Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

ÁLGEBRA Y GEOMETRÍA ANALÍTICA EN wxMaxima

Segunda Parte

Laboratorio de Matemática
UDB Matemática
2011

<u>Realizado por:</u> Daniela Scarcella
<u>Supervisado por:</u> Susana Estévez

Cónicas

En Maxima las cónicas se grafican en el plano usando las ecuaciones paramétricas como también las ecuaciones implícitas. Para graficar, como vimos anteriormente, se pueden utilizar el menú desplegable *Plot* como también los comandos `plot2d` y `draw2d`. Vamos a ver cómo se grafican las distintas cónicas con el comando `draw2d`.

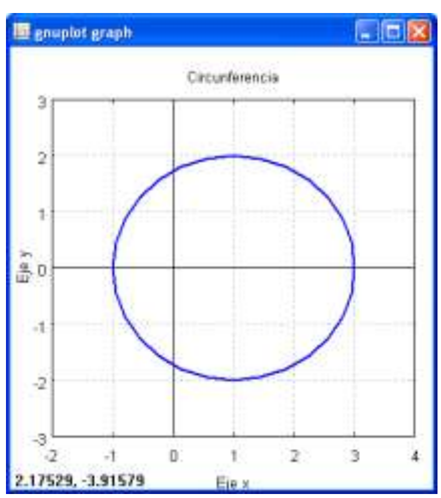
Circunferencia

La ecuación cartesiana de la circunferencia es de la forma: $(x-a)^2+(y-b)^2=r^2$

Sus ecuaciones paramétricas son: $\overrightarrow{r(t)}\begin{cases} x = a + r\cdot\cos t \\ y = b + r\cdot sent \end{cases}, 0 \leq t < 2\pi$

Ejemplo 1:

```
(%i1)   load(draw);
        draw2d(
                line_width=2,
                color = blue,
                parametric (1+2*cos(t),2*sin(t),t,0,2*%pi),
                grid=true,
                title="Circunferencia",
                xlabel="Eje x",
                ylabel="Eje y",
                xaxis=true,
                yaxis=true,
                xrange=[-3,3],
                yrange=[-3,3])$
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```

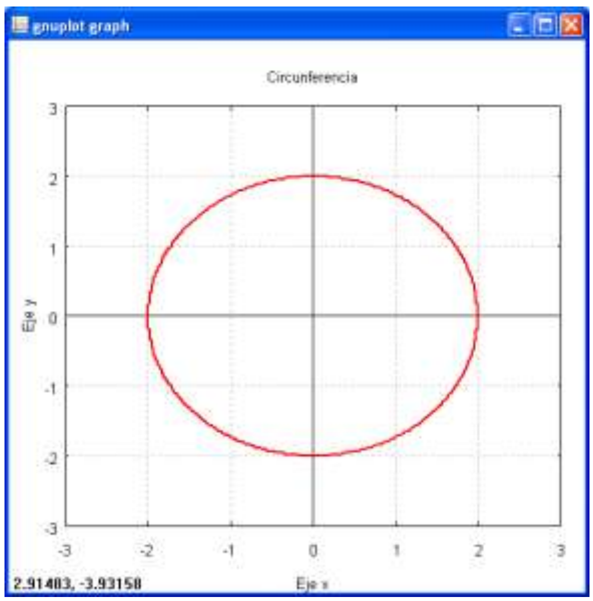


Ejemplo 2:

```
(%i1)   load(draw);

        draw2d(
                line_width=2,
                color=red,
                implicit((x^2)+(y^2)=4,x,-3,3,y,-3,3),
                xlabel="Eje x",
                ylabel="Eje y",
                title="Circunferencia",
                grid=true,
                xaxis=true,
                yaxis=true,
                xrange=[-3,3],yrange=[-3,3])$

(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



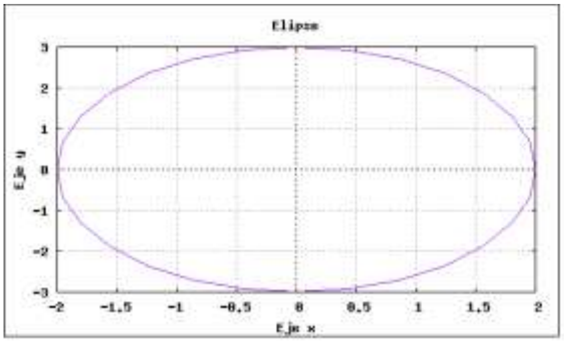
<u>Elipse</u>

La ecuación cartesiana de la elipse es de la forma: $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \vee \frac{x^2}{b^2}+\frac{y^2}{a^2}=1$

