臺灣二００七年國際科學展覽會

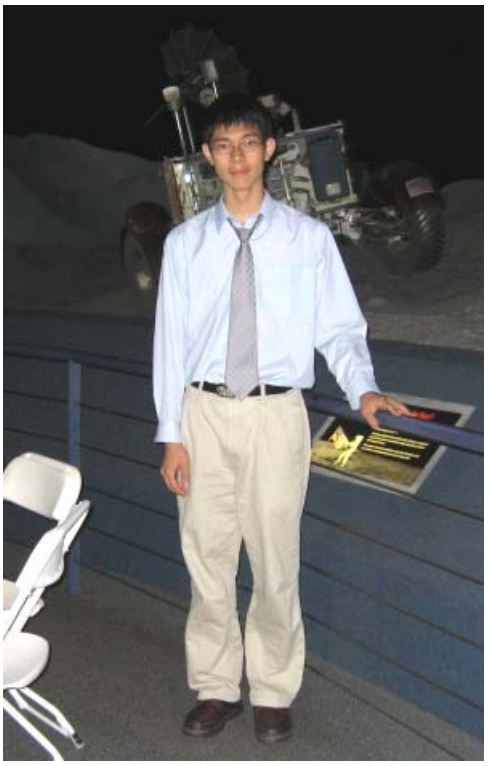
科　　別：電腦科學

作 品 名 稱：Equtatetor-新一代智慧型數學處理器

得 獎 獎 項：第二名
英特爾電腦科學獎：第一名

學校 / 作者：國立臺灣師範大學附屬高級中學　趙任康

作者簡介



從小我的興趣廣泛，不論是人文科學、文學、自然科學皆各有所好。考上師大附中數理班後，漸漸對數理等相關學科產生了濃厚的興趣，尤其是數學、物理及資訊。我喜愛研究數學、物理及資訊的題目，並藉由多項的競賽經驗，磨練並充實自我。也由於自身酷愛數學及程式設計的關係，因而對電腦或計算機等工具深感興趣，而有了這次的研究活動。希望藉由種種的學習經驗，使自身的能力更上一層樓，以求對社會有所貢獻。

研究報告

作品名稱：Equatetor － 新一代智慧型數學處理器

英文摘要(Abstract)

Equatetor – A New Intellectual Mathematical Processor

The object of this study is to design a method and processor which is able to edit, display a mathematical expression representing a number, calculate and output the answer. The executor of this task is called Equatetor. Normal calculators are not adequate for this kind of task. The main reason is that they can't reveal the original expression, such as fractions, radicals, exponents or mathematic functions.

Therefore, a simple and convenient method is needed. To perform the possible way of handling those tasks, a computer program has been written. Several techniques were used, such as MathML, computing algorithms, data structures, and so on. Following are main purposes:

(1) Displaying mathematical expressions.

(2) Editing mathematical expressions simply.

(3) Calculating mathematical expressions.

(4) Outputting the answers(in different expressions).

And the achievements:

(1) Structured methods of editing of mathematical expressions.

(2) Displaying mathematical expressions completely.

(3) Calculating mathematical expressions precisely.

(4) Offering answers in different expressions.

中文摘要

此研究的目的是要設計出一套完整編輯顯現數學式、加以計算，並求出解的一套方法與成品。而這項工作的執行者，在此稱之 Equatetor。一般的數學式子，若要計算的話，普通的計算機是不足夠的。原因是它們沒有辦法表現出數學式的「原貌」，例如分號、指數、函數、根號等數學符號混在一起時的情況。

於是，我便擬定了一個研究，希望設計出一套更方便且實用的方法。換句話說，我要設計出一個功能強大的工程計算機程式。其中，自然牽扯到數學式子的顯現方式（以 MathML 實現），以及計算機科學的演算法及資料結構。我主要的目的有四：

(1) 顯示數學式
(2) 方便編輯數學式
(3) 計算數學式
(4) 處理可以以不同形式輸出解答的計算（如輸出分數、根號、函數解等）。

研究結果中，成功地運用 XML 中的 MathML 與二分逼近分數等演算法及若干資料結構，達到了以下實用的幾點：

(1) 結構化的數學式編輯
(2) 完整地顯示數學式
(3) 正確運算並輸出運算式的答案
(4) 提供一般數學形式之解（非小數之解）

壹、 研究動機

數學式的運算，一向是國高中生常在做的數學功課。常常我們會用手算，但計算機也是一項不錯的選擇。它的便利性，在於有效率及零失誤的計算。

然而，對於現在是高中生的我，卻又少了那麼一份便利。原因是高中的數學比國中艱澀許多，數學式的內容與變化更超出國中的好幾倍。一般只能加減乘除的計算機，是沒有辦法應付的。經由同學的口中，得知市面上有販售工程型的計算機，功能十分強大，於是我便買了一台試試。用過之後，不但沒有滿足，反而發現了遺憾之處：一般的工程計算機只能支援「一長串」的數學式。所謂一長串的式子，就是有違一般我們在書本上看到的數學式原貌：不但是平面豐富的，而且數與數、符號與符號的關係一目了然。這對簡單的式子也許影響不大，但對複雜的式子，卻顯的些許牽強。舉例來說：

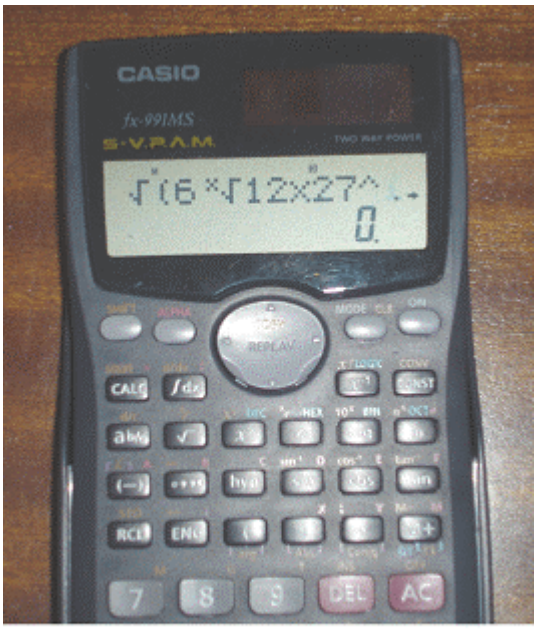
$$\frac{\sqrt{\sqrt[6]{12}\cdot 27^{\frac{1}{4}}}}{\sqrt[3]{72}}$$ 這樣的一個數學式子

$$\sqrt{(6\sqrt{12\times 27\wedge(1\div 4)})\div(3\sqrt{72})}$$

在計算機的顯示模式下便成這樣

↑圖1 書上看到與工程計算機中數學式的差別

在上圖後者的式子中，可發現其中含有許多的括號，其中還穿插不少數學符號。這樣雖然一樣可求出答案，但輸入的效率與美觀性卻大大打了折扣。無法顯示出數學式的原貌，是計算機領域中的一大遺憾。雖然近數月來，市面上推出了新款的工程計算機，而可以支援上述的一般數學式顯示。然而其編輯方式與易上手性仍嫌不足，其技術亦不夠成熟。



↑圖2 對於一般的工程計算機，只能支援「一長串」的數學式。另外，較複雜的數學式連螢幕都塞不下。

此外，現今的工程計算機仍有許多重要的缺憾。例如數學式的輸入效率不足、數學上的無理數無法依照精準度的輸入而求得相接近的分數、無法做到數學函數或根號的輸出（例如

輸入$\log_{10} 5 \cdot \log_5 2$無法輸出$\log_{10} 2$等等）、多元多次方程式或一般方程式只能求得小數解而無法以分數或根號表示、隱形乘號的功能太弱（如$2\sqrt{3} = 2 \times \sqrt{3}$）…等等功能，想必是有待加強的。

因此，我希望透過電腦程式的快速與便捷性，開發出一個功能強大的工程計算機，而能將上述種種的遺漏給一網打盡，提供現今的高國中學生們一套好用的計算機程式。更重要的是，它是完全免費的。

