

Chaos Project

Elizabeth Kenney

July 29, 2002

Abstract

For this project, the Lorentz Model was studied. The equilibrium states and the stability of these equilibrium states were examined. Numerical experiments were performed and a periodic solution and a chaotic solution were found. For both solutions, various statistical quantities were evaluated. Next white noise was added to both the periodic and chaotic solutions. Again, the same statistical quantities were evaluated. It was observed that when adding white noise to the periodic solution, the response statistics were greatly affected, which is quite evident in the time history plots. When adding white noise to the chaotic solution, there isn't a huge difference in the response statistics. The Lorentz Model was also expanded using the Karhunen-Loeve expansion. This illustrated that there was a high amount of energy surrounding the mode corresponding to one of the constants in the Lorentz Model. This energy played a role in the reconstruction of the x direction, while it did not significantly affect the reconstruction in the y and z directions.

1 Equilibrium States and the Stability of the Equilibrium States

The following equations are the nonlinear system known as the Lorentz model.

$$\frac{dx}{dt} = \sigma(y - x) \quad (1)$$

$$\frac{dy}{dt} = rx - y - xz \quad (2)$$

$$\frac{dz}{dt} = -b + xy \quad (3)$$

In these equations r , b , and σ are constants. To begin, the critical or equilibrium points were determined. To do this, $\frac{dx}{dt}$, $\frac{dy}{dt}$, and $\frac{dz}{dt}$ were set to zero and the equations 1,2, and 3 were solved. From this, the following critical points were determined.

$$x = y = z = 0 \quad (4)$$

$$\text{for } r > 1 \quad x = \pm\sqrt{r-1} \quad (5)$$

$$y = \pm\sqrt{r-1} \quad (6)$$

$$z = r - 1 \quad (7)$$

To check the stability, the equations were linearized about the critical point $x=y=z=0$. By doing this, the characteristic determinate was found to be:

$$\begin{vmatrix} -\sigma - \lambda & \sigma \\ r & -\lambda - 1 \end{vmatrix} \quad (8)$$

From this determinate, the eigenvalues were determined to be:

$$\lambda_{1,2} = \frac{-\sigma}{2} - \frac{1}{2} \pm \frac{1}{2} \sqrt{\sigma^2 - 2\sigma + 1 + 4r\sigma} \quad (9)$$

For the system to be stable, all the eigenvalues must be negative. Therefore, for this to occur, the parameter r must be < 1 . For the system to be unstable, at least one eigenvalue must be positive, therefore $r > 1$ will make the solution unstable. When $r < 1$, $x=y=z=0$ is the only stable solution.

2 Numerical Experiments

To perform the numerical experiments, equations 1, 2, and 3 were linearized and equations 10, 11, and 12 below were used.

$$\frac{dx}{dt} = \sigma(y - x) \quad (10)$$

$$\frac{dy}{dt} = x - y \quad (11)$$

$$\frac{dz}{dt} = -bz \quad (12)$$

Using these equations, $x(t)$, $y(t)$, and $z(t)$ were found using the Adams Bashforth step scheme. By solving for these equations, various plots of stochastic parameters were created. These plots include a 3-D plot, time history plots, poincare plots, cross correlation plots, auto-correlation plots, and power spectral density plots. The constants in equations 1, 2, and 3 were selected to create cases where both periodic and chaotic solutions existed.

2.1 Periodic Solution

By selecting the following constants, a periodic solution was found

$$\sigma = 10 \quad r = 15 \quad b = \frac{8}{3} \quad (13)$$

Figure 1 is a phase plot of the system. From this plot, it can be seen that when the constants above and the initial condition of $(0,1,0)$ are both selected, the Lorentz system is periodic and it approaches the stable point of $(-6.11, -6.11, 15)$.

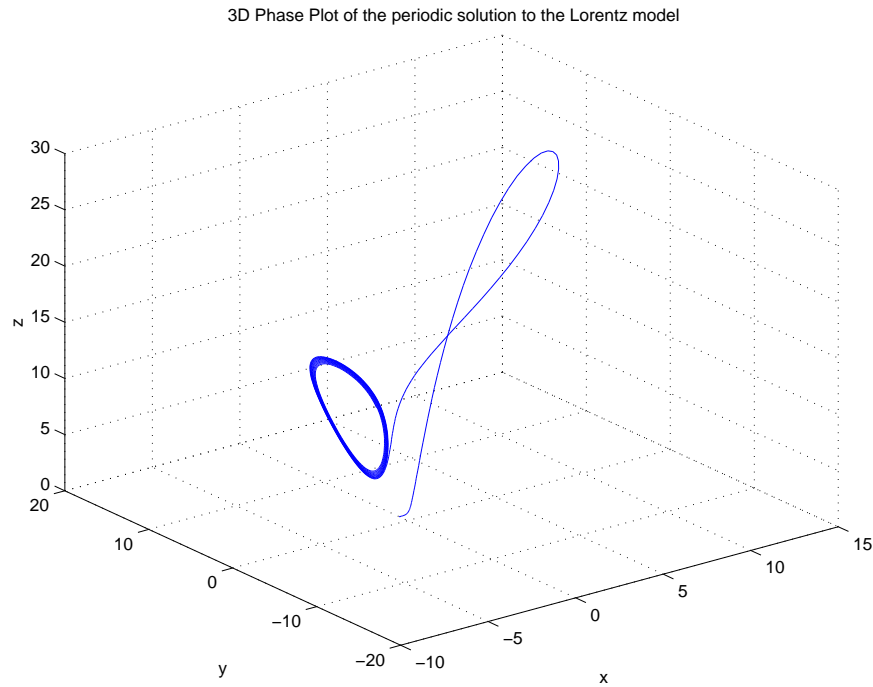


Figure 1: 3D Phase Plots of Periodic Solution to the Lorentz Model

Next, the time history responses were determined (Figure 2) as well as the Poincare plots (Figure 3).

