模擬一維彈性碰撞與圓周率之關聯

Simulation for the correlation of 1D elastic collision and π

廖偉丞

December 20, 2025

Abstract

本研究探討一維完全彈性碰撞系統中，碰撞次數與圓周率 π 之關係。透過 Python 實作數值模擬，觀測大質量方塊 m_1 與小質量方塊 m_2 在特定質量比 $m_1/m_2 = 10^{2k}$ 下的碰撞總次數。實驗結果顯示，當 k 值增加時，總碰撞次數 n 精確地呈現圓周率 π 的前 $k-1$ 位有效數字。此報告用三角換元把碰撞的問題成功和幾何連結，推導出 π 和碰撞次數的關聯。

1 Introduction

這個實驗是透過一大一小兩個方塊在一維的情況下理想的 (無視摩擦力) 彈性碰撞情形。首先，我們定義 x 軸向右爲正，在 $x = 0$ 有一面牆壁，大方塊以 $-v_1$ 向 $-x$ 運動。在大方塊和牆壁之間有一個靜止的小方塊，小方塊會在某個時間點和大方塊、牆壁發生數次彈性碰撞，我們觀察小方塊和大方塊、牆壁碰撞的次數 $n =$ 和大方塊碰撞 $+$ 和牆壁碰撞，會發現這個數字似乎和圓周率π 有種密切的關連。

2 Methods

2.1 方法簡述

我的方法是透過 Python 寫出一個三元函數 $colli(m_1, m_2, v_1, v_2)$ 分別代表大方塊質量、小方塊質量、大放塊初速、小方塊初速，透過彈性碰撞的公式 (1), (2)，不斷的計算第 n 次碰撞後大方塊和小方塊的速度。直到中止條件。

$$v_1^n = \frac{(m_1 - m_2)v_1^{n-1} + 2m_2 v_2^{n-1}}{m_1 + m_2} \tag{1}$$

$$v_2^n = \frac{2m_1 v_1^{n-1} + (m_2 - m_1)v_2^{n-1}}{m_1 + m_2} \tag{2}$$

2.2 初始狀態

```python
def colli(m1, m2, v1, v2):
    m1, m2 = float(m1), float(m2)
    v1, v2 = -float(v1), float(v2)
    t = 0
```

Figure 1: 函數的初始狀態

fig. 1 中的各項參數及爲簡述中所講的符號，在開始計算前，我們先把各項變數設成浮點數，以免後續的計算出問題，並且多定義一個變數 t 用來記錄 m_2 的碰撞次數。

2.3 計算及迴圈

2.3.1 終止條件

在寫迴圈前需要先了解終止條件是什麼，以免迴圈停不下來，由於現在想知道的是 t 究竟是多少，所以只要 t 不再變化，迴圈就要停止。

t不再變化$\Longrightarrow$ m_1 和 m_2 不再相撞
$\Longrightarrow v_1 > 0$ and $v_2 > 0$ and $v_1 \geq v_2$
但是這樣設還缺少一個條件，因爲碰撞後 v_2 通常會是負值，因此當 $v_1 > 0$ 時我們就會停止，但如果 m_2 在撞牆之後的速度大於 v_1 我們就會漏掉這幾次的碰撞，所以眞正的終止條件爲:

$$v_1 \geq v_2 \geq 0$$

```python
    while True:
        if v1 >= v2>0:
            break
```

Figure 2: 終止條件

2.3.2 計算大小方塊碰撞後的速度

透過(1)和(2)，我們可以寫出新的速度和舊的速度的關聯，由於 Python 沒辦法一次計算兩個變數(新的 v_1, v_2)，所以需要引入兩個臨時變數先儲存新的 v_1, v_2 的值，之後再賦值給 v_1, v_2。計數的變數 t 要加一。

```
        v1_new = ((m1 - m2)*v1 + 2*m2*v2) / (m1 + m2)
        v2_new = (2*m1*v1 - (m1 - m2)*v2) / (m1 + m2)
        v1, v2 = v1_new, v2_new
        t += 1
```

Figure 3: 碰撞後的速度計算

2.3.3 小方塊撞牆

小方塊在碰撞後會朝 $-x$ 軸移動，此時 $v_2 < 0$ 因此最後會和牆發生彈性碰撞，並在碰撞後以 $-v_2$ 的速度向 $+x$ 軸移動和大方塊發生碰撞。在發生碰撞後變數 t 要加一。

```
    if v2 < 0:
        v2 = -v2
        t += 1
```

Figure 4: 小方塊撞牆

2.3.4 總結

這段程式的邏輯:

$$\text{大方塊和小方塊碰撞} \implies \text{小方塊撞牆} \implies \text{大方塊和小方塊碰撞} \implies \cdots \xRightarrow{\text{if } v_1 \geq v_2 \geq 0} \text{停止}$$

