專欄 省立教育學院 科學教育系 數學組

速能補拙

林濟卿

有人給電子計算機起了一個封號「偉大的傻瓜」，堪稱爲頗有哲理的妙稱。因爲它只認識兩個字「是」與「不」，也就是「0」與「1」；它只會加法，其他演算都是祇若加法的延伸，如減法是加法的反算，乘法除法則分別是加法減法的反覆而已。這種傻瓜式的反覆計算我們人類不但不屑於一顧也是不能忍受的，可是計算機另有一招絕技「快」，快得令人難以想像。它一秒鐘可做（如5位數）加法幾百萬次，幾十億次。如使用5組0～9數字所作成的數字組合鎖就有 10^5 次不同的組合法，若每一秒鐘試一種也要10萬秒，一天有86400秒，幾乎要整整28個小時不眠不休的試才有一定可以打開的把握。當然人是不會這麼傻的，定會想盡辦法如投機、靈感、技巧等等，總不肯老老實實的從00000、00001、……、99990、……、99999的嘗試錯誤。刁滑如電視上的妙賊不消幾秒鐘便可打開，普通人嘛，搞個幾十分鐘或幾點鐘，管它開不開也就算了，可是我們的“傻瓜”計算機就不然，它絕不投機也不取巧更不氣餒，硬是按步就班的一一嘗試，祇要一切正常百分之百可以打開。所費時間，假定試一次等於作加法一次（它一秒可作十億次加法），則等於 $10^5\times10^{-9}=10^{-4}$ 即萬分之一秒，我們那會說它傻？祇有目瞪口呆感嘆不已的份而已！這種神速加上「不會錯」「不會忘」就使計算機變成爲一個可聖可魔的怪物。我們大可不必過份羡慕它的不會錯、不會忘，因爲不會錯不會忘所以敵我不分是非不明，祇要操作合法，密碼符號誰都可以指揮它。不過若能善加以鞭策使用，它確是我們最大的幫手。

爲了要發揮電子計算機「快」和「百作不厭」的特性，我們就必須在可能範圍內遷就它或改變我們的思考路徑。而這些思考的路徑，若以常理來判斷，常常是很單純、很基本，甚至會給人“笨拙”的印象。關於這些，我們在下面舉幾個實例來加以說明。

例1. 解 $f(x)=0$ 的數值根

如解 $f(x)=x^2-2=0$，即求 $\sqrt{2}$ 之值，很快便可筆算求出。但相信很少人會想到或願意先用一個數如2代進去試算

$2^2=4>2$

所以折半減少 $(0+2)/2=1$

$1^2=1<2$

所以折半增加 $(2+1)/2=1.5$

$1.5^2=2.25>2$

所以折半減少 $(1+1.5)/2=1.25$

………

直至得到滿意的數值解爲止。因爲這種方法計算繁瑣，尤其當 $f(x)$ 複雜時更是缺乏實用性。可是有計算機便改觀了，計算反覆幾百次，幾千次

，並不算一回事立即可完成。此法稱爲「二分法」是一種最原始最簡便的實用法，是計算機以快制勝的典型例子。

例 2. 兩個不同的日期之間共有多少天？分別是星期幾？

袖珍計算機常附有表演用小程式，只要按入任意的甲、乙兩個日期（比如說你的出生年月日和當天的年月日）立即可顯出此兩天之間共有幾天，以及分別各爲星期幾。其實道理非常簡單，所要考慮的條件不過是：

(1) 4 、 400 的倍數年爲潤年，2 月有29天

(2) 100 的倍數年雖也是 4 的倍數年，但是平年，它的 2 月祇有28天

(3) 4 、 6 、 9 、11每月有30天；1 、 3 、 5 、 7 、 8 、10、12每月有31天。

我們很容易的會想到以甲日爲起點，乙日爲終點，再依上面的條件計算中間的日數。當然這也是可行的方法之一，不過要再計算星期幾就有麻煩很難兩事一氣呵成。我們何不轉個念頭，一律以公元元年元旦爲起點計算至甲日及乙日的天數，然後兩者再相減。至於星期幾可先推公元元年元旦的曜日，再求上面所得差距除以 7 的餘數便可。此一念之差足可使計算程式驟減一半。

