Video : https://youtu.be/5Rkt-0MDFUk

普物期末程式專題
布朗運動

製作者:B07901026吳禎軒

前言

1.程式簡潔易懂，vpython 獨力編寫

2.秉持模擬精神，不要導果為因

3.以物理觀念探討其他學術領域

4.從錯誤中學習、檢討

3D動畫模擬

分子位移分布與機率

幾何布朗運動

3D動畫模擬

簡介

1.液體分子的熱擾動繼續，懸浮粒子便會繼鎖顯出隨機運動，即布朗運動

2.原子分子的存在乃氣體運動的基礎，而布朗運動則是證明分子存在的一種現象

目的

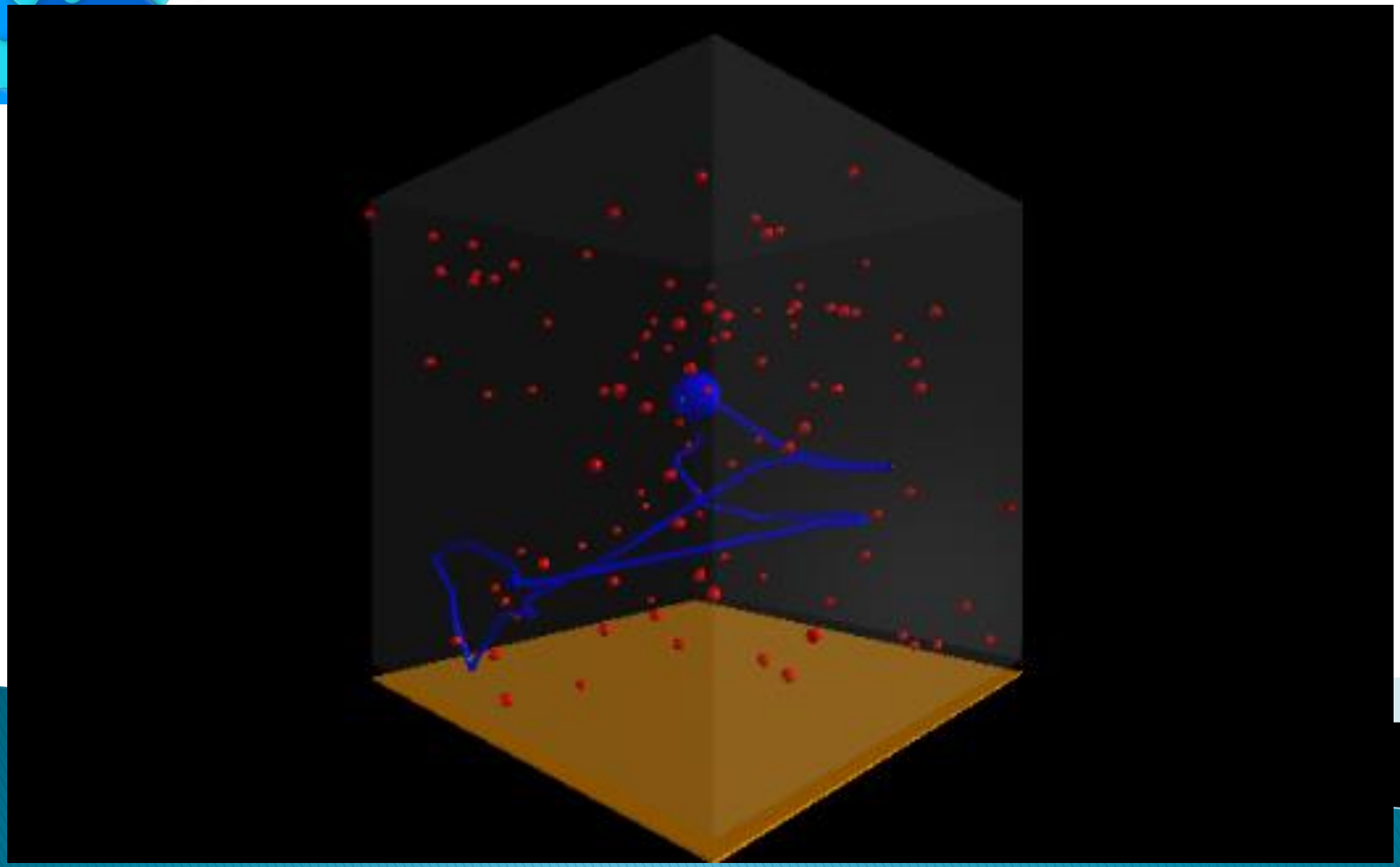
1.3D布朗運動模擬(boundary condition):

