% No. of points

N = 5;

% Governing equation => d = G\*m

% Defining x

x = linspace(5,25,N);

% Defining y

dT = randi(10, [1, 5]) %y transpose = dT

d = dT.' % y = d

% Calculating x^0, x^1, x^2, x^3, x^4

x0 = [1, 1, 1, 1, 1]

x1 = x

x2 = x.\*x

x3 = x.\*x2

x4 = x.\*x3

% Defining G

GT = [x0; x1; x2; x3; x4] % Transpose of G = GT

G = GT.' % The required matrix G for the governing equation d = G \* m

% Defining d

dT = randi(10, [1, 5]) % Transpose of d = dT

d = dT.' % The required matrix d for the governing equation d = G \* m

% We have calculated d and G in the govering eqn. d = G\*m, now we want to

% calculate m, which is given by m = Inv(G) \* d. Hence, we need to

% calculate inv(G) first, as shown below

% Calculating Inverse of G

GI = inv(G) % GI = inverse of G

% Calculating m

m = GI \* d

% Plotting 1000 points on the function obtained

xthousand = linspace(5, 25, 1000)

ythousand = linspace (0, 1000)

for i = 1:1000

ythousand(i) = Result(xthousand(i),m)

end

plot(xthousand, ythousand)

% Defining function A = f(b)

function A = Result(b,m)

A = (b.^0).\*(m(1)) + (b.^1).\*(m(2)) + (b.^2).\*(m(3)) + (b.^3).\*(m(4)) + (b.^4).\*(m(5))

end