

程式設計融入中小學數學探究式教學與討論活動設計之研究

胡耀仁¹、劉恩睿^{2*}

摘要

本研究目的為激發學生學習數學之動機，結合中小學數學與程式設計，透過「先理解數學原理，再以程式實作出計算機」之方式，引導學生將抽象的數學公式轉化為具體可執行的運算流程。數學中的許多觀念，在傳統教學中多以紙筆推導為主，學生往往僅停留在公式記憶層次，難以理解其運算邏輯與實際應用。本研究為開發一套以程式設計課程 C 語言實作，設計一系列以中小學數學觀念為核心的程式實作活動，包含等差數列級數、二次函數、費式數列，並引導學生完成一套對應計算機系統。透過實作歷程，提升學生在數學理解、運算思維與問題解決能力上的學習表現。依學生反饋，程式實作有助於釐清數學運算步驟，並提升其學習動機與概念理解程度。本研究可作為數學與程式跨領域教學之參考。

關鍵字：數學教育、程式設計、計算機實作、運算思維,跨領域教學

¹ 胡耀仁，臺南市私立快樂營美語數理學校老師。Email: abillcc5566@gmail.com

² 劉恩睿(通訊作者)，國立臺東大學綠能與資訊科技學系助理教授。Email: ejliu@nttu.edu.tw

A Study on Integrating Programming into Inquiry-Based Mathematics Teaching and Discussion Activities in Primary and Secondary Education

Yao-Ren Hu¹, En-Jui Liu^{2*}

This study investigates the integration of elementary and secondary mathematics education with computer programming through a programming-based calculator development approach. The instructional design requires students to first conceptualize mathematical principles and subsequently translate these principles into executable algorithms implemented in C programming language. The objective is to facilitate the transformation of abstract mathematical formulas into structured computational procedures.

Traditional mathematics instruction frequently relies on symbolic manipulation and paper-based derivations, which may limit students' understanding to formula memorization rather than procedural comprehension. To address this issue, a series of programming activities was developed based on core mathematical topics in elementary and secondary curricula, including arithmetic sequences and series, quadratic functions, the Fibonacci sequence, and time-based carry operations. Students were guided to design and implement a functional calculator system that encapsulates these mathematical models.

Through algorithm design, code implementation, testing, and debugging processes, students engaged in systematic problem decomposition and logical structuring, thereby strengthening their computational thinking skills. Empirical observations and student feedback indicate that programming-based mathematical implementation enhances procedural clarity, conceptual understanding, and learning motivation. The findings suggest that integrating programming practice into mathematics instruction provides an effective interdisciplinary framework for fostering algorithmic thinking and applied problem-solving competence.

Keywords: Mathematics Education, Programming, Calculator Implementation, Computational Thinking, Interdisciplinary Teaching

¹ Yao-Ren Hu, Tainan Private Cram School Teacher, Email: abillcc5566@gmail.com

² En-Jui Liu (Corresponding Author), Assistant Professor, Department of Green Energy and Information Technology, National Taitung University, Email: ejliu@nttu.edu.tw

壹、前言

數學與程式設計皆強調邏輯推理與問題解決能力，兩者在學習本質上具有高度關聯性。然而，在實際教學中，數學課程多以公式推導與紙筆計算為主，而程式設計課程則偏重語法與結構練習，學生往往難以將兩者加以連結，導致學習成效有限。

隨著資訊科技的普及，程式設計逐漸成為重要的基礎能力，若能將程式設計作為數學學習的工具，協助學生將抽象數學概念轉化為具體運算流程，將有助於深化數學理解。因此，本研究嘗試以數學計算機之程式實作為教學核心，讓學生在理解數學概念後，透過程式設計實際完成計算功能，藉此提升學習成效。

