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 regiões em coordenadas polares. Veja os exemplos

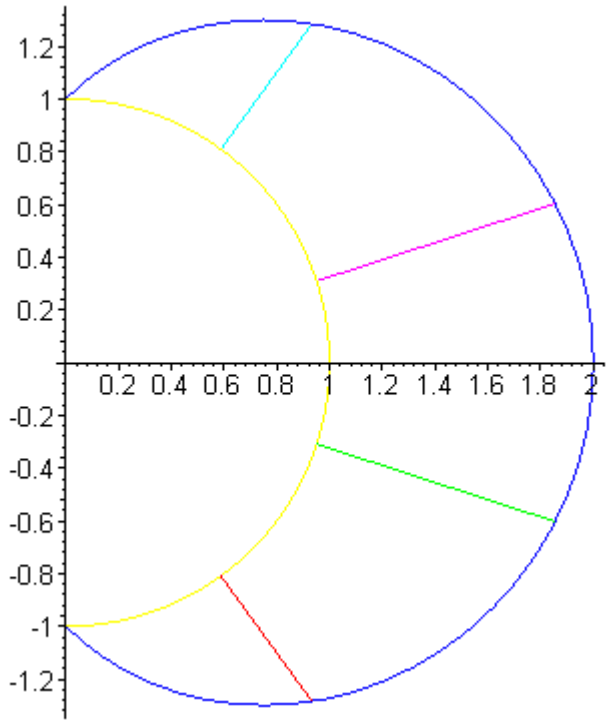
Execute o procedimento e faça os exemplos.

O Procedimento (execute-o)

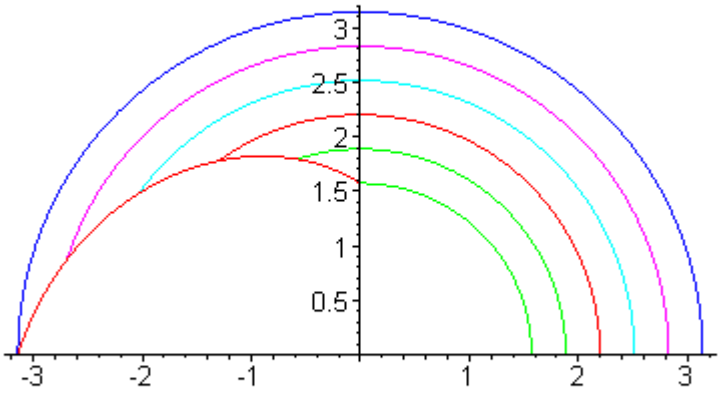
Exemplos

Exemplos de drdtplot e de dtdrplot

```
> drdtplot(r=1..1+cos(theta),theta=-Pi/2..Pi/2);
```




```
> dtdrplot(theta=0..r,r=Pi/2..Pi);
```



> **#plots[display]({drdtplot(r=sin(3*theta)..sin(theta),theta=Pi/4..Pi/3), #drdtplot(r=0..sin (theta),theta=Pi/3..Pi/2)}, scaling=constrained, title=` #region entre r=sin(3*theta) e r=sin(theta) `);**

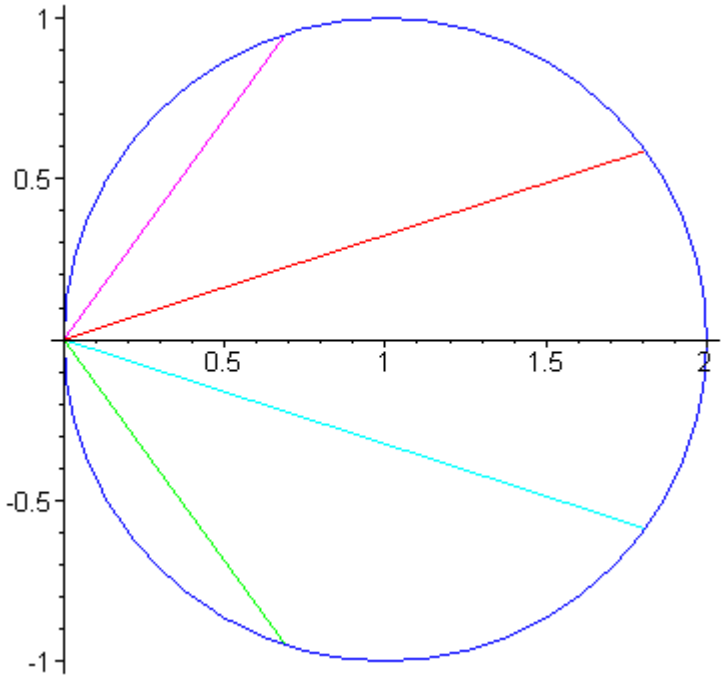
> **#p1:=drdtplot(r=1..3+(sin(theta))^2,theta=0..Pi/2):**