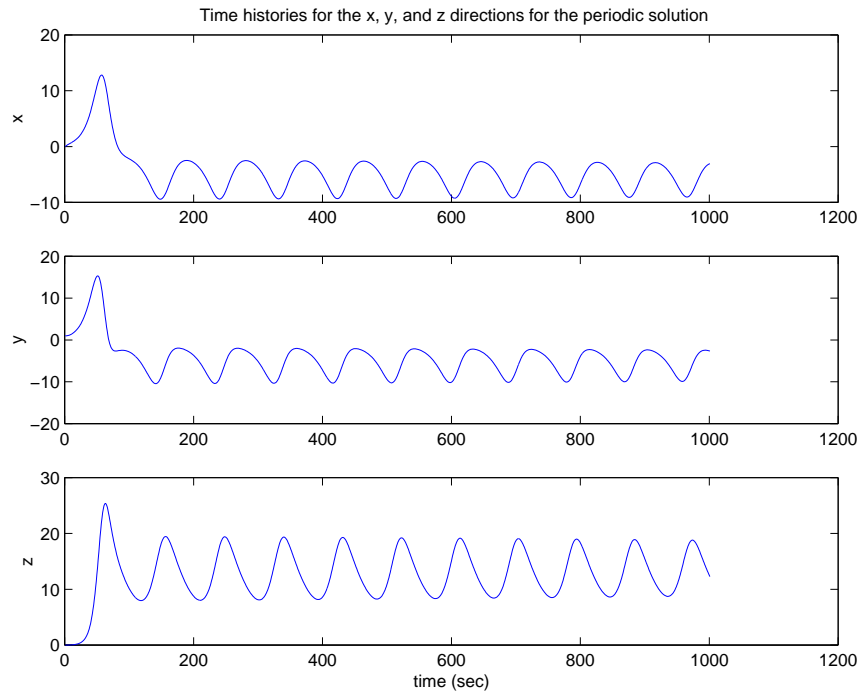


Figure 2: Time History Responses of Periodic Solution to the Lorentz Model

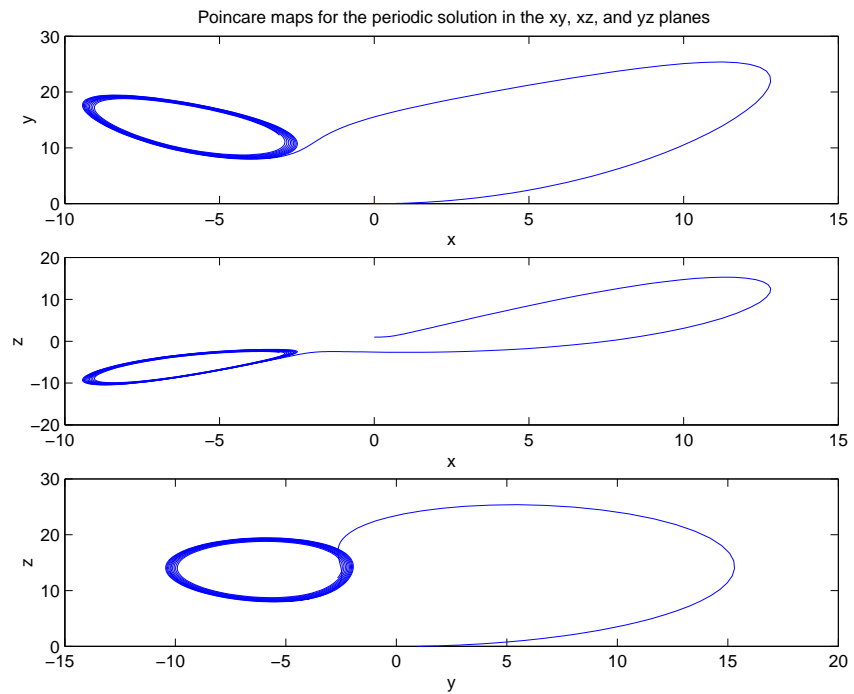


Figure 3: Poincare Plots of Periodic Solution to the Lorentz Model

The response statistics of the periodic solution were then analyzed. Table 1 contains the mean, mean square, and various moments for this system in each the x, y, and z directions.

	X	Y	Z
Mean	-4.5164	-4.5466	12.7164
Mean Square	35.2522	38.7074	183.2012
Moment (n=6)	4.9164e4	8.8002e4	1.489e5
Moment (n=3)	0	0	0
Moment (n=20)	3.4243e20	2.3845e21	1.3775e22

Table 1: Response Statistics for Periodic Solution to the Lorentz System

The next statistics that were investigated were the cross correlation and the autocorrelation. The cross correlation is a measure of the similarities or shared properties between two signals. In this case, the x-y correlation, y-z correlation, and x-z correlation were evaluated and are illustrated in Figure 4. This figure shows that there is a strong positive correlation between the x and y signals with the maximum value at a lag time of 1000 seconds. Conversely, there is a strong negative correlation between both the x and z signals and the y and z signals with the most negative point at a lag time of 1000 seconds.

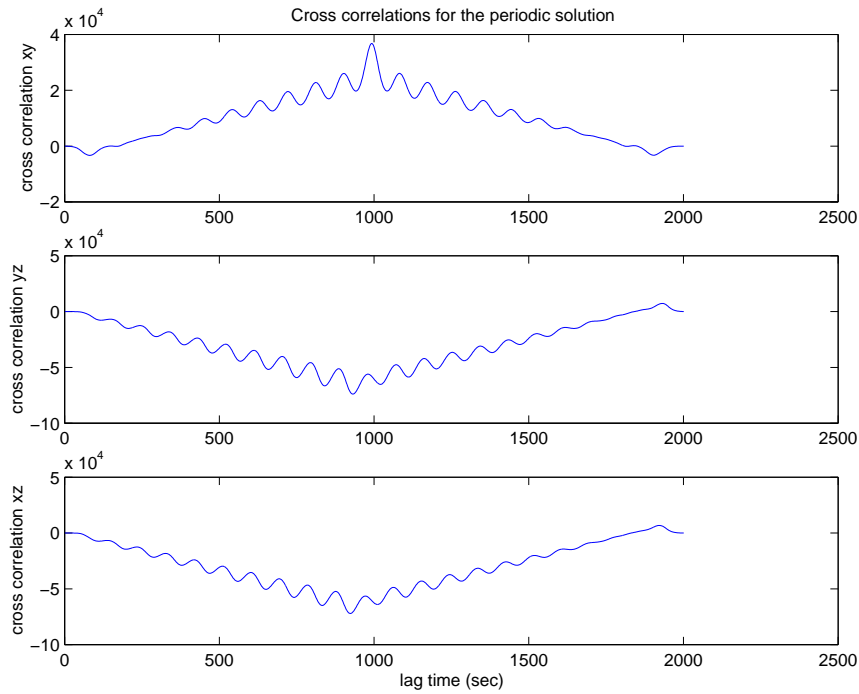


Figure 4: Cross Correlations of the x-y, y-z, and x-z signals for the Periodic Solution to the Lorentz Model

The autocorrelations for the x , y , and z signals are found in Figure 5. An autocorrelation involves only one signal and provides information about the structure of the signal or its behavior in the time domain. For a periodic signal, it is expected that when the lag time is zero, the amplitude of the autocorrelation should be one and should decrease as time increases. Looking at the autocorrelation for the x , y , and z signals, this is exactly what occurs.

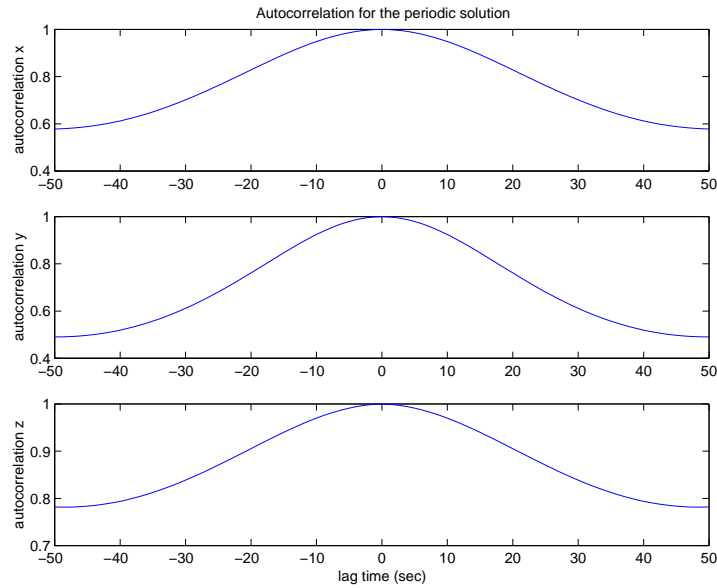


Figure 5: Autocorrelations of the x , y , and z signals for the Periodic Solution to the Lorentz Model

The final response statistic that was analyzed was the power spectral density. Figure 6 illustrates the estimated power spectral density for the x , y , and z signals.

