1.安裝步驟

http://maxima.sourceforge.net/

→Download

→Sourceforge download page

→Maxima-Windows

→5.19.2-Windows

→Maxima-5.19.2.exe

2.Maxima 簡介

Maxima 是一個所謂的「電腦代數系統」（Computer Algebra System, CAS），什麼是 Computer Algebra System？簡單的說，就是可以幫你解代數方程式的解的程式！可以幫你解代數題目，舉凡解一元 N 次方程式、二元一次方程式、N 元 N 次方程式、因式分解、求導函數、求積分(當然反導函數要存在)等等，都沒問題。搭配上 Gnuplot 還可以幫你繪出 2D 和 3D 的函數圖。Windows 版的執行檔叫做 wxMaxima。

Macsyma 是由 MIT (麻省理工學院，Massachusetts Institute of Technology) 在美國 APAR (Advanced Research Projects Agency，這個計劃最成功的成果就是你現在正在用的 Internet) 計劃和能源署贊助下，在 1960s' 年代所成立的 MAC project 的目標。在那個美好的年代，只要政府出錢的計劃，所有的美國大學都可以享用成果。因此，UC Berkery (加州柏克萊大學) 也拿到了一份原始碼烤貝，就像當初柏克萊電機系拿到 Unix 的原始碼時狀況一樣，柏克萊大學也有人不但開始試著維護，還試著改良它，這個人就是 Bill Schelter 教授。但是這套好用的 CAS 免不了最後還是成爲 MIT 賺取學校經費的主要來源方法之一：把它賣給想要商品化的公司，而且要求拿到原始碼烤貝的其它單位不准再使用及修改它。幸好 Schelter 教授不接受這樣的要求，不但沒有銷毀它，還在私底下默默地改良它，把原本 Macsyma 的 bugs 修掉，並一直嘗試重寫，試圖將它改成 open source 軟體。這項努力一直到 1998 年才達成，並且把名稱改成 Maxima。而 Bill Schelter 教授卻不幸在 2001 年去世，但是他留下了這套功能驚人的軟體。事實上，在 1998 年至今，Maxima 仍然經過不少的修正，邏輯性也比 2001 年時期好很多。

3.基本概念

Maxima 當計算機

以 maxima 來做計算：

```
(%i1) 1+1;
 (%o1) 2
```

出現紅色-->符號，再輸入指令後輸入分號，結束時要打上分號「；」讓 Maxima 知道我們下的指令已結束，再按 Ctrl+enter 執行

(%i1)指的是輸入(input)第 1 條，(%o1)指的是輸出(output)第 1 條。

進階的運算：

```
(%i3) 7/3;
```

$$(\%o3)\ \frac{7}{3}$$

```
(%i4) 1/2+2/3;
```

$$(\%o4)\ \frac{7}{6}$$

從 (%o3)我們看到，$\frac{7}{3}$ 這種運算， Maxima 不是告訴我們 2.3333…，而是分數的形式！這就是所謂電腦代數系統 (CAS) 的特長。

如果想把分數轉換爲小數，指令爲 float：

```
(%i5) float(7/3);
 (%o5) 2.333333333333334
```

浮點數指令：float

再介紹指數，根號，階乘表示法：

```
(%i6) 2^10;
 (%o6) 1024
```

指數指令：^

(%i7) sqrt(9);　　平方根指令：sqrt

```
(%o7) 3
```

除了平方根有特別指令外，其餘皆爲^形式，例如：$3^{1/3}$　　指令爲：3^(1/3)

(%i8) 5!;　　階乘指令：!

```
(%o8) 120
```

指令結尾

在 Maxima 下指令，結束時一定要打上分號「;」，讓 Maxima 知道我們下的指令已結束。爲什麼要多這一個動作，主要是爲了有時打比較長的指令可以換行之故。

另一個結束方式是打入「 $ 」的符號。不同於分號的地方是「運算結果不會顯示出來」：

(%i9) 1+4$

(%i10) 1+4;

```
(%o10) 5
```

結果的引用

我們時常會需要引用前面的結果，這時就用百分比符號 「%」。比方說：

(%i11) 7/3;

$$(\%o11)\ \frac{7}{3}$$

(%i12) float(%);

```
(%o12) 2.333333333333334
```

有時運算式子相距很遠，這時自己定一個標籤可能是最好的方式，比較容易辨識：

```
(%i14) myresult:2+5*4/3;
```

(%o14) $\frac{26}{3}$

```
(%i15) float(myresult);
(%o15) 8.666666666666666
```

```
(%i19) float(%o14);
(%o19) 8.666666666666666
```