> **#p2:=drdtplot(r=1..3,theta=0..Pi/2):**

> **#plots[display]({p1,p2},title=` as regoes acima`);**

> **#plots[display]({drdtplot(r=1..3+(sin(theta))^2,theta=0..Pi/2), #drdtplot(r=1..3,theta=0..Pi/2)}, scaling=constrained, title=` region #entre as regioes`);**

> **drdtplot(r=0..2*cos(theta), theta = -Pi/2 .. Pi/2);**


1
0.5
-0.5
-1
0.5
1
1.5
2

>

O Procedimento (execute-o)

```
> drdtplot:=proc()

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

> tbound, tloc;

> if nargs < 2 then

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

> rbound:=args[1];

> tbound:=args[2];

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

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

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

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

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

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

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

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

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

> ERROR(`the first variable name must be theta`) 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:=lhs(tbound);

> d:=lhs(rbound);

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

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

> g1:=unapply(g,c);
```

```
> h1:=unapply(h,c);
> pset:={[g1(tloc),tloc,tloc=a..b],[h1(tloc),tloc,tloc=a..b]};
> for i from 0 to 5 do
> dt:= (5-i)/5*a + i/5*b ;
> dr:=evalf(h1(dt) - g1(dt));
> if (dr < - 0.000001) then
> ERROR(`the positions of the angle functions are not correct`) fi;
> if (dr > 10^(-3) ) then
> pset:=pset union
> {[tloc*h1(dt)+(1-tloc)*g1(dt),dt, tloc=0..1]} fi;
> od;
> if nargs =2 then
> p1:=plot(pset,coords=polar, axes=NORMAL, scaling=CONSTRAINED); fi;
> if nargs > 2 then
> opt_seq:=seq(args[i], `i`=3..nargs);
> p1:=plot(pset, coords=polar), opt_seq; fi;
> plots[display](p1);
> end:
> #Angle as a function of radius.
> dtdrplot:=proc()
> local a, b, c, d, dr, dt, g, g1, h, h1, i, opt_seq, p1, pset, rbound,
> tbound, tloc;
> if nargs < 2 then
> ERROR(`there must be at least two arguments`) fi;
> tbound:=args[1];
> rbound:=args[2];
```

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

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

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

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

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

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

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

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

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

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

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

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

> c:=lhs(rbound);

> d:=lhs(tbound);

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

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

> g1:=unapply(g,c);

> h1:=unapply(h,c);

> pset:={[tloc,g1(tloc),tloc=a..b],[tloc,h1(tloc),tloc=a..b]};

> for i from 0 to 5 do

> dr:= (5-i)/5*a + i/5*b ;

> dt:=evalf(h1(dr) - g1(dr));

> if (evalf(dr) < - 0.000001) then

> ERROR(`the radius must always be positive`) fi;

> if (evalf(dr) > 10^(-3) ) then
```

```
> pset:=pset union
> {[dr,tloc*h1(dr)+(1-tloc)*g1(dr), tloc=0..1]} fi;
> od;
> if nargs = 2 then
> p1:=plot(pset,coords=polar, axes=NORMAL, scaling=CONSTRAINED); fi;
> if nargs > 2 then
> opt_seq:=seq(args[i],`i`=3..nargs);
> plot(pset,coords=polar),opt_seq; fi;
> plots[display](p1);
> end:
```