Michał Gradzik 253880

Big Data Analytics

Kac’s model

The task was to make an algorithm which calculate Kac’s model. This model is a ring with N sites. Each of this sites can be either black or white. On start t=0 every site is black

For N = 100, μ = 0.1 and time = 500 I got:

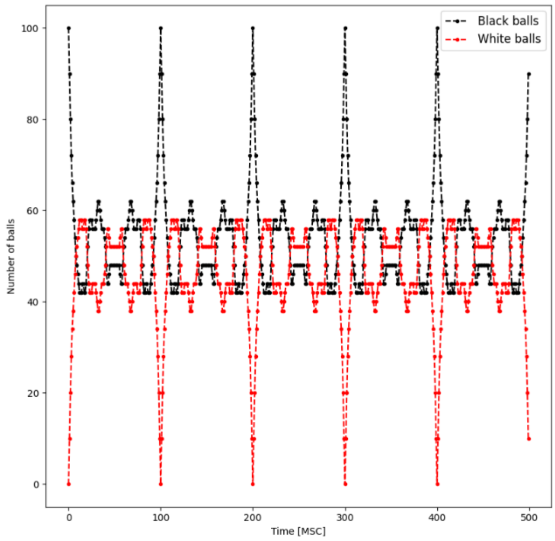


Fig. 1 Graph representing how number of black and white balls is changing

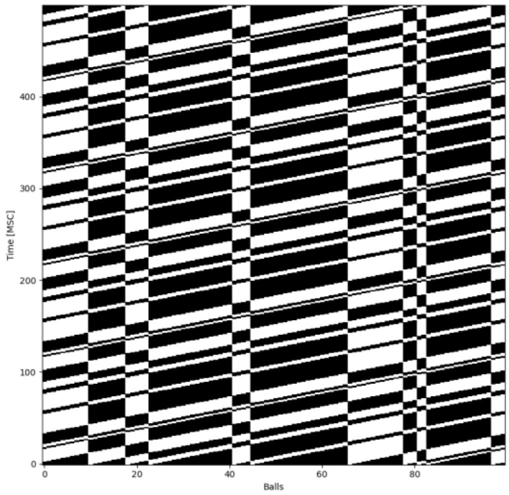


Fig. 2 Graphical representation of change of a model over time

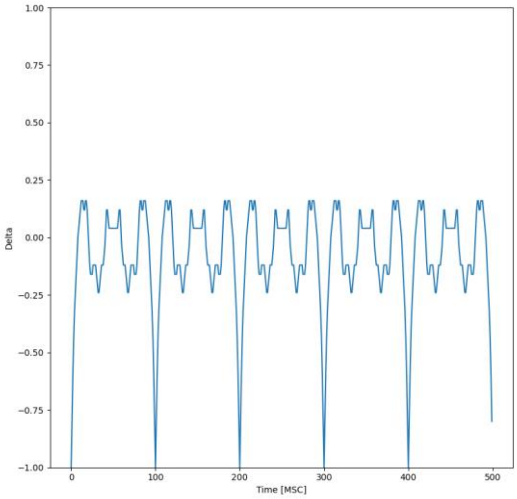


Fig. 3 Changing of delta over time (overall color of the ring)

For N = 800, µ = 0.2 and time = 5000 I got:

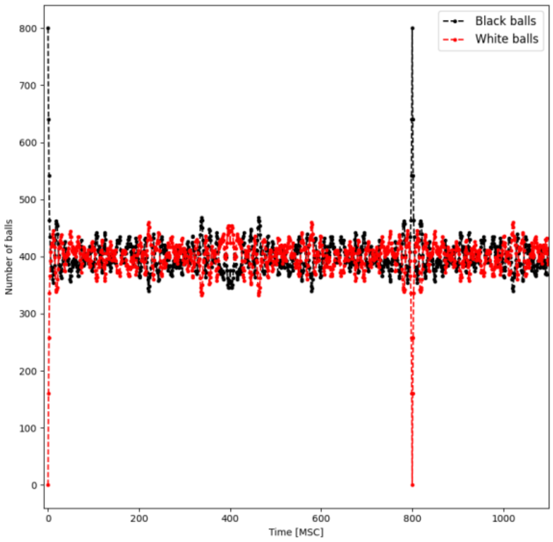


Fig. 4 Graph representing how number of black and white balls is changing (this graph was zoomed to make it more readable)