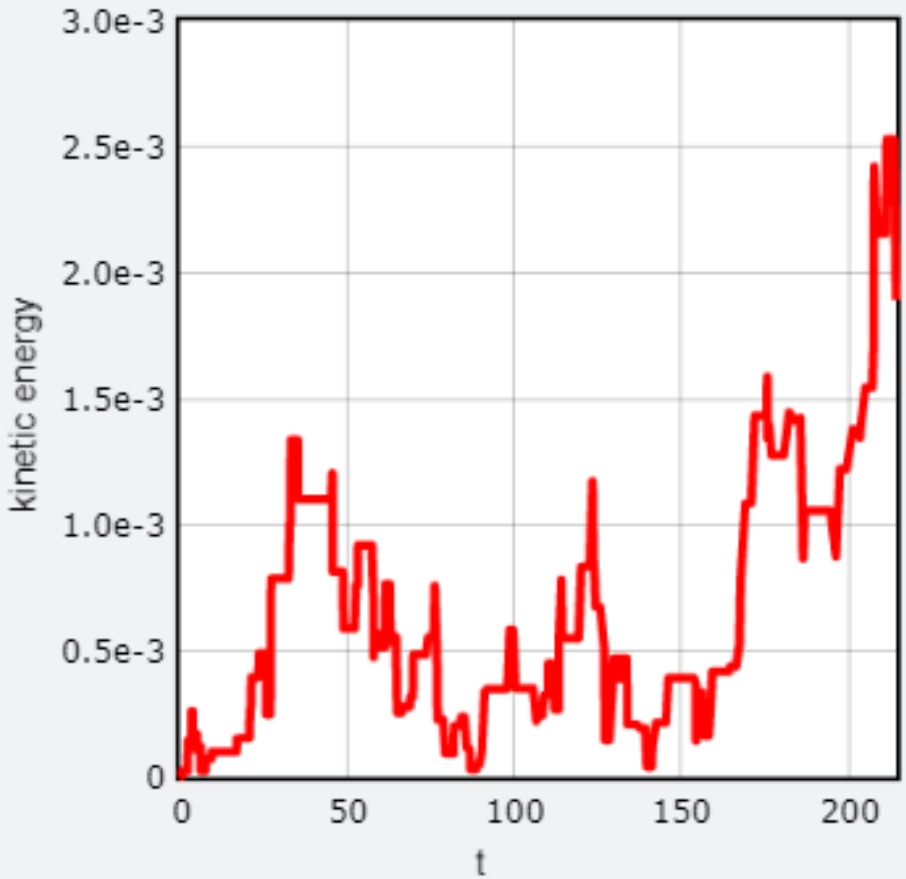
#體積一單位的箱子

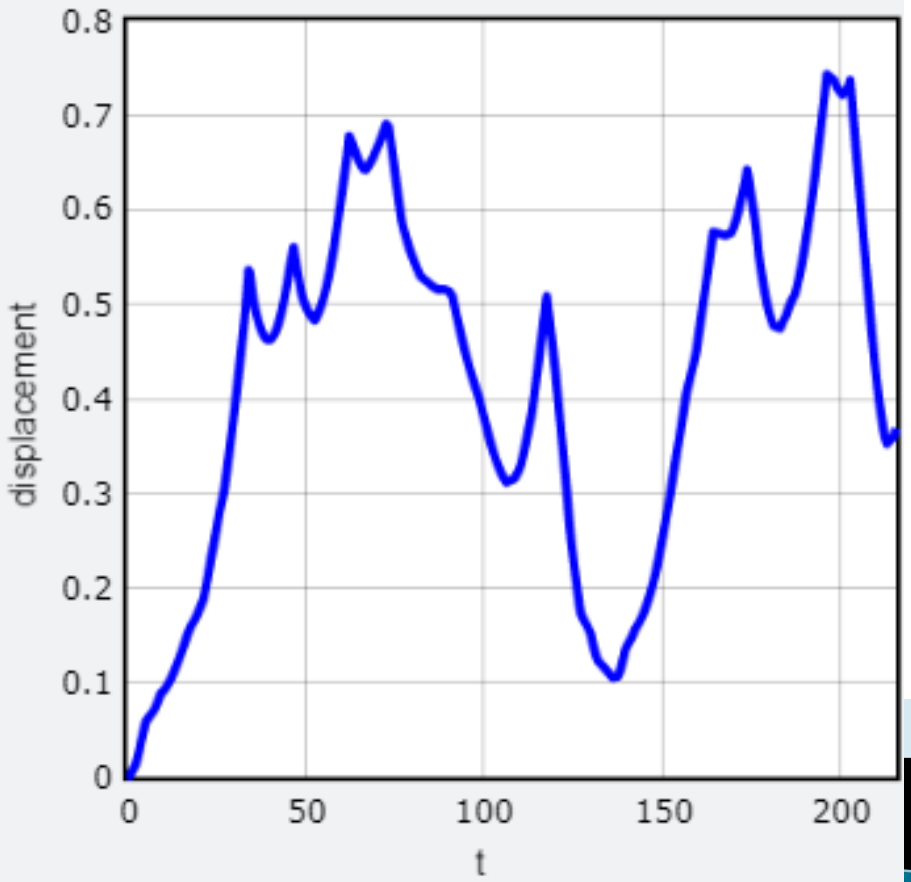
#一個大分子(r=0.05, m=1,原點,v0=0)

#100個小分子(r=0.01, m=0.01)

2.旁邊顯示大分子隨時間變化的動能和位移




3.0e-3
2.5e-3
2.0e-3
1.5e-3
1.0e-3
0.5e-3
0
kinetic energy
0
50
100
150
200
t


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
displacement
0
50
100
150
200
t

程式碼

```
def make_molecules(n_molecules, radius, mass):
    molecules = [Dust(
        pos=vector(0, 0, 0),
        vel=vector(0, 0, 0),
        col=color.blue,
        rad=0.05,
        mass=1
    )]
    for i in range(n_molecules):
        molecules.append(Particle(
            pos=vector(r() - 0.5, r() - 0.5, r() - 0.5),
            vel=vector(r() - 0.5, r() - 0.5, r() - 0.5),
            col=color.red,
            rad=radius,
            mass=mass
            ))
    return molecules

def update_velocity(molecules, dt):
    for m in molecules:
        calc_wall_collision(m)

    calc_part_collision(molecules[0], molecules)


def update_position(molecules, dt):
    for m in molecules:
        m.pos += m.vel * dt
```