重要常數

Maxima 當然有內建 e 或是 π 常常用到的數，只是表示法奇怪一點。 e 是 %e 而 π 是 %pi。

Constant	Description
%e	Base of the natural logarithms (e)
%i	The square root of (-1) (i)
%pi	The transcendental constant pi (π)
%phi	The golden mean $(1+\sqrt{5})/2$
%gamma	The Euler-Mascheroni constant
inf	Real positive infinity (∞)
minf	Real negative infinity $(-\infty)$

定義變數

Maxima 定義變數會給定一個標籤，表示後面的某個數字、矩陣，或想要定義的任何式子等等。例子：

```
(%i16) a:37;
(%o16) 37
```

```
(%i17) a;
 (%o17) 37

(%i18) b:22+89*(100-82);
 (%o18) 1624

(%i19) a+b;
 (%o19) 1661
```

函數

Maxima 函數的定義和使用：

```
(%i20) f(x):=3*x^2+5;
                                  2
 (%o20)                  f(x):=3 x +5

(%i21) f(2);
 (%o21) 17

(%i22) g(x,y):=sin(x)*cos(y);
 (%o22)                  g(x, y):=sin(x) cos(y)

(%i23) g(2*%pi,4);
 (%o23) 0
```

重點就是，在定義函數時要用「:=」去定義。定義變數跟定義函數是大不相同的，通常我們常將數學式存放到變數，方便後面操作調用，而函數是視需要而採用:

4.進階使用

列式而不運算

積分的計算：　積分指令：integrate(數式，變數，範圍)

(%i1) integrate(x^2,x,0,1);

$$(\%o1)\ \frac{1}{3}$$

若我們只是要列出式子。就加個 ‘ 號在前面，例如：

(%i2) 'integrate(x^2,x,0,1);

$$(\%o2)\ \int_0^1 x^2\,dx$$

快捷列：微積分→Integrate→數式：x^2

變數：x

☑定積分：從 0 到 1

Kill 指令

kill 指令可以幫助我們將全部計算過的變數函數歸零。從以下的例子可看出：

(%i24) f(x):=3*x^2+5;

$$(\%o24)\ f(x):=3x^2+5$$

(%i25) f(x);

$$(\%o25)\ 3x^2+5$$

(%i26) kill(all);

(%o0) done

(%i1) f(x);

(%o1) f(x)

5.微積分指令

極限指令

limit(數式，極限變數，範圍)

快捷列：微積分→Find Limit→數式：x^2

變數：x

點：0　特殊：$\pi,\infty,-\infty$

方向：2 邊、左極限、右極限

範例：

$$\lim_{x \to 3} \frac{x^2+x-12}{x-3}$$

```
(%i1) f:(x^2+x-12)/(x-3);
```

$$(\%o1)\quad \frac{x^2+x-12}{x-3}$$

```
(%i2) limit(f,x,3);
(%o2)  7
```

```
(%i3) 'limit(f,x,3);
```

$$(\%o3)\quad \lim_{x \to 3} \frac{x^2+x-12}{x-3}$$

若取右極限指令爲 **plus**，左極限指令爲 **minus**

(%i4) 'limit(f,x,3,plus);

(%o4) $\lim_{x \to 3+} \frac{x^2+x-12}{x-3}$

(%i5) limit(f,x,3,plus);

(%o5) 7

(%i6) 'limit(f,x,3,minus);

(%o6) $\lim_{x \to 3-} \frac{x^2+x-12}{x-3}$

(%i7) limit(f,x,3,minus);

(%o7) 7

無限大或負無限大時，用「inf」和「minf」來表示

(%i2) 'limit(1/x^8,x,inf);

(%o2) $\lim_{x \to \infty} \frac{1}{x^8}$

(%i3) 'limit(1/x^8,x,minf);

(%o3) $\lim_{x \to -\infty} \frac{1}{x^8}$

微分指令

微分指令：diff（數式，變數，次數）

快捷列：微積分→Differentiate→數式：x^5

變數：x

次數：2

(%i1) diff(x^5,x); //若不特別指定次數，maxima 一律視爲 1 次微分

(%o1) $5\,x^4$

(%i2) diff(x^5,x,2);

