4. Forward Euler on Lorenz equations

function ode = forward\_euler(xyz, h, N)

%xyz holds the initial x y z values

%h is the time step size

%N is the number of time steps to take

x = []; y = []; z = []; t = [];

%fill in dynamic arrays with initial conditions

x(1) = xyz(1);

y(1) = xyz(2);

z(1) = xyz(3);

t(1) = 0;

for i = 2:N

t(i) = t(i-1) + h; %keep track of actual times

%and update associated position values for "time t"

x(i) = x(i-1) + h\*(16\*(y(i-1) - x(i-1)));

y(i) = y(i-1) + h\*(45\*x(i-1) - y(i-1) -x(i-1)\*z(i-1));

z(i) = z(i-1) + h\*(x(i-1)\*y(i-1) - 4\*z(i-1));

end

%plot(t,x)

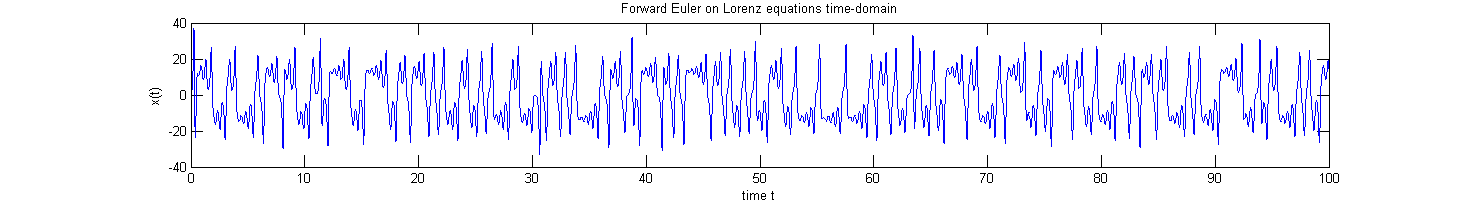
%title('Forward Euler on Lorenz equations time-domain');

plot3(x, y, z)

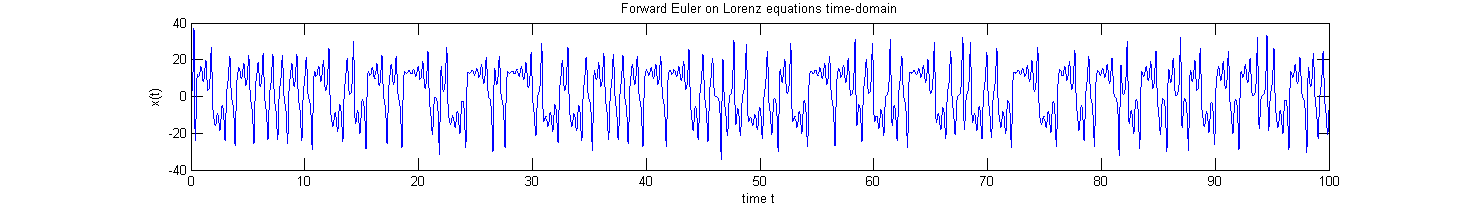
title('Forward Euler on Lorenz equations State-Space');

ode = ('check the plot');

end



x(t) vs t Time-domain plots for [1, 1 , 1] and [1.01, 1.01, 1.01]

Note the obvious differences in position x at time t between the two; chaotic

and x(t) vs y(t) vs z(t) State-space 3D plots