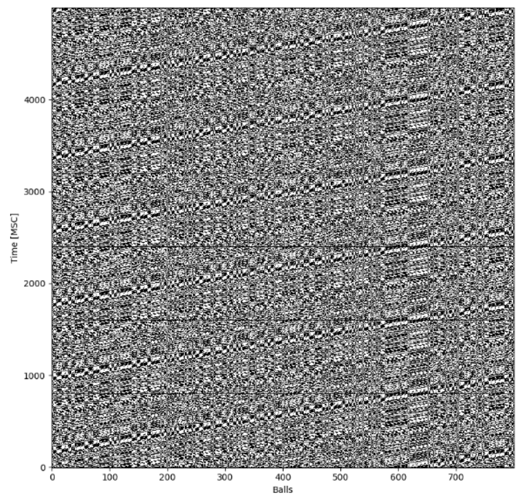


Fig. 5 Graphical representation of change of a model over time

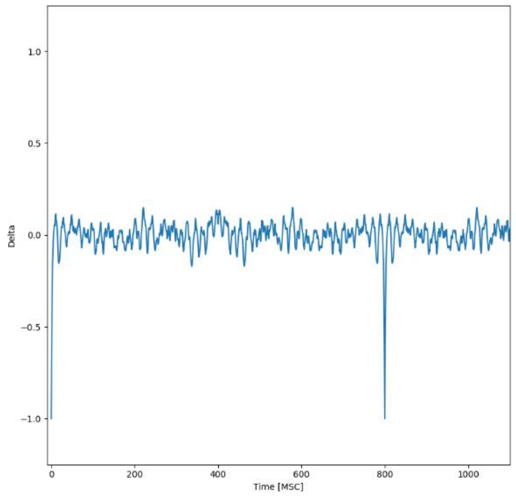


Fig. 6 Changing of delta over time (overall color of the ring)

For N = 800, μ = 0.05 and time = 5000 I got:

