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

Traça Gráficos sobre regiões polares. O primeiro argumento (uma fun **ção**) deverá ser expressa em

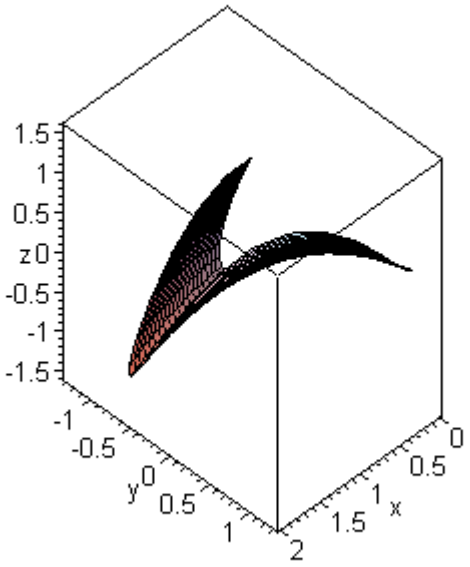
coordendas polares. *r* como função de θ ou θ como função de *r* .

Execute esta worksheet e faça os exemplos.

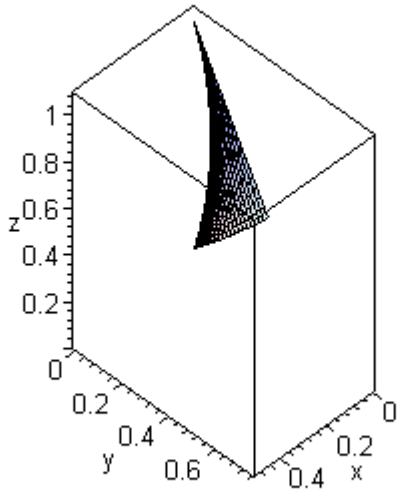
O Procedimento (execute-0)

Exemplos

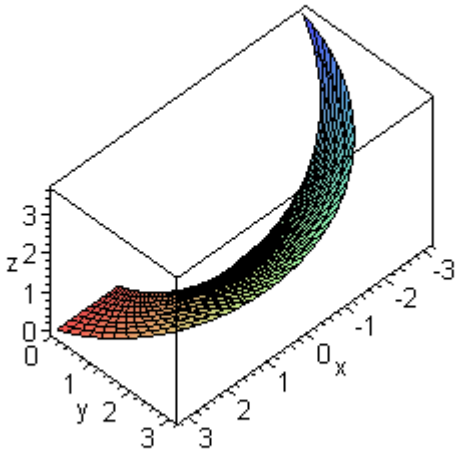
> **rtgraphplot(r^2*sin(theta)*cos(theta),r=1..1+cos(theta),theta=-Pi/2..Pi/2);**



> **plots[display](rtgraphplot((p,q)->q,r=sin(3*theta)..sin(theta),theta=Pi/4..Pi/3),view=0..1.1);**



