DiffEqSolutions

September 11, 2018

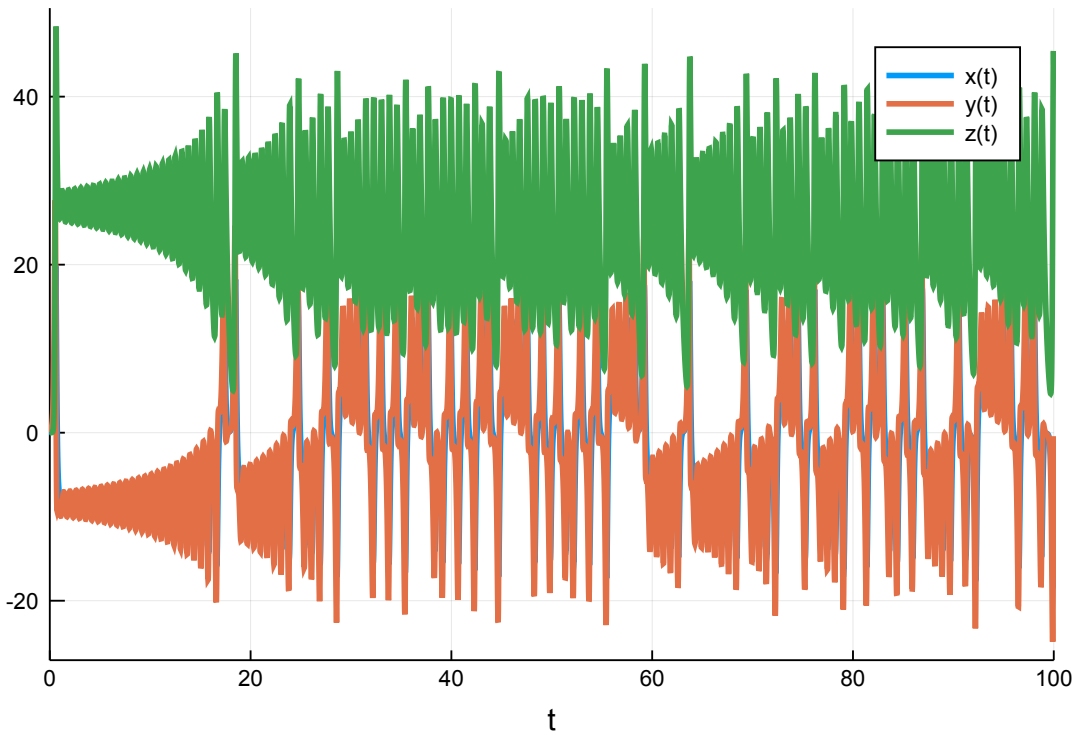
1 DiffEq Solutions

1.1 Solution to the Lorenz Problem

In [1]:
```
using DifferentialEquations

        f = @ode_def_nohes LorenzExample begin
          dx = *(y-x)
          dy = x*(-z) - y
          dz = x*y - *z
        end

        u0 = big.([0.1;0.0;0.0])
        tspan = (big(0.0),big(100.0))
        prob = ODEProblem(f,u0,tspan,(10.0,28.0,2.6666))
        sol = solve(prob);
```

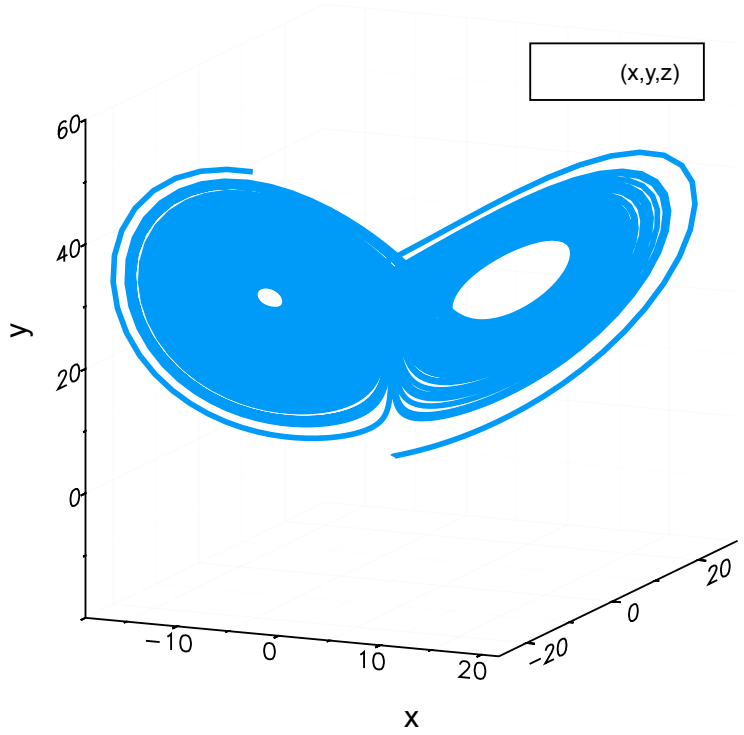
In [2]:
```
using Plots; gr(); plot(sol)
```

