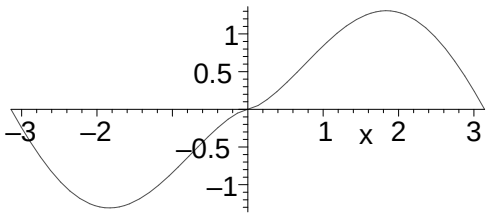
Graphiques avec Maple

En 2 dimensions.

L'instruction de base est plot avec toutes ses possibilités...elle figure dans le noyau de base sans avoir à charger de librairies supplémentaires

```
> f := x -> sqrt(abs(x))*sin(x):plot(f,-2*Pi..2*Pi):
> plot(f(x),x=-Pi..Pi,scaling=constrained);
```



Dans la première expression on ne précise pas la variable, car f est une fonction d'une seule variable... dans la deuxième on considère une expression et alors il faut indiquer la variable. En outre l'option scaling=constrained force Maple a utiliser un repère orthonormé, sinon Maple ajuste au mieux l'échelle sur Oy (parait-il !!?)

Dès que le curseur se trouve dans une zone de graphique, de nouvelles icones apparaissent sur la barre de menu permettant de choisir (ou modifier) le style du tracé : toutes ces icones correspondent aussi à des options Maple, mais certaines options ne correspondent à aucune icone...

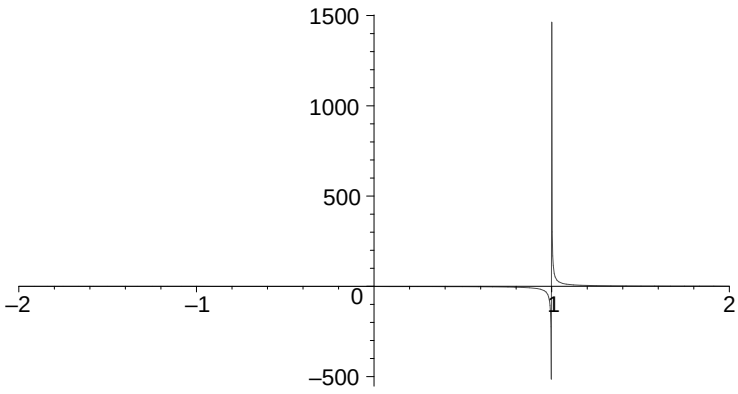
```
> plot(f(x),x=-Pi..Pi,-2..2): plot(f(x),x=-Pi..Pi,beurk=-2..2):
```

La première forme réalise le tracé avec fixation des unités sur Oy, la deuxième permet en outre d'affecter un label a l'axe... généralement mal placé, empiètant sur les graduations etc...

Attention, si x ou beurk ne sont pas libres...

```
> y := 3: plot(f(x),x=-Pi..Pi,y=-2..2);
Error, (in plot) invalid arguments
```

```
> f := x -> 1/(x-1): plot(f,-2..2);
```



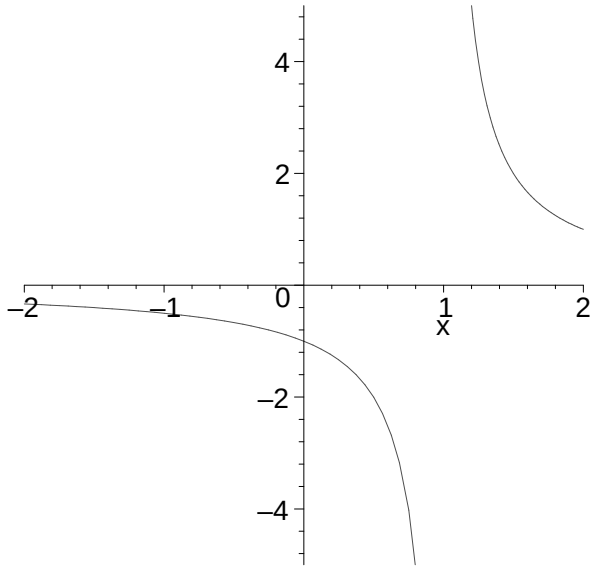
Maple fixe automatiquement et donc stupidement les échelles...il est alors obligatoire de le faire