爲配合高級中學的數學教材，本程式擬定以製作高中數學所使用到的數學元件爲原則。內容橫跨高級中學數學之數與座標系、指數與對數、三角函數、排列組合、多元多次方程式等。

貳、 研究目的

一、 數學式的編輯

研究並開發出便利性高的數學式輸入方法，以使使用者能完整地完成數學式的編輯，且不受限於數學式的規模。

二、 數學式顯現

發展出一套支援符號顯示的數學式顯示器，藉由表單使使用者加入元件，可將各符號間的關係顯現在程式中。

三、 數學式運算

將顯現出來的運算式加以運算，輸出解答。

四、 處理可以以不同形式輸出解答的計算（如輸出分數、根號、函數解等）

例如求得逼近無理數的分數、以根號或數學函數顯示答案、方程式求得自然顯示解（分數、根號）等。

參、 研究過程

一、 尋求顯示符號及函數元件的方法

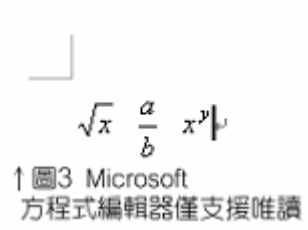
本程式第一項難關在於，如何能有別於一般工程計算機地顯現出完整的符號及函數系統。在研究的初步時期，整理了若干項表現出符號的方式。茲整理如下：

1. 手工方式繪製根號、分數圖形

簡單的線條，便可以表現出許多的數學符號。最簡單的圖形，運用小畫家就可以繪出。然而用此種方式，效率並不高且不易控制其大小，因而放棄此法。

2. 運用元件中的 Microsoft 方程式編輯器顯示

Microsoft 方程式編輯器可有效地編輯出美觀的數學式圖形，然而其格式爲既有，除在起始製作圖形時可編輯圖形外，爾後使用者便無法改變其屬性，而成爲唯讀，因而亦放棄此法。（見圖 3）



↑圖3 Microsoft 方程式編輯器僅支援唯讀

3. 運用 XML 的標籤元素處理

MathML 提供了一項有效率地顯示出數學式的規則。運用 XML 標籤的屬性，可藉程式作爲使用者與標籤之間的媒介，有效地控制其標籤語言，達到顯示數學式的目的。此方式不只可讀可寫，對於數學式的運算更是一大幫助，因而採用此法。

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>...</head>

  <h1>Example</h1>
  ....
  <math xmlns="http://www.w3.org/1998/Math/MathML">
   <mi>x</mi><mo>+</mo><mn>3</mn>

