探索小數點以下十兆位的秘密

從 π 的 BBP 公式浪潮談起

作者：貝利（David H. Bailey）・波宛（Jonathan M. Borwein）・麥廷理（Andrew Mattingly）・外特威克（Glenn Wightwick）
譯者：林真
貝利是勞倫斯柏克萊國家實驗室計算研究部門的資深科學家。波宛是澳大利亞紐卡索大學電腦輔助研究研究與其應用中心的數學教授。
麥廷理是澳大利亞 IBM 的資深資訊技術工程師。外特威克是澳大利亞 IBM 研究中心主任。

我們最近剛完成一場非常龐大的計算，發現一些以前被認為永遠無法計算的東西。我們的計算源於計算 π 值的 BBP 公式，那是 1997 年利用電腦執行 PSLQ 整線性關係演算法（integer relation algorithm）時所發現的。這個絕妙的公式，可從任意指定的第 d 位數開始，計算 π 的二進位數字（binary digit），完全不需要計算之前的 $d-1$ 位數字。自 1997 年開始，許多計算其他數學常數的 BBP 型公式陸續被發現。其中包括 π^2（包含二進位、三進位）和卡塔蘭常數（Catalan's constant）。

在這篇文章中，我們將描述如何計算 64 進位與 729 進位的 π^2，以及 4096 進位的卡塔蘭常數。每個情形都是從第十兆位起算，總共大約需要 1.549×10^{19} 個浮點運算。我們也會討論 BBP 型公式和一些歷史悠久的待解問題之間的關連，譬如 π、π^2、$\log 2$、卡塔蘭常數是否與為何其數字出現的模式是「隨機」的。❶

歷史巡禮

自從文明肇始，數學家對 π 的著迷便遠勝過其他數學常數。西元前三世紀，阿基米德（Archimedes）就發展出一套算法，用圓的內接和外切 $3\cdot2^n$ 邊形，計算 π 精準至十進位小數點後第二位。然而，諸如此類的古老數值方法往往受限於原始的算術系統。

歷史上最重要的科學進展之一，發生在西元六世紀或更早之前的印度，有一位或一些數學家發展出包含 0 在內的十進位算術。一些最早的文獻，包括數學家阿耶波多（Aryabhata）於西元 499 年所著的《阿耶波多曆算書》（*Āryabhatīya*）；一部包含天文觀測的宇宙論著《羅卡毗巴迦》（*Lokavibhaga*，宇宙的組成），現代學者根據書中記錄推測該書成於西元 458 年 8 月 25 日（參見 [9]）；以及最遲於七世紀前（有些學者認為更古老）完成的古數學文件〈巴克夏利手稿〉（Bakhshālī manuscript）（參見 [7]、[8]、[2]）。在〈巴克夏利手稿〉中記載了底下，從某近似值 x_0 計算 q 平方根的有趣演算法：

$$a_n = \frac{q - x_n^2}{2x_n}$$

$$x_{n+1} = x_n + a_n - \frac{a_n^2}{2(x_n + a_n)}$$

這個算法每遞迴一次，大約能讓正確位數變成前一次的四倍，因此是四次收斂的。儘管在手稿的例子裡，遞迴次數都不超過一次（參見 [2]）。

西元十世紀，歐里亞克的吉爾伯特（Gerbert of Aurillac）已嘗試將十進位算術引進歐洲，但直到 1202 年費波納契（Fibonacci）的著作《算盤書》（*Liber Abaci*）問世前，效果並不顯著。歐洲普遍接受十進位算術被是數百年後的事情（或許有些太遲）。吉爾伯特後來被加冕成為教宗席維斯特二世（Pope Sylvester II），他身處的時代局勢混亂，在庇護者奧圖三世（Otto III）駕崩後不久，他也於 1003 年撒手人寰。想像如果他能活久一點，歷史將會如何改寫 ❷，其實饒富趣味。圖 1 是他的數學著作《幾何學》（*De Geometria*）中的一頁。

牛頓的年代

有了十進位系統，再加上剛發現的微積分方法所

圖 1 教宗席維斯特二世（999-1003 在位）著作《幾何學》中的一頁。

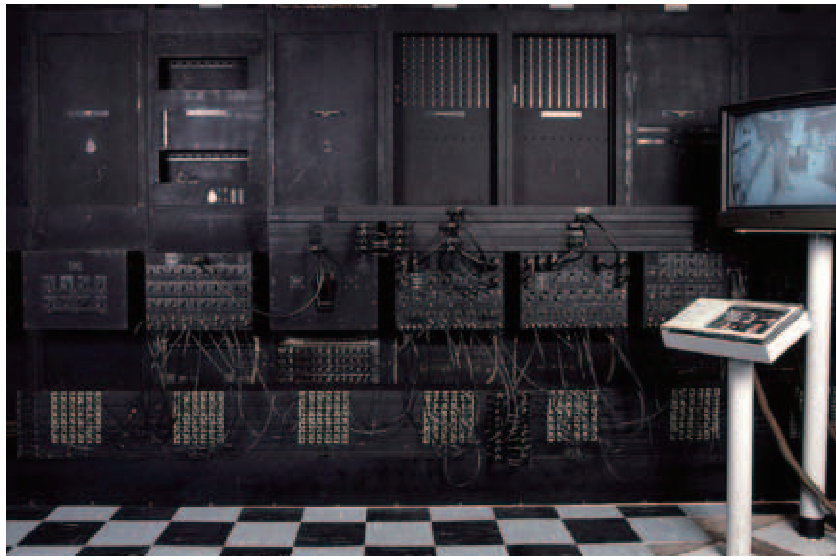


圖 2 史密森尼（Smithsonian）學會美國歷史博物館中的 ENIAC。（Division of Medicine & Science, National Museum of American History, Smithsonian Institution）

帶來的激勵，數學家對計算益發游刃有餘。π 的數值再次成為矚目焦點。牛頓發展出一套使用類反正弦函數的算法來計算 π 值，計算到小數點下 15 位。不過他語帶尷尬的坦承：「我不好意思告訴大家，在我沒其他事情可做時，為了這幾個數字做了多少計算。」牛頓是在發生瘟疫的 1666 年寫下這段話的，當時他在鄉下莊園避難，並發展出基本微積分、三大運動定律以及萬有引力理論。

