

建模引導之科學程式設計教學

李敏瑄 蕭聖哲 陳思如 林育慈*

國立臺灣師範大學 資訊教育研究所

摘要

本研究設計建模引導之科學程式設計教學，引導學生透過「現象描述」、「資料建模」、「流程建模」、「程式化」以及「觀察與除錯」等建模歷程逐步設計解題策略並編寫程式碼以解決科學問題。研究以準實驗研究法驗證所發展的建模引導教學模式之成效，教學實驗於大學通識課程實施，根據實驗結果發現：一、建模引導教學能幫助學生將複雜的科學現象擷取出關鍵解題元素、將解題相關的變數以及變數間的關係做較精確的描述，並將解題的步驟與流程較邏輯地陳述出來，因此整體來說，能提升學生「資料建模」與「流程建模」之品質，此亦分別對應運算思維的「資料表示」與「演算法設計」；對於低先備能力的學生來說，能同時提升「現象描述」、「資料建模」、「流程建模」之建模品質；二、建模引導教學能提升低先備能力學生之建模品質，藉以逐步發展解題策略，更精確表達資料並設計演算法，進而能撰寫程式以解決科學問題，因此能增進科學問題解決能力；三、由於建模輔助教學能逐步引導學生在複雜的科學問題情境中找到解題的脈絡，因此實驗組學生能感受「現象描述」與「流程建模」對科學問題解決的助益，亦較能清楚說明科學程式設計的優點，且能肯定程式設計在科學問題解決、科學理解與探索中的角色。

關鍵詞：STEM科際整合教學、建模、科學運算、程式設計教學、運算思維

壹、研究動機

資訊科技已然與生活及職業密不可分，各先進國家皆紛紛將運算思維(computational thinking)視為國民重要的基本能力(Barr & Stephenson, 2011; Magana et al., 2011)。運算思維是利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的思維模式(Wing, 2006)，包含抽象化、演算法設

計、資料表示、流程控制、拆解問題、建模(modelling)等概念，可以幫助我們透過資訊科技這個運算工具來解決各個領域的問題(Selby & Woollard, 2013)。科學領域經常需要透過運算處理大量而複雜的計算問題，或透過運算進行科學的探索與預測等，因此運算思維成為現今科學教育的重點之一，可協助培養學生跨學科問題解決、量化和批判性思維的能力(Qin, 2009)。為了提升運算思維，學習程式

*通訊作者：林育慈，liny@ntnu.edu.tw；ORCID：0000-0002-2138-538X

投稿：2023/6/1，修訂：2023/9/14，接受：2023/9/18，線上出版：2024/1/29

設計是有效的方法(García-Peñalvo & Mendes, 2018; Magana et al.)，因此運算思維和程式設計在各領域的教育上變得越來越重要。

程式設計除了理論與概念之外，尚須同時具備程式設計實作技能與策略運用能力(Bayman & Mayer, 1988; Davies, 1993; Soloway, 1986)。然而，目前程式設計教學多著重程式設計理論與概念，例如：程式語法、程式結構、程式設計慣例或演算法設計概念等(Davies; Kordaki et al., 2008; McKeithen et al., 1981; Pennington, 1987)，較缺乏培養學生程式設計應用與實作能力及透過運算思維解決問題的教學，所進行的程式練習規模亦較小，缺乏讓學習者將所習得的程式設計概念與理論應用於大型複雜問題的機會，導致學習者空有概念而無法將其有效運用於程式設計，在面對實體世界的複雜問題時，並無足夠能力解決問題(Davies)。因此，程式設計教學除了語法、結構等概念知識之外，亦須提供學生以程式設計解決複雜問題的機會，方能培養學生應用與實作程式之能力(Bayman & Mayer; Soloway)。然而，真實世界的問題多同時涉及多種領域知識，因此許多學者提出須整合不同的領域進行教學，以提供學生解決實際情境中複雜問題的機會(Connor et al., 2013; Matarić et al., 2007)，STEM科際整合教學隨之應運而生。STEM是指科學、科技、工程與數學(Science, Technology, Engineering, and Mathematics)四大領域，STEM素養是指具備將四個領域整合運用於問題解決的能力，透過STEM能力的培養，能促進學生創新與批判性思維等能力，以面對職涯中複雜的挑戰(Maltese et al., 2014; National Research Council, 2011; Sadler et al., 2012)。許多學者認為STEM領域的教學須為四個領域的跨科際(interdisciplinary)教學，以提供學生解決實際

情境中複雜問題的機會(Connor et al.; Matarić et al.)。

不僅程式設計與運算思維的教學可透過STEM科際整合的方式進行，從STEM領域的教學來看，亦無法與運算分離(Grover & Pea, 2013; Hambrusch et al., 2009)。現有研究發展許多運算融入科學與工程領域的教學，例如：進行天文研究時，天文望遠鏡所取得的大數據需要進行分析，或是氣象模型或工程材料的大量模擬等，因此在進行科學與工程教學時，亦應導入運算思維。Hambrusch等為科學科系的學生設計一套運算思維的課程，利用程式設計工具(Python的變數、型態、數學運算、字串、陣列等；利用VPython製作圖形、函數、參數等)以及運算工具與方法(利用NumPy處理數字、模擬方法、物理模擬軟體、生物資訊工具等)，並納入電腦科學議題來進行教學，其中包含語音處理、網格滲透、模擬物理系統與蛋白質交互作用等專題。透過運算的模擬與建模，可以讓STEM的問題解決更有效率。

建模是運算思維的重要元素，透過建模，可將問題描述為運算能有效解決的形式。在學習程式設計方面，建模有助於學生理解概念之間的關係(Abdullah & Shariff, 2008)，使學生更能掌握解決程式問題的關鍵(Teodoro & Neves, 2011)。在科學教育中，建模亦扮演相當重要的角色(Gilbert, 2004; Gilbert et al., 2000; Greca & Moreira, 2002)，建模過程可重現科學家做研究時的認知歷程與變化(Gilbert)，亦可提供學生思考和實踐科學概念的機會(Brewe & Sawtelle, 2018; Sins et al., 2005)，進而幫助學生建立科學概念(Brewe, 2008; Brewe & Sawtelle; Shen et al., 2014; Svoboda & Passmore, 2013)。

儘管運算思維與程式設計日益重要，

但一般程式設計教學仍較少提供學生真實而複雜的問題解決情境。為使學生能透過複雜的真實世界問題情境學習程式設計，本研究透過STEM科際整合的方式，讓學生利用程式設計解決科學問題。然而，初學者面對複雜的真實問題，往往難以掌握解題策略，而傳統由教師示範解題程序的教學，較缺乏有效的解題思考引導，難以幫助學生發展解題策略。為了幫助學生將複雜問題的解題歷程表徵成有效的形式以透過程式設計解題，本研究並設計與發展一套建模引導教學策略與輔助學習平臺，以引導學生逐步發展解題策略並撰寫程式碼，進而解決科學問題。本研究聚焦於科學程式設計，因此論文中之「科學問題解決能力」，乃指「透過程式設計解決科學問題之能力」，而「建模」則指「欲以程式設計解決科學問題時，用以表徵解題歷程的方法」。研究將探討建模引導之科學程式設計教學是否能提升學生的建模品質、科學問題解決能力與學習態度。待答問題如下：

- 一、建模引導之科學程式設計教學，是否能提升學生的建模品質？
- 二、建模引導之科學程式設計教學，是否能提升學生的科學問題解決能力？
- 三、建模引導之科學程式設計教學，是否能提升學生的學習態度？

貳、文獻探討

一、運算思維與程式設計

運算思維是利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的思維模式(Wing, 2006)，是一種解決問題的能力與思考方式，因此涉及制定問題與解決問題的思維過程(Grover & Pea, 2013)。Rodríguez

del Rey等(2021)認為運算思維應該包含了抽象化、演算法設計、資料表示和拆解問題。自運算思維一詞被提出開始，許多學者提出運算思維應包含哪些元素，以及如何應用於各領域之學習，多數研究皆認為運算思維應包含抽象化、問題解析、建模、模擬、演算法思維、資料表示、重複、遞迴與平行化思考(Barr & Stephenson, 2011; Grover & Pea; Selby & Woollard, 2013)。

運算思維是一種分析性思維，它能夠幫助我們利用數學和工程思維來理解和解決現實世界中的複雜問題(Voskoglou & Buckley, 2012)，若想要利用資訊科技的方法來解決複雜問題，運算思維是不可或缺的能力(Sanford & Madill, 2007; Wing, 2006)。而藉著程式設計，可發揮電腦強大的運算能力，幫助我們解決複雜的科學問題、數學運算和模擬科學現象，因此運算思維與程式設計的能力在STEM領域中相當重要(Chonacky & Winch, 2008)。

然而，運算思維並不同於程式設計，但學習程式設計是有效提升運算思維的方法(García-Peñalvo & Mendes, 2018; Magana et al., 2011)，寫程式的過程便是培養運算思維的歷程(Román-González et al., 2017)。儘管目前各學者或組織對於運算思維並沒有確切的定義，但有學者認為學生需要利用程式或科技工具來創作才算是具有運算思維(Lye & Koh, 2014)。從另一方面來看，有學者認為我們在寫程式前會先分析問題，確立解題目標和重要元素、思考資料該如何表示以及解題流程為何等，因此程式設計最基本的能力就是運算思維(Román-González et al.)。當學生在利用程式解決問題時他們就能以運算思維的角度看待問題。除此之外，用程式來解決問題時涉及多種能力，包含：溝通和合作力、

創造力、批判性思維、問題辨識與解決的能力(Durak et al., 2019; Voskoglou & Buckley, 2012)。程式設計教學時若可聚焦運算思維，可讓學生回歸到概念本質，有助於學生對於問題有更深刻理解(Voskoglou & Buckley)，並幫助學生以更多元的方式來解決問題(Barr & Stephenson, 2011)。

傳統程式設計教學較少將程式設計的思維歷程與系統化程序呈現給學生(Lin et al., 2022)，因此較難培養學生發展自己的解題策略以具備運算思維。許多研究嘗試教導學生問題解決策略，認為給予適當的鷹架(scaffolding)或是範本(template)可以提升學生程式設計學習的表現(Xie et al., 2019)。例如：透過學習單引導解題方向或程序(Angeli, 2022; Angeli & Valanides, 2020; Huang et al., 2012)，或在解題時提供不同的程式樣版進行填空(Weinman et al., 2021)等。但此等研究多聚焦於引導學生解決特定問題，未能針對發展學生通用的問題解決策略提供有效的教學引導。再者，儘管科學運算日益重要，科學領域專家卻仍較常使用數值運算之程式語言(如MATLAB、FORTRAN等)(Prabhu et al., 2011)，且相關教學研究較著重模擬軟體的操作(Psycharis, 2013)，較少探討如何引導學生透過程式設計解決科學問題。