> **trgraphplot(theta*log(r),theta=0..r,r=Pi/2..Pi);**

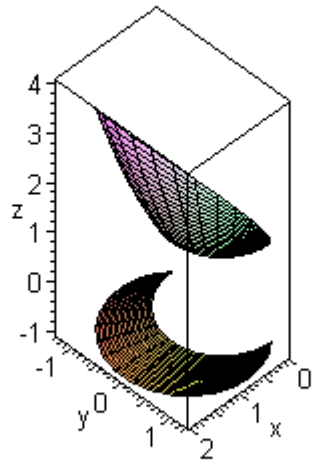


> **p1:=rtgraphplot(r^2,r=1..1+cos(theta), theta=-Pi/2..Pi/2):**

> **p2:=rtgraphplot(-1,r=1..1+cos(theta), theta=-Pi/2..Pi/2):**

> **plots[display3d]({p1,p2},title=`O grafico de r^2 sobre seu dominio`);**

O grafico de r^2 sobre seu dominio

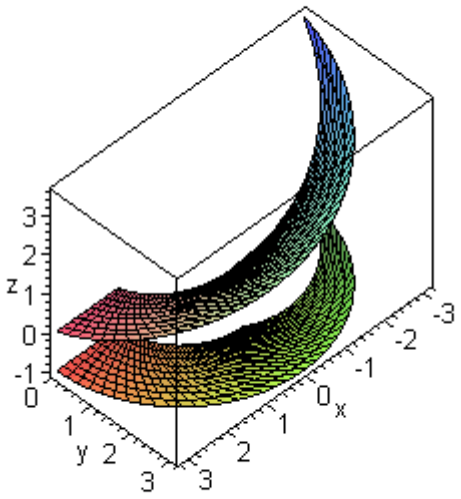


> **q1:= trgraphplot(theta*log(r),theta=0..r,r=Pi/2..Pi):**

> **q2:= trgraphplot(-1,theta=0..r,r=Pi/2..Pi):**

> **plots[display3d]({q1,q2},title=`O grafico de theta*log(r) sobre seu dominio`);**

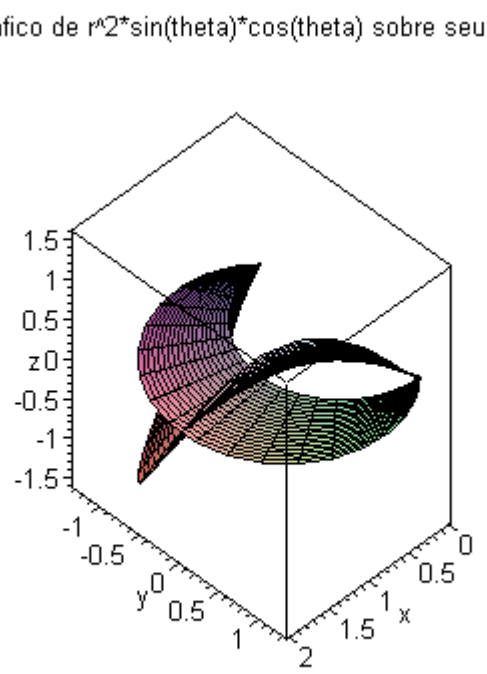
O grafico de theta*log(r) sobre seu dominio



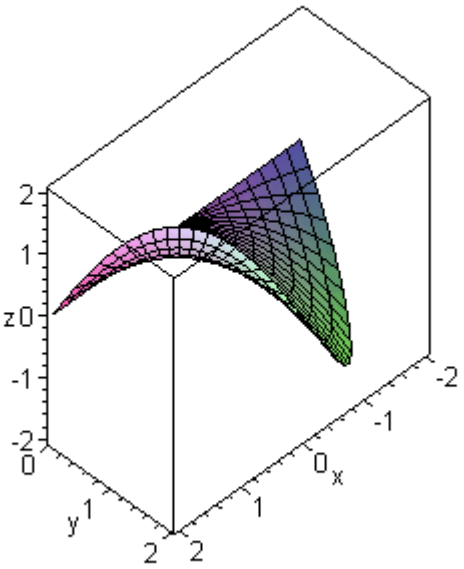
> **L1:= rtgraphplot(r^2*sin(theta)*cos(theta),r=1..1+cos(theta),theta=-Pi/2..Pi/2):**

> **L2:=rtgraphplot(0,r=1..1+cos(theta),theta=-Pi/2..Pi/2):**

> **plots[display3d]({L1,L2},title=`O grafico de r^2*sin(theta)*cos(theta) sobre seu dominio`);**