直到 1980 年之前，π 的計算都仰賴馬欽（Machin）公式或其各種變形：

$$\frac{\pi}{4}=4\tan^{-1}\left(\frac{1}{5}\right)-\tan^{-1}\left(\frac{1}{239}\right)$$

諸如此類的成果，在 1853 年善克斯（William Shanks）利用上式計算出 π 的小數點下 527 位時來到顛峰，後來又（錯誤的）擴增到 707 位。在出版這項計算的序言中，善克斯說這項工作「對作為數學家的名聲幾無幫助，但或許對作為計算員（Computer）有所裨益。」（1950 年之前，computer 指稱人，而 calculator 則指稱機器。）

做這些計算的目的之一，是要確認 π 的小數表示是否循環，進而確認 π 是不是有理數。這項爭論在 1761 年蘭伯特（Johann Lambert）證明 π 是無理數後得以終結，因此 π 不論是哪種進位的小數表示都不會循環。1882 年，林德曼（Ferdinand Lindemann）更證明 π 是超越數 [3]，因此諸如 π^2 或 π 的整係數多項式，它們的小數表示都也不會循環。由此完整解決了古希臘三大數學問題之一的「化圓為方」問題 [4]。答案是做不到，因為有限次尺規作圖能做出的數必然是代數數。

電腦的年代

在電腦誕生的早期，馮諾曼（John von Neumann）已建議用電腦來計算重要的數學常數如 π 和 e 的值，並研究其統計分析上的性質。在他的倡議下，π 值在 1949 年已經計算到小數點下 2037 位，使用的是**電子數值整合計算器**（Electronic Numerical Integrator and Calculator，ENIAC，見圖 2）。1965 年，數學家發現新發展的快速傅立葉變換（fast Fourier transform）能加速高精度乘法（high-precision multiplication）演算法。這不只

❶ 譯註：讀者留意本文的自然對數都用 log 表示，而非另一種常用的 ln。

❷ 譯註：西元 946~1003 年，享年 57 歲。

❸ 譯註：整係數多項式的根稱為代數數（algebraic number），非代數數的數稱為超越數（transcendental number）。

❹ 譯註：古希臘三大問題分別為以尺規作圖解決「倍立方體」、「化圓為方」、「三等分任意角」。現在都已被證明為不可能。

表 1　計算 π 的現代簡史

姓名	時間	十進位小數位數
三好和則、金田康正	1981 年	2,000,036
金田康正、吉田紗也佳、田村家晃	1982 年	16,777,206
賈斯柏	1985 年	17,526,200
貝利	1986 年 1 月	29,360,111
金田康正 / 田村家晃	1986 年 9 月	33,554,414
金田康正 / 田村家晃	1986 年 10 月	67,108,839
金田康正等人	1987 年 1 月	134,217,700
金田康正 / 田村家晃	1988 年 1 月	201,326,551
楚德諾夫斯基兄弟	1989 年 5 月	480,000,000
金田康正 / 田村家晃	1989 年 7 月	536,870,898
金田康正 / 田村家晃	1989 年 11 月	1,073,741,799
楚德諾夫斯基兄弟	1991 年 8 月	2,260,000,000
楚德諾夫斯基兄弟	1994 年 5 月	4,044,000,000
金田康正 / 高橋大介	1995 年 10 月	6,442,450,938
金田康正 / 高橋大介	1997 年 7 月	51,539,600,000
金田康正／高橋大介	1999 年 9 月	206,158,430,000
金田康正／後保範／黑田久保	2002 年 12 月	1,241,100,000,000
高橋大介	2009 年 1 月	1,649,000,000,000
高橋大介	2009 年 4 月	2,576,980,377,524
貝拉	2009 年 12 月	2,699,999,990,000
近藤茂／余智恒	2010 年 10 月	5,000,000,000,000

編按：在 2016 年 6 月，π 值計算的世界紀錄，目前是 "houkouonchi" 的 13,300,000,000,000 位。從 2009 年之後，現在的計算（尤其是余智恆發表 y-cruncher 之後）很多都是使用增強版的家用電腦，也出現許多匿名的網路算手。

另附上表中部分人名之原名：賈斯柏（William Gosper）、楚德諾夫斯基兄弟（Chudnovskys）、貝拉（Fabrice Bellard）。

表 2　其他數學常數的位數紀錄

常數	十進位小數位數	姓名	時間
$\sqrt{2}$	10,000,000,000,000	華特金斯	2016 年 6 月
ϕ	2,000,000,000,000	華特金斯 / 柯克蘭	2015 年 7 月
e	1,500,000,000,000	華特金斯	2016 年 2 月
$\log 2$	500,000,000,000	華特金斯	2016 年 4 月
$\log 10$	500,000,000,000	華特金斯	2016 年 4 月
$\zeta(3)$	400,000,000,000	納格	2011 年
G	250,000,000,000	華特金斯	2016 年 4 月
γ	250,000,000,000	華特金斯	2016 年 5 月

編按：本表是 2016 年 6 月時的最新紀錄。本文發表時，世界紀錄都是近藤茂、余智恒、陳漢夫所建立的。但從去年 5 月起，華特金斯異軍突起，目前幾乎橫掃所有紀錄。

附上表中人名之原名：華特金斯（Ron Watkins）、納格（Dipanjan Nag）、柯克蘭（Dustin Kirkland）。

讓 π 和其他數學常數的高精度計算受惠，對計算數論的研究也助益甚大。

1976 年，薩拉敏（Eugene Salamin）和布蘭特（Richard Brent）獨立發現計算基礎指數與三角函數的新方法（可進而計算諸如 π 或 e 等常數），比使用傳統級數展式的方式快上許多。他們的算略（scheme）是以橢圓積分以及高斯算幾平均遞迴式（Gauss arithmetic-geometric mean iteration）為基礎，每次遞迴後能讓正確位數翻倍。藉由這種方法，在 1973 年時 π 值已經算到百萬位，1989 年超過十億位，2002 年超過一兆位，而到本文撰寫的 2013 年 6 月為止，已超過五兆位（見表 1）。