soi-même.

```
> plot(f,-2..2,-5..5):
```

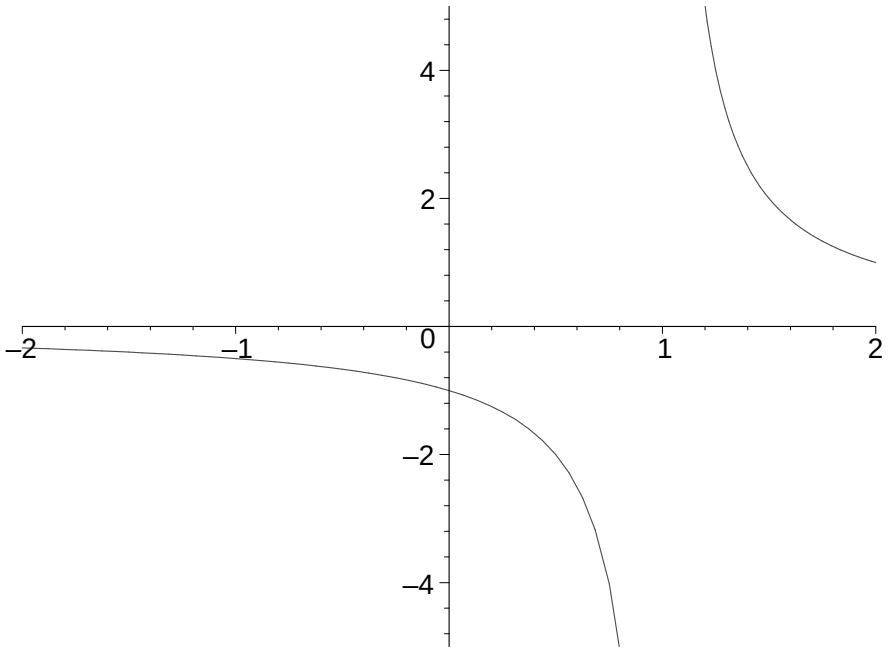
Maple ne vous fournit pas le tracé de l'asymptote verticale en prime...simplement il passe sur la discontinuité sans lever le crayon !

```
> plot(f(x),x=-2..2,-5..5,discont=true);
```



L'option discont oblige Maple a tester la continuité de l'expression avant le tracé !
Remarquez que cela ne marche pas avec une fonction !!??

```
> plot(f,-2..2,-5..5,discont=true);
```



Pour écrire la valeur d'une variable dans une chaîne ou dans un titre, il faut la convertir en chaîne et la concaténer avec l'opérateur .

```
> m := 3/2:
> mm := convert(m,string);
```

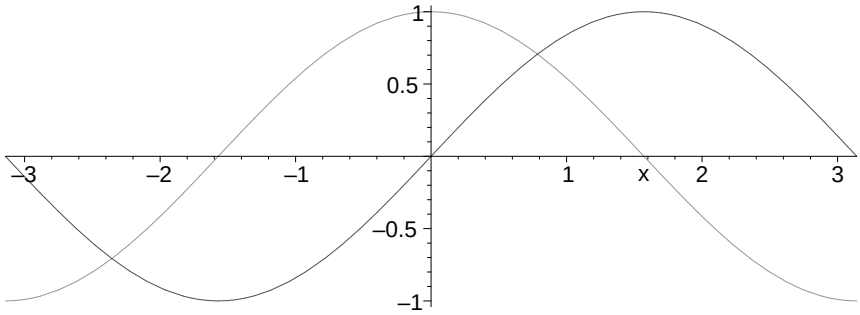
$$mm := \text{"3/2"}$$

```
> plot(sin(m*x),x=-Pi..Pi,title='sin('.mm.'x)'):
```

Notez les guillemets inversés (AltGr 7)! Laid et guère convaincant...et vous pouvez toujours essayer de mettre des accents !

