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 é capaz de plotar as seis diferentes regiões de integação tripla expressas em coordendas cilíndricas (r, θ, z).

Sintaxe: dzdrdtplot(z=f(r,theta)..g(r,theta), r=h(theta)..k(theta), theta=a..b, opts).

Este procedimento plota a fronteira da região no espaço, quando descrita em coordenadas cilíndricas:

$z = \mathrm{f}(r, \theta) \,..\, \mathrm{g}(r, \theta)$, $r = \mathrm{h}(\theta) \,..\, \mathrm{k}(\theta)$, $\theta = a \,..\, b$

Região entre os gráficos de $z = \mathrm{f}(r, \theta)$ **e** $z = \mathrm{g}(r, \theta)$ **e sobre a região descrita quando** r **varia entre** $\mathrm{h}(\theta)$ **e** $\mathrm{k}(\theta)$ **e** θ **varia entre** a **e** b **.**

Os cinco procedimentos análogos são:dzdtdrplot,

drdzdtplot,

drdzdtplot,

dtdrdzplot,

dtdzdrplot,

todos com sintaxes análogas.

Execute o procedimento e faça os exemplos.

O Procedimento (execute-o)

#1. drd(theta)dz

```
> drdtdzplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, r1, t1, z1, u,v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq, rbound,
> tbound, zbound;
```

```
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> rbound:=args[1];
> tbound:=args[2];
> zbound:=args[3];
> a:=op(1, rhs(zbound));
> b:=op(2, rhs(zbound));
> if not type (zbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of drdtdz type` ) fi;
> f:=op(1, rhs(rbound));
> g:=op(2, rhs(rbound));
> h:=op(1, rhs(tbound));
> k:=op(2, rhs(tbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(t1,z1));
> g1:=unapply(g,(t1,z1));
> h1:=unapply(h,z1);
> k1:=unapply(k,z1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
```

```
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[f1(vloc,uloc),vloc,uloc],[g1(vloc,uloc),vloc,uloc],
> [wloc,h1(uloc),uloc], [xloc,k1(uloc),uloc],[yloc,uloc1,a],[zloc,uloc2,b]};
> if nargs = 3 then
> p1:=plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); 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 :
#2. dzd(theta)tdr
> dzdtdrplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, r1, t1, z1, u,v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,rbound,
> tbound, zbound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> zbound:=args[1];
> tbound:=args[2];
> rbound:=args[3];
> a:=op(1, rhs(rbound));
> b:=op(2, rhs(rbound));
```

```
> if not type (rbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
> or not type (evalf(a), numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dzdtdr type` ) fi;
> f:=op(1, rhs(zbound));
> g:=op(2, rhs(zbound));
> h:=op(1, rhs(tbound));
> k:=op(2, rhs(tbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(t1,r1));
> g1:=unapply(g,(t1,r1));
> h1:=unapply(h,r1);
> k1:=unapply(k,r1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[uloc,vloc,f1(vloc,uloc)],[uloc,vloc,g1(vloc,uloc)],
> [uloc,h1(uloc),wloc], [uloc,k1(uloc),xloc],[a,uloc1,yloc],[b,uloc2,zloc]};
> if nargs = 3 then
```

```
> p1:=plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); 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 :
#3. drdzd(theta)
> drdzdtplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, r1, t1, z1, u,v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq, rbound,
> tbound, zbound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> rbound:=args[1];
> zbound:=args[2];
> tbound:=args[3];
> a:=op(1, rhs(tbound));
> b:=op(2, rhs(tbound));
> if not type (rbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
> or not type (evalf(a), numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of drdzdt type` ) fi;
> f:=op(1, rhs(rbound));
> g:=op(2, rhs(rbound));
> h:=op(1, rhs(zbound));
```

```
> k:=op(2, rhs(zbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(z1,t1));
> g1:=unapply(g,(z1,t1));
> h1:=unapply(h,t1);
> k1:=unapply(k,t1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[f1(vloc,uloc),uloc,vloc],[g1(vloc,uloc),uloc,vloc],
> [wloc,uloc,h1(uloc)], [xloc,uloc,k1(uloc)],[yloc,a,uloc1],[zloc,b,uloc2]};
> if nargs = 3 then
> p1:=plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); 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 :
```