  </math>
</html>
```

圖4 MathML數學標籤語言的誕生，使數學的符號能更方便的表示。

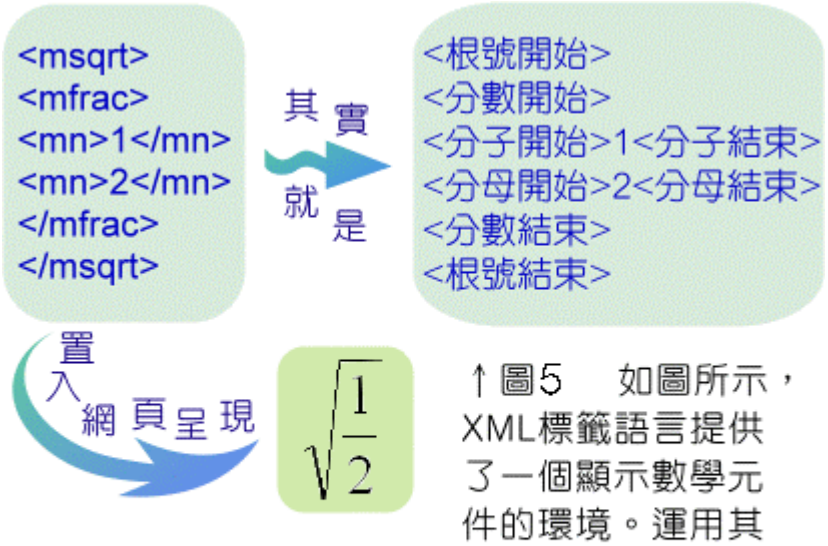
二、 XML 的數學標籤語言—MathML 專題研究

數學標籤語言(Mathematical Markup Language，簡稱 MathML)是 XML 於數學領域的一種應用，由於以往在 Web 網頁上難以表示數學資料，因而發展出 MathML。MathML 的設計使數學公式及科學資料等在 Web 上更容易呈現，可廣泛應用於科學、數學、工程等領域。在 MathML 標籤語言中，任何數學式皆由多層的標籤及內容所組成(見圖 5)。

除此之外，使用 MathML，必須先做到以下兩個步驟：

1. 用 XHTML 建立一個內建 MathML(inlined MathML)的網頁
2. 加入樣式表

完成以上兩步驟後，便可加入數學符號的標籤而達到顯示的目的。藉由此法，不僅可具結構地表示出數學式，在程式過程中也不易出錯；更重要的是，利用此一結構化的特性對於運算有較便利的幫助。

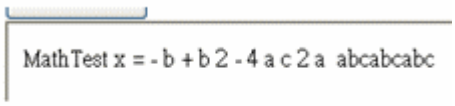


↑圖5 如圖所示，XML標籤語言提供了一個顯示數學元件的環境。運用其語言的規則，可編輯出一個完整的數學式。其結構化的語言，不僅便於顯現，亦便於運算。

三、 尋求適當的介面

1. HTML 的 iframe 元件

若能有效地藉由使用者編輯，良好的介面乃是一大要務。HTML 中的 iframe 元件提供這樣一個環境。它可將網頁檔顯示在框架之中，使用者也可以簡單地修改。在一開始的網頁測試情況仍順利，然而加入 XML 標籤後卻發生錯誤而失敗（見圖 6）。

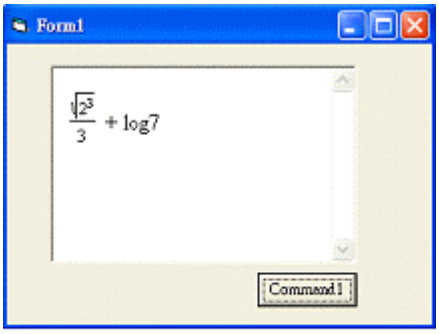


↑圖 6 HTML的iframe元件提供了一個編輯視窗，卻不支持XML。

2. Microsoft Visual Basic 的 WebBrowser 元件

WebBrowser 是爲瀏覽器，本作爲在程式介面中瀏覽網頁之用，用途類似網頁中的框架(Frame)。製作數學式顯示器，正可使用此元件作一介面，作爲顯示之用（見圖 7）。因而採用此元件。

本程式的原理，即是透過程式使使用者輸入運算式，爾後程式便輸出一個由 MathML 標籤語言所構成的 XML 文件，再行利用 WebBrowser 元件瀏覽，便可達到易編輯易顯示的功效。



↑圖 7 Visual Basic的WebBrowser元件，不但可呈現也可經由事件來編輯數學式。

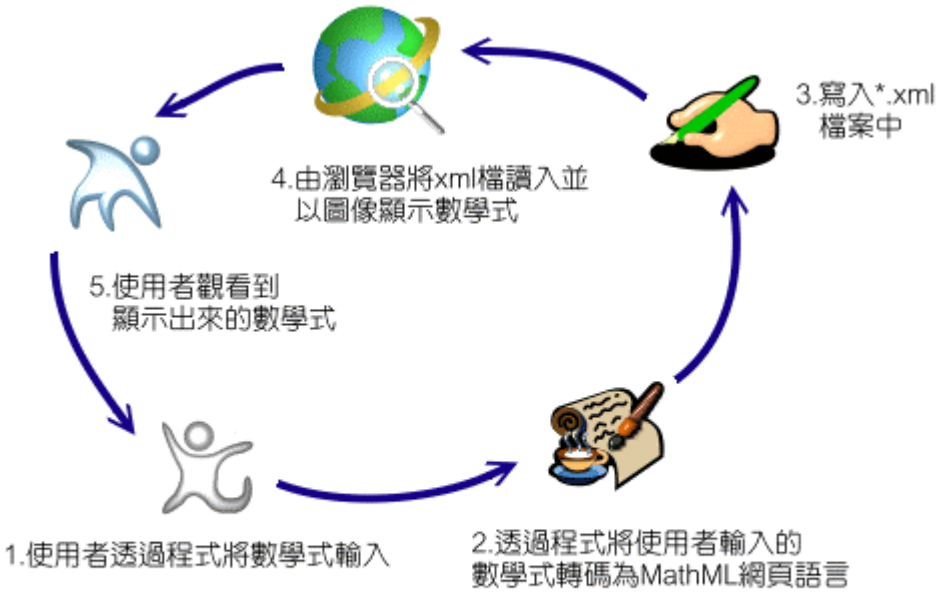
四、 實地測試：加入第一個符號元素—根號

選定了製作的程式與元件後，進而測試第一個元素—根號。諸如 $\sqrt{3}$ 之類的元素，成功地透過程式介面顯現至元件中，這是初步的一大成功。由於根號在 XML 語言中轉變成了標籤，因而只要在前後加上標籤便可將數學式加入根號之中。選取適當的部分外加標籤，便可自由地選定根號的範圍。（見圖 8）

↑圖 8 最早測試的程式畫面，加入了第一個元素-根號，僅能拿根號相加。

五、　確立輸入並顯示數學式的流程

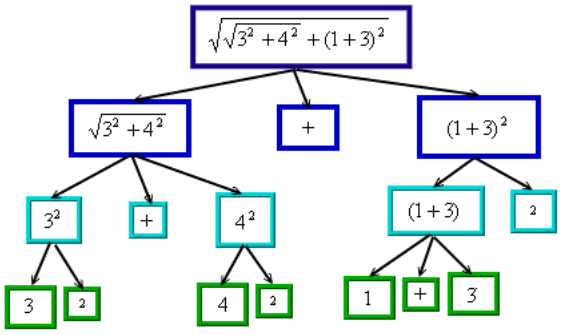
從使用者輸入數學式到顯示出數學式給使用者，可以下圖的流程表示：



↑圖9 程式中輸入並顯示數學式的流程

六、　將輸入的資料運算輸出

如何使 XML 的標籤加以運算？XML 中的 DOM，可將標籤視爲一顆樹(Tree) （見圖 10），可在節點間無止盡地往下走。熟悉 XML 的標籤規則後，建立讀取標籤的副程式，便可將輸入的數、符號、函數，一一地剖析出來。

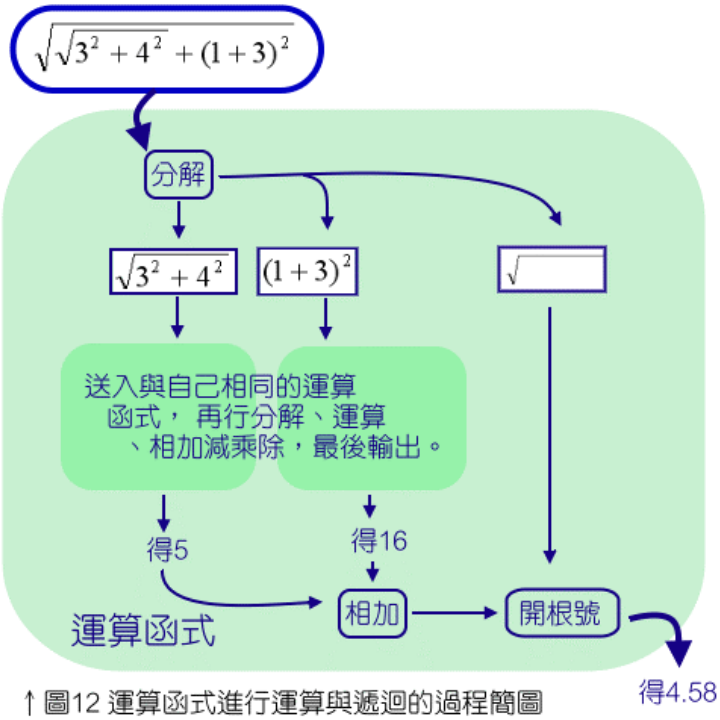


↑圖10 數學式，就好比是一棵大樹，
可以分解成更小的數學式。
如此分支下來，最小的元素
便是運算元跟運算子。

爲方便運算，在程式中特開發了一個函式（或稱副程式），主導運算。他的任務很簡單：將使用者輸入的數學式子（以上述之 MathML 標籤語言輸入），經過一系列演算法的運作後，將正確答案輸出給使用者。目的雖簡單，但運作過程卻相對顯的複雜許多。其主要步驟有以下數點：

↑圖11 運算函式簡要運作流程

在函式中，運用了呼叫自己的功能，換言之，便是遞迴。以具體的例子呈現，可由圖12一窺究竟：



↑圖12 運算函式進行運算與遞迴的過程簡圖

運用電腦來幫忙計算，自然需要讓電腦知道做運算的規則。在所有的運算子中，並不