Out[2]:



In [3]: `plot(sol,vars=(:x,:y,:z))`

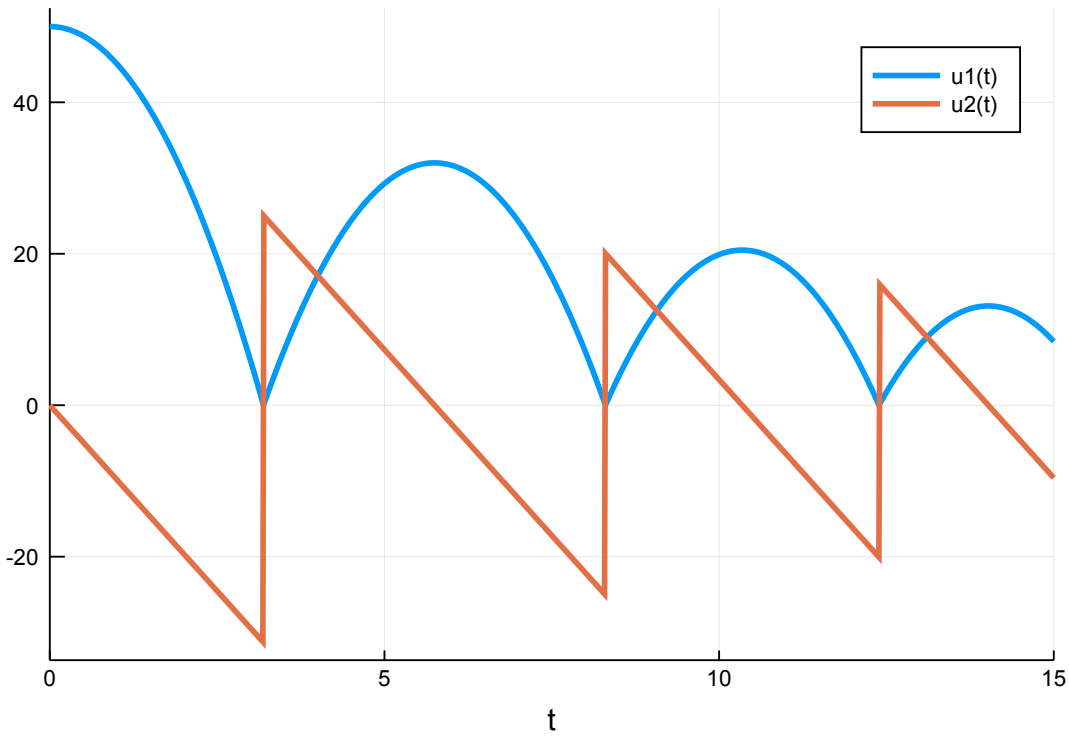
Out[3]:



1.2 Solution to the Ball Bounce Problem

In [5]:
```
f = function (du,u,p,t)
  du[1] = u[2]
  du[2] = -9.81
end

condtion = function (u,t,integrator) # Event when event_f(t,u,k) == 0
  u[1]
end

affect! = nothing
affect_neg! = function (integrator)
  integrator.u[2] = -0.8integrator.u[2]
end

callback = ContinuousCallback(condtion,affect!,affect_neg!,interp_points=100)

u0 = [50.0,0.0]
tspan = (0.0,15.0)
prob = ODEProblem(f,u0,tspan)


sol = solve(prob,Tsit5(),callback=callback,adaptive=false,dt=1/4)
plot(sol)
```

Out[5]:


u1(t)
u2(t)
40
20
0
-20
0
5
10
15
t