(%o2) $20\,x^3$

6.線性代數指令

解線性方程組

線性代數的核心問題，就是解線性方程組。解線性方程組可以用 solve 指令來解。

快捷列：方程式→Solve Linear System→方程式個數→方程式 1：x+2*y+3*z=6

方程式 2：2*x-3*y+2*z=14

方程式 3：3*x+y-z=-2

變數：x，y，z

線性方程組：

$$x+2y+3z=6$$

$$2x-3y+2z=14$$

$$3x+y-z=-2$$

我們可以用 solve 指令來解：

(%i1) eq1:x+2*y+3*z=6;

(%o1) $3\,z+2\,y+x=6$

(%i2) eq2:2*x-3*y+2*z=14;

(%o2) $2\,z-3\,y+2\,x=14$

```
(%i3) eq3:3*x+y-z=-2;
  (%o3)  -z+y+3 x=-2

(%i4) solve([eq1,eq2,eq3],[x,y,z]);
  (%o4)  [[x=1,y=-2,z=3]]
```

令一種指令是將上式合併來解，就不用給定方程式名稱，通常用在方程式少或簡單時：

```
(%i1) solve([x+2*y+3*z=6,2*x-3*y+2*z=14,3*x+y-z=-2],[x,y,z]);
  (%o1)  [[x=1,y=-2,z=3]]
```

矩陣及向量

定義矩陣的指令爲 matrix，給定列向量即可求出：

快捷列：代數→Enter Matrix→列：3
行：3
類型：一般、對角、對稱、反對稱
Name：A
→輸入矩陣：將元素一個個填入

```
(%i1) A:matrix([1,2,3],[4,5,6],[7,8,9]);
        [ 1  2  3 ]
        [         ]
  (%o1) [ 4  5  6 ]
        [         ]
        [ 7  8  9 ]

(%i2) v:[2,3,5];
  (%o2)  [2,3,5]
```

「行向量」表示方式。我們可以用下面兩種不同的方式達成：

1. 將列向量作轉置
2. 定義每個列向量元素都是 1 個

```
(%i3) transpose([2,3,5]);
```

$$(\%o3)\quad \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

或者

```
(%i4) matrix([2],[3],[5]);
```

$$(\%o4)\quad \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

一般來說因爲矩陣乘法的關係, 我們寫成列向量和行向量差別很大。不過 Maxima 其實不太在意這點: 它可以聰明地發現你要做的事, 並且正確得計算出來！簡單的說, 一般而言, 我們不需要麻煩得定義行向量, 用列向量即可。

矩陣的表示與截取

矩陣的抽象表示和取出一個矩陣行，列。這在很多理論和計算的嘗試會用到。

Maxima 是一個 CAS 系統，所以我們可以完全用符號去定義一個矩陣，比方說：

```
(%i5) A:matrix([a[1,1],a[1,2]],[a[2,1],a[2,2]]);
```

$$(\%o5)\quad \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$

你也可以做完全抽象的代數計算：

```
(%i6) c*A;
```

$$(\%o6)\quad \begin{bmatrix} a_{1,1}\,c & a_{1,2}\,c \\ a_{2,1}\,c & a_{2,2}\,c \end{bmatrix}$$

現在，我們重新把 A 定義成一個實數矩陣，再看看怎麼樣找出 A 的某一列，某一行，或某個 entry。

```
(%i1) A:matrix([1,2,3],[4,5,6],[7,8,9]);
```

$$(\%o1)\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

```
(%i2) row(A,1);
```

$$(\%o2)\begin{bmatrix}1 & 2 & 3\end{bmatrix}$$

```
(%i3) col(A,2);
```

$$(\%o3)\begin{bmatrix}2\\ 5\\ 8\end{bmatrix}$$

```
(%i4) A[2,3];
(%o4) 6
```

矩陣相關函數

用 Maxima 計算矩陣的行列式值，求轉置矩陣, 矩陣的秩：

```
(%i1) A:matrix([1,-2,1],[2,8,4],[3,3,9]);
```

$$(\%o1)\begin{bmatrix}1 & -2 & 1\\ 2 & 8 & 4\\ 3 & 3 & 9\end{bmatrix}$$

```
(%i2) determinant(A);
```

