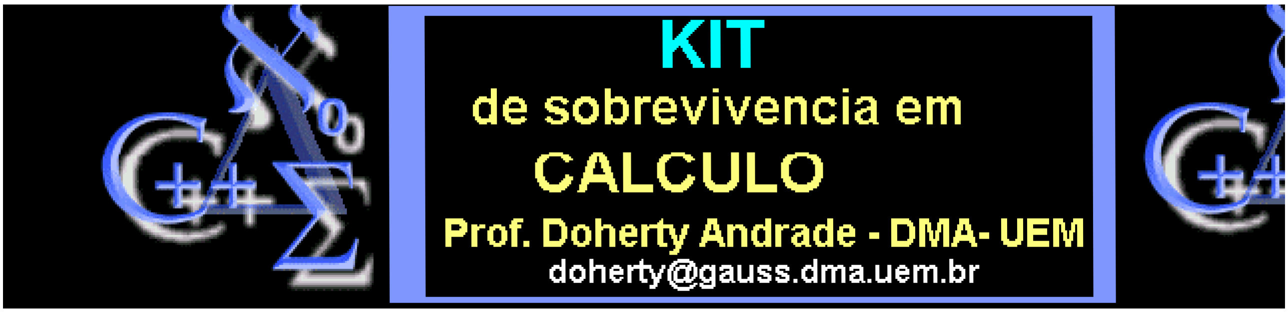


As aulas de 1 a 5 foram elaboradas juntamente com o Prof. Ma To FU (UEM)

Noções Básicas

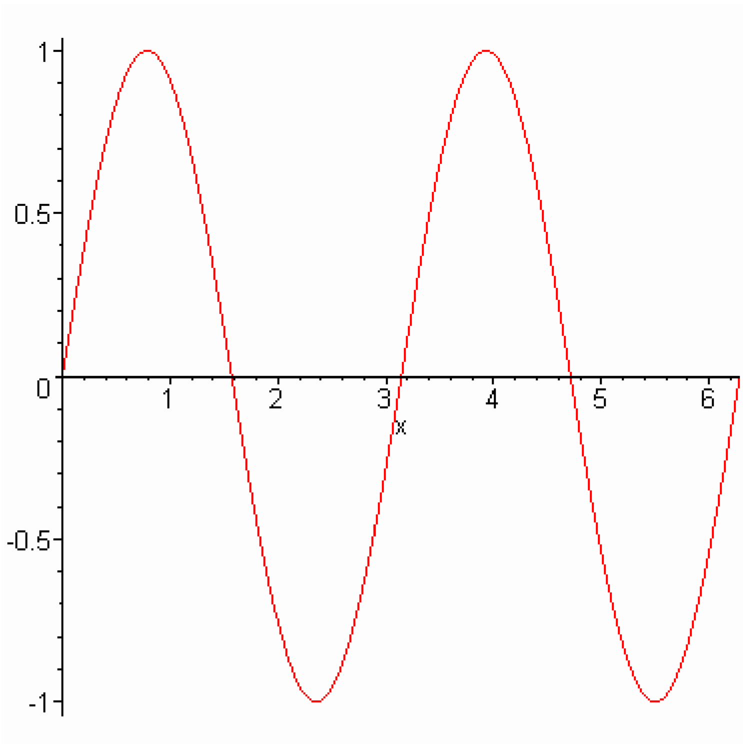
<u>Plotando gráficos com o MapleV</u>

<u>PLOTANDO GRÁFICOS</u>

Vamos iniciar com gráficos simples.

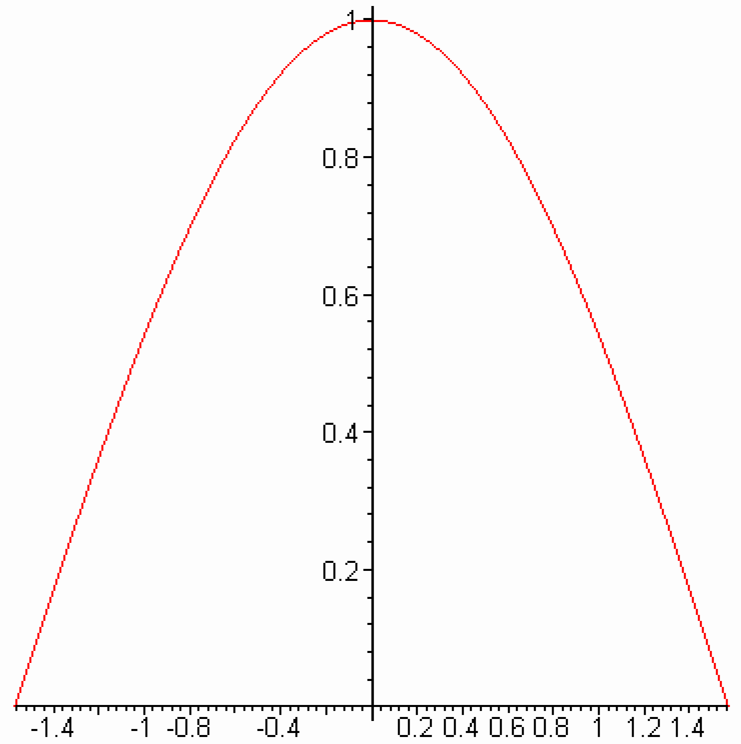
Plotar o gráfico de f(x)=sin(2x) com x em [0,2Pi]