二、STEM教學

STEM是指科學、科技、工程與數學四個領域。傳統教學的課程間關聯性較低，使得學習者學習動機薄弱，且無法將習得的知識連結到生活中(Bybee, 2013; Gonzalez & Kuenzi, 2012)，而現實生活中所面臨的問題經常涉及多種知識，因此許多學者認為STEM的人才也必須具備能融會貫通這四大領域知識的複合型能力，STEM教學應能將四個分散

的領域進行科際整合(Morrison, 2006; Sanders, 2009)，希望透過跨域的整合與應用，讓學習者具備較高層次和統整性的思考去解決真實世界的問題(Breiner et al., 2012; Bybee, 2010; Xie et al., 2015)。

許多學者都提出多元化的課程設計，能引發學生對STEM的興趣，增加其對STEM的職涯傾向，且針對不同年段學生的特質，採用不同的設計(Maltese et al., 2014; Xie & Reider, 2014)。科技工具的使用、動手實作、與合作式學習是許多研究所使用的STEM教學方式，例如：透過可程式機器人的學習，學會程式設計、無線通訊、與機器人的三維動態模擬(Matarić et al., 2007)；學生利用麵包版等設備進行實驗，並撰寫MATLAB程式進行模擬，如此同時學到電子學、程式設計、模擬等知識(Connor et al., 2013)。許多STEM教學透過跨域的方式討論工程與科學議題，例如：透過討論製作生質柴油的研究及教育，使學生瞭解生物柴油燃料的永續發展(Burrows et al., 2014)；融入海洋與環境保護議題，讓學生學習環境議題、電機、齒輪、電流與電壓等(Schnittka & Bell, 2011)。

電腦科學本身就是同時具備STEM四個領域特質的學科(Reed, 2019)，很適合作為STEM學科間的橋梁(English, 2017)。許多STEM教學將運算思維融入課程，透過建模探索工程與科學問題，根據系統的結構與需求以適切形式描述概念之間的關係，並設計與建立技術過程的模型，以期能精確地分析歷程，並有效檢視複雜的過程與除錯(Schmidt, 2006)。將運算思維加入STEM中，能採用基於運算思維的策略，並適用於各種情境，且能促進善用運算的思考，因此運算思維能有效幫助STEM領域的問題解決(Swaid, 2015)。

三、科學問題解決

傳統科學教學方式大多是將複雜的科學現象整理為一些規則、數學公式，老師再加以做說明，而學生以死記硬背的方式學習。因此學習者的科學概念是由分散和獨立的規則以及公式所組成的(Gilbert, 2004; Jimoyiannis & Komis, 2001; Marlino, 2001; Neves et al., 2013)，如此使學習者無法掌握推論科學概念的方法(Neves et al.)，甚至無法正確解釋公式或規則(Jimoyiannis & Komis)。因此，當學習者遇到複雜的科學問題時，常無法正確應用所學的科學概念解決問題(K.-E. Chang et al., 2008; Jimoyiannis & Komis)。科學問題解決涉及規則知識(rule knowledge)、規則應用(rule application)和規則辨別(rule identification)三個技能(Bühner et al., 2008; Wüstenberg et al., 2012)。學生亦須能辨識變數以及變數間的關係，以確定解題所需要使用到的科學概念(Scherer & Tiemann, 2012)。為了培養學生科學問題解決能力，有些科學教學採用問題導向式教學，透過真實問題情境幫助學生經由探究問題更深入瞭解概念(Akçay, 2009)。為了幫助學生發展解決問題的策略，教師須引導學生思考問題，例如：如何定義問題、如何定義問題的重要資訊、如何分析與整合問題的重要資訊以及發想解決問題的可能方案(Delisle, 1997)，並讓學生體會理解以及應用相關知識的重要性(Ferreira & Trudel, 2012)，以便他們能在現實世界中應用科學知識(Foshay & Kirkley, 1998; Lim, 2000)。

在資訊時代中，運算思維已成為探究各種領域知識之基本工具(Chi & Jain, 2011)，因此科學研究與問題解決亦已無法與運算分離(Hambrusch et al., 2009)，例如：進行天文研究時，天文望遠鏡所取得的大數據需要進

行分析，或是氣象模型或工程材料的大量模擬等，因此在進行科學與工程教學時，亦應導入運算思維。Hambrusch等為科學科系的學生設計一套運算思維的課程，利用程式設計工具(Python的變數、型態、數學運算、字串、陣列等；利用VPython製作圖形、函數、參數等)以及運算工具與方法(利用NumPy處理數字、模擬方法、物理模擬軟體、生物資訊工具)，並納入電腦科學議題來進行教學，其中包含語音處理、網格滲透、模擬物理系統與蛋白質交互作用四個專題。透過編寫程式有助於提高學生的科學問題解決能力，因為透過程式模擬科學現象學生可以觀察到各個物件之間的關係並及時的得到回饋，因此可以讓學生更專注於科學理論而不只是特定的情況(Chabay & Sherwood, 2008; Taub et al., 2015)。此外，透過科學建模歷程能使學生對於科學現象形成適當的表徵，用以表達個人的想法、進行溝通，並能有效解決科學問題(Justi & Gilbert, 2002)。

四、建模教學

建模是運算思維的元素之一，能幫助我們用更精確的方式描述系統，例如：描述系統中各元素的互動、描述系統中的時間與空間關係、設定系統中所須的參數等(Wing, 2008)。在科學教育中，建模亦扮演相當重要的角色(Gilbert, 2004; Gilbert et al., 2000; Greca & Moreira, 2002)，就如同真實現象和科學理論的銜接者(Gilbert)。科學模型是一個自然現象的簡易表徵，而建立此一表徵的過程就是科學建模，它重現科學家進行研究時認知的變化與過程(Gilbert; Gilbert et al.)，並嘗試將現實世界中的物體或現象轉化為數學公式或電腦關係，也稱之為數學模型或模擬模型(Voskoglou & Buckley, 2012)。不同的研究者提出了不同的建模歷程，但無論是分析、

歸納式推論、量化、解釋、評估(Sins et al., 2005)，它們都符合質性模型到量化模型，再到模型測試和改進的過程(Brewe, 2008)。經由科學建模歷程能讓學習者對於科學現象形成適當的表徵，透過產生、測試或修改模型等，更精確表達個人對科學現象的想法、進行溝通以及建構與學習科學概念並解決科學問題(Brewe; Justi & Gilbert, 2002; Sins et al.; Zawojewski, 2013)。

傳統科學教學常忽略培養學生透過科學家的思考方式探索科學的能力，導致學生無法真實掌握科學概念(Crawford, 2015)。透過科學建模的教學方式，能連結學習者的先備知識，在建模過程中發現尚未理解的科學概念(Sins et al., 2005)，亦可讓科學知識和實驗相互結合，以幫助學生重新組織他們的科學概念(Hutchins et al., 2020)。運算模型是指將科學現象先以數學方式表示為模型，再用程式實現或模擬出來(Psycharis, 2013)，程式設計強大的運算能力可以幫助我們解決複雜的科學問題、數學運算和模擬科學現象，所以程式設計的能力在科學領域也變得相當重要(Chonacky & Winch, 2008)。若在科學課程中加入運算建模的概念，學生能利用教學者提供的模型對各種情況或問題進行描述、解釋或預測問題(Hestenes, 1987)。藉由建構科學模型的過程，可以幫助學生理解科學公式並降低了這些概念的學習門檻，進而強化學生的學習動機，促進批判性思考的能力，以達到學習成效(Redish & Wilson, 1993)。

Stacewicz與Włodarczyk (2010)提出建模的歷程包含抽象化、形式化、簡化與查驗，而Sins等(2005)提出的建模程序則為：分析、歸納推論、量化與解說。進行建模時，可以反覆的修正和驗證，直到模型和科學理

論相符為止，透過這樣的方式能刺激學生的先備能力，並發現自己尚未理解的科學觀念(Siegmund et al., 2014; Sins et al.; Stacewicz & Włodarczyk)，以進一步建立完整的科學知識概念(Brewe, 2008; Svoboda & Passmore, 2013)。透過幫助學習者專注於現實的科學現象的概念重構，進而幫助學生建立統一而連貫的科學知識(Hestenes, 1987)。此外，建模可以讓學生積極參與科學探究活動，以獲取科學知識和技能(Oh & Oh, 2011)，同時也是實現認知、科學研究和解決問題的基礎，因此也有助於學生解決科學問題(Justi & Gilbert, 2002)。

參、研究方法

一、研究設計與流程

本研究採用準實驗研究法，課程參與者為臺北市某大學通識課的學生，實驗組25人，控制組10人，實驗組採建模引導之科學程式設計教學，控制組採傳統講述式科學程式設計教學。本研究採用的程式語言為Python，並使用Python內建的圖型化函式庫Turtle Graphics進行程式設計教學(Hunt, 2019)。實驗期程共16週，一週兩節課，一節課為50分鐘。第一週為課程簡介，第二至四週進行基本Python Turtle教學，並進行態度問卷、運算思維與科學問題解決能力前測，第六週起實驗組加入建模引導，而控制組維持傳統講述式教學，進行Python科學程式設計與科學問題解題。此處之傳統講授式教學，是指教師先說明程式語法與概念，接著示範解題並說明流程，然後由學生進行練習。第十六週則實施態度問卷與科學問題解決能力後測。實驗流程如表1。

表1：實驗流程

週次	實驗組上課內容	控制組上課內容
1	課程簡介	
2	Python Turtle	
3	Python Turtle、態度問卷前測	
4	Python Turtle、運算思維前測、科學問題解決能力前測	
5	國定假日	
6	透過建模歷程進程式設計以解決科學問題，並學習Python的input、print	透過程式設計以解決科學問題，並學習Python的input、print
7	透過建模歷程進程式設計以解決科學問題，並學習Python的integer和float	透過程式設計以解決科學問題，並學習Python的integer和float
8	透過建模歷程進程式設計以解決科學問題，並學習Python的if	透過程式設計以解決科學問題，並學習Python的if
9	透過建模歷程進程式設計以解決科學問題，並學習Python的for	透過程式設計以解決科學問題，並學習Python的for
10	期中考	
11	透過建模歷程進程式設計以解決科學問題—物理—一維等速運動	透過程式設計解決科學問題—物理—一維等速運動
12	透過建模歷程進程式設計以解決科學問題—物理—拋體運動	透過程式設計解決科學問題—物理—拋體運動
13	透過建模歷程進程式設計以解決科學問題—數學—碎形	透過程式設計解決科學問題—數學—碎形
14	透過建模歷程進程式設計以解決科學問題—生物—生命遊戲	透過程式設計解決科學問題—生物—生命遊戲
15	透過建模歷程進程式設計以解決科學問題—化學—理想氣體方程式	透過程式設計解決科學問題—化學—理想氣體方程式
16	科學問題解決能力後測、態度問卷後測	