#4. dzdrd(theta)

```
> dzdrdtplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, r1, t1, z1, u,v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq, rbound,
> tbound, zbound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> zbound:=args[1];
> rbound:=args[2];
> tbound:=args[3];
> a:=op(1, rhs(tbound));
> b:=op(2, rhs(tbound));
> if not type (zbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dzdrdt type` ) fi;
> f:=op(1, rhs(zbound));
> g:=op(2, rhs(zbound));
> h:=op(1, rhs(rbound));
> k:=op(2, rhs(rbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(r1,t1));
> g1:=unapply(g,(r1,t1));
> h1:=unapply(h,t1);
```

```
> k1:=unapply(k,t1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[vloc,uloc,f1(vloc,uloc)],[vloc,uloc,g1(vloc,uloc)],
> [h1(uloc),uloc,wloc], [k1(uloc),uloc,xloc],[uloc1,a,yloc],[uloc2,b,zloc]};
> if nargs = 3 then
> p1:=plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); 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 :
#5. d(theta)dzdr
> dtdzdrplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, r1, t1, z1, u,v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq, rbound,
> tbound, zbound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
```

```
> tbound:=args[1];
> zbound:=args[2];
> rbound:=args[3];
> a:=op(1, rhs(rbound));
> b:=op(2, rhs(rbound));
> if not type (zbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dtdzdr type` ) fi;
> f:=op(1, rhs(tbound));
> g:=op(2, rhs(tbound));
> h:=op(1, rhs(zbound));
> k:=op(2, rhs(zbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(z1,r1));
> g1:=unapply(g,(z1,r1));
> h1:=unapply(h,r1);
> k1:=unapply(k,r1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
```

```
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[uloc,f1(vloc,uloc),vloc],[uloc,g1(vloc,uloc),vloc],
> [uloc,wloc,h1(uloc)], [uloc,xloc,k1(uloc)],[a,yloc,uloc1],[b,zloc,uloc2]};
> if nargs = 3 then
> p1:=plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); 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 :
#6. d(theta)drdz
> dtdrdzplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, r1, t1, z1, u, v,
> uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq, rbound,
> tbound, zbound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> tbound:=args[1];
> rbound:=args[2];
> zbound:=args[3];
> a:=op(1, rhs(zbound));
> b:=op(2, rhs(zbound));
> if not type (zbound, string = range) or not (op(1, rbound) = 'r')
> or not (op(1, tbound) = 'theta') or not (op(1, zbound) = 'z')
```

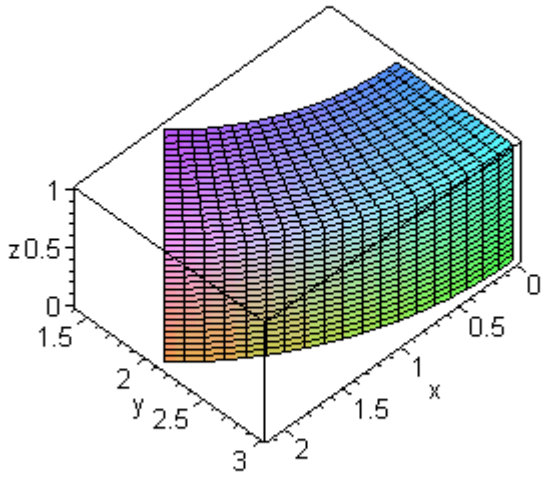
```
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dtdrdz type` ) fi;
> f:=op(1, rhs(tbound));
> g:=op(2, rhs(tbound));
> h:=op(1, rhs(rbound));
> k:=op(2, rhs(rbound));
> r1:=lhs(rbound);
> t1:=lhs(tbound);
> z1:=lhs(zbound);
> f1:=unapply(f,(r1,z1));
> g1:=unapply(g,(r1,z1));
> h1:=unapply(h,z1);
> k1:=unapply(k,z1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[vloc,f1(vloc,uloc),uloc],[vloc,g1(vloc,uloc),uloc],
> [h1(uloc),wloc,uloc], [k1(uloc),xloc,uloc],[uloc1,yloc,a],[uloc2,zloc,b]};
> if nargs = 3 then
> plots[cylinderplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[20,20],scaling=CONSTRAINED,style=PATCH); fi;
```

```
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> plots[cylinderplot](surface, u=0..1, v=0..1),opt_seq; fi;
> end :
```

