Simulating dynamics of a bi-stable charge state interacting with an oscillating tip for the purpose of a paper on ‘Quantum dissipation driven by electron transfer within a single molecule investigated by means of atomic force microscopy’

Martin Ondráček, Pavel Malý, Tomáš Mančal, Pavel Jelínek

July 22, 2019

File `AFM_charge-hopping_wavedata.py` contains the code that was used to carry out the simulations shown in our paper (Berger et al., Nature Chemistry 2019). More specifically, it generates data shown in Figure 4 of the paper. The simulated spectra we show in Figure 3 can be also obtained with the code after editing a single line in it, namely removing or commenting out the line that sets `zpoints=1`.

The code requires *Python3* and the *NumPy* library to run. We specifically used it with Python 3.6.2 and Numpy 1.16.1, but we expect it to run with most other versions of the software too, as it uses only pretty standard tools of the language and the library. Generating some plots on the fly also requires *Matplotlib*, but the corresponding lines are now commented out in the code, as we used the code to generate ASCII files with numerical data and then created the plots using external tools.

The code has no user interface. Any input parameters have to be changed by directly editing the source code. The same holds for any adjustments as to what the code outputs and in what format.

The algorithm implements the model described in the Supplementary Online material of our paper, in particular what we call "the full version" of the model.

Output

The output files generated by the code are in general named according to the template *prefix_parameters*`.dat`, where the "prefix" indicates what kind of data the file contains and the part denoted as parameters encodes numerical values of some of the parameters of the model. The names of the parameters used to compose the names of the output files may unfortunately differ from both the names of the corresponding variables inside the code and the names of the quantities we discuss in the paper. Sorry for that, we left the code as we used it ourselves, without any attempt to make it user friendly. The correspondence between the variables in the code and quantities in the paper is indicated in the list below, under section "Physical parameters".

Output files

`de_*.dat` $\Delta E(t) = E_A(t) - E_B(t) + E_0$
`V1_*.dat` $q, V_A(q)$ (potential well on site A)
`V2_*.dat` $q, V_B(q)$ (potential well on site B)
`wave1_*t?.?????.dat` $q, \rho_A(q)$ probability densities on site A for time $t =$?.????? μs
`wave2_*t?.?????.dat` $q, \rho_B(q)$ probability densities on site B for time $t =$?.????? μs
`rate1_*.dat` $t, k_{AB}(t)$, where $k_{AB}(t) = \int k_{AB}(q,t)\rho_B(q)dq$ is the q-integrated immediate rate for the A←B transfer
`rate2_*.dat` $t, k_{BA}(t)$, where $k_{BA}(t) = \int k_{BA}(q,t)\rho_A(q)dq$ is the q-integrated immediate rate for the B←A transfer
`occ1_*.dat` $t, p_A(t)$ (occupation probabilities on site A)
`occ2_*.dat` $t, p_B(t)$ (occupation probabilities on site B)

After removing the line with `zsize=1`, files with spectra will be generated. These files are named `z-df-exc_*.dat` and contain $z_0, \Delta f(z_0), E_{\mathrm{diss}}(z_0)$, for z_0 starting with `leftz` and increasing by a step of `dz`.

Physical parameters

The list below indicates the correspondence between internal variables of the code and the notation that was used for the parameters of the model in the paper.

`kBT` $= k_\mathrm{B}T$ (but see the note about a loop over temperatures below)
`reorgE` $= \Lambda$
`hwrel` $= 2E_\Lambda$
`rate` $= k_0$ (but rescaled to temperature so that the numerical value is true for $T = 1.2$ K while it corresponds to the same transfer matrix element $|M|$ for other temperatures
`Lambda` $= K_\Lambda$
`alpha` $= Q_\mathrm{tip}$

'Technical' parameters (some of them)

height (z_0-)grid

`zsize` = number of points in the $\Delta f(z)$ and E_diss(z) spectra. When set to one, no spectra are generated and instead the temporal evolution of quantities like $\rho_{A,B}(q,t)$, $p_{A,B}(t)$, etc. is written for a tip oscillating around the position given by `z0`.
`leftz` = starting value of `z0`—the mean height around which the the tip oscillates—for calculating the $\Delta f(z_0)$ and $E_\mathrm{diss}(z_0)$ spectra
`dz` = incremental step of the tip height in $\Delta f(z_0)$ and $E_\mathrm{diss}(z_0)$ spectra

temporal (t-)grid

`dt` = time step
`tsteps` = number of time steps in the simulation (so `tsize*tsteps` is the total simulation time)

In general, time dependent quantities are not written out for the whole simulation but rather only for some number of steps at the end of the simulation. This is controlled by parameter `tsize`.

`tsize` = number of time steps at the end of the simulation to be written to the output.

explicit DOF (q-)grid

`qleft` = leftmost (smallest) value of the q-coordinate `dq` = increment of the q-coordinate grid `qsize` = size (number of points) of the q-coordinate grid

The code contains a `for` loop that allows the user to generate results for several different values of some parameter (or more parameters, if one uses a loop over ordered tuples). Currently, it is used for temperature, so the `kBT` variable is set inside the loop and the default value set previously in the code is being ignored.

Units used in the code: 10^{-22} J for energy (but E_diss and $V_{A,B}$ already outputs in meV!), pN for forces, Å for length, elementary charge for electric charge, μs for time.