CURVAS SOBRE SUPERFÍCIES

FUNÇÃO: anima uma curva parametrizada bidimensional sobre uma superficie.

Sintaxe: Loopimate(f,x=a..b,y=c..d,[x(t),y(t)],t=e..f) ou

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

PARÂMETROS:

f - a superficie tridimensional`,

x - a primeira variavel independente de f,

a..b - variação da primeira variavel`,

y - a segunda variavel independente de f,

c..d - a variação da segunda variavel,

x(t) - a primeira componente da curva parametrizada,

y(t) - a segunda componente da curva parametrizada,

t - a variavel independente da curva,

e..f - variação da variavel parametrica,

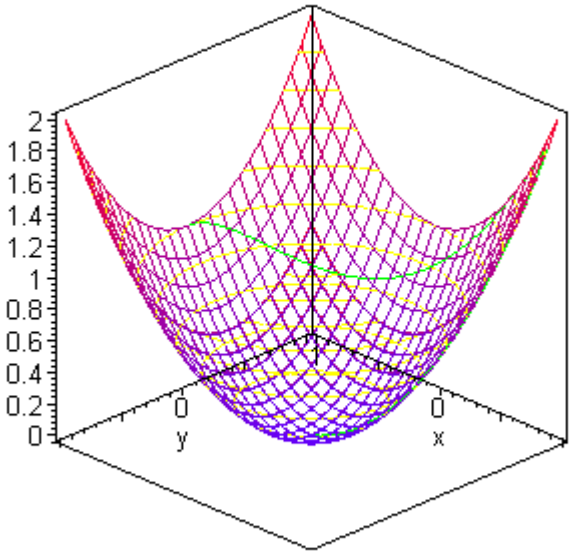
ops - opções

Execute este procedimento e faça os exemplos .

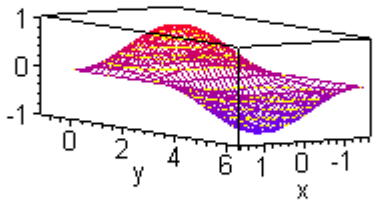
O Procedimento (execute-o)

Exemplos

> **Loopimate(x^2+y^2,x=-1..1,y=-1..1,[cos(3*t),sin(2*t)],t=0..2*Pi);**



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



>

O Procedimento (execute-o)

```
> Loopimate := proc(surf::{algebraic,procedure},xrange::name=range(constant),
> yrange:name=range(constant),loo:{vector(algebraic),
> list(algebraic)},trange:name=range(constant))
> local x,y,t,S,piece2,piece3,n,plotset,tend,tstart,loop,frms,grd,nmpts;
> 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 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 between arguments.`)
> fi
> else
> ERROR(`the first argument contains parameters that must be defined.`)
> fi
> fi;
> if nops(loop)<>2 then
> ERROR(`the fourth argument must be a list of only two components.`) fi;
> if nops(indets(loop,name) minus indets(loop,constant) minus {t})>0 then
> ERROR(`the fourth argument has parameters that must be defined.`) fi;
> grd := [25,25];
> nmpts := 100;
> frms := 8;
> if nargs>5 then
> for n from 6 to nargs do
> if op(1,args[n])=grid and type(op(2,args[n]),list) then
> grd := op(2,args[n])
> elif op(1,args[n])=numpoints then
> nmpts := op(2,args[n])
> elif op(1,args[n])=frames then
> frms := op(2,args[n])
> else
> ERROR(`optional argument not supported.`)
> fi
> od
> fi;
```

```
> tstart := op(1,op(2,trange));
> tend := op(2,op(2,trange));
> piece2 := plot3d(S(x,y),xrange,yrange,shading=Z,style=WIREFRAME,grid=grd):
> piece3 := plot3d(S(x,y),xrange,yrange,style=CONTOUR,color=yellow,grid=grd):
> plotset := [plots[display]([piece3,piece2,PLOT3D(CURVES([evalf(subs(t=0,
> [loop[1],loop[2],S(loop[1],loop[2])]))]))])];
> for n from 1 to frms do
> plotset := [plotset[],plots[display]([piece3,piece2,
> plots[spacecurve]([loop[1],loop[2],S(loop[1],loop[2])],
> t=tstart..tstart+(tend-tstart)/frms*n,color=green,
> numpoints=nmpts)])]:
> od;
> plots[display](plotset,insequence=true);
> end:
```