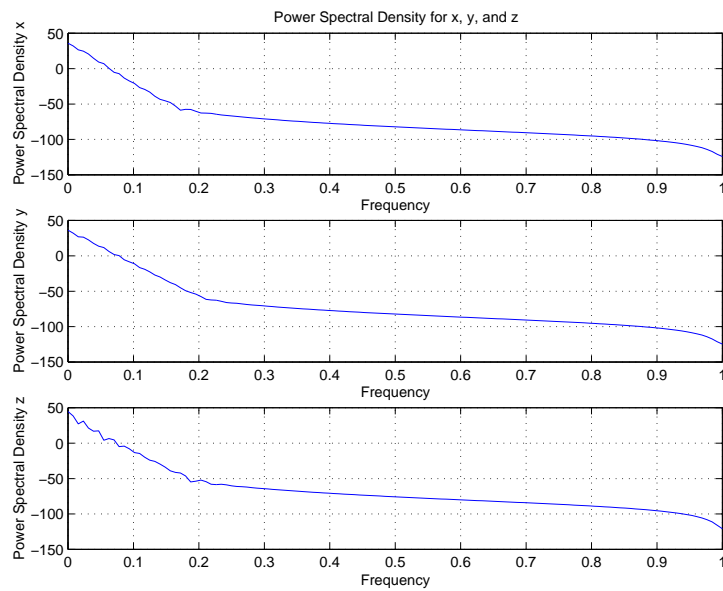


Figure 6: Power Spectral Density of the x , y , and z signals for the Periodic Solution to the Lorentz Model

2.2 Chaotic Solution

Figure 7 is a phase plot of the system when the following constants are selected.

$$\sigma = 10 \quad r = 50 \quad b = \frac{8}{3} \quad (14)$$

From this plot it can be seen that when these constants and an initial condition of (0,1,0) are selected, the Lorentz system is chaotic.

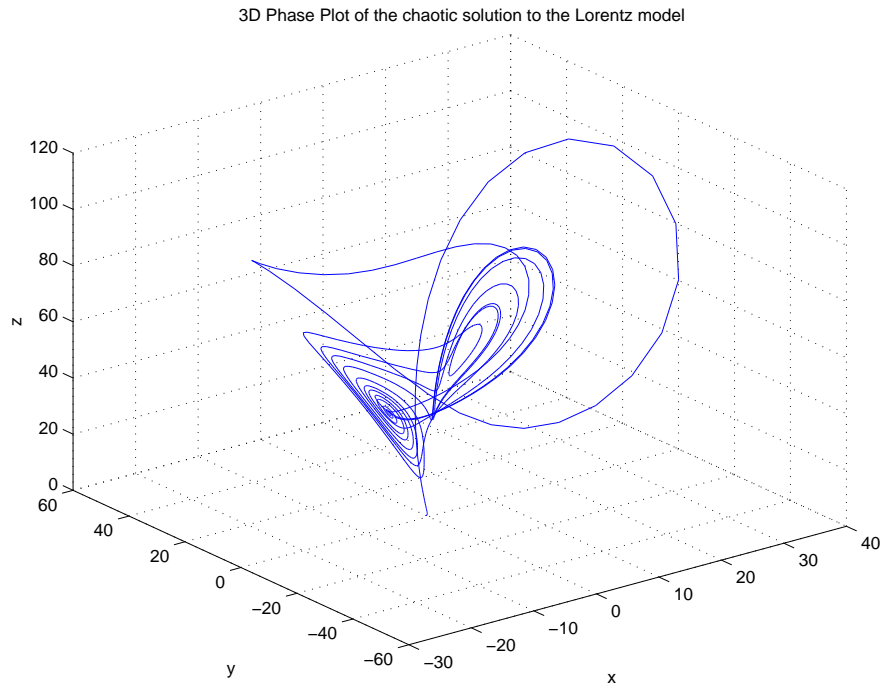


Figure 7: 3-D Phase Plot for the Chaotic Solution to the Lorentz Model

Next the time history responses were determined (Figure 8) as well as the Poincare plots (Figure 9). These plots also illustrate how by selecting the constants above, the Lorentz system becomes chaotic. Unlike Figure 2, the time history plots for the chaotic system (Figure 8) do not show a nice periodic signal. Also, the Poincare plots in Figure 3 show how the system finds its stable point, while Figure 9 illustrates how there are 2 unstable points that cause this system to be chaotic.

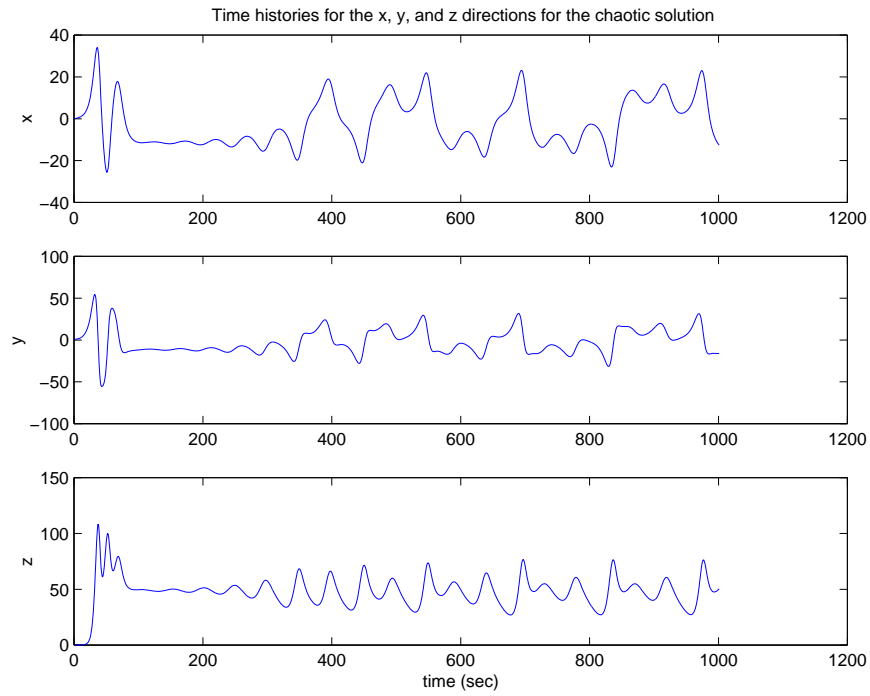


Figure 8: Time History Responses for the Chaotic Solution to the Lorentz Model

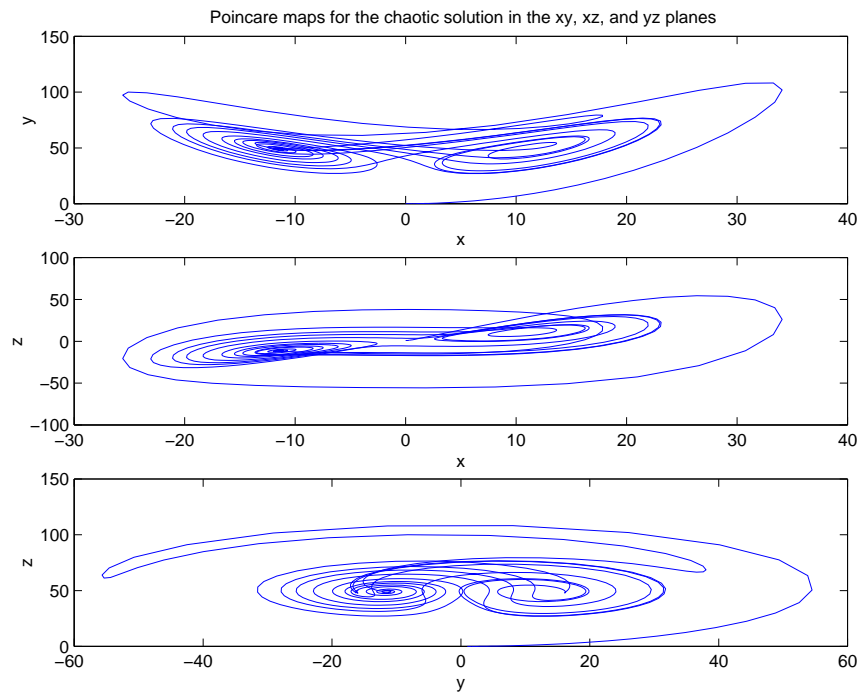


Figure 9: Poincaré Plots for the Chaotic Solution to the Lorentz Model

The response statistics of the chaotic solution were analyzed next. Table 2.2 contains the mean, mean square, and various moments for this system in each the x, y, and z directions.