> **plot(sin(2*x), x=0..2*Pi);**



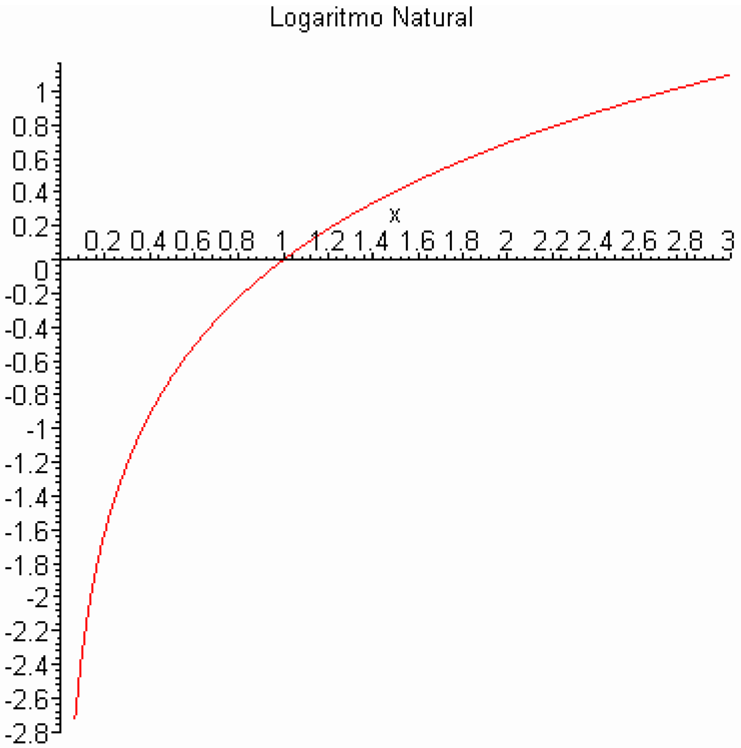
As vezes não é necessário escrever o argumento x

> **plot(cos, -Pi/2 .. Pi/2);**



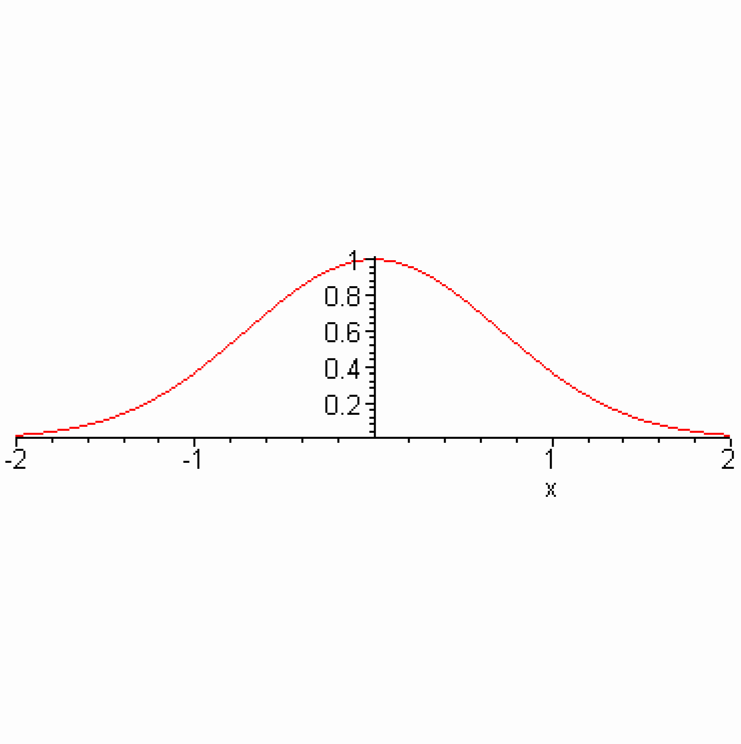
Acrescentando a opção "title"

> **plot(ln(x), x=0..3 , title=`Logaritmo Natural`);**



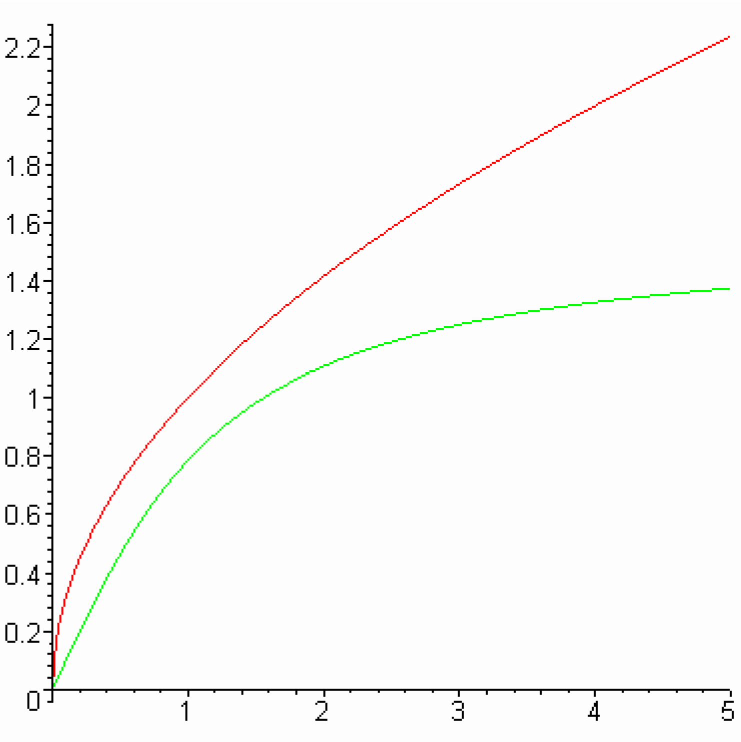
Usando opção "scaling=constrainded": escala 1-1.