同樣的，其他常數像是 e、黃金比例 ϕ、$\sqrt{2}$、$\log 2$、$\log 10$、黎曼 ζ 函數值 $\zeta(3)$、卡塔蘭常數 $G=\sum_{n=0}^{\infty}(-1)^n/(2n+1)^2$、歐拉常數 γ，都已算到小數點下令人難以想像的位數，參見表 2。[10]

有趣的是，歷史上每當計算的位數取得進展，往往就會有見識廣博的人士跳出來，宣稱未來的研究將不再能有太大的起色。例如不久前的 1963 年，自己將 π 計算到小數點下十萬位的丹尼爾·善克斯（Daniel Shanks），告訴戴維斯（Philip Davis）說想要計算到十億位「門都沒有」。然而，這項任務在不到 30 年的 1989 年，便被日本人金田康正（Kanada Yasumasa）達成。就在 1989 同一年，著名英國物理學家潘洛斯（Roger Penrose）在他的暢銷書《皇帝新腦》（*The Emperor's New Mind*）首版中宣稱，人類應該永遠無法知道 π 的十進位表示中是否會連續出現 10 個 7。結果，這串數字在八年後的 1997 年就被金田找到了，出現在小數點下第 22869046249 位開始。經本文作者之一提醒後，潘洛斯才在第二版中把敘述改為連續 20 個 7。

在這個方向上，數學直覺主義倡導者布勞爾（Luitzen E. J. Brouwer）與赫汀（Arend Heyting）

曾經為了說明某些猜測將永遠無法被形式證明，舉了 π 的十進位表示是否存在 0123456789 為例。然而金田已在第 17387594880 位找到它 [5]。另外，天文學家薩根（Carl Sagan）1985 年出版的小說《接觸未來》（*Contact*，電影版由朱蒂福斯特主演）中，主角還在確認 π 的 11 進位表示是否有特別模式。結果書出版後，薩根很訝異 π 已經被計算到小數點下好幾百萬位。

π 的 BBP 公式

1997 年，由本文作者之一（貝利）、彼得・波宛（Peter Borwein）[6] 和普洛夫（Simon Plouffe）合撰的論文中（參見 [3]），給出下列計算 π 值的新公式，現在稱為 BBP 公式：

（1）

$$\pi=\sum_{k=0}^{\infty}\frac{1}{16^k}\left(\frac{4}{8k+1}-\frac{2}{8k+4}-\frac{1}{8k+5}-\frac{1}{8k+6}\right)$$

這個公式有個絕妙的性質，讓人可以直接從任意位數開始計算 π 的二進位或 16 進位數字，完全不需算出任何前面位數的數值。由此得到的演算法只需要最小限度的記憶體、不需使用多精度計算（multiple-precision arithmetic），非常適合高度平行計算。而且隨著計算起點的位數往後延，這項算略的計算成本卻僅些微上升。

這道公式的證明意外的簡單。首先注意到對所有 $k<8$，

$$\int_0^{1/\sqrt{2}}\frac{x^{k-1}}{1-x^8}\,\mathrm{d}x=\int_0^{1/\sqrt{2}}\sum_{i=0}^{\infty}x^{k-1+8i}\,\mathrm{d}x=\frac{1}{2^{k/2}}\sum_{i=0}^{\infty}\frac{1}{16^i(8i+k)}$$

因此，可以將

$$\sum_{i=0}^{\infty}\frac{1}{16^i}\left(\frac{4}{i+1}-\frac{2}{8i+4}-\frac{1}{8i+5}-\frac{1}{8i+6}\right)=\int_0^{1/\sqrt{2}}\frac{4\sqrt{2}-8x^3-4\sqrt{2}x^4-8x^5}{1-x^8}\,\mathrm{d}x$$

以 $y=\sqrt{2}x$ 代換後，利用部分分式分解改寫為

$$\int_0^1\frac{16y-16}{y^4-2y^3+4y-4}\,\mathrm{d}y=\int_0^1\frac{4y}{y^2-2}\,\mathrm{d}y-\int_0^1\frac{4y-8}{y^2-2y+2}\,\mathrm{d}y=\pi$$

1997 年時，Maple 和 Mathematica 都還沒辦法透過直接運用（1）去計算 π 值，但現在這兩套軟體都能輕鬆做到了。

$\log 2$ 的二進位表示

值得注意的是，BBP 公式（1）並非用傳統分析方法推導的，而是使用數學家兼雕刻家佛格森（Helaman Ferguson）所發展的 PSLQ 整線性關係演算法（參考後面「尋找 π 的公式」一節），透過電腦計算搜尋發現的，有人甚至把箇中過程描述成「逆向數學工程」（reverse mathematical engineering）的習題。這項研究緣起於 [3] 的作者觀察到，$\log 2$ 具有能從任意位數開始計算二進位數字的性質，這可從分析下列經典公式得到：

[5] 編註：這個例子似乎不能作為反例，因為他們的重點在形式證明或判定，而非暴力計算。

[6] 編註：Peter Borwein 是本文作者之一 Jonathan Borwein 的弟弟，後文還有一位 David Borwein 則是他們的父親。

$$\log 2 = \sum_{k=1}^{\infty} \frac{1}{k\,2^k}$$

這道公式至少在歐拉的時代就已經出現，並且和二次對數（dilogarithm）常數的函數方程關係密切[7]。

令 $r \bmod 1$ 為非負實數 r 的小數部分，d 為非負整數。那麼在把「小數點」右移 d 位後，log 2 以二進位展開的「小數」部分就能寫成：

$$\begin{aligned} &2^d \log 2 \bmod 1 \\ &= \left(\sum_{k=1}^{d} \frac{2^{d-k}}{k} \bmod 1 + \sum_{k=d+1}^{\infty} \frac{2^{d-k}}{k} \bmod 1 \right) \bmod 1 \\ &= \left(\sum_{k=1}^{d} \frac{2^{d-k} \bmod k}{k} \bmod 1 \right. \\ &\qquad\qquad \left. + \sum_{k=d+1}^{\infty} \frac{2^{d-k}}{k} \bmod 1 \right) \bmod 1 \end{aligned}$$

