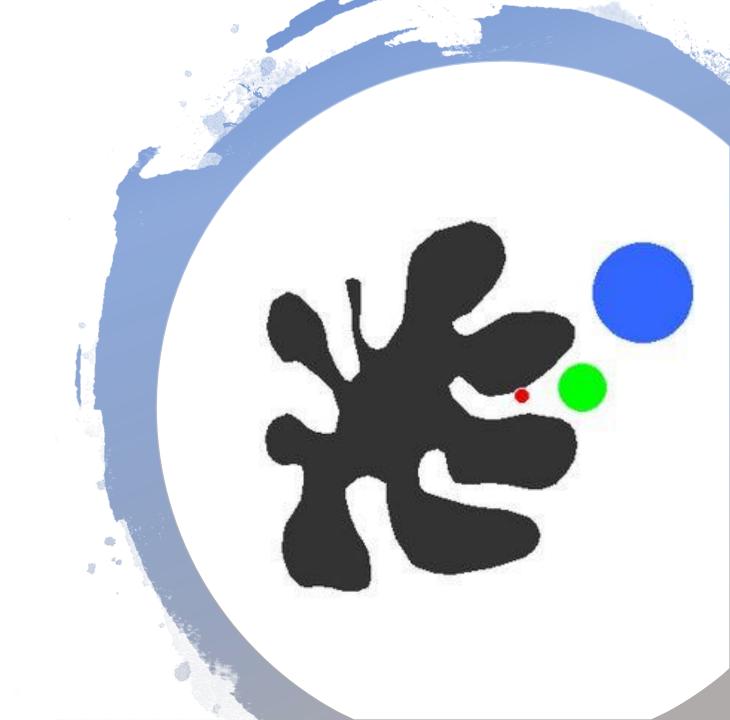
Gel Permeation

104020006余宇恒 104020010李偉志 104021230黃偉誌 104022140許孟楷



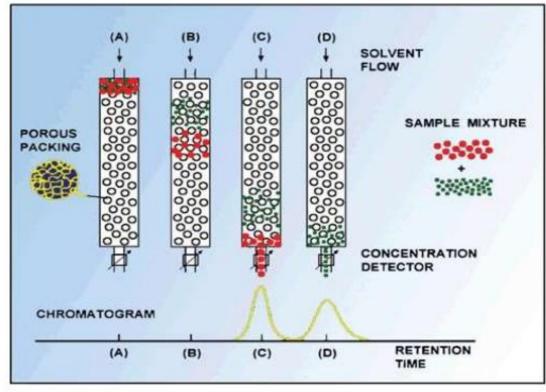


- 1. Gel Permeation 簡介
- 2. 計畫目標
- 3. 計算方法與步驟
- 4. 結果分析
- 5. 參考資料

模擬膠體滲透 層析儀

- 原理:大顆粒子與小顆粒子通過多孔材料所需時間不同
- 應用:分析一個未知混合 物的比例與成分

Separation Mechanism

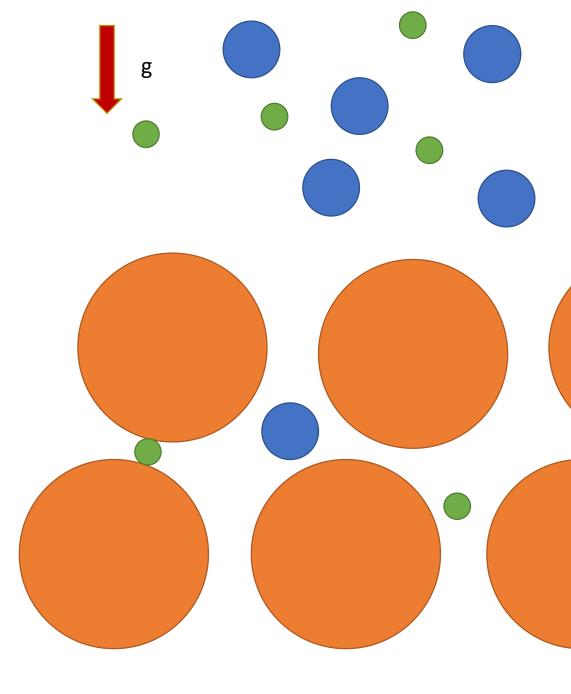


Reference: https://www.slideshare.net/KausarAhmed15/gel-permeation-chromatography-gpc-by-kausar

