No.

Date

ME564 L18

Forward Euler : $\dot{y} = f(y)$

$$y_{k+1} = y_k + \Delta t\, f(y_k)$$

$$y_{k+1} \approx \underbrace{y(t_k + \Delta\tau)}_{\text{true solution}} \quad (\text{assume } y_k = y(t_k))$$

$$= y(t_k) + \dot{y}(t_k)\Delta t + \frac{1}{2!}\ddot{y}(t_k)\Delta t^2 + \cdots$$

$$= \underbrace{y_k + f(y_k)\Delta t}_{\text{forward difference}} + \underbrace{\mathcal{O}(\Delta t^2)}_{\text{error}}$$

y_{k+1}

forward, backward euler → too much error

↓

use Runge-Kutta Integration

use this for a full step of euler

get this vector field

half step of euler

e.g. fluid flow

$\dot{y} = f(t, y)$

Runge-Kutta 2nd Order ('ode23')

$$y_{k+1} = y_k + \Delta t\, f\left(t_k + \frac{\Delta\tau}{2},\ y_k + \frac{\Delta\tau}{2} f(t_k, y_k)\right) + \mathcal{O}(\Delta\tau^3)$$

f_1

f_2

Global $\mathcal{O}(\Delta t^2)$

Runge-Kutta 4th order ('ode45')

tuned to cancel the terms @ taylor expansion

$$y_{k+1} = y_k + \frac{\Delta\tau}{6}\left[f_1 + 2f_2 + 2f_3 + f_4\right] + \mathcal{O}(\Delta\tau^5)$$

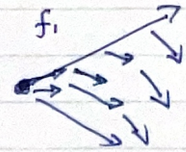
Global $\mathcal{O}(\Delta\tau^4)$

$f_1 = f(t_k, y_k)$ what is the vector field

$f_2 = f\left(t_k + \frac{\Delta\tau}{2},\ y_k + \frac{\Delta\tau}{2} f_1\right)$

$f_3 = f\left(t_k + \frac{\Delta\tau}{2},\ y_k + \frac{\Delta\tau}{2} f_2\right)$

$f_4 = f\left(t_k + \Delta t,\ y_k + \Delta\tau f_3\right)$

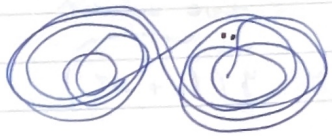


Lorentz

chaos

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$

$\sigma = 10$, $\beta = 8/3$, $\rho = 28$ } classic chaotic



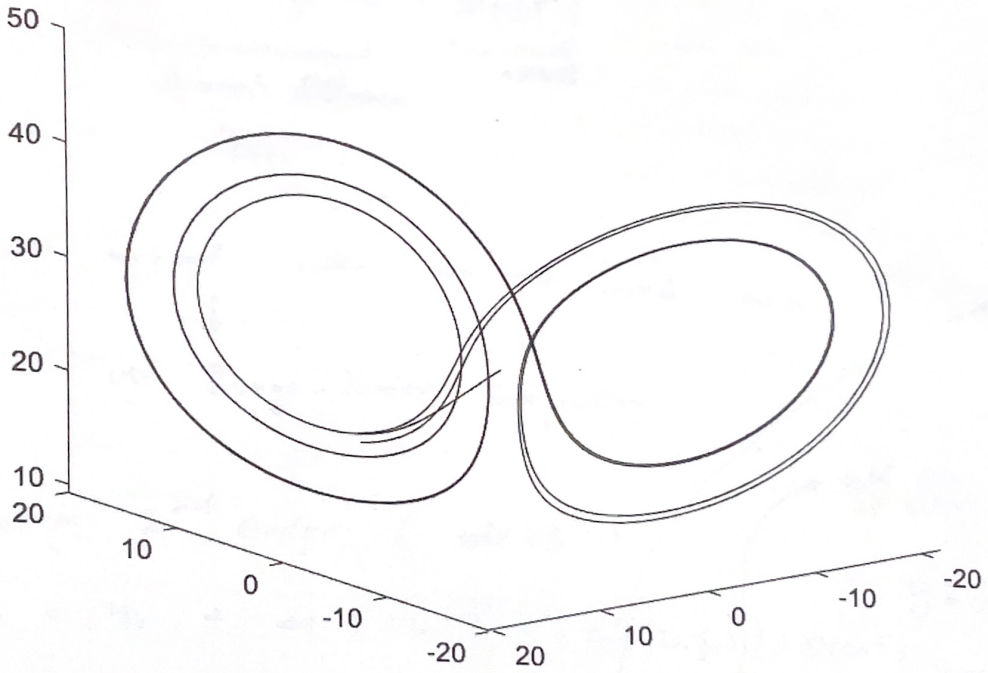
```
close all; clear all; clc;

% Lorenz's parameters (chaotic)
sigma = 10;
beta = 8/3;
rho = 28;

% Initial condition
y0=[-8; 8; 27];

% Compute trajectory
dt =0.01;
tspan=[0:dt:4];

Y(:,1)=y0;
yin = y0;
for i=1:length(tspan)
    time = i*dt;
    yout = rk4singlestep(@(t,y)lorenz(t,y,sigma,beta,rho),dt,time,yin);
    Y = [Y yout];
    yin = yout;
end
plot3(Y(1,:),Y(2,:),Y(3,:),'b')
hold on
[t,y] = ode45(@(t,y)lorenz(t,y,sigma,beta,rho),tspan,y0);
plot3(y(:,1),y(:,2),y(:,3),'r')


function yout = rk4singlestep(fun,dt,t0,y0)

f1 = fun(t0,y0);
% size(f1)
% size(y0)
f2 = fun(t0+dt/2,y0+(dt/2)*f1);
f3 = fun(t0+dt/2,y0+(dt/2)*f2);
f4 = fun(t0+dt,y0+dt*f3);

yout = y0 + (dt/6)*(f1+2*f2+2*f3+f4);
end
function dy = lorenz(t,y,sigma,beta,rho)
% y is a three dimensional state-vector
dy = [
sigma*(y(2)-y(1));
y(1)*(rho-y(3))-y(2);
y(1)*y(2)-beta*y(3);


];
end
```

function handles :

@(t,y) lorenz(t,y,1,1,1)

myquadratic = @(x) x.^2

myquadratic(2) = 4


50
40
30
20
10
20
10
0
-10
-20
20
10
0
-10
-20