設甲日 B＝ Y_1 年M 1 月D 1 日

A_1＝公元元年元旦到甲日的日數

乙日 T＝ Y_2 年M 2 月 D 2 日

A_2＝公元元年元旦到乙日的日數

則 $A1=365\times(Y_1-1)+N+L$

$N=Y_1$ 年的元旦到M1月 D 1日的日數

$$L=\mathrm{int}\left(\frac{Y_1}{4}\right)+\mathrm{int}\left(\frac{Y_1}{400}\right)-$$

$$\mathrm{int}\left(\frac{Y_1}{100}\right)=\text{要增加的潤日數}$$

不過還要判斷乙日之當年是否潤年，當月是否已超過 2 月，以便決定是否減一天潤日

$A=A_2-A_1$ 便是兩天之間的日數

若我們已推得公元元年元旦是星期一，則 A_1 除以 7 的餘數便是甲日的星期幾，（ 0 是星期日）。 A_2 同理可求。

說是簡單，眞要動筆計算可能還不是一件易事，而且很容易算錯。可是計算機却可瞬間完成，而且錯不了。

例 3. 倒數計算

$\frac{1}{n}$（ n 爲自然數）不是除盡便是循環小數。

我們感興趣的是分母爲質數的倒數，因爲其他的倒數均由它的乘積所構成。數學上有很多函數的展開級數都是用累乘倒數表示的。

如 $$e^x=1+\frac{1}{1}+\frac{x^2}{2\,!}+\cdots+\frac{x^n}{n\,!}+\cdots\cdots$$

$$\pi=16\left(\frac{1}{5}-\frac{1}{3\cdot5^3}+\frac{1}{5\cdot5^5}-\cdots\cdots\right)$$

$$-4\left(\frac{1}{239}-\frac{1}{3\cdot239^3}+\frac{1}{5\cdot239^5}-\cdots\cdots\right)$$

質倒數雖是無限小數，但祇要求得一循環節便可獲得任意位數的精度，故若能把質倒數的循環節預先求出保存，對於種種級數和的高精度計算定有不少幫助。

平常我們很少認眞動手去作這些繁瑣無味的倒數計算，其實確也是人手所不能勝任，因爲 $\frac{1}{n}$ 的循環節最高可達 $n-1$ 位，如 $\frac{1}{983}$ 就有 982 位的循環節。

普通都用除法求答

$$
\begin{array}{r}
0.142857 \\
7\overline{)\,10} \\
\underline{7} \\
30 \\
\underline{28} \\
20 \\
\underline{14} \\
60 \\
\underline{56} \\
40 \\
\underline{35} \\
50 \\
\underline{49} \\
1
\end{array}
\qquad \therefore \frac{1}{7} = .\dot{1}4285\dot{7}
$$

在此筆者願提出另一種計算方法，它的根據就是下面極為簡單的一個原理：

若 $\frac{1}{n} = a$，則 $na = .999\cdots\cdots$

例 1

$$
\begin{array}{rl}
\underline{\big|\,7} & \\
4\cancel{9} & 7 \\
\underline{35} & 5 \\
3\cancel{9} & 8 \\
\underline{56} & \\
5\cancel{9} & 2 \\
\underline{14} & \\
1\cancel{9} & 4 \\
\underline{28} & \\
2\cancel{9} & 1 \\
\underline{7} & \\
\cancel{9} &
\end{array}
\qquad
\begin{aligned}
&\therefore \frac{1}{7} = .142857 \\
&\because 1 = 7 \times .\dot{1}4285\dot{7} \\
&\qquad = .999999\cdots\cdots
\end{aligned}
$$

