ZMC(Zeolite Monte Carlo) Manual

Anshuman Goswami*

Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, Indiana 46556, United States

E-mail: agoswami@nd.edu

1. Introduction

ZMC is a kinetic Monte Carlo(kMC) code, developed for simulating Standard Catalytic Reduction(SCR) (equation 1), catalyzed by Cu-exchanged Chabazite (Cu-CHA) zeolite framework.

$$4\mathrm{NO}(g) + 4\mathrm{NH_3}(g) + \mathrm{O_2}(g) \rightarrow 4\mathrm{N_2}(g) + 6\mathrm{H_2O}(g) \tag{1}$$

Previous experimental and theoretical investigations for the above system have demonstrated that the SCR catalysis is assisted by Cu ions which are solvated and mobilized by NH_3 under typical reaction conditions. Furthermore, their catalysis is enabled through a redox cycling between Cu species in +1 or +2 oxidation states. Previous studies have also shown that electrostatic tethering of Cu(I) ions to the zeolite framework restricts their mobility and this limits their participation in the oxidation half cycle, where two Cu(I) species diffuse in each other's vicinity and activate an O_2 molecule to form a binuclear Cu(II) intermediate. This impacts the SCR performance of the catalyst, as evidenced by a compositional dependence in apparent rate constants deduced from prior kinetic experiments. To rationalize these observations through a modelling perspective, we write a physically motivated microkinetic model which accounts for the observed behavior.

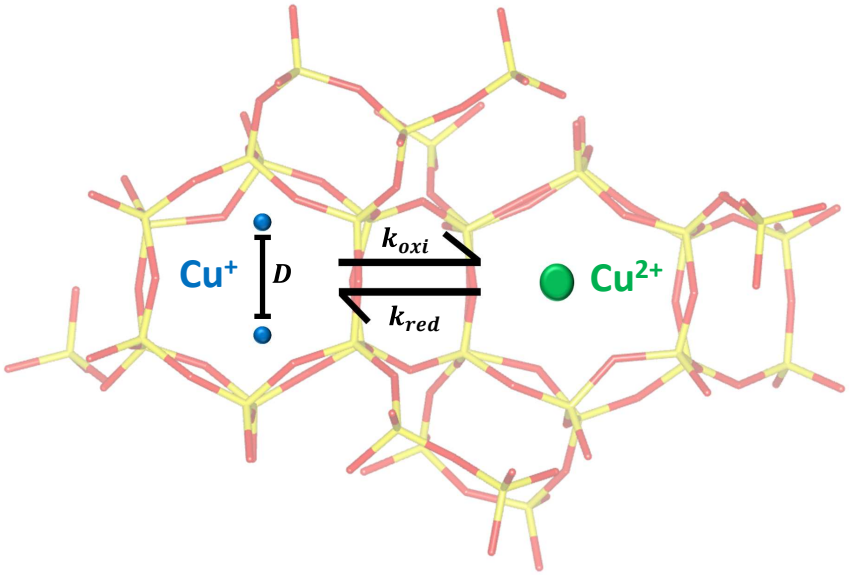

Fig. 1: Physical picture of the model system

$$
\begin{aligned}
2Cu^{+} + \mathrm{O}_2(g) &\xrightarrow{k_{oxi}} 2Cu^{2+} \\
Cu^{2+} + \mathrm{NO}(g) + \mathrm{NH}_3(g) &\xrightarrow{k_{red}} Cu^{+}
\end{aligned}
\tag{2}
$$

The constant interchange of ionic charge of the solvated Cu species advances the SCR catalysis. In addition, we postulate that the reactivity of a pair of Cu(I) species is related to the diffusion distance between them. We introduce this behavior through a pairing probability function λ(d), where d is the inter-ionic distance between 2 Cu(I) species. In principle, this function can have many forms but for our purposes, we assume a shape of a step or sigmoidal function.