	X	Y	Z
Mean	-2.3072	-2.4362	47.9134
Mean Square	137.4242	235.8749	2.4998e3
Moment (n=6)	3.4579e7	1.8236e8	1.2757e8
Moment (n=3)	0	0	0
Moment (n=20)	1.0595e30	2.7052e32	8.2209e31

Table 2: Response Statistics for Chaotic Solution to the Lorentz System

The next statistics that were investigated were the cross correlation, the autocorrelation, and the power spectral density of the signals. Figure 10 illustrates the cross correlation of the x and y signals, the x and z signals, and the y and z signals. These show that there is no real correlation between any of the signals. Again, similar to the periodic solution, there seems to be a negative correlation between the y and z signals and the x and z signals, although there isn't point lag time where the correlation is stronger than the rest. Also, there once again is a positive correlation between the x and y signals, with the strongest correlation located at a lag time of 1000 seconds. However, at every other lag time, there appears to be almost no correlation. Figure 11 is the autocorrelation. Similar to the periodic case, at a lag time of 0, all the autocorrelations are at the maximum which is 1. Figure 12 is the power spectral density. This plots looks similar to the periodic case (Figure 6) with the same values at the frequencies 0 and 1. The main difference is in the slopes of the curves.

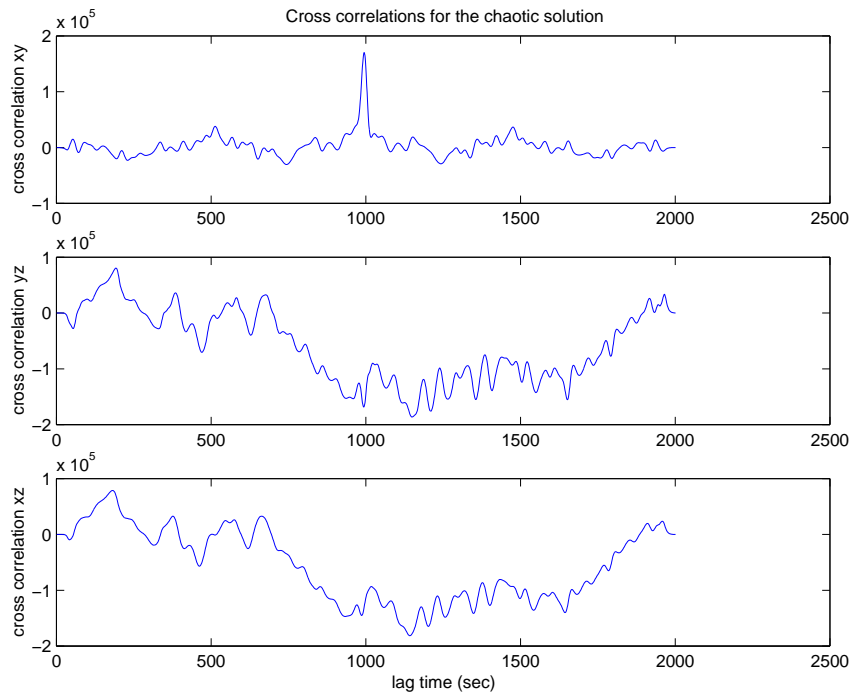


Figure 10: Cross Correlations of the x-y, y-z, and x-z signals for the Chaotic Solution to the Lorenz Model

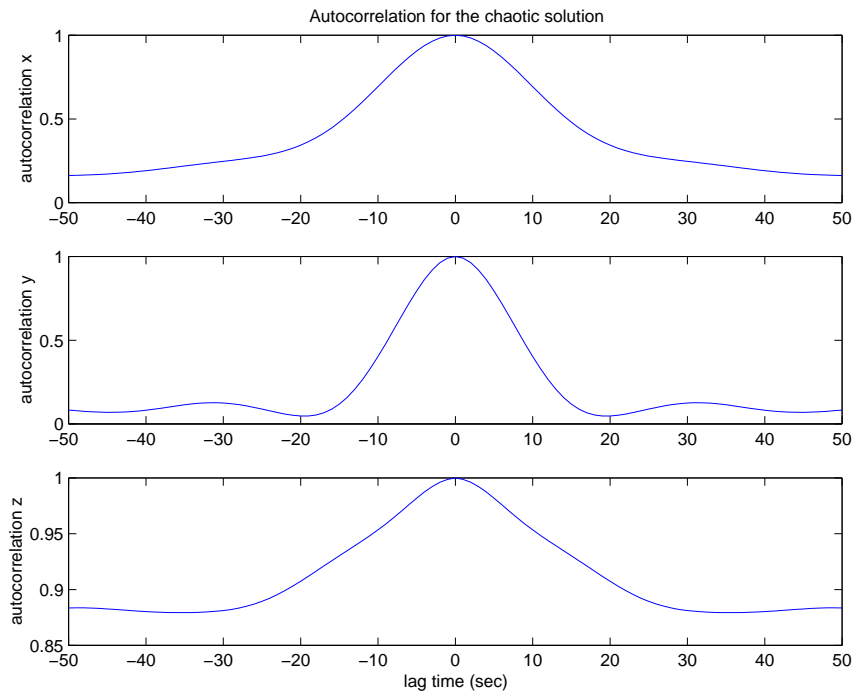


Figure 11: Autocorrelations of the x, y, and z signals for the Chaotic Solution to the Lorenz Model

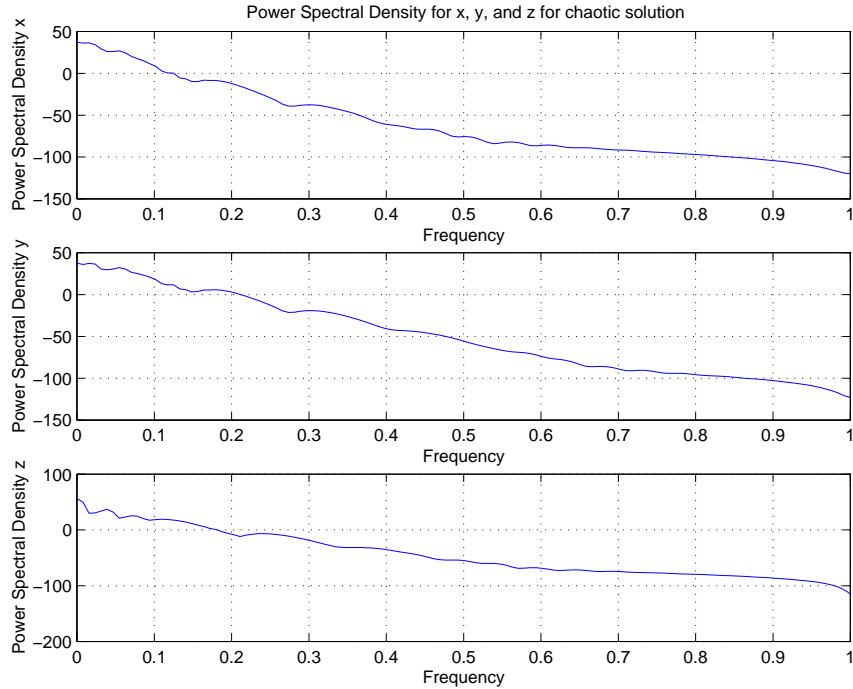


Figure 12: Power Spectral Density of the x, y, and z signals for the Chaotic Solution to the Lorenz Model

The periodic and chaotic solutions produced very different results. The time history response of the periodic solution eventually oscillated about a fixed point, while the chaotic solution's time history response had no obvious pattern to it. The Poincare plots for both cases were also very different. The periodic case illustrated that there is a definite stable point, while the chaotic case appeared to have two unstable points. The cross correlations for the periodic case illustrated a strong positive or a strong negative correlation at a lag time of 1000, while in the chaotic case there were no strong correlations that could be determined for the y and x signals and the x and z signals, while there appeared to be a correlation at a lag time of 1000 seconds for the x and y signals. The autocorrelations for the periodic case started with an amplitude of 1 and slowly the amplitude of the wave began to decrease. In the chaotic case, the slope of the autocorrelation was slightly different than in the periodic case. The power spectral density for both the chaotic and periodic case appeared to be somewhat similar, although the chaotic power spectral density is not as smooth of a curve as the power spectral density for the periodic case. The values for the mean, mean square, and moments in the periodic case were much smaller than what was determined for the chaotic case.

