## Points following Lorenz differential equations

MATLAB CODE

Following code has been used to generate points following Lorenz function

Parameters rho, sigma, beta and initial point is configurable and can be changed to generate different set of point.

|  |
| --- |
| clear all; clc; [x, y, z] = lorenz(28, 10, 8/3, [10.01, 20.01, 30.01], [0 25], 0.000001); res = [x(:), y(:), z(:)]; writematrix(res, 'M3.csv'); **function** [x,y,z] = **lorenz**(rho, sigma, beta, initV, T, eps) % LORENZ Function generates the lorenz attractor of the prescribed values % of parameters rho, sigma, beta % % [X,Y,Z] = LORENZ(RHO,SIGMA,BETA,INITV,T,EPS) % X, Y, Z - output vectors of the strange attactor trajectories % RHO - Rayleigh number % SIGMA - Prandtl number % BETA - parameter % INITV - initial point % T - time interval % EPS - ode solver precision % % Example. % [X Y Z] = lorenz(28, 10, 8/3); % plot3(X,Y,Z); **if** nargin<3  error('MATLAB:lorenz:NotEnoughInputs','Not enough input arguments.'); **end** **if** nargin<4  eps = 0.000001;  T = [0 25];  initV = [0 1 1.05]; **end** options = odeset('RelTol',eps,'AbsTol',[eps eps eps/10]); [T,X] = ode45(@(T,X) F(T, X, sigma, rho, beta), T, initV, options); plot3(X(:,1),X(:,2),X(:,3)); axis equal; grid; title('Lorenz attractor'); xlabel('X'); ylabel('Y'); zlabel('Z'); x = X(:,1); y = X(:,2); z = X(:,3); **return** **end** **function** **dx** = **F**(T, X, sigma, rho, beta) % Evaluates the right hand side of the Lorenz system % x' = sigma\*(y-x) % y' = x\*(rho - z) - y % z' = x\*y - beta\*z % typical values: rho = 28; sigma = 10; beta = 8/3;  dx = zeros(3,1);  dx(1) = sigma\*(X(2) - X(1));  dx(2) = X(1)\*(rho - X(3)) - X(2);  dx(3) = X(1)\*X(2) - beta\*X(3);  **return** **end** |