貳、文獻探討

一、數學概念學習與迷思概念形成

相關研究指出，學生在學習數學概念時，若僅透過符號操作與公式記憶進行學習，容易形成迷思概念，特別是在比例、函數與統計等主題中更為明顯（Dewey, 1938；Goodman et al., 2006）。研究亦顯示，透過具體操作、反思與多元表徵方式進行教學，有助於學生修正錯誤概念並建立正確的數學理解（Hearns et al., 2010）。

二、程式設計與運算思維

程式設計強調問題拆解、流程規劃與邏輯判斷，其學習歷程與數學解題所需之邏輯推理高度相似。學者指出，透過程式設計活動，學生能將抽象問題轉換為可執行的演算法，進而培養運算思維與系統性思考能力（Papert, 1980；Wing, 2006）。此外，多階段程式實作與反覆練習的學習方式，亦能提升學生對程式與數學概念的整體理解（Rother et al., 2010）。

三、探究式教學在跨領域學習中的應用

探究式教學強調學生主動提出問題、進行實驗驗證與反思的學習歷程，有助於深化概念理解與提升學習動機（Dewey, 1959）。相關研究指出，結合動手做與探究活動的教學設計，能在學習成效與學習動機上優於傳統講述式教學（Gerstner & Bogner, 2010）。近年來，STEM 教育與跨領域課程的推動，更凸顯將工程設計與程式實作融入中小學數學與科學課程之重要性（Hynes et al., 2011；Breiner et al., 2012）。

四、等差數列級數

數列是由數字組成的序列，若數列中每項與其前一項之差值相等（該差值稱為公差），該數列即為等差數列，例如數列：7, 10, 13, 16, 19, 22, ...即為一個等差數列，在這個數列中，自第二項起，每項與其前一項之差值皆為 3，公差即為 3；如果一個等差數列的首項記作 a_1 ，公差記作 d ，而因首項與第 n 項之間有 $n-1$ 個間隔，每一間隔大小為公差 d ，故該等差數列第 n 項 a_n 即為： $a_n = a_1 + (n-1) \cdot d$ 。等差數列中首項至第 n 項加總即為等差級數 S_n ，因等差數列中 a_i 與 a_{n-i+1} 兩兩一組（所以共 $n/2$ 組）之值皆相同且等於 $a_1 + a_n$ ，故 $S_n = (a_1 + a_n) \cdot n/2$ 或 $[2a_1 + (n-1) \cdot d] \cdot n/2$ 。

五、二次函數

二次函數為多項式函數的一種，其一般形式為：

$$f(x) = ax^2 + bx + c$$

其中 a 、 b 、 c 為常數，且 $a \neq 0$ 。二次函數的圖形為拋物線，其開口方向取決於係數 a 的正負：當 $a > 0$ 時拋物線開口向上；當 $a < 0$ 時拋物線開口向下。拋物線的頂點為函數的極值點，其座標為：

$$\left(-\frac{b}{2a}, \frac{4ac - b^2}{4a}\right)$$

在函數分析中，常需求解 $f(x) = 0$ 的情形，即求二次方程式

$$ax^2 + bx + c = 0$$

之解。其解可由求根公式表示為：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

其中 $b^2 - 4ac$ 稱為判別式 (discriminant)，通常以 Δ 表示。判別式可用以判斷方程式根的性質：當 $\Delta > 0$ 時，方程式具有兩個相異實根；當 $\Delta = 0$ 時，具有一個重根；當 $\Delta < 0$ 時，則無實數解。二次函數在數學與科學領域中具有重要地位，常應用於拋體運動分析、最佳化問題與面積計算等問題。

六、費式數列

費式數列是一個前二項分別為 0 和 1 而後續每個數字皆等於前兩個數字加總的數列，故前幾項為 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...。費式數列之重要性質之一為相鄰兩項之比值隨 n 增加而趨近黃金比例：

$$\phi = \frac{1 + \sqrt{5}}{2} \approx 1.618$$

，此數列廣泛應用於大自然生長規律、計算機科學與金融分析中。

七、時間進位計算

時間進位計算為小學常見教學內容，教學重點於秒、分、時與日之間的轉換，其關係為 60 秒為 1 分、60 分為 1 時、24 時為 1 日。此類運算不同於十進位系統，屬於多層級進位結構。