其中最後式第一項的分子之所以要安插「mod k」，是因為我們只在乎除完結果的小數部分。

透過二進位乘方演算法（binary algorithm for exponentiation），我們能很快地計算出$2^{d-k} \bmod k$。舉例來說，不需要花到 16 步，我們只需要五步就能計算出 $3^{17} = (((((3^2)^2)^2)^2) \cdot 3$。我們能透過在每次乘法後取對$k$的餘數來進一步減少計算成本。這項演算法和第一項中的除法、加法一般可以在雙倍位精度浮點運算（double-precision floating-point arithmetic）下運算，或者在更大型的計算，則使用四倍位精度或八倍位精度的浮點運算。

將最後的結果以二進位表示出來，便是 log 2 從第$d+1$ 位開始的二進位展式。計算的結果可以簡單地透過分別計算從靠近的位數，如第 $d-1$ 和 d 位開始的值，並比對結果是否重合來確認。

尋找 π 的公式

受到上述求 log 2 二進位展式的方法啟發，[3] 的作者立刻猜想對 π 是否也有類似的公式（那時沒人知道結果）。他們的方法是蒐集文獻上具有和 log 2 相似公式的數學常數 (α_i)，接著透過 PSLQ 整線性關係演算法確認是否有如下列形式的關係式：

$$\alpha_0\pi + a_1\alpha_1 + a_2\alpha_2 + \cdots + a_n\alpha_n = 0$$

其中 a_i 都是整數（如此一來便能解出 π 的公式）。在數個月的嘗試錯誤後，[3] 的作者發現了下列關係：π

$$4 \cdot {}_2F_1\left(\begin{matrix}1, \frac{1}{4} \\ \frac{5}{4}\end{matrix} \middle| -\frac{1}{4}\right) + 2\tan^{-1}\left(\tfrac{1}{2}\right) - \log 5$$

其中第一項為高斯超幾何函數。以取和的方式重新改寫上式後，π 的 BBP 公式便無所遁形：

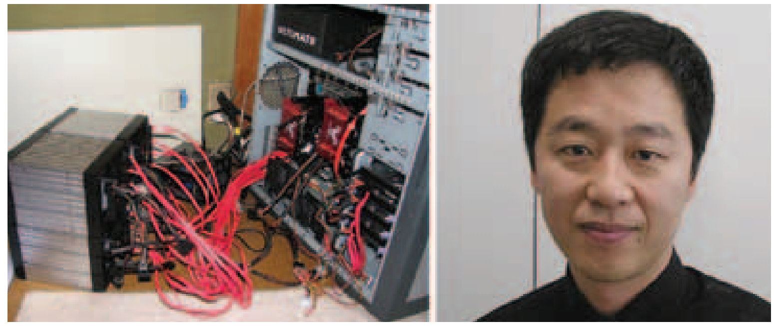
$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

在發現 π 的 BBP 公式後，很自然的問題就是有無類似公式但以其他數為底——換言之，在 BBP 公式中 16 的角色，可否換成其他數字如 3 或 10 而得到類似公式？在 2004 年，本文作者之一的波宛、高衛（Will Galway）以及大衛．波宛（David Borwein）證明除非以 2 的次方展開，否則不存在 π 的一次馬欽形式 BBP 公式（Degree-1 BBP-type formulas of Machin-type for π），也讓電腦的搜尋暫告段落 [5, pp.131-133]。

實地使用 BBP 公式

BBP 公式及其變形被廣泛運用於求 π 展式中，

圖 3 近藤茂和他的 π 計算器。



小數點下極高位的數字。1998 年，一位當時僅 17 歲，就讀於加拿大西門佛瑞瑟大學（Simon Fraser University）的柏西佛（Colin Percival）計算了 π 從千兆位（10^{15}）開始的二進位數。在當時，這大概是史上規模最大的分散式運算（distributed computation）。過了幾年，2010 年 7 月，**雅虎雲端運算**的施子和（Tsz-Wo Sze）在 Apache Hadoop 叢集，花費了約 500 CPU- 年算出 16 進位展式中，從第 5×10^{14} 位開始（二進位的第 2000 兆位）的值為

0 E6C1294A ED40403F 56D2D764 026265BC
A98511D0 FCFFAA10 F4D28B1B B5392B8

在離現今更近的 2013 年，聖塔克拉拉大學的卡列斯（Ed Karrels）使用配備了 NVIDIA 顯示卡的系統，計算以 16 為底從千兆位開始的 26 個數字。結果是

8353CB3F7F0C9ACCFA9AA215F2

BBP 公式也被用來驗算其他方式所算出的 π 值。舉例來說，在 2010 年 8 月，硬體工程師近藤茂（Kondo Shigeru）和大學生軟體工程師余智恆（Alexander Yee）使用僅 18000 美金的自製裝置，計算 π 十進位展式中第五兆位的數字。他們發現自第五兆位起算，前面 30 位為

7497120374 4023826421 9484283852

近藤茂和余智恆（見圖 3 和圖 4）使用下列楚德諾夫斯基 / 拉曼努真（Chudnovsky-Ramanujan）級數：

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k (6k)!(13591409 + 545140134k)}{(3k)!(k!)^3 640320^{3k+3/2}}$$

他們使用一套準符號算略（quasi-symbolic scheme）的機制，避開了直接計算上述公式所需的全精準運算。

近藤茂和余智恆一開始以 16 進位計算結果，接著他們將尾端的數值和使用 BBP 公式計算出來的值進行比對。通過這步關鍵測試後，他們才將結果轉成十進位。整個計算過程花費了 90 天，其中包含 64 小時的 BBP 公式檢測及八天的位值基底轉換。注意 BBP 檢測花費的時間相較其他兩個步驟少了很多，計算成本也因此大幅降低。他們的研究內容可參照 [11]。

