

彰化縣 114 學年度國民中小學學生獨立研究作品徵選

作品說明書（封面）

作品編號：（由承辦單位編列）

組別：

☒ 國小組

☐ 國中組

☒ 數學類

☐ 自然、科技類

☐ 人文社會類

作品名稱：撒豆成兵算圓周：蒙地卡羅法的大挑戰

◎封面切勿出現校名、作者、校長及指導者姓名，違者不予收件。

目錄

第一階段 研究訓練階段.....	II
第二階段 獨立研究階段.....	1
壹、研究動機與研究目的.....	1
貳、擬定正式計畫、研究問題及工作進度表.....	2
參、相關文獻摘要及實驗工具.....	4
肆、實驗設計與流程.....	7
伍、資料分析.....	9
陸、研究結果與討論.....	20
柒、評鑑與檢討.....	25
捌、參考資料.....	29

第一階段 研究訓練階段

一、近二年學校獨立研究課程之規劃

本校近兩年針對中高年級資優生，以及對獨立研究有興趣的普通班學生，規劃了一系列「獨立研究」課程與指導模式，期望透過多元主題與研究方法訓練，培養學生自主探究與團體合作的能力。在課程設計上，除中年級學生透過實驗操作初探研究樂趣、高年級學生組隊進行主題深耕與競賽準備外，亦提供團體課程與個別指導並行的方式，協助學生在研究歷程中獲得專業的引導與支持。

課程實施特色與流程：

1. 對象與時間安排：

- 中年級(三、四年級)：

提供自然、人文、數學等主題的研究基礎課程。透過歷屆優秀作品的案例分析與實務操作，讓三年級初學者感受研究的樂趣、掌握基本研究方法；四年級則逐步熟悉研究流程，為未來參與獨立研究競賽奠基。

- 高年級(五、六年級)：

結合興趣專長，採混齡分組進行獨立研究主題探究。此階段學生將參與團體課程以培養合作能力，同時進行個別研究指導，以完成獨立研究作品並參加彰化縣獨立研究競賽。

2. 教學與指導模式：

- 團體課程(Group Work)：

利用「外加式課程」進行。課程旨在透過小組活動、實境操

作與討論交流，培養學生的研究方法、合作能力與問題解決能力。此方式可激勵組間良性競爭，提高學習動機。

- 個別指導(Individual Mentoring)：

根據每組學生的研究進度與個人特質，提供一對一或小組預約指導時間。指導教師將就研究主題、方法與報告撰寫等方面，給予具體建議與回饋，確保每位學生都能在研究歷程中獲得適切支援。

3. 成果展現與競賽：

- 於校內舉辦獨立研究發表會，提供學生展現研究初探成果的舞台，藉此肯定其投入與成長。
- 配合彰化縣獨立研究競賽時程，規劃有興趣參加比賽的學生組隊參賽，讓學生透過正式競賽平台磨練研究實力、深化學術素養。

課程規劃表

年級	課程定位與內容	實施方式	成果展現
三年級	研究初階訓練：自然、人文、數學主題探索	外加式課程 操作實務	校內發表會
四年級	進階研究方法學習： 強化研究流程與技能	團體課程與 個別指導	校內發表會
五年級	主題深化與組隊： 依興趣專長分組研究	團體課程(合作 訓練)個別指導	縣級獨立研究競賽
六年級	成果展現與競賽準備： 參賽及專業指導	團體課程(分享 經驗)個別指導	縣級獨立研究競賽

透過上述課程規劃、實施流程與多層次的指導模式，本校期望建立一個兼具創新、探索、協同與競賽平台的學習環境，使學生在獨立研究過程中展現潛能、拓展視野、奠定研究素養的堅實基礎。

二、學校如何提供該生獨立研究訓練

本校秉持十二年國教「自發」、「互動」、「共好」的教育精神，鼓勵學生從生活經驗中發現問題，透過科學研究方法來探索並解決實際疑問。為協助學生從初階研究素養的培養到實際競賽舞台的展現，我們以多層次、多元化的方式設計並推行獨立研究訓練計畫，使學生能在合適的環境中逐步成長與發展。

訓練實施特色：

1. 課程內容設計與資源整合：

自中年級(約三年級起)即開始為學生建立研究素養，透過整合資優班教師、跨領域教師及家長資源，提供多元化的閱讀材料和充實課程，引導學生初步接觸並理解研究方法。以此基礎上，每學年會定期舉辦校內獨立研究發表會，為學生提供展示研究成果的公開平台，並借此激發其學習動機與創新思維。進入高年級階段(四、五、六年級)後，學校更配合彰化縣獨立研究競賽的時程，鼓勵學生依興趣組隊參賽，讓他們在更廣闊的舞台中實踐所學，將研究主題與社會脈動相連結，使獨立研究的進行更貼近生活實務，進而培養出能夠融會貫通、解決問題的未來人才。

2. 個別化教學與多元引導策略：

在獨立研究教學方面，深刻考量學生的個別差異，針對不同學習特質的學生，設計出不同程度的研究任務。指導過程中，若遇到研究能力較佳的學生，教師傾向採用較低結構的引導方式，給予其更多彈性與自主空間，鼓勵發揮創意與自我探索；而對於需要更多協助的學生，則採取高度結構性的引導策略，提供明確的研究步驟與參考方向，使學生在明確的學習軌道上降低挫折感，同時提升其自信與學習動機。透過此種差異化指導模式，學生得以在適合自己的研究挑戰中成長，並逐步培養獨立探究的能力。

3. 學習環境與資源支持：

教學環境注重提供多元而完善的學習環境，包括軟硬體資源、人力支援及心理氛圍的營造。硬體資源方面，學校備有電腦教室、資優班與圖書館，提供豐富的書籍與歷屆研究作品，便利學生查詢與參考；人力資源部分，由行政人員與班級導師共同支持研究相關的行政事務與課程安排，形成穩固的協作網絡，協助學生有序而高效地進行研究活動；而在心理環境方面，教師尊重學生的興趣，採分組研討方式鼓勵討論交流，並給予充分的發揮空間，進而塑造出舒適、支持性的研究氛圍，使學生能在愉悅而具啟發性的環境中穩健成長、拓展研究視野。

4. 評量與回饋機制：

課程設計評量與回饋機制，於每學年舉辦校內獨立研究發表會，為學生提供公開展示研究成果的平台。透過檔案評量、學生自我評量或教師觀察與質性描述等多元評量方式，學生得以清楚掌握自身的研究能力與特質。

第二階段 獨立研究階段

壹、研究動機與研究目的

這學期數學課在講解圓周長與面積時，我盯著課本上代表圓周率的符號 π ，對老師所說「這是一個無限不循環的無理數」感到不可思議。既然它永遠算不完，那我們現在使用的 3.14 究竟是如何求得的？這個疑問如種子般埋入心中。適逢學校安排獨立研究課程，雖然組員們曾提出生物觀察或物理實驗等想法，但大家始終對親手算出那個神祕的 π 念念不忘，最終一致通過，決定將尋找圓周率作為本次獨立研究的主題。