3 Addition of White Noise

3.1 Periodic

The following are the plots that correspond to the periodic solution to the Lorentz model with the addition of white noise.

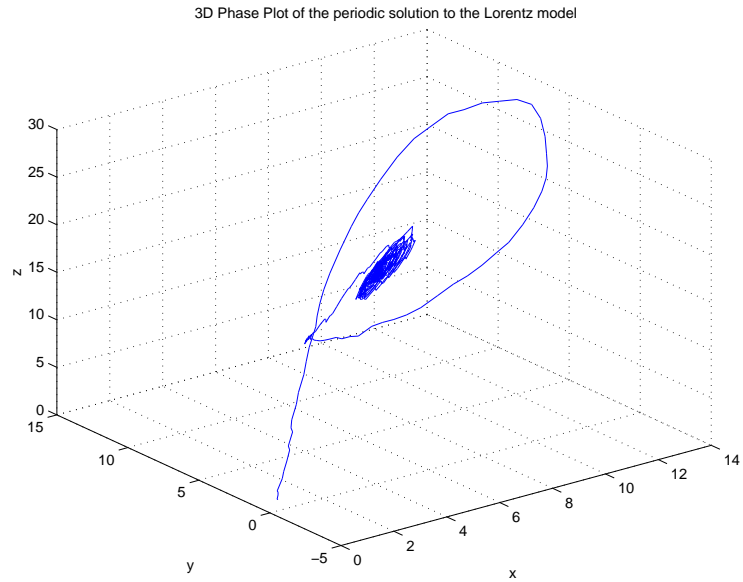


Figure 13: 3D Phase Plots of Periodic Solution to the Lorentz Model with the Addition of White Noise

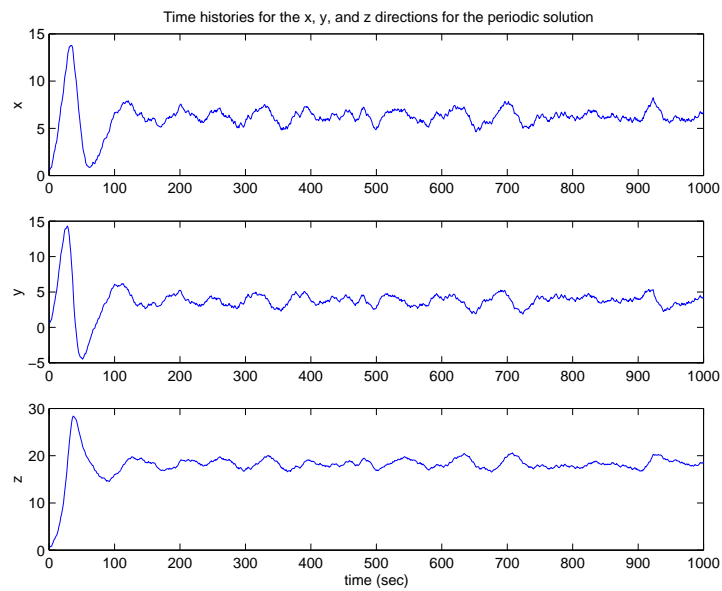


Figure 14: Time History Responses of Periodic Solution to the Lorentz Model with the Addition of White Noise

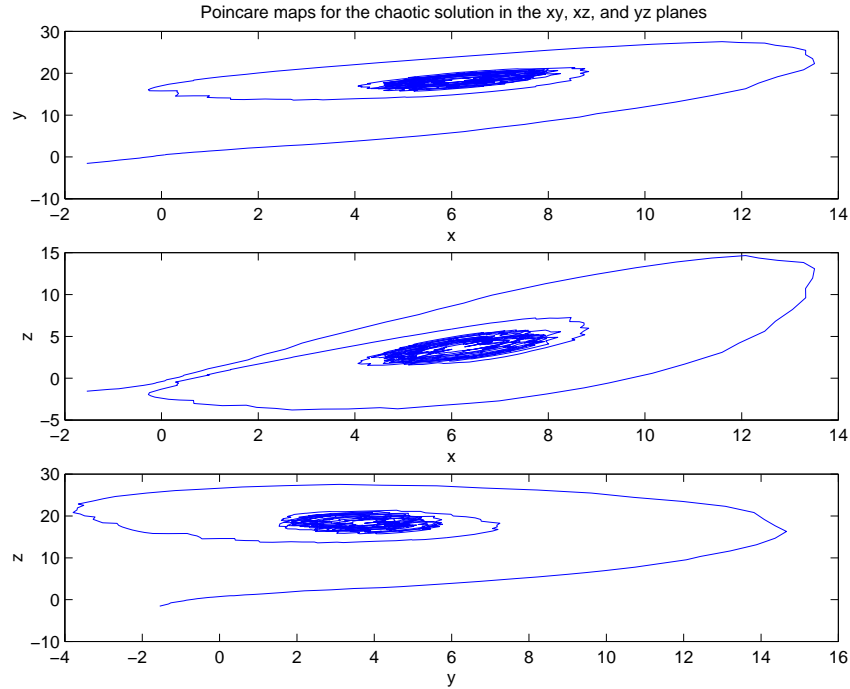


Figure 15: Poincaré Plots of Periodic Solution to the Lorentz Model with the Addition of White Noise

	X	Y	Z
Mean	6.1910	3.7685	18.0536
Mean Square	40.4063	17.5954	332.9565
Moment (n=6)	134.5193	586.4132	5.2001e3
Moment (n=3)	0	0	0
Moment (n=20)	9.8108e11	1.3277e14	1.9163e17

Table 3: Response Statistics for Periodic Solution to the Lorentz System with the Addition of White Noise

The addition of white noise to the periodic case is obvious when looking at Figures 13, 14, and 15. The main structure of the signal is still there, but it has obviously been disturbed by the white noise added. The response statistics (Table 3.1) are also greatly affected by the addition of white noise. The cross correlations are now all positive, while originally there were negative correlations. The autocorrelations are somewhat similar, although there is a difference between them. The power spectral density is also different. Not only are the values at the frequencies 0 and 1 different, there is also more noise in the signal.