其他常數的 BBP 型公式

1997 年後，數學家們或是使用 PSLQ 演算法搜尋，或是使用傳統的分析手法，發現了許多其他數學常數的 BBP 型公式，諸如 π^2、$\log^2 2$、$\pi \log 2$、$\zeta(3)$、π^3、$\log^3 2$、$\pi^2 \log 2$、π^4、$\zeta(5)$、卡塔蘭常數。許多反正切函數及當 $2 < k < 22$ 時的對數函數值 $\log k$ 也有 BBP 型公式。不過 $\log 23$ 的 BBP 型公式尚未被發現。這些公式及其參考資料都可以在線上手冊 [1] 中找到。

❼ 編註：將 k 次對數函數（k-th polylogarithm function）定義為 $L_k(z) = \sum_{i=1}^{\infty} z^i / i^k$
此函數因自然對數滿足 $\log(z) = -L_1(1-z)$，故得名。已知
$\pi^2 = 36L_2(\frac{1}{2}) - 36L_2(\frac{1}{4}) - 12L_2(\frac{1}{8}) + 6L_2(\frac{1}{64})$
$\log^2 2 = 4L_2(\frac{1}{2}) - 6L_2(\frac{1}{4}) - 2L_2(\frac{1}{8}) + L_2(\frac{1}{64})$
因此 π^2、$\log^2 2$ 稱為二次對數型的常數，其中牽涉到不同二次對數函數的關係。

圖 4 余智恆和他的大象。余所發展的 y-cruncher 現在已經是以家用電腦做這類計算的利器。

有趣的是，儘管 π 只有二進位的公式，π^2 卻同時有二進位和三進位的公式：

(2)

$$\pi^2 = \frac{9}{8}\sum_{k=0}^{\infty}\frac{1}{64^k}\left(\frac{16}{(6k+1)^2}-\frac{24}{(6k+2)^2}-\frac{8}{(6k+3)^2}-\frac{6}{(6k+4)^2}+\frac{1}{(6k+5)^2}\right)$$

(3)

$$\pi^2 = \frac{2}{27}\sum_{k=0}^{\infty}\frac{1}{729^k}\left(\frac{243}{(12k+1)^2}-\frac{405}{(12k+2)^2}-\frac{81}{(12k+4)^2}-\frac{27}{(12k+5)^2}-\frac{72}{(12k+6)^2}-\frac{9}{(12k+7)^2}-\frac{9}{(12k+8)^2}-\frac{5}{(12k+10)^2}+\frac{1}{(12k+11)^2}\right)$$

公式（2）出現在 [3] 中，而公式（3）由博德赫斯（David Broadhurst）找出。π^3 和 $\zeta(3)$ 都有二進位的 BBP 型公式，但迄今尚未有人發現三進位的公式。

卡塔蘭常數

另外一個備受關注的數學常數是卡塔蘭（Eugéne C. Catalan）常數：

$$G=\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)^2}=0.91596559417722\ldots$$

這大抵是目前尚未確定是否為無理數和超越數（雖然大多數人認為是）的常數中，最基本的一個。注意它和 π^2 緊密相關：

$$\frac{\pi^2}{8}=\sum_{n=0}^{\infty}\frac{1}{(2n+1)^2}=1.2337005501362\ldots$$

以上兩道公式都可視為最簡單的狄立克雷 L 級數（Dirichlet L -Series）在 2 取值的情形。這正是本文著重計算這兩個常數的背景。

卡塔蘭常數已是眾多大型計算研究的目標。如同先前所提及的，2009 年余智恆和陳漢夫（Raymond Chan）計算 G 到小數點後 310.26 億位。他們使用了兩道公式，其中一道來自拉曼努真的公式：

$$G=\frac{3}{8}\sum_{n=0}^{\infty}\frac{1}{\binom{2n}{n}(2n+1)^2}+\frac{\pi}{8}\log(2+\sqrt{3})$$

這道式子來自下列性質：

$$G=-T(\pi/4)=-\frac{3}{2}\cdot T(\pi/12)$$

其中 $T(\theta)=\int_0^{\theta}\log\tan\sigma\,\mathrm{d}\sigma$。

BBP 手冊中列出了兩道卡塔蘭常數的 BBP 型公式。第一道由貝利透過數值方法發現；艾德郭克（Kunle Adegoke）隨後以博德赫斯的結果為基礎，導證了這兩道公式。

現在我們希望找出一道 G 的公式，其中非零項越少越好，因為計算 BBP 型公式所需的時間大致

圖 5 卡塔蘭（Eugene Catalan）是 19 世紀的法國 / 比利時數學家。（維基）。

隨著非零項的數目線性增長。BBP 手冊中的兩道公式分別有 22 和 18 項非零項係數。因此我們利用 PSLQ 演算法，在這兩組 G 係數和兩道「0 關係式」（zero relations，和為 0 的 BBP 型公式）所張出的線性空間中搜尋 G 的公式。這些分析和計算讓我們找到了下列的公式，它只有 16 項非零項，應當是目前計算卡塔蘭常數中最經濟的 BBP 公式：

(4)

$$
\begin{aligned}
G = \frac{1}{4096}\sum_{k=0}^{\infty}\Bigg(& \frac{36864}{(24k+2)^2} - \frac{30720}{(24k+3)^2} \\
& - \frac{30720}{(24k+4)^2} - \frac{6144}{(24k+6)^2} - \frac{1536}{(24k+7)^2} \\
& + \frac{2304}{(24k+9)^2} + \frac{2304}{(24k+10)^2} + \frac{768}{(24k+14)^2} \\
& + \frac{480}{(24k+15)^2} + \frac{384}{(24k+11)^2} + \frac{1536}{(24k+12)^2} \\
& + \frac{24}{(24k+19)^2} + \frac{120}{(24k+20)^2} - \frac{36}{(24k+21)^2} \\
& + \frac{48}{(24k+22)^2} - \frac{6}{(24k+23)^2}\Bigg)
\end{aligned}
$$