> **plot(exp(-x^2), x=-2..2 , scaling=constrained);**



Juntando dois Gráficos. plot({f(x),g(x)}, x=a..b)

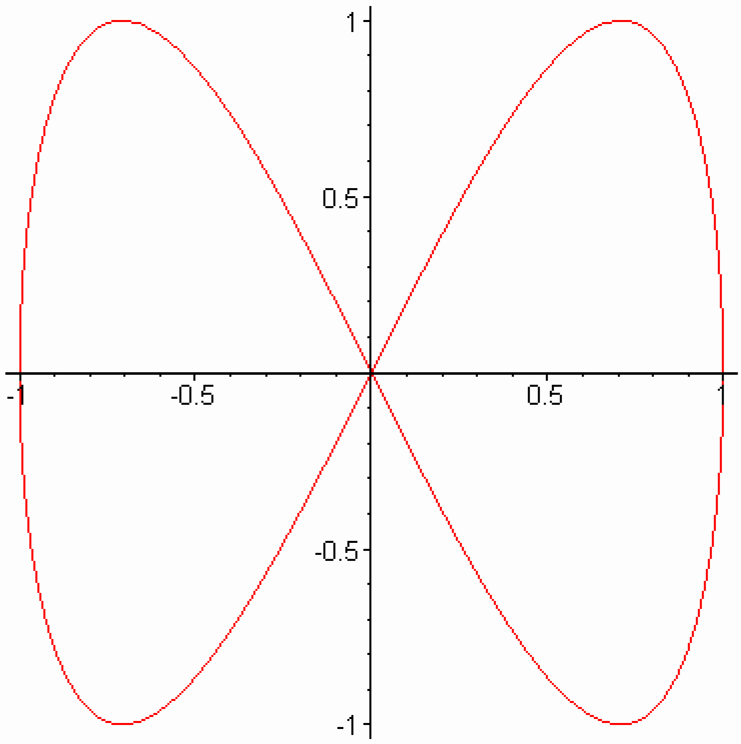
> **plot({arctan, sqrt}, 0..5);**



Plotando a curva parametrizada

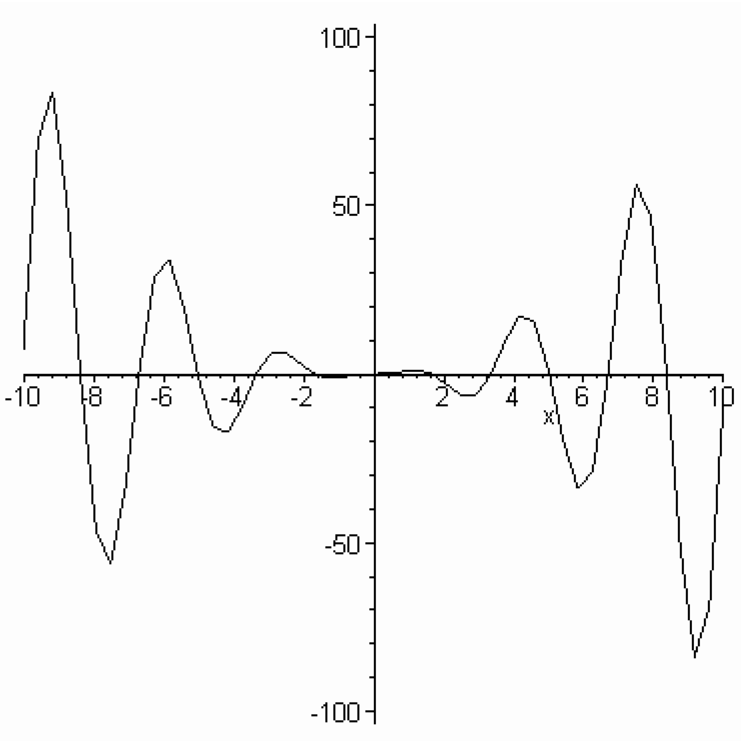
alpha(t)=(cos(t),sin(2t)) com t em [0,2Pi]

> **plot([cos(t), sin(2*t), t=0..2*Pi]);**



Vamos ao cinema ?

```
> with(plots):
animate( x^2*sin(x*t),x=-10..10,t=1..2,frames=50);
```



Os procedimentos em "plot" requerem bastante

recursos de máquina. É costume "zerar a memória"

após 10 plots.

> **restart;**

Dois Problemas interessantes.