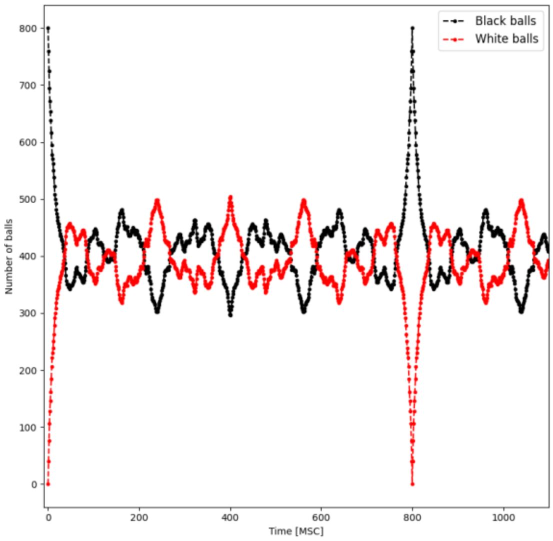


Fig. 7 Graph representing how number of black and white balls is changing

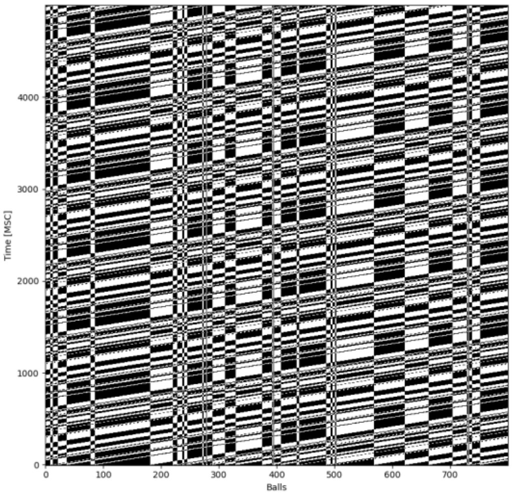


Fig. 8 Graphical representation of change of a model over time

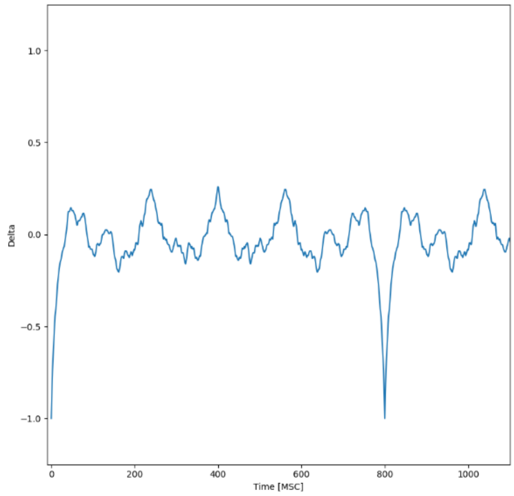


Fig. 9 Changing of delta over time (overall color of the ring)

For N = 2000, μ = 0.1 and time = 10000 I got:

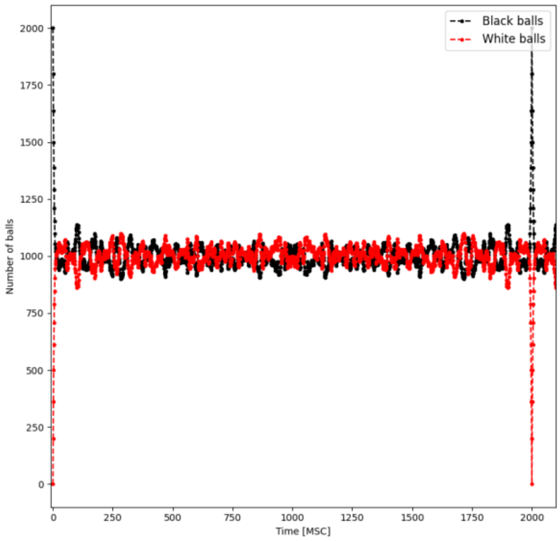


Fig. 10 Graph representing how number of black and white balls is changing

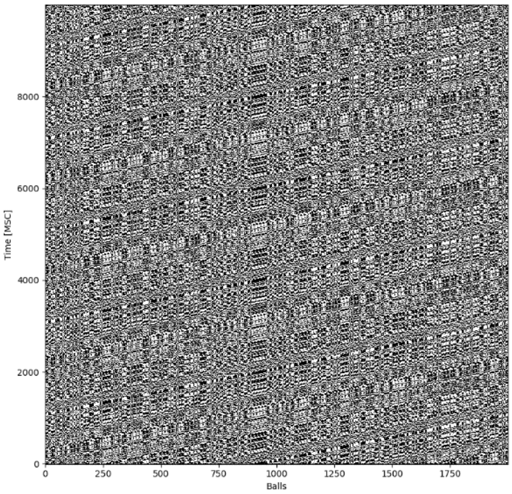


Fig. 11 Graphical representation of change of a model over time

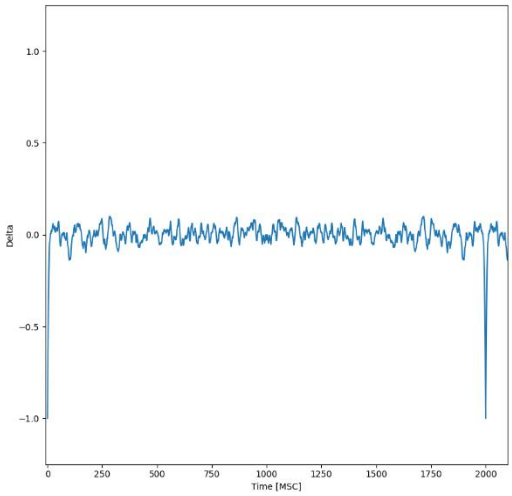


Fig. 12 Changing of delta over time (overall color of the ring)

For N = 2000, μ = 0.01 and time = 10000 I got:

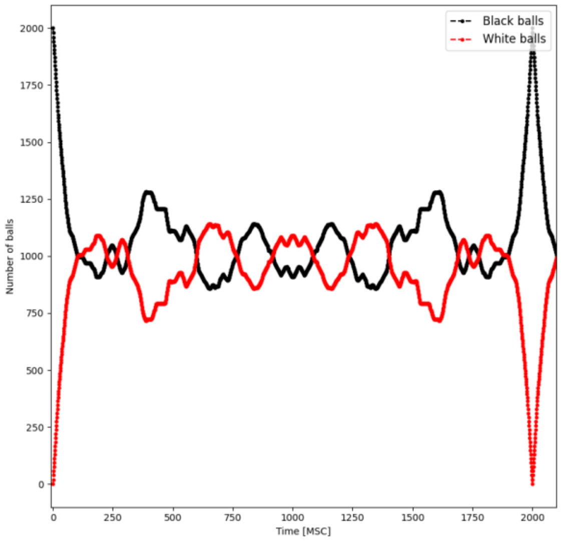


Fig. 13 Graph representing how number of black and white balls is changing

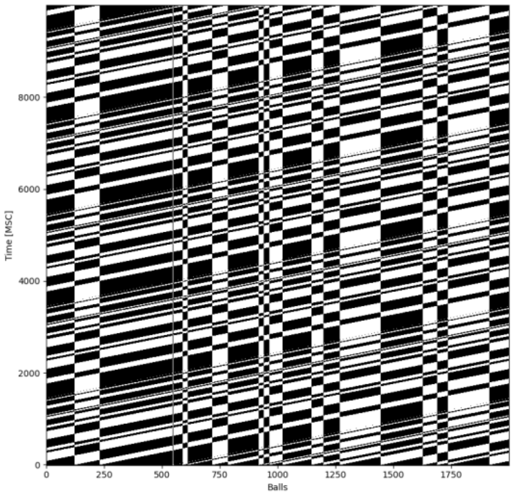


Fig. 14 Graphical representation of change of a model over time

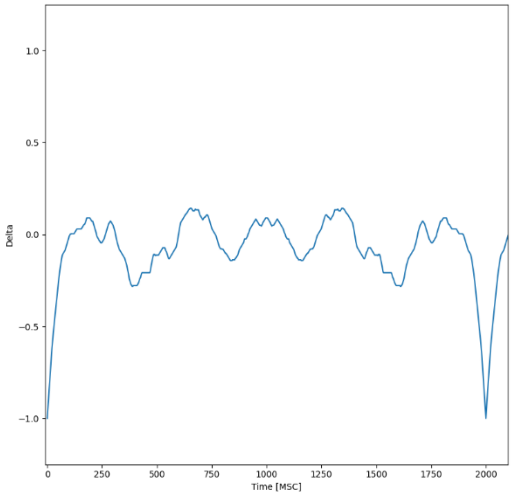


Fig. 15 Changing of delta over time (overall color of the ring)

We can see that when MCS = N we are back at the begging and whole model repeats.

We can see that more active sites make heatmap more dense.

In my simulation I was moving active sites not balls which gives the same graphs and results.