例 2

$$
\begin{array}{rl}
\underline{\big|\,13} & \\
3\cancel{9} & 3 \\
\underline{26} & 2 \\
2\cancel{9} & 9 \\
\underline{97} & \\
9\cancel{9} & 0 \\
\underline{0} & \\
\cancel{9} &
\end{array}
\qquad
\begin{aligned}
&\therefore \frac{1}{13} = .\dot{0}7692\dot{3} \\
&\because 1 = 13 \times .\dot{0}7692\dot{3} \\
&\qquad = .999999\cdots\cdots
\end{aligned}
$$

計算原則除第一次以外均把末位和湊成 9 再消去，直到不能計算為止，順序恰與普通除法相反，由最後一位逆算，故姑名之為「倒算法」。此法與普通法在筆算上並不能佔便宜，但無除法減法計算所以算起來總覺輕鬆些。這點對計算機來說也是一個優點，原來計算機最擅長的也是加法和乘法。使用計算機計算時，此法與普通的算法所需的時間在計算10位或 100 位的數字時尚無任何差別，但在計算萬位以上的數字時，則此法所需的時間會顯著地減少。

計算熟練後尚可把 9 省略不寫，同時把算式改直以便節省地方。

例 3

$$
\begin{array}{rl}
\underline{\big|\,11} & \\
9\cancel{9} & 9 \\
\underline{0} & 0 \\
\cancel{9} &
\end{array}
\qquad \therefore \frac{1}{11} = .\dot{0}\dot{9}
$$

例 4

$$
\begin{array}{rl}
\underline{\big|\,37} & \\
25\cancel{9} & 7 \\
7\cancel{4} & 2 \\
\cancel{9} & 0 \\
0 &
\end{array}
\qquad \therefore \frac{1}{37} = .\dot{0}2\dot{7}
$$

例 5

)31	18	19	7
279 .9	31 1	.0 0	62 2
62 2	4	1	6
3 1	155 5	248 8	93 3
31	15 4	24	9
3 6	124	155 5	0 0
186	13 6	17	
	186	62 2	

$$\therefore \frac{1}{31} = .\dot{0}3225806451612\dot{9}$$

再列幾個結果數請查驗

*1/13	RECUR 6	*1/23	RECUR 22
076923		0434782608	6956521739 13
*1/17	RECUR 16	*1/29	RECUR 28
0588235294	117647	0344827586	2068965517 24137931
*1/19	RECUR 18	*1/31	RECUR 15
0526315789	47368421	0322580645	16129

筆者曾應用此法使用PDP Ⅱ V03 微計算機（28KW）實際計算 e 及 π 值，分別正確至第235 位及第 232 位如下：

e： 2718281828 4590452353 6028747135 2662497757 2470936999 5957496696 7627724076 6303535475
9457138217 8525166427 4274663919 3200305992 1817413596 6290435729 0033429526 0595630738
1323286279 4349076323 3829880753 1952510190 1157383418 7930702154 0891499348 8416723866

π： 3141592653 5897932384 6264338327 9502884197 1693993751 0582097494 4592307816 4062862089
9862803482 5342117067 9821480865 1328230664 7093844609 5505822317 2535940812 8481117450
2841027019 3852110555 9644622948 9549303819 6442881097 5665933446 1284756482 3327526591

詳請見拙著「倒數計算及其應用」（教育學院學報第四期P.563～P.578）。

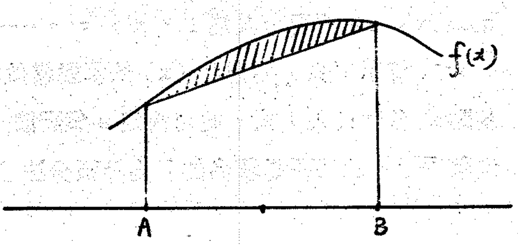
例 4. 積分的計算

積分的計算就是面積的計算，因曲線的面積難算就以直線的梯形面積近似是一種最自然的想法如圖㈠。

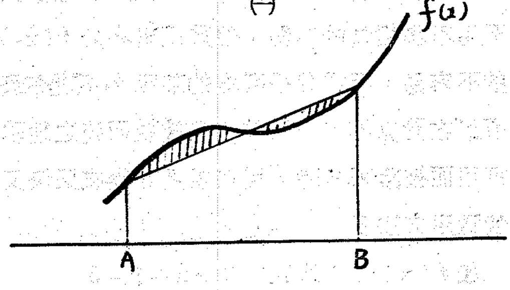
$$\int_A^B f(x)\,dx = \frac{1}{2}(f(A)+f(B))\,h$$