PROBLEMA 1

A cardióide é uma curva parametrizada em coordenadas

polares(ρ, θ **). Consulte o help do plot para**

plotar a cardióide $\rho = 5(1 - \cos(\theta))$ **.**

>

PROBLEMA 2

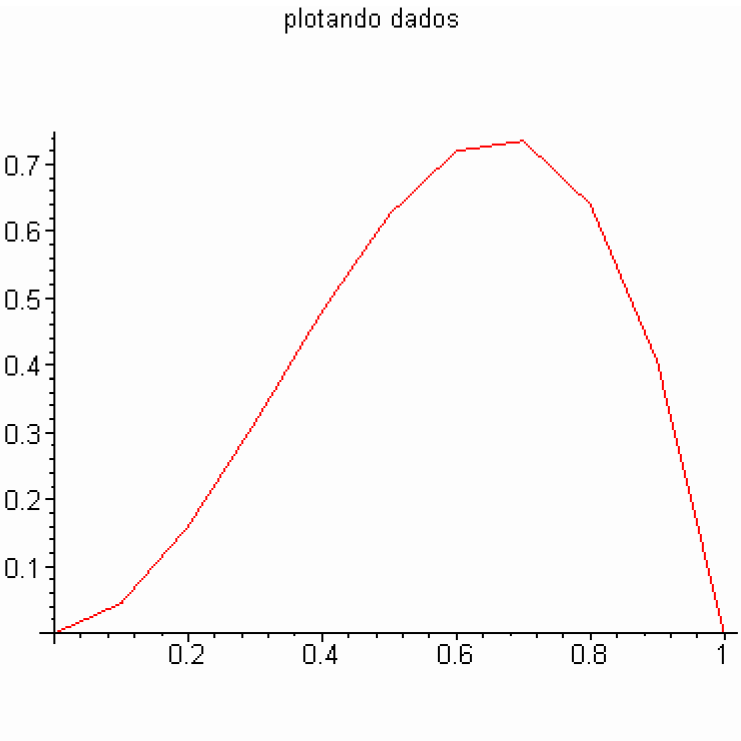
Um certo algorítmo gerou uma lista de pontos x

e uma lista de pontos y abaixo

```
> X:=[seq( .1*k , k=0..10 ) ]:
> Y:=[seq( 5*( (.1*k)^2-(.1*k)^3 ) , k=0..10 ) ]:
> L:=[ seq( [ X[k], Y[k] ], k=1..11) ];
```

$L :=$
[[0, 0], [.1, .045], [.2, .160], [.3, .315], [.4, .480], [.5, .625], [.6, .720], [.7, .735], [.8, .640], [.9,

```
> plot(L, scaling=constrained, title=`plotando dados`);
```



Veremos agora gráficos mais complicados.

Plotando gráficos tridimensionais

> **restart;**

Para plotar gráficos em três dimensões precisamos

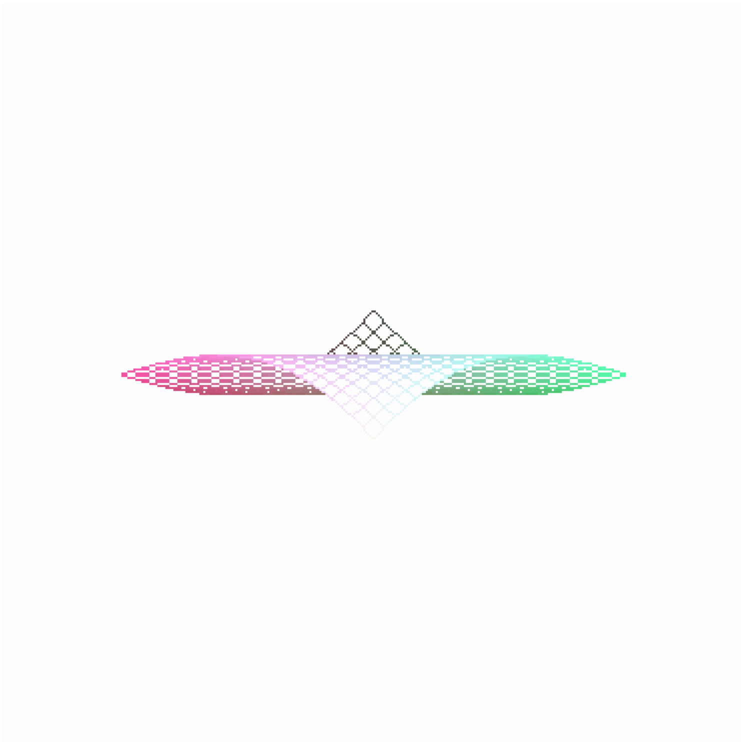
chamar o pacote para gráficos, fazemos isto digitando