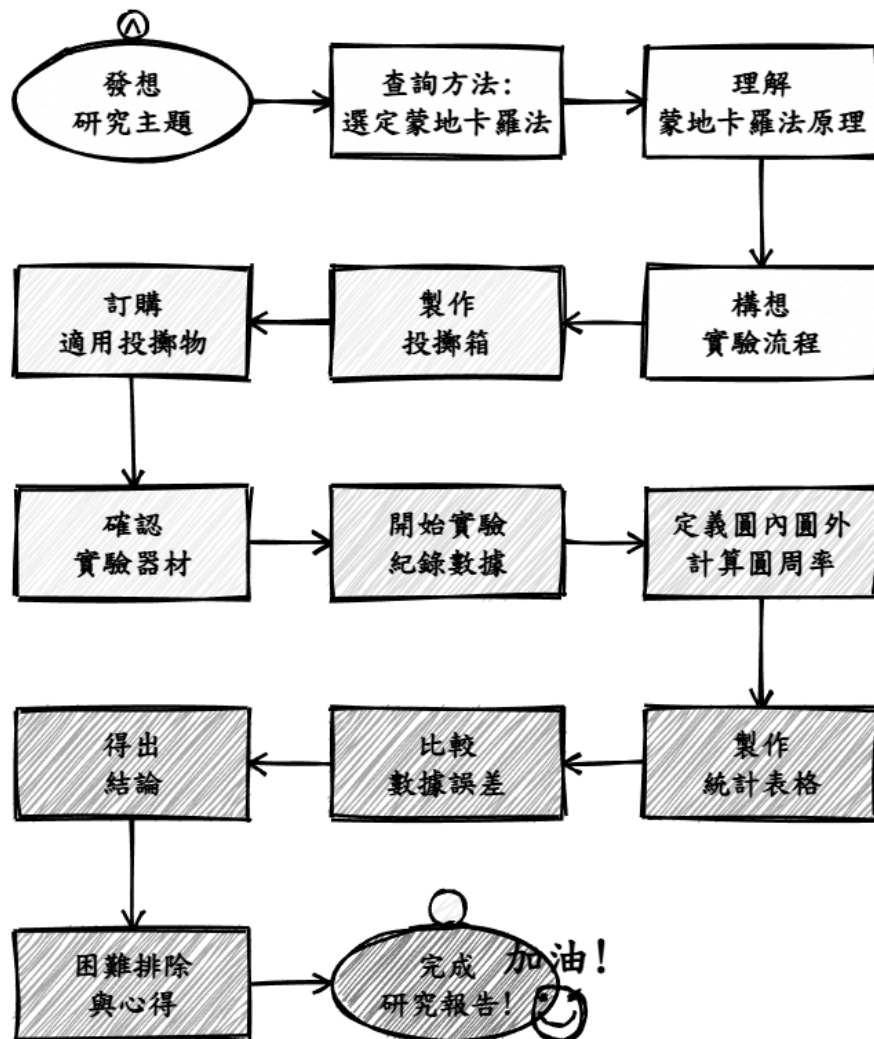
確定方向後，我們查閱資料發現古人如祖沖之利用割圓術計算圓周率。然而深入研究後，發現該方法涉及我們尚未學過的三角函數及繁複的開根號計算，難度遠超我們負荷。正當灰心之際，我們在網路上發現了蒙地卡羅法。這是一種利用機率統計與隨機投擲來解決數學問題的方法，僅需具備基礎統計與除法概念，無需艱深公式即可執行。這對我們而言是完美的解決方案，因此我們毅然放棄複雜的割圓術，轉而採用蒙地卡羅法進行實驗。

雖然原理簡單，但在設計實驗細節時，我們面臨了一個實際問題：當投擲物剛好壓在圓的邊界線上時，應判定為圓內還是圓外？這引發了小組激烈討論。為了秉持實驗精神，我們決定不流於爭辯，而是將此判定標準轉化為實驗變因。我們設計了兩種不同的計分標準來進行比較：第一種是寬鬆計算法，將所有落在線上的數值皆視為圓內；第二種是嚴格計算法，將所有落在線上的數值視為圓外。

本研究旨在透過人工投擲方式實地操作，分別以上述寬鬆與嚴格兩種標準計算圓周率，並與公認值 3.141592 進行比較，找出誤差最小的方法。此外，我們也將設計電腦程式模擬投擲情形，同樣套用這兩種計算法，再將電腦模擬結果與人工投擲數據進行交叉比對。我們期望透過實際動手做與數據分析的過程，探討不同判定方式對準確度的影響，進而揭開蒙地卡羅法求取圓周率的奧秘。

貳、擬定正式計畫、研究問題及工作進度表

一、擬定正式計畫



二、研究問題

我們的研究是利用實地實驗的方式檢驗：

1. 我們把落在線上的數字視為圓內，然後利用蒙地卡羅法求出圓周率。
2. 我們把落在線上的數字視為圓外，然後利用蒙地卡羅法求出圓周率。
3. 比較第一點跟第二點的誤差值，並討論結果。
4. 製作程式，用程式跑出來的理論值來和實地實驗結果比較。

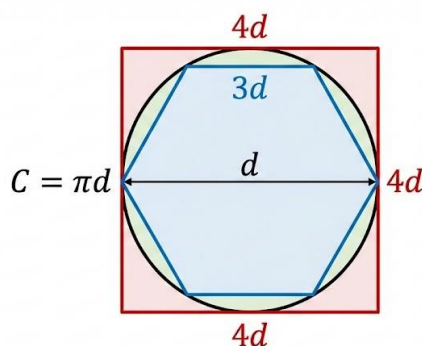
三、工作進度表

日期	預定進度
9/4 - 9/5	💡 發想研究主題
9/11 - 9/26	🔍 查詢算圓周率的方法 並選擇蒙地卡羅法
10/2 - 10/3	📖 理解蒙地卡羅法 算圓周率
10/9 - 10/16	📦 製作投擲箱
10/17 - 10/23	🛒 訂購投擲物
10/30	✓ 確認實驗器材
10/31 - 11/7	📝 開始實驗 並紀錄數據
11/7 - 11/21	📊 計算圓周率（線上視為圓內 vs 圓外） 製作表格，比較誤差值
11/21 - 11/28	☆ 得出結論
11/29 - 12/5	📁 寫出遇到的困難跟實驗心得 並完成報告

參、相關文獻摘要及實驗工具

一.割圓術

我們查到我們該如何不去利用實際的測量，而去證明說圓周率小於 4，傳統上我們可以利用尺或其他東西實際測量，而知道圓周率比 4 小。而我們就會延伸出一個問題，我們是否能用最純粹的幾何邏輯來證明圓周率小於 4。首先我們先想想第一個問題，我們可以設定一個半徑為 1cm 的圓，再設定一個 2cm 的正方形我們可以發現這個正方形可以將這個圓完全包覆起來，而圓的面積為 π 正方形面積為 4，所以我們便可以得知 $\pi < 4$ 。



內接正六邊形周長 ($3d$) < 圓周長 (πd) < 外切正方形周長 ($4d$)

因此： $3 < \pi < 4$

二.蒙地卡羅法：用「運氣」找到答案的數學魔術

蒙地卡羅法 (Monte Carlo Method)，又叫做統計試驗法，其實是一個超級酷的方法，專門用來解決那些很難用普通數學公式算出來的問題。你可以把它想像成一種「用不斷的猜謎和運氣（隨機）來找出答案」的技術。我們特別需要它，是因為在真實世界中，很多事情都有「運氣」的成分，像是武器的命中機率、東西會不會剛好壞掉等等。

這個方法是以機率和統計學為基礎，最主要的工作就是讓電腦去進行大量的「隨機試驗」。電腦會一直不斷地製造出很多隨機數字（就像一直丟骰子一樣），然後我們就可以從這些大量的試驗結果中，計算出一個非常接近真實情況的機率值，例如平均能打中多少目標。這樣就能更準確地評估一個裝備到底厲不厲害！其實，早在 18 世紀，

就有一個叫蒲豐的學者，用「丟針」的方式來模擬圓周率的值，這就是蒙地卡羅法最早期的應用概念。

三、實驗工具

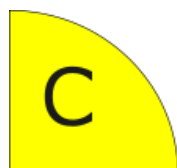
- 1.投擲箱(邊 50x20CM、底 50x50CM)，我們在投擲箱裡面畫上四分之一圓，依照投擲物的落點，去統計圓內、圓外及線上的數據。



