



Introduction to Kinetic Monte Carlo Method

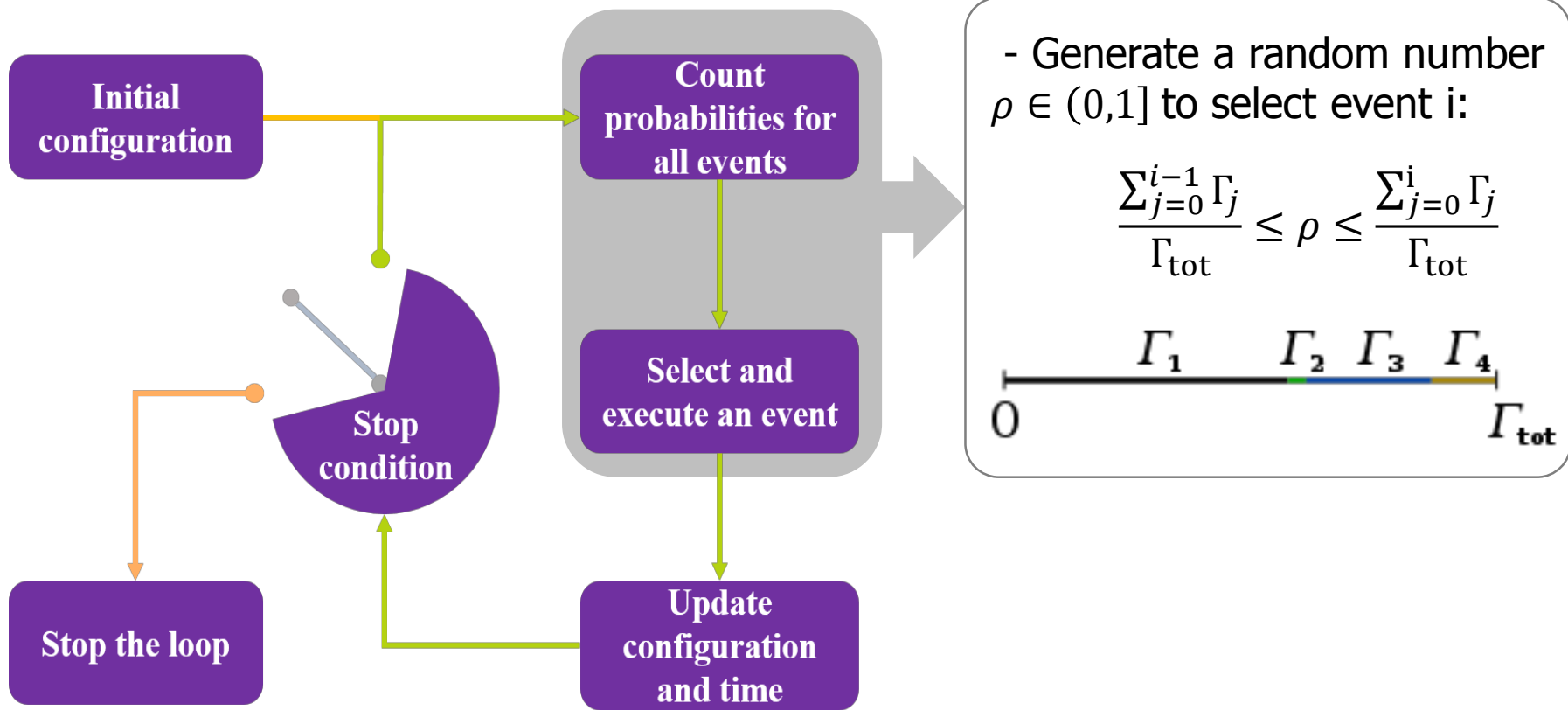
Matias A. Quiroga, Yinghui Yin, Alejandro A. Franco