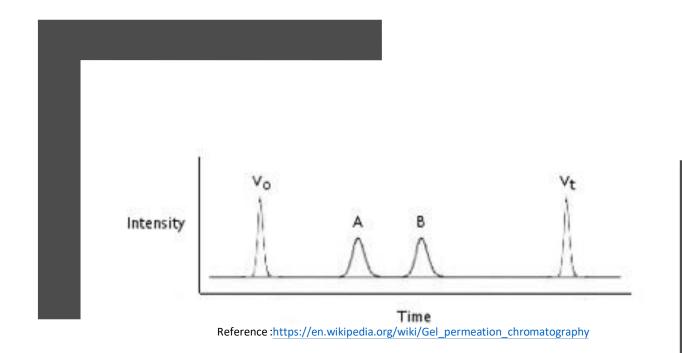
計畫目標

• 橘色球是固定的篩子,藍色與綠色顆粒受重力往下,設為完全彈性碰撞

• 模組:pygame math Matplotlib time Numpy random scipy.optimize sys



計畫目標



- · 畫出落下數量vs時間關 係圖
- · 畫出落下數量vs路徑關 係圖



背景與多孔材料設計:

許孟楷 (104022140)

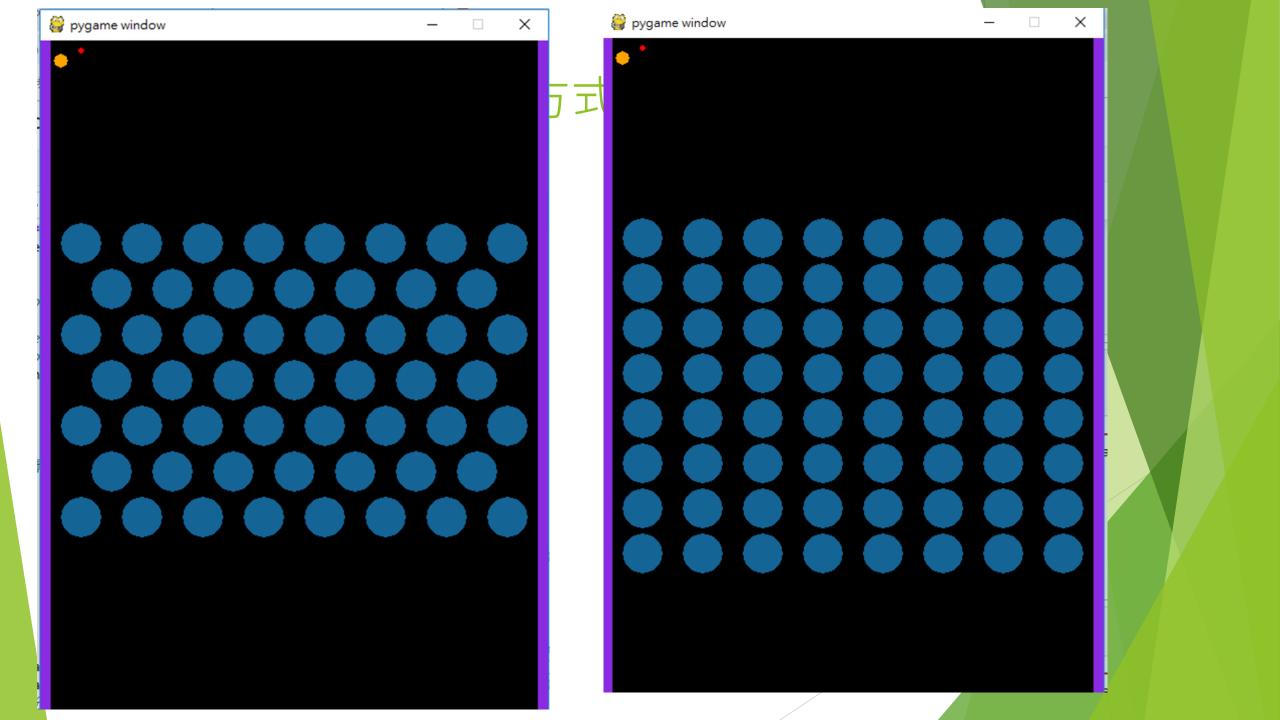
碰撞機制處理:

余宇恒 (104020006)、 李偉志 (104020010) 落下時間vs數量 histogram:

黄偉誌 (104021230)

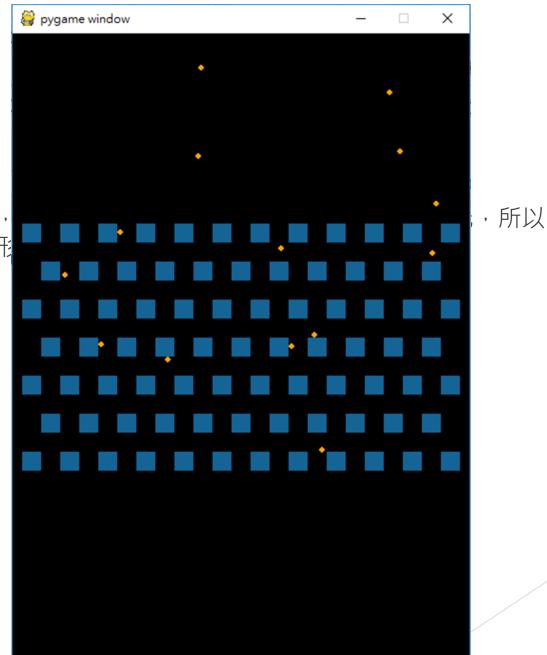
背景與多孔材料設計(許孟楷)

▶ 設置多孔材料的邊界、障礙物大小、排列



形狀改變

因為以圓形為背景的多孔材料, 我們把多孔材料以交錯的正方形



碰撞機制與 球的運動 (余字恒)

- 架構球及障礙物的物件
- 球的移動
- 球與障礙物和邊界的碰撞



物件: 使用class來建立物件,每個物件可以有自己的 屬性(attribute)以及方法(method)

```
class Ball:
    def __init__(self, x, y, vx, vy, color, radius):
        blablabla
    def draw(self, x, y):
        blablabla
    def move(self):
        blablabla
```



物體移動:用一個迴圈不斷更新物件狀態以及畫面,達成動畫的效果

```
while True:
    blablabla
    for b in balls:
        if b.y>bar_height:
            balls.remove(b)
     b in balls:
        b.vy=b.vy+g
        b.draw(b.x, b.y)
        b.move()
    blablabla
```

碰撞機制(李偉志)

粒子間的自由碰撞公式

自由碰撞判定

碰撞問題

- 膠著在一起
- 抖動

solution

```
def collision(ball1, ball2):
    r12 = [ball2.x-ball1.x, ball2.y-ball1.y]
    v1 = [ball1.vx, ball1.vy]
    v2 = [ball2.vx, ball2.vy]
    dist = ((ball1.x - ball2.x)**2 + (ball1.y - ball2.y)**2)**0.5
    if dist <= ball1.radius + ball2.radius:</pre>
        if np.dot(v1, r12)<0 and np.dot(v2, r12)<0:</pre>
            return False
        elif np.dot(v1, r12)*np.dot(v2, r12)>0:
            if np.dot(v1, r12)>0 and np.dot(v2, r12)-np.dot(v1, r12)>0:
                return False
            elif np.dot(v1, r12)<0 and np.dot(v2, r12)-np.dot(v1, r12)<0:
                return False
            else:
                return True
        else:
            return True
    else:
        return False
```