- 2.投擲物(每個約 1.5cm 直徑，厚度 0.5cm 的造型橡皮擦)，我們利用造型橡皮擦不規則的邊緣彈跳，增加隨機性。



- 3.筆記本電腦(使用 google sheet 計算表、Gemini AI 產生圖表、notebookLM AI 整理參考資料、python 製作程式模擬蒙地卡羅法運行過程)
- 4.蒙地卡羅法應用在圓周率上：在這個研究中我們列出了四種數字。投擲物的落點落在 1/4 圓內的數量[C]:我們生成隨機落點時，如果投擲物落在 1/4 圓內的話，我們則會把它設為[C]。



四分之一圓面積

投擲物的總數量[N]:我們生成隨機落點時，我們把我們生成的點的總數量則設為[N]。



正方形跟四分之一圓的面積:我們根據推算，假設在一個邊長為 2 的正方形內剛好放進一個四分之一圓，那麼我們可以說正方形面積為(2X2=4)而四分之一圓的面積為(2*2* π *1/4= π)



圓周上[A]:由於我們的實驗目的是判斷在圓周上算圓內還是圓外，所以我們增加了一個圓周的數字。

所以我們可以說 C:N=圓內:總數= π :4 而後面的算式如下:



C 四分之一圓面積
N 正方形面積

$$\text{C} : \text{N} = 2 \times 2 \times \pi \times 1/4 : 2 \times 2$$

$$\text{C} : \text{N} = \pi : 4$$

內項乘積=外項乘積

$$N \times \pi = 4 \times C$$

兩邊除以N(N不等於0)

$$\pi = \frac{4C}{N}$$

肆、實驗設計與流程

實驗目的：驗證蒙地卡羅法求圓周率。

實驗問題：如第二章所示。

一、實驗設計：

- 1.用投擲物體的實地實驗，收集物體落於模型中的位置。
- 2.記錄下數據，完全落在圓內的、落於圓周上的以及一次全部投擲次數。
- 3.提出問題：落在圓周上的數字，算成圓內或算成圓外，哪個比較貼近 π ?

二、實驗流程：

- 1.正方形紙箱底部繪製一個四分之一圓，準備好投擲物。
- 2.我們準備好投擲箱，前方一個摺疊墊子的距離開始實驗，我們投擲的方式是一次把一小把的投擲物丟進投擲箱，並且實驗者的手的高度大約要在胸前高度，並手臂成彎曲狀(避免實驗者伸長手臂丟造成距離上的誤差)，之後往前撒或拋出投擲物投擲橡皮擦，一次N個。



- 3.等待橡皮擦彈跳完記錄數據。而我們的紀錄方式是：把掉出投擲箱的投擲物忽略不計之後，紀錄我們的圓內、圓外、線上。最後我們會把投擲物撿回來。



4.將數據輸入電腦，用試算表及試算表公式計算各種比較數據。



我們一開始先把箱子跟墊子放好後，我們會先投擲出投擲物(不限投擲方法)，隨後我們紀錄了 C 圓內 N 總數 A 線上之後，我們就分出了(線上視為圓內)跟(線上視為圓外)這兩種數據，最後算出他們的誤差值跟將這兩種數據比較，來求出落在線上到底要不要算在圓內。

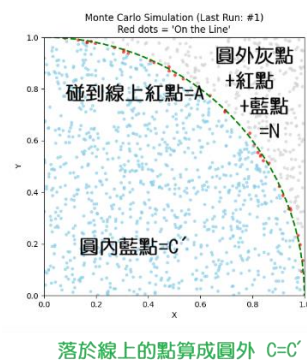
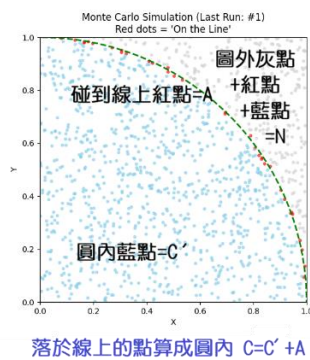


圖 4-1 不同計算法示意圖

伍、資料分析

因為每次投擲時常會有橡皮擦落於範圍外，所以在做累進統計時常常無法剛好取到正數或剛好的數字。所以使用電腦模擬時，取累進總數接近正百的數字來模擬。

一、較寬鬆的設定：落於圓周線上的橡皮擦視為圓內

將原始數據累進計算後，先將落於圓周線上的橡皮擦視為圓內，再代入蒙地卡羅法公式計算圓周率：

$$\pi = 4 * C / N$$

$$C = (\text{累進落於圓內橡皮擦數} + \text{累進落於圓周線上橡皮擦數})$$

$$N = \text{累進落於正方形紙箱內橡皮擦總數}$$

在圓內	在邊緣 (圓外)	在線上	總累進 次數	累進 圓內+線 上	累進 總次 數	累進 pi (4*內/總)	和圓周率相差 (計算值-pi)取絕對 值	誤差 (相差 /pi)
24	8	3	35	27	35	3.085714	0.055878	1.78
23	14	7	79	57	79	2.886076	0.255516	8.13
24	6	3	112	84	112	3.000000	0.141592	4.51
11	6	3	132	98	132	2.969697	0.171895	5.47
20	6	2	160	120	160	3.000000	0.141592	4.51
23	3	4	190	147	190	3.094737	0.046855	1.49
37	5	3	235	187	235	3.182979	0.041387	1.32
31	9	6	281	224	281	3.188612	0.047020	1.50
29	8	7	325	260	325	3.200000	0.058408	1.86
25	13	4	367	289	367	3.149864	0.008272	0.26
25	5	4	401	318	401	3.172070	0.030478	0.97
15	7	4	427	337	427	3.156909	0.015317	0.49
29	9	9	474	375	474	3.164557	0.022965	0.73
25	13	5	517	405	517	3.133462	0.008130	0.26
29	8	6	560	440	560	3.142857	0.001265	0.04
33	11	3	607	476	607	3.136738	0.004854	0.15
5	2	0	614	481	614	3.133550	0.008042	0.26
37	8	2	661	520	661	3.146747	0.005155	0.16
5	2	0	668	525	668	3.143713	0.002121	0.07
6	3	0	677	531	677	3.137371	0.004221	0.13
5	1	0	683	536	683	3.139092	0.002500	0.08
7	1	1	692	544	692	3.144509	0.002917	0.09

7	1	0	700	551	700	3.148571	0.006979	0.22
8	0	0	708	559	708	3.158192	0.016600	0.53
7	3	0	718	566	718	3.153203	0.011611	0.37
4	2	0	724	570	724	3.149171	0.007579	0.24
5	4	0	733	575	733	3.137790	0.003802	0.12
9	7	2	751	586	751	3.121172	0.020420	0.65
14	3	3	771	603	771	3.128405	0.013187	0.42
18	2	0	791	621	791	3.140329	0.001263	0.04
16	2	0	809	637	809	3.149567	0.007975	0.25
10	4	1	824	648	824	3.145631	0.004039	0.13
8	7	2	841	658	841	3.129608	0.011984	0.38
11	2	1	855	670	855	3.134503	0.007089	0.23
13	4	0	872	683	872	3.133028	0.008564	0.27
13	5	2	892	698	892	3.130045	0.011547	0.37
11	8	1	912	710	912	3.114035	0.027557	0.88
7	4	0	923	717	923	3.107259	0.034333	1.09
15	5	0	943	732	943	3.104984	0.036608	1.17
11	7	1	962	744	962	3.093555	0.048037	1.53
12	2	3	979	759	979	3.101124	0.040468	1.29
11	4	3	997	773	997	3.101304	0.040288	1.28
13	5	2	1017	788	1017	3.099312	0.042280	1.35
16	3	1	1037	805	1037	3.105111	0.036481	1.16
12	3	1	1053	818	1053	3.107312	0.034280	1.09
16	4	0	1073	834	1073	3.109040	0.032552	1.04
13	5	2	1093	849	1093	3.107045	0.034547	1.10
11	7	0	1111	860	1111	3.096310	0.045282	1.44
12	5	3	1131	875	1131	3.094607	0.046985	1.50
10	7	1	1149	886	1149	3.084421	0.057171	1.82
13	4	3	1169	902	1169	3.086399	0.055193	1.76
14	5	0	1188	916	1188	3.084175	0.057417	1.83
13	4	3	1208	932	1208	3.086093	0.055499	1.77
14	5	1	1228	947	1228	3.084691	0.056901	1.81
13	4	2	1247	962	1247	3.085806	0.055786	1.78
9	6	2	1264	973	1264	3.079114	0.062478	1.99
7	7	4	1282	984	1282	3.070203	0.071389	2.27
13	4	2	1301	999	1301	3.071483	0.070109	2.23