在進行時間計算時，當數值超過進位基準需進行進位處理，例如秒轉分、分轉時等，通常透過整數除法與餘數概念完成。由於進位基底不一致，學生在學習過程中容易產生混淆，特別是在多單位同時計算時更為明顯。因此，理解時間進位計算有助於學生掌握不同進位制之運算規則，並強化其對數值轉換與運算邏輯之理解。

參、研究方法

本研究採用探究式教學與實作分析相結合之方式進行，透過將數學學習與程式實作整合於教學過程中，引導學生從理解數學原理出發，進一步將抽象的數學公式轉化為具體且可執行的運算流程。研究重點在於以程式實作目前中小學數學課程作為數學與程式跨領域教學之參考，以此提升學生在數學概念理解、邏輯推理與問題解決能力等方面的學習表現。整體教學流程分為三個階段：

一、數學概念與程式語法講解：

在教學初期，教師先針對本研究所涉及之數學概念進行說明與推導，使學生能理解相關數學公式的形成過程與其實際意義。教學內容包含數列級數、二次函數、費式數列以及時間進位計算等主題。透過課堂講解、範例演示與問題討論等方式，引導學生理解數學概念之間的關聯性與運算邏輯，使學生不僅能記憶公式，更能理解其背後的數學原理。

此外，考量部分學生可能為首次接觸程式設計，本研究亦於教學初期安排基礎程式語法的介紹，包括變數宣告、資料型態、輸入與輸出指令、條件判斷結構以及迴圈控制等基本概念。透過簡單範例示範，使學生逐步熟悉程式語言的基本架構與撰寫方式，為後續將數學公式轉換為程式運算流程奠定基礎。

二、程式實作階段：

在學生理解相關數學概念與基本程式語法後，進入程式實作階段。此階段主要引導學生將數學公式與運算過程轉換為程式邏輯，並逐步撰寫對應之程式碼。學生需依據不同數學問題的特性，設計適當的運算步驟與程式流程，包括資料輸入、數值運算以及結果輸出等功能。

在實作過程中，學生透過實際撰寫程式並執行運算，可進一步觀察數學公式在不同數值條件下的計算結果，進而理解其運算規律。例如在數列級數計算中，學生可透過程式觀察數列項數增加時總和的變化；在費式數列的計算過程中，則可理解遞迴或迴圈運算所呈現的數列生成規律。透過此種方式，使學生能將抽象的數學概念轉化為具體的運算過程，進一步加深對數學概念的理解。

三、測試與修正：

在完成程式撰寫後，學生需進程式測試與結果驗證。透過輸入不同數值進行多次測試，比較程式輸出的計算結果與人工計算結果是否一致，以確認程式運算的正確性。若測試過程中發現計算結果與預期不符，則需回溯程式流程進行除錯與修正，包括檢查公式轉換是否正確、運算順序是否合理，以及程式邏輯是否存在錯誤。

透過反覆測試與修正的過程，學生能逐步建立完整的問題分析與解決能力，同時強化邏輯思考與運算理解能力。此外，透過程式與數學計算結果的對照，學生亦能更清楚理解數學公式在實際運算中的應用方式，進一步提升其對數學概念的掌握程度。

肆、結果與討論

透過本研究所開發的程式實作，可以清楚地觀察到中小學數學課程中核心概念的呈現與學生互動學習的效果。四個程式分別針對等差數列級數、二次函數、費式數列以及時間進位計算，呈現不同的數學概念與邏輯運算過程。

一、等差數列與級數程式

本程式透過輸入首項、公差及項數，即可計算出第 n 項及總和，並提供公式與迴圈兩種方法進行驗證。依學生反饋與觀察發現，學生在操作程式後，對等差數列的項數、總和計算公式，以及迴圈累加概念理解明顯提升；程式的即時運算結果也能幫助學生快速檢驗手算的正確性，進而強化對數列規律與公式應用的理解。