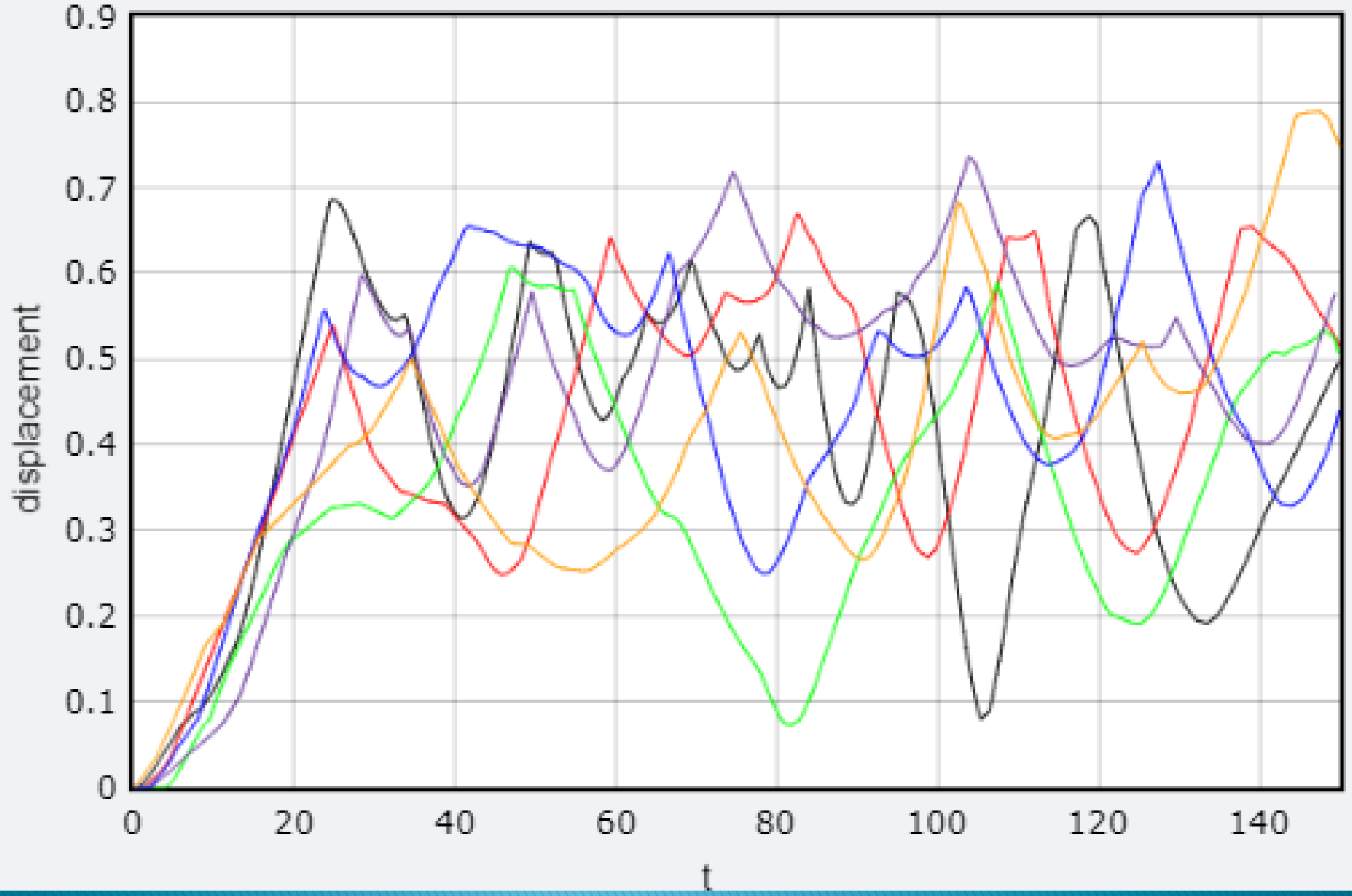
```python
# Stable (no oscillating velocity) because return to inside bounding box
# is guaranteed
def calc_wall_collision(particle):
    box_bounds = 0.5
    if abs(particle.pos.x) >= box_bounds:
        particle.vel.x *= -1
    if abs(particle.pos.y) >= box_bounds:
        particle.vel.y *= -1
    if abs(particle.pos.z) >= box_bounds:
        particle.vel.z *= -1


def calc_part_collision(p, molecules):
    for m in molecules:
        # Avoid collision with itself
        if p is m:
            continue

        # If collision detected, perform elastic momentum transfer
        # (Python simultaneous assignment ftw!)
        if mag(p.pos - m.pos) <= p.radius + m.radius:
            # If masses are equal, velocities are swapped

            if abs(p.mass - m.mass) < 1e-3:
                p.vel, m.vel = m.vel, p.vel
            else:
                p.vel, m.vel = \
                    (p.vel * (p.mass - m.mass) + 2 * m.mass * m.vel) / (p.mass + m.mass), \
                    (m.vel * (m.mass - p.mass) + 2 * p.mass * p.vel) / (p.mass + m.mass)
            # Process only one collision per frame
            break
```

