

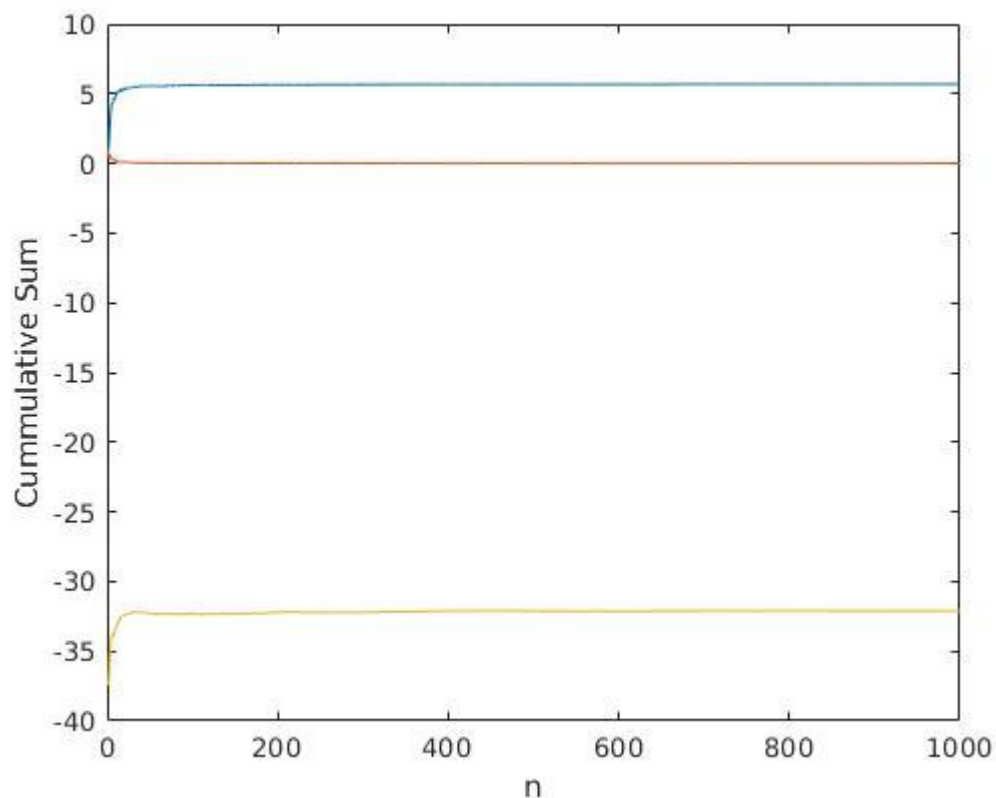
ME627 Assignment 7  
Name : Sudhanshu Mishra  
Roll no: 17807726

1.)

Params:

Sigma=10  
Beta =8/3  
Rho =28

The figure below shows the graph for the lyapunov exponents



Lyapunov exponent 1 = 5.6888  
Lyapunov exponent 2 = -0.0033  
Lyapunov exponent 3 = -32.1794

Discussion:

According to theory, as the Lorenz system is an autonomous system, one of the Lyapunov exponents must be 0. We can see that the 2nd Lyapunov exponent in our numerical calculation is very close to 0. Furthermore, we already know that the Lorenz system is a chaotic system, the positive Lyapunov exponent confirms that information. We see here that

the sum of the Lyapunov exponents at all times is a negative value, thus this shows that the system is a dissipative system.

