p34ed

(parcialillo 34 de ecuaciones)

Problema 1.

```
var('t');X=function('X',t);et=(1+t)*t*diff(X,t,2)-diff(X,t);
desolve(et,X);S=_.subs(k1=1,k2=0);S;taylor(S,t,0,5);limit(S,t=oo)
```

```
(t - log(t + 1))*k1 + k2
t - log(t + 1)
-1/5*t^5 + 1/4*t^4 - 1/3*t^3 + 1/2*t^2
+Infinity
```

```
R5.<c0,c1,c2,c3,c4,c5>=PolynomialRing(QQ,6)
R.<t>=PowerSeriesRing(R5)
x1=c0*t^2+c1*t^3+c2*t^4+c3*t^5+c4*t^6+c5*t^7+O(t^8)
xp=x1.derivative();xs = xp.derivative()
show(t*(1+t)*xs - xp)
```

$$(2c_0 + 3c_1)\,t^2 + (6c_1 + 8c_2)\,t^3 + (12c_2 + 15c_3)\,t^4 + (20c_3 + 24c_4)\,t^5 + (30c_4 + 35c_5)\,t^6$$

```
taylor(1/t/(1+t),t,0,7)
```

```
t^7 - t^6 + t^5 - t^4 + t^3 - t^2 + t + 1/t - 1
```

```
R.<t> = PowerSeriesRing(QQ, default_prec=10);
a=t^7-t^6+t^5-t^4+t^3-t^2+t-1+O(t^8); b=0+O(t^7);
f = a.solve_linear_de(prec=6,b=b,f0=1); f; f.derivative()-a*f-b
```

```
1 - t + t^2 - t^3 + t^4 - t^5 + O(t^6)
O(t^5)
```

```
search_doc("solve_linear_de")
```

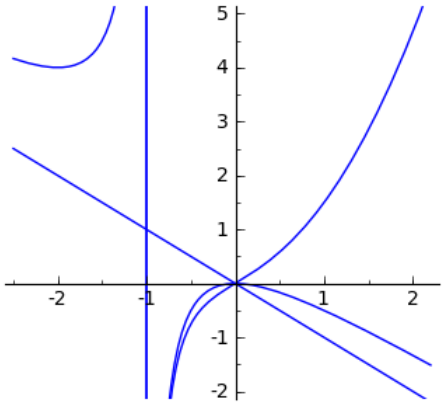
Search Documentation: "solve_linear_de"

1. constructions/calculus.html
2. reference/genindex-S.html
3. reference/genindex-all.html
4. reference/sage/rings/power_series_ring_element.html

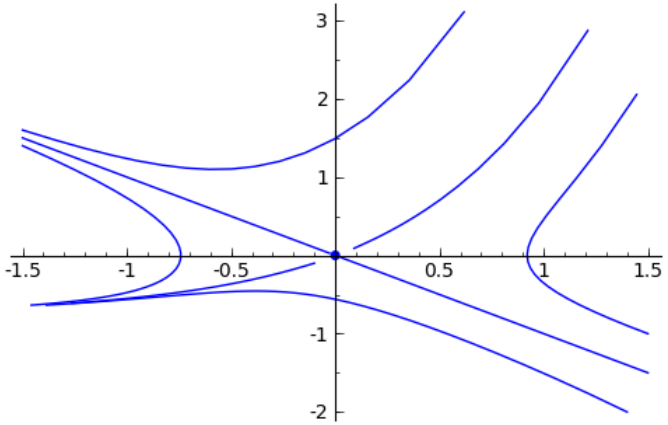
Problema 2.

```
var('x,y');f=(x+x*y+y^2)/y
mx=solve(f,x);nf=solve(diff(f,x)+diff(f,y)*f,x)
mn=[-y^2/(y+1),-y,(y^3+y^2+y)/(y+1)]
mx,nf;plot(mn,y,-2.5,2.2,ymin=-2,ymax=5,figsize=[4,3.5])
```

```
([x == -y^2/(y + 1)], [x == -y, x == (y^3 + y^2 + y)/(y + 1)])
```




```
var('X Y t');S=[Y,X+X*Y+Y^2];V=[X,Y]
A=desolve_system_rk4(S,V,ics=[0,1.5,-1.5],ivar=t,end_points=4)
Q1=[[j,k] for i,j,k in A];L1=list_plot(Q1,plotjoined=True)
B=desolve_system_rk4(S,V,ics=[0,-1.5,1.5],ivar=t,end_points=4)
Q2=[[j,k] for i,j,k in B];L2=list_plot(Q2,plotjoined=True)
C=desolve_system_rk4(S,V,ics=[0,0.09,0.09],ivar=t,end_points=2)
Q3=[[j,k] for i,j,k in C];L3=list_plot(Q3,plotjoined=True)
D=desolve_system_rk4(S,V,ics=[0,-0.1,-0.1],ivar=t,end_points=3.4)
Q4=[[j,k] for i,j,k in D];L4=list_plot(Q4,plotjoined=True)
F=desolve_system_rk4(S,V,ics=[0,1.4,-2],ivar=t,end_points=4)
Q5=[[j,k] for i,j,k in F];L5=list_plot(Q5,plotjoined=True)
G=desolve_system_rk4(S,V,ics=[0,1.5,-1],ivar=t,end_points=2)
Q6=[[j,k] for i,j,k in G];L6=list_plot(Q6,plotjoined=True)
K=desolve_system_rk4(S,V,ics=[0,-1.5,1.4],ivar=t,end_points=3)
Q7=[[j,k] for i,j,k in K];L7=list_plot(Q7,plotjoined=True)
L=desolve_system_rk4(S,V,ics=[0,-1.5,1.6],ivar=t,end_points=1.5)
Q8=[[j,k] for i,j,k in L];L8=list_plot(Q8,plotjoined=True)
L1+L2+L3+L4+L5+L6+L7+L8+point([(0,0)],pointsize=20)
```