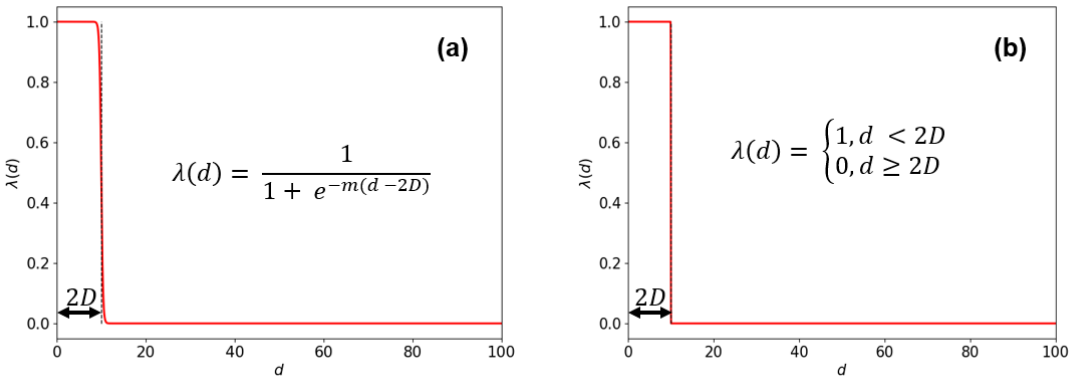

Fig. 2: (a) Sigmoidal function (b) Step function

The point of inflection in both these functions is twice the maximum diffusion distance D. As D increases, more Cu(I) species will become eligible for pairing. Eventually, at very high D, the fraction of Cu(I) species at steady state will be identical to that obtained from solving the microkinetic model by analytical equations that assume mean-field mixing:

$$2k_{oxi}P_{O_2}[Cu_I]^2 = k_{red}[Cu_{II}] \tag{3}$$

Here, the influence of P_{NH_3} and P_{NO} is lumped within k_{red}. Recasting in terms of Cu density ρ_{Cu} and Cu fractions x_I,x_{II}, we have:

$$2k_{oxi}P_{O_2}\rho_{Cu}x_I^2 = k_{red}(1 - x_I) \tag{4}$$

To summarize, the non-mean-field model is characterized by 5 degrees of freedom: (ρ_{Cu}, P_{O_2}, k_{oxi}, k_{red}, D). ZMC requires all these parameters, among others in order to solve the non-mean-field model using kinetic Monte Carlo (kMC) algorithms. Briefly, kMC simulates the time evolution of a system from given initial conditions using stochastic procedures. Post simulation, steady state kinetic outputs can be extracted from the time-dependent trajectories. Since it employs stochastic methodologies, kMC algortithms can be designed to simulate kinetics of complicated systems, such as the one described above, which cannot

be solved through standard ordinary differential equations.

Input parameters are specified to the code through the following input files.

1. `simulation_input.dat`
2. `box_input.dat`
3. `kinetic_input.dat`

After obtaining the inputs, Zmc generates a list of output files containing the relevant statistics. The output files depend on the mode of calculation. All input and output files will be elaborated upon in subsequent sections.

2. Package requirements and installation

Basic python packages:

- Numpy
- Scipy
- Math
- Matplotlib
- Random

In addition, this code also requires the ASE package (https://github.com/rosswhitfield/ase.git). This is used to read the POSCAR files of zeolite frameworks and determines spatial locations of the Copper species.

To install ZMC, clone this repository onto your HOME directory, using git clone https://github.com/Anshu2711/zmc.git. You will also need to set the PYTHONPATH environment variable to the cloned repository's location.

ZMC can be executed at a location by copying the input files to that location and running a python script as follows:

```
1 from zmc import Zmc
2 calc = Zmc()
3 calc.calculate()
```

Post calculation, the location will contain the appropriate output files.

3. Input files

3.1 `simulation_input.dat`

This file comprises of parameters defining the reaction conditions along with intervals and end points for collecting statistical information from the simulation. A sample textfile looks like this:

```
1  ############### SIMULATION_PARAMETERS ################################
2  random_seed              670311654                #random seed generator
3  temperature              473.0                    #Temperature(K) and partial pressures(kPa) of gas-phase reactants
4  O2_pressure              1.01122331209627
5  NO_pressure              1.00
6  NH3_pressure             1.00
7  ############## INTERVALS TO COLLECT DATA ############################
8  snapshots                event 1000           #Interval to collect snapshots.
9  process_statistics       event 1000000        #Interval to collect event statistics.
10 species_numbers          event 1000000         #Interval to collect specie numbers.
11 ############## END POINTS TO COLLECT DATA ###########################
12 max_events               50000000            # Maximum events to simulate
13 max_time                 1000000.0           # Maximum time to simulate(seconds)
14 wall_time                36000.0             # Real time limit for simulation to terminate(seconds)
```

Some points to note:

1. Comments can be added to this input file (and all input files). They must begin with a '#' symbol to be excluded from parsing.