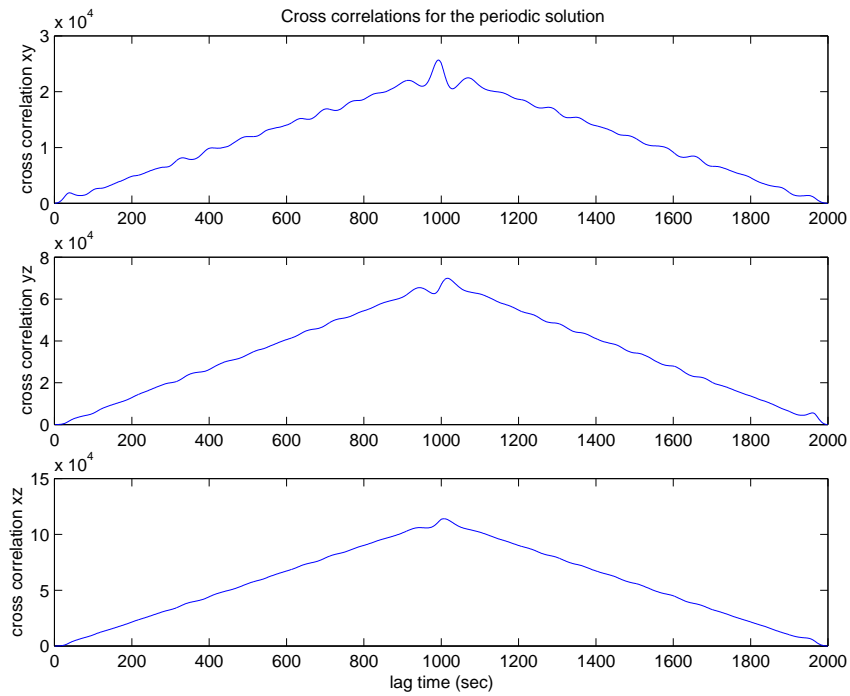


Figure 16: Cross Correlations of the x-y, y-z, and x-z signals for the Periodic Solution to the Lorentz Model with the Addition of White Noise

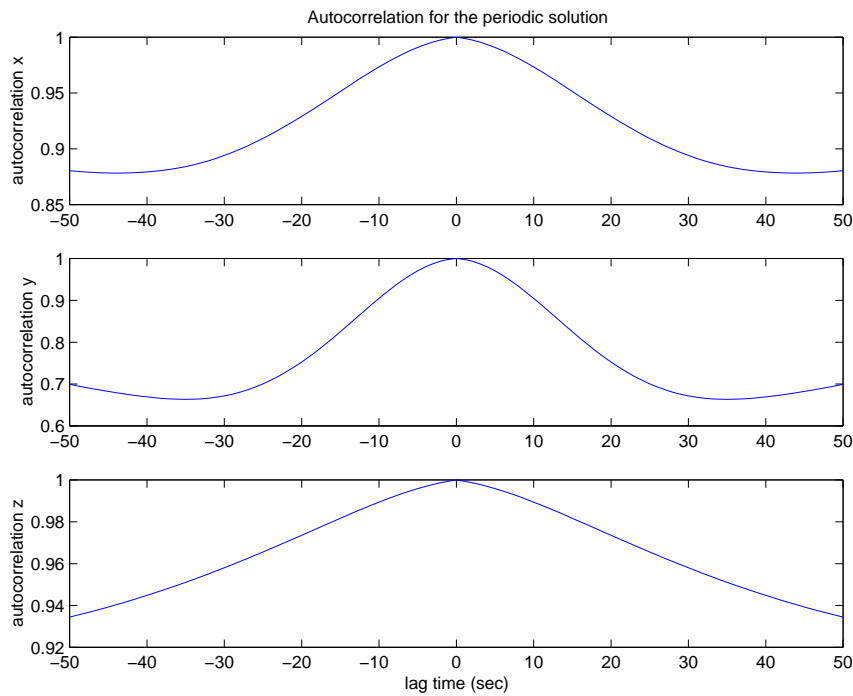


Figure 17: Autocorrelations of the x, y, and z signals for the Periodic Solution to the Lorentz Model with the Addition of White Noise

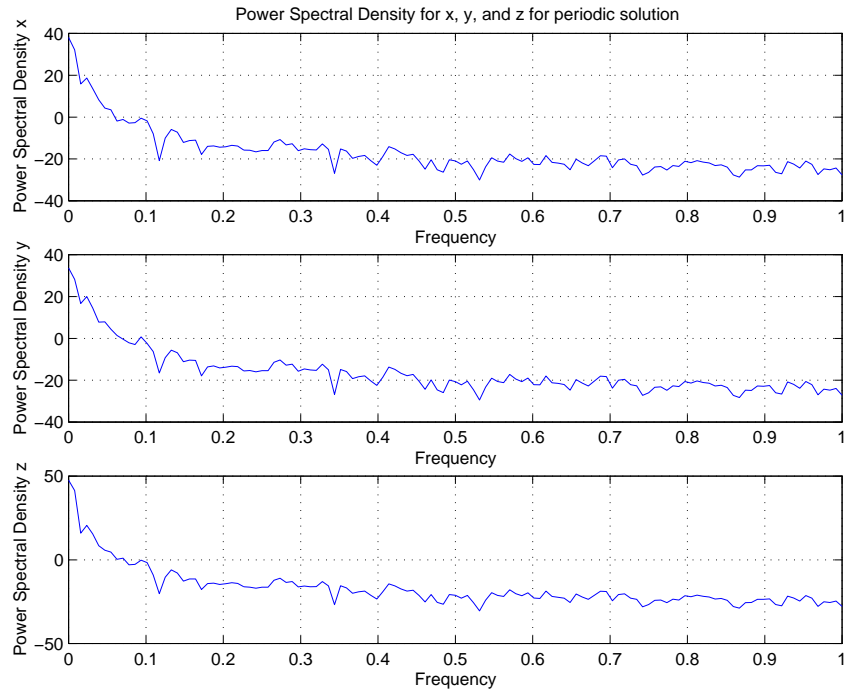


Figure 18: Power Spectral Density of the x , y , and z signals for the Periodic Solution to the Lorentz Model with the Addition of White Noise

3.2 Chaotic

The following are the plots that correspond to the chaotic solution to the Lorentz model with the addition of white noise.

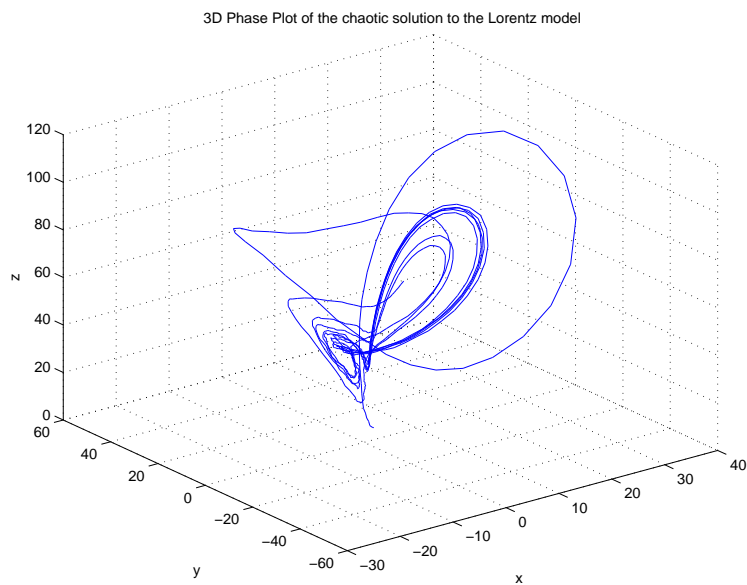


Figure 19: 3D Phase Plots of Chaotic Solution to the Lorentz Model with the Addition of White Noise