落下時間vs數量histogram (黃偉誌)

畫時間對路徑,個數的關係圖。球走了多少時間,多少距離

具體的實現方法

- 個數

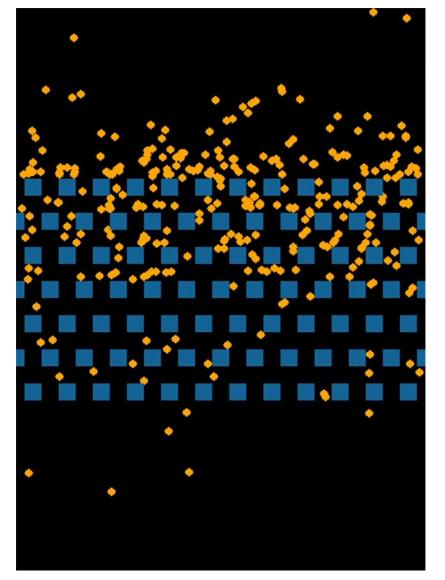
```
global beginning_time # 程式的開始時間
global times # 球通過sensor的時間
beginning_time = time.time() # 初始化,這會給執行這行當下的時間(秒)
times = [] # 初始化
times.append(time.time() - beginning_time) # 減掉開始時間,就得到球落下的總時間
plt.hist(times, bins='auto') # 長方圖
plt.show()#畫出來
```

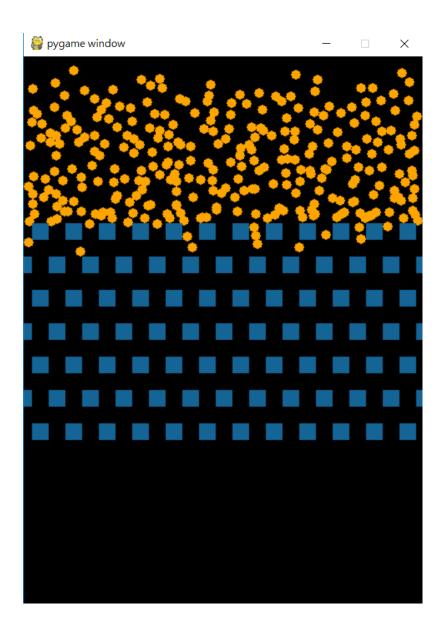
具體的實現方法

- 路徑

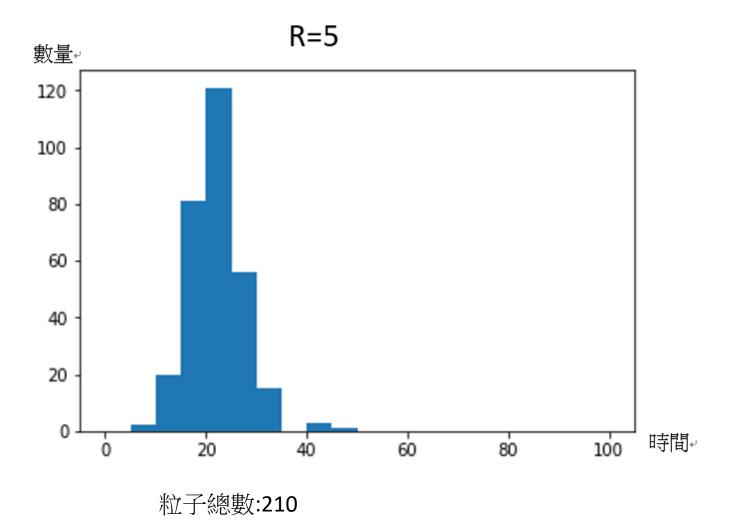
```
設計資料結構到球的class裡面: distance
self.distance = 0 # 初始化,零里程
moving_distance = ( self.vx **2 + self.vy ** 2 ) ** (1/2) # 用速度算距離(單位時間內)
self.distance += moving_distance # 加上去
all_balls_distance = [ball.distance for ball in balls] # 所有球的距離
plt.hist(all_balls_distance, bins='auto') # 長方圖
plt.show()#畫出來
```