La representación paramétrica es: $\mathrm{E}\begin{cases} x = a\cdot\cos t \\ y = b\cdot sent \end{cases}, 0 \le t < 2\pi$

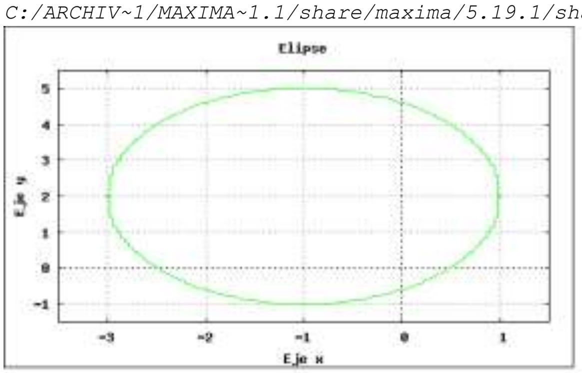
Ejemplo 1:

```
(%i1)   load(draw);
        wxdraw2d(
                color = purple,
                parametric (2*cos(t),3*sin(t),t,0,2*%pi),
                grid=true,
                xlabel="Eje x",
                ylabel="Eje y",
                xaxis=true,
                yaxis=true,
                xrange=[-2,2],yrange=[-3,3],
                title="Elipse")$
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
(%t2)
```



Ejemplo 2:

```
(%i1)   load(draw);
        wxdraw2d(
                color = green,
                implicit ((((x+1)^2)/4)+(((y-2)^2)/9)=1,x,-5,5,y,-5,5),
                grid=true,
                xlabel="Eje x",
                ylabel="Eje y",
                xaxis=true,
                yaxis=true,
                xrange=[-3.5,1.5],yrange=[-1.5,5.5],
                title="Elipse")$
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
(%t2)
```



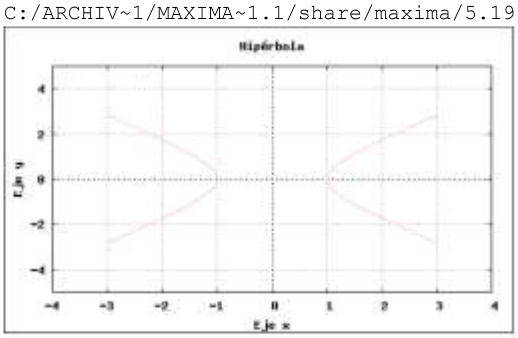
Hipérbola

La ecuación cartesiana de la hipérbola, por lo general, es de la forma:

$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1 \;\vee\; -\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$$

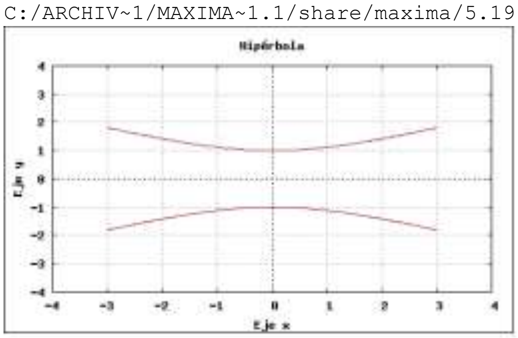
Ejemplo 1:

```
(%i1)   load(draw);
        wxdraw2d(
          color=pink,
          implicit(x^2-y^2=1,x,-3,3,y,-3,3),
          grid=true,
          xlabel="Eje x",
          ylabel="Eje y",
          xaxis=true,
          yaxis=true,
          xrange=[-4,4],yrange=[-5,5],
          title="Hipérbola")$
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
(%t2)
```



Ejemplo 2:

```
(%i1)   load(draw);
        wxdraw2d(
          color=brown,
          implicit(-(x^2)/4+y^2=1,x,-3,3,y,-3,3),
          grid=true,
          xlabel="Eje x",
          ylabel="Eje y",
          xaxis=true,
          yaxis=true,
          xrange=[-4,4],yrange=[-4,4],
          title="Hipérbola")$
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
(%t2)
```



Parábola

La ecuación cartesiana de la parábola, por lo general, es de la forma: $x^2 = 2py \vee y^2 = 2px$