On peut aussi utiliser un ensemble (éviter une liste !) avec plot pour représenter plusieurs fonctions simultanément Pour avoir toutes les courbes tracées en couleur noire utiliser l'instruction color=

```
> plot({sin(x),cos(x)},x=-Pi..Pi);plot({sin(x),cos(x)},x=-Pi..Pi,c
  olor=black):
```



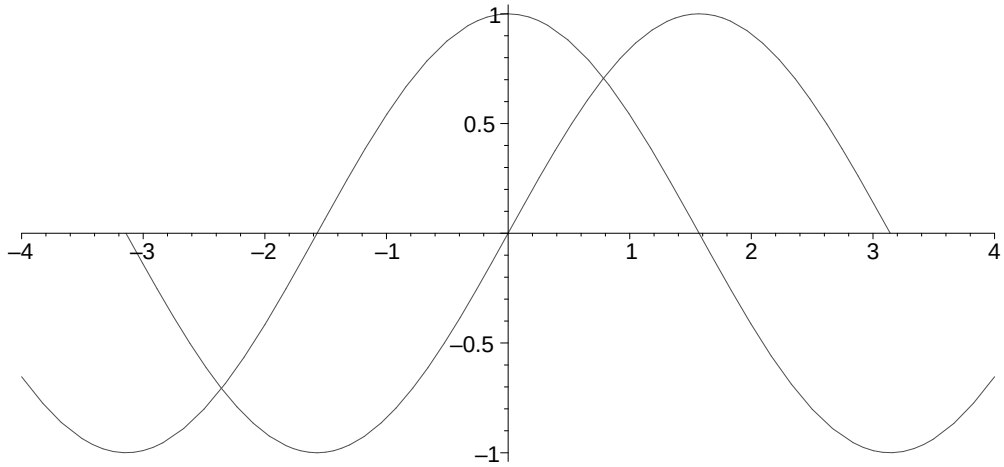
Les instructions suivantes (sauf plotd) nécessitent le chargement de la bibliothéque plots

```
> with(plots);
```

[*animate, animate3d, animatecurve, changecoords, complexplot, complexplot3d, conformal, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot, pareto, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot*]

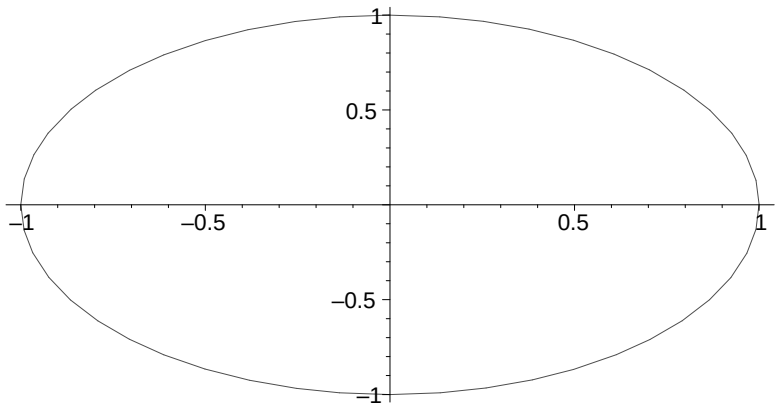
Pour tacer simultanément plusieurs fonctions, ou rajouter des commentaires à un graphique, on peut stocker chaque élément graphique dans une variable et utiliser la fonction display de la bibliothèque plots.

```
> g1 := plot(sin,-Pi..Pi) : g2 := plot(cos,-4..4):
  display({g1,g2});
```



Evitez de mettre un point virgule après l'instruction g1= ... Si vous voulez colorer les courbes, utilisez une instruction color pour chaque graphique. Remarquez que les échelles peuvent etre différentes. Notez que l'intervalle fait partie de la liste, à ne pas confondre avec l'exmple précédent !