BBP 公式和正則性

表 3 π 前一兆位各 16 進位數碼出現次數記錄表

16 進位數碼	出現次數
0	62499881108
1	62500212206
2	62499924780
3	62500188844
4	62499807368
5	62500007205
6	62499925426
7	62499878794
8	62500216752
9	62500120671
A	62500266095
B	62499955595
C	62500188610
D	62499613666
E	62499875079
F	62499937801
總數	1000000000000

注意平均值應為 62,500,000,000，偏差在第六位後才發生，和預期相去不遠。

從古至今，計算並分析 π 和其他數學常數數值的主要動機之一，便是想回答下述古老的問題——是否以及為何這些數字是「隨機」出現的。目前許多由電腦執行的統計檢定無法從任何統計準則否定對 π 的上述猜想（不像 e）。讀者可以參閱表 3，其中呈現了金田康正所計算，π 小數點後一兆位的 16 進位數碼出現的次數。

令 b 為正整數，若實數 α 的 b 進位展開中，每個長度 m 的數串出現的期望頻率都收斂到 $1/b^m$，則 α 稱為 b 正則（b -normal）。根據基本機率論，幾乎所有實數對任何特定的 b 進位，甚至同時對所有進位，都是 b 正則。然而要證明數學家感興趣的特定常數具有正則性，卻超乎想像的困難。

2001 年，本文作者之一（貝利）和克蘭多（Richard Crandall）注意到，像 π、π^2、$\log 2$ 和 G 這類有 BBP 公式的常數的正則性，可以化簡成關於一類混沌遞迴式（chaotic iteration）行為的猜想 [5，141-173 頁]。目前沒人能夠證明一般的猜想，但即使能有一些特殊結果就已饒富趣味。舉例來說，如果能證明遞迴數列 $w_0 = 0$ 以及


圖 6 本文作者之一的麥廷理。


圖 7 本文作者之一的外特威克。

(5)

$$w_n = \left(2w_{n-1} + \frac{1}{n}\right) \bmod 1$$

在 $[0,1)$ 均匀分布（即為「好」偽隨機（pseudorandom）亂數生成器），那麼根據貝利與克蘭多的結果，$\log 2$ 便是 2 正則。循著相同的脈絡，若我們能證明遞迴數列 $x_0 = 0$ 以及

(6)

$$x_n = \left(16x_{n-1} + \frac{120n^2-89n+16}{512n^4-1024n^3+712n^2-206n+21}\right) \bmod 1$$

在 $[0,1)$ 均匀分布，則 π 就是 2 正則。

上述研究被底下最近的結果給增強了。這個結果是擴展上述想法，因此嚴格證明某一類不可數無窮多實數的正則性。令 r 為 $[0,1)$ 中任意實數，r_k 為其二進位的第 k 位數字。則以下實數是 2 正則：

$$\alpha_{2,3}(r) = \sum_{k=0}^{\infty} \frac{1}{3^k\, 2^{3^k+r_k}}$$

例如 $\alpha_{2,3}(0)$ 等於

$$\sum_{k=0}^{\infty} \frac{1}{3^k\, 2^{3^k}} = 0.541883680831502985\ldots$$

就被證為 2 正則。如果 2 和 3 被代換為任意一對大於一的互質整數 (b,c) ，類似的結果也成立 [5，141-173 頁]。

有意思的 16 進位猜想

有趣的是，若我們使用（6）計算 16 進位的數列

(7)

$$y_n = \lfloor 16x_n \rfloor$$

（其中 $\lfloor\cdot\rfloor$ 代表高斯函數），那麼數列 (y_n) 就分毫不差地（並非只有近似）是 π 的 16 進位展開。藉由精確的計算，我們確認了這個數列的前千萬位和 $\pi - 3$ 的 16 進位展開式的前千萬位**一模一樣**。這個計算並非易事，它需要大約 n^2 位元操作，而且不容易在平行電腦系統上執行。實際上，我們用 Maple 和（7）計算了 2,000,000 位，在筆記型電腦上花了 17.3 小時。再用 Mathematica 和改良過後的演算法計算 4,100,000 位，花了 46.5 小時。至於完整確認，我們使用 C++ 在一台 IBM Power 780 上共花了 433,192 秒（120.3 小時）（型號：9179-MHB；時脈速度：3.864 GHz）。所有的結果都和 http://www.experimentalmath.info 中軟體（Software）頁面中所儲存，π 的 16 進位數字做對照與確認。

猜想 1

根據遞迴式（6）所定義的數列 (x_n)，則數列 $\lfloor 16x_n \rfloor$ 恰好生成 $\pi - 3$ 的 16 進位展式。

事實上，我們還能學得更多。令

$$\|x-y\| = \min(|x-y|, 1-|x-y|)$$

代表兩個 $[0,1]$ 間實數 x 和 y 的「環狀」距離。π 的 16 進位展式（以 π_n 表示）滿足 $\|\pi_n - x_n\|$ 不會大於下式：

$$\sum_{k=n+1}^{\infty} \frac{120k^2-89k+16}{16^{k-n}(512k^4-1024k^3+712k^2-206k+21)} \approx \frac{1}{64(n+1)^2}$$

因此將此式從某個 N 加至無限，我們有

圖 8 藍色基因 /P。不過本圖取自美國阿貢國家實驗室，而非文中的羅徹斯特基準測試中心。（維基）

（8）

$$\sum_{k=N}^{\infty} \|\pi_n - x_n\| \leq \frac{1}{64(N+1)}$$

以一種啟發式的角度來思考，我們假設 π_n 是各自獨立且在 $(0,1)$ 均勻分布的隨機變數，並令 $\delta_n = \|\pi_n - x_n\|$。注意只有當 π_n 離 0、$1/16$、$2/16$、$\cdots$、$15/16$ 中任一點的距離小於 δ_n 時，偏差（亦即 x_n 和 π_n 落在不同子區間，使得對應的 16 進位數字不一致）才會發生。因為對所有 n，$x_n < \pi_n$（當 x_n 很靠近 1 時，$<$ 應該用環狀距離的概念解釋），上述事件發生的機率為 $16\delta_n$。是以，因為在（8）中之總和有限，於是由第一波瑞耳 / 康塔利引理（Borel-Cantelli Lemma）知，只可能存在有限多個偏差。此外，即使取 $N = 1$，（8）中的總和也很小，這表示很不可能觀測到偏差。如果我們取 $N = 10,000,001$（因為我們知道前 10,000,000 位是一致的），我們得到上限為 1.563×10^{-9}，表示偏差真的不大可能發生。