"with(plots)".O comando é plot3d

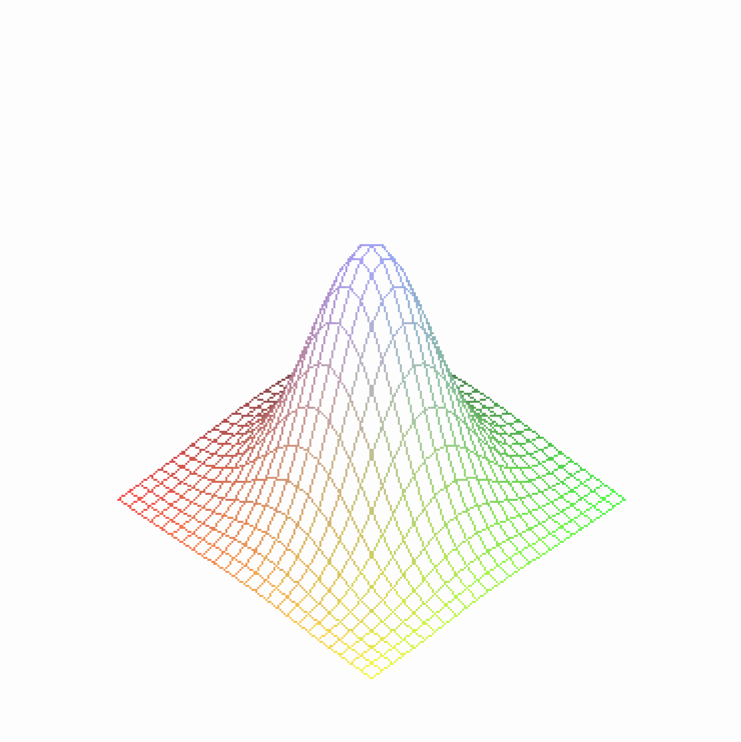
> **with(plots);**

[*animate*, *animate3d*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *contourplot*, *conto*
coordplot, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradpl*
implicitplot, *implicitplot3d*, *inequal*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3*
logplot, *matrixplot*, *odeplot*, *pareto*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3*
replot, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*,
textplot, *textplot3d*, *tubeplot*]

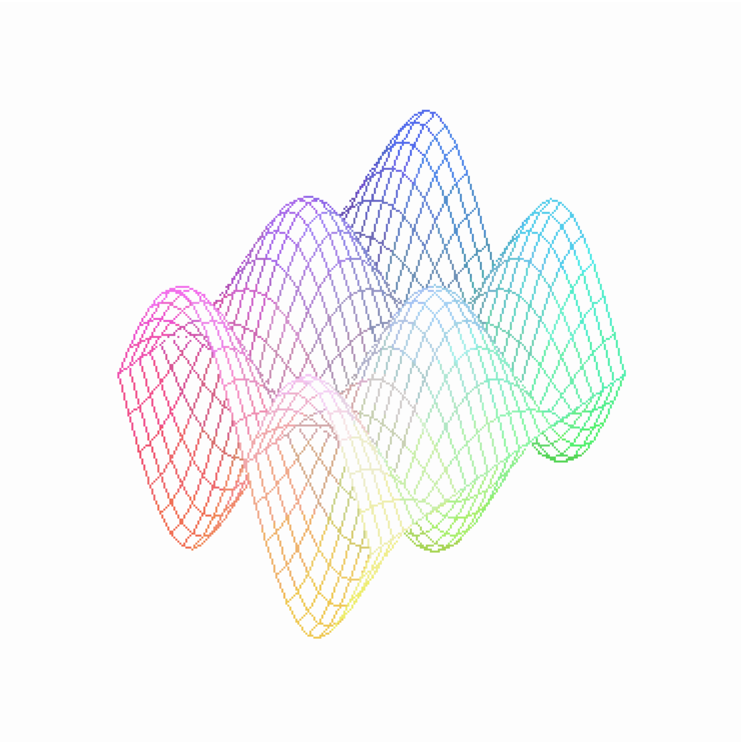
> **plot3d(sin(x+y),x=-1..1,y=-1..1);**



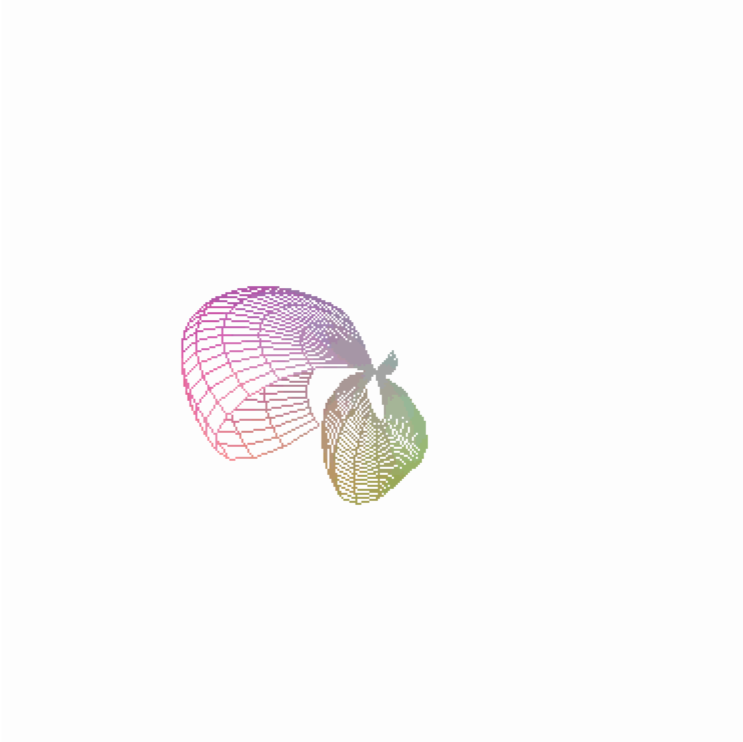
> **plot3d(exp(-x^2-y^2),x=-2..2,y=-2..2);**



> **animate3d({cos(t*x)*sin(t*y),-cos(t*x)*sin(t*y)},x=-Pi..Pi, y=-Pi..Pi,t=1..2);**



