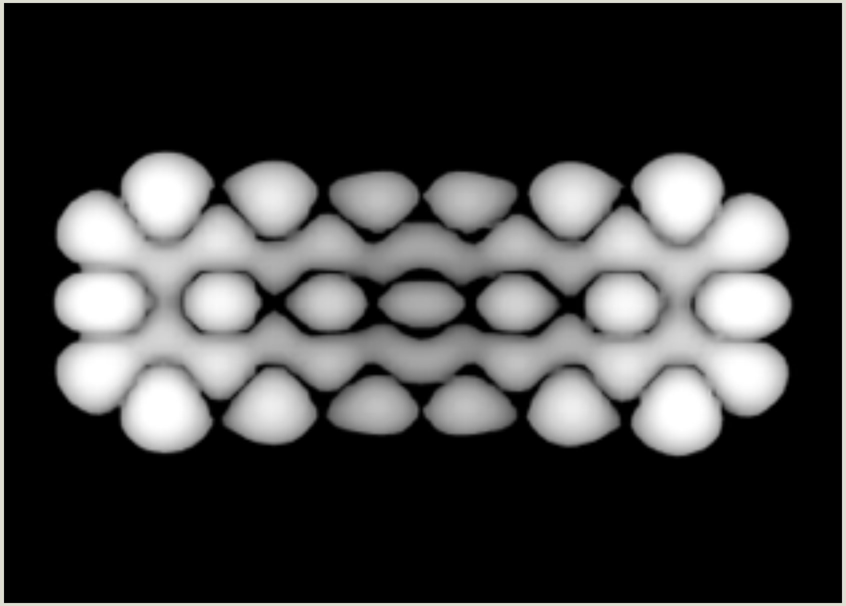
bias index

STM Images

Constant current measurements



Iso-current surface
in volume data

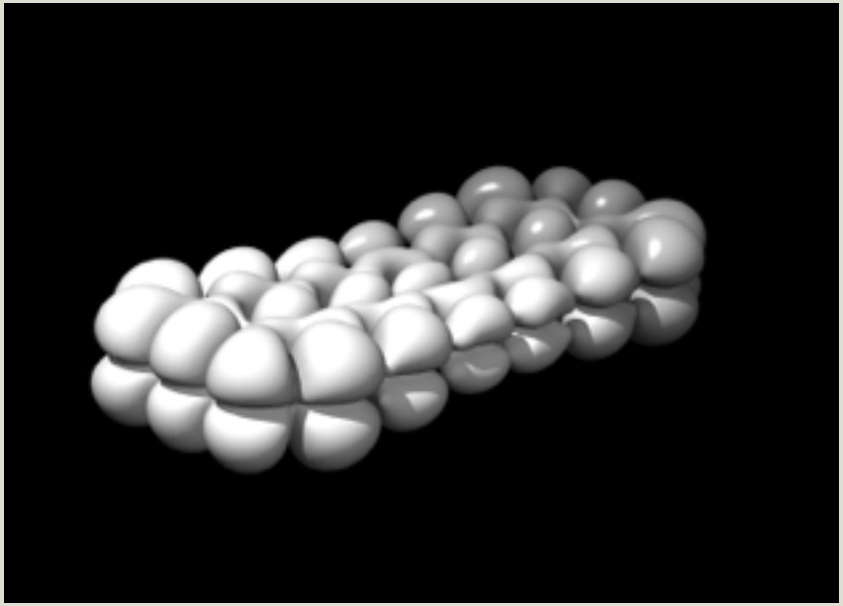


Z-contrast

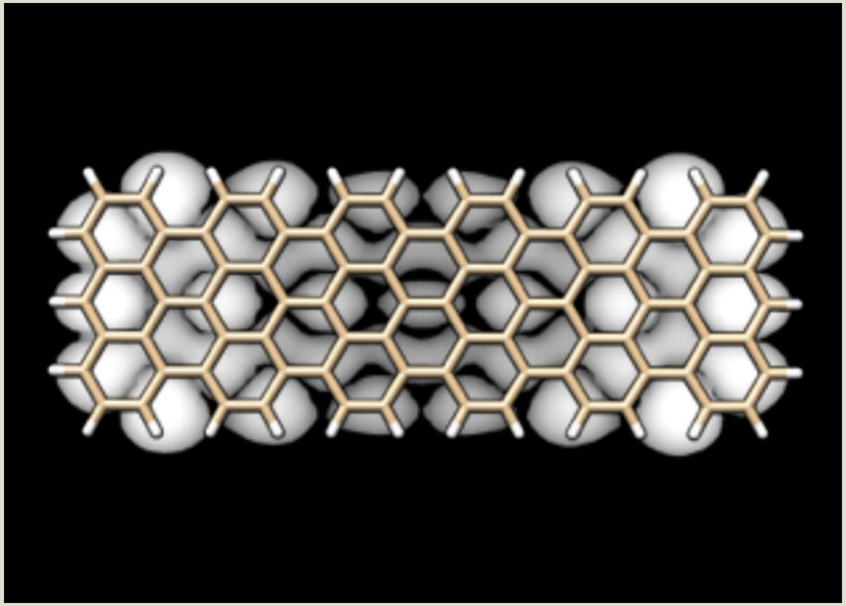
Visualisation using UCSF Chimera, *J Comput Chem. 2004 Oct;25(13):1605-12*

