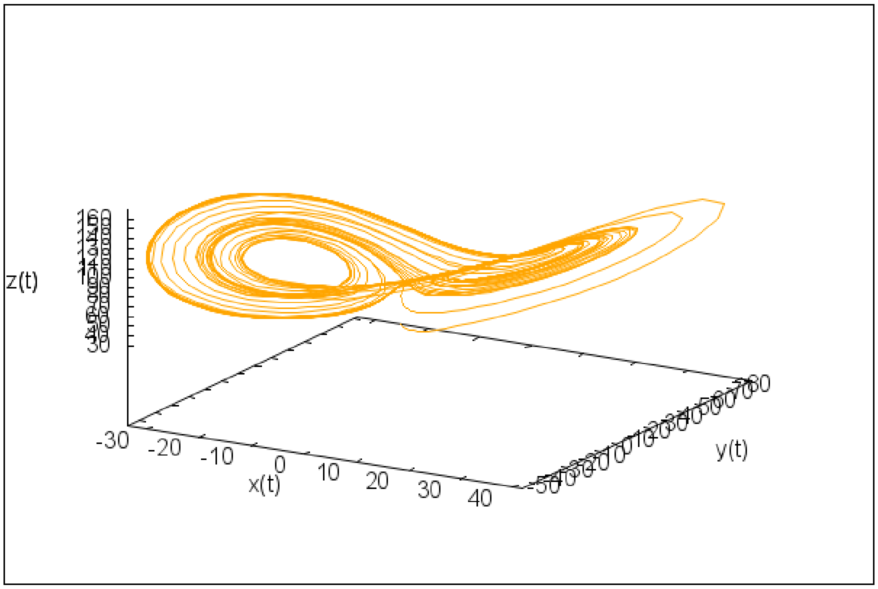
```
–> plotdf(r + 1*sin(y), [x, -50, 50], [parameters, ”r=0.9”], [sliders, ”r=0:10”])$

–> plot2d(0.9 + 1*sin(y), [y, -10, 10])$;

–> /* harmonic oscillator */ plotdf([v,-k*x/m], [x,v], [parameters,”m=2,k=2”], [sliders,”m=1:5”], [trajectory_at,6,0])$

–> /* damped harmonic oscillator */ plotdf([v,-k*x/m - 0.1*v], [x, v], [parameters,”m=2,k=2”], [sliders,”m=1:5”], [trajectory_at,6,0])$

–> /* Example 5.1.2 */ kill(a)$; eq1: a*x$; eq2: -y$; plotdf([eq1, eq2], [x, y], [x, -50, 50], [y, -50, 50], [parameters,”a=1”], [sliders,”a=-5:5”], [trajectory_at,0.5,0.5])$

–> /* Example 5.2.2 */ eq1: x + y$; eq2: 4*x - 2*y$; plotdf([eq1, eq2], [x, y], [x, -50, 50], [y, -50, 50], [trajectory_at,0.5,0.5])$

–> /* Exercise 5.3.2 */ R_: J$; J_: -R + J$; plotdf([R_, J_], [R, J], [R, -5000, 5000], [J, -5000, 5000], [trajectory_at,0.5,0.5])$

–> /* Exercise 5.3.3 */ kill(a)$; kill(b)$; R_: a*J$; J_: b*R$; plotdf([R_, J_], [R, J], [R, -5000, 5000], [J, -5000, 5000], [parameters,”a=1, b=1”], [sliders,”a=-5:5, b=-5:5”], [trajectory_at,0.5,0.5])$

–> /* Exercise 5.3.4 */ kill(a)$; kill(b)$; R_: a*R + b*J$; J_: -b*R - a*J$; plotdf([R_, J_], [R, J], [R, -5000, 5000], [J, -5000, 5000], [parameters,”a=1, b=1”], [sliders,”a=-5:5, b=-5:5”], [trajectory_at,0.5,0.5])$

–> /* Exercise 5.3.5 */ kill(a)$; kill(b)$; R_: a*R + b*J$; J_: b*R + a*J$; plotdf([R_, J_], [R, J], [R, -5000, 5000], [J, -5000, 5000], [parameters,”a=1, b=1”], [sliders,”a=-5:5, b=-5:5”], [trajectory_at,0.5,0.5])$

–> /* Exercise 5.3.6 */ kill(a)$; kill(b)$; R_: 0$; J_: a*R + b*J$; plotdf([R_, J_], [R, J], [R, -5000, 5000], [J, -5000, 5000], [parameters,”a=1, b=1”], [sliders,”a=-5:5, b=-5:5”], [trajectory_at,0.5,0.5])$

–> /* Example 6.1.1 */ X_: x + %e**(-y)$; Y_: -y$; scale: 5$; plotdf([X_, Y_], [x, y], [x, -scale, scale], [y, -scale, scale], [trajectory_at,-2,0], [trajectory_at,-0.5,0])$

–> /* Example 6.3.1 */ X_: -x + x**3$; Y_: -2*y$; scale: 5$; plotdf([X_, Y_], [x, y], [x, -scale, scale], [y, -scale, scale], [trajectory_at,-2,0], [trajectory_at,-0.5,0])$

–> /* Rabbits and Sheeps */ X_: x * (3 - x - 2*y)$; Y_: y * (2 - x - y)$; scale0: 0$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, -scale0, scale1], [y, -scale0, scale1], [trajectory_at,-2,0])$

–> /* Example 6.5.1 */ X_: y$; Y_: x - x**3$; scale0: -5$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [trajectory_at,-2,0])$