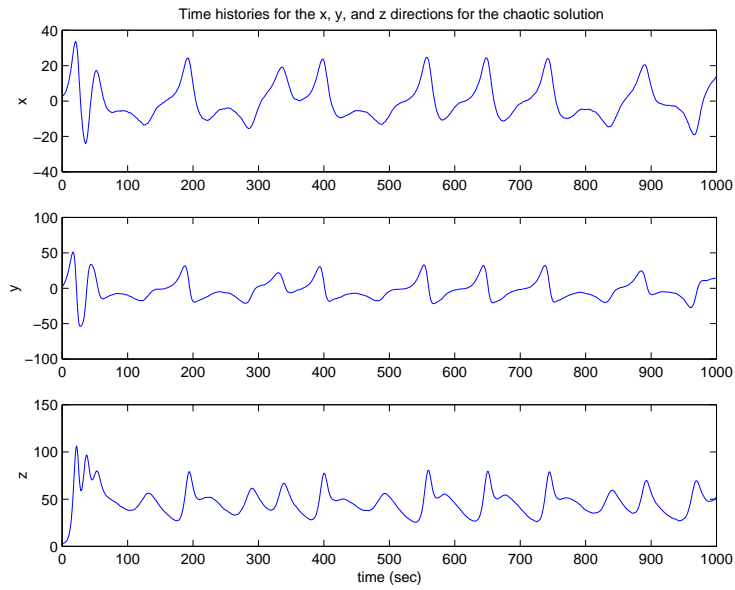


Figure 20: Time History Responses of Chaotic Solution to the Lorentz Model with the Addition of White Noise

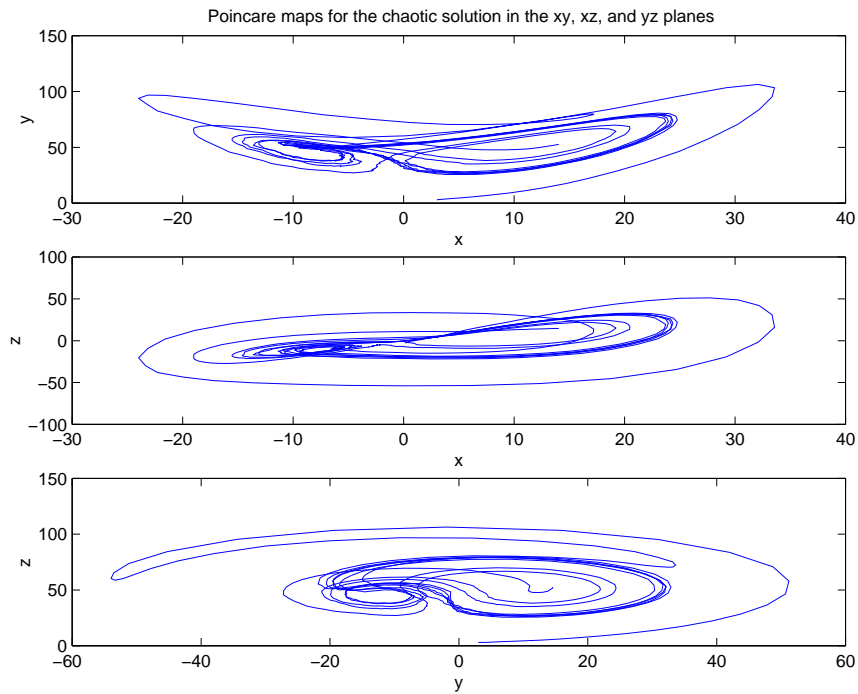


Figure 21: Poincaré Plots of Chaotic Solution to the Lorentz Model with the Addition of White Noise

Unlike the periodic case, where the addition of white noise was very evident, Figure 19, 20, and 21 illustrate that the chaotic solution with white noise doesn't look much different than the original chaotic solution.

	X	Y	Z
Mean	-0.2058	-2.5672	46.8334
Mean Square	108.6045	220.1332	2.3967e3
Moment (n=6)	1.9192e7	1.4606e8	1.2617e8
Moment (n=3)	0	0	0
Moment (n=20)	1.4888e29	1.2909e32	7.9233e31

Table 4: Response Statistics for Chaotic Solution to the Lorentz System with the Addition of White Noise

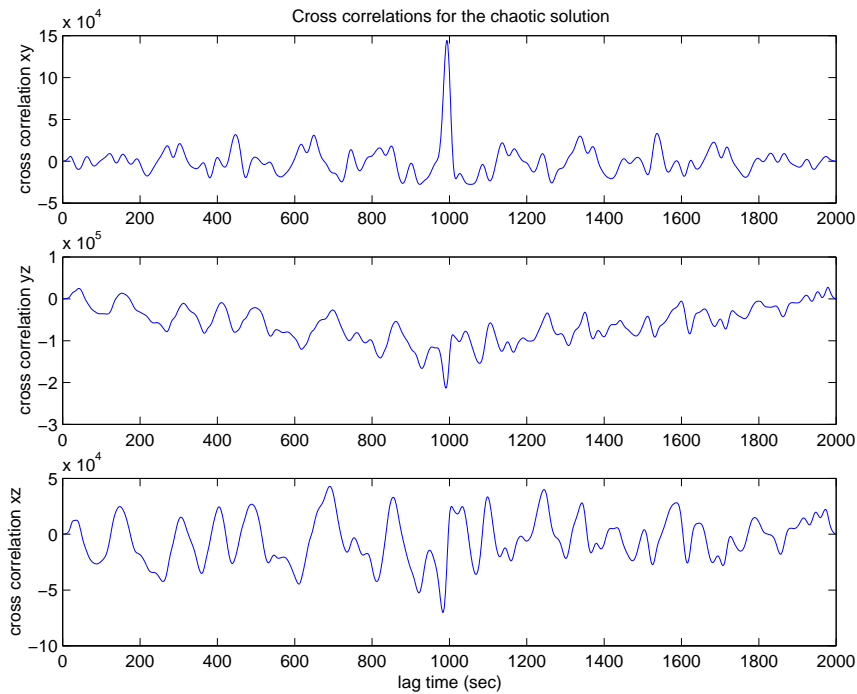


Figure 22: Cross Correlations of the x-y, y-z, and x-z signals for the Chaotic Solution to the Lorentz Model with the Addition of White Noise

The response statistics for the chaotic solution with the addition of white noise are more similar to the original chaotic solution than what was observed in the periodic cases. The parameters are not exactly the same, but they are much closer to one another than what was observed in the periodic cases. The same is true for the cross correlations and the autocorrelations. The power spectral density plots are not quite the same and appear to be similar to the power spectral density plots for the periodic case with white noise.

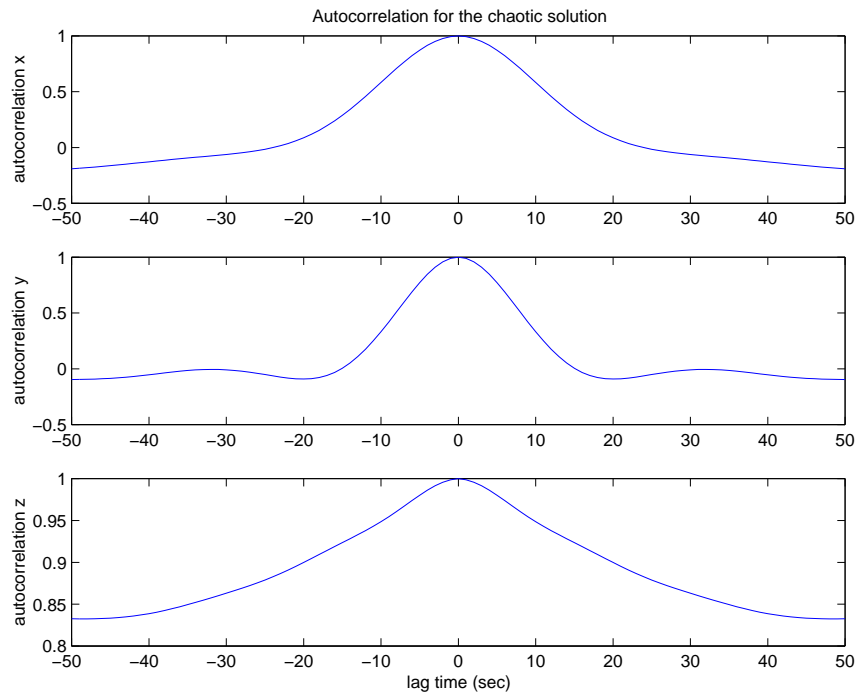


Figure 23: Autocorrelations of the x, y, and z signals for the Chaotic Solution to the Lorentz Model with the Addition of White Noise