STM Images

Constant current measurements

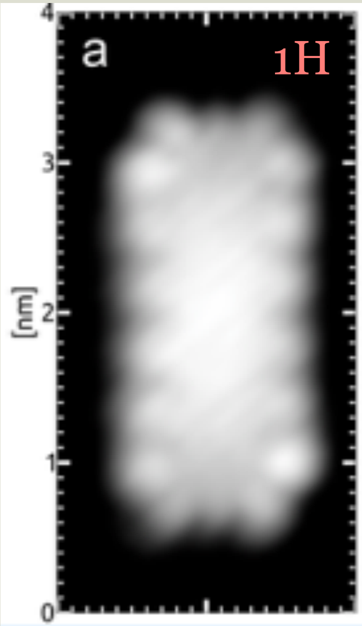


Iso-current surface
in volume data



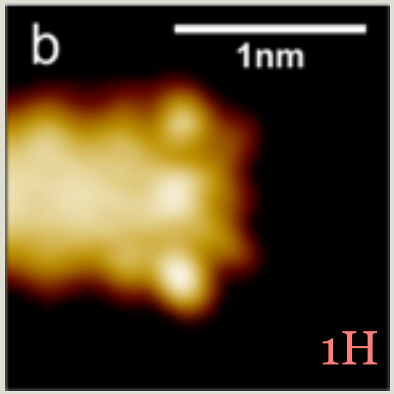
Z-contrast

Visualisation using UCSF Chimera, *J Comput Chem. 2004 Oct;25(13):1605-12*

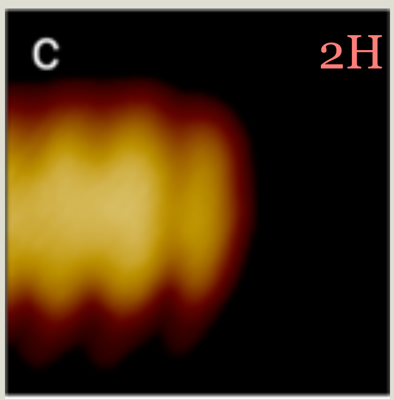


Exp. -1.89V
on Au(111)