11	5	3	1320	1013	1320	3.069697	0.071895	2.29
16	2	1	1339	1030	1339	3.076923	0.064669	2.06
8	5	5	1357	1043	1357	3.074429	0.067163	2.14
13	4	2	1376	1058	1376	3.075581	0.066011	2.10
9	7	4	1396	1071	1396	3.068768	0.072824	2.32
12	4	2	1414	1085	1414	3.069307	0.072285	2.30
12	5	2	1433	1099	1433	3.067690	0.073902	2.35
16	2	2	1453	1117	1453	3.075017	0.066575	2.12
14	5	1	1473	1132	1473	3.073999	0.067593	2.15
15	4	0	1492	1147	1492	3.075067	0.066525	2.12
12	6	2	1512	1161	1512	3.071429	0.070163	2.23

表 5-1 投擲累進表(較寬鬆設定)

以總累進次數為 X 軸，以累進 $\pi(4 \times \text{內}/\text{總})$ 為 Y 軸，繪製折線圖後，再將已知圓周率(以 3.141592 為例)繪製一條虛線。我們發現在累進次數較少的時候(400 次之前)，它與圓周率的值相差甚多且波動較大；投擲數量累進在 500 到 900 次之間較貼近圓周率，隨著之後慢慢誤差值趨近穩定。

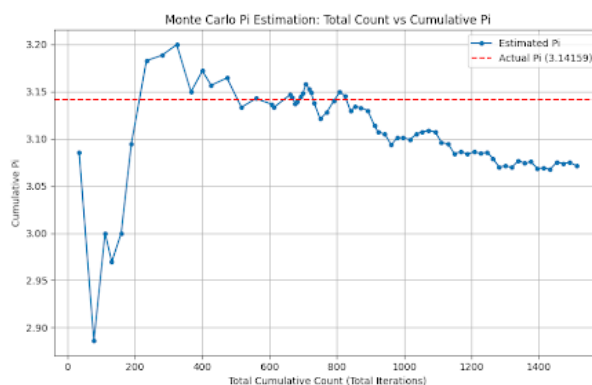


圖 5-1 寬鬆設定下的圓周率變化圖

我們再以誤差百分比為 Y 軸繪製一張圖，也證明了我們對數據的分析結果。使用以下公式計算誤差值：

誤差值 = (投擲所得的圓周率 / 3.141592) 取絕對值

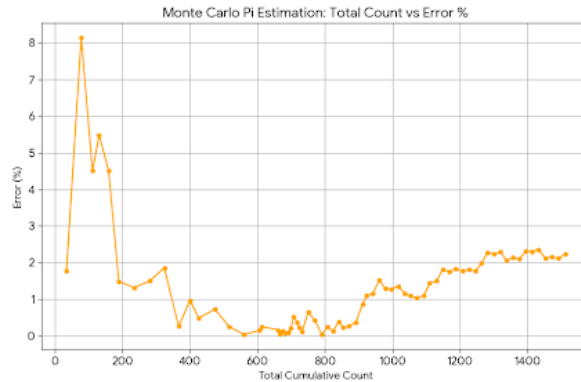


圖 5-2 寬鬆設定下的誤差變化圖

二、較嚴格的設定：落於圓周線上的橡皮擦視為圓外不計

接著使用相同的數據，但是將落於圓周線上的橡皮擦視為圓外(不計)，再代入蒙地卡羅法公式計算圓周率：

$$\pi = 4 * C' / N$$

C' = 累進落於圓內橡皮擦數

N = 累進落於正方形紙箱內橡皮擦總數

在圓內	在邊緣 (圓外)	總累進 次數	累進 圓內	累進 總次數	累進 pi (4*內/總)	和圓周率相差 (計算值-pi) 取絕 對值	誤差 (相差/pi)
24	8	3	35	24	2.742857	0.398735	12.69%
23	14	7	79	47	2.379747	0.761845	24.25%
24	6	3	112	71	2.535714	0.605878	19.29%
11	6	3	132	82	2.484848	0.656744	20.90%
20	6	2	160	102	2.550000	0.591592	18.83%
23	3	4	190	125	2.631579	0.510013	16.23%
37	5	3	235	162	2.757447	0.384145	12.23%
31	9	6	281	193	2.747331	0.394261	12.55%
29	8	7	325	222	2.732308	0.409284	13.03%
25	13	4	367	247	2.692098	0.449494	14.31%
25	5	4	401	272	2.713217	0.428375	13.64%
15	7	4	427	287	2.688525	0.453067	14.42%
29	9	9	474	316	2.666667	0.474925	15.12%
25	13	5	517	341	2.638298	0.503294	16.02%
29	8	6	560	370	2.642857	0.498735	15.88%
33	11	3	607	403	2.655684	0.485908	15.47%

5	2	0	614	408	614	2.657980	0.483612	15.39%
37	8	2	661	445	661	2.692890	0.448702	14.28%
5	2	0	668	450	668	2.694611	0.446981	14.23%
6	3	0	677	456	677	2.694239	0.447353	14.24%
5	1	0	683	461	683	2.699854	0.441738	14.06%
7	1	1	692	468	692	2.705202	0.436390	13.89%
7	1	0	700	475	700	2.714286	0.427306	13.60%
8	0	0	708	483	708	2.728814	0.412778	13.14%
7	3	0	718	490	718	2.729805	0.411787	13.11%
4	2	0	724	494	724	2.729282	0.412310	13.12%
5	4	0	733	499	733	2.723056	0.418536	13.32%
9	7	2	751	508	751	2.705726	0.435866	13.87%
14	3	3	771	522	771	2.708171	0.433421	13.80%
18	2	0	791	540	791	2.730721	0.410871	13.08%
16	2	0	809	556	809	2.749073	0.392519	12.49%
10	4	1	824	566	824	2.747573	0.394019	12.54%
8	7	2	841	574	841	2.730083	0.411509	13.10%
11	2	1	855	585	855	2.736842	0.404750	12.88%
13	4	0	872	598	872	2.743119	0.398473	12.68%
13	5	2	892	611	892	2.739910	0.401682	12.79%
11	8	1	912	622	912	2.728070	0.413522	13.16%
7	4	0	923	629	923	2.725894	0.415698	13.23%
15	5	0	943	644	943	2.731707	0.409885	13.05%
11	7	1	962	655	962	2.723493	0.418099	13.31%
12	2	3	979	667	979	2.725230	0.416362	13.25%
11	4	3	997	678	997	2.720160	0.421432	13.41%
13	5	2	1017	691	1017	2.717797	0.423795	13.49%
16	3	1	1037	707	1037	2.727097	0.414495	13.19%
12	3	1	1053	719	1053	2.731244	0.410348	13.06%
16	4	0	1073	735	1073	2.739981	0.401611	12.78%
13	5	2	1093	748	1093	2.737420	0.404172	12.87%
11	7	0	1111	759	1111	2.732673	0.408919	13.02%
12	5	3	1131	771	1131	2.726790	0.414802	13.20%
10	7	1	1149	781	1149	2.718886	0.422706	13.46%
13	4	3	1169	794	1169	2.716852	0.424740	13.52%
14	5	0	1188	808	1188	2.720539	0.421053	13.40%