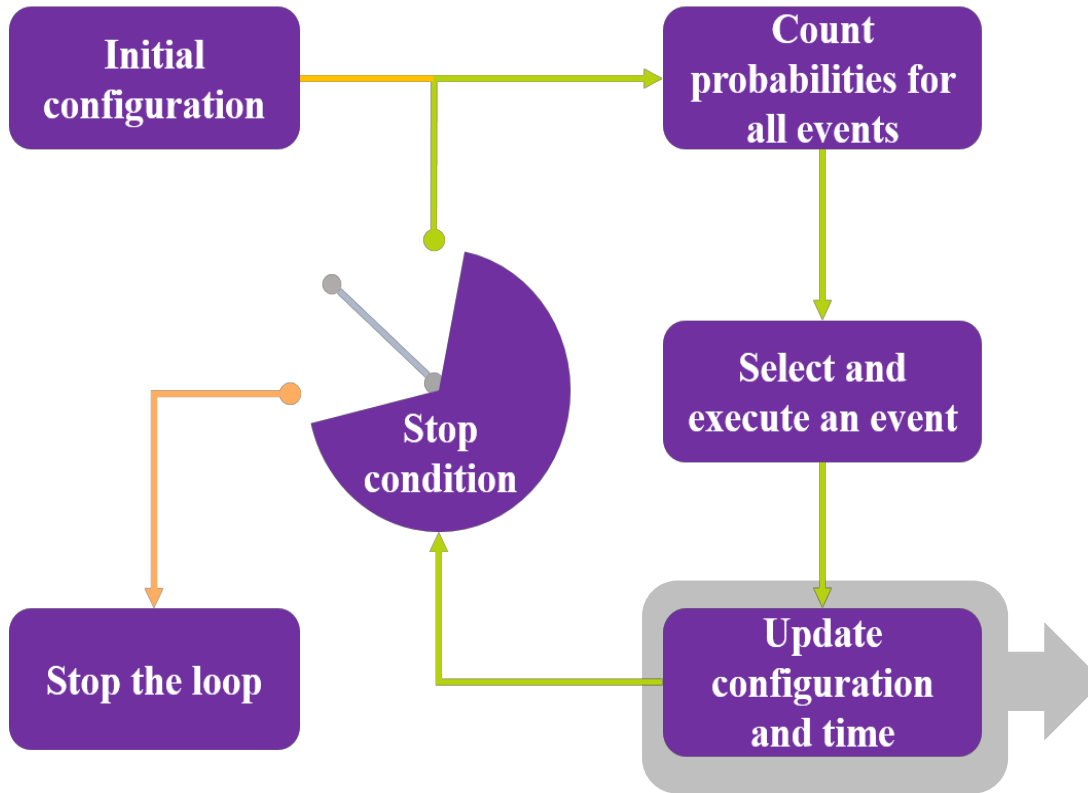
Université de Picardie Jules Verne (Amiens, France)
Laboratoire de Réactivité et Chimie de Solides • CNRS
Research Network on Electrochemical Energy Storage
ALISTORE European Research Institute



