Procedimentos Graficos em Calculo Integral

Universidade Estadual de Maringá
Departamento de Matemática
Prof. Doherty Andrade (DMA- UEM)
Prof. Timothy M. (WLU-USA)

Maple

Este procedimento plota superficies de revolução. A sintaxe é

rotxplot(f, x:a..b, y= c eixo) ou rotyplot(f, x=a..b, x=c eixo).

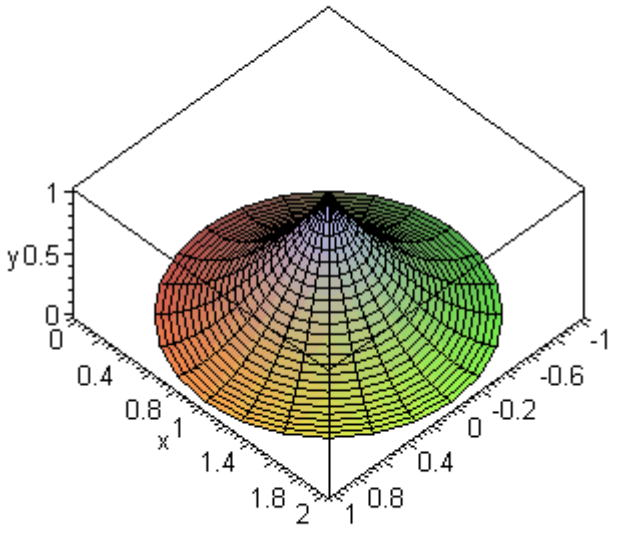
Execute o procedimento e faça os exemplos.

O Procedimento (execute-o)

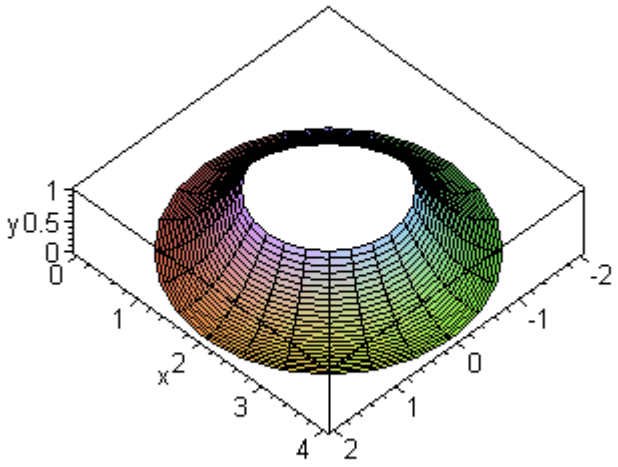
Exemplos

1. rotyplot

```
> rotyplot(x^2,x=0..1,x=1);
```

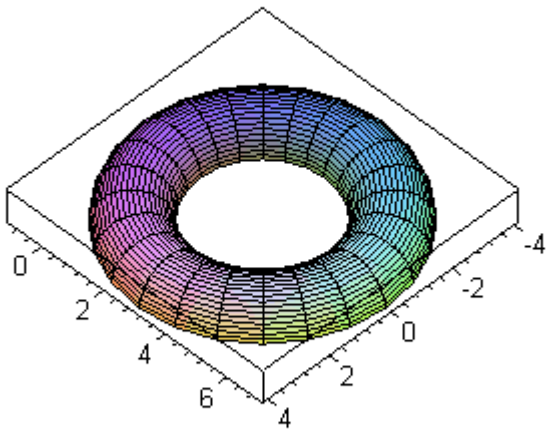



```
> rotyplot(t->t^2,x=0..1,x=2);
```



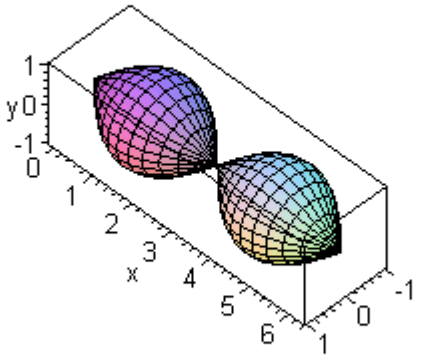
> **rotyplot([cos(x),sin(x)],x=0..Pi,x=3,title=`Toro`,scaling=constrained,style=patch,axes=boxed);**