13	4	3	1208	821	1208	2.718543	0.423049	13.47%
14	5	1	1228	835	1228	2.719870	0.421722	13.42%
13	4	2	1247	848	1247	2.720128	0.421464	13.42%
9	6	2	1264	857	1264	2.712025	0.429567	13.67%
7	7	4	1282	864	1282	2.695788	0.445804	14.19%
13	4	2	1301	877	1301	2.696387	0.445205	14.17%
11	5	3	1320	888	1320	2.690909	0.450683	14.35%
16	2	1	1339	904	1339	2.700523	0.441069	14.04%
8	5	5	1357	912	1357	2.688283	0.453309	14.43%
13	4	2	1376	925	1376	2.688953	0.452639	14.41%
9	7	4	1396	934	1396	2.676218	0.465374	14.81%
12	4	2	1414	946	1414	2.676096	0.465496	14.82%
12	5	2	1433	958	1433	2.674110	0.467482	14.88%
16	2	2	1453	974	1453	2.681349	0.460243	14.65%
14	5	1	1473	988	1473	2.682960	0.458632	14.60%
15	4	0	1492	1003	1492	2.689008	0.452584	14.41%
12	6	2	1512	1015	1512	2.685185	0.456407	14.53%

表 5-2 投擲累進表(較嚴格設定)

用相同的條件繪製圖形，發現誤差數字在約 700 次之後漸漸趨於平緩，但是實際計算出來的圓周率卻與實際圓周率相差較遠(比起較寬鬆的設定)，可以得知使用較嚴格的設定，總投擲數越大，它越趨近一個較穩定的數值。但是這個數值和實際圓周率有較大的差別。

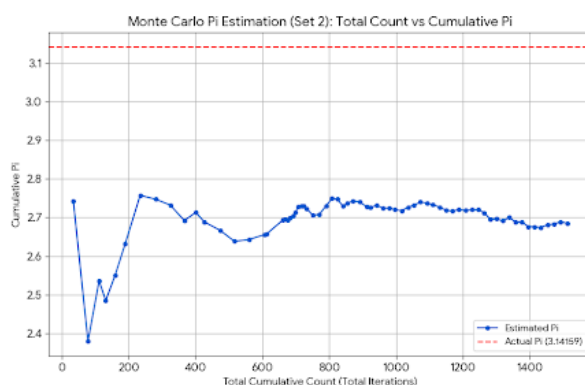


圖 5-3 嚴格設定下的圓周率變化圖

我們再單獨取出誤差值來看，得到以下圖形。而它隨著投擲數增加的誤差值也是一樣的漸漸穩定。使用以下公式計算誤差值：

誤差值=(投擲所得的圓周率/3.141592)取絕對值

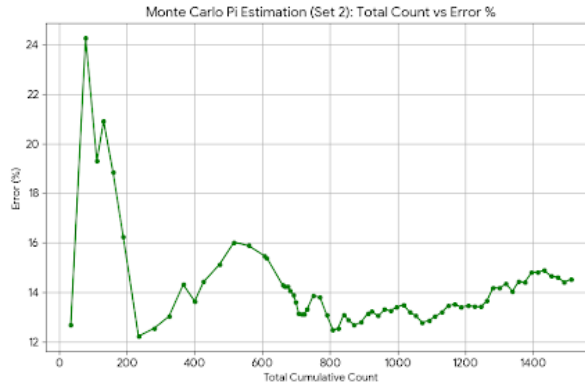


圖 5-4 嚴格設定下的誤差變化圖

三、自製的電腦程式模擬投擲數據

在方塊中繪製一個有寬度的四分之一圓周線，並模擬有體積的投擲物體隨機落於正方形區域中，再記錄落於圓內及圓周線上的數字，並記錄下總投擲數，使用第一、二節相同的公式計算寬鬆及嚴格設定下的圓周率和誤差。以下為本次的 python 模擬拋擲程式：

```
import random
import math
import pandas as pd
import matplotlib.pyplot as plt

# --- 設定實驗參數 ---
NUM_EXPERIMENTS = 1 # 實驗總組數
TOTAL_THROWS = 10000000 # 每一組實驗的投擲次數
LINE_THICKNESS = 0.01 # 模擬線條的粗細(模擬投擲橡皮擦產生的誤差)
REAL_PI = 3.1415926

def run_simulation():
    print(f"🚀 開始執行 {NUM_EXPERIMENTS} 組模擬實驗 (每組 {TOTAL_THROWS} 次投擲)...")

    all_results = [] # 用來存放每一組的結果

    # 預備變數供最後畫圖用
    last_points_x = []
    last_points_y = []
    last_colors = []

    for exp_id in range(1, NUM_EXPERIMENTS + 1):
        # 重置單次實驗的計數器
```

```

count_inside_clear = 0
count_on_line = 0
count_outside_clear = 0

# 清空畫圖數據 (只保留最後一組)
current_points_x = []
current_points_y = []
current_colors = []

for i in range(TOTAL_THROWS):
    # 1. 隨機產生落點
    x = random.uniform(0, 1)
    y = random.uniform(0, 1)

    # 2. 計算落點到圓心的距離
    distance = math.sqrt(x**2 + y**2)

    current_points_x.append(x)
    current_points_y.append(y)

    # 3. 判定位置
    if distance < (1.0 - LINE_THICKNESS):
        count_inside_clear += 1
        current_colors.append('skyblue')
    elif distance > (1.0 + LINE_THICKNESS):
        count_outside_clear += 1
        current_colors.append('lightgray')
    else:
        count_on_line += 1
        current_colors.append('red')

# --- 計算該組結果 ---
# 加計組：壓線視為圓內
loose_hits = count_inside_clear + count_on_line
pi_loose = 4 * (loose_hits / TOTAL_THROWS)
error_loose = abs(pi_loose - REAL_PI) / REAL_PI * 100

# 忽略組：壓線視為圓外
strict_hits = count_inside_clear
pi_strict = 4 * (strict_hits / TOTAL_THROWS)
error_strict = abs(pi_strict - REAL_PI) / REAL_PI * 100

# 將數據存入列表
all_results.append({
    "實驗編號": exp_id,
    "總投擲數": TOTAL_THROWS,
    "圓內次數": count_inside_clear,
    "圓外次數": count_outside_clear,
    "壓線次數": count_on_line,
    "Pi (加計)": round(pi_loose, 5),
    "誤差 (加計)%": round(error_loose, 2),
    "Pi (忽略)": round(pi_strict, 5),
    "誤差 (忽略)%": round(error_strict, 2)
})

# 如果是最後一組，保存數據用於畫圖
if exp_id == NUM_EXPERIMENTS:
    last_points_x = current_points_x

```

```

        last_points_y = current_points_y
        last_colors = current_colors

# --- 儲存結果至 CSV ---
df = pd.DataFrame(all_results)
csv_filename = "result.csv"
# encoding='utf-8-sig' 確保 Excel 打開中文不亂碼
df.to_csv(csv_filename, index=False, encoding='utf-8-sig')

print(f"\n✅ 已完成 {NUM_EXPERIMENTS} 組數據生成。")
print(f"📁 數據已存檔至: {csv_filename}")
print("\n📊 前 5 筆數據預覽:")
print(df.head().to_string(index=False))

# --- 畫出最後一次實驗的圖 (示意用) ---
plt.figure(figsize=(6, 6))
plt.scatter(last_points_x, last_points_y, c=last_colors, s=10, alpha=0.6)
circle = plt.Circle((0, 0), 1, color='green', fill=False, linewidth=2, linestyle='--')
plt.gca().add_patch(circle)
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.title(f'Monte Carlo Simulation (Last Run: #{NUM_EXPERIMENTS})\nRed
dots = 'On the Line')
plt.xlabel("X")
plt.ylabel("Y")
plt.savefig('experiment_plot.png')
print("\n📈 最後一組實驗的分布圖已產生: experiment_plot.png")

if __name__ == "__main__":
    run_simulation()