La representación paramétrica es: $\overrightarrow{r(t)} \begin{cases} x = t \\ y = t^2 \end{cases}, \forall t$

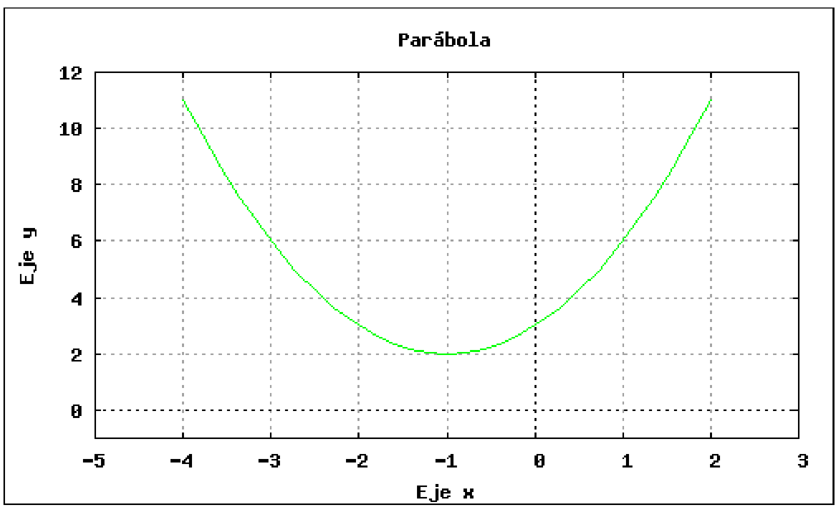
Ejemplo 1:

```
(%i1)   load(draw);

        wxdraw2d(
                color = green,
                parametric (t-1,t^2+2,t,-3,3),
                grid=true,
                title="Parábola",
                xlabel="Eje x",
                ylabel="Eje y",
                xaxis=true,
                yaxis=true,
                xrange=[-5,3],yrange=[-1,12])$

(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```

(%t2)



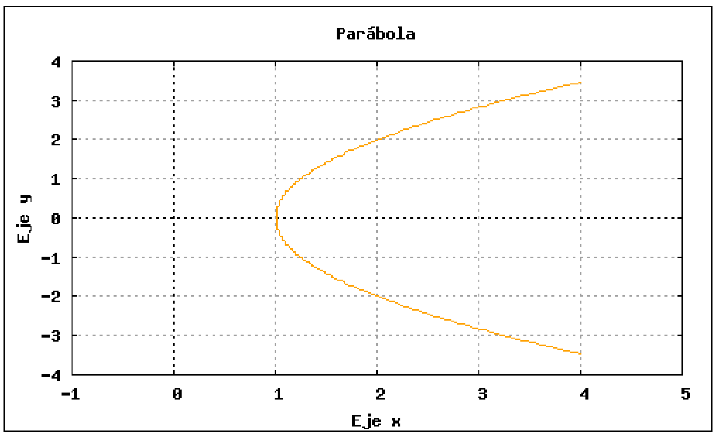
Ejemplo 2:

```
(%i1)   load(draw);

        wxdraw2d(
                color = orange,
                implicit(y^2=4*(x-1),x,-4,4,y,-4,4),
                grid=true,
                title="Parábola",
                xlabel="Eje x",
                ylabel="Eje y",
                xaxis=true,
                yaxis=true,
                xrange=[-1,5],yrange=[-4,4])$

(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```

(%t2)



<u>Aclaración:</u> Si queremos que el gráfico se muestre dentro de la ventana de xwMaxima, debemos escribir antes del comando las letras `wx` (Ver ejemplo anterior). *Para que se muestre en una ventana independiente no se le antepone* `wx` *al comando* (Ver ejercicio 1 de circunferencia).

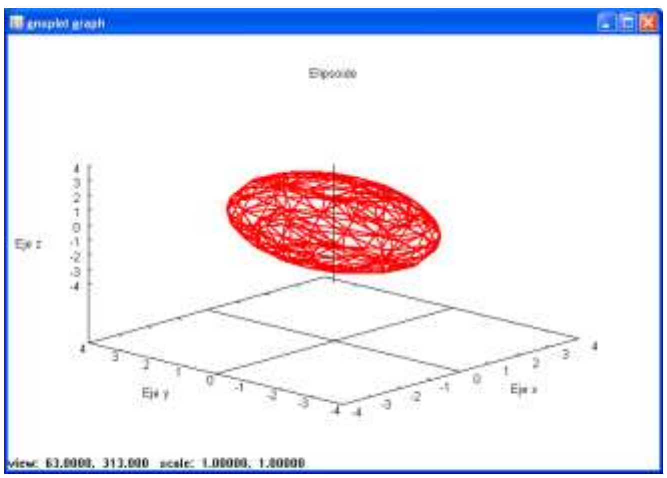
Superficies

Las diferentes superficies también las vamos a graficar con el comando draw3d.