Toro

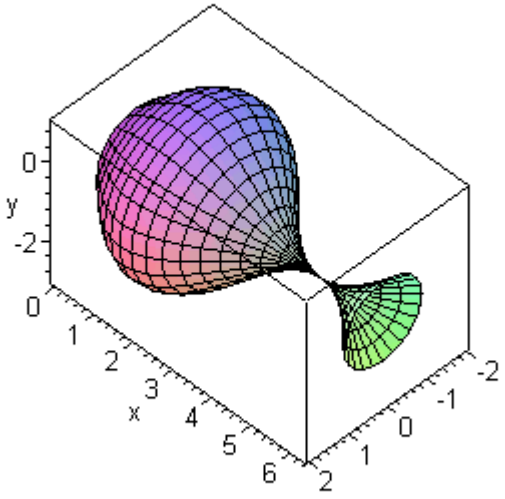


2. rotxplot

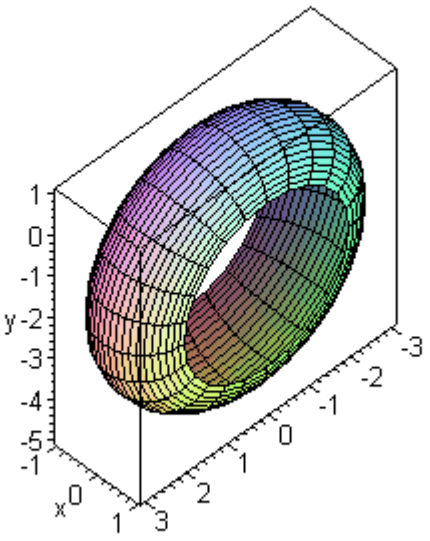
> **rotxplot(sin(x),x=0..2*Pi,y=0);**



> **rotxplot(x->sin(x),x=0..2*Pi,y=-1);**

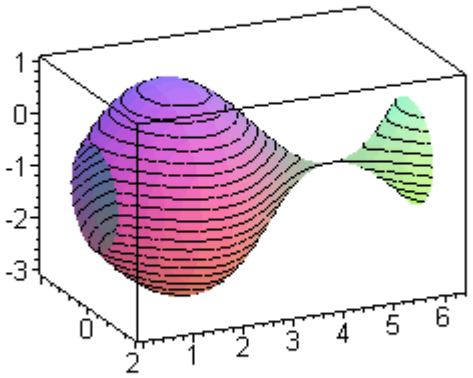


> **rotxplot([cos(x),sin(x)],x=0..Pi,y=-2);**

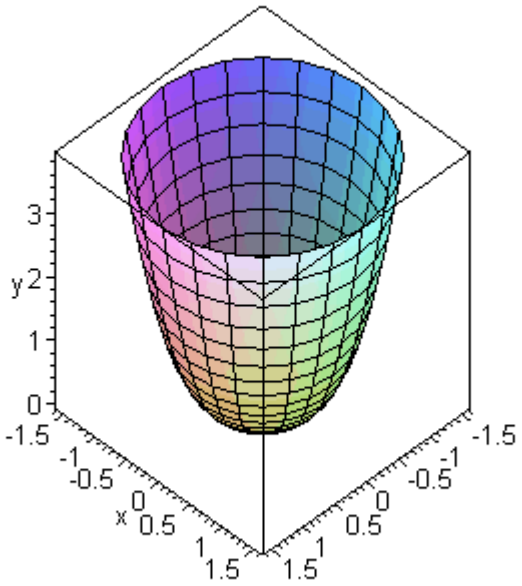


> **rotxplot(x->sin(x),x=0..2*Pi,y=-1,style=patchcontour, title=`Uma superf. conhecida `,scaling=constrained, axes=boxed,orientation=[-25,71]);**