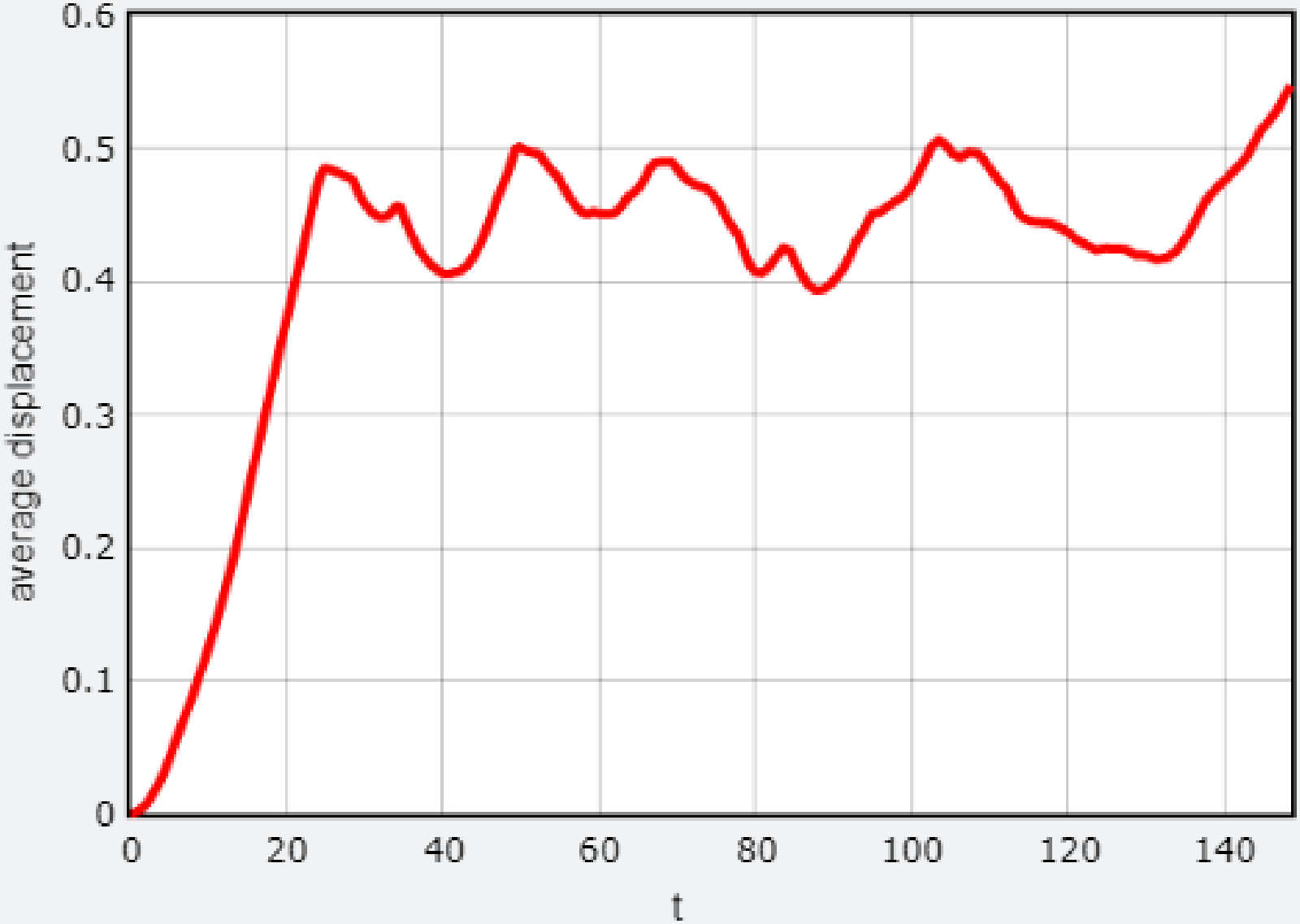
分子位移分布與機率

目的

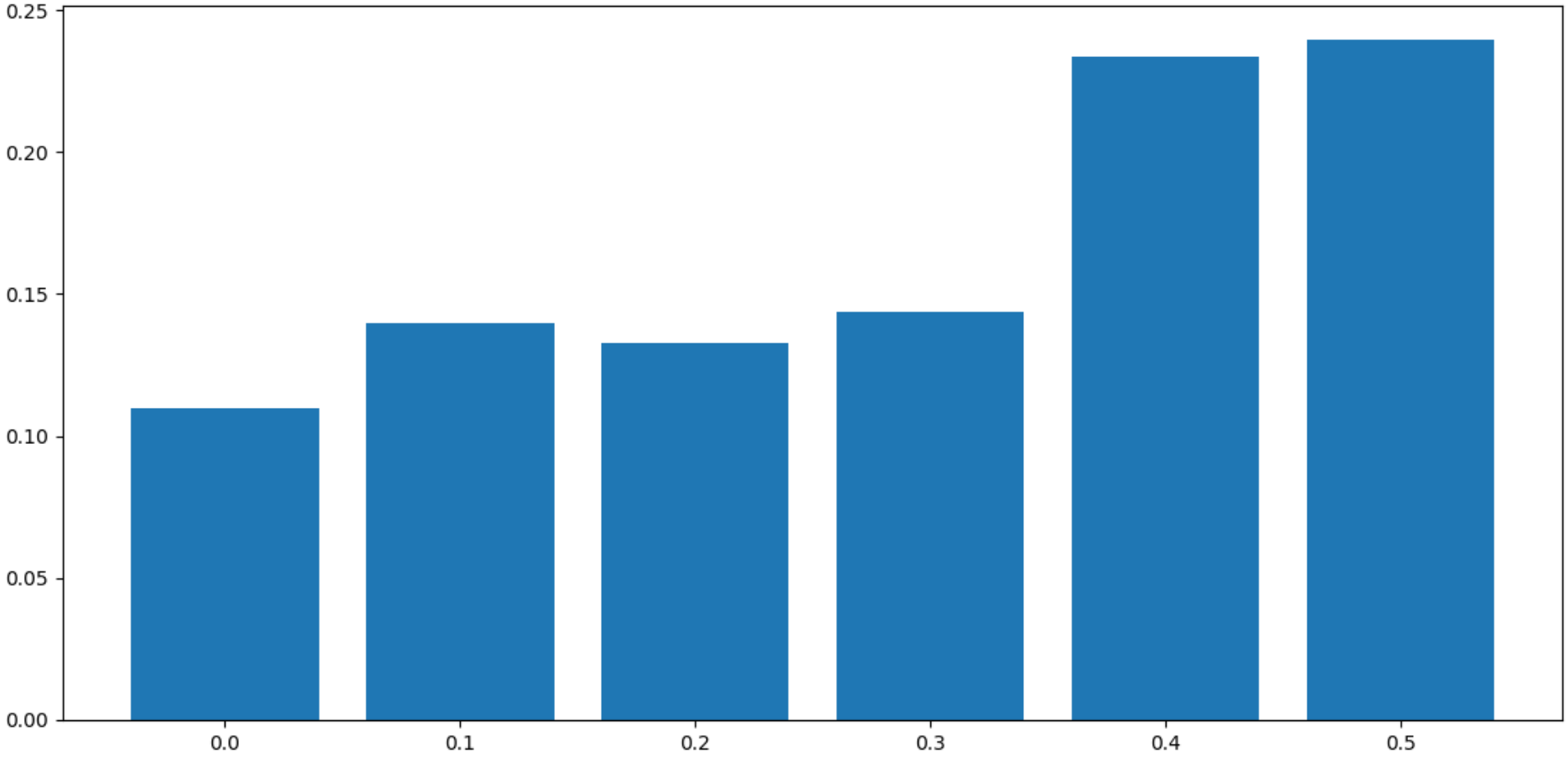
1.進行多個樣本模擬(預設6個)，求出平均位移量，隨時間變化顯示

2.比較不同時間後位移量的機率分布，預測其符合的數學分布