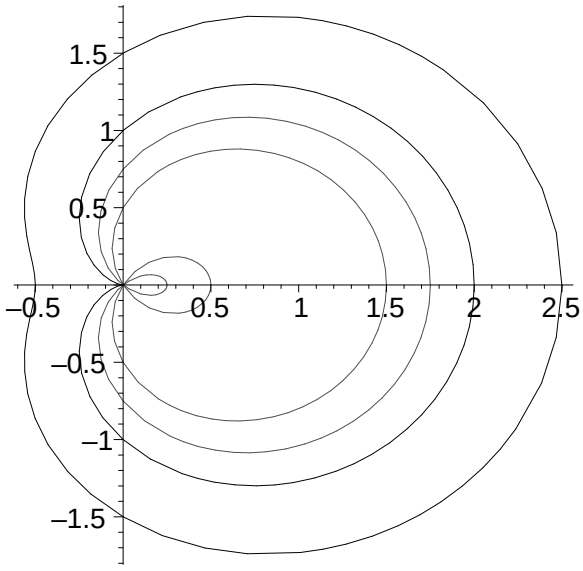
```
> plot([sin(x),cos(x),x=-Pi..Pi]); # Maple interprète ceci comme
  une représentation paramétrique cartésienne.
```




```
> G := a->[(a+cos(t))*cos(t),(a+cos(t))*sin(t),t=-Pi..Pi]:
```

On peut avoir des paramètres dans les graphiques, mais il doivent etre connus au moment du tracé et une fonction peut meme etre à valeurs graphique !

```
> ens_a := {0.5, 0.75,1,1.5}:
> plot(map(G,ens_a),color=[black,red]);
```



Là on utilise une liste de couleurs pour les tracés...

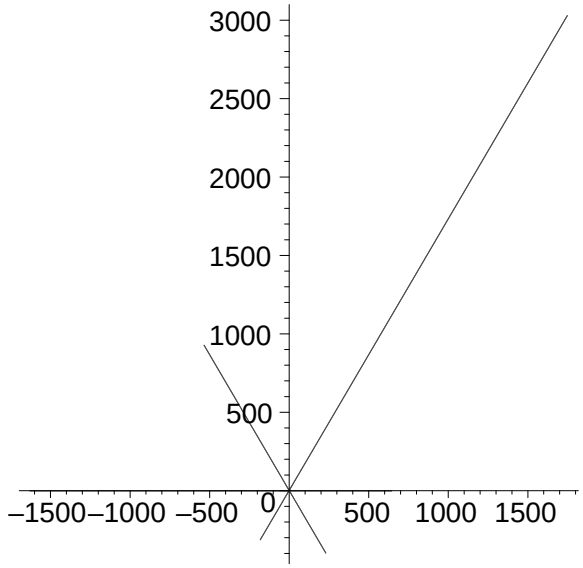
Coordonnées polaires

Bien sûr Maple sait faire des coordonnées polaires tout seul !

plot([p(t),q(t),t=a..b],coords=polar) trace la courbe définie en polaire par rho=p(t) et theta =q(t) pour t dans [a;b] ; généralement t=theta.

Remarquez qu'en cas de discontinuité vous aurez droit à l'asymptote sans possibilité d'utiliser l'option discont !

```
> plot([1/(sin(t)-sin(2*t)),t,t=-Pi/2..Pi/2],coords=polar);
```

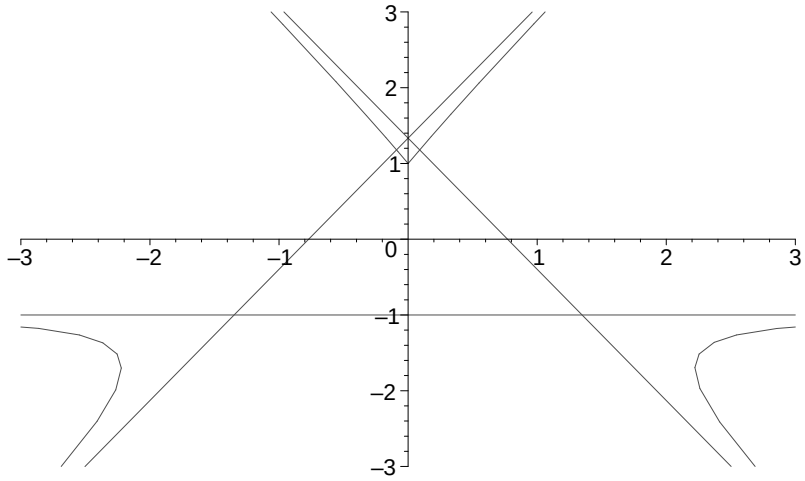


Funny spectacular ou funny ridiculous ?