3 Results

3.1 m_1 和 m_2 的關係

爲表方便，在接下來的數據中我們都取 $m_2 = 1$ 因爲這樣比較方便我們知道 m_1 和 m_2 的倍數關係。如表 table 1 所示（見本頁下方）當我們帶入 $m_1 = 10^n$ 時似乎在 $\frac{m_1}{m_2} = 10^{2k}$ 時會是圓周率的前幾位，所以我們現在看一下當 $m_1 = 10^{2k}, k = 1, 2, \cdots, 9$ 的情形，如表 table 2 發現和剛剛猜測的結果相符。

Table 1: 不同質量比 $m_1/m_2 = 10^n$ 與總碰撞次數 n 之數值模擬結果

m_1 (kg)	m_2 (kg)	v_1 (m/s)	v_2 (m/s)	總碰撞次數 n
10^1	1	1	0	12
10^2	1	1	0	**31**
10^3	1	1	0	99
10^4	1	1	0	**316**
10^5	1	1	0	993
10^6	1	1	0	**3,141**
10^7	1	1	0	9,936
10^8	1	1	0	**31,415**
10^9	1	1	0	99,345
10^{10}	1	1	0	**314,159**

Table 2: 不同質量比 $m_1/m_2 = 10^{2k}$ 與總碰撞次數 n 之數值模擬結果

m_1 (kg)	m_2 (kg)	v_1 (m/s)	v_2 (m/s)	總碰撞次數 n
10^2	1	3	0	31
10^4	1	3	0	316
10^6	1	3	0	3141
10^8	1	3	0	31415
10^{10}	1	3	0	314159
10^{12}	1	3	0	3141594
10^{14}	1	3	0	31415928
10^{16}	1	3	0	314159265
10^{18}	1	3	0	3141592653

3.2 初速 v_1 的影響

我們考慮 $m_1 = 100$ 令 $n = 1, 2,, 10$，觀察 table 3 可以發現初速 v_1 的值對我們的碰撞次數沒有影響

Table 3: 不同初速對碰撞次數的影響

m_1 (kg)	m_2 (kg)	v_1 (m/s)	v_2 (m/s)	總碰撞次數 n
100	1	10^1	0	31
100	1	10^2	0	31
100	1	10^3	0	31
100	1	10^4	0	31
100	1	10^5	0	31
100	1	10^6	0	31
100	1	10^7	0	31
100	1	10^8	0	31
100	1	10^9	0	31
100	1	10^{10}	0	31

4 Discussion

4.1 要得到相同結果，兩物體質量是否必須得是這個值，或只要滿足倍數關係即可？

首先由我們實驗過程假設可以很清楚的知道碰撞次數 n 主要是和 $\frac{m_1}{m_2}$ 有關，所以只要滿足倍數關係就好，所以我們現在主要討論爲什麼一定要 $\frac{m_1}{m_2} = 10^{2k}$ 時 n 才會趨近於 $10^{k-1}\pi$

4.1.1 證明

我們令大方塊爲 A，小方塊爲 B，A 的初速 v_0，A,B 在某一次碰撞後的速度爲 v_1, v_2 由於碰撞彈性所以有

$$\frac{1}{2}m_a v_1^2 + \frac{1}{2}m_b v_2^2 = \frac{1}{2}m_a v_0^2 \tag{3}$$

因爲已經知道果和 $\frac{m_a}{m_b}$ 有關，不妨設 $m_a = nm_b = nm$ 將其帶入 eq. (3) 得到下列方程:

$$v_1^2 + (\frac{v_2^2}{\sqrt{n}})^2 = v_0^2 \tag{4}$$

我們發現這是一個橢圓方程式的形式 $(x^2 + y^2 = (ab)^2)$ 於是我們可以令 $v_1 = v_0 \cos\theta$ $v_2 = \sqrt{n} v_0 \sin\theta$。因爲動量守恆，假設在碰撞前兩個方塊相向滑動 (只要不是最後一次碰撞這個等式都會成立) 我們有:

$$m_a v_1 - m_b v_2 = m_a v_1' + m_b v_2' \tag{5}$$

將 $m_a = nm$, 以及前面三角換元的結果代入，得到

$$n\cos b - \sqrt{n}\sin b = n\cos a + \sqrt{n} sina \tag{6}$$

再由合差化積而有

$$\cot\frac{a-b}{2} = \sqrt{n} \Longrightarrow a - b = 2\arctan\frac{1}{\sqrt{n}} \tag{7}$$