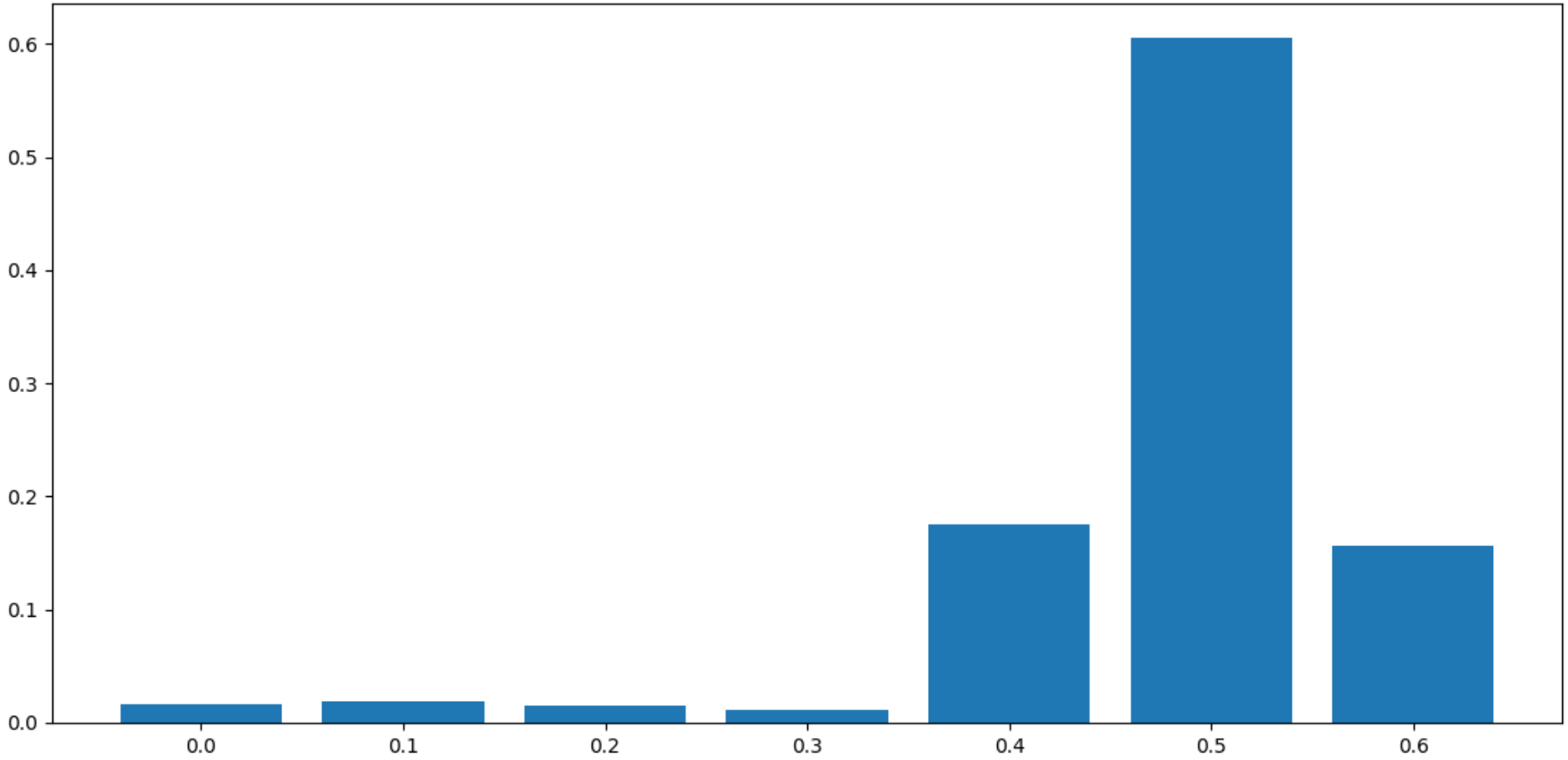

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
displacement
0
20
40
60
80
100
120
140
t