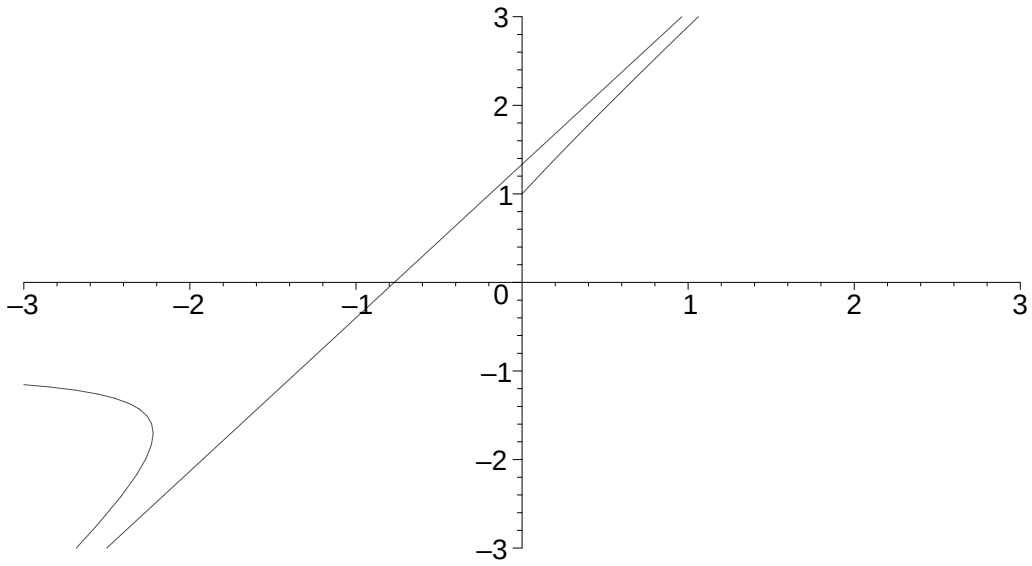
```
> plot([1/(sin(t)-sin(2*t)),t,t=-Pi/2..Pi/2],coords=polar,x=-3..3)
  ;
Error, (in plot/options2d) unknown or bad argument, x = -3 .. 3
```

Les options sont toujours données après les échelles...en outre la Release 4 n'admet plus la syntaxe x=a..b,y=c..d remplacée par l'option view=[a..b,c..d].

```
> plot([1/(sin(t)-sin(2*t)),t,t=-Pi/2..Pi/2],view=[-3..3,-3..3],co
  ords=polar);
```




```
> plot([1/(sin(t)-sin(2*t)),t,t=0..Pi/2],view=[-3..3,-3..3],coords
  =polar);
```



Maple supporte mal la discontnuité en 0 ! ! mais le message n'est guère utile !

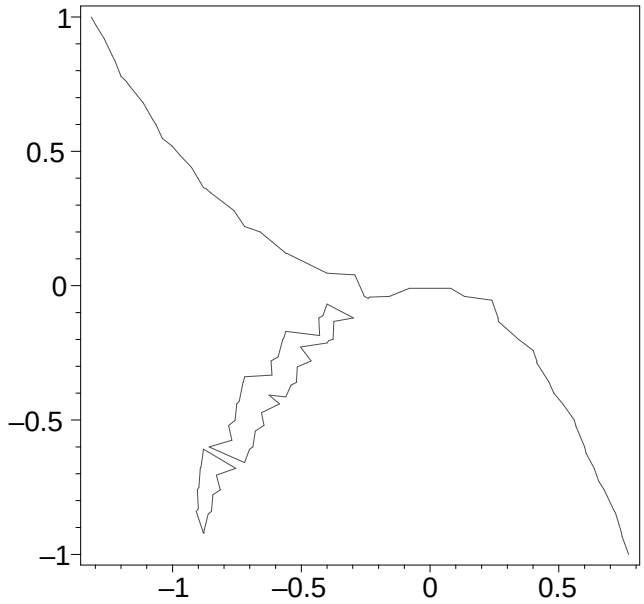
Dans la bibliothèque graphique, il existe une fonction polarplot qui fait la même chose que plot utilisé avec l'option coords=polar.