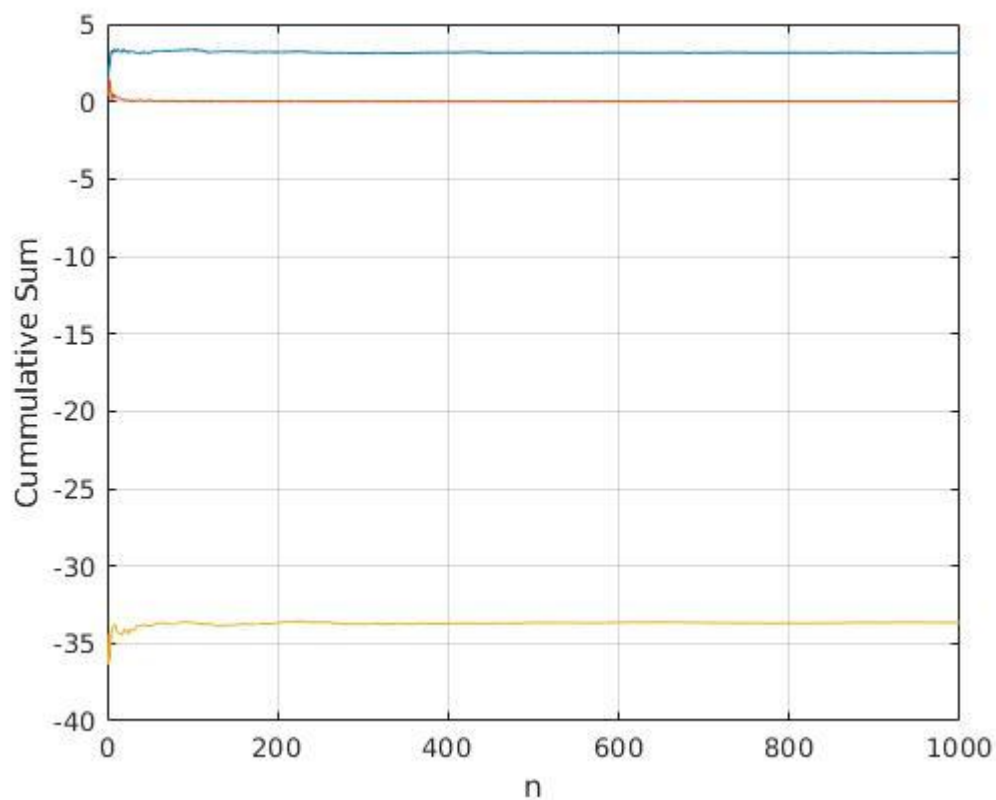
Params

$\sigma=10$

$\beta=1$

$\rho=28$

The figure below shows the graph for the lyapunov exponents



Lyapunov exponent 1 = 3.1507

Lyapunov exponent 2 = 0.0035

Lyapunov exponent 3 = -33.6459

Discussion:

According to theory, as the Lorenz system is an autonomous system, one of the Lyapunov exponents must be 0. The time period for integration is chosen as  $t=0$  to  $t=2\pi$ . We can see that the 2nd Lyapunov exponent in our numerical calculation is very close to 0. Furthermore,

we already know that the Lorenz system is a chaotic system, the positive Lyapunov exponent confirms that information. We see here that the sum of the Lyapunov exponents at all times is a negative value, thus this shows that the system is a dissipative system.

%MATLAB Code

%Main script to calculate LYPExpos and plot graphs

```
clc;
```

```
clear all;
```

```
close all;
```

```
z=ALE_LS(1000);
```

```
z=abs(z); z=log(z); n=[1:1000];
```

```
figure(1)
```

```
plot(n,cumsum(z(1,:))./n,n,cumsum(z(2,:))./n,n,cumsum(z(3,:))./n);
```

```
grid on
```

```
xlabel('n')
```

```
ylabel('Cummulative Sum')
```

```
figure(2)
```

```
plot(n,sum(z,1));
```

%Function to calculate all LYP Expos

```
function z=ALE_LS(N)
```

```
op=odeset('reltol',1e-8,'abstol',1e-8);
```

```
y0=randn(12,1);
```

```
A=[y0(4:6),y0(7:9),y0(10:12)];
```

```
[q,r]=qr(A);
```

```
y0(4:6)=q(:,1);
```

```
y0(7:9)=q(:,2);
```

```
y0(10:12)=q(:,3);
```

```
z=zeros(3,N);
```

```
for k=1:N
```

```
    [t,y]=ode45('LS2',[0,2*pi],y0,op);
```

```
    y0=y(end,:);
```

```
    A=[y0(4:6),y0(7:9),y0(10:12)];
```

```
    [q,r]=qr(A);
```

```
    y0(4:6)=q(:,1);
```

```
    y0(7:9)=q(:,2);
```

```
    y0(10:12)=q(:,3);
```

```
    z(:,k)=diag(r);
```

```
end
```

%Function for Lorenz System

```
function qdot=LS2(t,q)
```

```
sigma=10;  
beta=1;  
rho=28;
```

```
x=q(1);  
y=q(2);  
z=q(3);
```

```
xdot=sigma*(y-x);  
ydot=x*(rho-z)-y;  
zdot=x*y-beta*z;
```

```
qdot=[xdot;ydot;zdot];
```

```
D = [-sigma, sigma, 0; rho-z,-1,-x;y,x,-beta];  
xi1 = q(4:6);  
xi2 = q(7:9);  
xi3 = q(10:12);
```

```
qdot = [qdot;D*xi1;D*xi2;D*xi3];
```

---

2.