行列式指令：**determinant**

```
(%o2) 54
```

```
(%i3) rank(A);
```

秩的指令：**rank**

```
(%o3) 3
```

用 Maxima 求反矩陣：

```
(%i5) invert(A);
```

反矩陣指令：invert

$$(\%o5)\quad \begin{bmatrix} \frac{10}{9} & \frac{7}{18} & -\frac{8}{27} \\ -\frac{1}{9} & \frac{1}{9} & -\frac{1}{27} \\ -\frac{1}{3} & -\frac{1}{6} & \frac{2}{9} \end{bmatrix}$$

使用模組

用了 Maxima 一陣子，你可能會預期它該會的都會。比方說求一個矩陣的 trace，這應該夠容易了吧？

事情並不是那麼簡單。Maxima 本身是「不會」算 trace 的！當然我們可以自己寫個小程式，不過先別急。我們可以使用使用適當的模組來做這件事。

所謂模組就是一段小程式，通常是增加一些指令，供你使用。你也許會覺得奇怪，那爲什麼 Maxima 不一開始就把這些模組都加進來？那是因爲如此一來太佔用記憶體，也許很多對某些人重要的指令你永遠也不用去用！

我們要算一個矩陣的 trace，要使用 `nchrpl` 這個模組，這個模組提供了 `mattrace` 指令去計算 trace。

錯誤的作法

```
(%i1) A:matrix([1,-2,1],[2,8,4],[3,3,9]);
```

$$(\%o1)\quad \begin{bmatrix} 1 & -2 & 1 \\ 2 & 8 & 4 \\ 3 & 3 & 9 \end{bmatrix}$$

```
(%i2) mattrace(A);          //並不會去計算矩陣 A 的 trace
```

```
(%o2) mattrace([ 1  -2  1 ])
               [ 2   8  4 ]
               [ 3   3  9 ]
```

使用的方法如下，先讀入 nchrpl 模組，接著就可以使用這個模組提供的指令：

```
(%i3) load("nchrpl");          //讀取 nchrpl 模組
  (%o3)
C:/PROGRA~1/MAXIMA~1.2/share/maxima/5.19.2/share/matrix/nchrpl.mac
```

```
(%i4) mattrace(A);
  (%o4) 18
```

7.繪圖指令

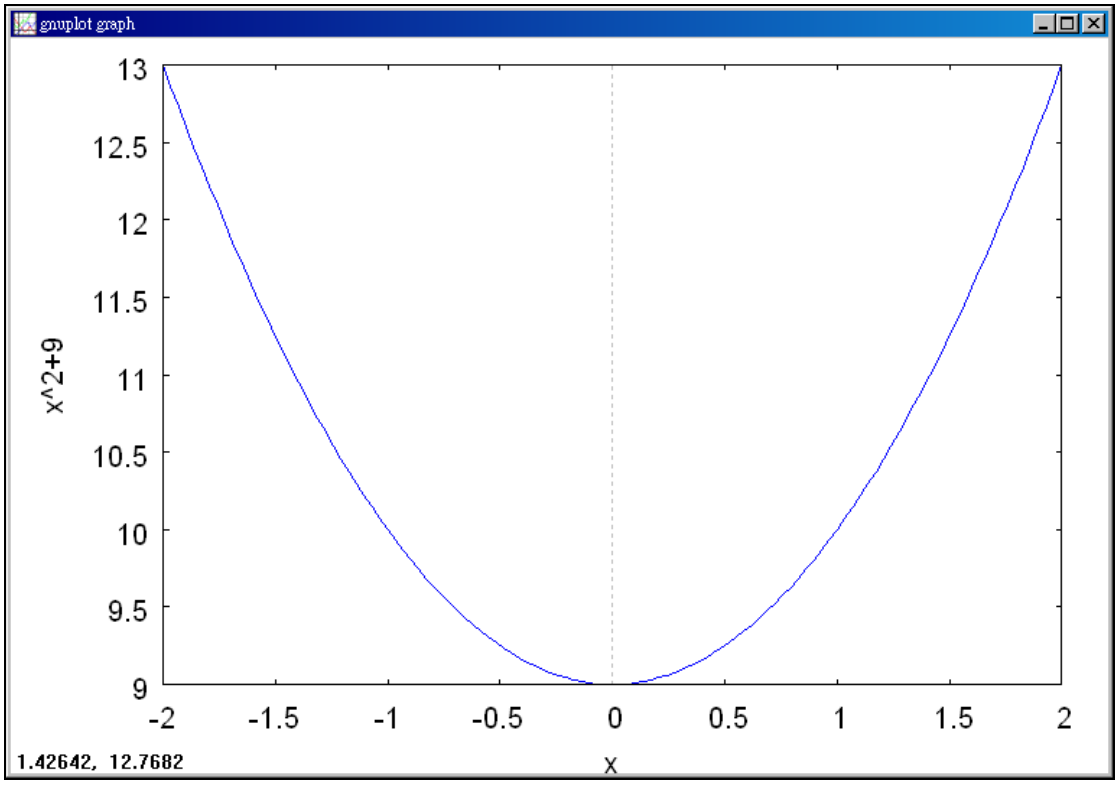
maxima 是免費的自由軟體，但是功能絲毫不比市面上的專業軟體遜色，接著我就要介紹 maxima 的繪圖功能。