(a)

```
[*] 等差數列級數.cpp
1  #include<stdio.h>
2  int main(){
3      int d,n,a1,i;
4      printf("請輸入公差、項數、首項\n");
5      scanf("%d %d %d",&d,&n,&a1);
6      int a,s1,s2;
7      a=a1+(n-1)*d;
8      s1=(a+a1)*n/2;
9      printf("第%d項為%d,級數總和為%d\n",n,a,s1);
10     s2=0;
11     for(i=1;i<=n;i++){
12         s2+=a1+(i-1)*d;
13     }
14     printf("若以迴圈計算總和為%d",s2);
15 }
```

(b)

求等差數列 1, 3, 5, 7, 9 的和。

• 解析： $a_1 = 1, a_n = 9, n = 5$ 。

• 計算： $S_5 = \frac{5(1+9)}{2} = \frac{5 \times 10}{2} = 25$ 。

```
請輸入公差、項數、首項
2 5 1
第5項為9,級數總和為25
若以迴圈計算總和為25
-----
Process exited after 22.57 seconds with return value 0
請按任意鍵繼續 . . . |
```

圖 1. 等差數列與級數程式碼與結果

二、二次函數程式

透過程式輸入係數 a, b, c 即可計算判別式、求解實根或重根，並自動判斷極值點及對應函數值。這種動態操作使學生能直觀理解判別式與根的關係，以及極值概念。討論結果顯示，透過程式呈現的極值與根的關係圖示，能有效提升學生對二次函數圖形特性、對稱性及應用情境的理解，並促進邏輯推理能力。

(a)

```
[*] 二次函數.cpp
1  #include <stdio.h>
2  #include <math.h>
3  int main(){
4      float a,b,c,D,x1,x2,x,y;
5      printf("請輸入二次函數a,b,c值分別為多少\n");
6      scanf("%f %f %f",&a,&b,&c);
7      D=b*b-4*a*c;
8      if(D>0){
9          printf("f(x)=0 時有兩個相異實數解\n");
10         x1 = (-b + sqrt(D)) / (2*a);
11         x2 = (-b - sqrt(D)) / (2*a);
12         printf("x1 = %f\n",x1);
13         printf("x2 = %f\n",x2);
14     }
15     else if(D == 0){
16         printf("f(x)=0 時有一個重根\n");
17         x1 = -b/(2*a);
18         printf("x = %f\n",x1);
19     }
20     else{
21         printf("f(x)=0 時沒有實數解\n");
22     }
23     x=-b/(2*a);
24     y=a*x*x+b*x+c;
25     printf("當x=%f時,f(x)有極值%f\n",x,y);
26     return 0;
27 }
```

(b)

1. 兩交點情況 ($D > 0$)

- 函數： $f(x) = 2x^2 - 5x + 2$
- 判別式： $D = (-5)^2 - 4(2)(2) = 25 - 16 = 9 > 0$
- 解(根)： $(2x - 1)(x - 2) = 0 \Rightarrow x = \frac{1}{2}, 2$
- 極值計算：頂點 $x = \frac{-b}{2a} = \frac{5}{4}$ ，代入得 最小值 $-\frac{9}{8}$ (即 $-\frac{D}{4a}$)

```
請輸入二次函數a,b,c值分別為多少
2 -5 2
f(x)=0 時有兩個相異實數解
x1 = 2.000000
x2 = 0.500000
當x=1.250000時,f(x)有極值-1.125000

-----
Process exited after 16.3 seconds with return value 0
請按任意鍵繼續 . . . |
```

(c)

2. 相切情況 ($D = 0$)