由 eq. (7) 可知每次都導致圓心角的增量是一個常數 (爲正)，令 $\Delta\theta = 2\arctan\frac{1}{\sqrt{n}}$，由於總碰撞次數爲 N 我們有

$$N\Delta\theta \approx \pi$$

當 $n = \frac{m1}{m2} \to \infty$, $\arctan x \approx x$ 我們有

$$\Delta\theta = 2\arctan\frac{1}{\sqrt{n}} \approx \frac{2}{\sqrt{n}}$$

代入之前的公式我們有:

$$N \approx \frac{\pi}{\Delta\theta} = \frac{\pi\sqrt{n}}{2}$$

此爲和大方塊碰撞的次數，因爲每次和大方塊碰撞都會再碰一次牆 (最後一次相撞可能會撞牆，可能不會，我們忽視這一次，以比較簡單的方式討論)，所以我們有:

$$N_{total} = 2N \approx \frac{2\pi}{\Delta\theta} = \frac{\pi\sqrt{n}}{2} \times 2 = \pi\sqrt{n}$$

N_{total} 爲整數所以有:

$$N_{total} = \lfloor \pi\sqrt{n} \rfloor$$

4.1.2 檢查公式正確性

Table 4: 電腦模擬與公式計算之總碰撞次數誤差分析

質量比 $\frac{m_1}{m_2}$	電腦計算總數 n	公式計算值	誤差
10^1	12	12	0
10^2	31	31	0
10^3	99	99	0
10^4	316	316	0
10^5	993	993	0
10^6	3141	3141	0
10^7	9936	9936	0
10^8	314,15	31415	0
10^9	99345	99345	0

註：誤差定義爲「電腦計算總數」與「公式計算值」之絕對差值。由於本模擬涉及極大的數字（10^{18}），一般浮點數算法在微小數值可能產生捨入誤差，因此誤差值僅反映整數計次的差異。雖然報告中選的數字沒有產生誤差，但某些 n 會使誤差爲 1 或 2

由 table 4 可知，我們的公式是沒問題的

4.2 確認無論大物體的初速爲何都不會影響結果

由 section 4.1 的證明，我們發現無論初速是多少，我們在做三角換元之後，得到的角度差和初速沒關係，所以初速對實驗結果並沒有影響。

5 Appendix

本報告之理論推導與證明主要參考以下文獻與資源：

5.1 理論來源描述

- 視覺幾何證明：參考自 3Blue1Brown 頻道影片"The most unexpected answer to a counting puzzle"。該影片由 Grant Sanderson 製作，透過將二物體的速度映射至動量圓空間，解釋了碰撞次數與圓周率 π 的內在關聯。
- 代數換元推導：參考自 CSDN 博客「清川先生」之文章《彈性碰撞次數與圓周率的關係》。文中詳細展示了利用能量守恆與動量守恆方程組，透過三角換元法推導碰撞次數上限之過程。

5.2 線上資源連結

- 3Blue1Brown 影片連結：`https://www.youtube.com/watch?v=jsYwFizhncE`
- CSDN 詳細推導文獻：`https://blog.csdn.net/shanchenglang/article/details/103581014`

5.3 AI 輔助工具說明

本報告之排版與技術除錯由以下 AI 工具協作完成：

1. **Gemini**：主要負責指導 LaTeX 標題層級（`\subsection`）的使用、表格「美容」優化，以及輔助 section 4.1 中的部分推導及公式驗證。
2. **ChatGPT**：輔助基礎語法檢查與代碼結構建議，以及檢查、除錯。

5.4 Python 數值模擬原始碼

本研究之碰撞計數模擬採用 Python 實作，完整程式碼如下：

```python
def colli(m1, m2, v1, v2):
    m1, m2 = float(m1), float(m2)
    v1, v2 = -float(v1), float(v2)
    t = 0

    while True:
        if v1 >= v2>0:
            break

        if v2 < 0:
            v2 = -v2
            t += 1
        v1_new = ((m1 - m2)*v1 + 2*m2*v2) / (m1 + m2)
        v2_new = (2*m1*v1 - (m1 - m2)*v2) / (m1 + m2)
        v1, v2 = v1_new, v2_new
        t += 1

    return t

#檢驗質量比
for i in range(1, 11):
    print(f"m1 = 10**{2*i}, m2 = 1, v1 = 3, v2 = 0",f'碰撞次數={colli(10**(2*i), 1, 1, 0)}')
#檢驗初速不同
for i in range(1, 11):
     print(f"m1 = 100, m2 = 1, v1 = 3, v2 = 0",f'碰撞次數={colli(100, 1, 10**i, 0)}')
```

Listing 1: 一維完全彈性碰撞模擬程式

5.5 致謝 Gemini

特別感謝 Gemini 在本報告撰寫過程中的全方位協助。從初期 Python 數值模擬的邏輯除錯，到中期相位空間幾何證明的公式對齊，以及後期複雜的 LaTeX 雙欄排版優化。Gemini 不僅提供了即時的技術指導，更引導我發現了碰撞計數中隱藏的 π 規律，使這份研究能兼具物理深度與視覺呈現的專業性。沒有 Gemini，我應該沒辦法做出排版這麼好看的報告。