> **animate3d(x*cos(t*u),x=1..3,t=1..4,u=2..4,coords=spherical);**



Mais um exemplos

> **restart;**

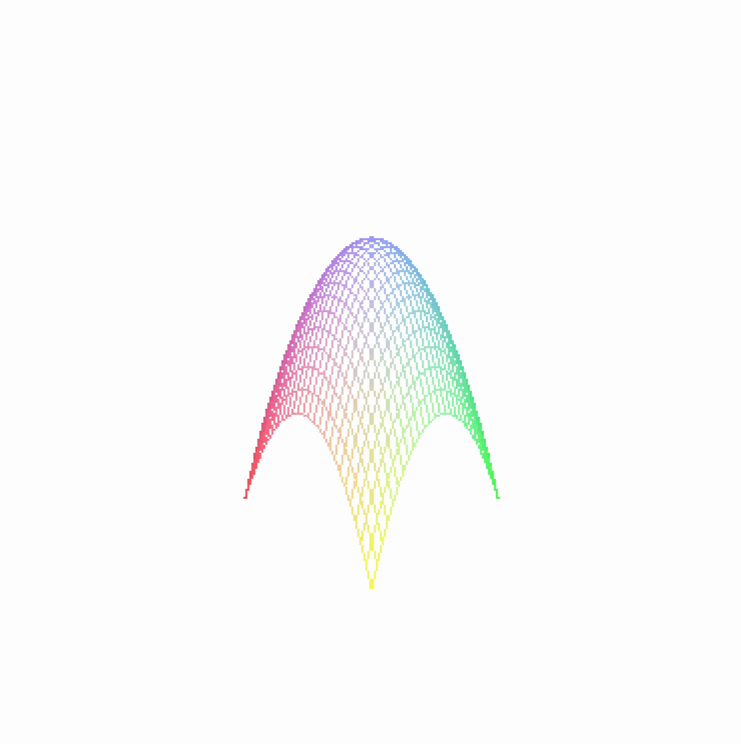
> **with(plots);**

[*animate*, *animate3d*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *contourplot*, *conto*
coordplot, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradpl*
implicitplot, *implicitplot3d*, *inequal*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3*
logplot, *matrixplot*, *odeplot*, *pareto*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3*
replot, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*,
textplot, *textplot3d*, *tubeplot*]

> **f:=(x,y) -> -(x^2+y^2);**

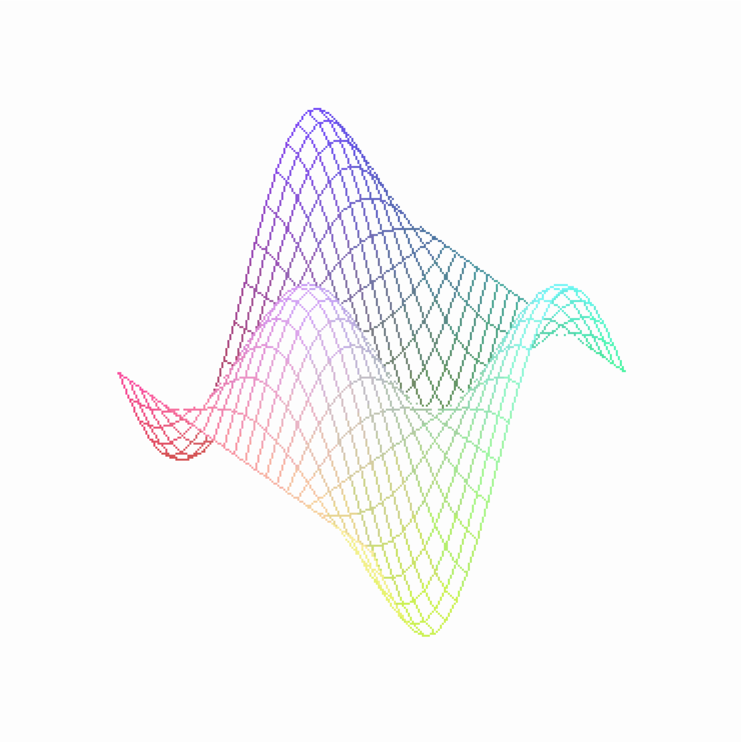
$$f := (x, y) \to -x^2 - y^2$$

> **plot3d(f(x,y), x=-2..2, y=-2..2, scaling=constrained);**



Plotando algo mais bonito

\> **plot3d(sin(x)*cos(y), x=-Pi..Pi, y=-Pi..Pi);**



Observe as funções epeciais que esse pacote tem.

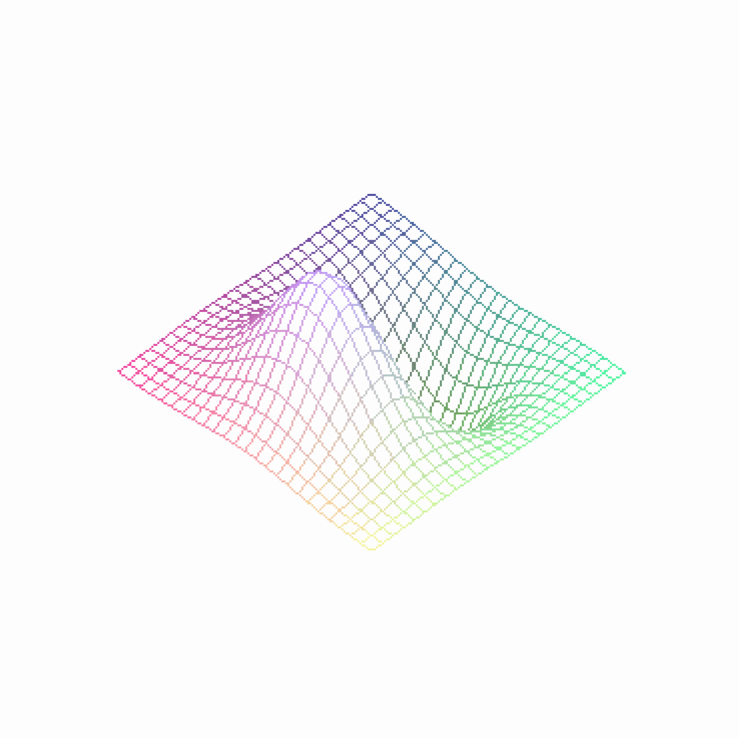
Para se saber sobre alguma delas execute ?nome.