Problema 3.

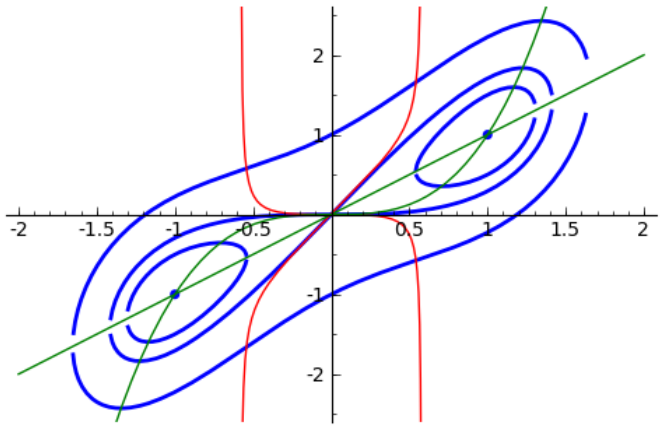
```
var('x,y,C');f=(y-x^3)/(y-x);Y=function('Y',x);
si=desolve(diff(Y,x)==f.subs(y=Y),Y)
nf=solve(diff(f,x)+diff(f,y)*f,y);
f1=factor(nf[0]).right();f2=factor(nf[1]).right()
si;H=x^4-4*x*y+2*y^2;show(solve(H==C,y));show(f2)
```

```
1/4*x^4 - x*Y(x) + 1/2*Y(x)^2 == c
```

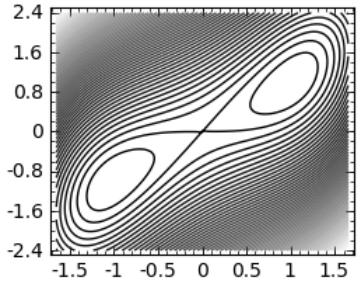
$$\left[y = -\frac{1}{2}\sqrt{-x^4+2\,x^2+C}\sqrt{2}+x, y = \frac{1}{2}\sqrt{-x^4+2\,x^2+C}\sqrt{2}+x\right]$$

$$\frac{\left(\sqrt{-3\,x^2+1}x^2-\sqrt{-3\,x^2+1}+3\,x^2-1\right)x}{\left(3\,x^2-1\right)}$$

```
sq=sqrt(C+x^2-x^4/2);s1(C)=x+sq;s2(C)=x-sq
d1=plot([s1(0),s2(0)],x,-1.41,1.41,thickness=2)
d2=plot([s1(1),s2(1)],x,-1.7,1.7,thickness=2)
d3=plot([s1(-1/4),s2(-1/4)],x,-1.3,-0.51,thickness=2)
d4=plot([s1(-1/4),s2(-1/4)],x,0.51,1.3,thickness=2)
d5=plot([f1,f2],x,-0.6,0.6,ymin=-2.5,ymax=2.5,color='red')
d6=plot([x^3,x],x,-2,2,ymin=-2.5,ymax=2.5,color='green')
dp=point([(0,0),(1,1),(-1,-1)],pointsize=20)
show(d1+d2+d3+d4+d5+d6+dp,figsize=[5,3])
```