Params

N=5000 % No. of points Taken

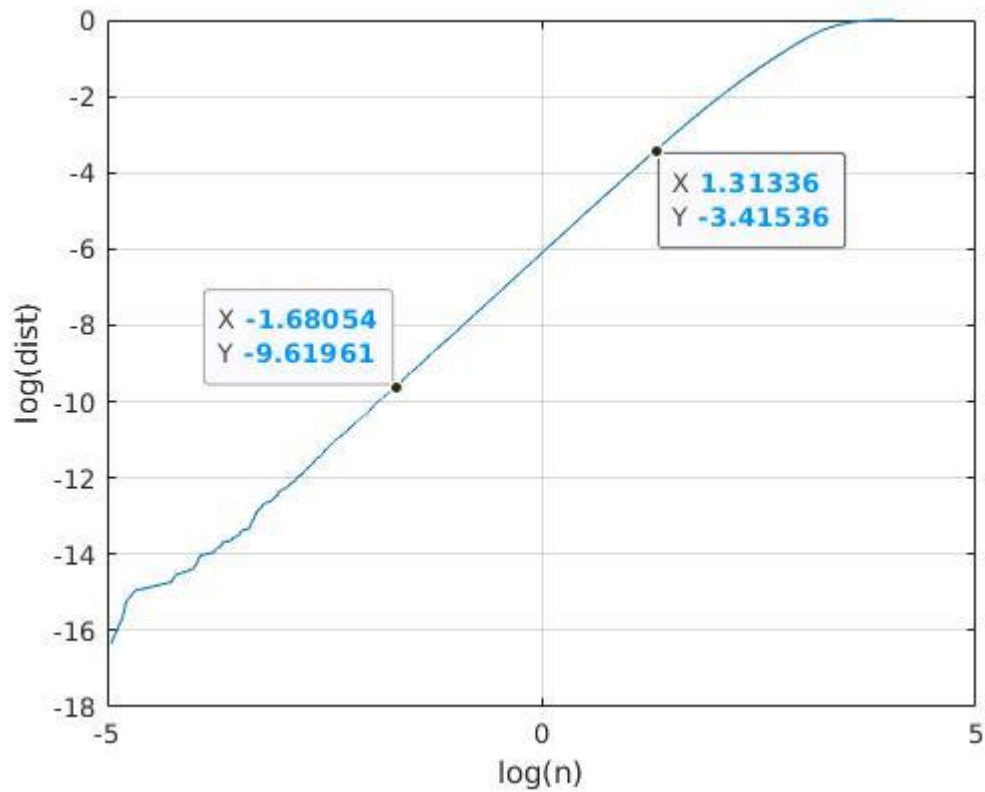
Sigma=10

Beta =1

Rho =28

A = 12.5510

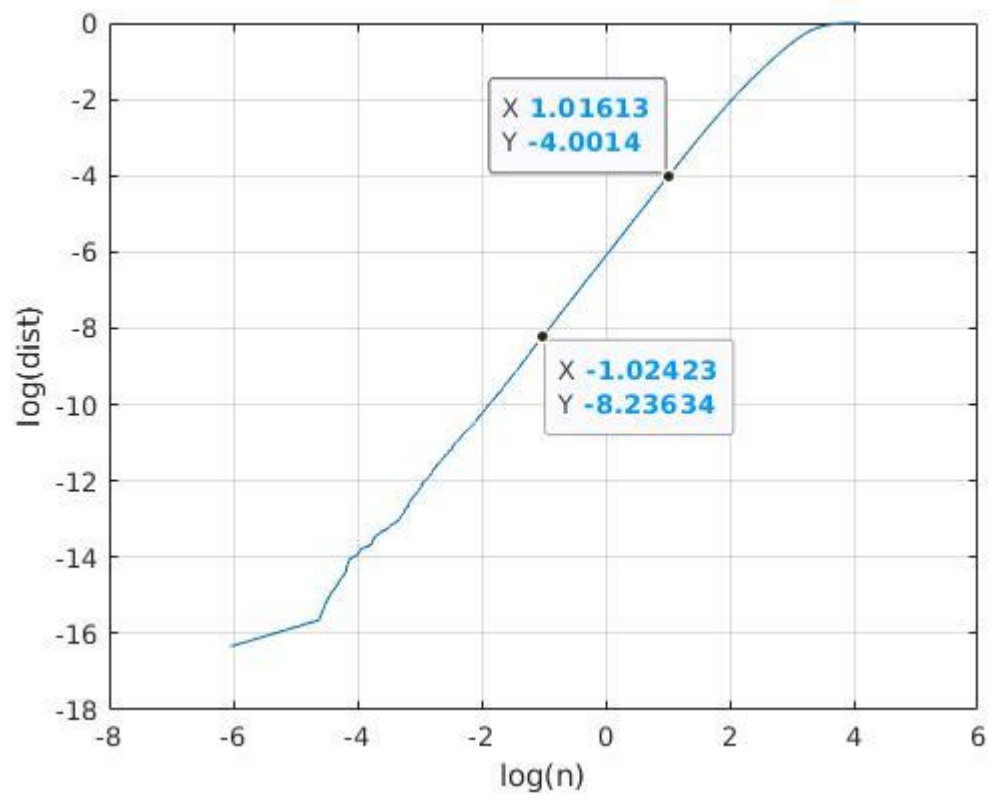
The figure below shows the graph for log Distances vs log n. The slope in the scaling region gives the correlation dimension.