–> /* Pendulum */ %theta_: v$; V_: -sin(%theta)$; scale0: -5$; scale1: 5$; plotdf([%theta_, V_], [%theta, v], [v, scale0, scale1], [%theta, scale0, scale1], [trajectory_at,-2,0])$

–> /* Damped Pendulum */ %theta_: v$; V_: -0.1*v -sin(%theta)$; scale0: -5$; scale1: 5$; plotdf([%theta_, V_], [%theta, v], [v, scale0, scale1], [%theta, scale0, scale1], [trajectory_at,-2,0])$
```

```
-->     load(physical_constants);

-->     /* Exercise 6.3.8 */ kill(a)$; m1: 10E10$; m2: 20E10$; G: 6.673e-11$ X_: y$; Y_: G * m2/(x - a) * G * m1/(x**2)$; scale0: -5$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”a=3”], [sliders,”a=-50:50”], [trajectory_at,-2,0])$

-->     /* Example 7.1.1 */ R_: r * (1 - r**2)$; %theta_: 1$; scale0: -5$; scale1: 5$; plotdf([R_, %theta_], [r, %theta], [r, scale0, scale1], [%theta, scale0, scale1], [trajectory_at,-2,0])$

-->     /* Van Der Pol Oscillator */ kill(MU)$; X_: y$; Y_: -MU*(x**2 - 1)*y -x$; scale0: -5$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”MU=3”], [sliders,”MU=-10:10”], [trajectory_at,-2,0])$

-->     /* Example 7.1.1 */ R_: r * (1 - r**2)$; %theta_: 1$; scale0: -5$; scale1: 5$; plotdf([R_, %theta_], [r, %theta], [r, scale0, scale1], [%theta, scale0, scale1], [trajectory_at,-2,0])$

-->     /* Example 7.3.3 */ kill(a)$; kill(b)$; X_: -x + a*y + x^(2)*y$; Y_: b - a*y - x^(2)*y$; scale0: -1$; scale1: 3$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”a=0.08,b=0.6”], [sliders,”a=0:10,b=0:10”], [trajectory_at,-2,0])$

-->     /* Example 8.1.3 */ kill(MU)$; X_: MU*x + y + sin(x)$; Y_: x - y$; scale0: -5$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”MU=3”], [sliders,”MU=-10:10”], [trajectory_at,-2,0])$

-->     /* Example 8.2.1 */ kill(MU)$; X_: MU*x - y + + x*y^2$; Y_: x + MU*y + y^3$; scale0: -5$; scale1: 5$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”MU=-0.2”], [sliders,”MU=-0.5:0.5”], [trajectory_at,-2,0])$

-->     /* Example 8.3.3 */ kill(a)$; kill(b)$; X_: a - x - 4*x*y/(1 + x**2)$; Y_: b*x*(1 - y/(1 + x**2))$; scale0: 0$; scale1: 500$; plotdf([X_, Y_], [x, y], [x, scale0, 20], [y, scale0, scale1], [nsteps, 4000], [tstep, 0.01], [parameters,”a=50,b=29.4”], [sliders,”a=0:100,b=0:100”])$

-->     /* Homoclinic Bifurcation */ kill(MU)$; X_: y$; Y_: MU*y + x - x**2 + x*y$; scale0: -2$; scale1: 2$; plotdf([X_, Y_], [x, y], [x, scale0, scale1], [y, scale0, scale1], [parameters,”MU=-0.92”], [sliders,”MU=-5:5”])$
```

