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 o gráfico de funções da forma z=h(x,y) sobre o seu

dominio do tipo I : f(x)<=y<=g(x), a<=x<=b ou

do tipo II: r(y)<=x<=s(y),c<=y<=d .

O Caso especial h=0 produz o desenho da região acima no plano

xy. #Note que estes tipos de plots podem ser obtidos diretamente

com o comando 3d plot; entretanto, a orientação nos plots dá um pouco

de trabalho.

Execute o procedimento e faça os exemplos.

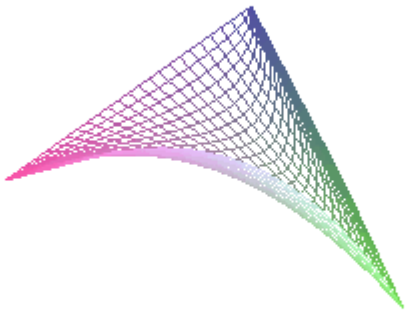
O Procedimento (execute-o)

Exemplos

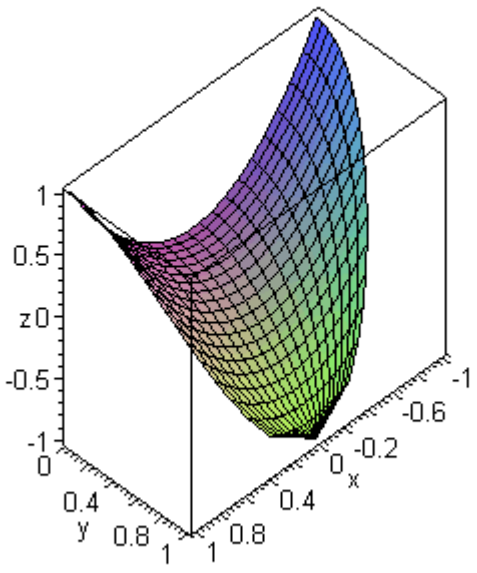
Exemplos yxgraphplot e xygraphplot

```
> yxgraphplot(x*y,y=0..sqrt(1-x^2),x=-1..1,title=`exemplo`);
```

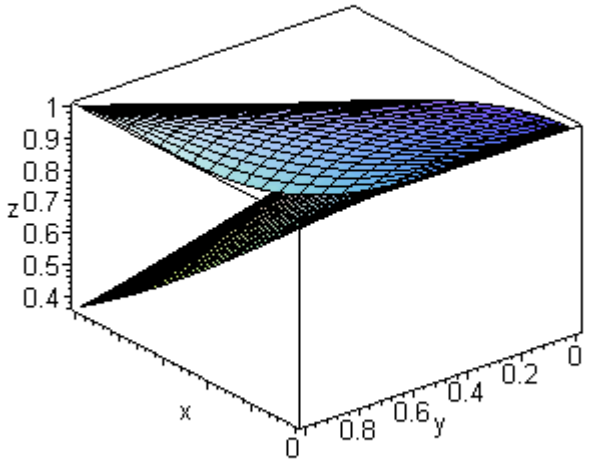
exemplo



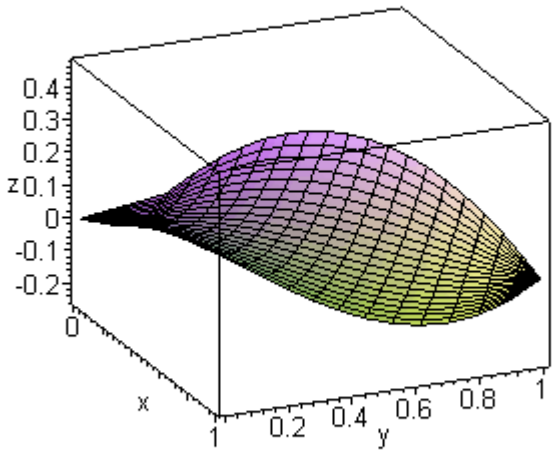
```
> #yxgraphplot((x,y)->sin(Pi*(x^2+y^2)),y=x^2..x^(1/4),x=0..1);
> #xygraphplot(x*y,x=-sqrt(1-y^2)..sqrt(1-y^2),y=0..1);
> #yxgraphplot(x^2+y^2,y=0..sqrt(1-x^2),x=-1..1);
> #yxgraphplot(x^2-y^2,y=0..sqrt(1-x^2),x=-1..1);
> xygraphplot(x^2-y^2,x=-sqrt(1-y^2)..sqrt(1-y^2),y=0..1);
```



```
> #yxgraphplot(x*y,y=0..sqrt(1-x^2),x=-1..1);
> #yxgraphplot(exp(-x*y),y=x^2..x^(1/4),x=0..1);
> #yxgraphplot(cos(x^2-y^2),y=x^2..x^(1/4),x=0..1);
> #yxgraphplot(cos(x^2-y^2),y=-1..1,x=-1..1);
> p1:= yxgraphplot(exp(-x*y),y=x^2..x^(1/4),x=0..1):
> p2:= yxgraphplot(cos(x^2-y^2),y=x^2..x^(1/4),x=0..1):
> plots[display3d]({p1,p2});
```




```
> xygraphplot(x^2-y^2,x=y^2..y^(1/4),y=0..1);
```


0.4
0.3
0.2
z 0.1
0
-0.1
-0.2
0
x
1
0.2
0.4
0.6
0.8
1
y

>

O Procedimento (execute-o)

Regioes do tipo I

```
> yxgraphplot:=proc()

> local a, b, c, d, g, h, i, k, f1, g1, h1, opt_seq, p1, tloc, sloc,

> surface, xbound, ybound;

> if nargs < 3 then

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

> h:=args[1];

> ybound:=args[2];

> xbound:=args[3];

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

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

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

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

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

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

> if not (lhs(xbound) = 'x') then

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

> if not (lhs(ybound) = 'y') then

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

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

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

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

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

> g:=lhs(xbound);
```

```
> k:=lhs(ybound);
> 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:=plot3d(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:=plot3d(surface, u=0..1,v=0..1),opt_seq; fi;
> plots[display3d](p1);
> end:
```

Regioes do tipo II

```
> xygraphplot:=proc()
> local a, b, c, d, g, h, i, k, f1, g1, h1, opt_seq, p1, sloc, tloc,
> surface, xbound, ybound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> h:=args[1];
> xbound:=args[2];
> ybound:=args[3];
```

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

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

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

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

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

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

> if not (lhs(xbound) = 'x') then

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

> if not (lhs(ybound) = 'y') then

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

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

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

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

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

> g:=lhs(xbound);

> k:=lhs(ybound);

> 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)];

> if nargs = 3 then
```

```
> p1:=plot3d(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:=plot3d(surface,u=0..1,v=0..1),opt_seq; fi;
> plots[display3d](p1);
> end:
```