Fonctions implicites

La fonction qui nous intéresse est implicitplot qui trace les courbes définies implicitement, f=0 est pris par défaut.

```
> f := (x,y)-> x^2+y^2-1:
> implicitplot(f,-2..2,-2..2):
> implicitplot(f(x,y)=2,x=-2..2,y=-2..2):
Error, (in plot/iplot2d/expression) bad range arguments x = -2 .. 2, 3 = -2 ..
2
```

```
> g := (x,y) -> x^7 + 3*x^2*y^2 + 2*y^3:
> implicitplot(g,-2..2,-1..1,axes=boxed);
```



Par défaut Maple divise le rectangle considéré en 25x25 points, ce qui est souvent insuffisant pour avoir un tracé précis. On peut fixer le nombre de points de la grille pour arranger les choses, mais attention au temps de calcul !

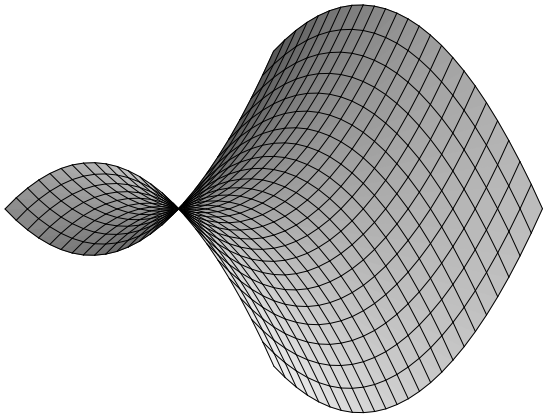
```
> t0 := time():
  implicitplot(g,-2..2,-1..1,grid=[100,100],axes=none):time()-t0;
```

1.891

En dimension 3.

C’est presque pareil...avec beaucoup plus d’options !

```
> restart:with(plots) : plot3d(x^2-y^2,x=-1..1,y=-1..1); # Tracé
  de z = f(x,y)
Warning, the name changecoords has been redefined
```



On peut fixer l'échelle sur Oz avec l'option view=a..b .
Si vous cliquez sur le graphique, de nouvelles icones apparaissent, en outre on peut changer le point de vue ! Les modifications ne sont pas immédiatement affichées, il faut utiliser l'icone **R** (pour redraw).
Admirez la position des labels des axes !!

```
> plot3d([u*cos(v),u*sin(v),cos(2*v)],u=0..1,v=0..2*Pi): # nappe
  paramétrée.
```

On peut utiliser des coordonnées cylindriques sphériques.
Ici on trace rho comme fonction de theta et z.

```
> plot3d(z+cos(t),t=0..2*Pi,z=0.1..0.9,coords=cylindrical):
```

On peut aussi tracer une nappe paramètrée en cylindrique.

```
> r := 2-v*sin(Pi/4+t/2):z := v*cos(t/2):
> plot3d([r,t,z],t=0..2*Pi,v=-1..1,coords=cylindrical):
> plot3d(1,theta=0..2*Pi,phi=-Pi..Pi,coords=spherical):
```

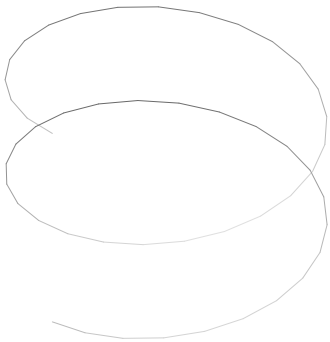
Le premier argument est rho comme fonction de theta et phi. On a aussi la variante paramètrée

```
> plot3d([1,theta,phi],theta=0..2*Pi,phi=-Pi..Pi,coords=spherical)
  :
```

Dans tous ces tracés on dispose des options numpoints , grid et view pour affiner les tracés.