```

使用電腦程式模擬投擲，因為人工投擲時累進次數比較難取整數值，所以取接近整百的累進總投擲數，交給電腦模擬投擲，得到以下數據：

實驗 編號	投擲 數	圓內 次數	圓外 次數	壓線 次數	Pi (寬鬆)	誤差 (寬鬆)	Pi (嚴格)	誤差 (嚴格)
1	112	84	21	7	3.250000	3.45	3.000000	4.51
2	190	149	36	5	3.242110	3.20	3.136840	0.15
3	281	202	71	8	2.989320	4.85	2.875440	8.47
4	401	320	72	9	3.281800	4.46	3.192020	1.61
5	517	399	101	17	3.218570	2.45	3.087040	1.74
6	607	472	117	18	3.229000	2.78	3.110380	0.99
7	700	551	137	12	3.217140	2.40	3.148570	0.22
8	809	585	198	26	3.021010	3.84	2.892460	7.93
9	912	701	188	23	3.175440	1.08	3.074560	2.13
10	1017	796	195	26	3.233040	2.91	3.130780	0.34
11	1093	843	220	30	3.194880	1.70	3.085090	1.8
12	1208	930	234	44	3.225170	2.66	3.079470	1.98

13	1301	989	277	35	3.148350	0.22	3.040740	3.21
14	1414	1095	263	56	3.256010	3.64	3.097600	1.4
15	1512	1145	321	46	3.150790	0.29	3.029100	3.58

表 5-3 電腦模擬投擲數據

接著我們以投擲數為 X 軸，以計算出的 Pi 值為 Y 軸。圓點(--) 代表 Pi(寬鬆) 的數值，而方塊(-.)代表 Pi(嚴格) 的數值，製作折線圖，觀察數值的變化。(上方折線：寬鬆)

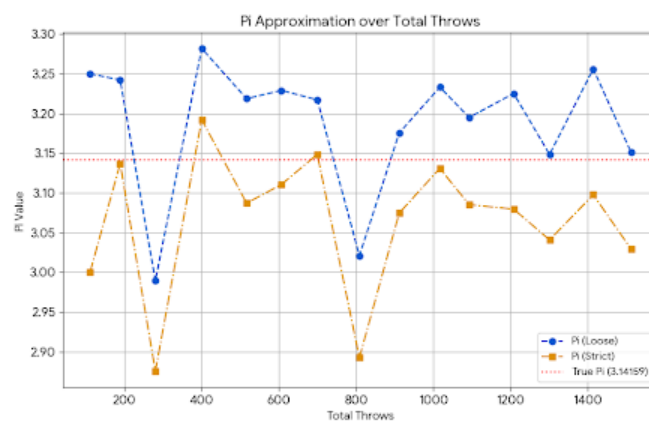


圖 5-5 投擲數與模擬 pi 值的變化趨勢圖

四、使用網路上的範例程式模擬

維基百科的範例程式(沒有考慮點的大小，把圓周線上算成圓內)

```
import numpy as np

# 生成 10 的 9 次方個隨機點
num_points = 10**9
points = np.random.rand(num_points, 2)

# 計算點到原點的距離
distances = np.sqrt(points[:,0]**2 + points[:,1]**2)

# 計算落在單位圓內的點的數量
points_inside_circle = np.sum(distances <= 1)

# 蒙地卡羅方法計算圓周率
pi_estimate = 4 * points_inside_circle / num_points

print(pi_estimate)
```

使用電腦程式模擬投擲，因為人工投擲時累進次數比較難取整數值，所以取接近整百的累進總投擲數，交給電腦模擬投擲，得到以下數據：

實驗編號	總投擲數	Pi	誤差
1	112	3.214286	1.02
2	190	2.800000	0.89
3	281	3.160142	1.01
4	401	3.012469	0.96
5	517	3.110251	0.99
6	607	3.149918	1.00
7	700	3.091429	0.98
8	809	3.179234	1.01
9	912	3.144737	1.00
10	1017	3.162242	1.01
11	1093	3.125343	0.99
12	1208	3.162252	1.01
13	1301	3.154497	1.00
14	1414	3.213579	1.02
15	1512	3.161376	1.01

表 5-4 電腦模擬投擲數據

因為這個程式並沒有考量投擲物的體積和圓周線的寬度，所以只有一組數據：

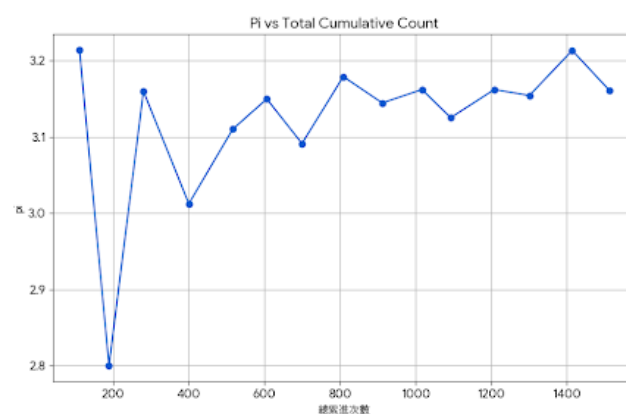


圖 5-6 投擲數與模擬 pi 值的變化趨勢圖

陸、研究結果與討論

一、研究結果

依第五章的研究數據所示，以下將使用名詞定義為：

圓內：投擲或模擬投擲且落於四分之一圓內部。

線上：投擲或模擬投擲且落於四分之一圓之圓周線上並與其接觸。

寬鬆：指把線上的數字計算成圓內。

嚴格：把線上的數字算成圓外。

π 值：3.141592

1.我們把落在線上的數字視為圓內，然後利用蒙地卡羅法求出圓周率。

使用寬鬆的算法，利用原始數據計算每單次投擲的平均圓周率（單次投擲原始數據如附件一），再將每次的累進數據視為獨立事件，求平均圓周率。我們發現使用個別單次計算的平均圓周率，比使用累進數據求得的平均圓周率還要不準確，這結果與上一章的數據分析結果吻合，在少量取樣時，使用蒙地卡羅法求圓周率，會得到較不穩定的折線；但在取樣數多時，折線會較穩定。

	累進投擲	單次投擲
平均 π	3.106373	3.043528
平均相差值 (計算值- π)取絕對值	0.043627	0.098064
平均誤差 (相差/ π)	1.39%	3.12%

表 6-1 寬鬆求法時單次與累進投擲相比較

圖左為第五章數據分析得到之圓周率數值；圖右為誤差值。但示意圖僅止於表示視覺上的直觀判定，並不是數學上數據分析結果。

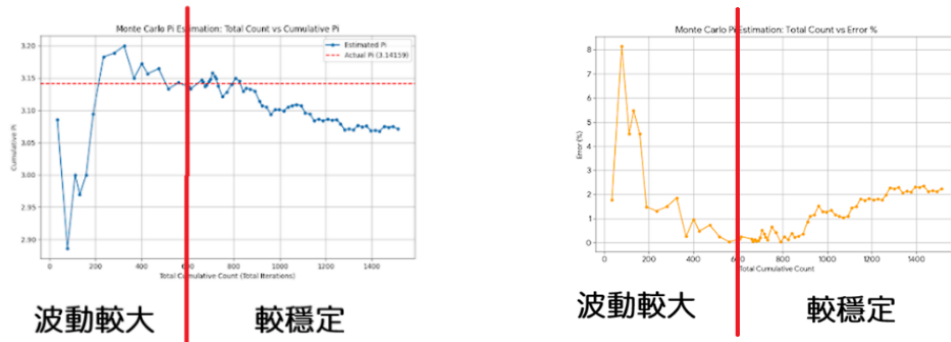


圖 6-1 寬鬆算法的折線穩定示意圖

2.我們把落在線上的數字視為圓外，然後利用蒙地卡羅法求出圓周率。

使用嚴格的算法，利用原始數據計算每單次投擲的平均圓周率，再將每次的累進數據視為獨立事件，求平均圓周率。我們發現使用嚴格算法時，單次投擲和累進投擲取得的各項數值非常相似。雖然求得的平均誤差較大(超過 10%)，但是使用不同的計算法，會得到類似的結果。但是它偏離實際圓率率較遠，可能需要加上一個常數，調整後才能得到較正確的圓周率數值。

