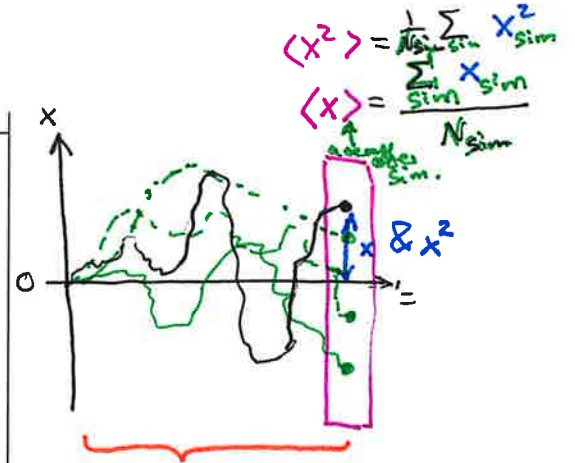
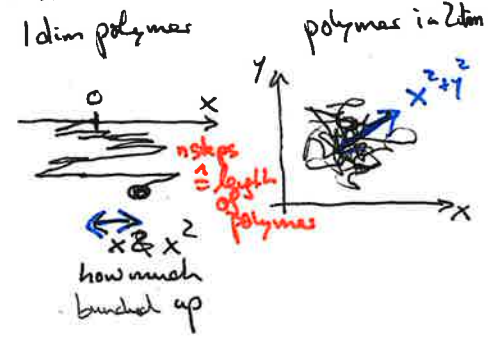
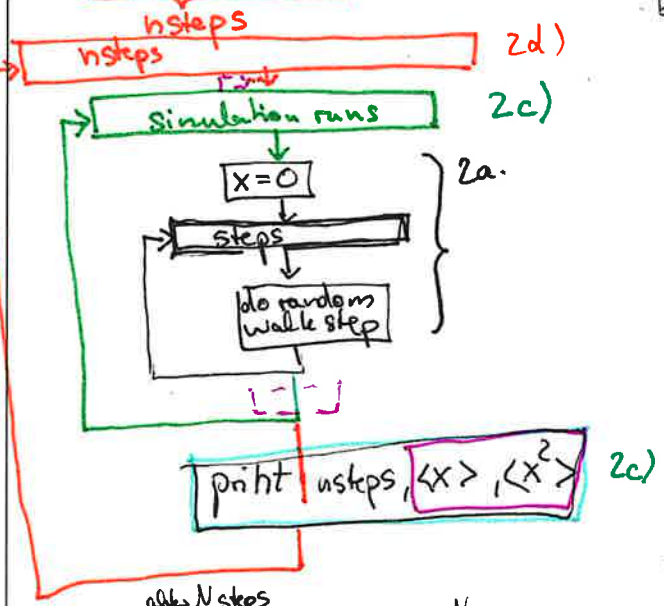
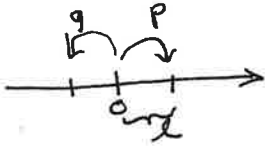
```python
#!/usr/bin/env python
# Random Walks In Class 2d

import scipy as sp
import numpy as np
from scipy import stats

import matplotlib as mpl
import matplotlib.pyplot as plt
#import pylab as pl

p=0.5 #probability to jump to the right
NSIMRUNS=10000

# for reproducible random numbers
sp.random.seed(15)    #argument is any integer

# loop over nsteps
for nsteps in range(100,2100,100):

# loop over indep. simul. runs
  xav = 0.0
  xto2av = 0.0
  for sim in range(NSIMRUNS):

# 1-dim. random walk:
    x=0
    for t in range(nsteps):
      rnd= np.random.random_sample() # rnd number uniformly from [0,1)
      if rnd < p:
        x += 1
      else:
        x -= 1
    xav += x
    xto2av += x**2

  print (nsteps,xav/float(NSIMRUNS),xto2av/float(NSIMRUNS))

# also possible for p=0.5 is following line to replace above if ...
#if np.random.randint(2) == 0:
```



OR







Theory

$$\langle x(N) \rangle = \left\langle \sum_{i=1}^{N} s_i \ell \right\rangle \quad s_i \in \{+1,-1\}$$

after N steps; average over simul. runs

$$= N(p\ell - q\ell) = N\ell(p-q) = 0 \qquad p=q=\tfrac{1}{2}$$

$$\langle x^2(N) \rangle = \left\langle \left(\sum_{i=1}^{N} \ell s_i \right)^2 \right\rangle = \left\langle \sum_{i=1}^{N} \underbrace{(\ell s_i)^2}_{\ell^2(\pm 1)^2 = \ell^2} \right\rangle + \left\langle \sum_{i \neq j} \ell^2 s_i s_j \right\rangle$$

$$= N\ell^2 + N(N-1)\left(p^2\ell^2 - 2pq\ell^2 + q^2\ell^2\right)$$

$$= N\ell^2 + N\ell^2(N-1)(p-q)^2$$

$$\langle x^2(N) \rangle \underset{\text{for } p=q=0.5}{=} N\ell^2$$

for many particle systems (Brownian motion)

1dim $\langle x^2(t) \rangle = \frac{2RT}{\delta N_A} t = 2Dt$ ← diffusion const.

$$\langle r^2(t) \rangle = 4Dt$$

KAlJ & SiO_2 $\langle r_\alpha^2 \rangle = \frac{1}{N_\alpha} \sum_{i=1}^{N_\alpha} |\vec{r}_i(t) - \vec{r}_i(0)|^2$

show msd of SiO_2 & KAlJ

Printed by