$$\text{Slope} = (-3.3415 + 9.6196) / (1.3133 + 1.6805) = 2.097$$

A = 19.5929

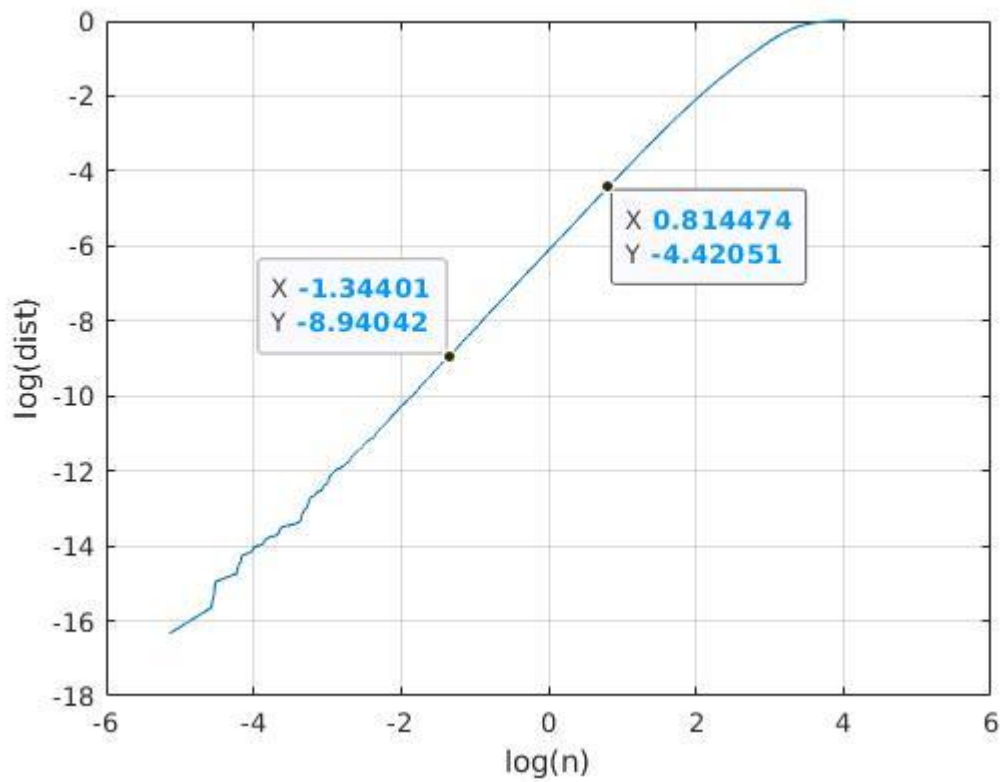
The figure below shows the graph for log Distances vs log n. The slope in the scaling region gives the correlation dimension.



$$\text{Slope} = (-4.0014 + 8.23634) / (1.0161 + 1.0242) = 2.075$$

$$A = 12.5751$$

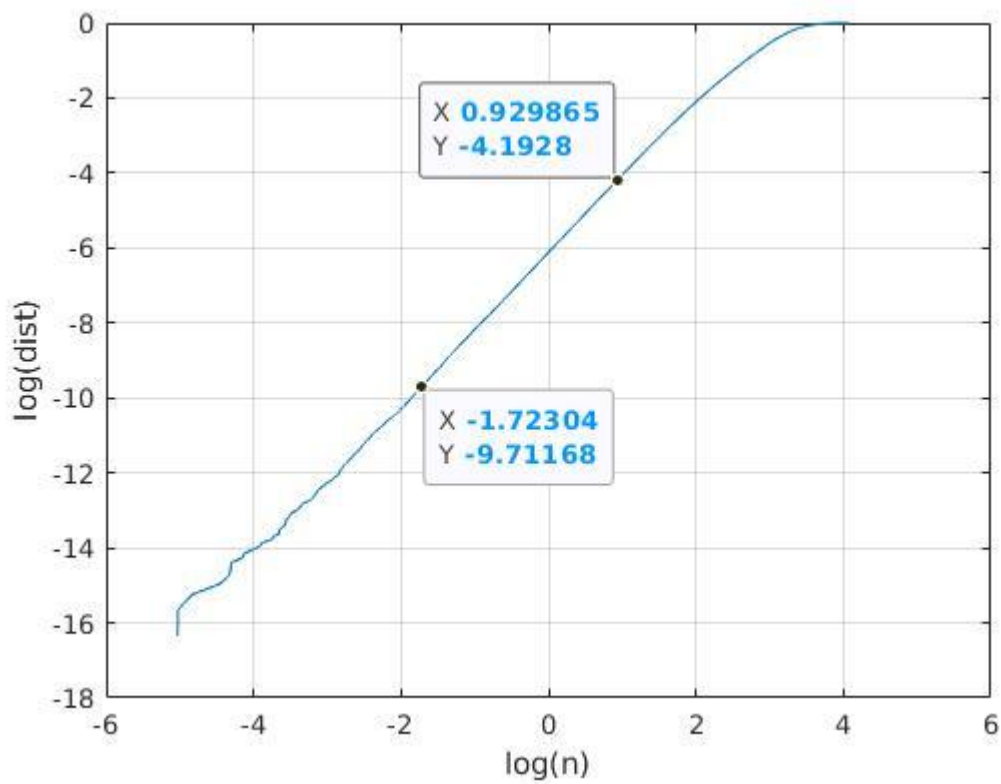
The figure below shows the graph for log Distances vs log n. The slope in the scaling region gives the correlation dimension.



$$\text{Slope} = (-4.4205 + 8.9404) / (0.8144 + 1.3440) = 2.094$$

$$A = 12.4352$$

The figure below shows the graph for log Distances vs log n. The slope in the scaling region gives the correlation dimension.