0.6
0.5
0.4
0.3
0.2
0.1
0
average displacement
0
20
40
60
80
100
120
140
t

t=100



t=250

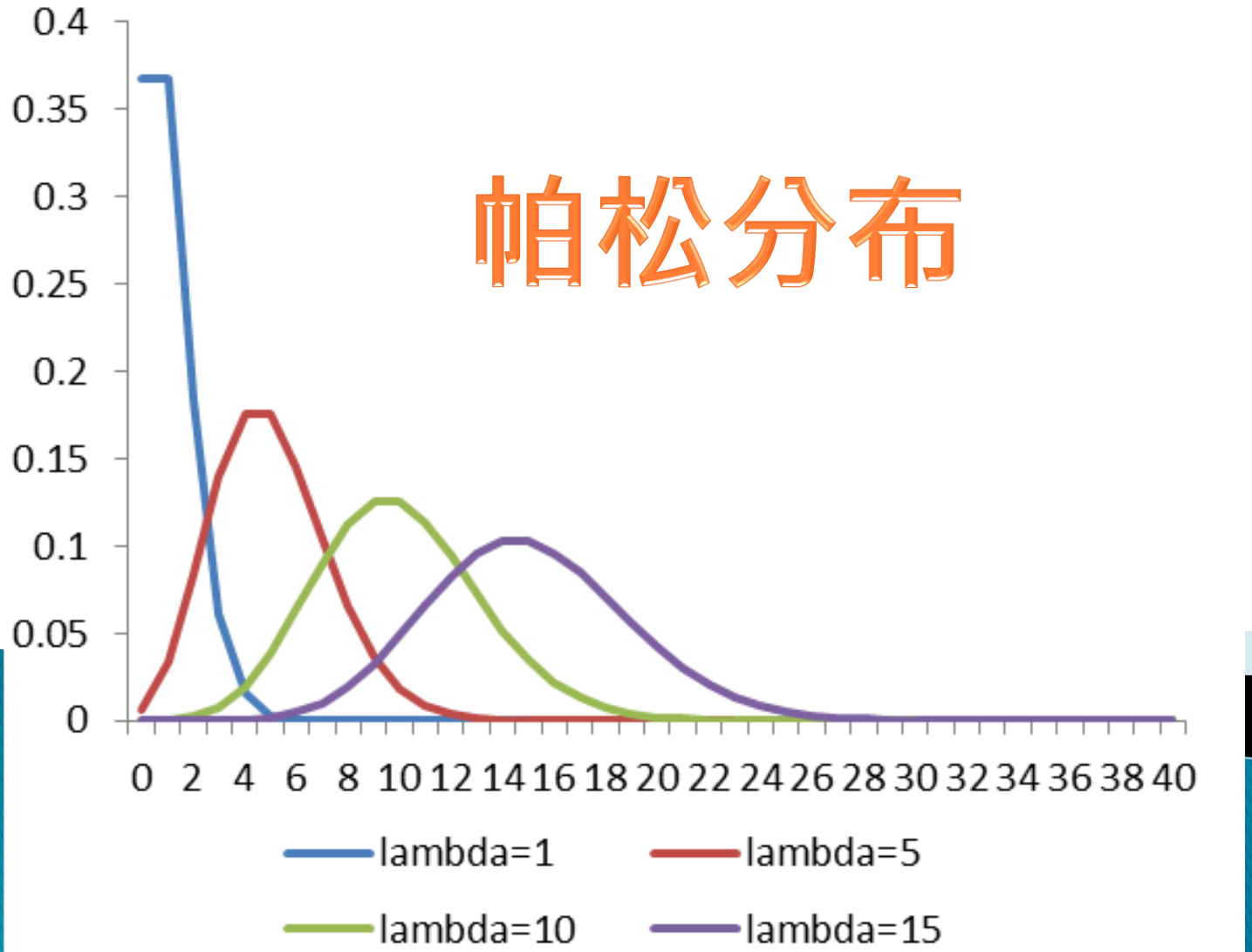


程式碼

```
class Dust:
    def __init__(self,pos,vel,rad,mass):
        self.pos = pos
        self.vel = vel
        self.rad = rad
        self.mass = mass

class Particle:
    def __init__(self,pos,vel,rad,mass):
        self.pos = pos
        self.vel = vel
        self.rad = rad
        self.mass = mass


class many_molecules:   #all molecules in one sample
    def __init__(self,n_molecules, radius, mass):
        self.num = []
        self.num.append(Dust(vector(0, 0, 0),vector(0, 0, 0),0.05,1))
        for i in range(n_molecules):
            self.num.append(Particle(vector(r() - 0.5, r() - 0.5, r() - 0.5),vector(r() -

class collection:   #collection of all samples
    def __init__(self,num):
        self.samples = []
        for i in range(num):
            self.samples.append(many_molecules(n_molecules=100, radius=0.01, mass=0.01).n
```

```
    if t>=250:
        break

prob_dict = {}
for sample in prob_collection:
    if sample not in prob_dict:
        prob_dict[sample]=1
    else:
        prob_dict[sample]+=1

all_samples = sum(prob_dict.values())
for sample in prob_dict:
    prob_dict[sample] = float(prob_dict[sample]/all_samples)

D = {}
for key in prob_dict:
    D[str(key)] = prob_dict[key]

plt.bar(range(len(D)), list(D.values()), align='center')
plt.xticks(range(len(D)), list(D.keys()))

plt.show()
```

預測數學分布


帕松分布
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
lambda=1
lambda=5
lambda=10
lambda=15