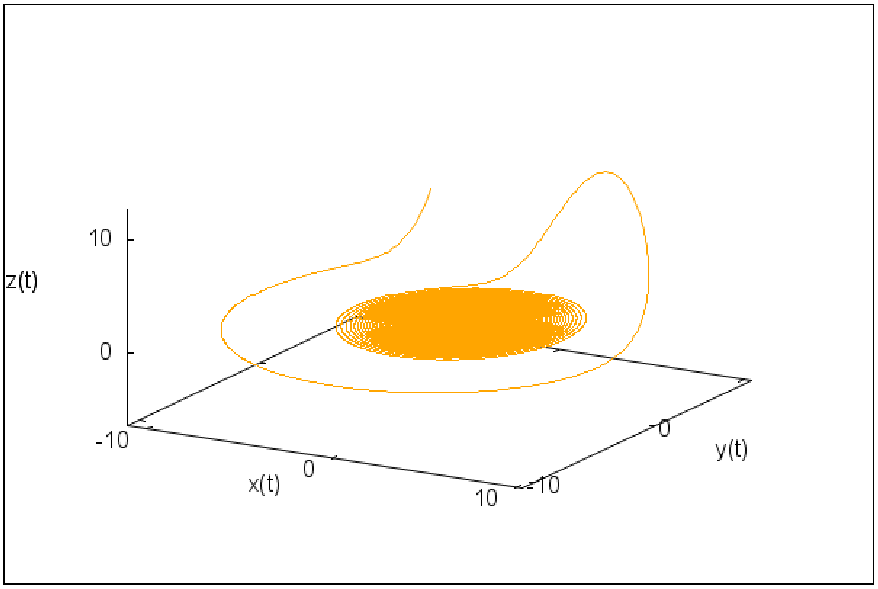
```
-> /* Lorenz equations */ load(draw)$; latractor: [10*(y-x), -x*z+100*x-y, x*y-8*z/3]$ linitial: [-2, 8, 27]$ lsolution:rk(latractor,[x,y,z],linitial,[t,0,50,0.01])$ lpoints: map(lambda([x], rest(x)), lsolution)$ wxdraw3d(point_type=none,points_joined=true,color=orange, xlabel=”x(t)”,ylabel=”y(t)”,zlabel=”z(t)”, xtics=10,ytics=10,ztics=10,points(lpoints));
```



