Mais sobre curvas

SurfLoop - plota uma curva bidimensional parametrizada sobre uma superficie

SINTAXE: SurfLoop(f,x=a..b,y=c..d,[x(t),y(t)],t=e..f)

SurfLoop(f,x=a..b,y=c..d,[x(t),y(t)],t=e..f,ops)

PARAMETROS:

f - a superficie

x - a primeira variavel independente de f

a..b - a variação do plot da primeira variavel

y - a segunda variavel independente de f

c..d - a variação do plot da segunda variavel

x(t) - a primeira componente da curva parametrizada

y(t) - a segunda componente da curva parametrizada

t - a variavel indenpendente

e..f - a variação da variavel parametrica

ops - comandos opcionais de plot3d

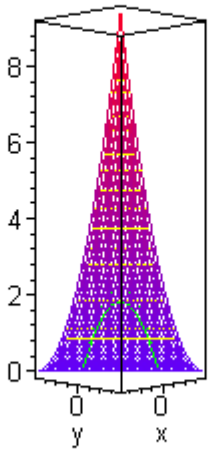
RESUMO: SurfLoop(f,x=a..b,y=c..d,[x(t),y(t)],t=e..f) desenha uma curva bidimensional parametrizada [x(t),y(t)] sobre uma superficie tridimensional dada por f.

Execute este procedimento e faça os exemplos.

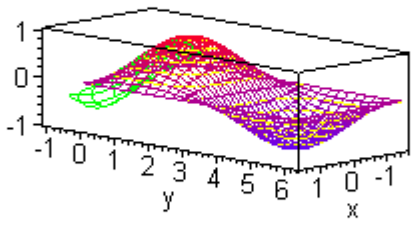
O Procedimento (execute-o)

Exemplos

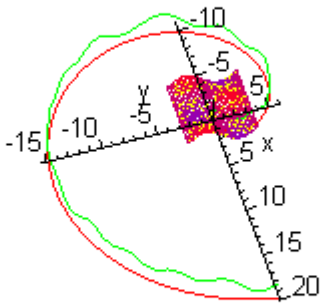
```
> SurfLoop((x+y)^2,x=-1.5..1.5,y=-1.5..1.5,[cos(t),sin(t)],t=0..2*Pi);
```



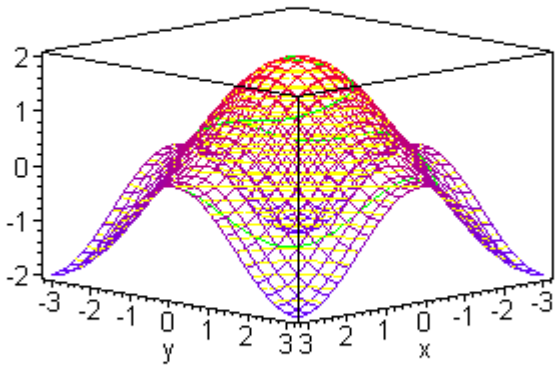
> **SurfLoop(cos(x)*sin(y),x=-Pi/2..Pi/2,y=0..2*Pi,[cos(5*t),sin(4*t)],t=0..2*Pi);**



> **SurfLoop(2+cos(x)*sin(y),x=-Pi..Pi,y=-Pi..Pi,[t*Pi*cos(t),t*Pi*sin(t)],t=0..2*Pi,axes=NORMAL,project,orientation=[-17,36]);**



> **SurfLoop(cos(x)+cos(y),x=-Pi..Pi,y=-Pi..Pi,[(Pi-t/4)*cos(t),(Pi-t/4)*sin(t)],t=0..4*Pi);**



>

O Procedimento (execute-o)

```
> SurfLoop := proc(surf:{algebraic,procedure},xrange:name=range(constant),
> yrange:name=range(constant),loo:{vector(algebraic),
> list(algebraic)},trange:name=range(constant))
> global piece4;
> local x,y,t,S,piece1,piece2,piece3,nmpts,opargs,n,loop,proj;
> proj := false;
> x := op(1,xrange);
> y := op(1,yrange);
> t := op(1,trange);
> loop := convert(loo,list);
> if type(surf,procedure) then
> if nops({op(1,op(1,surf))})=2 then
> if nops(indets(surf(x,y),name) minus
> indets(surf(x,y),constant) minus {x,y})=0 then
> S:=surf
> else
> ERROR(`the first argument contains undefined parameters.`) fi
> else
> ERROR(`first argument can have only two independent variables.`) fi;
> else
> if nops(indets(surf,name) minus indets(surf,constant) minus {x,y})=0 then
> if member(x,indets(surf,name)) or member (y,indets(surf,name)) then
> S := traperror(unapply(surf,x,y));
> if S=lasterror then
> ERROR(`unable to construct function.`) fi;
```

```
> else
> ERROR(`independent variable mismatch betwen arguments.`)
> fi
> else
> ERROR(`the first argument contains undefined parameters.`)
> fi
> fi;
> if nops(loop)<>2 then
> ERROR(`the fourth argument must be a two component list.`) fi;
> if nops(indets(loop,name) minus indets(loop,constant) minus {t})>0 then
> ERROR(`the fourth argument has undefined parameters.`) fi;
> nmpts := 100;
> opargs := [];
> if nargs>5 then
> for n from 6 to nargs do
> if op(1,args[n])=numpoints then
> nmpts:=op(2,args[n])
> elif op(1,args[n])=color then
> print(`The color option is not available.`)
> elif args[n]=project then
> proj := true
> else
> opargs := [opargs[],args[n]];
> fi;
> od;
> fi;
```

```
> piece1 := plots[spacecurve]([loop[1],loop[2],S(loop[1],loop[2])],
> trange,color=green,numpoints=nmpts):
> piece2 := plot3d(S(x,y),xrange,yrange,shading=Z,style=WIREFRAME):
> piece3 := plot3d(S(x,y),xrange,yrange,style=CONTOUR,color=yellow):
> if proj then
> piece4 := plots[spacecurve]([loop[1],loop[2],0],trange,color=red,
> numpoints=nmpts)
> else
> piece4 := PLOT3D()
> fi;
> plots[display]([piece3,piece1,piece2,piece4],opargs[]);
> end:
```