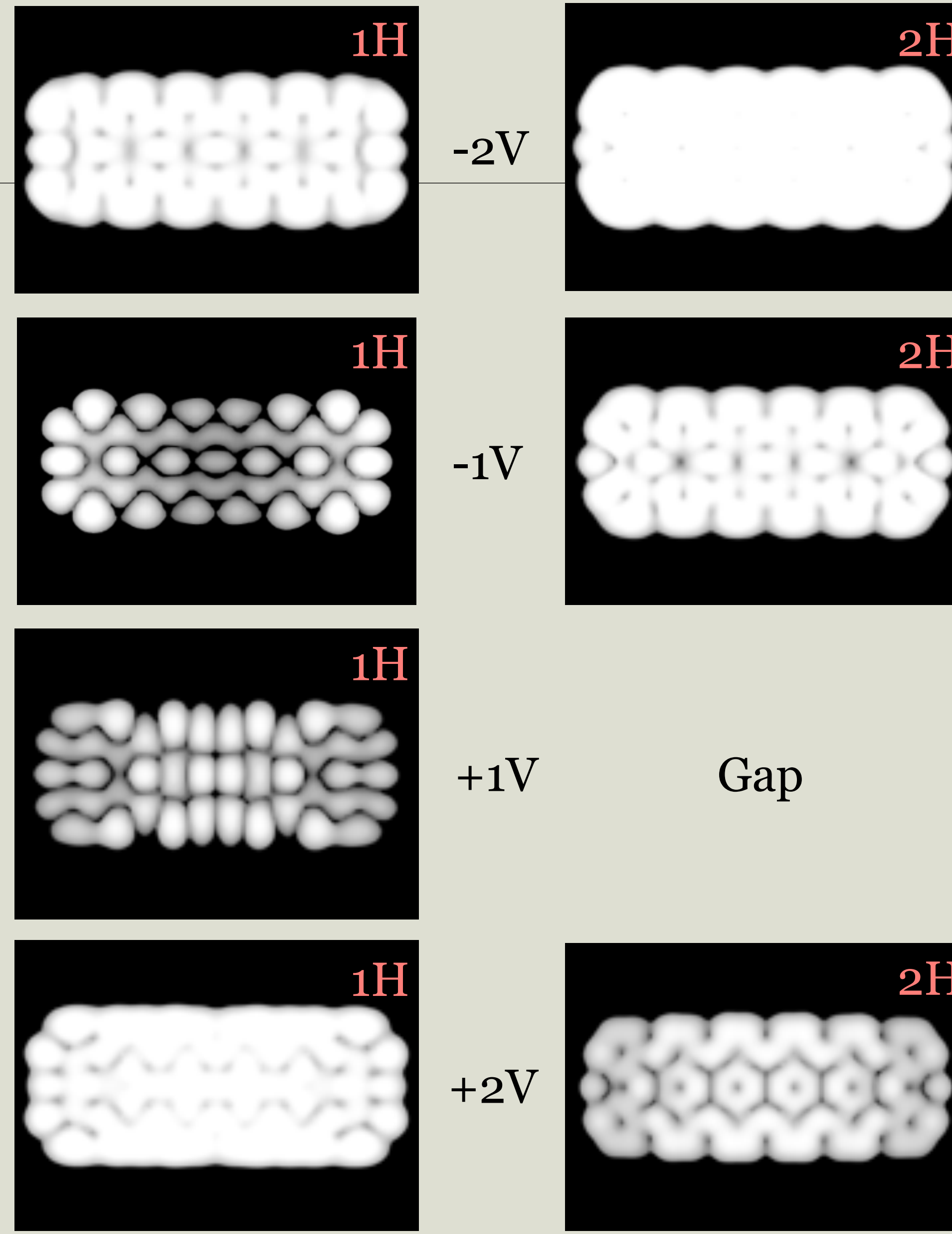
Talirz et al. J. Am. Chem. Soc., 2013, 135 (6), pp 2060–2063



Exp. -0.14V
on Au(111)



Exp. +0.10V
on Au(111)



Visualisation using UCSF Chimera, *J Comput Chem. 2004 Oct;25(13):1605-12*

To View Contributing Orbitals

```
&DFT
  ...
  &PRINT
    ...
    &MO_CUBES
      NHOMO 10
      NLUMO 10
      STRIDE 2 2 2
      WRITE_CUBE T
    &END MO_CUBES
    ...
  &END PRINT
&END DFT
```

NHOMO: Number of occupied orbitals to print out

NLUMO: Number of unoccupied orbitals to print out

STRIDE: How coarse is the output cube file

Outputs CUBE files: `GNR_A7_L11_1H_STM-WFN_00184_1.cube`

Job name

orbital number in order of its energy

To View Contributing Orbitals

Fermi Energy

GNR_A7_L11_1H_STM-WFN_00184_1.cube

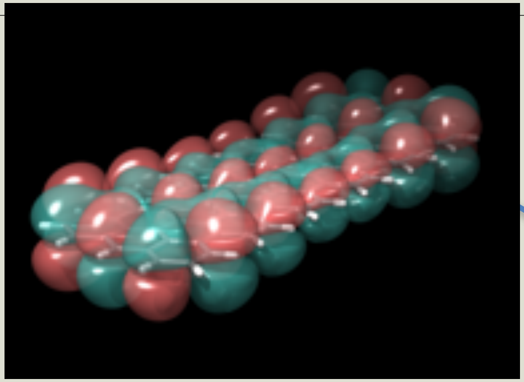
```
-Quickstep-
  WAVEFUNCTION          184  spin    1  i.e. LUMO + 0
  114     0.000000    0.000000    0.000000
  338     0.335951    0.000000    0.000000
  180     0.000000    0.314954    0.000000
  113     0.000000    0.000000    0.335951
    1     0.000000   28.145581   37.666021   18.897261
    1     0.000000   28.177779   33.065110   18.847665
    1     0.000000   28.125830   28.449643   18.831351
    6     0.000000   30.241584   33.058888   18.836194
    6     0.000000   30.187491   28.411612   18.825917
    6     0.000000   30.209663   37.703928   18.861812
    1     0.000000   30.448718   24.358968   18.829013
    1     0.000000   30.495028   41.755734   18.864860
    6     0.000000   31.484281   26.141256   18.824440
    6     0.000000   31.523286   30.736638   18.824450
    6     0.000000   31.536401   35.374029   18.838633
    6     0.000000   31.520564   39.967601   18.850293
```