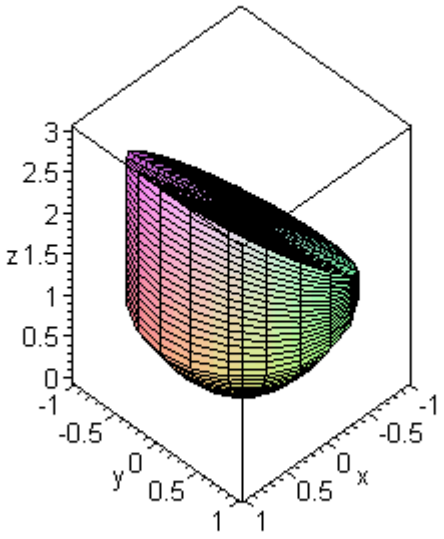
Exemplos

##Exemplos ## drdtdzplot e dzdtdrplot

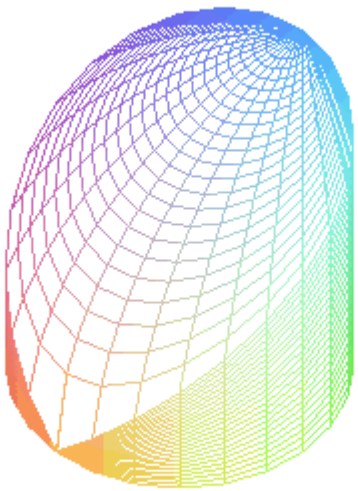
```
> drdtdzplot(r = 2..3, theta = Pi/4..Pi/2,z = 0..1);
```



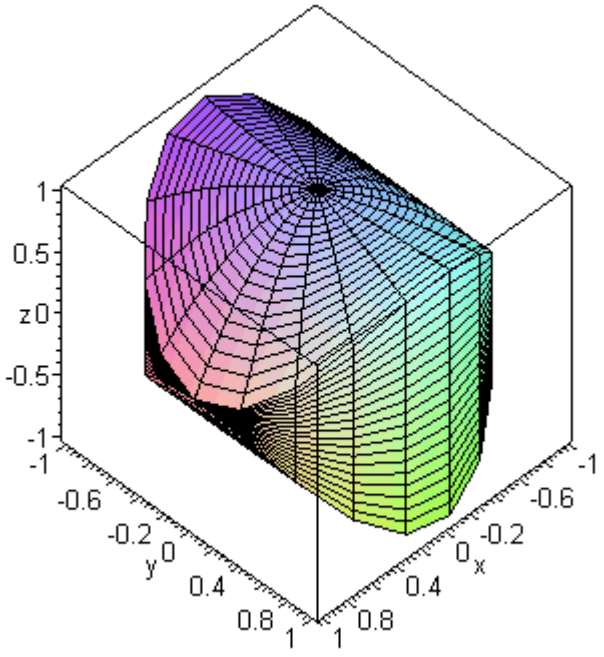

```
> dzdtdrplot(z = r^2..2+r*cos(theta), theta=0..2*Pi, r = 0..1);
```



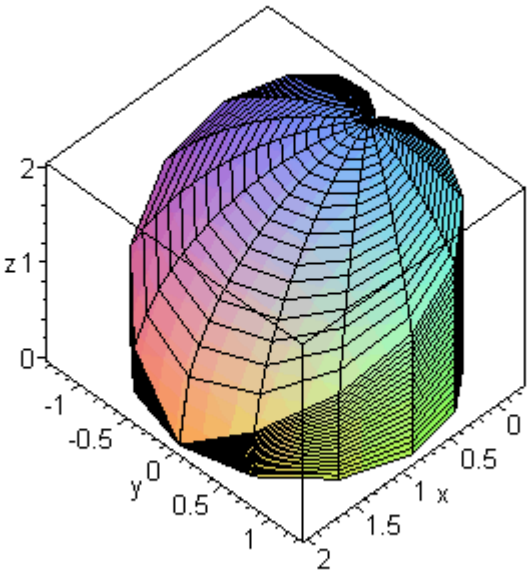
> **dzdrdtplot(z=0..(1-r^2)^(0.5),r=0..cos(theta),theta=0..Pi, grid=[40,15]);**



> **dzdtdrplot(z=-(1-r^2*cos(theta)^2)..(1-r^2*cos(theta)^2),theta=0..2*Pi,r=0..1);**



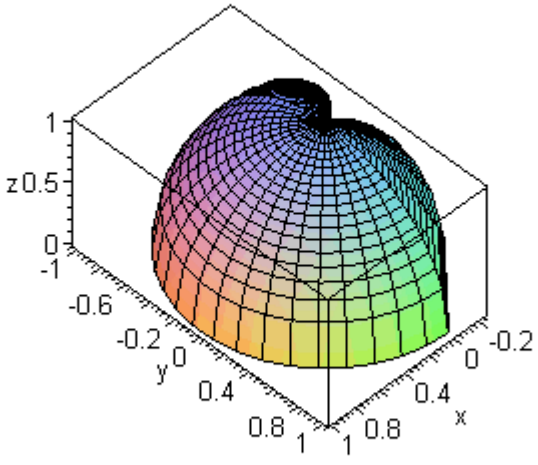
> **dzdrdtplot(z=0..sqrt(4-r^2),r=0..1+cos(theta),theta=0..2*Pi);**