2. All fields have to be specified by integer/floating point numbers. Entering any other information like characters/strings terminates the code.

3. Intervals for collecting data can be eventwise as shown in the above example. Alternately, data can also be collected at fixed simulation time intervals by writing '*time n*' instead of '*event n*' for every field. This enables recording after every n seconds of simulation time.

3.2 box_input.dat

This file comprises of information about the periodic supercell, as well as information about the locations of seeded Coppers. There are two versions of this file depending on the nature of the simulation.

Periodic box(Automatic seeding)

```
################### BOX DIMENSIONS in Angstrom ##########################################
x_length            100       #Dimensions of the orthorhombic box in x/y/z directions
y_length            100
z_length            100
################### Number of Coppers seeded/Seed mode ##################################
P_Copper            0.35    #Density of coppers to be seeded (/1000 Ang^3)
Cu1_frac            0.0       #Fraction of seeded coppers in +1 oxidation state
Seed_mode           auto      #Mode of seeding
```

Periodic box(Manual seeding)

```
################### BOX DIMENSIONS in Angstrom ##########################################
x_length            100       #Dimensions of the orthorhombic box in x/y/z directions
y_length            100
z_length            100
################### Number of Coppers seeded/Seed mode ##################################
Seed_mode           manual      #Mode of seeding
1 23.4 45.3 56.3     # Oxidation state, x coord, y coord, z coord
2 32.6 41.3 59.4
3 84.3 35.9 45.3
4 34.2 23.1 98.2
```

This mode allows the user to seed the Coppers at pre-specified locations and with pre-specified oxidation states.

CHA supercell

```
###################### CHA periodic unit repetitions ###################################
a_reps           9         # Repetitions of 36T site supercell along a,b,c directions
```

```
3  b_reps            9
4  c_reps            9
5  ################## Number of Coppers seeded/Seed mode ################################
6  P_Copper          0.35    #Density of coppers to be seeded (/1000 Ang^3)
7  Cu1_frac          0.0
8  Seed_mode         CHA      # Coppers seeded at Si locations following Lowenstein's rule. It can be replaced by AFX or MON or FER
```

In this option, the code randomly seeds Cu at Si locations in a CHA supercell which is repeated along a,b,c lattice vectors with number of repetitions specified by the 'reps' fields. The seeding is done in a manner obeying Lö wenstein's rule; Coppers are not seeded at nearest neighbouring Si locations.

3.3 `kinetic_input.dat`

This file provides information about the type of kMC calculation: steady state or transient oxidation. Also, kinetic information about the reduction and oxidation events are specified, along with the probability function for pairing.

Steady state with step probability function

```
Kinetic_mode      steady_state              #Type of calculation
##########################################
A_reduction       1.0E+10                   #Reduction prefactor
Ea_reduction      0.00                      #Reduction activation energy
Reduce_mode       single                    #Mode of reduction for Cu(II)
##########################################
A_pairing         1.0E+10                   #Pairing/Oxidation prefactor
Ea_pairing        0.00                      #Pairing/Oxidation activation energy
Decay             step                      #Decay function
Step_cutoff       10.0                      #Diffusion cutoff
Statistics        False
```

If statistics field is True, then two additional output files are written: `Static_distances.txt` which contains information about distances between all Cu(I) pairs present and `Distance_statistics.txt` which records distances of all Cu(I) pairs which participated in oxidation events.

Steady state with sigmoidal probability function

```
Kinetic_mode      steady_state              #Type of calculation
##########################################
A_reduction       1.0E+10                   #Reduction prefactor
Ea_reduction      0.00                      #Reduction activation energy
Reduce_mode       single                    #Mode of reduction for Cu(II)
##########################################
A_pairing         1.0E+10                   #Pairing/Oxidation prefactor
Ea_pairing        0.00                      #Pairing/Oxidation activation energy
Decay             sigmoidal                     #Decay function
Sigmoid_cutoff    10.0                      #Diffusion cutoff
```

```
Sigmoid_slope      5.0                    # Slope of sigmoid at point of inflection
Statistics         False
```

Transient oxidation using a step probability function

```
Kinetic_mode       transient_oxidation
##########################################
A_pairing          52.103
Ea_pairing         0.00
Decay              step
Step_cutoff        10.0
Statistics         False
```

4. Output files

4.1 `error_output.txt`

This file is produced whenever the code terminates prematurely due to insufficient or inappropriate information provided in the input files. The mistake in either case is noted in this file. Here is an example if `A_reduction` parameter in `kinetic_input.dat` is not entered as a floating point number.

```
1 The A_reduction isnt a numeric value. Please re-enter
```

If all parameters are entered correctly, then this file will be removed upon code execution and output files, as described subsequently, are written post-completion of the code.