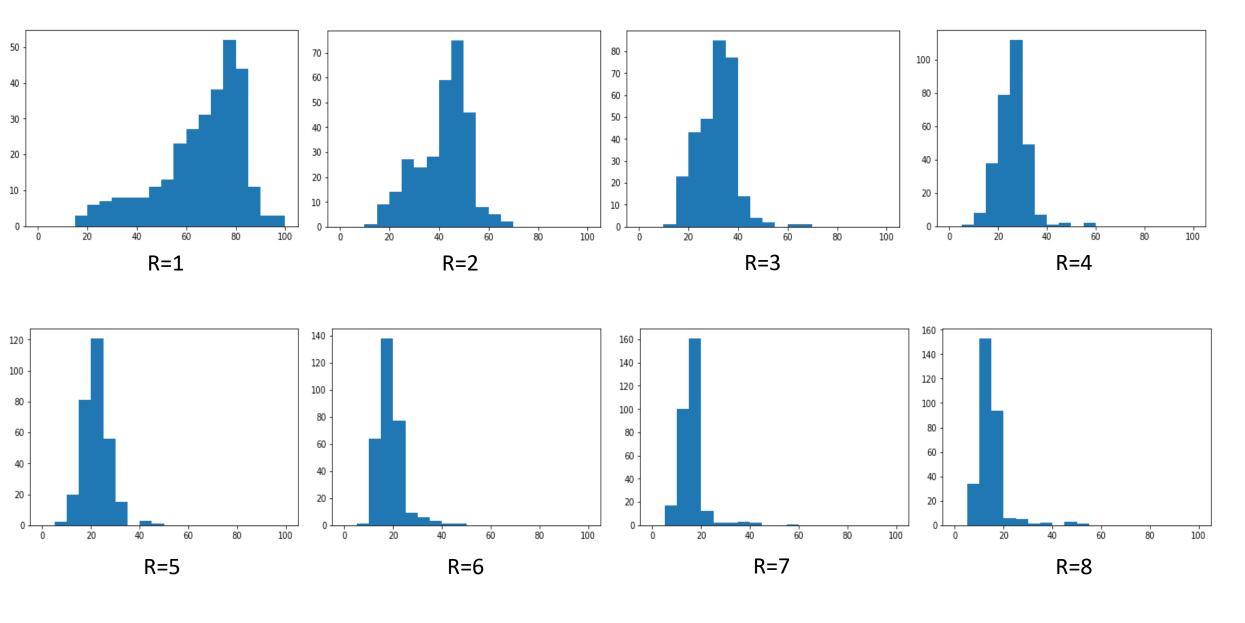
實際效果





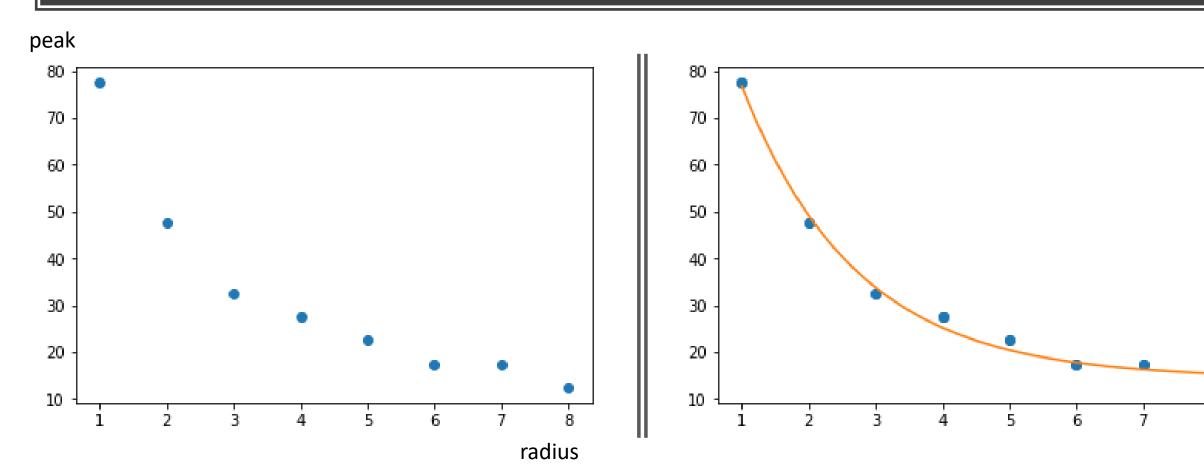
結果分析

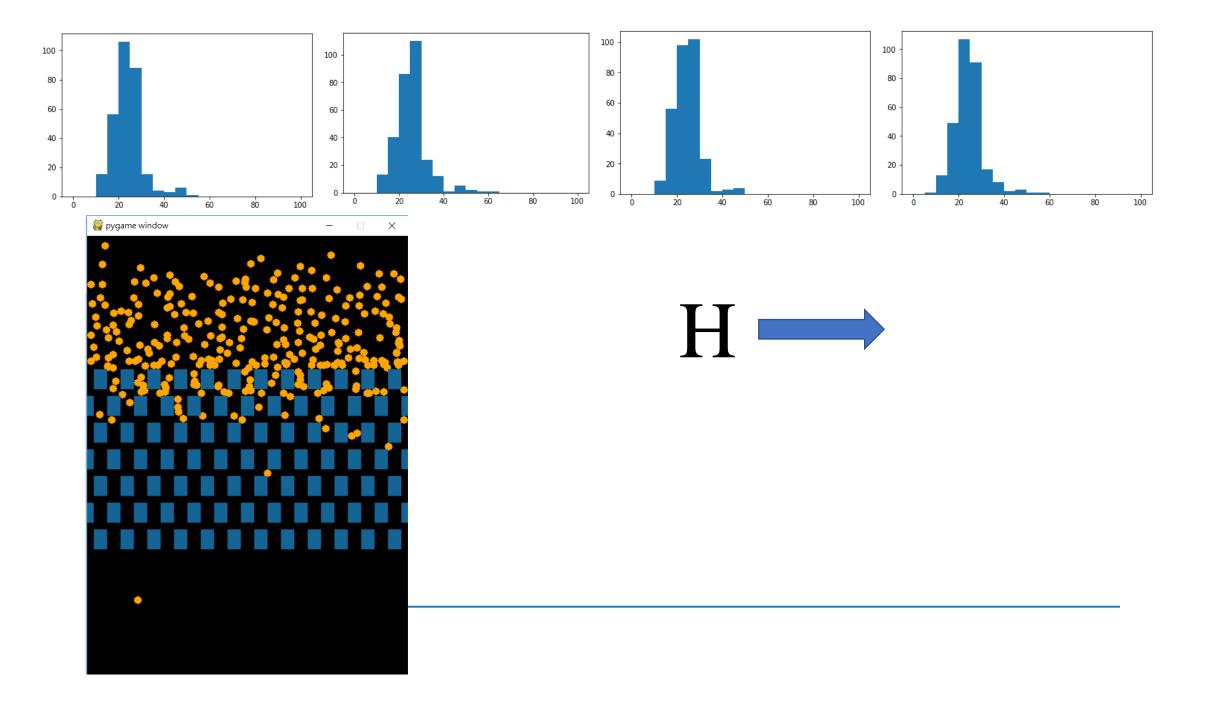






Curve fitting





參考資料

• https://en.wikipedia.org/wiki/Elastic_collision

• Https://en.wikipedia.org/wiki/Gel_permeation_chromatography

• https://www.agilent.com/cs/library/primers/Public/5990-6969EN%20GPC%20SEC%20Chrom%20Guide.pdf

Thanks for listening