Veremos exemplos de "gradplot",

\> **f :=(x,y) -> x*exp(-x^2-y^2);**

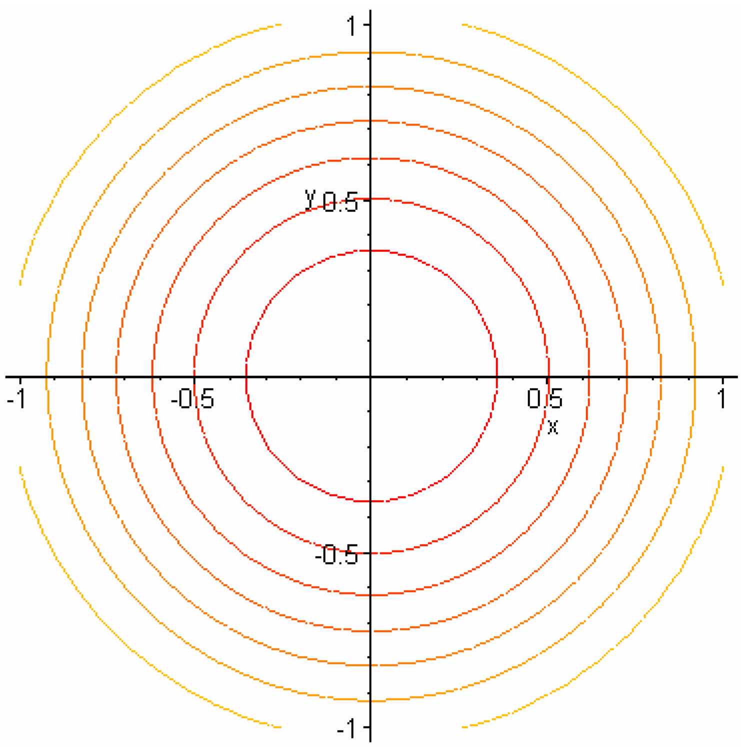
$$f := (x, y) \rightarrow x\, e^{(-x^2 - y^2)}$$

> plot3d(f(x,y), x=-2..2, y=-2..2);



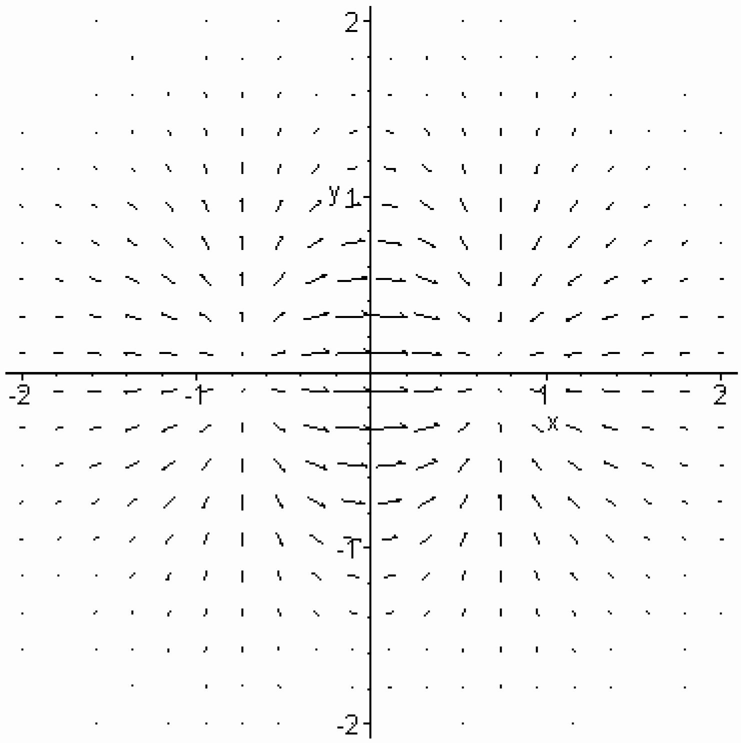
Vamos ver o CAMPO GRADIENTE da f com "gradplot" e "countourplot"

> contourplot(sin(x^2+y^2),x=-1..1,y=-1..1);