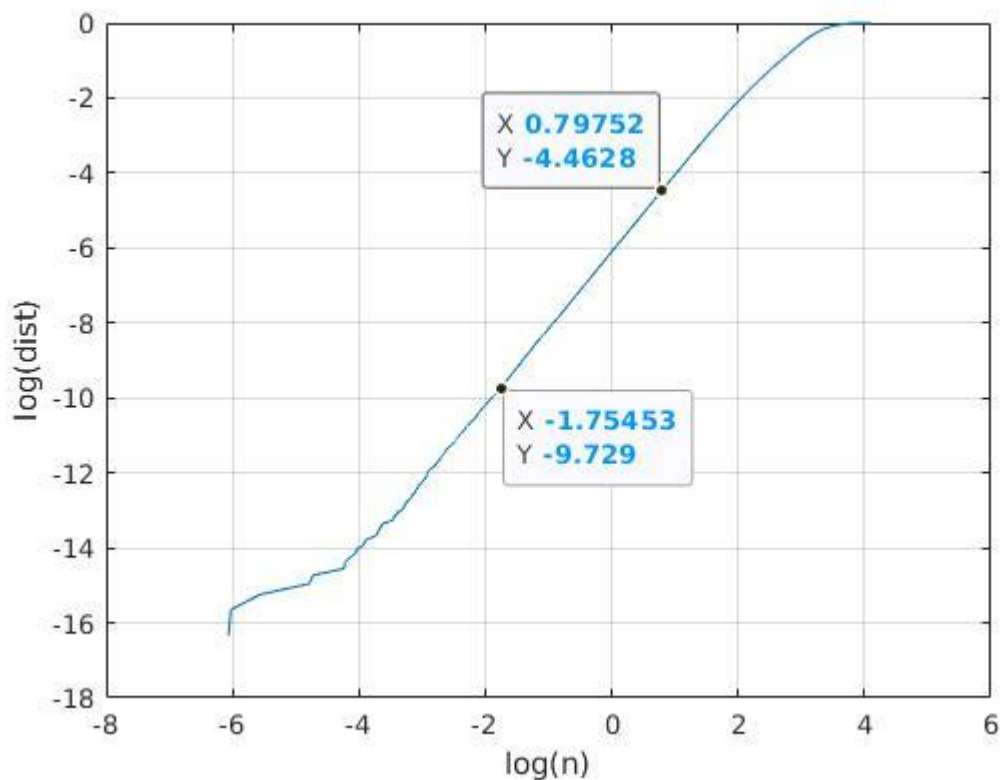


Slope = 2.08

A=15.9642

The figure below shows the graph for log Distances vs log n. The slope in the scaling region gives the correlation dimension.





Slope=2.063

Discussion:

First a random value of  $A$  is chosen between 10 and 20.

Then the Lorenz equations are integrated from  $t=0$  to  $t=A$  using ode45 using the same initial condition as the last state reached in the previous integration. This is done 100 times to remove any transients available. After this the Lorenz equations are integrated 5000 times, each time from  $t=0$  to  $t=A$ ; using the last state in the previous integration as the starting state, storing the last state at the end of each integration. Thus we are now left with 5000 data points. After this the distances between all pairs of points is calculated and stored, these distances are then sorted. The log of the distances are plotted against the log of the rank of the distance/ the total number of distances. From the generated plot, the slope is calculated using two points from a favourable scaling region.

The slope gives us the Correlation dimension.

From the experiments we get the following data :