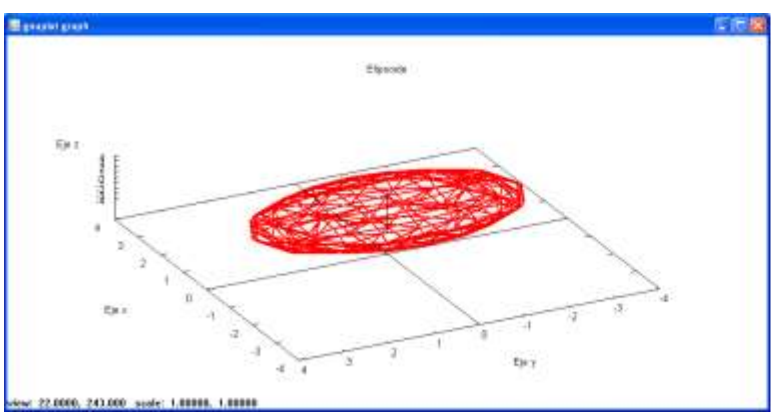
Elipsoide

Sea la ecuación $3x^2 + y^2 + z^2 = 9$ de un elipsoide vamos a graficarlo muy fácilmente:

```
(%i1)   load(draw);
        draw3d(
                line_width=2,
                color=red,
                implicit(3*x^2+y^2+z^2=9,x,-4,4,y,-4,4,z,-4,4),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Elipsoide");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



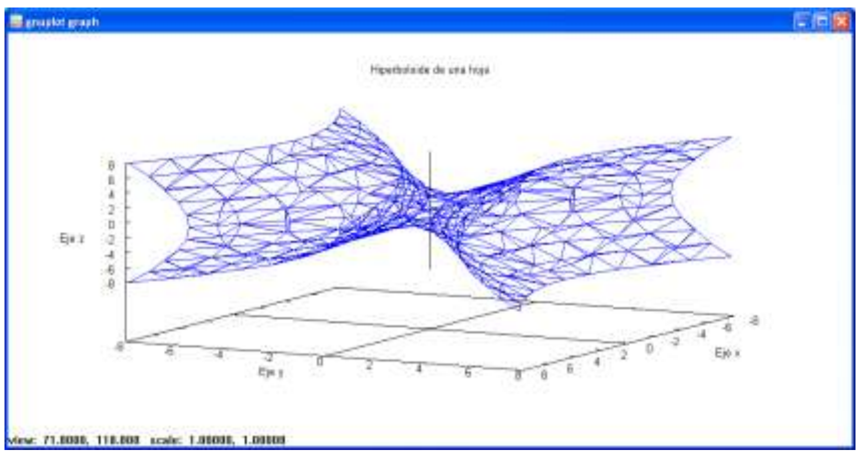
Rotando el gráfico, a continuación observamos otra vista:



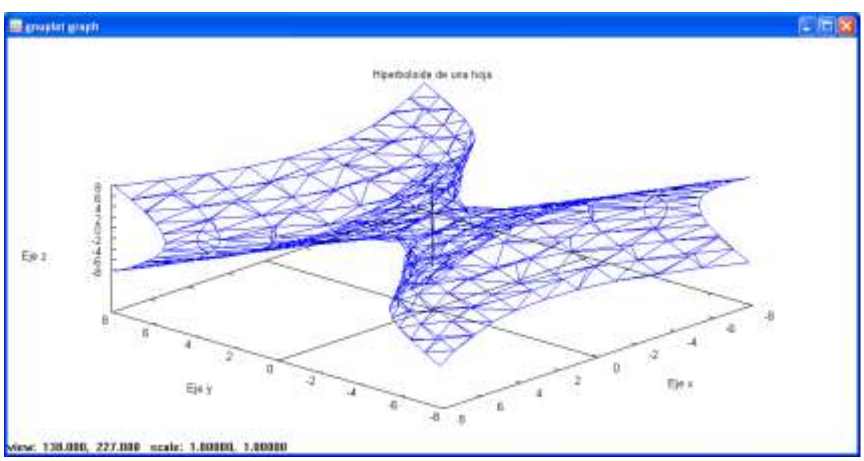
Hiperboloide de una hoja

Sea la ecuación $x^2 - 2y^2 + z^2 = 4$ de un hiperboloide de una hoja vamos a graficarlo en forma similar a la anterior:

```
(%i1)   load(draw);
        draw3d(
                line_width=1,
                color=blue,
                implicit(x^2-2*y^2+z^2=4,x,-8,8,y,-8,8,z,-8,8),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Hiperboloide de una hoja");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



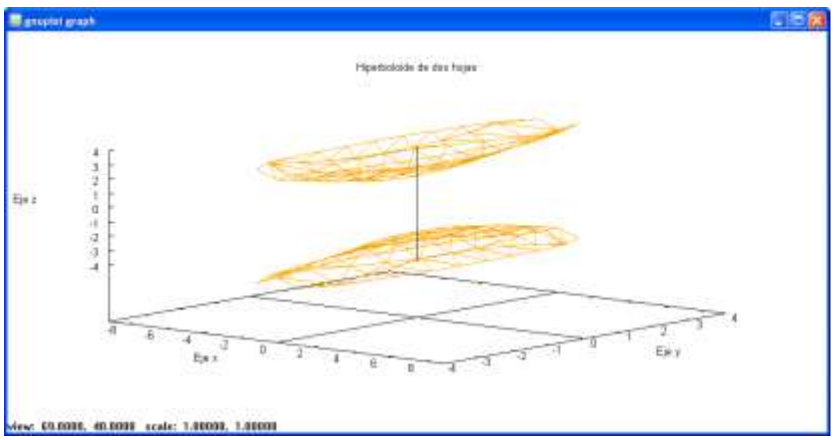
Rotando el gráfico, a continuación observamos otra vista:



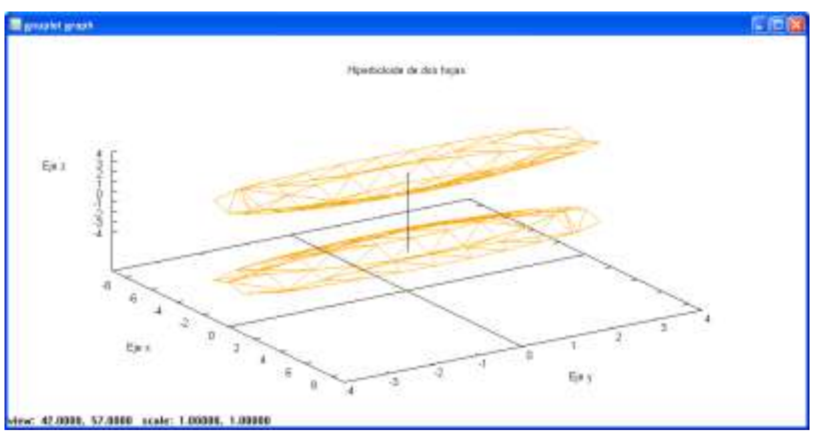
Hiperboloide de dos hojas

Sea la ecuación $-4x^2 - y^2 + 2z^2 - 8 = 0$ de un hiperboloide de dos hojas graficamos:

```
(%i1)   load(draw);
        draw3d(
                line_width=1,
                color=orange,
                implicit(-4*x^2-y^2+2*z^2-8=0,x,-9,9,y,-4,4,z,-4,4),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Hiperboloide de dos hojas");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



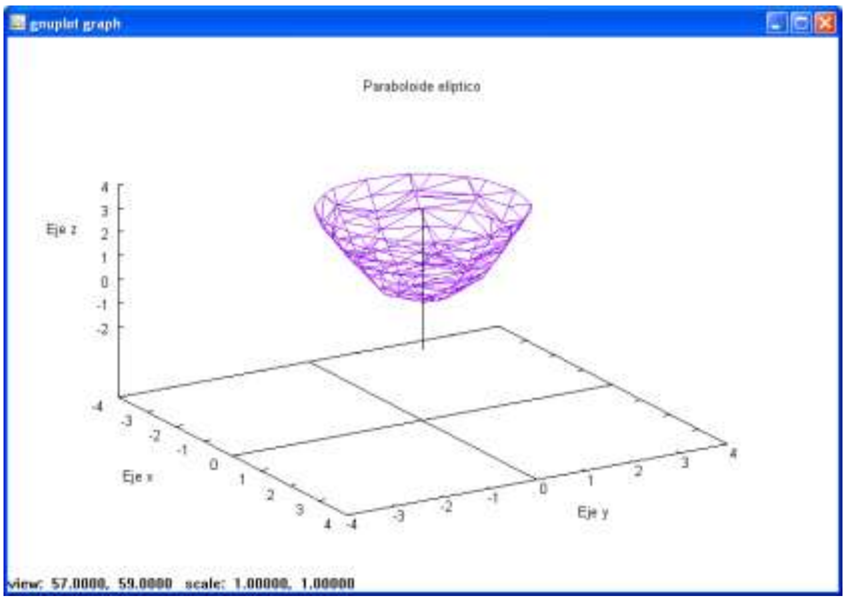
Otra vista:



Paraboloide elíptico

Sea la ecuación $x^2 + y^2 = z$ de un paraboloide elíptico:

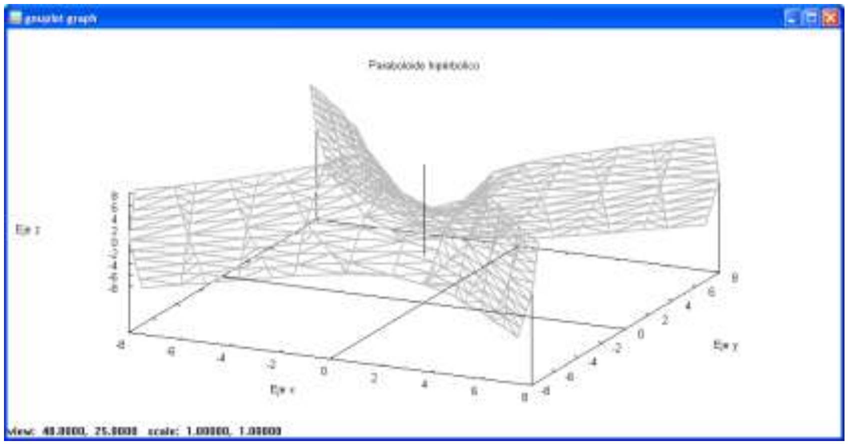
```
(%i1)   load(draw);
        draw3d(
                line_width=1,
                color=purple,
                implicit(x^2+y^2=z,x,-4,4,y,-4,4,z,-2,4),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Paraboloide elíptico");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