- 函數： $f(x) = 9x^2 + 6x + 1$
- 判別式： $D = (6)^2 - 4(9)(1) = 36 - 36 = 0$
- 解(根)： $(3x + 1)^2 = 0 \Rightarrow x = -\frac{1}{3}$ (重根)
- 極值計算：頂點 $x = \frac{-6}{18} = -\frac{1}{3}$ ，代入得 最小值 0 (即 $-\frac{D}{4a}$)

```
請輸入二次函數a,b,c值分別為多少
9 6 1
f(x)=0 時有一個重根
x = -0.333333
當x=-0.333333時,f(x)有極值0.000000

-----
Process exited after 5.793 seconds with return value 0
請按任意鍵繼續 . . . |
```

(d)

3. 無交點情況 ($D < 0$)

- 函數： $f(x) = -3x^2 + 2x - 1$
- 判別式： $D = (2)^2 - 4(-3)(-1) = 4 - 12 = -8 < 0$
- 解(根)：在實數範圍內 無實數解
- 極值計算：頂點 $x = \frac{-2}{-6} = \frac{1}{3}$ ，代入得 最大值 $-\frac{2}{3}$ (即 $-\frac{D}{4a}$)

```
請輸入二次函數a,b,c值分別為多少
-3 2 -1
f(x)=0 時沒有實數解
當x=0.333333時,f(x)有極值-0.666667

-----
Process exited after 12.87 seconds with return value 0
請按任意鍵繼續 . . . |
```

圖 2. 二次函數程式碼與結果

三、費式數列程式

本程式使用遞迴方法計算費式數列前 n 項，並顯示最後兩項的比值。結果顯示，學生在實作中能觀察到數列遞推的模式，以及費式數列相鄰兩項之比值確實將隨 n 增加而趨近黃金比例 1.618。透過程式運算，學生能加深對遞迴概念、序列規律與數學模式認識的掌握，並培養解決問題的邏輯思維。

(a)

```
["] 費式數列.cpp
1 #include <stdio.h>
2 /*遞迴函式*/
3 int fibonacci(int n){
4     if(n<=0) return 0; /*第0項為0 */
5     if(n==1) return 1; /*第1項為1 */
6     return fibonacci(n-1)+fibonacci(n-2);
7 }
8
9 int main(){
10     int n,i;
11     float q;
12     printf("請輸入欲計算Fibonacci 數列前幾項的值?\n");
13     scanf("%d",&n);
14     for(i=0;i<n;i++){
15         printf(" %d",fibonacci(i));
16         if(i==n-1){
17             q=(float)fibonacci(i)/fibonacci(i-1);
18             printf("\n 最後相鄰兩項之比值為%.3f",q);
19         }
20     }
21     return 0;
22 }
```

(b)

```
請輸入欲計算Fibonacci 數列前幾項的值?
10
0 1 1 2 3 5 8 13 21 34
最後相鄰兩項之比值為1.619

-----
Process exited after 2.04 seconds with return value 0
請按任意鍵繼續 . . . |
```

圖 3. 費式數列程式碼與結果

四、時間進位計算程式

本程式透過輸入日、時、分、秒等數值並進行倍數運算，程式能自動完成跨單位之進位處理，將結果轉換為正確之時間表示形式。由測試結果可發現，學生能正確處理秒轉分、分轉時及時轉日之進位關係，並理解不同單位間之換算規則。此過程有助於學生掌握多層級進位系統之運算邏輯，並降低在手動計算時可能產生的錯誤。

(a)

```
[*] 時間進位計算機.cpp
1  #include<stdio.h>
2  int main(){
3      int d,h,m,s,x;
4      printf("請輸入幾日幾時幾分幾秒\n");
5      scanf("%d %d %d %d",&d,&h,&m,&s);
6      printf("請輸入要乘多少\n");
7      scanf("%d",&x);
8      d=d*x;
9      h=h*x;
10     m=m*x;
11     s=s*x;
12     m=m+s/60;
13     h=h+m/60;
14     d=d+h/24;
15     s=s%60;
16     m=m%60;
17     h=h%24;
18     printf("%d日%d時%d分%d秒",d,h,m,s);
19 }
```

(b)