是每一種運算子的階級都一樣；因而需要告訴電腦做運算的先後次序。爲達到這項技巧，我實習了運算式的這一章節，運用資料結構的輔助及中後序式的轉換來達到此效果。至於函數或符號中的運算式，則將其視爲一體輸入運算函式中計算，再行輸出成爲單獨的運算元，如此一層層疊上來，配合要求的運算邏輯，便可達到完整運算的功效。

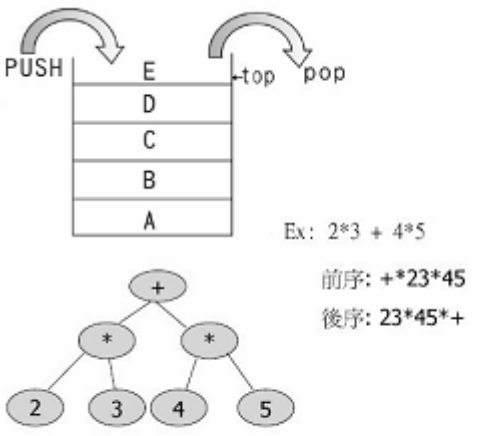


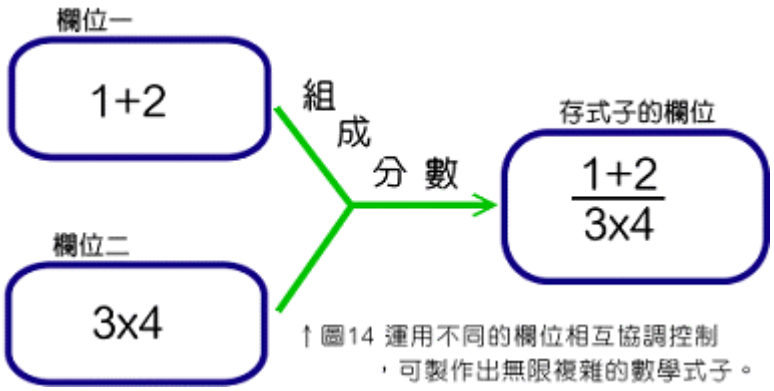
圖13 資料結構的模擬圖示
借用資料結構的輔助，可使中序式有效地轉為後序式，
並加以運算。圖上為堆疊，圖下為二元數的模擬圖示。

七、 多重欄位的編輯

現今的工程計算機，僅能在單一個欄位做編輯。因此，若遇到較複雜的數學式時，便顯得不易輸入而缺乏效率。爲打破此一限制，本程式開創了多欄位的編輯模式。在製作一個大數學式時，可先由其中所含的小數學式下手。編好小數學式後，再複製至其他的欄位組合起來，最後便能完成大數學式的編輯。（見圖 14）

對於不同的數學元件（符號、函數等），其所含有的參數（或稱引數）也不盡相同。例如對數函數 log 有底數與真數兩個參數，分數亦有分子與分母兩個參數，sin 函數卻只有一個。因此本程式也分別開創了不同的欄位，以存放即將送到數學元件中的小數學式。例如對數函數可以 $\log_x y$ 表示，x, y 爲不同的欄位，可分別插入不同的數學式（因爲一個數學式代表一個值）。

在不同的欄位，也可先行求得值，以確保數學式輸入無誤。



↑圖14 運用不同的欄位相互協調控制
，可製作出無限複雜的數學式子。

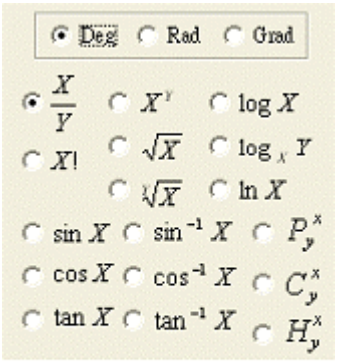
八、　符號、函數、常數的開發

待各輸入、輸出部門都完備後，便著手研究各符號、函數的顯示規則及推導計算的相關公式。重要的數學符號與函數茲列舉如下：

1. 分數(Fraction)
2. 階乘(Factorial)
3. 指數(Exponent)
4. 根號(Radical)
5. 對數函數(Function of Logarithm)
6. 自然對數函數(Function of Napierian Logarithm)
7. 正弦函數及反正弦函數(Function of Sine and Arc sine)
8. 餘弦函數及反餘弦函數(Function of Cosine and Arc cosine)
9. 正切函數及反正切函數(Function of Tangent and Arc tangent)
10. 排列與組合(Permutation and Combination)

至於常數方面，則有：

1. π 表圓周率 ≒ 3.1415926535
2. e 表自然對數 ≒ 2.718281828



↑圖15
豐富的符號函數元件

九、 分數解的製作

在運算的時候，往往不僅需要以小數的方式求解，也希望能求得分數的解。除了方便記憶某數值之外，方便運算也是一大分數解的功能。

由於數學中的實數分爲無理數及有理數，而其處理方式也大異其趣。爲此，針對兩種數值分別討論其適用的演算法：

1. 有理數—以數學方法求得

$$\frac{q}{p}$$

由於有理數可表示爲 $\frac{q}{p}$ 的形式，因而能以數學方法求得其分數值。有理數可分爲有限循環小數及無限循環小數：

(1) 有限循環小數

令 t 爲一有限循環小數，則 t 可表爲 $n_0 + 0.n_1n_2n_3.....n_m$ 的形式 $(n_0, m \in Z, m \geq 0, \bigvee_{i \in N} n_i \in N, 0 \leq \bigvee_{i \in N} n_i \leq 9)$，即

$$t = n_0 + 0.n_1n_2n_3.....n_m = n_0 + \frac{n_1n_2n_3.....n_m}{10^{m+1}} = \frac{10^{m+1} \cdot n_0 + n_1n_2n_3.....n_m}{10^{m+1}}$$

再以輾轉相除法求得分子與分母之最大公因數，約分至最簡分數即得所求。

(2) 無限循環小數

令 s 爲一無限循環小數，則 s 可表爲 $n_0 + 0.p_1p_2...p_qn_1n_2...n_mn_1n_2...n_m...$ 的形式 $(n_0, m, q \in Z; m, q \geq 0; \bigvee_{i \in N} n_i, \bigvee_{i \in N} p_i \in N; 0 \leq \bigvee_{i \in N} n_{i,}, \bigvee_{i \in N} p_i \leq 9)$，即