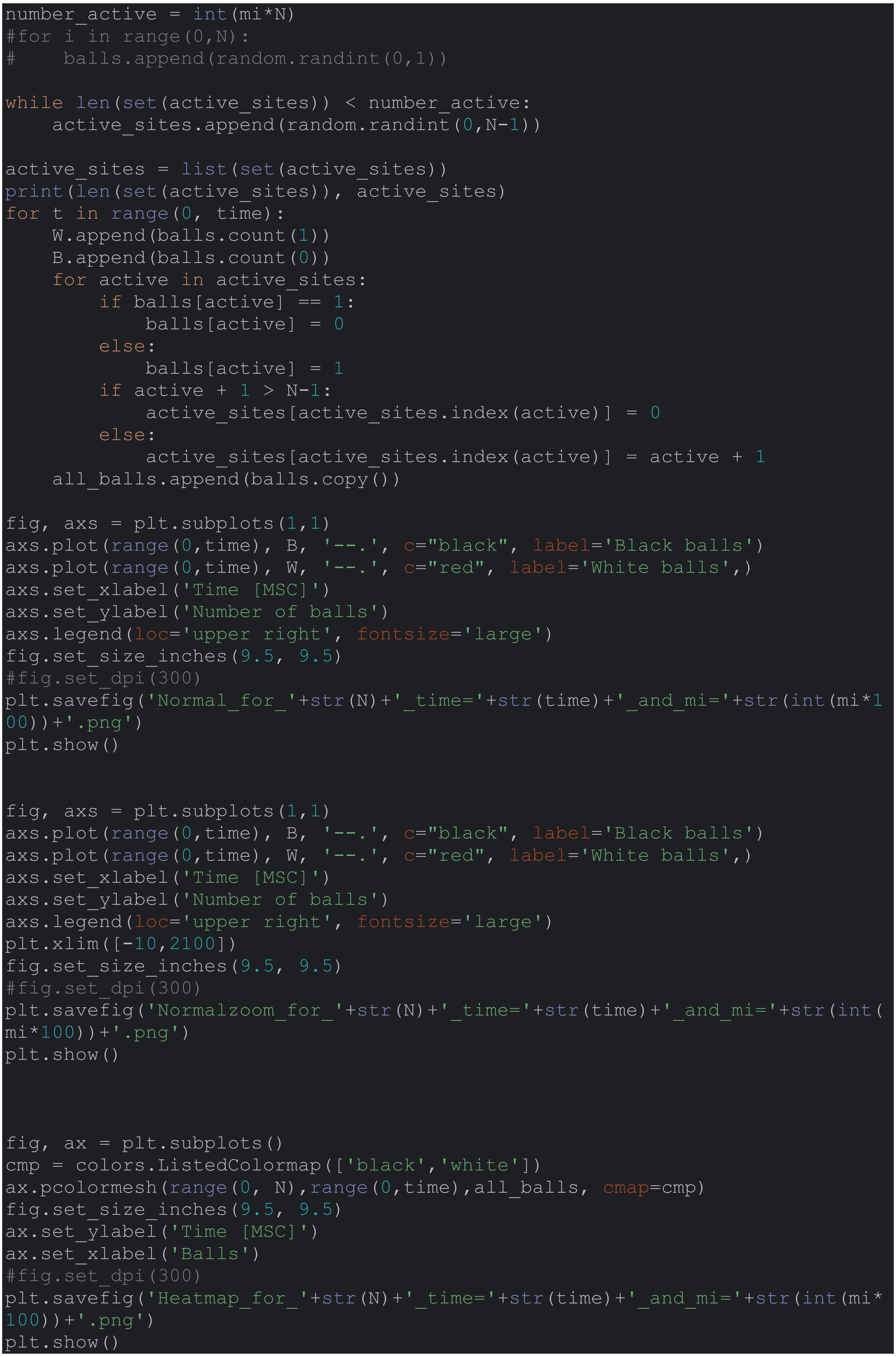
In simulation I used 2 loops with variables: t, active.

Variable t was used to generate each Monte Carlo step. In each Monte Carlo step, I was counting how many black and white balls I have and making switches between black and white. Variable active was used for switching color of ball and moving active sites.

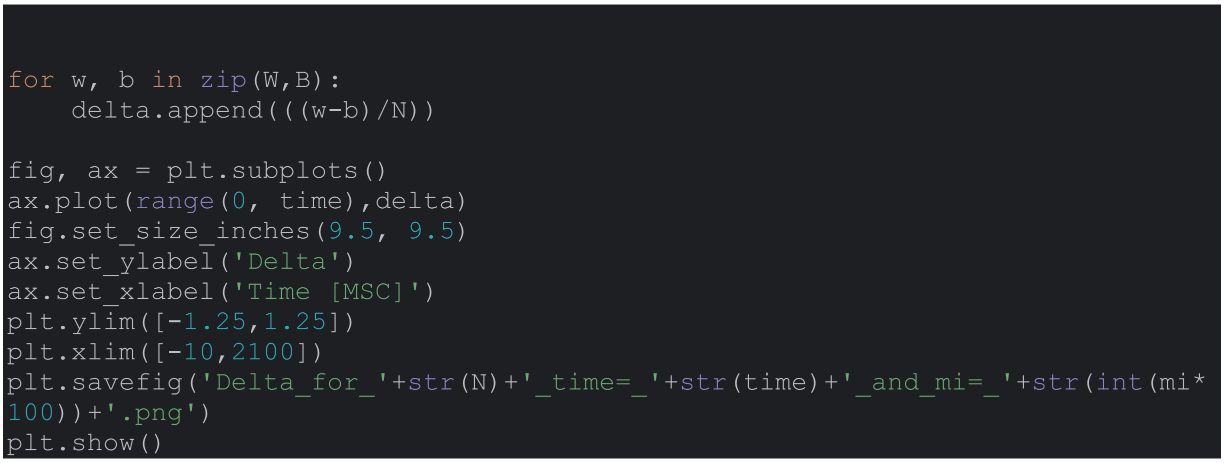
The rest of the code is for plotting graphs and heatmaps.