題目

「小強參加環島馬拉松接力賽，他跑完一棒平均需要 7 小時 45 分 50 秒。如果他一共要跑 4 棒，總共花了多少時間？」

解題步驟（直式思考）：

1. 先分別乘出來：

1. 秒： $50 \times 4 = 200$ 秒2. 分： $45 \times 4 = 180$ 分3. 時： $7 \times 4 = 28$ 時

2. 由小單位開始進位：

1. 秒進分： $200 \div 60 = 3$ 餘 20 → 留 20 秒，進 3 分。2. 分進時： $(180 + 3) \div 60 = 3$ 餘 3 → 留 3 分，進 3 時。3. 時進日： $(28 + 3) \div 24 = 1$ 餘 7 → 留 7 時，進 1 日。

答案：

1 日 7 小時 3 分 20 秒

請輸入幾日幾時幾分幾秒

0 7 45 50

請輸入要乘多少

4

1日 7時 3分 20秒

Process exited after 16.96 seconds with return value 0

請按任意鍵繼續 . . . |

圖 4. 時間進位計算程式碼與結果

總結來說，本研究實作活動成功結合了程式與中小學數學課程內容可作為數學與程式跨領域教學之參考，研究過程依學生反饋與觀察可發現此跨領域教學可加深學生對抽象數學概念的理解，並強化邏輯推理與問題解決能力。在數學學習過程中，學生常因計算或觀念錯誤而答錯題目。將錯題重新計算，並適度改變題目數字，有助於學生重新檢視解題步驟與概念，找出錯誤原因，進而加深對數學概念的理解。此外，題目數字改變後，學生無法僅依賴記憶答案，必須再次運用公式與邏輯進行解題，有助於提升問題解決能力與概念應用能力，但教師若需手動計算每一題修改後的答案，往往較為耗時。未來可進一步透過此程式輔助快速得到正確答案，不僅提升教學效率，也能提供更多不同數值的練習題目。綜合而言，錯題再計算結合程式輔助，能同時提升學生學習成效與教師教學效率，對數學學習具有正面幫助。

伍、結論

本研究成功將程式實作與中小學數學教學結合，教學活動每周 1 至 3 小時持續一年以上，依學生反饋與觀察證實此跨領域教學方式有助於學生理解數學運算流程，並培養運算思維與問題解決能力。由實作結果顯示，程式設計不僅是資訊科技學習工具，亦可結合傳統學科成為深化學習的重要媒介。未來可進一步以此實做進行錯題修改，利用程式計算快速得到正確答案，以提升教學成效與實用性。

參考文獻

- Breiner, J. M., Harkness, S. S., Johnson, C. C., & Koehler, C. M. (2012). What is STEM? A discussion about conceptions of STEM in education and partnerships. *School Science and Mathematics*, 112(1), 3–11.
- Dewey, J. (1938). *Experience and education*. Macmillan.
- Dewey, J. (1959). *Democracy and education*. Macmillan.
- Gerstner, S., & Bogner, F. X. (2010). Cognitive achievement and motivation in hands-on and teacher-centered science classes: Does an additional hands-on consolidation phase (concept mapping) optimize cognitive learning at work stations? *International Journal of Science Education*, 32(7), 849–870.
- Goodman, B. E., Freeburg, E. M., Rasmussen, K., & Di, M. (2006). Elementary education majors experience hands-on learning in introductory biology. *Advances in Physiology Education*, 30(4), 195–203.
- Hearns, M. K., Miller, B. K., & Nelson, D. L. (2010). Hands-on learning versus learning by demonstration at three recall points in university students. *OTJR: Occupation, Participation and Health*, 30(4), 169–171.
- Hynes, M., Portsmore, M., Dare, E., Milto, E., Rogers, C., Hammer, D., & Carberry, A. (2011). *Infusing engineering design into high school STEM courses*. National Center for Engineering and Technology Education.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Rother, K., Rother, M., Pleus, A., & Belzen, A. U. Z. (2010). Multi-stage learning aids applied to hands-on software training. *Briefings in Bioinformatics*, 11(6), 582–586.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.