$$h = B - A$$

如果 $f(x)$ 成波形凹凸可相抵，效果更好如圖(二)。



(一)



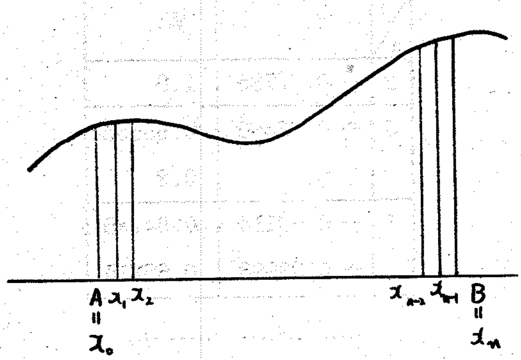
(二)

如果要提高精密度可細分 AB 增加梯形個數如圖(三)，分得越細就可期待越高的精密度，但這樣也就增加了須要計算的基點數如圖(三)

$$\int_A^B = \left(\frac{1}{2}\left(f(x_0) + f(x_n)\right)\right.$$

$$+ \sum_{i=1}^{n-1} f(x_i) h$$

$$h = \frac{B-A}{n}$$



(三)

用人手算不了多少基點，但是用計算機幾百、幾千基點也是輕而易舉的事，這就是大家所最能瞭解並接受的梯形法。

不過我們總希望要計算的基點少而精密度又高的妙法。這種似乎是要馬兒好又要馬兒不吃草的如意算盤，若能好好的利用增減相抵的觀念也並非不可能。

如 $f(x) = a_1 x + a_0$ 爲一次函數即一直線

則 $\int_A^B f(x)dx = f(M)h$

$$M = \frac{1}{2}(A+B) \qquad h = B - A$$

祇利用一個基點M就可以。

二次以上函數祇要基點取得巧，似乎也可達到類此增減相抵的效果，所得計算公式型式應如

$$\int_A^B f(x)dx = \sum_{i=1}^{n} W_i f(x_i)$$

n：基點數視 $f(x)$ 的次數而定

W_i：與基點 x_i 相配合的調節係數

爲著達到此目的，可把積分範圍定在〔-1，1〕間，同時規定具有下面特性的一系列特別函數 $P_n(x)$

$$\int_{-1}^{1} P_n(x) G_{n-1}(x) dx = 0$$

$P_n(x)$：次數等於 n 的特別函數

$G_{n-1}(x)$：次數 $\leq n-1$ 的任意函數

$$\because \int_{-1}^{1} 1 \cdot 0\, dx = 0 \qquad \therefore 設\ P_0(x) = 1$$

$$\because \int_{-1}^{1} 1 \cdot (x+a)\, dx = 0$$

解之得 $a = 0$ $\therefore$ 設 $P_1(x) = x$

$$\because \begin{cases} \int_{-1}^{1} 1 \cdot (x^2 + ax + b)\, dx = 0 \\ \int_{-1}^{1} x \cdot (x^2 + ax + b)\, dx = 0 \end{cases}$$