適合於描述單位時間內隨機事件發生的次數的機率分佈。

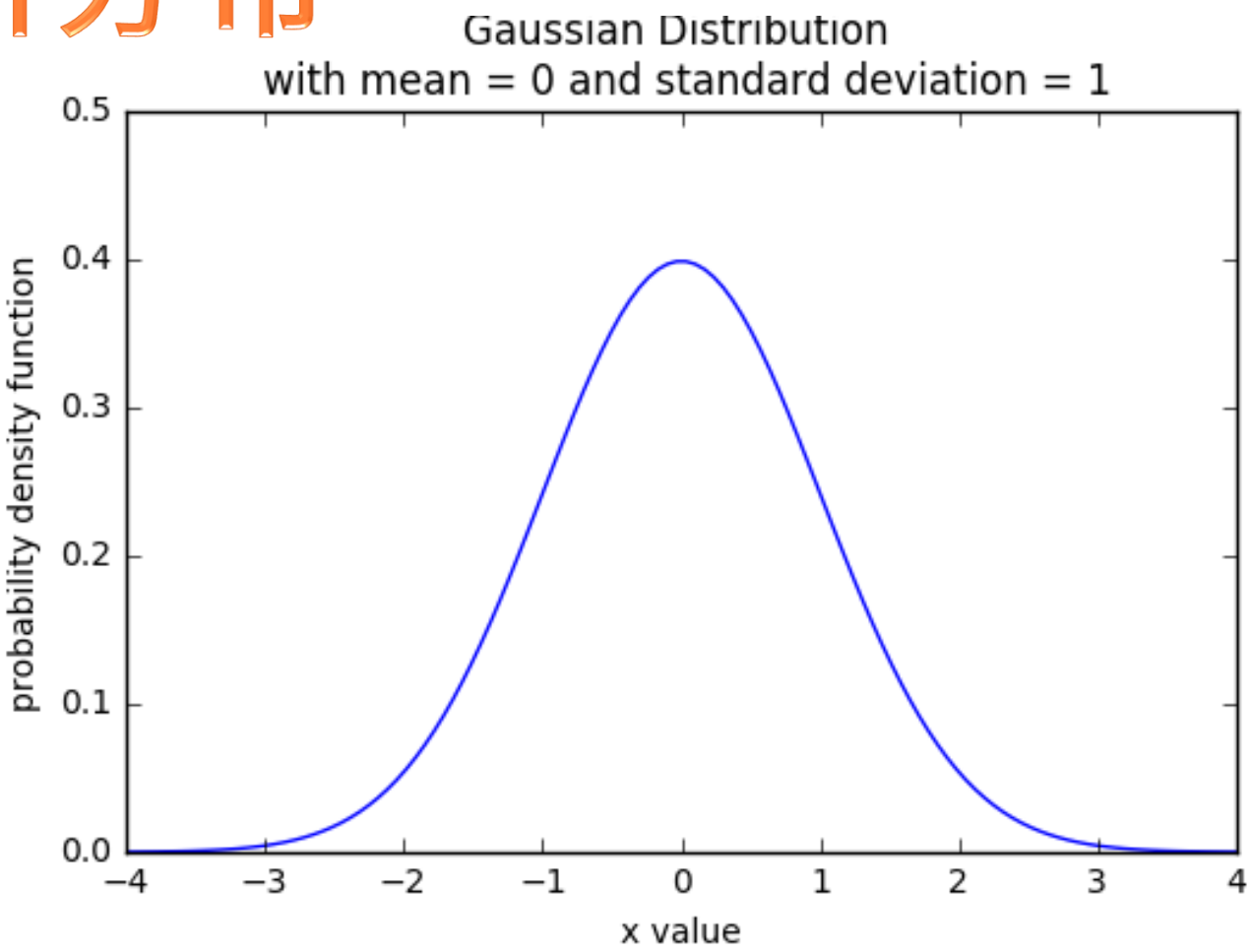
#參數λ是單位時間(或單位面積)內隨機事件的平均發生率

$$P(X = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$


ERROR

真實數學分布

高斯分布


Gaussian Distribution
with mean = 0 and standard deviation = 1
probability density function
0.5
0.4
0.3
0.2
0.1
0.0
−4
−3
−2
−1
0
1
2
3
4
x value

考慮三維情形，$\langle R^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$

且 $\langle x^2 \rangle = \langle y^2 \rangle = \langle z^2 \rangle$

$$\Rightarrow \langle R^2 \rangle = 3\langle x^2 \rangle$$

TA reference

$$\therefore \langle R^2 \rangle = \frac{6KT \cdot t}{\mu}$$

$$P_{n(x)} = \frac{2}{\left(2\pi n l^2\right)^{\frac{1}{2}}} \cdot e^{-\frac{x^2}{2l^2 n}}$$

$$\equiv \frac{2}{(2\pi)^{\frac{1}{2}}\sigma} \cdot e^{-\frac{(x-x)^2}{2\sigma^2}}$$ …… 高斯分佈

TA reference

其中

$$\sigma^2 = nl^2 = \alpha t$$

$$= 0$$

推斷可能錯誤原因

1.規模小，粒子數不足
2.未考慮溫度
3.未考慮液體黏性
4.只有衡量位移量(純量)
5.Boundary condition 大分子會與容器壁相撞

幾何布朗運動

簡介

1.連續時間情況下的隨機過程，其中隨機變量的對數遵循布朗運動，也稱維那過程

2.幾何布朗運動在金融數學中有所應用，用來模仿股票價格

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t$$

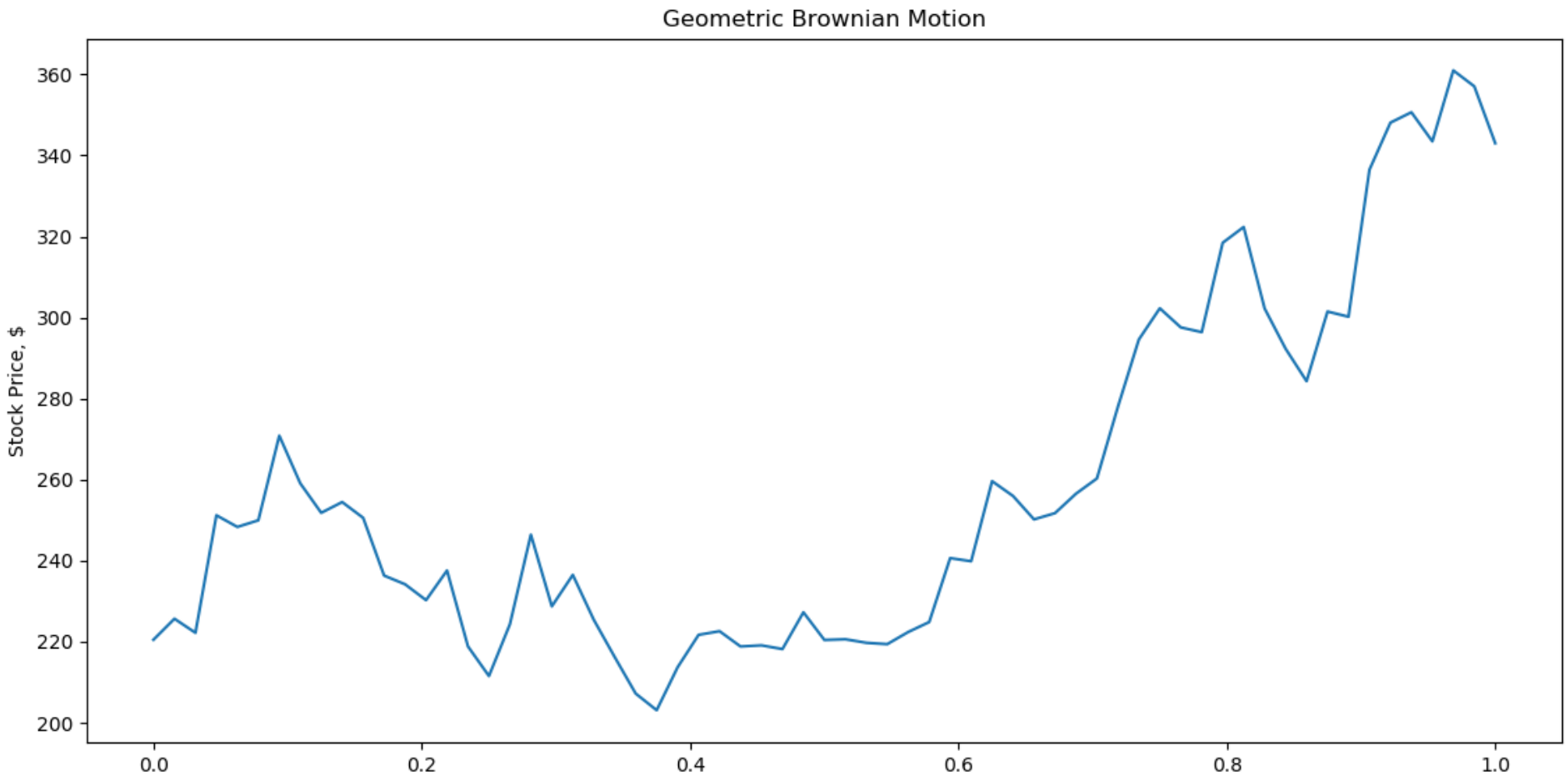
Wt是一個維那過程，也是布朗運動，而μ (‘百分比drift’) 和σ(‘百分volatility’) 則是常數。