	累進投擲	單次投擲
平均 π	2.696240	2.706182
平均相差值 (計算值- π)取絕對值	0.445352	0.435410
平均誤差 (相差/ π)	14.18%	13.86%

表 6-2 嚴格求法時單次與累進投擲相比較

圖左為圓周率數值；圖右為誤差值。但示意圖同樣僅止於表示視覺上的直觀判定，並不是數學上數據分析結果。

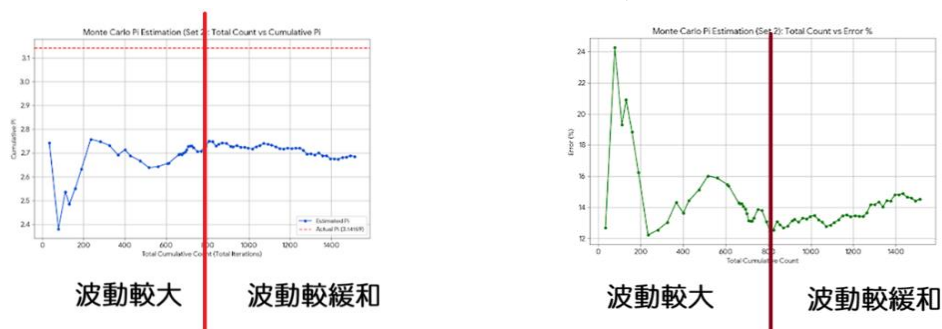


圖 6-2 嚴格算法的折線穩定示意圖

3.比較第一點跟第二點的誤差值，並討論結果。

將每次的累進數據視為獨立事件，使用累進將寬鬆計算法與嚴格計算法求得的圓周率數字統一合成一張圖來比較，更能看出人工投擲的情況下，用兩種不同計算法，其實它們的折線圖在視覺上是差不多的，只是使用嚴格條件計算的圓周率，離實際圓周率較遠。但是由先前結論得知，雖然嚴格計算法取得的圓周率較不可靠，但若將其加上一個常數(兩種計算法的相差值)移動折線，便可以得到一個有效的圓周率。而且如上述第 2 個問題結果，嚴格的算法不管是將每一次投擲視為獨立事件，或是將每次累進數字視為獨立事件，算出來的 π 值非常相近，顯示嚴格算法比起寬鬆算法較有穩定性，計算時採樣時得到的數據較可靠；而寬鬆算法比較貼近真實圓周率，可以用來校正嚴格算法的數值。

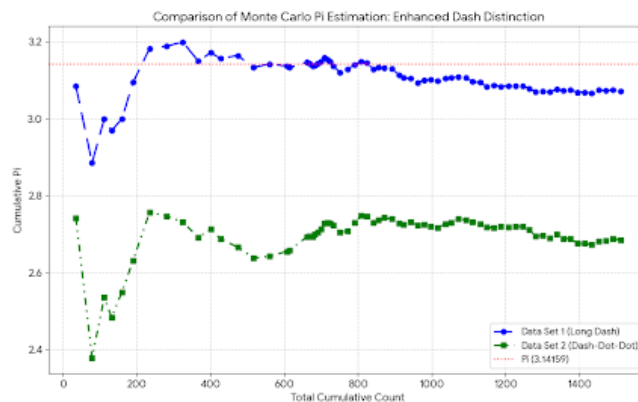


圖 6-3 寬鬆計算法與嚴格計算法(下方折線)圓周率比較圖

4.製作程式，用程式跑出來的理論值來和實地實驗結果比較。

我們使用了兩種程式，A 程式是模擬每個點有體積且圓周線有體積，點隨機落於正方形內部時需考量是否有沾到圓周線的考量；另 B 程式是利用維基百科列出來的範例程式。投擲次數參考人工投擲時的累進次數，以貼近”百”為取樣考量。

左圖為 A 程式得到的結果，一樣是寬鬆算法得到的折線會在嚴格算法的上方。

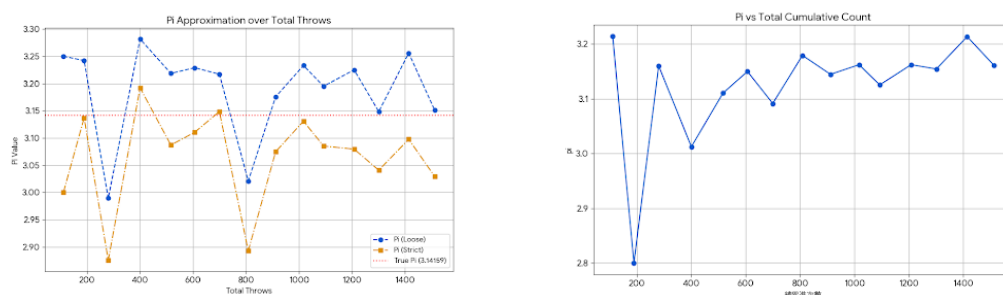


圖 6-4 電腦亂數落點算出來的圓周率圖

A 程式的點有體積，線也有體積，但它是用電腦取亂數，所以得到的誤差較小於人工投擲，這也和實際情境一樣，人工投擲會摻雜較多不可控的變因。

	寬鬆算法	嚴格算法
平均 π	3.188842	3.065339
平均相差值 (計算值- π)取絕對值	0.08	0.08
平均誤差 (相差/ π)	2.66%	2.67%

表 6-3 使用 A 程式模擬投擲的誤差比較

B 程式是完美程式，不計體積的點，用來模擬投擲時，會得到較高的準確度和較理論的圓周率值。

平均 π	3.122783
平均相差值 (計算值- π)取絕對值	0.06
平均誤差 (相差/ π)	1.81%

表 6-4 使用 A 程式模擬投擲的誤差

比較手投與電腦投的數據，我們可以得到一個結論，就是投擲樣本數很少的時候，算出來的圓周率數字波動很大；而且累進算法得到的折線相較於模擬單次投擲的數字穩定。也就是說，蒙地卡羅法求圓

周率適合投擲大量數字後，才能得到較準確的圓周率值；而且用理想狀況下的模擬投擲，可以得到更加準確的圓周率值。

二、討論

1.研究結論

我們的實地實驗是人為操作投擲，尤其是我們的數據是由一堆不同的陌生同學們一起累積起來而得到。有些人投擲時會加上人為的干擾(故意投向某方向、集中一把灑或是甩手散開撒等等)，這都會干擾最終實驗計算結果。不過因為我們實驗設計得當，我們發現手投和電腦投，在巨觀上來看，它們的折線圖都很相近，可以得知本次實地實驗很成功。

2.未來可進行之研究方向

在進行人工投擲和電腦模擬投擲後，我們發現嚴格算法和寬鬆算法都各有不錯的地方，但有沒有一個算式或是一個參數，可以將這兩種算法結合起來而取得更貼近圓周率的值呢？這是一個值得我們繼續探究的方向。

柒、評鑑與檢討

本章為所有作者於上述每一階段的省思與收穫，分列如下：

一、第一位作者

從不會到會的過程

這一切的開始，其實是因為數學課。那時候老師正在教圓面積和周長，提到了 π (圓周率)。老師說它是一個「無限不循環小數」，我就覺得超神奇，心裡一直想：「那以前的人到底是怎麼發現這個數字的啊？」

為了找答案，我們決定做這個研究。一開始查資料的時候其實超挫折，怎麼查都找不到我們要的東西，還好老師教了我們搜尋文獻的技巧，才讓我們順利找到資料。我們發現求 π 有很多方法，像祖沖之的割圓術或是蒙地卡羅法。經過討論，我們一致決定挑戰蒙地卡羅法，因為可以用實驗的方式來玩。