解之得 $\begin{cases} b=\dfrac{-1}{3} \\ a=0 \end{cases}$

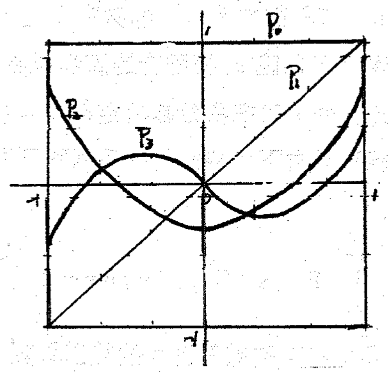
$\therefore$ 設 $P_2(x)=x^2-\frac{1}{3}$

………

此特別函數系稱爲Legendre 的直交系

k	$P_K(x)$
0	1
1	x
2	$x^2-\frac{1}{3}$
3	$x^3-\frac{3x}{5}$
4	$x^4-\frac{6x^2}{7}+\frac{3}{35}$

…………



次數≤ $2n-1$ 的任意多項式 $f_{2n-1}(x)$ 均可利用此直交系表現如下

$$f_{2n-1}(x)=q_{n-1}(x)+P_n(x)h_{n-1}(x)$$

$q_{n-1}(x)$，$h_{n-1}(x)$ 均爲次數≤ $n-1$ 的多項式

兩邊積分得 $\int_{-1}^{1} f_{2n-1}(x)\,dx$

$$=\int_{-1}^{1} q_{n-1}(x)\,dx$$

故可假設 $\int_{-1}^{1} q_{n-1}(x)\,dx=\sum_{i=1}^{n} W_i f(t_i)$

此式應適用於次數≤ $n-1$ 的任意多項式 $q_{n-1}(x)$，我們可令其爲 1，x，x^2，……，x^{n-1} 等單項式，而把 $P_n(x)$ 的零點當做基點 x_i 分別代入上式，便可決定 n 個調節係數 W_i。此法便是著名的「高斯積分法」。

n 個基點便可積分 $2n-1$ 次的任意多項式，不可不謂相當的高明，但其困難在於 $f(x_i)$ 計算的不容易，因爲幾乎所有的基點 x_i 都是無理數，所以在計算機未出世前也是被敬而遠之地視爲不實用而被冷藏多時，現在當然是變成又快又方便的實用方法了。

如 $f(x)$ 爲 3 次式 $\because 2n-1=3$

$\therefore n=2$ 即夠

則 $\int_{-1}^{1} f(x)\,dx=f(0.57735)+f(-0.57735)$

$f(x)$ 爲 5 次式 $\because 2n-1=5$

$\therefore n=3$ 即夠

則 $\int_{-1}^{1} f(x)\,dx=0.55556f(0.77460)+0.8f(0)+0.55556f(-0.77460)$

2 次與 3 次式可共用 $n=2$ 點，4 次與 5 次可共用 $n=3$ 點，……

n \ i	x_i	W_i
2	± 0.57735	1.0
3	± 0.77460	0.55556
	0.0	0.8
4	± 0.86114	0.34785
	± 0.33998	0.65215

………………………

例 5　Monte Carlo 法

要求單位圓的面積可解

$$x^2+y^2=1 \quad 得\ y=\sqrt{1-x^2}$$

再積分便可。

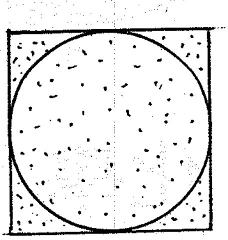
若要求 5 次元單位球的體積當然可解

$$x_1^2+x_2^2+x_3^2+x_4^2+x_5^2=1$$

再多重積分便可。

但何不轉個念頭而使用較爲憨直簡便的辦法謀求解決呢？

我們可把大小均一的細粒如粟粒等任意的撒到單位圓的外切正方形上去，若數得正方形上的細粒總數爲 100 而單位圓上的爲75，則單位圓的近似面積不就是 $2^2 \times \frac{75}{100}=3$ 嗎？



此法之精密度可能不甚高，但總比束手無策強得多。我們也可預料撒得越多越均勻所得的精密度越高，不過若想到細粒計數的麻煩可能又會放棄而另想他法了。但我們有不嫌麻煩的計算機在旁待命呀！祇要命令得宜，那怕是幾千粒，就是幾萬粒照算不誤而且快得很。