目的

1.用SDEs 模擬股價的浮動
　簡單版:許多變數因子未考慮

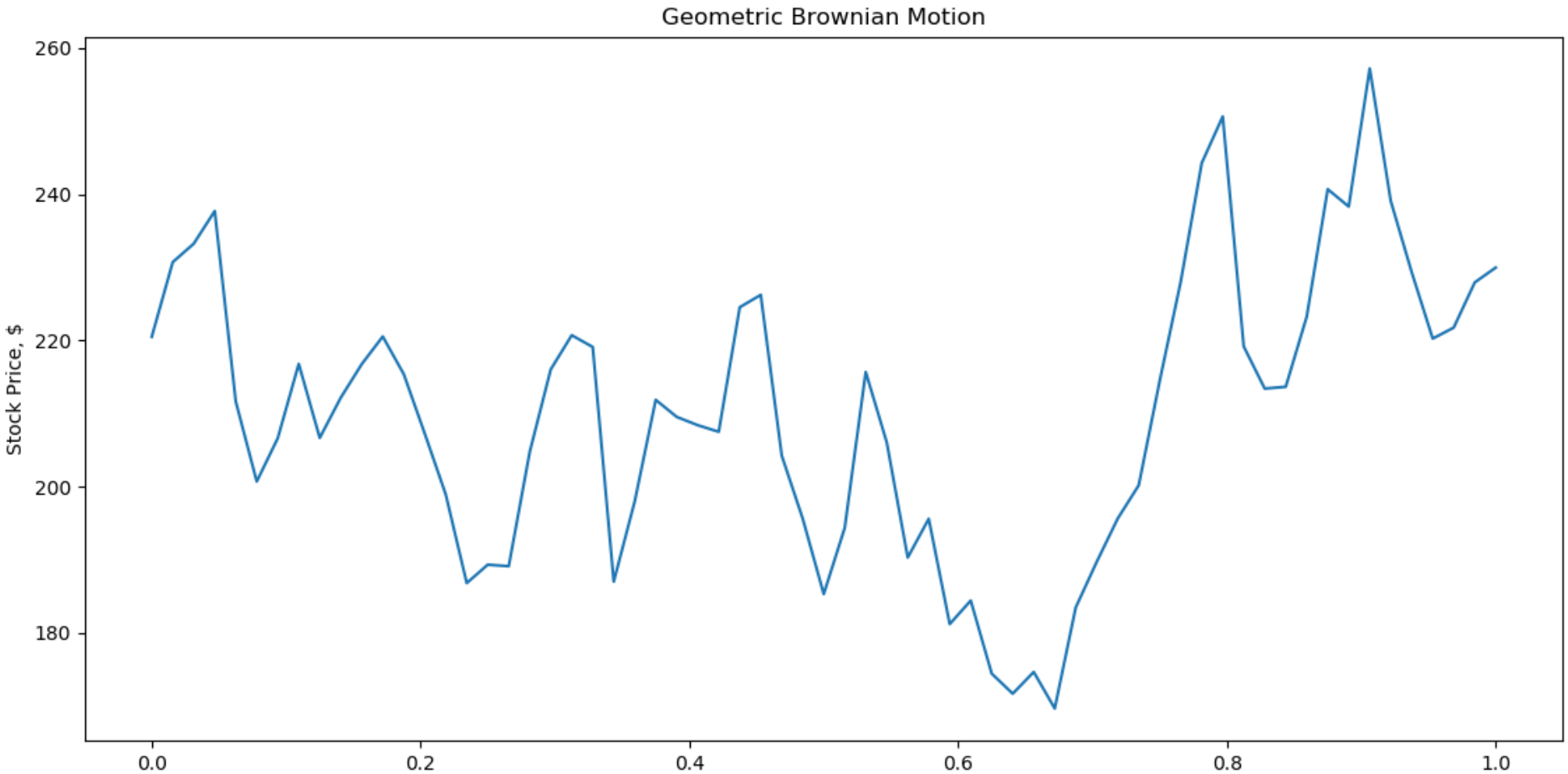
2.探討布朗運動和GBM的關係，把理論物理學的觀念應用於其他領域

seed=5



x=0.62980

seed=20



x=0.59135

程式碼

```
def Brownian(seed, N):

    np.random.seed(seed)
    dt = 1./N                                          # time step
    b = np.random.normal(0., 1., int(N))*np.sqrt(dt)  # brownian increments
    W = np.cumsum(b)                                    # brownian path
    return W, b

def GBM(So, mu, sigma, W, T, N):
    t = np.linspace(0.,1.,N+1)
    S = []
    S.append(So)
    for i in range(1,int(N+1)):
        drift = (mu - 0.5 * sigma**2) * t[i]
        diffusion = sigma * W[i-1]
        S_temp = So*np.exp(drift + diffusion)
        S.append(S_temp)
    return S, t

So = 220.5  #tsmc
mu = 0.15
sigma = 0.4
seed = 20
W = Brownian(seed, N)[0]
T = 1.


soln = GBM(So, mu, sigma, W, T, N)[0]     # Exact solution
t = GBM(So, mu, sigma, W, T, N)[1]        # time increments for  plotting
```

可進步的空間

1.不同變因（溫度、黏滯係數）的布朗運動
2.驗證波茲曼常數
3.考慮阻滯力、推導擴散係數
4.愛因斯坦方程
5.讓程式有更大規模，不影響模擬速率

Reference

1.陳昱蓁、許耘慈兩位課程助教提供的額外參考資料
2.https://matplotlib.org/tutorials/introductory/pyplot.html
3. http://cs231n.github.io/python-numpy-tutorial/

4.https://www.itsfun.com.tw/%E5%B9%BE%E4%BD%95%E5%B8%83%E6%9C%97%E9%81%8B%E5%8B%95/wiki-4755729-3108498
5.https://web.math.sinica.edu.tw/math_media/d164/16408.pdf

報告結束!