> **trgraphplot(r^2*sin(theta)*cos(theta),theta=0..Pi,r=0..2);**



>

>

O Procedimento (execute-0)

```
> rtgraphplot:=proc()

> local a, b, c, d, dr, dt, g, h, i, k, f1, g1, h1, opt_seq, p1,rbound,

> tbound, tloc, sloc, surface;

> if nargs < 3 then

> ERROR(`there must be at least three arguments`) fi;

> h:=args[1];

> rbound:=args[2];

> tbound:=args[3];

> a:=op(1,rhs(tbound));

> b:=op(2,rhs(tbound));

> ### WARNING: semantics of type `string` have changed
if not type (tbound, string = range) then

> ERROR(`range expression for theta is incorrect`) fi;

> ### WARNING: semantics of type `string` have changed
if not type (rbound, string = range) then

> ERROR(`range expression for r is not correct`) fi;

> if not (lhs(tbound) = 'theta') then

> ERROR(`the second variable name must be theta`) fi;

> if not (lhs(rbound) = 'r') then

> ERROR(`the first variable name must be r`) fi;

> if not type (evalf(a), numeric) or not type(evalf(b), numeric) then

> ERROR(`range limits for theta are not real numbers` ) fi;

> c:=op(1,rhs(rbound));

> d:=op(2,rhs(rbound));

> g:=lhs(rbound);

> k:=lhs(tbound);
```

```
> tloc:=(1-u)*a+u*b;
> f1:=unapply(c,k);
> g1:=unapply(d,k);
> sloc:=(1-v)*f1(tloc)+v*g1(tloc);
> if not type (h, procedure) then
> h1:=unapply(h,(g,k)); fi;
> if type(h,procedure) then h1:=h fi;
> surface:=[sloc,tloc,h1(sloc,tloc)];
> for i from 0 to 10 do
> dt:= (10-i)/10*a + i/10*b ;
> dr:=evalf(g1(dt) - f1(dt));
> if (dr < - 0.000001) then
> ERROR(`the radius must always be positive`) fi; od;
> if nargs = 3 then
> p1:=plots[cylinderplot](surface,u=0..1,v=0..1,style=PATCH,axes=BOXED,
> scaling=CONSTRAINED, grid=[20,20], labels=[x,y,z]); fi;
> if nargs >3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> p1:=plots[cylinderplot](surface, u=0..1,v=0..1),opt_seq; fi;
> plots[display3d](p1);
> end:
> #Reverse the order: angle as a function of radius.
> trgraphplot:=proc()
> local a, b, c, d, g, k, f1, g1, h1, i, h, opt_seq, p1, rbound,tbound,
> tloc, sloc, surface;
> if nargs < 3 then
```

```
> ERROR(`there must be at least three arguments`) fi;

> h:=args[1];

> tbound:=args[2];

> rbound:=args[3];

> a:=op(1,rhs(rbound));

> b:=op(2,rhs(rbound));

> ### WARNING: semantics of type `string` have changed
if not type (tbound, string = range) then

> ERROR(`range expression for theta is incorrect`) fi;

> ### WARNING: semantics of type `string` have changed
if not type (rbound, string = range) then

> ERROR(`range expression for r is not correct`) fi;

> if not (lhs(tbound) = 'theta') then

> ERROR(`the second variable name must be theta`) fi;

> if not (lhs(rbound) = 'r') then

> ERROR(`the first variable name must be r`) fi;

> if not type (evalf(a), numeric) or not type(evalf(b), numeric) then

> ERROR(`range limits for theta are not real numbers` ) fi;

> c:=op(1,rhs(tbound));

> d:=op(2,rhs(tbound));

> g:=lhs(rbound);

> k:=lhs(tbound);

> tloc:=(1-u)*a+u*b;

> f1:=unapply(c,g);

> g1:=unapply(d,g);

> sloc:=(1-v)*f1(tloc)+v*g1(tloc);

> if not type (h, procedure) then
```

```
> h1:=unapply(h,(g,k)); fi;
> if type(h,procedure) then h1:=h fi;
> surface:=[tloc,sloc,h1(tloc,sloc)];
> if nargs = 3 then
> p1:=plots[cylinderplot](surface,u=0..1,v=0..1,style=PATCH,axes=BOXED,
> scaling=CONSTRAINED, grid=[10,40], labels=[x,y,z]); fi;
> if nargs >3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> p1:=plots[cylinderplot](surface, u=0..1,v=0..1),opt_seq; fi;
> plots[display3d](p1);
> end:
```