$$s = n_0 + 0.p_1p_2...p_qn_1n_2...n_mn_1n_2...n_m... \text{---式 1}$$

等號兩側同乘 10^m 得

$$10^m \cdot s = 10^m(n_0 + 0.p_1p_2...p_qn_1n_2...n_mn_1n_2...n_m...) \quad \text{---式 2}$$

式 1 減式 2 得

$$(10^m - 1)s = 10^m(n_0 + 0.p_1p_2...p_qn_1n_2...n_m) - (n_0 + 0.p_1p_2...p_q)$$

得

$$s = \frac{10^m(n_0 + 0.p_1p_2...p_qn_1n_2...n_m) - (n_0 + 0.p_1p_2...p_q)}{10^m - 1}$$

配合上述有限循環小數作法，得

$$s = \frac{10^m(n_0 + \frac{p_1p_2...p_qn_1n_2...n_m}{10^{q+m}}) - (n_0 + \frac{p_1p_2...p_q}{10^q})}{10^m - 1}$$

化簡後，再以輾轉相除法求得分子與分母之最大公因數，約分至最簡分數即得所求。

2. 無理數—以程式迴圈方法逼近

如同中國古代數學家祖沖之所提出圓周率的密率與約率一樣，若能依照使用者的需求輸入精準度，便可以簡單的分數代表無理數。爲此，便著手研究求得逼近分數解的方法。若使用數學方法取適當位數而視爲有限循環小數，便會出現以下的窘境（以圓周率 π 爲例）：

$$\pi \fallingdotseq 3.14159265358979 = \frac{314159265358979}{100000000000000}$$

不但因過度複雜而缺乏實用性，其分數與小數差異也不大而失去意義。因此，改以另一套演算法來達到逼近的效果。程式中先取出整數部份，使輸入值小於 1，待分數值輸出後再加上原先整數部分。

副程式中，數值(<1)及誤差值爲輸入的兩筆資料，另外設有左界、中界、右界，每界有分子分母兩個變數，共六個變數。初始令

左界 = 0/1 = 0.0

右界 = 1/1 = 1.0

迴圈的每一步驟中，執行以下的演算法：

令中界 = (左界分子+右界分子) / (左界分母+右界分母)

若 |中界-數值(原小數數值)| < 誤差值，則

程式輸出中界，跳出迴圈

若輸入值大於中界，則

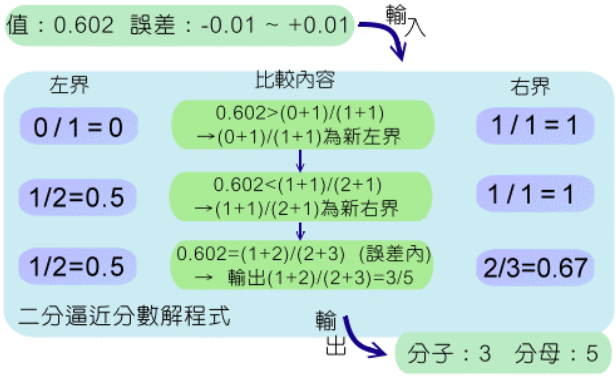
令左界 = 中界，右界依舊

若輸入值小於中界，則

令右界 = 中界，左界依舊；

重複迴圈動作

於是可得誤差範圍內分母最小的分數。



↑圖16 二分逼近分數解副程式 運用分數的性質，可將小數步步逼近於分數。本副程式的特點為，可依使用者的需求調整精準度（及誤差值的大小），以切合實際需求。

十、 數學自然顯示

所謂數學自然顯示，便是以帶有數學符號的方式輸出答案。例如答案帶有根號、分數、函數或常數等。此一功能，在現今的數學式求解中十分重要，原因是如此方能零誤差地表示出正確的值，也較容易在人爲計算中化簡。然而現今的工程計算機中，此項技術並未成熟，於是便有研究的價值。

數學自然顯示中，自然分爲許多種如根號、函數…等。程式中主要利用比較方式，將小數反求取較接近的自然表示法。如同前述的逼近分數法，由使用者輸入精準度（及誤差範圍），若某元件的值與所求的差小於給定的誤差範圍，便輸出此元件。每種數學元件，自然各自有不同的優先順序，茲介紹如下（下文中之所求值即爲計算後之答案）：

1. 整數

顯示條件爲計算後之數值與其最接近的整數數值之差在輸入之誤差值內。

作法：令 INT (x)爲 x 四捨五入後的整數。令值爲 t，若 $|t - INT(t)| <$ 誤差值 ，則輸出 t。

2. 根號

顯示條件爲計算後之數值不符合以上之條件，且其與最接近之根號整數值（如1.414213=$\sqrt{2}$ 爲一根號整數值）之差在誤差值之內，以下相似之其餘數學元件依此類推。共分 3 種形式，列作法於下：

(1) 形式 1：$\sqrt[n]{s}$

令所求值爲 t、$s = t^n$ 、$s' = INT(s)$，其中 n 爲大於 1 之任意正整數，若 $\left|t - \sqrt[n]{s'}\right| <$ 誤差值，則輸出 $\sqrt[n]{s'}$（取可能之 s' 中的最小值）。