Source code:

```
import random
import matplotlib.pyplot as plt
from matplotlib import colors
import numpy as np
import math


N = 2000
mi = 0.01

balls = [0]*N
time = 10000
W = []
B = []
all_balls = []
delta = []
active_sites = []
```



```
number_active = int(mi*N)
#for i in range(0,N):
#    balls.append(random.randint(0,1))

while len(set(active_sites)) < number_active:
    active_sites.append(random.randint(0,N-1))

active_sites = list(set(active_sites))
print(len(set(active_sites)), active_sites)
for t in range(0, time):
    W.append(balls.count(1))
    B.append(balls.count(0))
    for active in active_sites:
        if balls[active] == 1:
            balls[active] = 0
        else:
            balls[active] = 1
        if active + 1 > N-1:
            active_sites[active_sites.index(active)] = 0
        else:
            active_sites[active_sites.index(active)] = active + 1
    all_balls.append(balls.copy())

fig, axs = plt.subplots(1,1)
axs.plot(range(0,time), B, '--.', c="black", label='Black balls')
axs.plot(range(0,time), W, '--.', c="red", label='White balls',)
axs.set_xlabel('Time [MSC]')
axs.set_ylabel('Number of balls')
axs.legend(loc='upper right', fontsize='large')
fig.set_size_inches(9.5, 9.5)
#fig.set_dpi(300)
plt.savefig('Normal_for_'+str(N)+'_time='+str(time)+'_and_mi='+str(int(mi*1
00))+'.png')
plt.show()


fig, axs = plt.subplots(1,1)
axs.plot(range(0,time), B, '--.', c="black", label='Black balls')
axs.plot(range(0,time), W, '--.', c="red", label='White balls',)
axs.set_xlabel('Time [MSC]')
axs.set_ylabel('Number of balls')
axs.legend(loc='upper right', fontsize='large')
plt.xlim([-10,2100])
fig.set_size_inches(9.5, 9.5)
#fig.set_dpi(300)
plt.savefig('Normalzoom_for_'+str(N)+'_time='+str(time)+'_and_mi='+str(int(
mi*100))+'.png')
plt.show()



fig, ax = plt.subplots()
cmp = colors.ListedColormap(['black','white'])
ax.pcolormesh(range(0, N),range(0,time),all_balls, cmap=cmp)
fig.set_size_inches(9.5, 9.5)
ax.set_ylabel('Time [MSC]')
ax.set_xlabel('Balls')
#fig.set_dpi(300)
plt.savefig('Heatmap_for_'+str(N)+'_time='+str(time)+'_and_mi='+str(int(mi*
100))+'.png')
plt.show()
```



```
for w, b in zip(W,B):
    delta.append(((w-b)/N))

fig, ax = plt.subplots()
ax.plot(range(0, time),delta)
fig.set_size_inches(9.5, 9.5)
ax.set_ylabel('Delta')
ax.set_xlabel('Time [MSC]')
plt.ylim([-1.25,1.25])
plt.xlim([-10,2100])
plt.savefig('Delta_for_'+str(N)+'_time=_'+str(time)+'_and_mi=_'+str(int(mi*
100))+'.png')
plt.show()
```