

# Introduction to AFM Line Scan Simulator

Samuel Ng

## 1 Marcus Hopping Simulator

Marcus theory models the rate of thermally assisted electron transfers. The rate equation for an electron hopping from an initial site to a final site is given by:

$$\nu_{ij} = \frac{|t_{ij}|^2}{\hbar} \sqrt{\frac{\pi}{\lambda k_B T}} \exp\left(-\frac{(\Delta G_{ij} + \lambda)^2}{4\lambda k_B T}\right) \quad (1)$$

Where  $t_{ij}$  is the overlap integral between the DB bound states;  $\lambda$  is the self-trapping energy;  $\Delta G_{ij}$  is the Coulombic energy difference between the configurations before and after electron exchange.

**Implementation** The *MarcusModel* class takes an arrangement of DBs and simulates the charge dynamics over time. Marcus theory is used to determine the intervals between hopping events and which DBs are involved in each hop. Currently, the overlap integrals are approximated by a  $1/r$  falloff with DB distance, a good choice of  $\lambda$  for matching the experimental results was found to be 40 meV (close to the  $\sim 100$  meV discussed), and each DB can be exposed to an arbitrary external electric potential such as from the tip.

## 2 AFM Line Scan Simulation

To attempt to reproduce the experimental AFM line scans, an *AFMLine* class was implemented which simulates the action of scanning the AFM over the DBs in the *MarcusModel*. The line scan is described by the type of scan (read-read, write-right, write-left), the scan rate, the amount of padding on either side of the line, and the number of scans. The class determines the list of upcoming DBs and the times before each measurement and queries the *MarcusModel* class for the relevant charge state of the DBs. In write mode, the tip is primitively modeled by decreasing the energy of the next DB by a set amount (10 meV) when the tip is within a set distance (0.2 nm).

## 3 Current Work

- The tip is a result of a crude parameterization to match some of the results and is not very physically consistent, Lucian's model should be able to fill this gap.
- The number of charges is currently fixed so inclusion of a mechanism for electrons to hop in/out of the surface will need to be added in the future if the surface population can vary.

## 4 Sample Usage

Example simulation setup options can be found at *afm-sim/src/python/demo.py*, executing it yields a line scan plot. The contents of *demo.py* have also been included below for your convenience:

```
1 #!/usr/bin/env python
2
3 from afm import AFMLine
4
5 # set the coordinates of the dangling bonds in lattice units
6 #X = [1, 12, 15, 22, 25]          # 1-2-2
7 X = [1, 8, 10, 15, 17, 24]       # 1-2-2-1
8
9
10 afm = AFMLine(X)                 # create AFM handler for the given DB arrangement
11
12 # set the scan type
13 afm.setScanType(0)               # read-read, set by default
14 #afm.setScanType(1, .01)         # write(right)-read(left) with a strength of 10meV
15 #afm.setScanType(-1, .01)        # write(left)-read(right) with a strength of 10meV
16
17 # set the bias
18 #afm.setBias(.00012)             # 120 ueV/angstrom bias gradient
19
20 # run the AFM and produce the AFM image, image will be saved to tempfig.pdf
21 # Nel:      number of electrons in the system, half filled by default
22 # nscans:    number of line scans, 2 per sweep with read-read else 1
23 # pad:       number of lattice sites included on the [left, right] edge of the scan
24 # srate:     tip speed in angstroms/second
25 afm.run(Nel=3, nscans=200, pad=[3,3], srate=82)
```

If you wish to alter the behavior of the simulation beyond what's shown above, you may tweak the 'magic numbers' in *afm-sim/src/python/marcus.py* within the section *# somewhat magic numbers*.