(%t42)

(%o42)

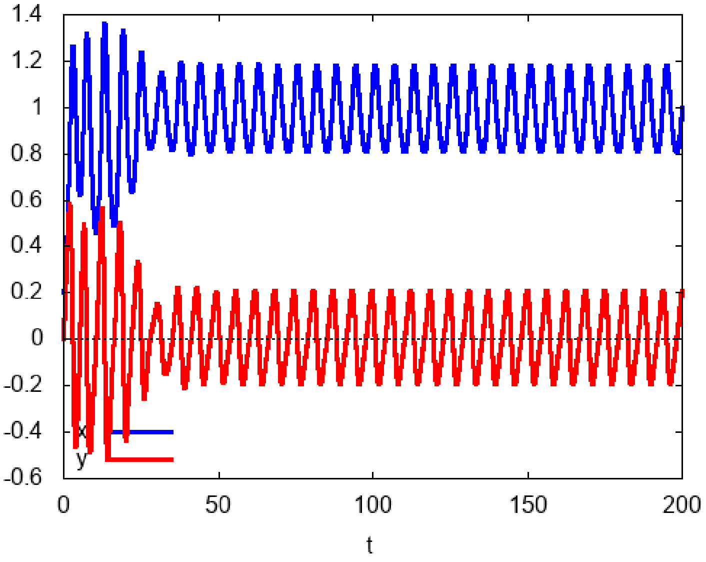
```
-->   /* Rossler System */ load(draw)$; latractor: [-y-z, x+0.2*z, 0.2+z*(x-5.7)]$ linitial: [-7, 8, 6]$ lsolution:rk(latractor,[x,y,z],linitial,[t,0,500,0.01])$ lpoints: map(lambda([x], rest(x)), lsolution)$ wxdraw3d(point_type=none,points_joined=true,color=orange, xlabel=”x(t)”,ylabel=”y(t)”,zlabel=”z(t)”, xtics=10,ytics=10,ztics=10,points(lpoints));
```



(%t90)

(%o90)

```
–>  /* Example 12.5.1 */ kill(x)$; kill(t)$; delta: 0.25$ w: 1$ F: 0.18$ dxdt: y$
    dydt: F*cos(w*t) - delta*y + x - x**3$ points : rk([dxdt, dydt], [x, y], [0.2, 0],
    [t, 0, 200, 0.01] )$ txL : makelist ([points[i][1], points[i][2]], i, 1, length(points))$
    tyL : makelist ([points[i][1], points[i][3]], i, 1, length(points))$ wxplot2d([ [discrete,txL], [discrete,tyL]],[t,0,280], [style,[lines,3]],[xlabel,"t"], [legend, "x", "y"],
    [gnuplot_preamble,"set key bottom left;"], [gnuplot_term, aqua]);
```

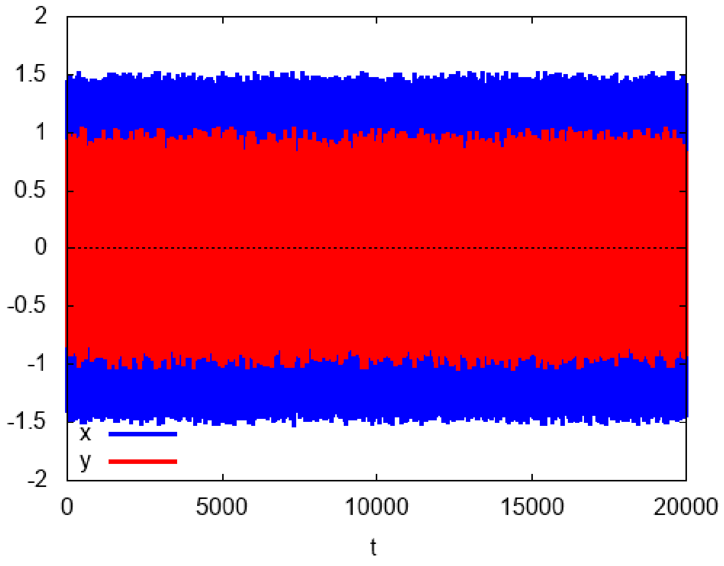


(%t85)

(%o85)