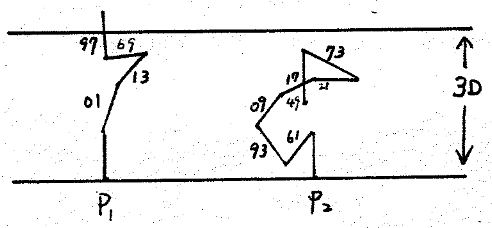
每一細粒之位置可用（ x ， y ）座標表示，若把正方形之中心定爲原點，則 $|x|$，$|y| \le 1$。每一細粒是否在圓內可由 $\sqrt{x^2+y^2} \le 1$ 來判斷。至於如何任意的撒細粒呢？這可借用隨週機數（亂數）解決。既然 x ， y 值的出現全靠偶然的機會，由其組合的（ x ， y ）也理該是靠運氣出現毫無做作才是。

此法簡單易行，且可直接推廣至 3 次元以上，可惜精密度不高。若要極爲準確的結果，還得依靠理論方法慢慢的推究。不過它總可提供大概的指針以資查驗，也提供了大原則、大方向使我們避免踏入歧途。以下我們再考慮另一例子。

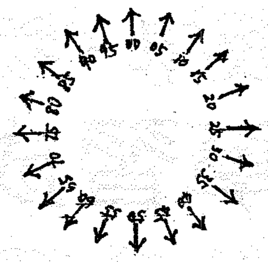
假定有中子流垂直衝擊厚度 3 D的反應爐鉛壁有多少比值的中子會透過去？我們已知的條件是：

(1)中子 P 衝進一個 D 的距離便要碰到一個鉛原子而彈射至任意方向。

(2)一個中子在碰到十個鉛原子後，若仍不能跳出鉛壁外，則此中子將力竭而停留在鉛壁內。



(甲)



(乙)

在此問題中，我們也可借用隨週機數系列來決定中子被鉛原子彈射後的任意方向。如 2 位系列　　01 ， 13 ， 69 ， 97 ， 61 ， 93 ， 09 ， 17 ， 21 ， 73 ， 49 ， 37 ， 81 ， 53 ， 89 ， 57 ， 41 ， 33 ， 29 ， 77 ，……

模擬中子 P_1，P_2 的運動如甲圖，而以隨週機數系列所表示的任意方向則根據乙圖。在甲圖中 P_1 碰 4 次便跳出壁外， P_2 碰十次後仍不得跳出，故停留在壁內。「模擬實驗」（ Simulation ）的中子越多，所得結果也應更爲可靠，但所需要的隨週機數系列也越長。幸好一般計算

機都具備有製造如此系列的性能故可源源供應不絕。我們甚至還可把這種模擬情況圖顯示在螢幕機上，而使眼睛直接看到其運動。

至於爲什麼中子衝進一個D就會碰擊到一個鉛原子，以及爲什麼碰擊十次便要衰竭，這些當然都要有所依據。這種依據可以是理論計算，也可以是多次實驗的結果，但總離不了機率的推測與隨遇機數系列的應用，而這種機率理論又起源於賭博的輸贏率，因之此法遂以賭博聖地Monte Carlo稱之。

因爲一切狀況都憑想像並無實體上的損失故用途很廣，如紅綠燈的管制、車禍的發生、地震、火災、太空旅行等祇要能設立情況，再把它數值化便無所不適用，也不受時間、空間的限制。故若着迷的話，眞可以樂此設計而不疲。不過我們也應該瞭解千百次的模擬實驗有時還不如一次事實的演變，如經過模擬實驗保證不會出事的耐火耐震摩天大樓，居然被地震震裂起火倒坍了，錯的當然是模擬實驗，而絕不是地震、火災。但模擬實驗若重複多次，終不失爲一種可靠的指標，自有其不容忽視的功能，要能驅使那些千頭萬緒十分龐雜的資料而在可容許的時間內整理出一個具體結果出來確實不簡單，也祇有身懷「超速」絕技的計算機才能辦得到。