二、建模引導之科學程式設計教學

為引導學生透過程式設計解決科學問題，本研究設計建模引導程式設計學習平臺以幫助學生逐步撰寫程式。Stacewicz與Włodarczyk (2010)提出的建模程序為：抽象化、形式化、簡化與查驗，而Sins等(2005)提出的建模程序則為：分析、歸納推論、量化與解說(Siegmund et al., 2014)，本研究參考此二文獻的建模程序，再根據程式設計的特性修改建模程序歷程為：「現象描述」、「資料建模」、「流程建模」、「程式化」以及

「觀察與除錯」，並據以設計輔助學習平臺，逐步引導實驗組學生形塑解題策略並撰寫程式以解決科學問題。

(一)現象描述

現象描述的目的是幫助學生在解題前先確認科學問題相關的重要元素，例如：題目的輸出與輸入、與問題相關的要素等。描述問題重要元素可幫助學生釐清解題方向，現象描述功能如圖1所示。

在平臺上「現象描述」的畫面中，會呈現需要解題的問題內容，題目可由教材匯

氧原子計算

現象描述 資料建模 流程建模 程式化 觀察與除錯

1

莫耳是科學上用來表示物質所含粒子（例如電子、原子或分子等）的單位，一莫耳的氧原子大約是 6×10^{23} 個氧原子。請輸入一個數為氧原子的莫耳數，並輸出此數相當於幾個氧原子。你認為在運算的過程中需要哪些運算的要素？

上傳文字 上傳圖片

文字

送出

1 2020年4月17日 14:41 -- 刪除

```
In [1]: n = float(input())
        num = 6*10**23*n
        print(num)
5
3e+24
```

In []:

2 2020年4月17日 14:45 -- 刪除 修改(0)

個數=莫爾數* 6×10^{23}

圖1：建模引導之科學程式設計——現象描述

入，亦可由教師手動輸入。學生須先分析以瞭解題目，並在下方欄位輸入對科學現象的描述，在操作上述環節時，學生必須寫下與問題解決相關的重要元素，亦可上傳圖片來作答。完成第一次現象描述後，授課教師會帶領學生釐清題意，學生可以再持續新增自己的描述。

(二)資料建模

資料建模步驟引導學生確認與問題相關的變數，並完整地描述所需的變數以及變數之間的關聯性，以更精確陳述與問題相關的資料屬性與關係，藉以建立資料模型，資料建模功能如圖2所示。

進行資料建模時，學生須依照在現象描述時找到的解題元素建構變數，並以式子陳述變數間的關聯性。學生可拖曳平臺所提供的各種變數、符號的方塊，以選取所需要的

變數類型並定義變數間的關聯性，進一步建立資料模型。待學生完成建模後，教師會引導學生找出模型錯誤並釐清想法。

(三)流程建模

流程建模目的是引導學生描述解題步驟與流程，藉以發展演算法模型，流程建模功能如圖3所示。

透過現象描述與資料建模後，學生已瞭解問題重點與元素，並將重點元素與其間之關聯性以變數與式子陳述，接著便須發展演算法以進行解題。為了減輕學生的認知負荷，在此「流程建模」程序中，學生以中文虛擬碼(pseudocode)陳述指令，並搭配平臺上的基本指令圖示(包含「處理」、「輸入」、「輸出」、「迴圈」、「判斷」)，組合成為解題的演算法。其中「迴圈」的流程框格用以輸入欲重複的指令，並用黃底區別迴圈內

庵原嘉峰 氧原子計算

現象描述 資料建模 流程建模 程式化 觀察與除錯

1

根據現象問題的描述，你認為這些運算的要素之間有什麼樣的關係及規則？請用下面的方塊表示。

一般變數 + Output Input

陣列變數 +

字面值 字串 數值

符號 () + - * / ^

新增運算式 請將上面的元件方塊拖曳至下方運算式的虛線框內...

Output = Input * 6 + 10 - 23

圖2：建模引導之科學程式設計——資料建模

庵原嘉峰 氧原子計算

現象描述 資料建模 流程建模 程式化 觀察與除錯

1

根據你在資料建模的描述，寫下運算過程，並詳細說明每個流程的內容。

新增流程 > 處理 輸入 輸出 迴圈 判斷

輸入 輸入要算幾個莫耳的原子數

↓

處理 輸入乘一個莫耳的原子數

↓

輸出 輸出處理過程的結果

圖3：建模引導之科學程式設計——流程建模(循序結構示意)

外區域，如圖4所示。教師會在學生完成建模後，引導學生找出錯誤與問題並釐清觀念。

(四)程式化

在學生建立了資料與流程的模型後，便可依此將演算法與資料結構轉化為程式碼，平臺之程式化畫面如圖5所示。



圖4：建模引導之科學程式設計——流程建模(迴圈結構示意)

庵原嘉峰 氧原子計算

現象描述 資料建模 流程建模 程式化 觀察與除錯

上傳 1

創作過程：全部靠自己想

程式碼：

```
1x = int(input())
2print(x*6*10**23)
```

執行結果：

copy

圖5：建模引導之科學程式設計——程式化

(五)觀察與除錯

寫完程式後，引導學生觀察執行結果並進行除錯，此步驟是使用外部程式直譯器進

行除錯，再回到平臺記錄除錯歷程，藉以進行驗證與反思，如圖6所示。

庵原嘉峰 氧原子計算

現象描述 資料建模 流程建模 程式化 觀察與除錯

錯誤記錄 1 程式碼:
1x = int(input())
2print(x*6*10**23)

錯誤類型:
程式語法錯誤

錯誤描述:

改進方法:

送出

圖6：建模引導之科學程式設計——觀察與除錯

三、資料蒐集與分析

本研究為驗證所提出的建模引導之科學程式設計教學的效能，在教學實驗前實施運算思維測驗與科學問題解決能力測驗，並透過問卷瞭解初期的學習態度，教學後則進行科學問題解決能力及學習態度問卷後測，實驗結束之後並進行學生訪談。根據所蒐集的問卷與測驗成績等量化研究資料，以曼－惠特尼 U 檢定(Mann-Whitney U test)比較實驗組與控制組在歷經不同的教學策略後，其學習成效與學習態度是否有差異，並以訪談結果及學生科學問題解決後測之作答內容為質性資料，佐證量化資料以進行更深入的討論。

為瞭解教學策略對不同先備能力學生之影響，本研究依據學生運算思維與科學問題解決能力前測之結果，將其分為運算思維高、低先備能力與科學問題解決高、低先備能力之學生分組。在運算思維先備能力方

面，乃依據實驗組與控制組所有學生之運算思維前測平均分數(3.9分)進行分組，大於等於平均分數者為高運算思維先備能力組(實驗組：14人、控制組：5人)，小於平均分數者為低運算思維先備能力組(實驗組：11人、控制組：5人)。在科學問題解決能力方面，則依據實驗組與控制組所有學生之科學前測平均分數(8.1分)進行分組，大於等於平均分數者為高科學問題解決先備能力組(實驗組：13人、控制組：6人)，小於平均分數者為低科學問題解決先備能力組(實驗組：12人、控制組：4人)。相關分析將於研究結果呈現。以下更詳細說明本研究有關運算思維、科學問題解決能力與學習態度之檢驗。

(一)運算思維

運算思維前測是以Bebras國際運算思維挑戰賽的題目為主，從中挑選模式辨識、資料表示與演算法思維三個面向的題目修改而成。此前測用來區分高、低先備能力之學生，並為後續進行Mann-Whitney U 檢定時之進步幅度計算基準。

(二)科學問題解決能力

如前所述，本研究之「科學問題解決能力」乃是指「透過運算解決科學問題的能力」，但由於課程實施前學生並未具備程式設計能力，因此科學問題解決能力前測僅包含科學概念問題，包含：拋體運動、基因遺傳、波以爾定律三個主題，主要是為檢驗學生科學問題解決之先備能力，用以區分高、低先備能力之學生，並為後續進行Mann-Whitney U 檢定時之進步幅度計算基準。題目範例見附錄一。

科學問題解決能力後測旨在檢驗課程後學生是否能透過程式設計解決科學問題，一共有七個題目，題目範例見附錄二。

(三)建模品質

為比較建模引導之科學程式設計教學對學生建模品質的影響，我們亦評鑑學生的建模品質，依照建模引導之步驟進行評分，但由於學生在「觀察與除錯」步驟時若執行結果無錯誤則無需填寫，因此此項目不計入評分。建模品質前測分數乃針對平臺上學生初期的建模內容進行評分，後測則針對學生科學問題解決測驗之作答內容評分。評分準則如表2所示，符合一項評分準則即加1分。程式化的步驟直接由學生寫的程式碼進行評分，前三步驟的評分則透過問答題瞭解學生的解題程序，如圖7之作答內容，在現象描述中，學生能正確找到解題元素(溫度、壓力、體積、容器、空氣分子)，且有描述輸入(體積、溫度)，但未能清楚描述程式輸出，並且未能用完整的句子清楚描述問題與期望的執行畫面，因此只得1分；在資料建模中，學生能設定壓力、溫度、體積的變數為P、T與V，且能解釋其意義及其與解題的關係，並能描述變數之間的關聯性(T與P成正比)，因此得到3分；在流程建模中，學生能知道解題

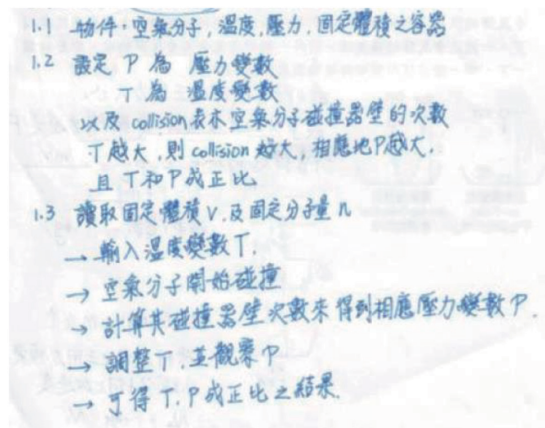


圖7：建模評分範例之學生作答內容

方法並循序寫出解題步驟，但未能以抽象形式(流程圖等)表達，因此得到2分。

(四)學習態度

為瞭解學生在科學程式設計課程中的學習態度，本研究參考Lin等(2019)的研究發展學習態度問卷，於實驗前後各施測一次。題目包含四個面向：科學學習興趣、科學學習感受、科學問題解決自我效能以及科學程式設計態度與感受，共15題李克特五點量表