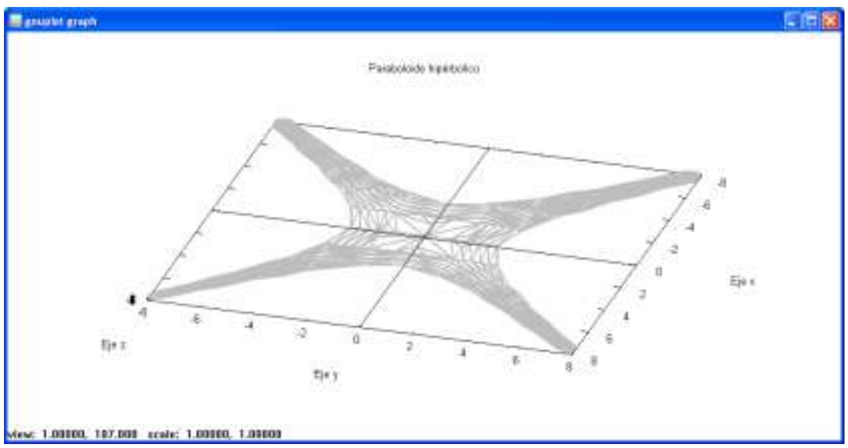
Paraboloide hiperbólico

La ecuación $x^2 - y^2 = z$ es de un paraboloide hiperbólico y se grafica:

```
(%i1)   load(draw);
        draw3d(
                line_width=2,
                color=grey,
                implicit(x^2-y^2=z,x,-8,8,y,-8,8,z,-8,8),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Paraboloide hipérbolico");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



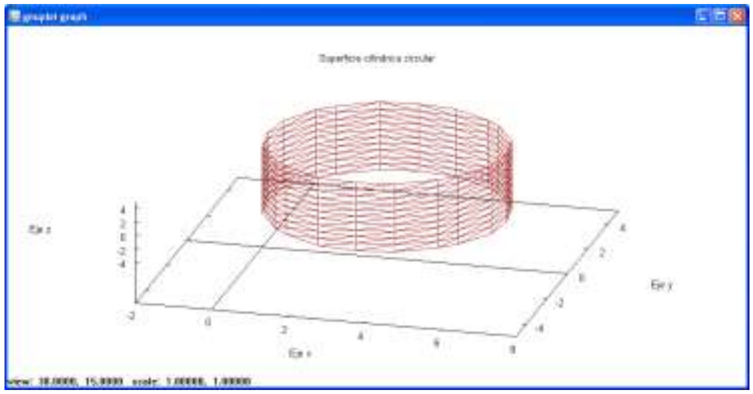
Otra vista rotando el grafico:



<u>Superficie cilíndrica</u>

Sea la ecuación $x^2+y^2+6x-2y=0$ de una superficie cilíndrica circular se grafica de la siguiente manera:

```
(%i1)   load(draw);
        draw3d(
                line_width=1,
                color=brown,
                implicit(x^2+y^2-6*x-2*y=0,x,-2,8,y,-5,5,z,-5,5),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Superficie cilíndrica circular");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```

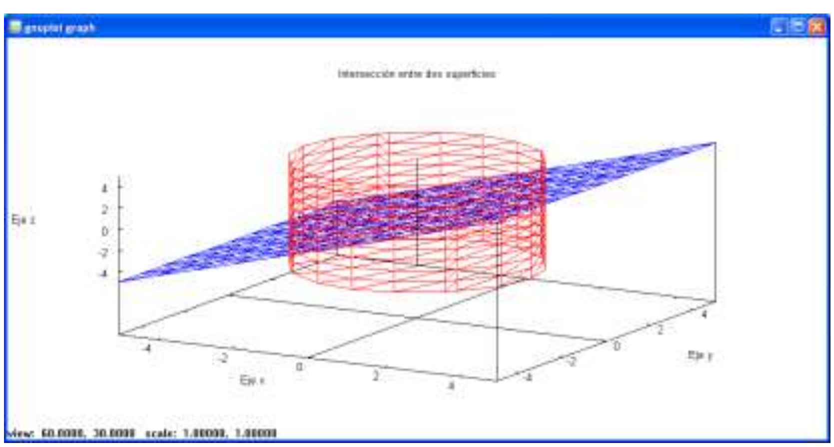


<u>Curvas determinadas como intersección de superficies y Proyecciones sobre los planos coordenados</u>

Ahora vamos a ver con ejemplos cómo se grafican dos superficies para obtener la curva de intersección.

Ejemplo 1: $$C:\begin{cases} z = x \\ x^2 + y^2 = 9 \end{cases}$$

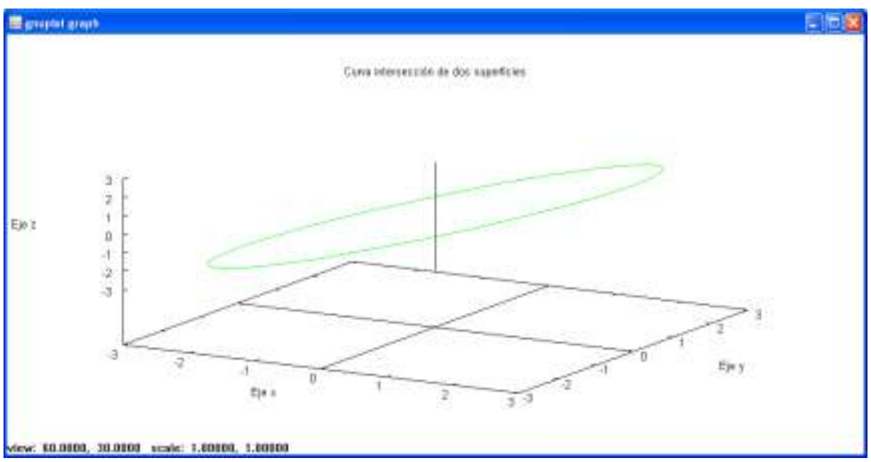
```
(%i1)   load(draw);
        draw3d(
                color=blue,
                line_width=1,
                implicit(z=x,x,-5,5,y,-5,5,z,-5,5),
                color=red,
                line_width=1,
                implicit(x^2+y^2=9,x,-5,5,y,-5,5,z,-5,5),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Intersección entre dos superficies");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



Entonces para graficar solamente la curva intersección utilizamos las ecuaciones paramétricas de C que son:

$$C:\begin{cases} x = 3\cdot\cos t \\ y = 3\cdot sent\ ,0 \le t < 2\pi \\ z = 3\cdot\cos t \end{cases}$$