Uma superf. conhecida



> **rotyplot(x^3*sin(x),x=0..Pi/2,x=0);**


3
y 2
1
0
-1.5
-1
-0.5
0
x 0.5
1
1.5
-1.5
-1
-0.5
0
0.5
1
1.5

>

O Procedimento (execute-o)

```
> rotyplot:=proc()
> local a, b, c, d, f, i, xbound, rotaxis, f1, opt_seq, p1, profile, tloc;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> f:=args[1];
> xbound:=args[2];
> rotaxis:=args[3];
> a:=op(1, rhs(xbound));
> b:=op(2, rhs(xbound));
> if not type (xbound, string = range) then
> ERROR(`range expression for x is incorrect`) fi;
> if not type (rotaxis, string = numeric) then
> ERROR(`axis expression is not correct`) fi;
> if not (lhs(xbound) = 'x') or not (lhs(rotaxis) = 'x') then
> ERROR(`the variable name must be x`) fi;
> if not type (evalf(a), numeric) or not type(evalf(b), numeric) then
> ERROR(`range limits are not real numbers` ) fi;
> c:=op(1, rhs(rotaxis));
> d:=lhs(xbound);
> if not type (f, procedure) or not type(f, list) then
> f1:=unapply(f,d); fi;
> if type(f,procedure) then
> f1:=f; fi;
> if not type(f, list) then
> profile:=[(d-c)*sin(tloc),(d-c)*cos(tloc)+c,f1(d)]; fi;
```

```
> if type(f,list) then f1:=unapply(f,d);
> profile:=[(f1(d)[1]-c)*sin(tloc),(f1(d)[1]-c)*cos(tloc)+c,f1(d)[2]]; fi;
> if nargs = 3 then
> p1:=plot3d(profile,d=a..b,tloc=0..2*Pi,style=PATCH, scaling=CONSTRAINED,
> axes=BOXED, grid=[25,25],labels=[` `,x,y]); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> p1:=plot3d(profile,d=a..b,tloc=0..2*Pi),opt_seq; fi;
> plots[display3d](p1);
> end:
> #A y-axis rotation plotter. Same input as xrotplot, except that the axis of
> #rotation is given in the form y=c.
> rotxplot:=proc()
> local a, b, c, d, f1, f, i, xbound, rotaxis, opt_seq, p1, profile, tloc;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> f:=args[1];
> xbound:=args[2];
> rotaxis:=args[3];
> a:=op(1, rhs(xbound));
> b:=op(2, rhs(xbound));
> if not type (xbound, string = range) then
> ERROR(`range expression for x is incorrect`) fi;
> if not type (rotaxis, string = numeric) then
> ERROR(`axis expression is not correct`) fi;
> if not (lhs(xbound) = 'x') or not (lhs(rotaxis) = 'y') then
```

```
> ERROR(`the variable name must be x`) fi;
> if not type (evalf(a), numeric) or not type(evalf(b), numeric) then
> ERROR(`range limits are not real numbers` ) fi;
> c:=op(1, rhs(rotaxis));
> d:=lhs(xbound);
> if not type (f, procedure) or not type (f, list) then
> f1:=unapply(f,d); fi;
> if type (f, procedure) then
> f1:=f; fi;
> if not type(f,list) then
> profile:=[(f1(d)-c)*sin(tloc),d,(f1(d)-c)*cos(tloc)+c]; fi;
> if type(f,list) then f1:=unapply(f,d);
> profile:=[(f1(d)[2]-c)*sin(tloc),f1(d)[1],(f1(d)[2]-c)*cos(tloc)+c]; fi;
> if nargs = 3 then
> p1:=plot3d(profile,d=a..b,tloc=0..2*Pi,style=PATCH, scaling=CONSTRAINED,
> axes=BOXED, grid=[25,25], labels=[` `,x,y]); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> p1:=plot3d(profile,d=a..b,tloc=0..2*Pi),opt_seq; fi;
> plots[display3d](p1);
> end:
```