> **p1:=dzdrdtplot(z=0..sqrt(1-r^2),r=0..1,theta=-Pi/2..Pi/2):**

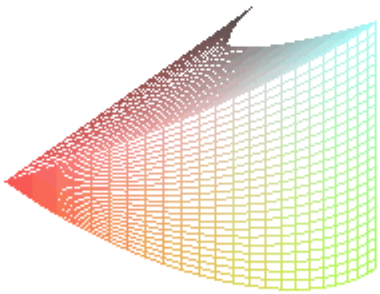
> **p2:=dzdrdtplot(z=0..sqrt(1-r^2),r=0..1+cos(theta),theta=Pi/2..3*Pi/2):**

> **plots[display3d]({p1,p2});**



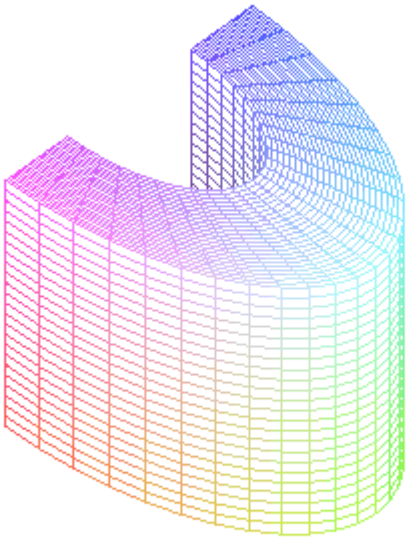
> **dzdrdtplot(z = 0..r*sin(theta),r=2*sin(theta)..2*cos(theta),theta= 0..Pi/4,title=`regiao do 18.7-2`);**

regiao do 18.7-2



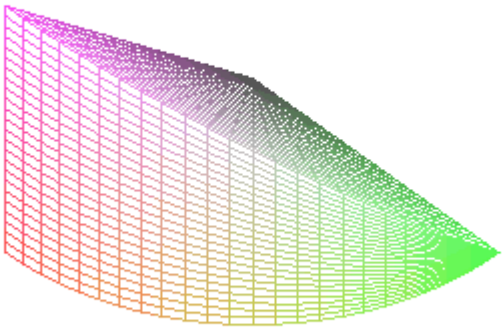
> **dzdrdtplot(z = 0..1,r=2..4,theta= 0..Pi, title=`regiao do 18.7-3`);**

regiao do 18.7-3



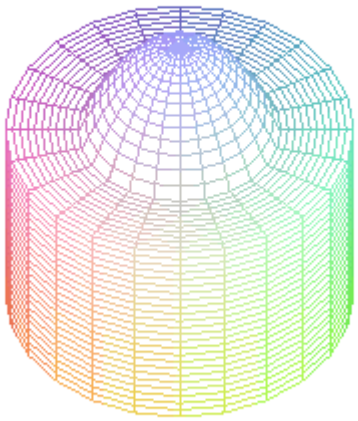
> **dzdrdtplot(z = 0..r*cos(theta),r=0..1,theta= 0..Pi/2, title=`regiao do 18.7-9`);**

regiao do 18.7-9



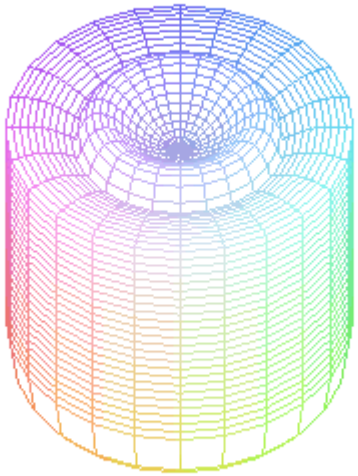
> **dzdrdtplot(z = 8..12-r^2,r=0..sqrt(12),theta= 0..2*Pi, title=`regiao do 18.7-11`);**

regiao do 18.7-11



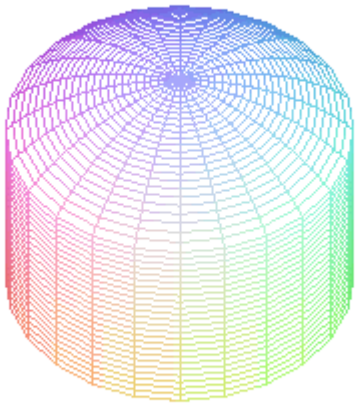
> **dzdrdtplot(z =10*r^2*cos(r^2)..sqrt(4-10*sin(cos(r^2))),r=0..2,theta= 0..2*Pi,style=PATCH, title=`outra regiao- logotipo`);**

outra regiao- logotipo



> **dzdrdtplot(z =-sqrt(25-r^2)..sqrt(25-r^2),r=0..4,theta= 0..2*Pi, title=`esfera inters cilin.`);**

esfera inters cilin.



>