A	CorrDim
12.5510	2.097
19.5929	2.075
12.5751	2.094
12.4352	2.08
15.9642	2.063

---

Params

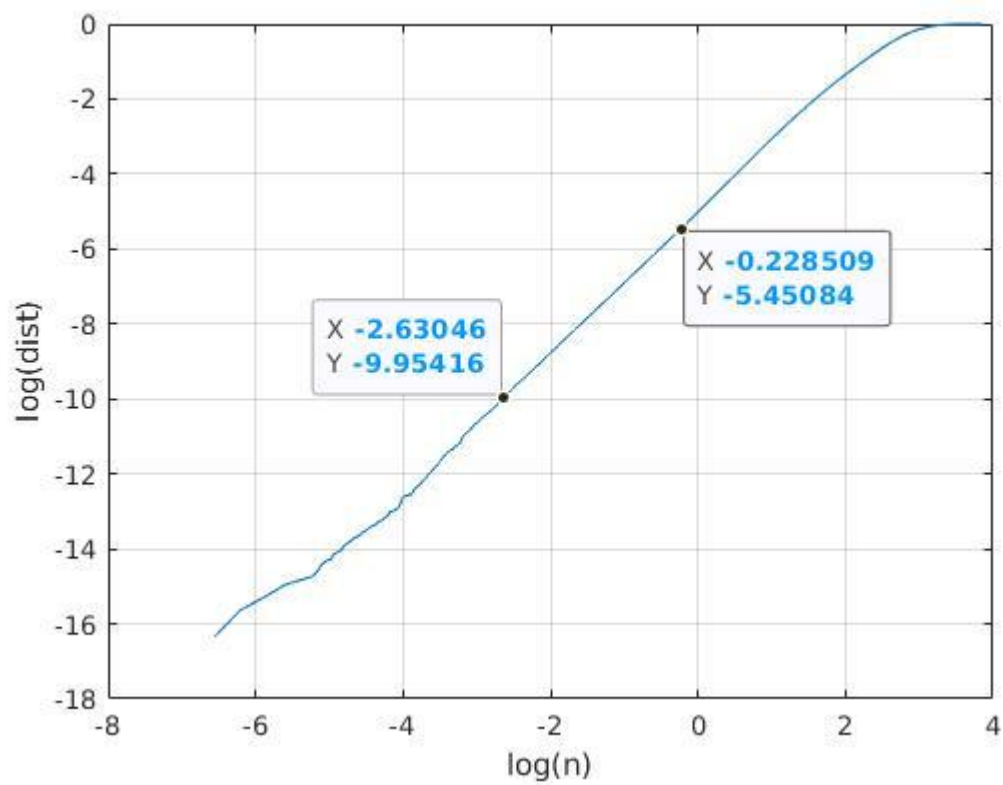
N=5000 % No. of points Taken

Sigma=10

Beta =1

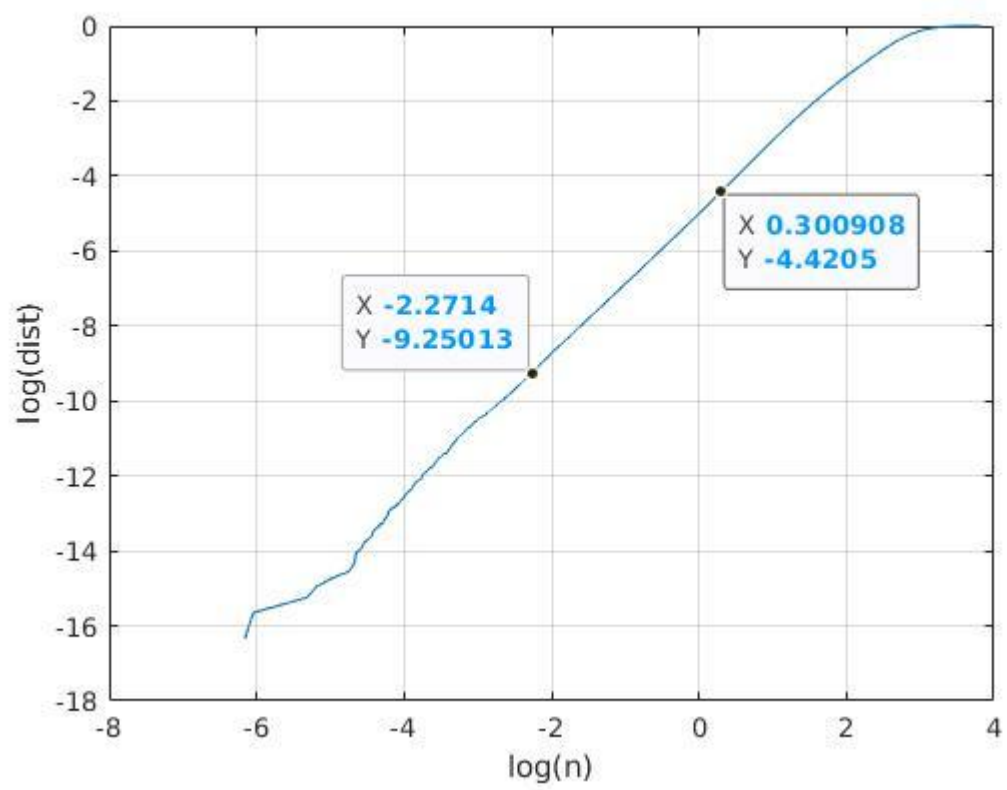
Rho =28

A=12.1510



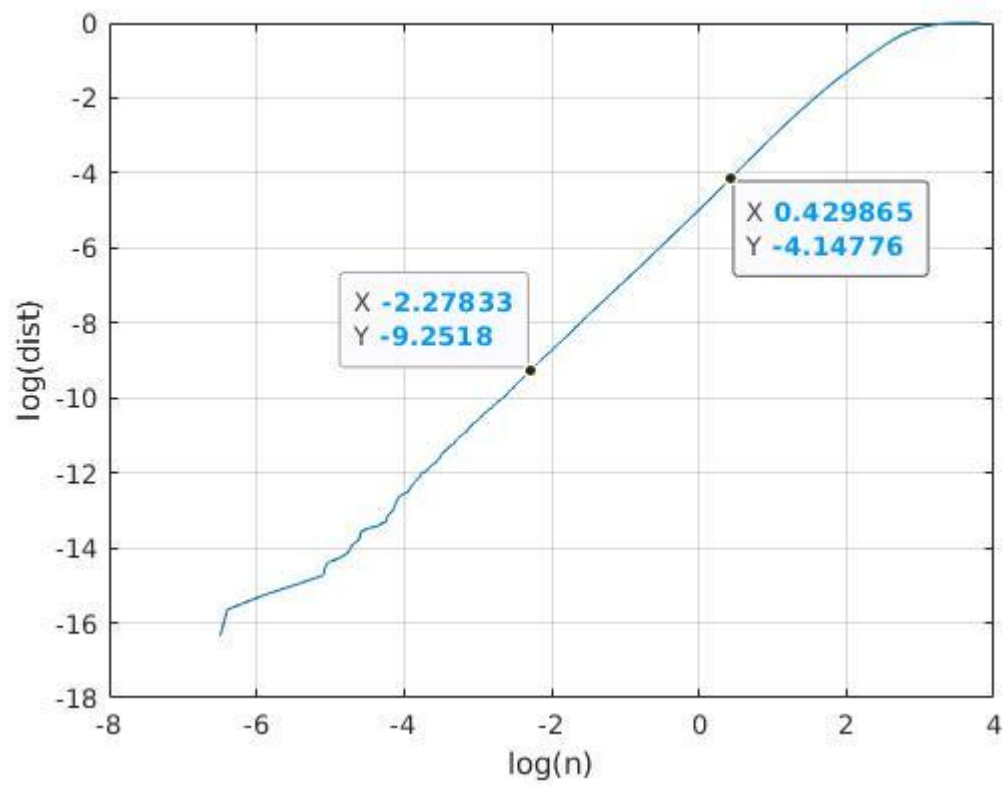
Slope = 1.8748

A = 14.9270



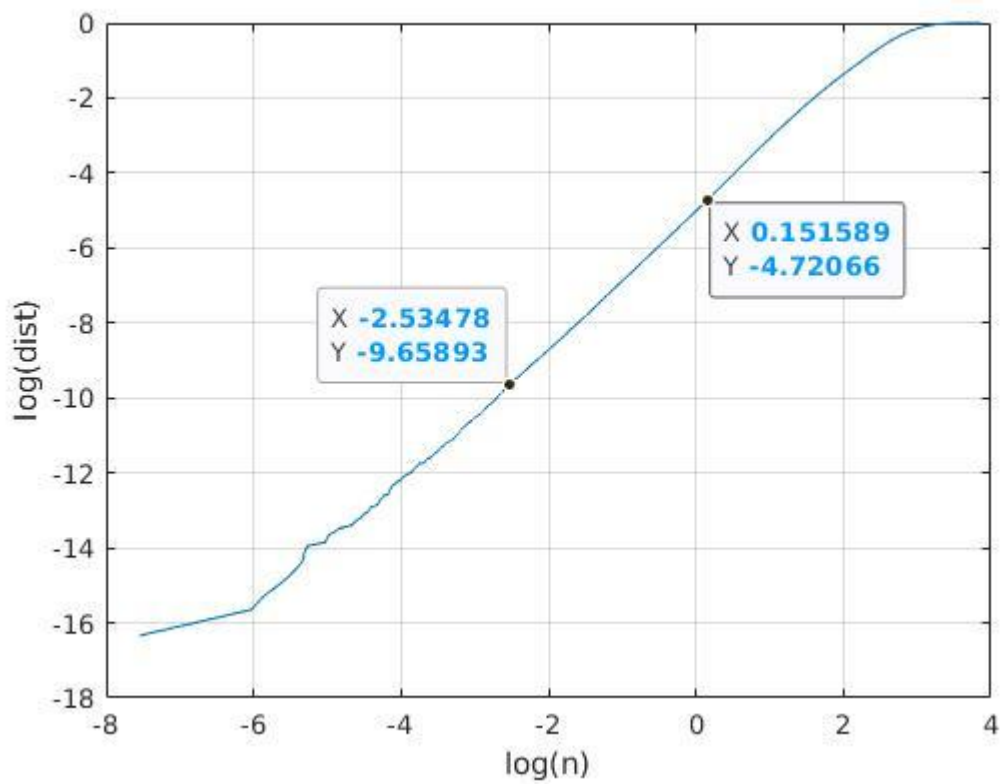
Slope = 1.8775

A=16.1236



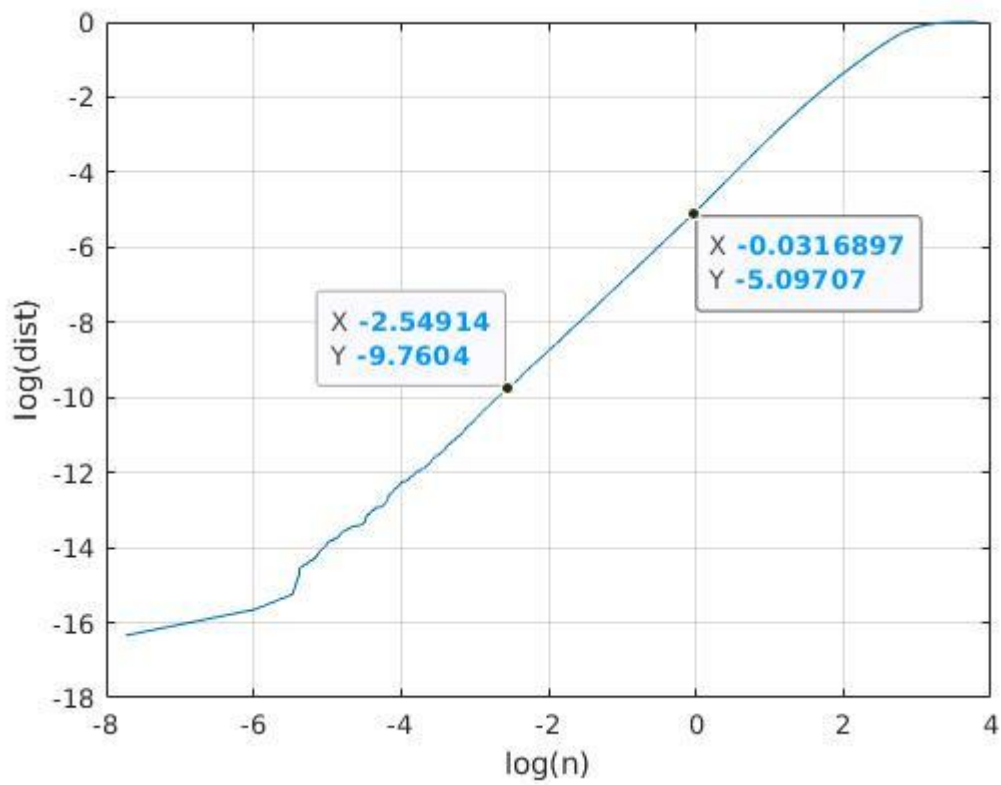
Slope = 1.8847

A=18.3137



Slope = 1.8383

A = 19.2587



Slope = 1.8523

Discussion :

First a random value of A is chosen between 10 and 20.

Then the Lorenz equations are integrated from  $t=0$  to  $t=A$  using ode45 using the same initial condition as the last state reached in the previous integration. This is done 100 times to remove any transients available. After this the Lorenz equations are integrated 5000 times, each time from  $t=0$  to  $t=A$ ; using the last state in the previous integration as the