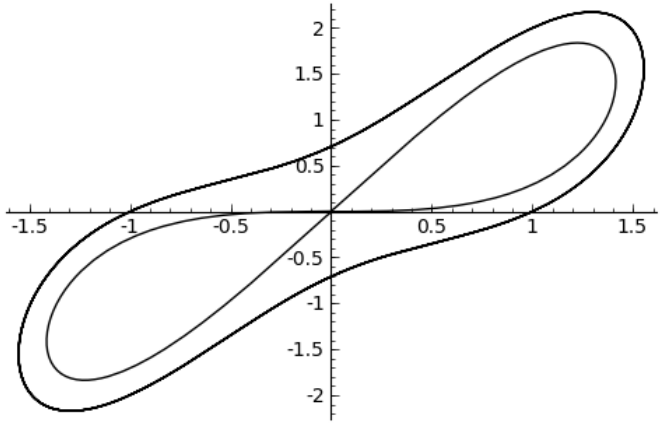

```
contour_plot(H,(x,-1.65,1.65),(y,-2.4,2.4),
contours=80,fill=False,figsize=[3,2])
```



Utilizando **Math 265 Introduction to Systems** obtenida en http://www.sagenb.org/pub/

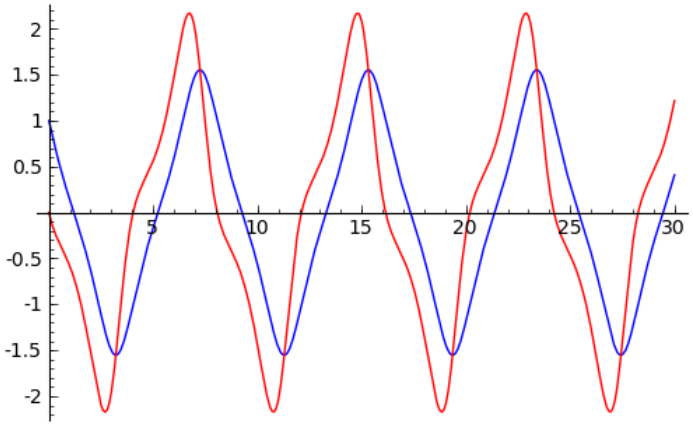
```
x,y,t=var('x y t')
class SA:
    def __init__(ff,point_list):
        ff.T=[i for (i,j,k) in point_list]
        ff.X=[j for (i,j,k) in point_list]
        ff.Y=[k for (i,j,k) in point_list]
    def MF(ff,color='black'):
        return line(zip(ff.X,ff.Y),color=color)
    def Coords(ff,info=True):
        if info: print "x en azul, y en rojo."
        return
line(zip(ff.T,ff.X),color='blue')+line(zip(ff.T,ff.Y),color='red')
class Flujo():
    def __init__(ff,de_list):
        ff.des=de_list
    def MF(ff):

colores=['black','blue','green','red','brown','orange','yellow']
        colors=[choice(colores) for i in range(len(ff.des))]
        return sum([z.MF(color=colors[i]) for i,z in
enumerate(ff.des)])
    def Coords(ff,info=True):
        if info: print "x en azul, y en rojo."
        return sum([z.Coords(info=False) for z in ff.des])
def SS(ss,IvT,CI,stepsize=0.05):
    try:

soln=SA(desolve_system_rk4(ss,[x,y],ics=[IvT[1],CI[0],CI[1]], \
        ivar=IvT[0],end_points=IvT[2],step=stepsize))
    except:
        print "Oops!  Falló la integración numérica."
        soln=[]
    return soln
def SF(ss,IvT,rect,stepsize=0.05,flows=20):
    flowsfix=max(1,int(flows))
    a,b=rect[0];c,d=rect[1]
    def r(): return (a+(b-a)*random(),c+(d-c)*random())
    icdata=[r() for i in range(flowsfix)]
    return Flujo([SS(ss,IvT,z,stepsize=stepsize) for z in icdata])
```

```
F=[y-x,y-x^3]
s1=SS(F,[t,0,30],[1,0]);s2=SS(F,[t,0,30],[0.1,0]);s1.MF()+s2.MF()
```




```
s1.Coords()
```

x en azul, y en rojo.




```
fl=SF(F,[t,0,12],[[-1.0,1.0],[-0.5,0.5]]);fl.MF()
```