表2：建模品質評分標準

建模歷程	評分準則
現象描述	A、學生能以完整和適當的呈現方式表達題目，例如：以文字、繪圖、表格等形式表達。 B、學生知道題目的輸出和輸入。 C、學生知道題目的重要元素(題目所提供的資訊)。
資料建模	A、學生能寫出變數之間的關係。 B、學生能寫出選擇或需要它們的原因。 C、學生能寫出該題所需的變數。
流程建模	A、學生能以抽象化的方式表達解題流程。 B、學生能寫出解題步驟。 C、學生能寫出解題方法。
程式化	A、學生能使用正確的資料型態宣告題目所需的變數。 B、學生能使用正確的程式結構(if, while, for)來進行解題。 C、學生所撰寫的程式碼能考量一般化或程式風格。 D、學生所撰寫的程式能正確運作以顯示欲模擬的現象。

(Likert scale)的題目(非常同意：5分、同意：4分、普通：3分、不同意：2分、非常不同意：1分)，題目如表3所示。透過Cronbach's α 值分析問卷之信度，科學學習興趣面向信度為.822、科學學習感受為.713、科學問題解決自我效能為.575、科學程式設計態度與感受為.694，而整體信度為.823。後測題目除了與前測相同的問題外，另有開放式的問答題，並有半結構式訪談，藉以更深入瞭解學生的學習態度。

(五)質性分析資料

課程結束後進行訪談，抽選控制組與實驗組高、中、低科學問題解決能力各1人(每組共3人)，訪談內容包含對課程的感受、對

科學程式設計的想法等，以更進一步瞭解學生學習狀況。訪談內容皆以錄音記錄，並分析逐字稿作為量化資料的佐證。主要訪談問題如表4。除了訪談，另一項質性資料為科學問題解決測驗之作答內容，這些質性資料可作為量化資料之佐證，並藉以交互討論建模與科學問題解決之關聯性。

肆、研究結果

一、建模引導之程式設計教學對建模品質的影響

為探討建模引導之科學程式設計課程中的建模引導是否能增進學生建模品質，以Mann-Whitney U 檢定分析實驗組與控制組學

表3：態度問卷題目

面向	編號	題目
科學學習興趣	1	我覺得科學是有趣的。
	2	我會主動學習科學新知。
	3	當我解決科學難題時，我會覺得很有成就感。
科學學習感受	4	我認為在日常生活中時常會使用到科學觀念。
	5	學習科學對於許多領域的工作都會有幫助。
	6	隨著科技的進步，科學的研究與發展也會有新的方法。
科學問題解決自我效能	7	當我遇到科學問題時，我知道該從何下手或知道需要用到那些觀念。
	8	當我遇到科學難題時，我可以自己思考可能的解決辦法。
	9	只要花足夠的時間，我通常能釐清科學問題解題的方向。
科學程式設計態度與感受	10	我認為課堂中使用科學問題進行程式設計的教學可以幫助我學習程式設計。
	11	未來我遇到較複雜的科學問題，我會嘗試透過程式解決問題。
	12	我能夠使用程式設計解決科學問題。
	13	我認為程式設計對科學是很重要的。
	14	很多困難、複雜的科學問題，透過程式會更容易解決。
	15	程式設計的過程可以幫助我重新思考科學問題的解題方法。
問答題(後測)	16	請問您對於此課程中將程式設計與科學整合有何看法？
	17	請問科學程式設計平臺中的解題步驟引導是否幫助你更能掌握用程式解決科學問題的流程？為什麼？
	18	請問科學程式設計平臺中哪個步驟最能幫助你透過程式解決科學問題？為什麼？

表4：實驗組與控制組訪談問題

問題	實驗組	控制組
1	在經歷過課程後，你現在對於科學程式設計的想法？	在經歷過課程後，你現在對於科學程式設計的想法？
2	利用程式來模擬科學現象或是解題是否能幫助你理解或學習科學？為什麼？	利用程式來模擬科學現象或是解題是否能幫助你理解或學習科學？為什麼？
3	科學程式設計的課程是否能幫助你學習程式？為什麼？	科學程式設計的課程是否能幫助你學習程式？為什麼？
4	在課程中的那些教學活動(建模平臺、老師講解、Snakify、完成作業)對你的學習最有幫助？	在課程中的那些教學活動(老師講解、Snakify、完成作業)對你的學習最有幫助？
5	若未來遇到複雜的科學問題是否會嘗試利用程式來解題？為什麼？	若未來遇到複雜的科學問題是否會嘗試利用程式來解題？為什麼？
6	你覺得科學運算平臺中的建模歷程引導是否能幫助你思考和釐清科學問題與概念？如何幫助？	在學習科學程式設計的時候有遇到那些困難嗎？有沒有什麼樣的引導可以協助你更能順利解決問題、完成程式？
7	你覺得科學運算平臺中的哪個建模歷程最能幫助你寫出程式來解決科學問題？如何幫助？	
8	你覺得有現象描述的引導是否能幫助你從複雜的科學問題中理出重要的解題元素(例如：輸出、輸入或問題目標等)，以幫助後續的程式設計？	
9	你覺得有資料建模的引導是否能幫助你找到用程式解決科學問題時所需的變數或變數之間的關係？如何幫助？	
10	你覺得流程建模的引導是否能幫助你思考科學程式問題的解題步驟與流程，以幫助後續的程式撰寫？如何幫助？	

生建模品質進步幅度的差異，包含：現象描述、資料建模、流程建模、程式化四個面向的建模程序，其Mann-Whitney U 檢定之 U 值分別為72.000、69.000、66.500、117.500， p 值分別為.054、.041、.031與.788，結果顯示建模引導之程式設計教學相較於傳統程式設計教學，學生能有較佳的資料建模與流程建模品質($p < .05$)，建模品質之描述統計與進步幅度之Mann-Whitney U 檢定結果如表5所示。

進一步針對低運算思維與低科學問題解決先備能力的學生進行分析，發現建模引導之程式設計教學能幫助學生在現象描述、資料建模與流程建模三項建模歷程都有較佳的表現，顯著性皆小於.05或.01(表6、表7)。

二、建模引導之程式設計教學對科學問題解決能力的影響

為探討建模引導之科學程式設計教學與傳統科學程式設計教學是否對學生的科學問題解決能力造成不同的影響，將實驗組與控制組學生科學問題解決測驗之前後測進步幅度以Mann-Whitney U 檢定進行分析，Mann-Whitney U 檢定之 U 值為73.000， $p = .059 (> .05)$ ，表示兩組學生的科學問題解決能力進步幅度沒有顯著差異，科學問題解決能力之描述統計與進步幅度之Mann-Whitney U 檢定結果如表8所示。

進一步分析教學策略對低運算思維能力學生的影響，以Mann-Whitney U 檢定分析

表5：建模品質前後測進步幅度之Mann-Whitney U 檢定結果

面向	測驗	實驗組($n = 25$)		控制組($n = 10$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
現象描述	前測	6.20	3.19	7.30	2.00	72.000	-1.951	.054
	後測	6.00	1.80	4.90	2.56			
資料建模	前測	9.88	5.53	13.20	3.22	69.000	-2.051	.041
	後測	6.44	2.18	5.90	2.51			
流程建模	前測	10.52	6.46	14.60	5.38	66.500	-2.144	.031
	後測	5.36	2.04	5.20	3.08			
程式化	前測	4.08	2.08	3.60	2.07	117.500	-0.276	.788
	後測	10.60	1.83	9.90	2.60			

表6：低運算思維先備能力學生建模品質前後測進步幅度之Mann-Whitney U 檢定結果

面向	測驗	實驗組($n = 11$)		控制組($n = 5$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
現象描述	前測	3.91	2.88	6.60	2.30	5.500	-2.515	.009
	後測	5.55	1.97	3.40	2.51			
資料建模	前測	5.64	3.41	11.80	3.90	5.500	-2.505	.009
	後測	5.36	2.11	4.60	3.05			
流程建模	前測	5.73	3.07	12.40	5.77	6.500	-2.406	.013
	後測	4.64	2.01	3.20	2.95			
程式化	前測	2.00	1.18	2.00	1.41	25.000	-0.288	.827
	後測	10.34	2.62	9.60	3.78			

表7：低科學問題解決先備能力學生的建模品質前後測進步幅度之Mann-Whitney U 檢定結果

面向	測驗	實驗組($n = 12$)		控制組($n = 4$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
現象描述	前測	3.92	2.75	5.50	1.29	6.500	-2.157	.030
	後測	5.33	1.50	3.25	2.50			
資料建模	前測	5.58	3.12	10.75	2.75	7.500	-2.017	.042
	後測	5.58	1.98	5.50	3.32			
流程建模	前測	5.42	2.39	10.00	3.56	5.500	-2.279	.020
	後測	4.67	1.83	4.00	2.71			
程式化	前測	2.83	2.03	2.00	1.83	21.500	-0.307	.770
	後測	10.33	2.42	8.75	4.03			

實驗組與控制組低運算思維先備能力學生的科學問題解決能力進步幅度之差異，Mann-Whitney U 檢定之 U 值為 9.500， $p = .038$ ($< .05$)，顯示建模引導之程式設計教學對運算思維能力較不足的學生有顯著的幫助，科學問題解決能力之描述統計與進步幅度之 Mann-Whitney U 檢定結果如表 9 所示。

我們亦想瞭解教學策略對於低科學問題解決先備能力學生的影響，因此以 Mann-Whitney U 檢定分析實驗組與控制組低科學問題解決先備能力學生科學問題解決能力進步幅度的差異，Mann-Whitney U 檢定之 U 值為 6.000， $p = .030$ ($< .05$)，顯示建模引導之程式設計教學對科學問題解決先備能力較不足的學生有顯著的幫助，描述統計與 Mann-Whitney U 檢定結果如表 10 所示。

三、建模引導之程式設計教學對學習態度的影響

為探討學生經歷科學程式設計課程後的學習態度是否與傳統教學有所差異，將實驗組與控制組態度問卷的結果差異分別做 Mann-Whitney U 檢定。兩組學生學習態度進步幅度之 Mann-Whitney U 檢定之 U 值分別為 77.000、88.000、116.000、105.000、102.500， p 值分別為 .083、.186、.760、.483、.418，顯著性皆大於 .05，顯示接受建模引導之程式設計教學與傳統教學之學生學習態度並未有顯著差異，描述性統計與 Mann-Whitney U 檢定分析結果如表 11。

表 8：科學問題解決能力之描述性統計與進步幅度之 Mann-Whitney U 檢定結果

面相	測驗	實驗組($n = 25$)		控制組($n = 10$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
科學問題解決能力	前測	7.56	3.70	9.50	2.22	73.000	-1.912	.059
	後測	16.36	3.26	13.90	4.99			

表 9：低運算思維先備能力學生之科學問題解決能力進步幅度之 Mann-Whitney U 檢定結果

面向	測驗	實驗組($n = 11$)		控制組($n = 5$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
科學問題解決能力	前測	4.82	3.09	8.80	2.59	9.500	-2.051	.038
	後測	15.45	4.32	12.40	5.32			