(2) 形式 2：$p \cdot \sqrt[n]{s}$

承上，若能表示爲形式 1，則繼續判斷：若 $p^n \mid s'$（p 爲任意正整數，取可能之 p 的最大值），則輸出 $p \cdot \sqrt[n]{\frac{s'}{p^n}}$，其中 $\frac{s'}{p^n}$ 可以一整數表示。證明：$\sqrt[n]{s'} = \sqrt[n]{\frac{p^n s'}{p^n}} = p \cdot \sqrt[n]{\frac{s'}{p^n}}$。形式 2 之優先順序高於形式 1，因爲此形式一般較易爲使用者使用。

(3) 形式 3：$\frac{p \cdot \sqrt[n]{s}}{q}$

與形式 2 相似，若所求值乘上一任意正整數 q 後可表爲形式 2 之形式，則以此形式呈現。

3. 對數函數

顯示之條件與上述相似。共分 3 種形式，列作法於下：

(1) 形式 1：$\log s$

令所求值爲 t、$s = 10^t$ 、$s' = INT(s)$，若 $\left|t - \log s'\right| <$ 誤差值，則輸出 $\log s'$。

(2) 形式 2：$\log_p s$

令所求值爲 t、$s = \mathrm{p}^t$ 、$s' = INT(s)$，其中 p,s 爲正整數，$p \neq 1$，若 $\left|t - \log_p s'\right| <$ 誤差值，則輸出 $\log_p s'$。

(3) 形式 3：$\ln s$

令所求值爲 t、$s = \mathrm{e}^t$、$s' = INT(s)$，其中 s 爲正整數，若 $\left|t - \ln s'\right| <$ 誤差值，則輸出 $\ln s'$。

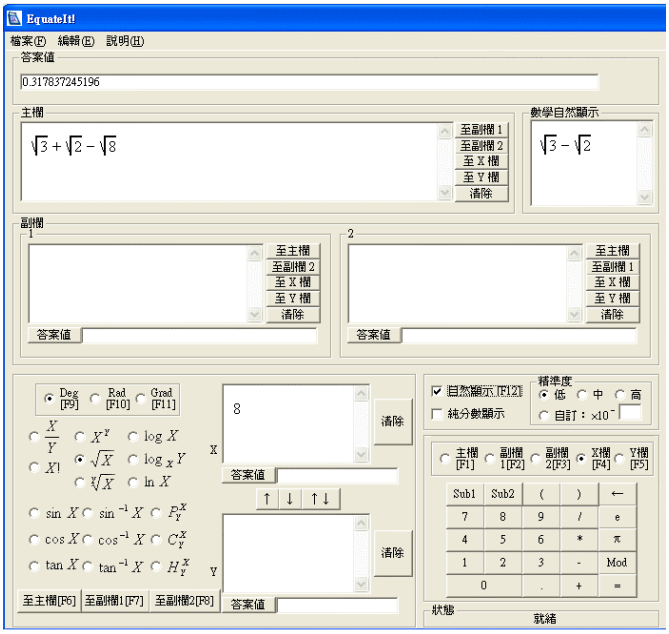
4. 三角函數（0≤所求值≤1）

顯示之條件與上述相似。共分 3 種三角函數，列作法於下：

令函數爲 f(x)（f(x) = sinx, cosx, tanx）、所求值爲 t、$s = f^{-1}(t)$ 、$s' = INT(s)$，若 $\left|t - f(s')\right| <$ 誤差值，則輸出 $f(s')$。（s 爲弧度或角度視使用者而定）

5. 分數

當上述之情況皆不符合時，採用分數。如前所述，有理數可表示爲一分數，無理數亦可以迴圈方式逼近。

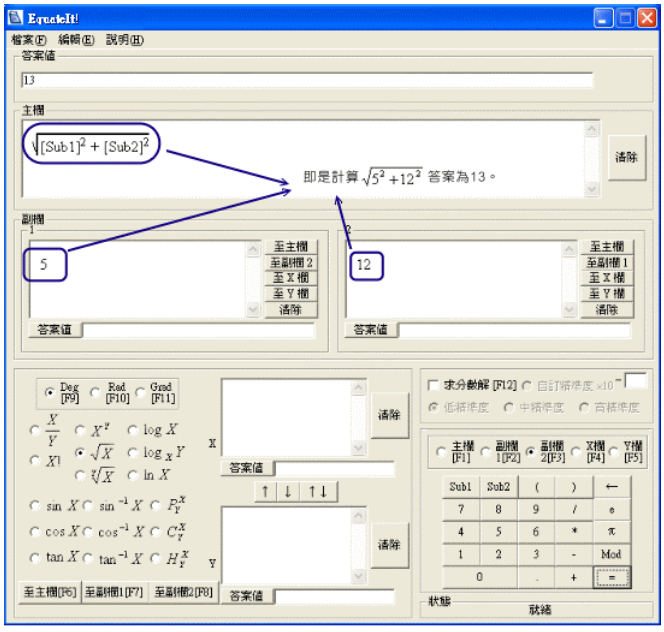


↑圖17　以數學自然顯示方式輸出，對使用者做運算有更大的便利。

十一、　自製函數 — 以欄位參數實現

在探討數學運算式的問題時，經常須討論不同的參數值，在同樣的參數位置與同一運算式中，其值的變化情況。換句話說，便是運用現有的元件自行製作數學函數，而探討此函數在不同的參數時，函數值的變化。好比探討 $f(x,y)=\sqrt{x^2+y^2}$ ，在不同的 x 與 y 時，其函數值的變化。若一次又一次的帶值進入式子運算，則實在缺乏效率。

爲實現此一目標，本程式中設計了欄位參數（見圖 18），設兩個副欄的內容爲參數，只要在主欄中的運算式嵌入欄位參數，便可以只修改副欄內容，而達到輕鬆討論自製函數值的目的。



↑圖18　運用欄位參數，每回編輯數學式時，只需更換參數值即可。
（例如打入3與4，可算出5，不需重新編輯數學式）

十二、 程式的細節控制與錯誤訊息

一個完美的程式，不只要功能強大，其細節亦十分重要。下述幾項與數學式相關的細節逐步被加入程式考量當中：

1. 隱形乘號(Invisible Times)

若我們仔細觀察一般所使用的數學式，會發現數與數之間帶有乘號的存在。例如 2π，雖然 2 跟 π 都代表著一個數，然而將他們擺在一起時，卻定義它們其中帶有乘號，因而 2π 實際上代表 $2 \times \pi$。此時需額外定義一個新的運算子，即隱形乘號。

2. 倒退功能(Backspace)

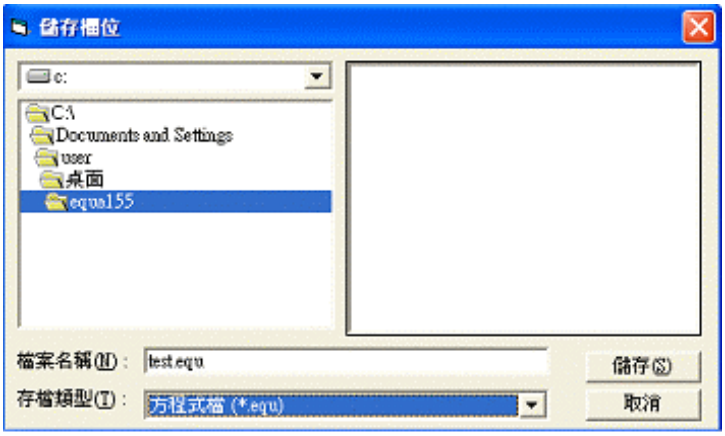
編輯數學運算式時，往往會發生不經意的錯誤，例如輸入錯誤的符號或數字。此時若將編輯欄全數清除以重新編輯，則不但太浪費也沒效率。利用倒退功能（程式中以「←」鍵代表），則可清除前一個錯誤的資料。

3. 自動括號判斷

當加入函數或符號的參數爲一運算式時，有時必須加入括號才可避免資料發生視覺上的錯誤。例如 $\sin(\frac{2\pi}{3}+\frac{\pi}{3})$若寫成 $\sin\frac{2\pi}{3}+\frac{\pi}{3}$則意義便完全不同，前者值爲 0，後者值爲 2.96。