maxima 的繪圖功能是使用「GunPlot」，Linux 的使用者應該不陌生，它可以讓使用者藉由輸入函數的方式來產生 2D 和 3D 圖形。

二維繪圖

Maxima 二維繪圖的指令是用 plot2d。比方說，我們要畫 $-3x^5+5x^3$ 這個函數，設定 x 軸的範圍是從 -5 到 5，就下這個指令：

```
(%i1) plot2d([-3*x^5+5*x^3],[x,-2,2]);
```



2D 繪圖快捷列：繪圖→Plot 2d→數式： $-3x^5+5x^3$

變數：x，從-2 到 2

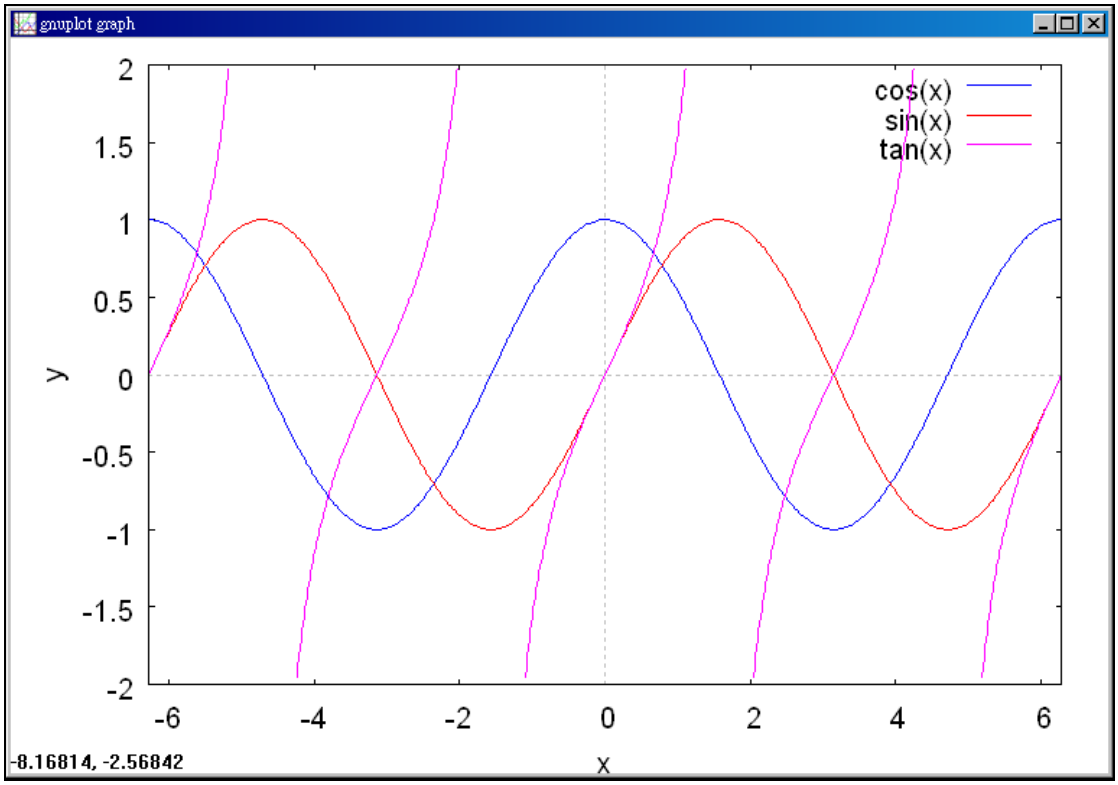
變數：y ，y 的範圍

多個函數的繪圖

如果要比較幾個函數，要如何下指令呢？我們來看個例子就明白了：

(%i4) plot2d([cos(x),sin(x),tan(x)],[x,-2*%pi,2*%pi],[y,-2,2])$