Tracé de courbes

```
> spacecurve([cos(t),sin(t),t,t=-2*Pi..2*Pi]);
```



Comme son nom l'indique, spacecurve trace une courbe paramétrée de l'espace, on peut choisir son ssytème de coordonnées.

```
> spacecurve([1,t,t,t=0..2*Pi],coords=spherical):
```

Exercice : essayer de tracer une géodésique et une loxodromie sur la sphère, après avoir consulté un dictionnaire si besoin est !

```
> implicitplot3d(x^2+y^2+z^2=1,x=-2..2,y=-2..2,z=-2..2);
Error, (in implicitplot3d) need to specify range of three variables
```

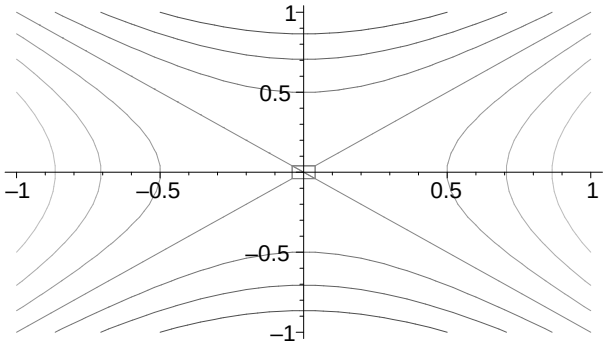
Il faut libérer les variables !

```
> x := 'x' : y := 'y' : z:= 'z' :
  implicitplot3d(x^2+y^2+z^2=1,x=-2..2,y=-2..2,z=-2..2):
```

Comme son nom l'indique, trace une surface définie implicitement i.e. f(x,y,z)=0 ; le résultat est généralement assez laid .

On peut aussi tracer un ensemble de lignes de niveau

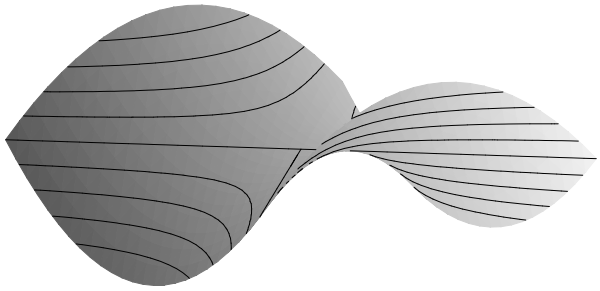
```
> f := (x,y) -> x^2-y^2:
> contourplot(f,-1..1,-1..1);
```



Trace les lignes de niveau de la fonction spécifiée ... la plupart des options de plot sont acceptées.

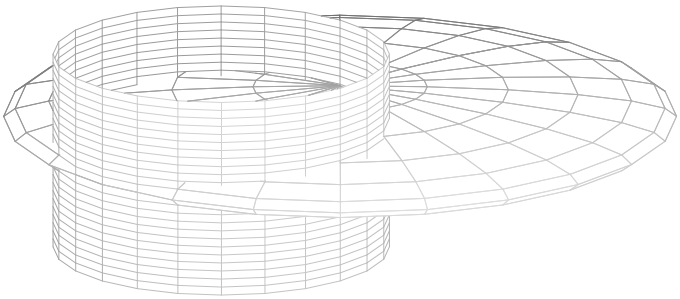
On peut aussi les tracer en perspective 3D, c'est-à-dire placées à leur niveau respecif.

```
> contourplot3d(f,-1..1,-1..1);
```



On peut superposer plusieurs surfaces

```
> s1 := [sin(u)*cos(v),sin(u)*sin(v),cos(u)] : c1 :=
  [(1+cos(u))/2,sin(u)/2,v] :
> plot3d({s1,c1},u=0..2*Pi,v=-Pi..Pi,style=HIDDEN);
```



```
> s2 :=
  plot3d(1,t=0..2*Pi,p=0..Pi,coords=spherical,scaling=constrained)
  :
> c2 := plot3d(c1,u=0..2*Pi,v=-1..1):
> display({s2,c2},style=HIDDEN):
```

Enfin on peut animer des séquences de graphiques (2D ou 3D)

```
> animate3d(cos(t*x)*sin(t*y),x=-Pi..2*Pi,
  y=-Pi..2*Pi,t=1..2,frames=64):
>
```

Il faut cliquer sur le dessin pour avoir les icones qui lancent l'animation...au moins sur ma machine !

En guise de conclusion

A signaler les packages geom3d et geometry qui viennent completer l'ensemble et le package DEtools qui comporte de nombreuses fonctions graphiques.