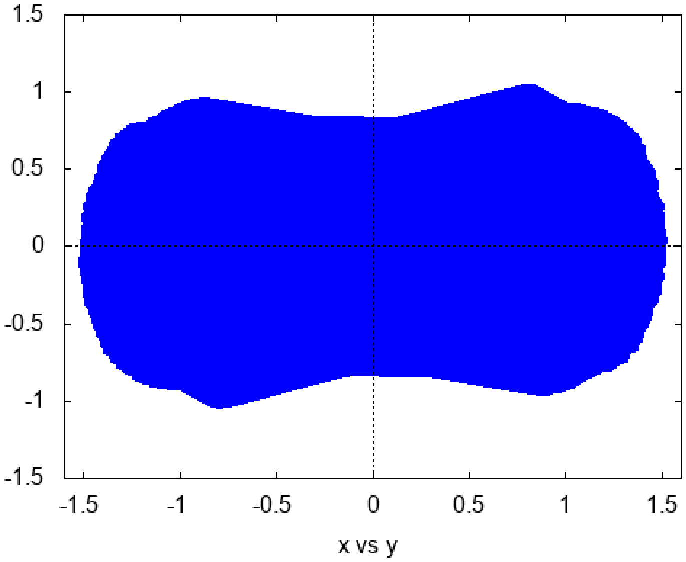
−>

```
/* Example 12.5.2 */ kill(x)$; kill(t)$; delta: 0.25$ stepSize: 2*float(%pi)/10.0$
w: 1$ F: 0.4$ dxdt: y$ dydt: F*cos(w*t) - delta*y + x - x**3$
rkRes : rk([dxdt, dydt], [x, y], [0, 0], [t, 0, 20000, stepSize] )$ txL
: makelist ([rkRes[i][1], rkRes[i][2]], i, 1, length(rkRes))$ tyL : makelist
([rkRes[i][1], rkRes[i][3]], i, 1, length(rkRes))$ wxplot2d([ [discrete,txL],
[discrete,tyL]],[t,0,280], [style,[lines,3]],[xlabel,”t”], [legend, ”x”, ”y”], [gnuplot_preamble,”set key bottom left;”], [gnuplot_term, aqua]); /* Phase */ xyL
: makelist ([rkRes[i][2], rkRes[i][3]], i, 1, length(rkRes))$ wxplot2d ( [discrete,
xyL],[x, -1.6, 1.6],[y, -1.5, 1.5], [style, [lines, 3]], [ylabel, ” ”],[xlabel, ” x vs y ”]
)$ /* Poincaré */ pL : makelist ([rkRes[i][2], rkRes[i][3]], i, 1, length(rkRes),
floor(2*float(%pi)/stepSize))$ wxplot2d ( [discrete, pL],[x, -1.6, 1.6],[y, -1.5,
1.5], [style, [points, 0.5]], [ylabel, ” ”],[xlabel, ” x vs y ”] )$
```

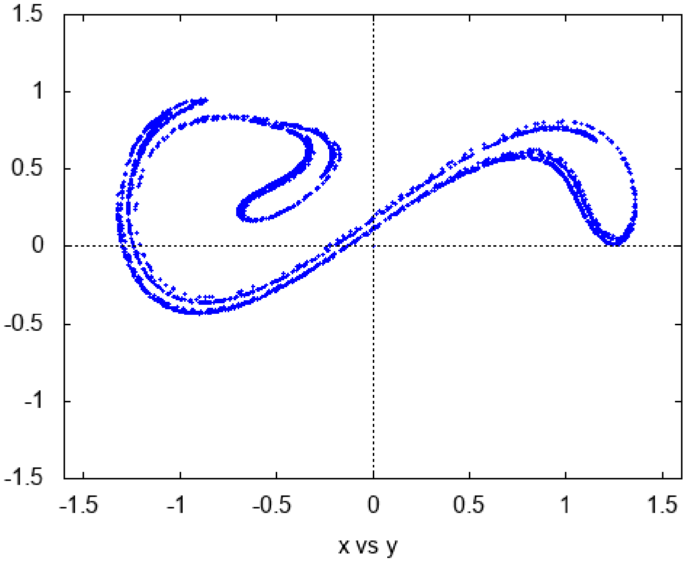


(%t731)

(%o731)


1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
x vs y

(%t733)


1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
x vs y

(%t735)