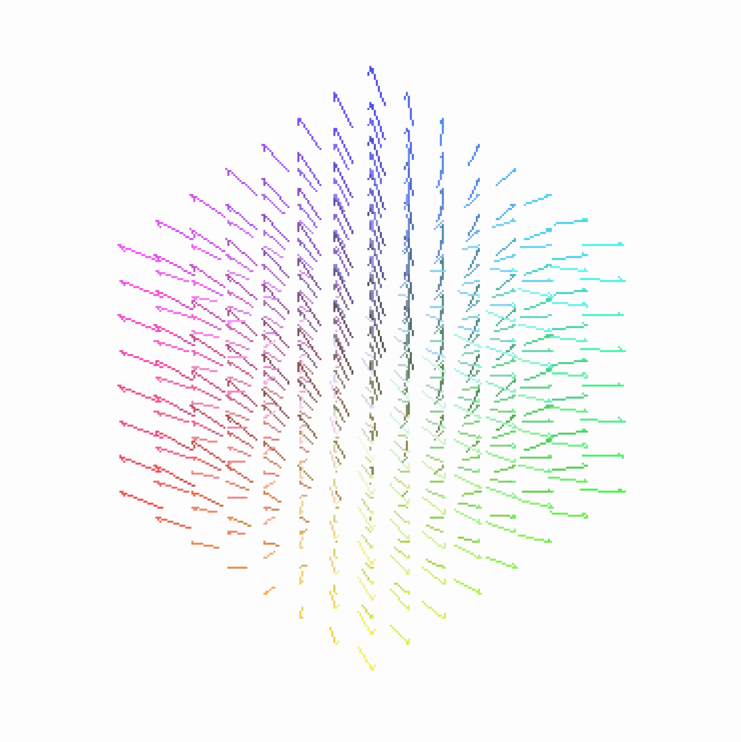


>

> **gradplot(f(x,y), x=-2..2, y=-2..2);**

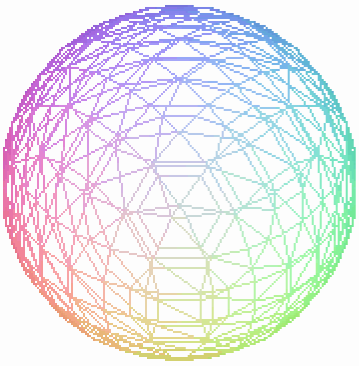


> **gradplot3d(x^2+2*y^2+z+1,x=-1..1,y=-1..1,z=-1..1);**



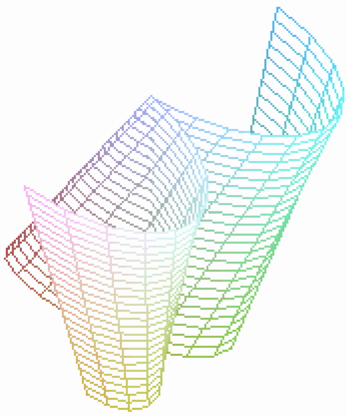
Vamos desenhar a esfera, dada implicitamente.

> **implicitplot3d(x^2+y^2+z^2=4, x=-2..2, y=-2..2, z=-2..2 , scaling=constrained);**



Coordenadas cilindricas

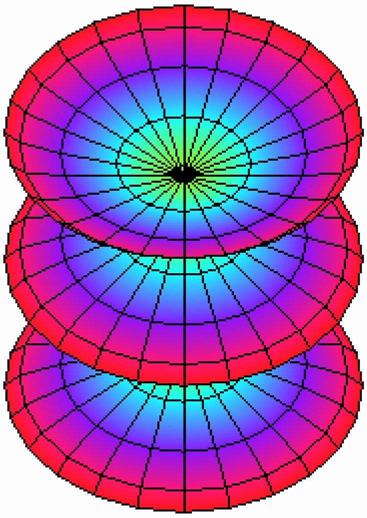
> **cylinderplot(z+ 3*cos(2*theta),theta=0..Pi,z=0..3);**



> **f := (5*cos(y)^2 -1)/3;**

$$f := \frac{5}{3}\cos(y)^2 - \frac{1}{3}$$

> **cylinderplot(f, x=0..2*Pi,y=-Pi..Pi,style=PATCH, color = f);**

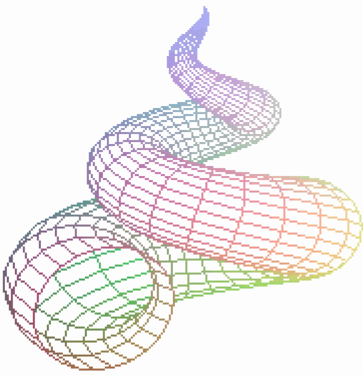


>

Um exemplo tubular com "tubeplot"

> **c:=[(t- 5*Pi)*sin(t)/3,(t-5*Pi)*cos(t)/3,(t-5*Pi)*.9, t=0..5*Pi]:**

> **tubeplot(c, radius=(t-5*Pi)*.2, orientation=[-37,81],tubepoints=25, style=hidden);**



>

> **# UM OUTRO PASSATEMPO PARA ESPANTAR O TÉDIO**

```
> # Leia "?plots[spacecurve]" e desenhe uma espiral.

> # DICA: use a opção "color=black"

> # plot([5*(1-cos(t)) ,t , t=0..2*Pi], coords=polar);

> # X:=[seq( .1*k , k=0..10 ) ]:

> # Y:=[seq( 5*( (.1*k)^2-(.1*k)^3 ) , k=0..10 ) ]:

> # L:=[ seq( [ X[k], Y[k] ], k=1..11) ];

> # plot(L, scaling=constrained,title=`plotando dados`);

>
```