CP2K main output for STM calculation

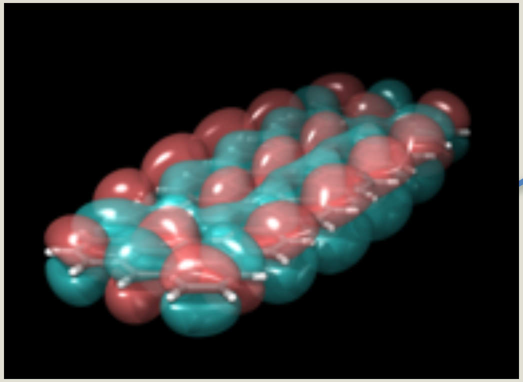
```
STM : Reference energy    -0.133627 a.u.

Preparing for STM image at bias [a.u.]        -0.073499
  Using a total of 7 states

Preparing for STM image at bias [a.u.]        -0.036749
  Using a total of 2 states

Preparing for STM image at bias [a.u.]         0.036749
  Using a total of 1 states

Preparing for STM image at bias [a.u.]         0.073499
  Using a total of 6 states
```

Contributing Orbitals

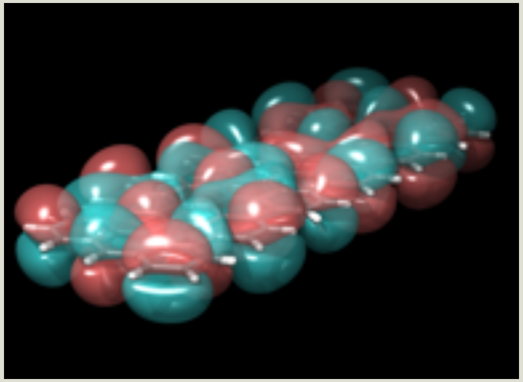
HOMO
E = -0.13362650
Ha

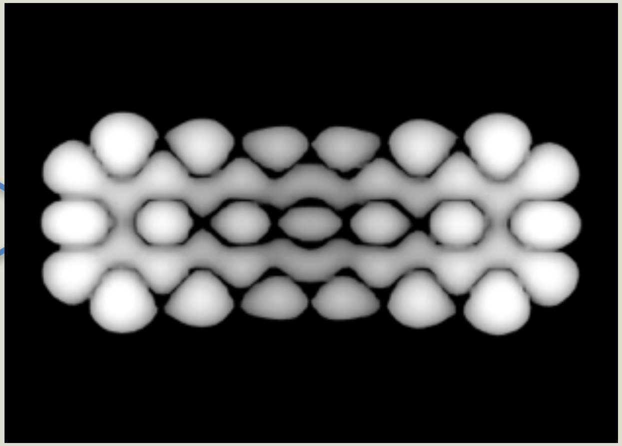


HOMO - 1
E = -0.17065594
Ha

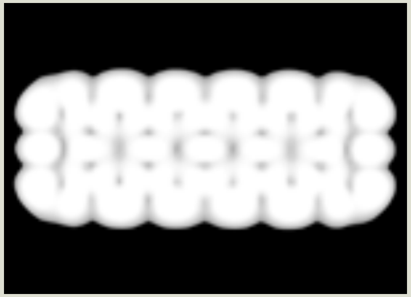


HOMO - 3
E = -0.18709203
Ha





```
Preparing for STM image at bias [a.u.]  -0.036749
   Using a total of 2 states
```



Visualisation using UCSF Chimera, *J Comput Chem. 2004 Oct;25(13):1605-12*

Ensure you have the full band structure

- CP2K currently supports Gamma point calculation only
  - K-point implementation expected to be released early next year.
- One must ensure you have the complete band structure by having a large enough system

```
MULTIPLE_UNIT_CELL 8 8 8
```

- Efficient for large systems:
  - Orbital Transform
  - Filtered Matrix Diagonalisation— up to 10 times speed up with little lose of accuracy for large systems: Please see poster

THANK YOU!

Any Questions?