4. 數學式的儲存

對於使用者，往往會需要編輯較長的數學式；或是常常需要使用類似的數學式。因此，本程式設計了儲存數學式的功能，可自任一欄儲存至檔案，亦可自檔案讀入至任一欄。



↑圖19 建立與Windows一般應用程式相當的儲存與讀取視窗，使使用者能夠容易上手。

5. 錯誤訊息

程式進行的過程中有可能發生使用錯誤，爲避免無故跳出程式，故設計若干錯誤訊息於程式錯誤時提出（見圖 20）。錯誤可分爲以下三類：

(1) 敘述錯誤(Syntax Error)

當使用者輸入不合邏輯的數學運算式時，即發出此訊息。

(2) 數學錯誤(Math Error)

當使用者輸入與數學符號或函數定義矛盾的數學運算式時，即發出此訊息。

(3) 資料錯誤(Data Error)

當使用者輸出答案，必要資訊卻遺漏時，即發出此訊息。

↑圖20 當使用者輸入的數學式發生錯誤時，答案欄便輸出錯誤訊息，以防程式自動關閉。

6. 智慧性捨去不精準位數

由於程式中的資料型態所儲存的位數有限，在不精準的值與不精準的值相互運算的情況下，誤差值將愈算愈大。在此情況下，若可自動捨去不精準的位數，便可防止誤差擴大的情況。程式中便加入了此一功能，寧可輸出的值沒有那麼多位數，而不要輸出末幾位數含有離譜誤差的值。

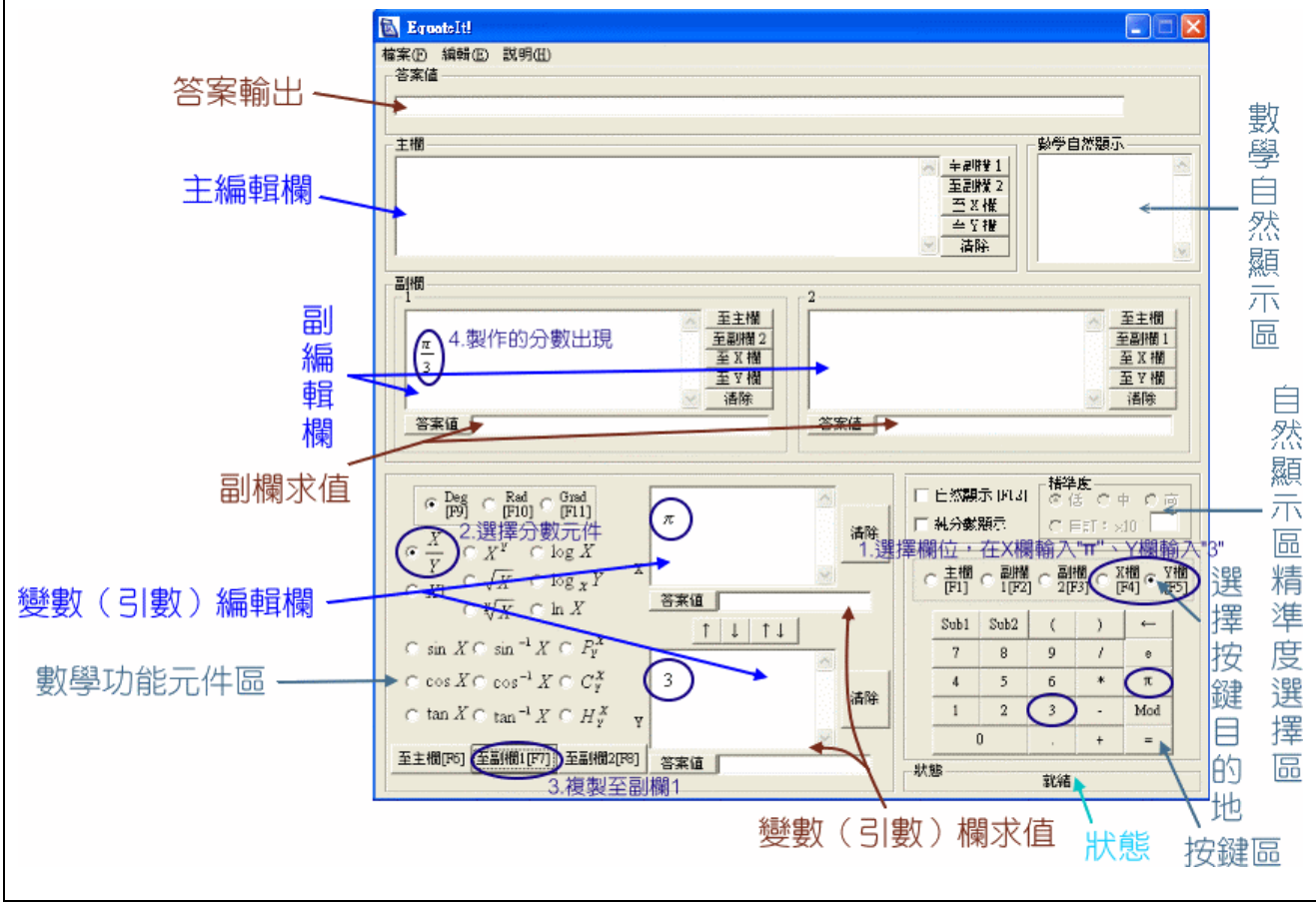
7. 鍵盤的輸入設計

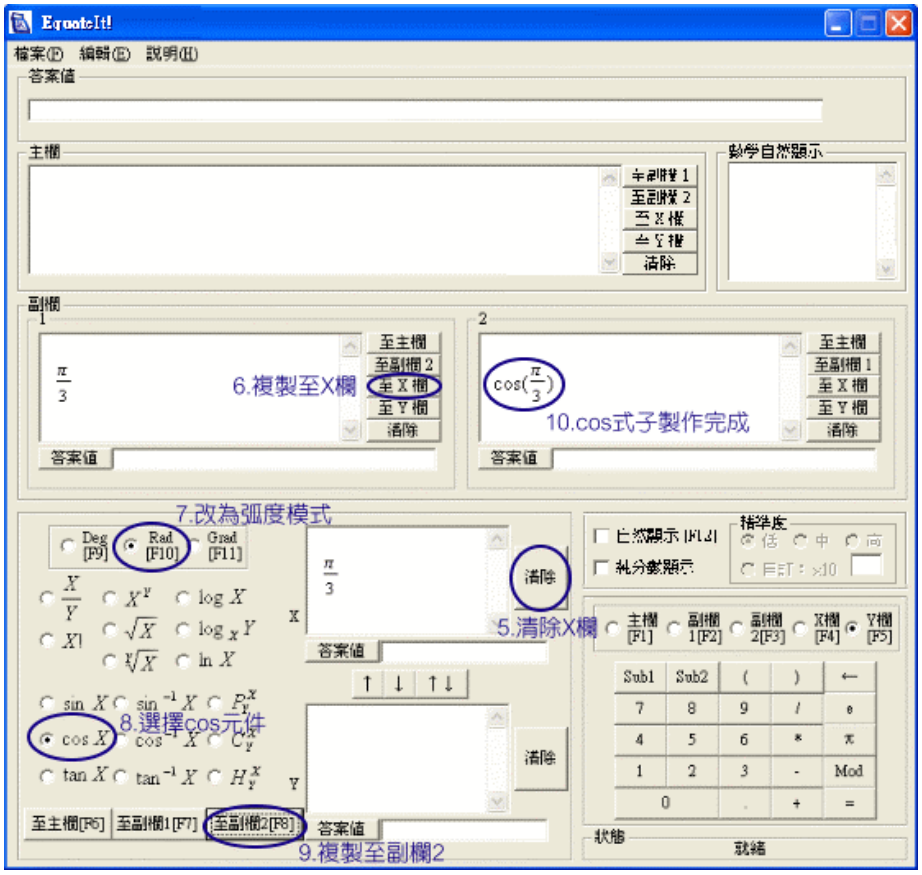
有鑑於許多電腦使用者較習慣於使用鍵盤，因而訂定出一套鍵盤輸入的規則，藉此不一定需要動用滑鼠來達到編輯的目的。

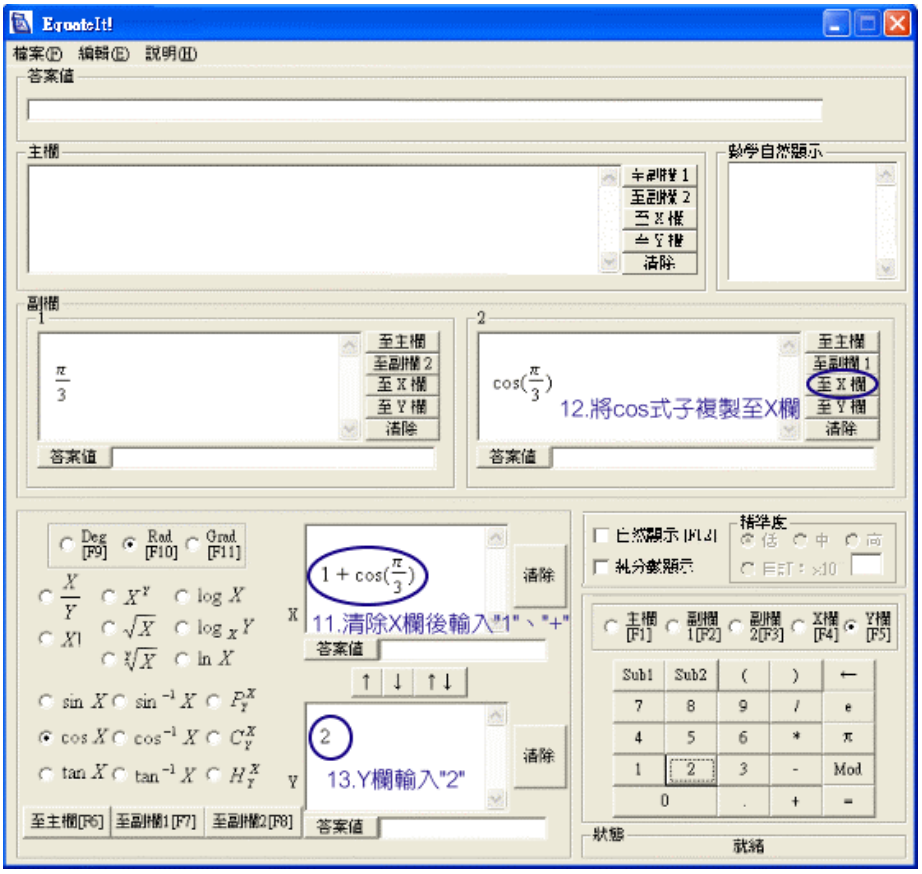
十三、程式畫面與使用過程示範

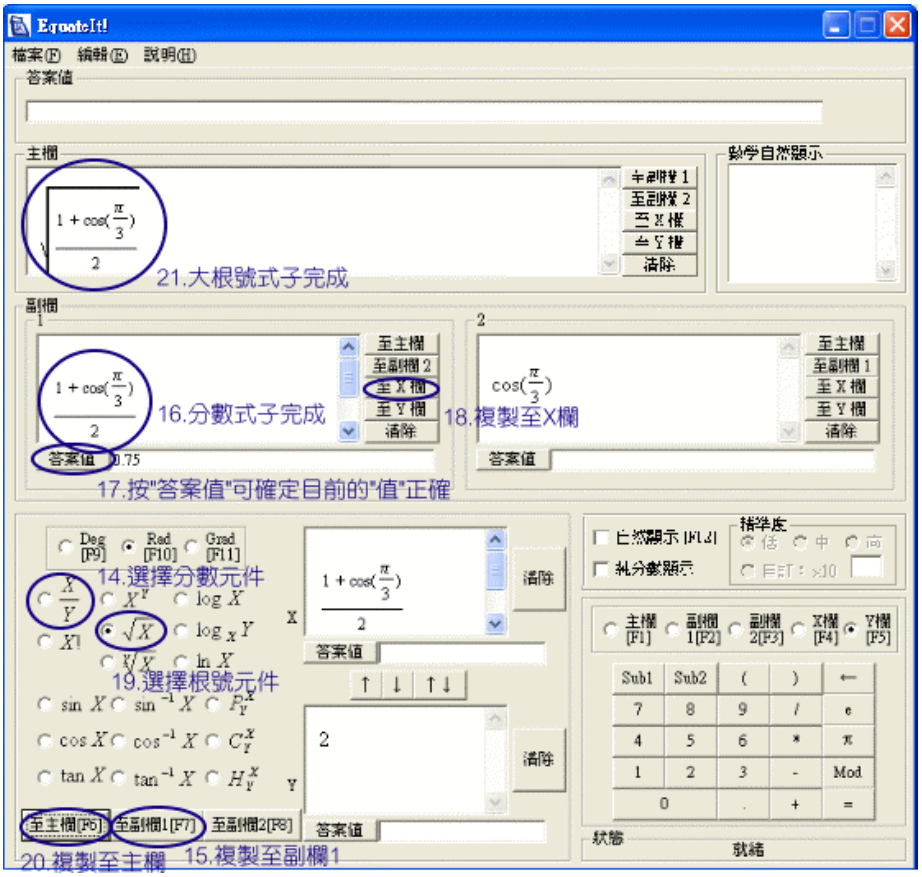
假定欲設計數學式 $\sqrt{\frac{1+\cos\frac{\pi}{3}}{2}}+\sqrt{\frac{1-\cos\frac{\pi}{3}}{2}}$ 並運算，過程示範如下：

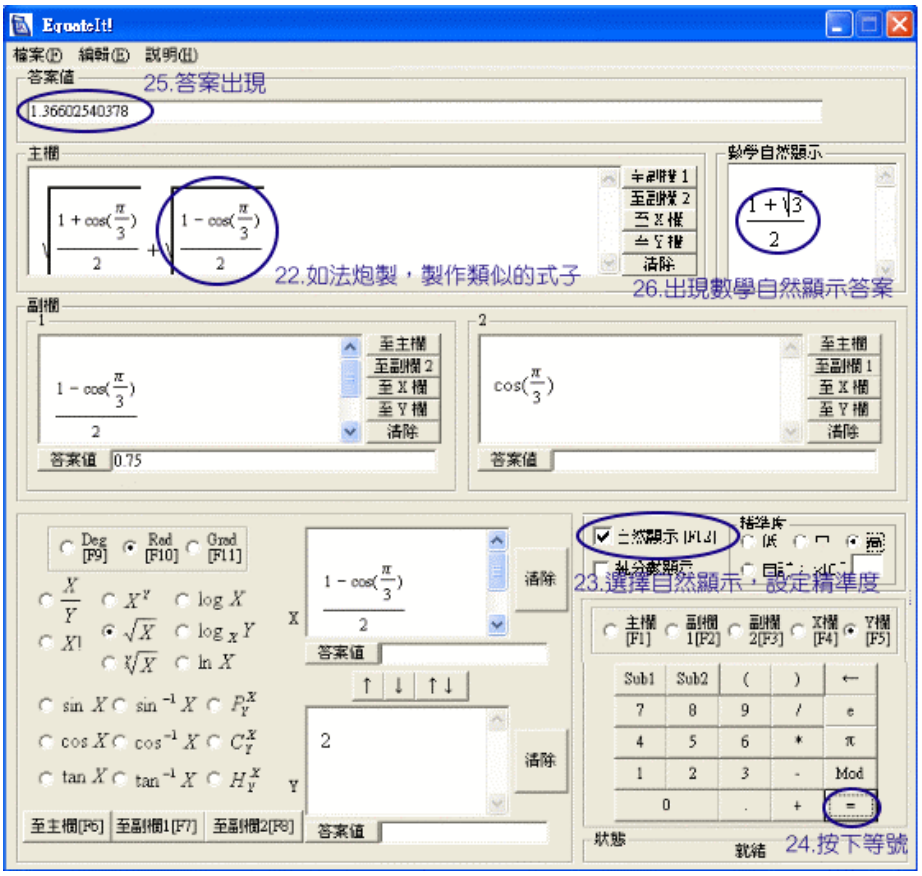
$$\sqrt{\frac{1+\cos\frac{\pi}{3}}{2}}+\sqrt{\frac{1-\cos\frac{\pi}{3}}{2}}$$




EquateIt!
檔案(F) 編輯(E) 說明(H)
答案值
主欄
數學自然顯示
至副欄 1
至副欄 2
至 X 欄
至 Y 欄
清除
副欄
1
2
$\frac{\pi}{3}$
6.複製至X欄
$\cos(\frac{\pi}{3})$
10.cos式子製作完成
答案值
7.改為弧度模式
Deg [F9]
Rad [F10]
Grad [F11]
$\frac{X}{Y}$ X^Y $\log X$
$X!$ $\sqrt{X}$ $\log_X Y$
$\sqrt[Y]{X}$ $\ln X$
$\sin X$ $\sin^{-1} X$ P_Y^X
$\cos X$ $\cos^{-1} X$ C_Y^X
$\tan X$ $\tan^{-1} X$ H_Y^X
8.選擇cos元件
X
Y
清除
5.清除X欄
自然顯示 [F12]
科學度
主欄 [F1]
副欄 1[F2]
副欄 2[F3]
X欄 [F4]
Y欄 [F5]
Sub1 Sub2 ( ) ←
7 8 9 / e
4 5 6 * π
1 2 3 - Mod
0 . + =
至主欄[F6]
至副欄1[F7]
至副欄2[F8]
9.複製至副欄2
狀態
就緒


EquateIt!
檔案(F) 編輯(E) 說明(H)
答案值
主欄
數學自然顯示
副欄
1
2
$\frac{\pi}{3}$
$\cos(\frac{\pi}{3})$
12.將cos式子複製至X欄
$1+\cos(\frac{\pi}{3})$
11.清除X欄後輸入"1"、"+"
2
13.Y欄輸入"2"
Deg [F9]
Rad [F10]
Grad [F11]
X
Y
清除
自然顯示 [F12]
科學度
主欄 [F1]
副欄 1[F2]
副欄 2[F3]
X欄 [F4]
Y欄 [F5]
Sub1 Sub2 ( ) ←
7 8 9 / e
4 5 6 * π
1 2 3 - Mod
0 . + =
至主欄[F6]
至副欄1[F7]
至副欄2[F8]
狀態
就緒


EquateIt!
檔案(F) 編輯(E) 說明(H)
答案值
主欄
數學自然顯示
副欄
21.大根號式子完成
16.分數式子完成