在式（5）中的遞迴值和 log 2 的二進位展式也有類似的對應。特別令 $z_n = \lfloor 2w_n \rfloor$，其中 w_n 如（5）定義。因為對應到（8）的 log 2 偏差項總和發散，由第二波瑞耳 / 康塔利引理，可預期 (z_n) 和 log 2 二進位展式間的不一致將會出現無限多次，但頻率會隨位數增加遞減。事實上，在我們完成的計算裡，可發現 (z_n) 和 log 2 二進位展式在前 20 位有 10 處不同，但在第 5,000 至 8,000 位間的數字只有一處不同。

計算 π^2 與卡塔蘭常數

作為上述理論的實例，我們現在給出分別由（2）、（3）、（4）得到之 π^2 的二進位與三進位展式，以及卡塔蘭常數的二進位展式的計算結果（圖 6 和圖 7）。這個計算是在 IBM 明尼蘇達州羅徹斯特（Rochester）基準測試中心（Benchmarking Center）裡的四機架式「藍色基因 /P」（BlueGene/P）系統進行的（圖 8）。由於這是共享資源，計算長達數個月，有時我們可以用到部分甚至全部資

源，但有時完全無法運用。由於我們是從澳洲遠端執行程式，因此往往能在離峰時段使用系統。時區不同有時還是挺有助益的！

- **從第十兆位開始的 π^2 64 進位展式**：第一輪結果從第 $10^{13}-1$ 位開始，我們將計算分割成七段，每段有 2048 個執行緒（thread），每一執行緒上平均花費 253,529 秒，所以最後花費的 CPU-秒數為 $7\cdot 2048\cdot 253529=3.6\times 10^9$

  IBM 系統中每機架配備 4096 個核心，因此總花費為 10.3 **機架-天**（rack-days）。第二輪計算從第 10^{13} 位開始，執行時間差不多（只差數分鐘）。兩次計算的結果以八進位列於表 4 的 A（每兩個八進位數字對應一個 64 進位數字）。我們以 | 記號分出一致的位數。注意 53 個連續的八進位數字（對應到 159 個連續的二進位數字）完美重合。

- **從第十兆位開始的 π^2 729 進位展式**：計算分割方式和前項類似，兩輪各自在執行緒上花費平均 795,773 秒。因此總花費是 6.5×10^9 CPU-秒，或 18.4 機架-天。兩輪計算的結果以九進位列於表 4 的 B（每三個連續九進位數字對應到一個 729 進位數字）。注意這次有連續 47 個九進位數字（連續 94 個三進位數字）重合。

- **從第十兆位開始的卡塔蘭常數 4096 進位展式**：兩輪在執行緒上平均花費 707,857 秒，但這次我們將計算分割成八段，每段有 2048 個執行緒。因此總花費為 1.2×10^{10} CPU- 秒，或 32.8 **機架-天**。結果以八進位列於表 4 的 C（每四個八進位數字對應到一個 4096 進位數字）。有連續 47 個八進位數字（連續 141 個二進位數字）重合。

這些從目標位數開始連續重合的長串數值，讓我們對結果在統計上具有相當高的信心。例如第一種情況，隨意挑選出的 32 對八進位數字完美重合的機率大約是 1.2×10^{-29}。即使考慮數值上的計算誤差，丟棄最後六對 262,144 分之一的機率作為統計上的保險，仍然有 24 對完美重合的數字，機率是 2.1×10^{-22}。嚴格來說，我們無法在 π^2 展式的數字定義機率測度。不過從實用的眼光來看，這樣的分析已經提供很高的信心，可以認為計算結果是正確的。

因此，計算 π 及類似常數，成為電腦系統硬體和軟體整合性測試的好方法。要是在兩輪計算 π 的過程中，在某一輪發生即使是一次閃失，通常結果比較起來就會相去甚遠。舉例來說，1986 年的一組 π 的計算，就揭露了超級電腦 Cray-2 早期模型中一些細微卻嚴重的錯誤。事實上，我們這次的計算大抵也是藍色基因 /P 系統所歷經最強的整合性測試。表 5 記錄了上述三次計算的規模，其中使花費了超過 135 機架-天、連續 1378 CPU- 年，超過 1.549×10^{19} 個浮點運算。這和目前 2011 年最複雜的動畫電影耗費相當。

表 4 π^2 和卡塔蘭常數的十兆位展開

<table>
<tr><td>A</td><td>75|60114505303236475724500005743262754530363052416350634|573227604</td></tr>
<tr><td></td><td>|60114505303236475724500005743262754530363052416350634|220210566</td></tr>
<tr><td>B</td><td>001|12264485064548583177111135210162856048323453468|10565567635862
|12264485064548583177111135210162856048323453468|04744867134524</td></tr>
<tr><td>C</td><td>0176|34705053774777051122613371620125257327217324522|6000177545727
|34705053774777051122613371620125257327217324522|57035105166602</td></tr>
</table>

（A）π^2 的 8 進位展開。（B）π^2 的 9 進位展開。（C）卡塔蘭常數的 8 進位展開，均從小數點後第十兆位開始。

表 5 計算的規模

常數	n'	d	疊代次數（$\times 10^{15}$）	疊代時間（微秒）	時間（年）	連同驗算（年）	總時間（年）	額外負擔（%）	精度浮點運算量（$\times 10^{18}$）
π^2 2^6 進位	5	10^{13}	2.16	1.424	97.43	194.87	230.35	18.2	2.58
π^2 3^6 進位	9	10^{13}	3.89	1.424	175.38	350.76	413.16	17.8	4.65
G 4^6 進位	16	10^{13}	6.91	1.424	311.79	623.58	735.02	17.9	8.26