表 10：低科學問題解決先備能力學生之科學問題解決能力進步幅度之 Mann-Whitney U 檢定結果

面向	測驗	實驗組($n = 12$)		控制組($n = 4$)		Mann-Whitney U		
		M	SD	M	SD	U	Z	p
科學問題解決能力	前測	4.33	2.42	7.25	0.96	6.000	-2.204	.030
	後測	15.75	4.16	12.25	5.25			

表11：學習態度問卷之描述性統計與進步幅度之Mann-Whitney *U*檢定結果

面向	測驗	實驗組(<i>n</i> = 25)		控制組(<i>n</i> = 10)		Mann-Whitney <i>U</i>		
		<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>U</i>	<i>Z</i>	<i>p</i>
整體學習態度	前測	4.12	0.38	4.28	0.49	77.000	-1.753	.083
	後測	4.12	0.35	4.11	0.57			
科學學習興趣	前測	4.19	0.57	4.70	0.40	88.000	-1.370	.186
	後測	4.28	0.50	4.37	0.62			
科學學習感受	前測	4.45	0.47	4.10	0.93	116.000	-0.334	.760
	後測	4.47	0.44	3.93	0.86			
科學問題解決自我效能	前測	3.89	0.61	3.97	1.26	105.000	-0.762	.483
	後測	3.91	0.52	4.17	0.82			
科學程式設計態度與感受	前測	4.07	0.43	4.28	0.60	102.500	-0.837	.418
	後測	4.06	0.47	4.07	0.69			

伍、討論

一、建模引導之程式設計教學對學習成就(建模品質與科學問題解決能力)的影響

本節將合併討論建模引導的兩種學習成就：建模品質與科學問題解決能力，以利彼此佐證，更清楚闡釋兩者的交互影響。根據Mann-Whitney *U*檢定結果，發現建模引導之程式設計教學與傳統科學程式教學對整體學生科學問題解決能力的影響無顯著差異，但針對低運算思維先備能力與低科學問題解決能力的學生，建模輔助教學能顯著提升其科學問題解決能力。從實驗組與控制組低運算思維或科學問題解決先備能力學生在科學問題解決能力後測的答案(圖8、圖9)中來看，實驗組學生能較完整且精確地寫出科學概念的定義或其因果關係，雖然是低先備能力，實驗組學生仍能詳細描述變數之間的關係(例如：動量變化與其他變項的關係、溫度與壓力的關係等)、變數的假設(例如：對質量 m 與時間差 Δt 的假設、將壓力定義為 a 與溫度定為 T 的假設等)以及逐步透過因果關係陳述解題流程(例如：逐步推論到 F 的大小關係)，這

些部分是在控制組低先備能力學生的答案中缺乏的。

事實上，建模品質的分析結果亦支持此研究發現，如第肆章所示，建模引導之程式設計教學能提升所有學生的「資料建模」與「流程建模」品質，而這兩項也分別代表「運算思維」中的「資料表示」與「演算法設計」。對於低運算思維或科學問題解決先備能力的學生來說，建模引導教學則是能同時提升其「現象描述」、「資料建模」與「流程建模」品質。因此，對於低先備能力的學生來說，實驗組相較於控制組，學生能在問題元素的定義與描述(現象描述)、變項與其關係的描述(資料建模)以及因果關係的陳述(流程建模)三個方面有較佳的品質。在本研究的建模引導之程式設計教學中，學生須於建模引導學習平臺上逐步進行建模程序，藉由現象描述，能找到複雜科學問題中的重點元素，因而能掌握問題的解題方向，且透過資料建模更精確地描述所需的變數及其關係，接著再透過流程建模將解題步驟程序化。在此建模過程中，學生透過引導能不斷調整建模的想法(van Joolingen, 2004)，尤其在解決大型且複雜的問題時，建模教學能提升建模

實驗組

二、問答題

1. 若要利用 python 程式模擬[給呂薩克定律]的動畫:

- 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
- 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度?它們的關係為何?
- 1.3 請以流程圖或虛擬碼描述程式流程?

[給呂薩克定律]:定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速, 會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

$$PV = nRT \quad P \propto T$$

1.1 可參照波以爾定律所建構之內容, 固定幾何形狀 = 斜角板
由一面牆紀錄單位時間內分子碰撞(壁面/球體前後)
當 T 溫度上升, 則調整或分子移動速率 \propto 倍, 則壓力也會隨之提升

1.2 $\frac{\text{壓力}}{T} = \frac{\text{撞擊次數}}{\text{單位時間}}$
T: 溫度 每上升 1°C 增 \propto 倍

1.3 給定空間大小, 分子數, 位置方向, 給隨機初速度
→ 計時, 計次, 溫度調整滑桿(輸入框)
→ 顯示 $\frac{PV}{nR}$ 繪製動畫圖。

控制組

二、問答題

1. 若要利用 python 程式模擬[給呂薩克定律]的動畫:

- 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
- 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度?它們的關係為何?
- 1.3 請以流程圖或虛擬碼描述程式流程?

[給呂薩克定律]:定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速。會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

1.1 容器 → 一個桶
原子 → 小球

1.2 壓力 → 小球的密度 壓力大 → 速度快
溫度 → 小球的顏色 溫度 ↑ 速度 ↑

1.3

實驗組

二、問答題

1. 若要利用 python 程式模擬[給呂薩克定律]的動畫:

- 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
- 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度?它們的關係為何?
- 1.3 請以流程圖或虛擬碼描述程式流程?

[給呂薩克定律]:定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速, 會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

1.1 物件: 空氣分子, 溫度, 壓力, 固定體積之容器

1.2 設定 P 為 壓力變數
T 為 溫度變數
以及 collision 表示空氣分子撞擊器壁的次數
T 越大, 則 collision 越大, 相應地 P 越大,
且 T 和 P 成正比

1.3 讀取固定體積 V, 及固定分子數 n
→ 輸入溫度變數 T,
→ 空氣分子開始碰撞
→ 計算其撞擊器壁次數來得到相應壓力變數 P。
→ 調整 T, 並觀察 P
→ 可得 T, P 成正比之結果

控制組

二、問答題

1. 若要利用 python 程式模擬[給呂薩克定律]的動畫:

- 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
- 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度?它們的關係為何?
- 1.3 請以流程圖或虛擬碼描述程式流程?

[給呂薩克定律]:定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速。會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

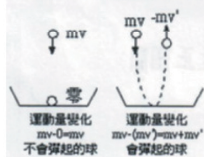
1.1 動畫會有一個房間, 一堆氣體分子。氣體分子會在房間內隨機初速度移動, 碰到房間壁會反彈, 當溫度數值越大, 氣體分子運動速度越快

1.2 壓力以固定時間內, 氣體分子撞擊每單位的面數, 溫度 T 以數值表示
壓力 \propto 溫度

圖8：低運算思維先備能力學生之科學問題解決能力測驗後測答案示例

實驗組

3. 大家小的時候應該都有玩過一種橡皮球吧，當這種橡皮球彈到地面的時候會反彈的很高，那如果今天我們利用兩種不同的球從磅秤上方高度處放下，一種是會反彈的橡皮球，另外一種則是完全不會反彈的球。那麼請問一下，哪一種球可以讓磅秤的讀數最大呢？並詳述原因或你的想法？



動量變化 = 衝量 = $m\Delta v = F\Delta t$

反彈球 $\Delta v_1 = v_0 - (-v_1)$, $|v_1| < |v_0|$

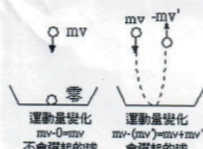
完全不反彈球 $\Delta v_2 = |v_0|$

設兩者 m 相同又 $\Delta v_1 > \Delta v_2$, 再設 Δt 兩者相同

$\Rightarrow F_1 > F_2 \Rightarrow$ 會彈起的秤讀數大。

控制組

3. 大家小的時候應該都有玩過一種橡皮球吧，當這種橡皮球彈到地面的時候會反彈的很高，那如果今天我們利用兩種不同的球從磅秤上方高度處放下，一種是會反彈的橡皮球，另外一種則是完全不會反彈的球。那麼請問一下，哪一種球可以讓磅秤的讀數最大呢？並詳述原因或你的想法？



$F_W = ma$, 而會彈起的球除 W 外, 有 $-F$ 正力
故於磅秤上, 故會影響讀數變大

實驗組

二、問答題

- 若要利用 python 程式模擬[給呂薩克定律]的動畫:
 - 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
 - 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度? 它們的關係為何?
 - 1.3 請以流程圖或虛擬碼描述程式流程?

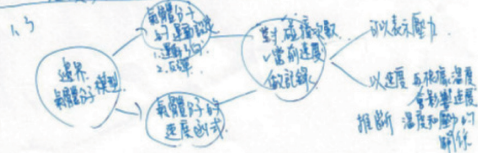
[給呂薩克定律]: 定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速。會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

1.1 做一個矩形的邊界, 4個邊。
很多氣體分子隨機移動。

1.2 氣體分子碰撞邊界的次數 \rightarrow 壓力

氣體分子的移動速度 \rightarrow 溫度

溫度 \uparrow 速度 \uparrow 單位時間碰撞次數 \uparrow 壓力 \uparrow




控制組

二、問答題

- 若要利用 python 程式模擬[給呂薩克定律]的動畫:
 - 1.1 簡單描述你的動畫會長什麼樣子, 或需要那些物件
 - 1.2 請問程式中需要哪些變數或要如何表示壓力、溫度? 它們的關係為何?
 - 1.3 請以流程圖或虛擬碼描述程式流程?