我們的實驗工具很簡單：一個大紙箱當投擲區、100 個造型橡皮擦當投擲物，還有一台電腦。實驗過程真的很有趣，當我在操場拿出大箱子開始丟橡皮擦時，好多同學都好奇圍過來看，最後大家乾脆一起輪流幫忙丟，多虧了大家的幫忙，我們的實驗數據很快就蒐集完成了。

不過，真正的挑戰在後面。我們原本完全不會用試算表軟體，是老師一步步教我們輸入公式，讓電腦幫忙算數據，然後統計圖形我們選擇使用 AI 工具來美化。但在計算時，我們遇到大麻煩——不管怎麼算，求出來的圓周率都天差地遠。那時候真的很慌覺得實驗完全失敗，但是靜下來思考後，決定從最根本的算式開始檢視。我趕快回去把蒙地卡羅法的文章再讀一遍，比對後才發現真的把公式搞錯了。幸好最後有改回來，實驗數據算出來的圓周率才順利成功。

這次研究雖然中間一度算不出答案很緊張，但能親手做實驗、學會用電腦分析，真的很有成就感。

二、第二位作者

原來算數學也可以像玩遊戲

為什麼我會選這麼數學的題目？其實理由很簡單，我覺得這個研究不只是坐在教室算圓周率，更重要的是可以和同學一起合作。比起自己一個人埋頭苦算，我更喜歡跟大家一起動手實驗的感覺。當然，我也希望能透過這次研究，讓自己的數學變得更厲害一點。

說實話，剛開始聽到蒙地卡羅法這個名字時，我根本不知道那是為了什麼？我上網查了很多資料，前幾篇文章寫得好深奧，充滿了看不懂的專有名詞，當下真的有點想放棄。但我沒有氣餒，繼續換關鍵字搜尋，終於找到幾篇專門寫給小朋友看的文章。讀完的那一刻，那種解開謎題的感覺真的很有成就感。

實驗開始了，我們準備的器材很特別，只有一個手工改造的超大紙箱、還有好多好多橡皮擦，以及每人一台電腦。記得有一次，我忘記帶筆電，另一位同學馬上大方地說：「沒關係，我們一起用。」大家擠在螢幕前討論，這種互相幫忙的團隊精神，讓我覺得很溫暖。整個實驗最有趣的部分，絕對是「丟橡皮擦」。雖然這聽起來像是在玩，但我們可是很認真的。為了公平，我們輪流上場，讓每個人都能體驗到親手操作的樂趣。丟完之後還要仔細紀錄位置，雖然過程重複，但大家說說笑笑，一點也不覺得累。

最讓我印象深刻的一幕，是我們把大紙箱搬到操場做實驗的時候。一開始只有我們自己在丟，結果旁邊經過的小朋友全都好奇地圍了過來。突然有個小朋友問：「我可以丟丟看嗎？」我們開心地說：「當然可以呀。」結果一傳十、十傳百，越來越多人加入我們的行列，大家都說：「這也太好玩了吧。」看到我們的數學實驗變成全場焦點，大家都玩得這麼開心，我也覺得超級自豪。

除了玩得很開心，我也學到了真功夫。在分析資料時，老師教我們用電腦的「試算表」來計算。一開始看著密密麻麻的格子，我完全不知道該怎麼下手。但在老師和同學的教學下，我學會了怎麼輸入公式、怎麼讓電腦幫我自動加總。原來只要學會方法，複雜的計算也可

以變得這麼簡單。這次的研究讓我發現，原來數學不只是課本上的數字，它可以是丟橡皮擦的遊戲，也可以是和朋友一起合作的回憶。我很開心能完成這個研究。

三、第三位作者

從丟橡皮擦裡找答案——我的蒙地卡羅法實驗心得

大家聽到圓周率，直覺反應應該都是那串數字 3.14159.....吧？但我心裡一直有個疑問：這數字到底是怎麼算出來的？為什麼偏偏是 3.14？

為了找答案，我上網查了好多資料。說實話，過程蠻辛苦的，網路上的資訊有的寫錯、有的太深奧，我看都看不懂。就在我看到眼睛快脫窗、想放棄休息的時候，終於找到一篇既正確又好懂的文章。那一刻真的覺得，前面的辛苦都值得了。這篇文章教了我一個很酷的方法，叫做蒙地卡羅法。

於是，我們小組擬定了計畫，準備了紙箱、造型橡皮擦和筆電，開始動手做實驗。

實驗步驟聽起來很簡單：把紙箱放地上，大家輪流把橡皮擦往箱子裡的圓形區域丟，然後計算落在圓內、圓外和線上的數量，最後把數據記下來。剛開始丟的時候，大家都覺得超好玩，但因為要蒐集很多數據，丟到後來手很痠，真的會覺得有點無聊、快膩了。不過，好險我們是團隊合作，大家輪流投擲，互相打氣，過程其實還是蠻有趣的。

這次研究讓我最有成就感的地方，就是學會用試算表。一開始我根本不會用公式，覺得算圓周率的觀念好難，但為了分析這些數據，我學會了操作試算表，讓電腦幫我把圓周率算出來。發現原本很難的計算，用對工具就變得簡單又好玩，而且這項技能以後一定也用得到，真的很實用。

回顧這次的獨立研究，雖然理解原理很燒腦、重複丟橡皮擦很累，但我不只學到了蒙地卡羅法，還學會了用電腦處理數據，更體會到和同學同心協力完成報告的快樂。這場實驗，真的是讓我一舉多得。

捌、參考資料

永無止境的圓周率追尋之旅——《數學大觀念》

<https://pansci.asia/archives/117265>

蒙特卡罗法

<https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BE%85%E6%B3%95/1225057>

蒙地卡羅模擬法是什麼？該怎麼運用蒙地卡羅模擬法？

<https://blog.roboadvisor.com.tw/?p=1551>

附件一 原始人工投擲數據

記錄 序號	在圓內	在邊緣 (圓外)	在線上	小計		記錄 序號	在圓內	在邊緣 (圓外)	在線上	小計		記錄 序號	在圓內	在邊緣 (圓外)	在線上	小計
1	24	8	3	35		24	8	0	0	8		47	13	5	2	20
2	23	14	7	44		25	7	3	0	10		48	11	7	0	18
3	24	6	3	33		26	4	2	0	6		49	12	5	3	20
4	11	6	3	20		27	5	4	0	9		50	10	7	1	18
5	20	6	2	28		28	9	7	2	18		51	13	4	3	20
6	23	3	4	30		29	14	3	3	20		52	14	5	0	19
7	37	5	3	45		30	18	2	0	20		53	13	4	3	20
8	31	9	6	46		31	16	2	0	18		54	14	5	1	20
9	29	8	7	44		32	10	4	1	15		55	13	4	2	19
10	25	13	4	42		33	8	7	2	17		56	9	6	2	17
11	25	5	4	34		34	11	2	1	14		57	7	7	4	18
12	15	7	4	26		35	13	4	0	17		58	13	4	2	19
13	29	9	9	47		36	13	5	2	20		59	11	5	3	19
14	25	13	5	43		37	11	8	1	20		60	16	2	1	19
15	29	8	6	43		38	7	4	0	11		61	8	5	5	18
16	33	11	3	47		39	15	5	0	20		62	13	4	2	19
17	5	2	0	7		40	11	7	1	19		63	9	7	4	20
18	37	8	2	47		41	12	2	3	17		64	12	4	2	18
19	5	2	0	7		42	11	4	3	18		65	12	5	2	19
20	6	3	0	9		43	13	5	2	20		66	16	2	2	20
21	5	1	0	6		44	16	3	1	20		67	14	5	1	20
22	7	1	1	9		45	12	3	1	16		68	15	4	0	19
23	7	1	0	8		46	16	4	0	20		69	12	6	2	20