```
(%i1)   load(draw);
        draw3d(
                color=green,
                line_width=1,
                parametric(3*cos(t),3*sin(t),3*cos(t),t,0,2*%pi),
                xlabel="Eje x",
                ylabel="Eje y",
                zlabel="Eje z",
                xaxis=true,
                yaxis=true,
                zaxis=true,
                title="Curva intersección de dos superficies");
(%o1)   C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.19.1/share/draw/draw.lisp
```



Autovalores y Autovectores

Para realizar el cálculo de los autovalores y autovectores en wxMaxima lo único que debemos hacer es ingresar la matriz y luego utilizar los comandos `eigenvalues(matriz)` y `eigenvectors(matriz)`, respectivamente. O bien, obtener el polinomio característico y resolver la ecuación.

Veamos a través de un ejemplo las distintas maneras de obtener los autovalores y autovectores.

Sea la matriz $A = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 3 & -2 \\ 0 & 0 & 4 \end{pmatrix}$ vamos a ingresarla en Maxima de la misma forma que vimos en la unidad de Matrices. Una vez ingresada la matriz, vamos a seleccionarla y hacer clic en el menú desplegable *Álgebra* y seleccionamos la opción *Valores propios*. A partir de eso, Maxima calcula los autovalores de la matriz. Observemos como se muestra el ejemplo en dicho programa:

```
(%i1)   A: matrix(
         [1,2,-1],
         [0,3,-1],
         [0,0,4]
        );
```

(%o1) $\begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & -1 \\ 0 & 0 & 4 \end{bmatrix}$

```
(%i2)   eigenvalues(matrix([1,2,-1],[0,3,-1],[0,0,4]));
(%o2)   [[1,3,4],[1,1,1]]
```

El comando `eigenvalues` devuelve dos listas, la primera formada por los autovalores de la matriz (en este caso $\lambda_1 = 1$, $\lambda_2 = 3$ y $\lambda_3 = 4$) y la segunda lista por el orden de multiplicidad de cada autovalor respectivamente. Es decir el autovalor $\lambda_1 = 1$ tiene orden de multiplicidad 1, el $\lambda_2 = 3$, 1 y el $\lambda_3 = 4$ también 1.

Ahora calculemos los autovalores de un modo distinto utilizando el comando **`charpoly(A,x)`** que calcula el polinomio característico de la matriz A con incógnita **`x`**. Pero también vamos a utilizar el comando **`expand( )`** para expandir dicho polinomio.

```
(%i3)   expand(charpoly(A,x));
(%o3)   (1-x)(3-x)(4-x)
```

Una vez que se obtiene dicho polinomio ingresamos el comando **`solve( )`** y se obtienen los distintos autovalores de la matriz:

```
(%i4)   solve(%);
(%o4)   [x=3,x=4,x=1]
```

<u>Aclaración:</u> Recordemos que el símbolo % representa a la salida anterior.

Ahora calculemos los autovectores, seleccionamos la matriz A haciendo un clic sobre ella y luego accedemos al menú desplegable *Algebra* y seleccionamos la opción *Vectores Propios*. Veamos que devuelve:

```
(%i3)   eigenvectors(matrix([1,2,-1],[0,3,-1],[0,0,4]));
(%o3)   [[[1,3,4],[1,1,1]],[[[1,0,0]],[[1,1,0]],[[1,1,-1]]]]
```

El comando `eigenvectors` devuelve una lista formada por los autovalores junto a su orden de multiplicidad y a sus autovectores asociados. En el ejemplo, para el autovalor $\lambda_1 = 1$ con orden de multiplicidad 1, el autovector asociado es $v_1 = (1,0,0)$, para $\lambda_2 = 3$ con orden de multiplicidad 1 el autovector es $v_2 = (1,1,0)$ y para $\lambda_3 = 4$ también con orden de multiplcidad 1 el autovector asociado es $v_3 = (1,1,-1)$.

Entonces lo que devuelve `eigenvectors` es:

```
[[[autovalor1,autovalor2,…,autovalorN],[ordenDeMultip1, ordenDeMultip2,…,
ordenDeMultipN]],[[autovector1],[autovector2],[autovector3]]]
```