starting state, storing the last state at the end of each integration. Thus we are now left with 5000 data points. After this the distances between all pairs of points is calculated and stored, these distances are then sorted. The log of the distances are plotted against the log of the rank of the distance/ the total number of distances. From the generated plot, the slope is calculated using two points from a favourable scaling region.

The slope gives us the Correlation dimension.

From the experiments we get the following data :

A	CorrDim
12.1510	1.8748
14.9270	1.8775
16.1236	1.8847
18.3137	1.8383
19.2587	1.8523

---

MATLAB Code

```
%%Code to get data for dimension calculation
```

```
clc;  
clear all;  
close all;
```

```
A = 19.2587;%10*(1+rand(1));  
save Aused A  
q0 = randn(3,1);  
N=5000;  
X = zeros(3,N);
```

```
op=odeset('reltol',1e-8,'abstol',1e-8);
```

```
%Eliminating Transients
```

```
for k=1:100
```

```

[t,q]=ode45('LS',[0,A],q0,op);
q0=q(end,:);
end

```

```

for k=1:N
    k
    [t,q]=ode45('LS',[0,A],q0,op);
    q0=q(end,:);
    X(:,k)=q0;
end
save corrttest X

```

%%Function for Lorenz equations

```

function qdot=LS(t,q)
sigma=10;
beta=1;
rho=28;

```

```

x=q(1);
y=q(2);
z=q(3);

```

```

xdot=sigma*(y-x);
ydot=x*(rho-z)-y;
zdot=x*y-beta*z;

```

```

qdot=[xdot;ydot;zdot];

```

%% Code to calculate distances

```

function E=corrddim(X)

```

```

q=X;

```

```

[m,n]=size(q);
E=[];

```

```

while m>1
    t=q(1,:);
    q=q(2:end,:);
    [m,n]=size(q);
    EE=zeros(m,1);
    for n=1:m
        EE(n)=norm(q(n,:)-t);
    end
    m
    %size(EE)

```

```
    E=[E;EE];  
end  
end
```

```
%% Code to calculate correlation dimension
```

```
E = corrdim(X(:,1:5000));  
E=sort(E);  
ne=size(E,1);  
n=[1:ne]./ne;  
plot(log(E),log(n));  
xlabel('log(n)')  
ylabel('log(dist)')  
grid on  
%x=ginput(2)  
%p=polyfit(x(:,1),x(:,2),1)
```

---