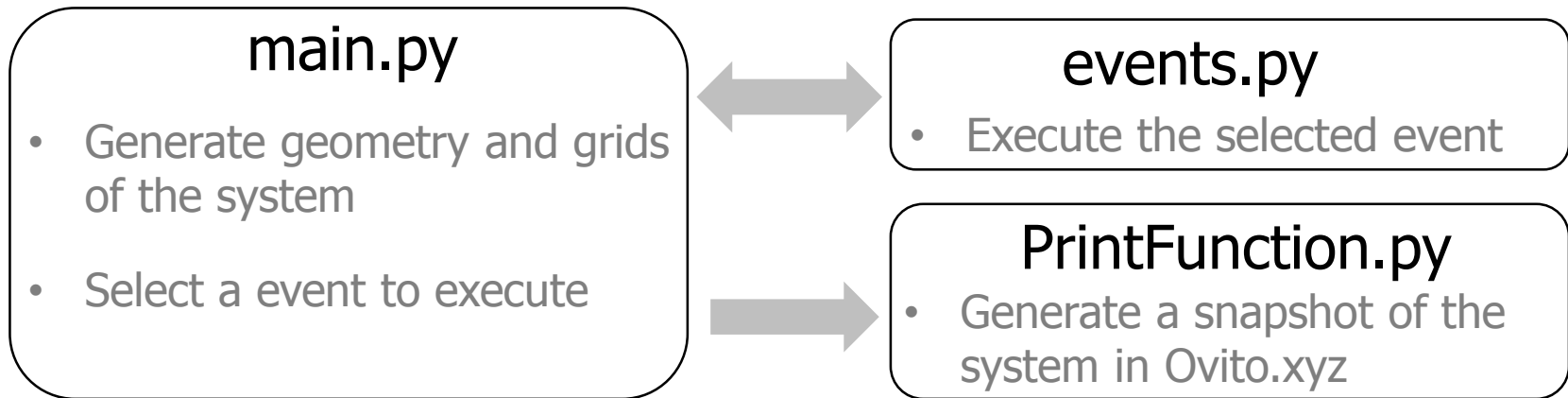
KMC Algorithm



KMC Algorithm

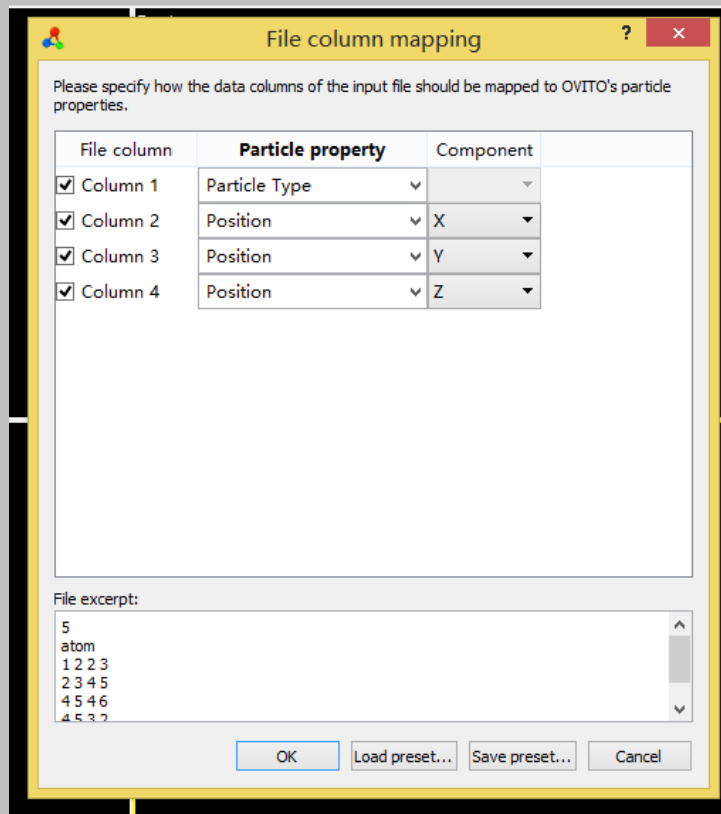


Architecture of the KMC Code

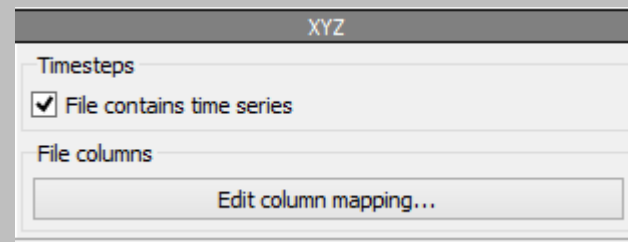


Exercise - 1

1. Place in the same folder the files main.py, PrintFunction.py and events.py. Run the code with command:
> `python main.py`
2. Open the file Ovito.xyz with OVITO, then you can visualize your simulation results.



Set the column mapping



Import time series



<Play> to see the video



Exercise - 2

1. Edit the main.py with command:
> edit main.py

Then change the set of diffusion rates, the amount of species and the size of the box to see their impacts.

```
25
26
27 # USER input Parmeter start
28
29 tTOT = 100      # total KMC time
30 Lx = 30        # x grid size
31 Ly = 30        # y grids size
32
33 # build the board
34 board = np.zeros((Lx, Ly), dtype=int) # here we build a board plenty of 0, and waiting to be filled with integer numbers
35
36
37 Dif_A = 1.3    # difussion rate for A
38 Dif_B = 0.4
39 Dif_C = 0.4
40
41 R_AB = 2      # chemical reaction between A and B
42
43 speciesA = 20 # initial amount of species A in the board
44 speciesB = 20 # initial amount of species B in the board
45
46 RF = 100     # recording frequency, the code just prints 1/RF snapshots
47
```



Exercise - 3

1. Uncomment line 209 and run the simulation with the command:

> python main.py >species.txt

Then plot the species.txt with graph software (e.g excel, origin..)

```
198 dt = -log(rand2)/(sumTOT)           # time step
199 t = t + dt                          # update the clock
200 cumulative = cumulative + 1          # this variable grows until we record the step
201
202 if cumulative == RF:                 # If we reach the recording frequency, we print our system
203     cumulative = 0                   # reset the variable
204     call = call + 1                 # it grows call
205     PrintFunction(board,call,4)      # it calls the printer function, in order to generate an *.xyz file
206
207
208 #uncoment the following line, if you want to see those variables in the screen
209 #print t, ' ',Ap, ' ',Bp, ' ',Cp
210 #print board # be carefull big baords cannot be seen in the screen (< 32 x 32)
211
212
```





Exercise - 4

1. Edit the main.py file in order to have all the species filling half of the domain.
2. Add some randomly distributed defects/ pined places .



Hint : a new specie (id=4) that is not able to diffuse

3. Add a new mobile specie D(id =5) into the system.
4. Create new chemical reaction between B and C to form D.



Exercise - 5

- Play with the code, find your own ideas. Can imagine how to transform it to a 3D KMC code?