表 5 我們估計每次遞迴平均將有 4.5 個四精度浮點運算，約相同於 266 個單精度浮點運算。最後一行紀錄總共所花費的單精度浮點運算量，其中包含因採用位元遮罩（bit-masking）產生許多捨入運算（rounding operation）的額外負擔（overhead）。

表 6 卡塔蘭常數的數碼分布

數碼	0	1	2	3	4	5	6	7
二進位（141）	0.454	0.546	-	-	-	-	-	-
四進位（70）	0.171	0.329	0.229	0.271	-	-	-	-
八進位（47）	0.085	0.128	0.213	0.128	0.064	0.128	0.043	0.213

從第十兆位開始，卡塔蘭常數依不同進位所得的數碼次數頻率分布。

為了完整性，表 6 分別呈現了在卡塔蘭常數的計算中，一、二、三位元數碼出現的頻率分布。

未來方向

說來諷刺，在這個時代，數學難題如費馬最後定理與龐卡赫猜想均已臣服於現代數學的慧智，但是一項最基本的數學假說——π 或其他常數如 $\log 2$、π^2、G 的展式，其數字之出現是否（及為何）「隨機」，仍舊等不到解答。特別是證明 π（或 $\log 2$、π^2、G）是否對某整數 b 之進位具有 b 正則性，前路仍舊曖昧不明。即使是更弱的結果，像是在 π（或 $\log 2$、π^2、G）的二進位展式中，1 的出現機率是否趨近 $1/2$（許多計算都支持這點）仍舊缺乏證明，現在看來離目標還遠得很。

對於簡單的無理代數數如 $\sqrt{2}$，我們也所知甚少。在這個情況，目前已知 $\sqrt{2}$ 二進位展式的前 n 位中，1 的數目至少佔了 $\sqrt{n}$ 個左右，而其他無理代數數也有類似結果 [5，141-173 頁]。但這是相當弱的結果，因為對於 $\sqrt{2}$ 和其他代數無理數，這個極限的比率幾乎確定是 $1/2$。

我們對許多常數的連分數展開知道的也不多，只有一些特殊情形廣為人知，例如二次無理數 8，貝塞爾函數（Bessel function）的比值，以及一些與指數函數有關的表式。

因為如此，大家才會對 BBP 型常數的興趣源源

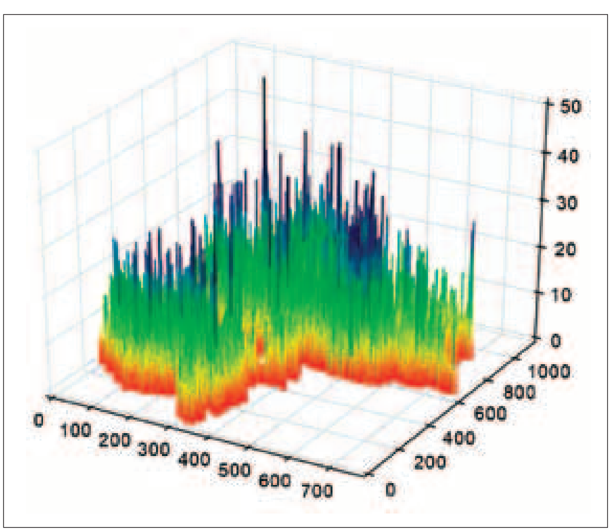


圖 9 利用卡塔蘭常數展式的 100 萬位所建構的「隨機」漫步圖。

8 譯註：形如 $\frac{b+c\sqrt{d}}{a}$，其中 a、b、c、d 為整數，d 不為完全平方數。

圖 10 本文作者之一的貝利。（維基）

圖 11 本文作者之一的波宛。（Australian Academy of Science/Mark Graham）

不絕。先前提過，BBP 型常數和特定的混沌遞迴有引人入勝的關聯，和偽隨機亂數產生器類似。假設我們能強化這種關聯，或許就能證明多次對數函數型常數（polylogarithmic constant）的正則性，其中可能包含 π、$\log 2$、π^2 和 G。

當理論的框架改變，問題自然也跟著改變。直到有人問說能否有效摘取展開式的任一位數後，我們方才發現自己對這個問題一無所知。底下的指數函數展式是個很有啟發性的例子。

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

由於就現在的目的來看，此式的收斂速度實在太快太好了，我們猜測：

猜想 2

e 沒有 BBP 型公式。此外，除了直接計算前面的展式數字外，沒有顯著更快的方法能直接取出展開式中的某一項。

我們猜想其他常數如歐拉常數 $\gamma = 0.57721566490153\ldots$，也有類似的性質。簡而言之，除非在這領域出現強而有力的結論，不然人們無疑還是會持續有興趣計算這些常數。在這段理論的真空期，這或許是我們唯一能做的。∞

本文參考資料請見〈數理人文資料網頁〉http://yaucenter.nctu.edu.tw/periodical.php

本文出處

Notices 60（2013）No.7, AMS。

譯者簡介

林真是臺大數學系四年級的學生。

延伸閱讀

▶《實驗數學》網站（*Experimental Mathematics Website*）。這是本文作者貝利和波宛主持的網站，目的在提供關於實驗數學與電腦輔助數學的各種資訊。

http://www.experimentalmath.info

他們兩人各自還擁有資訊豐富的個人網站：

貝利：http://www.davidhbailey.com

波宛：https://www.carma.newcastle.edu.au/jon/

▶余智恆的《數世界》網站。想要鑽研數學常數計算的現況與工具，這是必訪網站。

http://www.numberworld.org

▶ Beckmann, Petr, *A History of Pi*（1971）Golem Press。中譯本《π 的故事》（2000）姜家齊、朱建正、林聰源譯，凡異出版社。

▶ Blatner, David, *The Joy of Pi*（1997）Walker Books。中譯本《神奇的 π》（2007），商周。前書有比較多的歷史，本書編排前衛，有比較多的計算相關題材。