[給呂薩克定律]: 定量和定體積的條件下, 氣體的絕對溫度與壓力成正比, 也就是說絕對溫度越高, 壓力就越大。而溫度較高的氣體, 平均而言其中的原子(或分子)移動較為快速。會感受到容器裡的氣體具有壓力, 是因為無數的氣體分子不斷的撞擊容器壁, 使容器感受到向外的推力。當撞擊次數越多, 或是撞擊力道越大, 壓力就越大。

1.1  空氣分子, 碰撞次數, 溫度顯示

壓力可以反映碰撞次數

溫度可以反映碰撞分子的移動速度

溫度高, 速度快

圖9：低科學問題解決先備能力學生之科學問題解決能力測驗後測答案示例

品質，以引導系統化的解題程序，因而有較好的問題解決表現(S.-N. Chang, 2008)。但控制組學生缺乏建模引導，因此面對複雜的問題時，較無法逐步定義、分析問題並設計解題方法。過去的研究亦發現，學生進行建模時，透過反覆地修正和驗證，能刺激學生的先備能力，發現自己還不理解的科學觀念(Sins et al., 2005)，進一步建立完整的科學知識概念(Brewe, 2008; Svoboda & Passmore, 2013)，且能使學生對於科學現象形成適當的表徵，進而能有效解決科學問題(Justi & Gilbert, 2002)。

從訪談中亦可發現學生對於建模教學有正向的回饋，認為建模的引導能幫助他們解析問題：「建模歷程引導讓我能夠更快瞭解題目的問題，有利於我思考解決之道」(E-7007H)、「建模歷程能幫助我分析科學問題，並思考該如何用程式解題」(E-3020E)。每一個建模的步驟亦對學生有一定的幫助：

現象描述的引導可以讓我們從複雜的科學問題中，迅速找出關鍵的要素，以利我們後續的程式設計。(E-3013E)

資料建模能讓我清楚知道一個科學問題中變數與變數之間存在的關係，進而列出它們的關係式，方便下一個流程建模的進行。(E-7016H)

藉由流程建模的引導，可以得到基本的程式架構，再加上將變數之間的關係轉換成程式語法，就可以較為容易的撰寫程式了。(E-0119E)

二、建模引導之程式設計教學對學習態度的影響

在學習態度方面，雖然以平均數來看，

接受建模引導教學之實驗組學生在每一個向度皆高於接受傳統教學的控制組學生，但依據Mann-Whitney *U*檢定結果，兩組學習態度並無顯著差異，其可能原因為此次教學僅為期一個學期，對於學生科學學習興趣等面向恐無法在短時間內有顯著提升。

儘管如此，在開放式問題「您對於此課程中將程式設計與科學整合有何看法？」的回答中，實驗組有更多學生對於程式設計與科學結合的課程表達了正向的想法：

有學生感受到跨科整合是有趣的：

融合不同的學科，讓學習程式不再那麼無趣。(E-0226S)

之前曾經稍微接觸過程式設計，但從沒有想過可以將程式設計與科學問題的解決相互結合，真的很特別也很有趣。(E-3013E)

而難易適中的問題設計亦能引發學習動機：

非常能引起學習動機，有趣也有一些難度，但又不至於做不出來。(E-7012S)

跨科整合的設計亦能讓學生體驗程式設計在科學領域的實用性與必要性，過去的研究亦發現建模可以讓學生更積極參與的科學探究活動，以獲取科學知識和技能(Oh & Oh, 2011)：

有別於單純教導程式設計而多了實用性方面的應用我覺得很不錯。(E-7007H)

可以思考更多研究型的內容，以壓力來說吧，模擬氣體運動並不是主要目的吧？那化學感覺可以去思考更多需要模擬的題目，比方說電子軌域之類的。(E-7027S)

寫程式可以協助我們解決較複雜的科學問題。(E-7037S)

在一些繁瑣又重複的計算方面，還有現實中不容易做的實驗，透過程式能更方便地得出解答。(E-7016H)

這應該是必要的，有時候在做複雜或繁複的問題時很適合用程式解決。(E-7004H)

此外，實驗組學生亦提到了建模的逐步引導幫助他們理解及研究科學原理：

訓練逐步分析可以有助於理解及研究科學。(E-5038H)

可以透過不同的角度瞭解科學問題，而且因為以前習慣直接思考，學程式以後才知道，要用程式碼把想達成的效果表達出來是一件很不容易的事。(E-1022S)

科學問題本身可能很繁雜，但透過建模歷程能重新理解、甚至用不同角度去思考問題。(E-7016H)

相較於實驗組，控制組學生雖然也接受科學程式設計教學，但其回饋多較為表面：

我對MATLAB比對Python還有興趣。(C-4001S)

要與科學整合不如與數學整合。(C-7014S)

這可能是因為在缺乏引導的情況下，控制組學生對於要解決複雜的科學問題感到較為無助，而希望能使用函式庫較方便引用的程式語言直接撰寫，或能單純與數學結合，以避免在學習程式設計之外，還須學習複雜的科學問題解決。

事實上，實驗組學生在另一題開放式問題「請問科學程式設計平臺中的解題步驟引

導是否幫助你更能掌握用程式解決科學問題的流程？為什麼？」的回答中，有更多的肯定意見：

有，先設想好元素，在設計初期程式的大綱，再釐清程序，我覺得對解決問題是有幫助的。(E-3110S)

可以！能更清楚思路幫助邏輯思考的演進。(E-7007H)

有幫助，可以一步一步地思考，讓程式更順暢地完成。(E-0226S)

因為系統化、步驟化地解決問題可以避免徬徨與混亂。(E-0224E)

針對建模歷程的五個步驟，實驗組學生在開放式問題「科學程式設計平臺中哪個步驟最能幫助你透過程式解決科學問題？為什麼？」的回答中，多次肯定現象描述與流程建模對他們的幫助，此亦能說明實驗組學生在此兩步驟的建模品質較好的原因：

現象描述，因為可以釐清問題流程建模，建立一些順序，對之後程式碼的撰寫更為容易。(E-0226S)

第一個步驟，也就是界定問題與資源。因為我們無法順利解題往往都是因為不清楚問題在問什麼，當釐清問題後與手邊資源後，串接起來就變得容易許多。(E-0224E)

可能是第一步驟吧，先簡略的想需要什么，後續就會比較有目標。(E-1107H)

流程建模。因為他能讓整個問題的思路一目了然、方便思考判斷。(E-7007H)

流程建模，能一步步釐清問題的處理方式。(E-3001E)

流程建模，更清楚每個流程在做什麼。(E-7037S)

實驗組學生在訪談中，亦對於程式設計與科學的結合有較多正向的看法，認為此教學能讓他們多了一種解決科學問題的工具：

將程式與科學問題的結合令人感到意外，因為我從沒想過可以透過程式來解決科學問題。(E-0125S)

科學程式設計可以利用程式語言來表現科學現象跟解決科學問題，讓我瞭解可以用不一樣的觀點來解決科學問題，我認為這會是我以後在面對科學問題的時候會考慮的一個解題方向。(E-3013E)

亦能幫助他們更理解科學原理：

在修課前以為只能靠著背公式、規則來學習科學、解決科學問題也只能手算，但這堂課讓我學到如何利用程式的思維解決科學問題，這讓我對於以前學過的概念有了更多元的理解。(E-0119E)

此課程可以幫助我學習科學，因為利用程式可以清楚看到科學現象的展現，讓我們更容易瞭解這個科學現象的運作，所以能夠幫助我們理解。(E-7016H)

對於建模的引導，亦認為能幫助他們建立有邏輯的程式設計程序：

此課程以Python做為主要授課之程式語言，雖是以Python為主軸，但寫程式的邏輯跟其他語言差不了太多，所以可以幫助我在學習其他程式語言的時候能夠以相似的邏輯來寫出程式。(E-7046O)

然而，少數高先備能力的學生，認為平臺的輔助反而讓他們無法自由撰寫程式：

有時候腦袋裡的想法直接以程式撰寫似乎更加順利，平臺有一點硬要我們解構自己的想法，梳理成一個流程，有時候反而是負擔。(E-0119E)

這可能是由於高先備能力的學生原本就具備較好的建模能力，能在腦中形塑問題解決程序的表徵，因此較不需要逐步引導，此或許也是高先備能力的學生在實驗組與控制組間無顯著學習成就差異的原因。

陸、結論與建議

一、結論

根據教學實驗的結果，本研究主要有三項研究發現，分述如下：(一)建模引導之科學程式設計教學能提升學生建模的品質——本研究所發展的建模引導之科學程式設計教學，能顯著提升學生「資料建模」與「流程建模」的建模品質，此亦分別對應到運算思維的「資料表示」與「演算法設計」；而對低先備能力學生的影響更為顯著，能提升其「現象描述」、「資料建模」與「流程建模」三個步驟的建模品質。透過建模引導之程式設計學習平臺的引導，學生能在複雜的科學問題情境中抓出重點，藉以較清楚掌握科學問題的關鍵元素，能思考解題所需的變數與變數之間的關係，以建立程式設計所需的資料模型，並能以適當的順序陳述解題流程與步驟，以設計演算法的流程，這些建模步驟都有助於後續的程式碼編寫與問題解決。(二)建模引導之科學程式設計教學能提升低先備能力學生科學問題解決能力——透過建模輔助，原本對於複雜科學問題感到

無從著手的低先備能力學生，能逐步依照平臺的引導，將複雜的科學問題透過「現象描述」、「資料建模」與「流程建模」進行分析並規劃程式所需之資料與演算法，以進一步撰寫程式以解決科學問題，因而能有較佳之科學問題解決能力。(三)建模引導之科學程式設計教學能讓學生對於科學程式設計與科學問題解決有正向的看法——雖然實驗組與控制組都接受科學程式設計教學，但實驗組學生因有建模引導，其對科學與程式設計整合教學在開放式問答中有更多正面的回饋，並強調程式設計能協助解決複雜的科學問題，並有助於科學理解與探索。此乃由於建模引導之程式設計教學能逐步引導學生在複雜的科學問題情境中找到解題的脈絡，因此更能清楚說明科學程式設計的優點。再者，實驗組學生能明顯感受「現象描述」與「流程建模」對科學問題解決的助益，這也說明了為何他們在「現象描述」與「流程建模」步驟中都有較佳的建模品質。

二、建議

多數學生對於本研究所發展的建模引導之科學程式設計平臺持正向的看法，但少數

高成就的學生認為建模的引導限制了解題的程序，因此未來可針對高成就者的需求，簡化建模的引導步驟，並設計更為友善的使用介面，幫助學生更容易進行變數的設定與流程的設計。此外，本研究之研究對象人數較少，未來可於更多班級進行實驗，以提高樣本數與研究結果的正確性。此外，本研究乃針對科學領域學生開發教材，未來可將建模輔助教學推廣至人文社會等其他領域之程式設計教學，以幫助非資訊科學主修之學生逐步建構解題程序並培養其以程式設計解決領域問題之能力。

誌謝

本論文為國家科學及技術委員會計畫編號MOST 109-2511-H-003-010-MY3之計畫成果，且接受國立臺灣師範大學學習科學跨國頂尖研究中心之經費支持。論文的部分內容是出自作者的學位論文。此外，國立臺灣大學物理學系石明豐教授、臺北市立中山女子高級中學朱德清老師、臺北市立第一女子高級中學陳怡芬老師協助教材發展與教學實驗相關事項，使本研究得以順利進行，特此感謝。