4.2 `simulation_output.txt`

This file contains a summary of the calculation times incurred by the code, both in terms of simulation time as well as real time.

```
1 The simulation time taken by the code is 16917.91270841319 seconds
2 The total calculation time taken by the code is 60.00351901911199 seconds
```

4.3 indices.txt

This output file contains indices of the seeded Cu locations within the CHA supercell, as represented in the Atomic Simulation Environment (ASE) interface.

```
471  5634  7892  14013  19452  601  5595  21296  2566  13173  19673  14537  938  10266  4355  3445
```

4.4 raw_outputs.txt

This output file appears only if the mode of calculation was set as steady state. It comprises of the number of Copper species against simulation time and number of events.

Events	Time	CuI	CuII
250	0.000000000000	244	261
320	10.000000000000	275	230
563	20.000000000000	344	161
765	30.000000000000	372	133
962	40.000000000000	338	167
1169	50.000000000000	341	164
..	..	..	..
..	..	..	..

This information enables the user to compute the fractions of each Copper species at steady state.

4.5 `event_outputs.txt`

This output file appears only if the mode of calculation was set as steady state. It comprises of the information about number of oxidation and reduction events against the number of events and simulation time.

Events	Time	Oxi_events	Red_events
250	0.00000	001	249
548	100.00000	005	543
584	200.00000	016	568
615	300.00000	027	588
634	400.00000	032	602
650	500.00000	039	611
686	600.00000	049	637
708	700.00000	056	652
714	800.00000	058	656
736	900.00000	068	668
773	1000.00000	079	694
795	1100.00000	087	708
..	..	..	..
..	..	..	..

This information enables the user to compute the reaction rate at steady state.

4.6 raw_outputs_TOX.txt

This file is written if the mode of calculation was set as transient oxidation. It describes the number of Copper species against simulation time and number of events.

Events	Time	CuI	CuII
000	0.000e+00	1.000	0.000
001	6.592e-02	398.000	2.000
002	9.397e-02	396.000	4.000
003	3.202e+00	394.000	6.000
004	3.313e+00	392.000	8.000
005	3.994e+00	390.000	10.000
006	4.534e+00	388.000	12.000
007	4.835e+00	386.000	14.000
008	7.423e+00	384.000	16.000
009	8.230e+00	382.000	18.000
010	9.226e+00	380.000	20.000
011	9.733e+00	378.000	22.000
012	1.197e+01	376.000	24.000
013	1.266e+01	374.000	26.000
014	1.491e+01	372.000	28.000
015	1.519e+01	370.000	30.000
016	1.581e+01	368.000	32.000
017	1.895e+01	366.000	34.000
018	1.943e+01	364.000	36.000
019	1.955e+01	362.000	38.000
020	1.964e+01	360.000	40.000
021	2.133e+01	358.000	42.000
022	2.256e+01	356.000	44.000
023	2.399e+01	354.000	46.000
..	..	..	..
..	..	..	..

This information enables one to compute the residual fraction of Cu(I)/Cu(II) post-completion.

4.7 `event_outputs_TOX.txt`

This file is written if the mode of calculation was set as transient oxidation. It describes the number of oxidation events against simulation time and number of events.

Events	Time	Oxi_events	Red_events
000	0.000e+00	000	000
001	6.592e-02	001	000
002	9.397e-02	002	000
003	3.202e+00	003	000
004	3.313e+00	004	000
005	3.994e+00	005	000
006	4.534e+00	006	000
007	4.835e+00	007	000
008	7.423e+00	008	000
009	8.230e+00	009	000
010	9.226e+00	010	000
011	9.733e+00	011	000
012	1.197e+01	012	000
013	1.266e+01	013	000
014	1.491e+01	014	000
015	1.519e+01	015	000
016	1.581e+01	016	000
017	1.895e+01	017	000
018	1.943e+01	018	000
019	1.955e+01	019	000
020	1.964e+01	020	000
021	2.133e+01	021	000
022	2.256e+01	022	000
023	2.399e+01	023	000
024	2.409e+01	024	000
025	2.606e+01	025	000
026	2.801e+01	026	000
027	2.980e+01	027	000
..	..	..	..
..	..	..	..