```
import math
import random
import bisect
import numpy as np
import copy as copy
import matplotlib.pyplot as plt

R = 1 # interatomic spacing
num_atoms = 100

state = [0 for x in range(num_atoms)]
rate = [0 for x in range(num_atoms)]

def generate_rates():
    for k in range(num_atoms):
        list_of_atoms = list(range(num_atoms))
        list_of_atoms.remove(k)

        interaction_sum = 0
        for j in list_of_atoms:
            interaction_sum += state[j]/(abs(j-k)**6)

        rate[k] = 1/(1 + (R**12) * (interaction_sum**2))

def get_jump_time():
    return -math.log(random.random())/sum(rate)

def get_jump_atom():
    cum_rate = np.cumsum(rate) / sum(rate)
    return bisect.bisect_left(cum_rate, random.random())

generate_rates()

duration = 500

last_jump_time = 0
times = []
total_state = []

while (last_jump_time < duration):
    last_jump_time += get_jump_time()
    times.append(last_jump_time)

    flipped_atom = get_jump_atom()
    state[flipped_atom] = 1 if state[flipped_atom] == 0 else 0
    total_state.append(copy.copy(state))

    generate_rates()

# plot data
t = []
p = []

for i in range(len(times)):
    t.extend([times[i] for n in range(total_state[i].count(1))])
    p.extend([index for (index, value) in enumerate(total_state[i]) if value == 1])

plt.scatter(t, p, color='black',marker=',',lw=0, s=1)
```