Charge Transport and Mobility in Organic Semiconductors

Pranay Venkatesh
2019B2A11004P

A report presented for the course
Novel Project (NP)

.

Department of Chemical Engineering
BITS Pilani, Pilani Campus
Rajasthan
May, 2023

Contents

Acknowledgements

1 Introduction and Background

In this report, we will discuss a method we can employ to determine the mobility of charge carriers (electrons and holes) in organic semiconductor materials. Before that it is helpful to review what organic semiconductors are and why they are useful.

1.1 Organic Semiconductors

1.2 BBL Molecules for Gas Sensing

1.3 Challenge of determining mobility

Determining the mobility of charge carriers in materials that do not have long-range crystalline order is rather difficult. They do not obey traditional properties of transport and quantum mechanical effects start to dominate.

Given the disordered nature of the material and the fact that quantum effects dominate, we can no longer use convenient methods such as band structure theory or semiclassical transport equations. The movement and mobility of electrons and holes is no longer simply a function of an excitation from a valence band to a conduction band. To compute mobilities of electrons, we now need to consider how they "hop" from site to site. When we have a bunch of molecules bundled up together, these "sites" are basically the HOMO and LUMO levels of the donor and acceptor of the electron in a given hop.

One of the major challenges is to figure out which path an electron will take when it's at any given site. An electron at a given site has many available sites it can hop to. Each of those sites has some hopping likelihood associated with it. So, now the problem we try to address in this project is: given a morphology of equilibrated molecules, (1) how do we determine the likelihood of hopping from one site to another? , (2) with all hopping rates, how do we determine the mobility of electrons across the bulk of the material?

Addressing the first question, many kinetic models have been developed to try and figure out the hopping rates between two quantum states. The model we use here is the semi-classical Marcus model for hopping rates in a two state quantum system, as described in Section 2. To answer the second question: for taking an aggregate of all the hops for all the electrons, we need to run some probabilistic simulation for our system. Here, we use kinetic monte carlo simulations as described in Section 3.

2 Marcus Theory

3 Kinetic Monte Carlo Model

4 Literature Data for BBL Mobility

5 Candidate Software for Implementation

5.1 Schrodinger

5.2 Votca-XTP

5.3 MorphCT

6 Installing and using MorphCT

MorphCT is the software we finally settle on, since the entire workflow can be carried out in Python and it is completely an open source software.

6.1 Getting python and conda

To install MorphCT, first begin by installing python and anaconda. Understanding some of the basics of python, pip and anaconda is very helpful in working with MorphCT. Anaconda can be installed by running the following commands (in Linux / WSL / your Cyclops account) :

```
curl −sL \
"https://repo.anaconda.com/miniconda/Miniconda3−latest−Linux−x86_64.sh" \ >
"Miniconda3.sh"
```

This adds the Miniconda3 install script to your home directory. Now install miniconda by running:

```
bash Miniconda3.sh
```

6.2 Install MorphCT

Get the morphCT files to your home directory by using the following command (make sure you have git installed) :

```
git clone https://github.com/cmelab/morphct.git
```

Go into the MorphCT directory and create a conda environment for installing the package:

```
cd morphct
```

```
conda env create −f environment.yml
```

This will take some time, but it will set up the requirements for morphct. Now, complete the installation for morphct:

```
conda activate morphct
```

```
pip install −e .
```

The "." at the end of the command is important, since it specifies that you want to install it in that directory. This should complete the installation of morphct.

Now, while in the morphct environment, you can run python scripts and programs that utilise morphct or its dependencies.

7   LAMMPS-to-Mobility Workflow

LAMMPS is not a python-based software nor is it traditionally compatible with the python toolkits that we are using for MorphCT.

Hence for working with MorphCT, we need to convert the LAMMPS morphology into an appropriate file to use it. I have created a workflow to take a snapshot of a LAMMPS molecular dynamics (MD) simulation output. The workflow I developed is described in the package : lammps-carrier-mobility

To use this workflow, start by extracting the coordinate and bond data into separate files called "sorted_coords.data" and "sorted_bonds.data". The data should be sorted by atom number and bond number. To run MorphCT,we need to convert the LAMMPS data files of the morphology into gsd files. To convert files into gsd files, we can use the mbuild package, which can save atoms and molecules as gsd files. The script gsdwriter.py converts the sorted coordinates and sorted bonds files into a gsd file.

To run the script gsdwriter.py, we first require the mbuild package. In your conda environment, use the following command to install mbuild :

```
conda install −c conda−forge mbuild
```

Now, we can run the script by running:

```
python3 gsdwriter.py
```

Running the script takes time. It generates the output file : "system.gsd", for which we can perform the Kinetic Monte Carlo analysis.

The script "kmc-analysis.py" lays out a basic template for calculating the mobility of the system with MorphCT.

8 Results and Discussion

8.1 Toy System : 10 PEDOT molecules