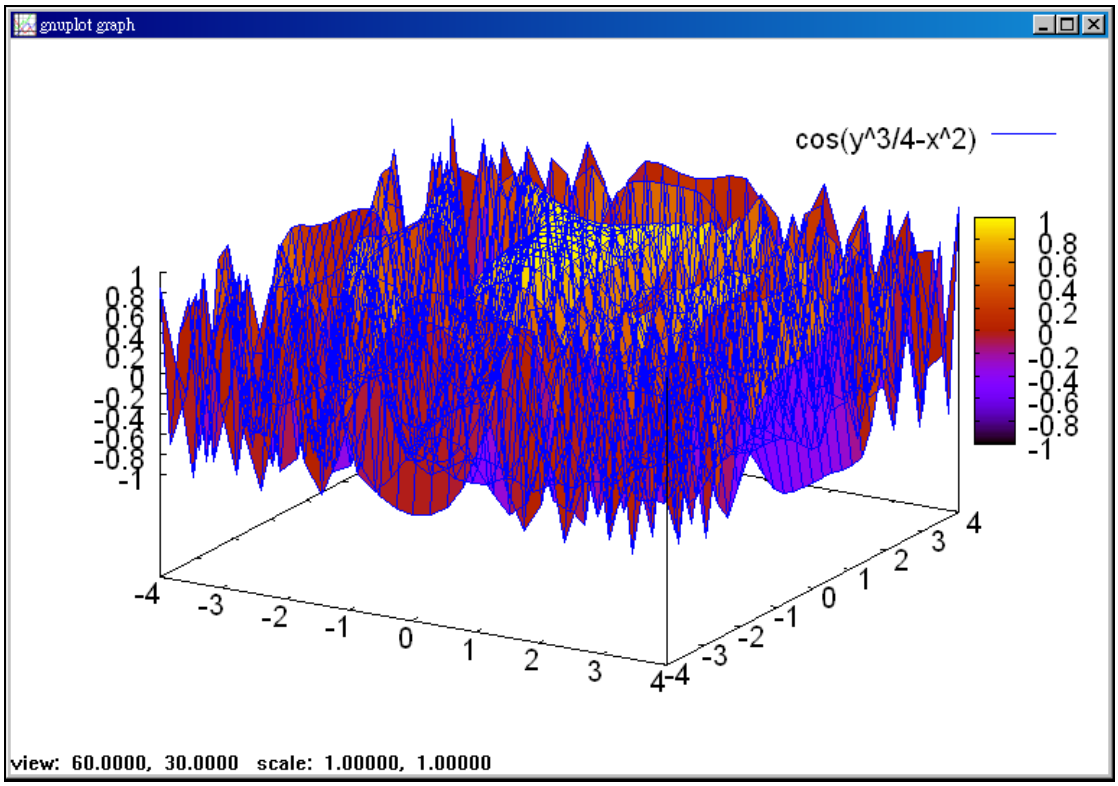
這個例子會同時畫出cos(x) ， sin(x) 和tan(x) 的圖形。



三維繪圖

三維繪圖也一樣容易，只要改用 plot3d 的指令即可，3D 和 2D 的繪圖指令，大同小異，不同的是，在 2D 只要指定 x 軸(平行軸)的範圍，而 3D 則是兩個軸的範圍都要指定：

```
(%i2) plot3d(cos(-x^2+y^3/4),[x,-4,4], [y,-4,4]);
```



maxima 的 3D 功能強大，你可以在圖形上按住滑鼠左鍵，移動滑鼠，就可以旋轉圖形

參考資料：蔡炎龍(2006)，maxima 在線性代數的應用，政治大學應用數學系。

詹勳國(2009)，MAXIMA 學習手冊，屏東教育大學應用數學系。

http://www.cheeren.com/?q=node/125