參考文獻

- Abdullah, S., & Shariff, A. (2008). The effects of inquiry-based computer simulation with cooperative learning on scientific thinking and conceptual understanding of gas laws. *Eurasia Journal of Mathematics, Science and Technology Education*, 4(4), 387-398. <https://doi.org/10.12973/ejmste/75365>
- Akçay, B. (2009). Problem-based learning in science education. *Journal of Turkish Science Education*, 6(1), 26-36. <https://bit.ly/49iy8hH>
- Angeli, C. (2022). The effects of scaffolded programming scripts on pre-service teachers' computational thinking: Developing algorithmic thinking through programming robots. *International Journal of Child-Computer Interaction*, 31, Article 100329. <https://doi.org/10.1016/>

j.ijcci.2021.100329

- Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105, Article 105954. <https://doi.org/10.1016/j.chb.2019.03.018>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>
- Bayman, P., & Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, 80(3), 291-298. <https://doi.org/10.1037/0022-0663.80.3.291>
- 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. <https://doi.org/10.1111/j.1949-8594.2011.00109.x>
- Brewe, E. (2008). Modeling theory applied: Modeling instruction in introductory physics. *American Journal of Physics*, 76(12), 1155-1160. <https://doi.org/10.1119/1.2983148>
- Brewe, E., & Sawtelle, V. (2018). Modelling instruction for university physics: Examining the theory in practice. *European Journal of Physics*, 39(5), Article 054001. <https://doi.org/10.1088/1361-6404/aac236>
- Bühner, M., Kröner, S., & Ziegler, M. (2008). Working memory, visual-spatial intelligence and their relationship to problem-solving. *Intelligence*, 36(6), 672-680. <https://doi.org/10.1016/j.intell.2008.03.008>
- Burrows, A. C., Breiner, J. M., Keiner, J., & Behm, C. (2014). Biodiesel and integrated STEM: Vertical alignment of high school biology/biochemistry and chemistry. *Journal of Chemical Education*, 91(9), 1379-1389. <https://doi.org/10.1021/ed500029t>
- Bybee, R. W. (2010). What is STEM education? *Science*, 329(5995), 996. <https://doi.org/10.1126/science.1194998>
- Bybee, R. W. (2013). *The case for STEM education: Challenges and opportunities*. National Science Teachers Association.
- Chabay, R., & Sherwood, B. (2008). Computational physics in the introductory calculus-based course. *American Journal of Physics*, 76(4), 307-313. <https://doi.org/10.1119/1.2835054>
- Chang, K.-E., Chen, Y.-L., Lin, H.-Y., & Sung, Y.-T. (2008). Effects of learning support in simulation-based physics learning. *Computers & Education*, 51(4), 1486-1498. <https://doi.org/10.1016/j.compedu.2008.01.007>
- Chang, S.-N. (2008). The learning effect of modeling ability instruction. *Asia-Pacific Forum on Science Learning and Teaching*, 9(2), Article 3. <https://bit.ly/49gf4k6>

- Chi, H., & Jain, H. (2011). Teaching computing to STEM students via visualization tools. *Procedia Computer Science*, 4, 1937-1943. <https://doi.org/10.1016/j.procs.2011.04.211>
- Chonacky, N., & Winch, D. (2008). Integrating computation into the undergraduate curriculum: A vision and guidelines for future developments. *American Journal of Physics*, 76(4), 327-333. <https://doi.org/10.1119/1.2837811>
- Connor, K. A., Ferri, B. H., & Meehan, K. (2013, June). *Models of mobile hands-on STEM education*. 2013 ASEE Annual Conference & Exposition, Atlanta, GA. <https://doi.org/10.18260/1-2--22295>
- Crawford, B. (2015). Authentic science. In R. Gunstone (Ed.), *Encyclopedia of science education* (pp. 113-115). Springer. https://doi.org/10.1007/978-94-007-2150-0_144
- Davies, S. P. (1993). Models and theories of programming strategy. *International Journal of Man-Machine Studies*, 39(2), 237-267. <https://doi.org/10.1006/imms.1993.1061>
- Delisle, R. (1997). *How to use problem-based learning in the classroom*. Association for Supervision and Curriculum Development.
- Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2), 173-197. <https://doi.org/10.30935/cet.554493>
- English, L. D. (2017). Advancing elementary and middle school STEM education. *International Journal of Science and Mathematics Education*, 15(Suppl. 1), 5-24. <https://doi.org/10.1007/s10763-017-9802-x>
- Ferreira, M. M., & Trudel, A. R. (2012). The impact of problem-based learning (PBL) on student attitudes toward science, problem-solving skills, and sense of community in the classroom. *Journal of Classroom Interaction*, 47(1), 23-30.
- Foshay, R., & Kirkley, J. (1998). *Principles for teaching problem solving*. <https://bit.ly/4axxcHu>
- García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. <https://doi.org/10.1016/j.chb.2017.12.005>
- Gilbert, J. K. (2004). Models and modelling: Routes to more authentic science education. *International Journal of Science and Mathematics Education*, 2(2), 115-130. <https://doi.org/10.1007/s10763-004-3186-4>
- Gilbert, J. K., Boulter, C. J., & Elmer, R. (2000). Positioning models in science education and in design and technology education. In J. K. Gilbert & C. J. Boulter (Eds.), *Developing models in science education* (pp. 3-17). Springer. https://doi.org/10.1007/978-94-010-0876-1_1
- Gonzalez, H. B., & Kuenzi, J. J. (2012). *Science, technology, engineering, and mathematics*

- (STEM) education: A primer. <https://reurl.cc/E1O5aK>
- Greca, I. M., & Moreira, M. A. (2002). Mental, physical, and mathematical models in the teaching and learning of physics. *Science Education*, 86(1), 106-121. <https://doi.org/10.1002/sce.10013>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Hambruch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, 41(1), 183-187. <https://doi.org/10.1145/1539024.1508931>
- Hestenes, D. (1987). Toward a modeling theory of physics instruction. *American Journal of Physics*, 55(5), 440-454. <https://doi.org/10.1119/1.15129>
- Huang, H.-W., Wu, C.-W., & Chen, N.-S. (2012). The effectiveness of using procedural scaffoldings in a paper-plus-smartphone collaborative learning context. *Computers & Education*, 59(2), 250-259. <https://doi.org/10.1016/j.compedu.2012.01.015>
- Hunt, J. (2019). Python turtle graphics. In J. Hunt (Ed.), *Advanced guide to Python 3 programming* (pp. 13-21). Springer. https://doi.org/10.1007/978-3-030-25943-3_3
- Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L., Basu, S., & McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1), 83-100. <https://doi.org/10.1007/s10956-019-09804-9>
- Jimoyiannis, A., & Komis, V. (2001). Computer simulations in physics teaching and learning: A case study on students' understanding of trajectory motion. *Computers & Education*, 36(2), 183-204. [https://doi.org/10.1016/S0360-1315\(00\)00059-2](https://doi.org/10.1016/S0360-1315(00)00059-2)
- Justi, R. S., & Gilbert, J. K. (2002). Modelling, teachers' views on the nature of modelling, and implications for the education of modellers. *International Journal of Science Education*, 24(4), 369-387. <https://doi.org/10.1080/09500690110110142>
- Kordaki, M., Miatidis, M., & Kapsampelis, G. (2008). A computer environment for beginners' learning of sorting algorithms: Design and pilot evaluation. *Computers & Education*, 51(2), 708-723. <https://doi.org/10.1016/j.compedu.2007.07.006>
- Lim, Y.-K. (Ed.). (2000). *Problems and solutions on atomic, nuclear and particle physics*. World Scientific. <https://doi.org/10.1142/4139>
- Lin, Y.-T., Wang, M.-T., & Wu, C.-C. (2019). Design and implementation of interdisciplinary STEM instruction: Teaching programming by computational physics. *The Asia-Pacific Education Researcher*, 28(1), 77-91. <https://doi.org/10.1007/s40299-018-0415-0>
- Lin, Y.-T., Yeh, M. K.-C., & Tan, S.-R. (2022). Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations. *IEEE Transactions on Education*, 65(4), 617-627. <https://doi.org/10.1109/TE.2022.3155884>

- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Magana, A. J., Marepalli, P., & Clark, J. V. (2011, October). *Work in progress—Integrating computational and engineering thinking through online design and simulation of multidisciplinary systems*. 2011 Frontiers in Education Conference (FIE), Rapid City, SD. <https://doi.org/10.1109/FIE.2011.6143083>
- Maltese, A. V., Melki, C. S., & Wiebke, H. L. (2014). The nature of experiences responsible for the generation and maintenance of interest in STEM. *Science Education*, 98(6), 937-962. <https://doi.org/10.1002/sce.21132>
- Marlino, M. R. (2001). Visualization in undergraduate geoscience science education: What do we really know? *Computers & Geosciences*, 27(4), 497-498. [https://doi.org/10.1016/S0098-3004\(00\)00127-8](https://doi.org/10.1016/S0098-3004(00)00127-8)
- Matarić, M. J., Koenig, N., & Feil-Seifer, D. (2007, March). *Materials for enabling hands-on robotics and STEM education*. AAAI Spring Symposium: Semantic Scientific Knowledge Integration, Stanford, CA. <https://bit.ly/3RJPay1>
- McKeithen, K. B., Reitman, J. S., Rueter, H. H., & Hirtle, S. C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, 13(3), 307-325. [https://doi.org/10.1016/0010-0285\(81\)90012-8](https://doi.org/10.1016/0010-0285(81)90012-8)
- Morrison, J. S. (2006). *Attributes of STEM education: The student, the school, the classroom*. <https://bit.ly/4atkolC>
- National Research Council. (2011). *Successful K-12 STEM education: Identifying effective approaches in science, technology, engineering, and mathematics*. National Academies Press.
- Neves, R. G. M., Neves, M. C., & Teodoro, V. D. (2013). Modellus: Interactive computational modelling to improve teaching of physics in the geosciences. *Computers & Geosciences*, 56, 119-126. <https://doi.org/10.1016/j.cageo.2013.03.010>
- Oh, P. S., & Oh, S. J. (2011). What teachers of science need to know about models: An overview. *International Journal of Science Education*, 33(8), 1109-1130. <https://doi.org/10.1080/09500693.2010.502191>
- Pennington, N. (1987). Stimulus structures and mental representations in expert comprehension of computer programs. *Cognitive Psychology*, 19(3), 295-341. [https://doi.org/10.1016/0010-0285\(87\)90007-7](https://doi.org/10.1016/0010-0285(87)90007-7)
- Prabhu, P., Jablin, T. B., Raman, A., Zhang, Y., Huang, J., Kim, H., Johnson, N. P., Liu, F., Ghosh, S., Beard, S., Oh, T., Zoufaly, M., Walker, D., & August, D. I. (2011). A survey of the practice of computational science. In S. Lathrop, J. Costa, & W. Kramer (Eds.), *SC '11: State of the practice reports* (Article 19). Association for Computing Machinery. <https://doi.org/10.1145/1955558.1955559>