18.複製至X欄
17.按"答案值"可確定目前的"值"正確
14.選擇分數元件
19.選擇根號元件
20.複製至主欄
15.複製至副欄1
至主欄[F6]
至副欄1[F7]
至副欄2[F8]
清除
狀態
就緒


EquateIt!
檔案(F) 編輯(E) 說明(H)
答案值
25.答案出現
1.36602540378
主欄
數學自然顯示
22.如法炮製，製作類似的式子
26.出現數學自然顯示答案
副欄
23.選擇自然顯示，設定精準度
24.按下等號
至主欄[F6]
至副欄1[F7]
至副欄2[F8]
清除
狀態
就緒

肆、 討論與應用

由於本程式提供了一個更爲方便且簡易的方式來做數學式的編輯與計算，因而有效地提升了一般計算時的速度（與一般工程計算機比較之下）。爲實地測試本程式使用效率優勢於一般工程計算機，特請數位同學對手邊的工程計算機及本計算機輸入同樣的數學式，而測試完整輸入的時間。數據茲列於下：

數學式：$\sqrt{\frac{\left(\sin\frac{\pi}{3}\right)^2}{\cos\frac{\pi}{4}}}$	本作者	測試同學 1	測試同學 2	測試同學 3	平均
一般工程計算機	42 秒	43 秒	41 秒	52 秒	44.5 秒
EquateIt 程式	29 秒	34 秒	35 秒	43 秒	35.25 秒

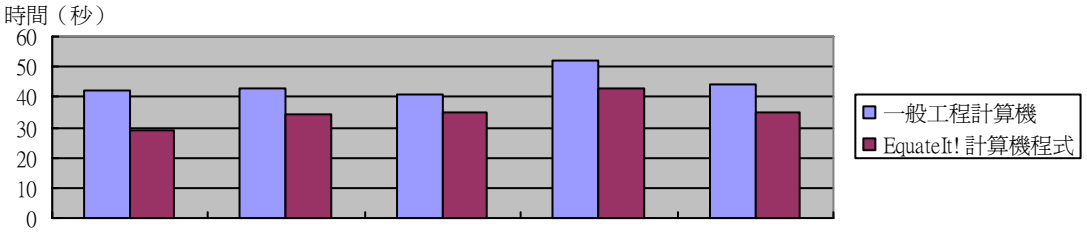


圖21 所需時間測試結果1

數學式：$\sqrt{\frac{1}{2}\frac{3}{4}\sqrt{\frac{5}{6}\frac{7}{8}}}$	本作者	測試同學 1	測試同學 2	測試同學 3	平均
一般工程計算機	48 秒	1 分 32 秒	59 秒	1 分 16 秒	1 分 9 秒
EquateIt 程式	43 秒	52 秒	54 秒	1 分 03 秒	53 秒

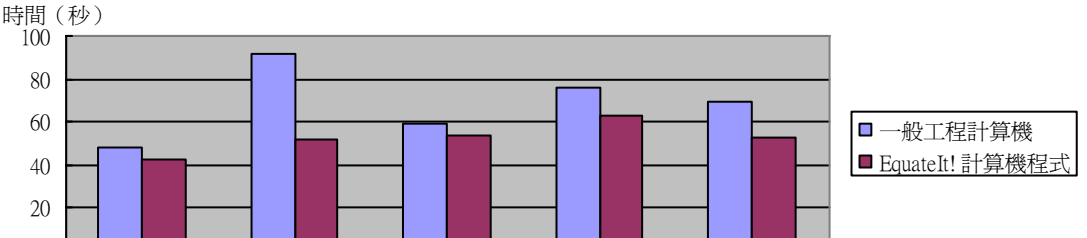


圖22 所需時間測試結果2

由以上數據顯示，使用本程式計算可有效降低輸入數學式的時間，達到有效率的效果。其主因爲，一般工程計算機僅能支援一長條式的式子，使使用者不容易輸入複雜的式子。耗時間的另一項因素在於，一般工程計算機有可能因括號或根號輸入數量過多或不足導致錯誤，使數學式必須重新編輯。

此外，本計算器提供了另一種答案輸出的方式，可以以小數以外的表示法將答案輸出。諸如分數、根號、指數或函數等的輸出方式，對使用者的便利性有一定的提升，因爲在許多的計算需求下僅有小數是不夠的。

伍、 結論

藉由以上的研究與開發製作，目前本計畫已達到了以下的目標：

一、 結構化的數學式編輯

藉由程式中欄位的控制，本程式已能使使用者完整地編輯所需要的數學式，且沒有層數的限制（在記憶體範圍之內可無限層延伸）。

二、 完整顯示數學運算式

運用程式的元件與介面，同 XML 的標籤語言結合，已能顯示數學式原本的樣式。相較於一般工程型計算機僅能支援「一行式」的數學式，本程式提供了一個更寬廣的數學式舞台。藉由此，使用者可明瞭地完整呈現數學式原有的風貌。

三、 正確運算並輸出運算式的答案

透過計算機資料結構與演算法的運作，目前已能正確輸出使用者所輸入的運算式之答案。以數學及程式方法，可輸出小數以外的答案。

四、 提供一般數學形式之解（非小數之解）

在提供一般的小數答案之外，另外利用了若干演算法，達到了不同數學形式的輸出。如：分數、根號、指數與函數等，有效地提高了實用性。

陸、 參考資料

一、 小雄資訊服務中心 VB 研究小站 http://vb.infoserv.com.tw/

二、 資料結構：使用 C 語言教學範本 碁峰出版

三、 新觀念的 Visual Basic 6.0 教本/實戰講座 旗標出版

四、 數學程式軟體 Wolfram Mathematica

五、 Mathematical Markup Language (MathML) Version 2.0 http://www.w3.org/TR/MathML2/

六、 Visual Basic 6 程式設計 18 堂特訓教材 文魁出版

評語

本作品以一般計算器的數學公式表達不夠”人性化”與”自然”出發，開發一套軟體系統，使得數學式的表現與教科書一樣自然美觀，同時可減少輸入的時間達 20%以上。

本作品的輸出，將來可直接 Cut & Paste 至 WORD 或 PowerPoint 等軟體，使得 Equation Editor 更爲自然。

綜而言之，本作品有創意、有實做，而且有實用價值，故推薦之。