org/10.1145/2063348.2063374

- Psycharis, S. (2013). Examining the effect of the computational models on learning performance, scientific reasoning, epistemic beliefs and argumentation: An implication for the STEM agenda. *Computers & Education*, 68, 253-265. <https://doi.org/10.1016/j.compedu.2013.05.015>
- Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. *ACM SIGCSE Bulletin*, 41(1), 188-191. <https://doi.org/10.1145/1539024.1508932>
- Redish, E. F., & Wilson, J. M. (1993). Student programming in the introductory physics course: M.U.P.P.E.T. *American Journal of Physics*, 61(3), 222-232. <https://doi.org/10.1119/1.17295>
- Reed, A. (2019). *The effects of community partnerships on K-12 computer science education* [Unpublished doctoral dissertation]. Auburn University. <https://bit.ly/3TzqBq5>
- Rodríguez del Rey, Y. A., Cawanga Cambinda, I. N., Deco, C., Bender, C., Avello-Martínez, R., & Villalba-Condori, K. O. (2021). Developing computational thinking with a module of solved problems. *Computer Applications in Engineering Education*, 29(3), 506-516. <https://doi.org/10.1002/cae.22214>
- Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sadler, P. M., Sonnert, G., Hazari, Z., & Tai, R. (2012). Stability and volatility of STEM career interest in high school: A gender study. *Science Education*, 96(3), 411-427. <https://doi.org/10.1002/sce.21007>
- Sanders, M. (2009). STEM, STEM education, STEMmania. *The Technology Teacher*, 68(4), 20-26.
- Sanford, K., & Madill, L. (2007). Recognizing new literacies: Teachers and students negotiating the creation of video games in school. In A. Baba (Ed.), *DiGRA '07—Proceedings of the 2007 DiGRA International Conference: Situated Play* (Vol. 4, pp. 583-589). Digital Games Research Association.
- Scherer, R., & Tiemann, R. (2012). Factors of problem-solving competency in a virtual chemistry environment: The role of metacognitive knowledge about strategies. *Computers & Education*, 59(4), 1199-1214. <https://doi.org/10.1016/j.compedu.2012.05.020>
- Schmidt, D. C. (2006). Guest editor's introduction: Model-driven engineering. *Computer*, 39(2), 25-31. <https://doi.org/10.1109/MC.2006.58>
- Schnittka, C., & Bell, R. (2011). Engineering design and conceptual change in science: Addressing thermal energy and heat transfer in eighth grade. *International Journal of Science Education*, 33(13), 1861-1887. <https://doi.org/10.1080/09500693.2010.529177>
- Selby, C. C., & Woollard, J. (2013). *Computational thinking: The developing definition*. <https://bit.ly/3v3xFRr>

- Shen, J., Lei, J., Chang, H.-Y., & Namdar, B. (2014). Technology-enhanced, modeling-based instruction (TMBI) in science education. In J. M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop (Eds.), *Handbook of research on educational communications and technology* (pp. 529-540). Springer. https://doi.org/10.1007/978-1-4614-3185-5_41
- Siegmund, J., Kästner, C., Liebig, J., Apel, S., & Hanenberg, S. (2014). Measuring and modeling programming experience. *Empirical Software Engineering*, 19(5), 1299-1334. <https://doi.org/10.1007/s10664-013-9286-4>
- Sins, P. H. M., Savelsbergh, E. R., & van Joolingen, W. R. (2005). The difficult process of scientific modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education*, 27(14), 1695-1721. <https://doi.org/10.1080/09500690500206408>
- Soloway, E. (1986). Learning to program = Learning to construct mechanisms and explanations. *Communications of the ACM*, 29(9), 850-858. <https://doi.org/10.1145/6592.6594>
- Stacewicz, P., & Włodarczyk, A. (2010). Modeling in the context of computer science—A methodological approach. *Studies in Logic, Grammar and Rhetoric*, 20(33), 155-179.
- Svoboda, J., & Passmore, C. (2013). The strategies of modeling in biology education. *Science & Education*, 22(1), 119-142. <https://doi.org/10.1007/s11191-011-9425-5>
- Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing*, 3, 3657-3662. <https://doi.org/10.1016/j.promfg.2015.07.761>
- Taub, R., Armoni, M., Bagno, E., & Ben-Ari, M. (2015). The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87, 10-23. <https://doi.org/10.1016/j.compedu.2015.03.013>
- Teodoro, V. D., & Neves, R. G. (2011). Mathematical modelling in science and mathematics education. *Computer Physics Communications*, 182(1), 8-10. <https://doi.org/10.1016/j.cpc.2010.05.021>
- van Joolingen, W. (2004). Roles of modeling in inquiry learning. In Kinshuk, C.-K. Looi, E. Sutinen, D. Sampson, I. Aedo, L. Uden, & E. Kähkönen (Eds.), *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings* (pp. 1096-1097). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ICALT.2004.1357768>
- Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computers in a learning environment. *Egyptian Computer Science Journal*, 36(4), 28-46. <https://doi.org/10.48550/arXiv.1212.0750>
- Weinman, N., Fox, A., & Hearst, M. A. (2021). Improving instruction of programming patterns with Faded Parsons Problems. In Y. Kitamura, A. Quigley, K. Isbister, T. Igarashi, P. Bjørn, & S. Drucker (Eds.), *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Article 53). Association for Computing Machinery. <https://doi.org/10.1145/3411461.3411461>

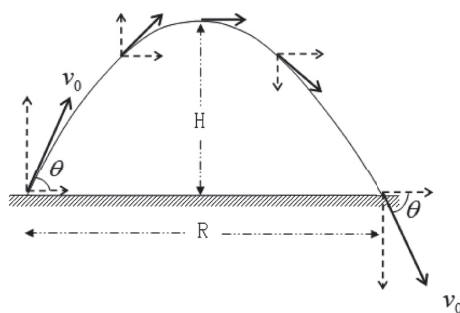
org/10.1145/3411764.3445228

- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wüstenberg, S., Greiff, S., & Funke, J. (2012). Complex problem solving—More than reasoning? *Intelligence*, 40(1), 1-14. <https://doi.org/10.1016/j.intell.2011.11.003>
- Xie, B., Loksa, D., Nelson, G. L., Davidson, M. J., Dong, D., Kwik, H., Tan, A. H., Hwa, L., Li, M., & Ko, A. J. (2019). A theory of instruction for introductory programming skills. *Computer Science Education*, 29(2-3), 205-253. <https://doi.org/10.1080/08993408.2019.1565235>
- Xie, Y., & Reider, D. (2014). Integration of innovative technologies for enhancing students' motivation for science learning and career. *Journal of Science Education and Technology*, 23(3), 370-380. <https://doi.org/10.1007/s10956-013-9469-1>
- Xie, Y., Fang, M., & Shauman, K. (2015). STEM education. *Annual Review of Sociology*, 41, 331-357. <https://doi.org/10.1146/annurev-soc-071312-145659>
- Zawojewski, J. (2013). Problem solving versus modeling. In R. Lesh, P. L. Galbraith, C. R. Haines, & A. Hurford (Eds.), *Modeling students' mathematical modeling competencies: ICTMA 13* (pp. 237-243). Springer. https://doi.org/10.1007/978-94-007-6271-8_20

附錄

附錄一：科學問題解決能力前測題目範例

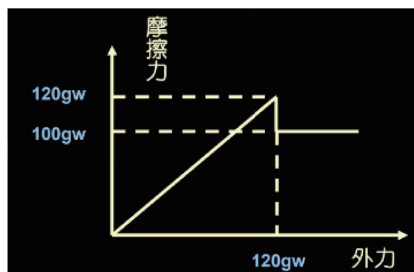
如圖所示，在地表附近，一物體以初速度 v_0 被拋射至空中。如果只受到地球的重力作用，則拋射體的運動軌跡必為一條拋物線。在拋物線路徑上，實線的箭頭向量，代表拋射體在該處的實際速度，向量 \rightarrow 代表水平速度分量，向量 \downarrow 或 \uparrow 代表鉛直速度分量。圖中的 θ 代表初速度 v_0 與水平面的夾角，稱為拋射仰角，而 v_0 則代表 v_0 的大小(即量值)，稱為初速。拋射體受到的地球重力加速度，係沿鉛直方向向下，因此其鉛直速度的大小，在上升過程中，每秒約減少10 m/s，而在下降過程中，每秒約增加10 m/s，但其水平速度則到處都相同。拋射體在軌跡頂端時，鉛直速度為零，故其速度等於水平速度(假設小石頭只受到地球的重力作用)。我們想探討以相同初速度自地面拋射一小石頭，拋射仰角是否影響小石頭在最高點的速度。請回答以下題目。



1. 這個問題與那些物理現象或定律相關？請簡述。
2. 小石頭運行過程中可以分為哪些速度分量？拋射仰角大小和重力分別會影響哪些速度分量？所造成的影響為何？請詳述。
3. 請問以相同初速自地面拋射一小石頭，拋射仰角大小是否會影響到石頭在最高點的速度？為什麼？

附錄二：科學問題解決能力後測題目範例

若有一物靜置於水平桌上，其摩擦力與水平推力的關係如圖所示：



此物體的最大靜摩擦力為120 gw，推力要大於最大靜摩擦力才能使物體移動，而此移動物體的動摩擦力皆為100 gw。

若推力小於等於最大靜摩擦力則物體無法移動，而此時物體的摩擦力剛好就是推力。

請寫一個程式，隨機產生一個推力的量，並輸出此推力對物體所產生的摩擦力為何。

Modelling-Based Instruction for Scientific Programming

Min-Hsuan Lee, Sheng-Che Hsiao, Si-Ru Chen and Yu-Tzu Lin *

Graduate Institute of Information and Computer Education, National Taiwan Normal University

Abstract

This study aims to design a modelling-based instruction for scientific programming. Through the guidance by a modelling-based learning platform, students had to follow a step-by-step modelling process, including “phenomenon description,” “data modelling,” “logic modelling,” “coding,” and “verification and debugging.” This instruction is to enable students to gradually develop problem-solving strategies and write program code to solve scientific problems. A quasi-experimental research was conducted to examine the effectiveness of the modelling-based instruction on learning of scientific programming. The experimental group received modelling-guided instruction, while the control group received traditional instruction. Based on the results, the following research findings were identified: (1) Modelling-based instruction helped students build their solutions in various ways: it enabled students to extract key elements of the problem from complex scientific phenomena, provide more accurate descriptions of variables and their relationships, and represent problem-solving steps more logically. Overall, this approach enhanced the quality of students’ “data modelling” and “logic modelling,” corresponding to “data representation” and “algorithm design” of computational thinking, respectively. For students with lower prior ability, it also improved the quality of “phenomenon description.” (2) Since the modelling-based instruction improved the modelling quality of students with lower prior knowledge, it helped them gradually develop problem-solving strategies, represent data more accurately, design algorithms, and then write code to solve scientific problems. As a result, it enhanced their ability to solve scientific problems. (3) As modelling-based instruction gradually guided students to find the strategies for problem-solving in complex scientific problem context, the experimental group students clearly perceived the benefits of “phenomenon description” and “logic modelling” in solving scientific problems. They could also articulate the advantages of scientific programming more clearly and affirm the role that programming plays in concept understanding, exploration, and problem-solving in the field of science.

Key words: STEM Education, Modelling, Computational Science, Programming Instruction, Computational Thinking

* Corresponding author: Yu-Tzu Lin, linyt@ntnu.edu.tw; ORCID: 0000-0002-2138-538X

Received: 2023/6/1, Revised: 2023/9/14, Accepted: 2023/9/18, Available Online: 2024/1/29