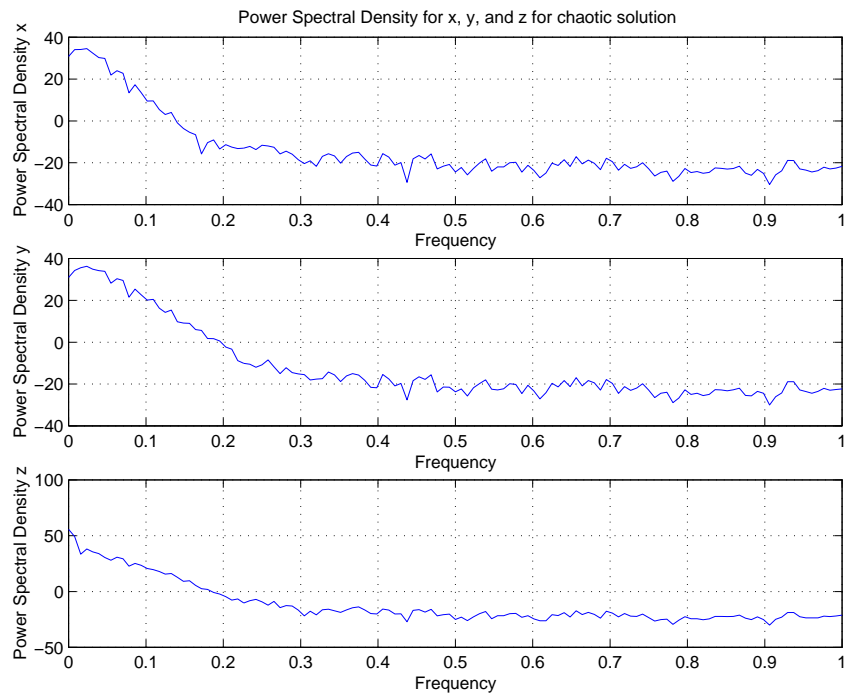


Figure 24: Power Spectral Density of the x, y, and z signals for the Chaotic Solution to the Lorentz Model with the Addition of White Noise

By comparing the cases when white noise is present and when it is absent, it can be seen that the white noise disrupts the solution to the Lorentz model significantly. Figure 25 illustrates the x signal for the periodic solution with and without white noise, while Figure 26 shows the chaotic case.

Not only are the phase plots, time history responses, and poincare plots altered, but all the response statistics are also affected. It is interesting to note that for both the chaotic and periodic solutions, the xy cross correlation has a similar spike at a lag time of 1000 for both the case with white noise and without white noise. There appears to be a similarity in the autocorrelation of the z signal as well.

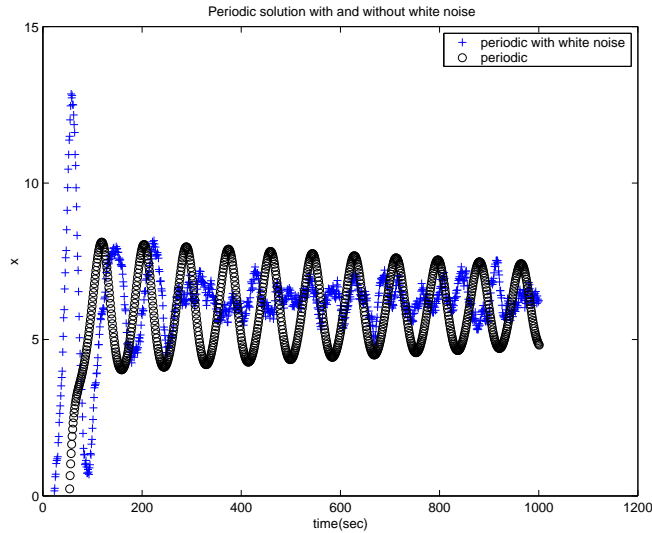


Figure 25: Periodic Solutions with and without White Noise

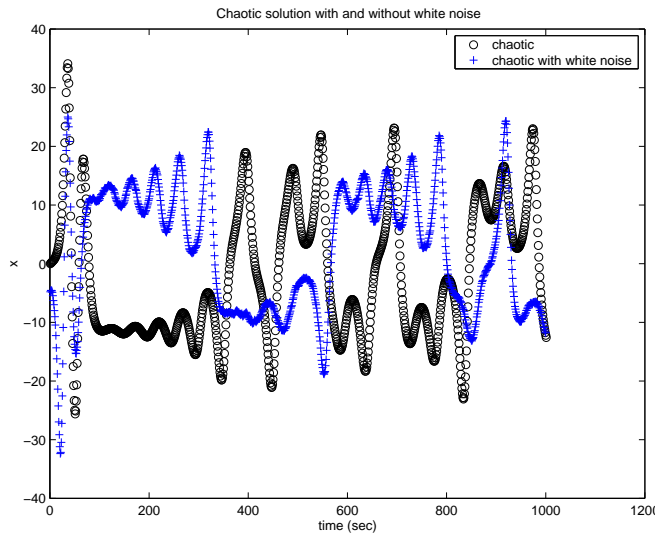


Figure 26: Chaotic Solutions with and without White Noise

4 Karhunen-Loeve Expansion

For the chaotic model with white noise, an orthogonal basis set for the Karhunen-Loeve expansions was analyzed. With the system of equations for the Lorentz model being defined as the vector (r), the integral eigenvalue problem can be written as follows:

$$\int \int R_{ij}(t, t') \psi_j(t') dt' = \lambda \psi_i(t) \quad (15)$$

R_{ij} is the ensemble averaged two-point correlation tensor and is defined as

$$R_{ij}(t, t') = \langle r_i(t) r_j(t') \rangle \quad (16)$$

This kernel R_{ij} will always be symmetric. Due to this property, the following properties hold:

- There is a discrete set of solution to equation 15

$$\int_D R_{ij}(t, t') \phi_j^n(t') dt' = \lambda^n \phi_i^n(t) \quad (17)$$

- The solution to equation 15 can be selected such that ϕ^n are orthonormal:

$$\int_D \phi_i^p(t) \phi_i^q(t) dt = \delta_{pq} \quad (18)$$

- The infinite series of solutions, ϕ^n , can be used to reconstruct the original fluctuating quantity, $r(t)$:

$$r_i(t) = \sum_{n=1}^{\infty} a_n \phi_i^n(t) \quad (19)$$

where the coefficients are defined as:

$$a_n = \int_D u_i(t) \phi_i^{(n)*}(t) dt \quad (20)$$

The eigenvalues and eigenvectors of the two-point correlation tensor were then determined.

Using the Karhunen-Loeve Expansion Matlab Script provided at the end of this report, reconstruction of the Lorentz model was performed. The kernel was constructed from 3000 realizations of the model, each excited with white noise. The white noise algorithm was seeded with the time to ensure independence between the blocks. Figure 27 illustrates the original x , y , and z values. The first parameter to be evaluated was the percent mean square energy that each eigenvalue carries. From Figure 28 it can be seen that most of the energy is located in the first few modes, with little energy is the higher modes. Next, the first three eigenfunctions were examined (Figure 29). There are a total of 300 eigenfunctions. Using the first set of eigenfunctions, reconstruction was done. This can be seen in Figure 30 for x , y , and z . Reconstruction was done for up to modes 5, 10, 45, 50, 55, 100, and 300 (Figures 31 - 37). At mode 300, all modes are used, and therefore, the reconstruction is exactly the same as the original data set.

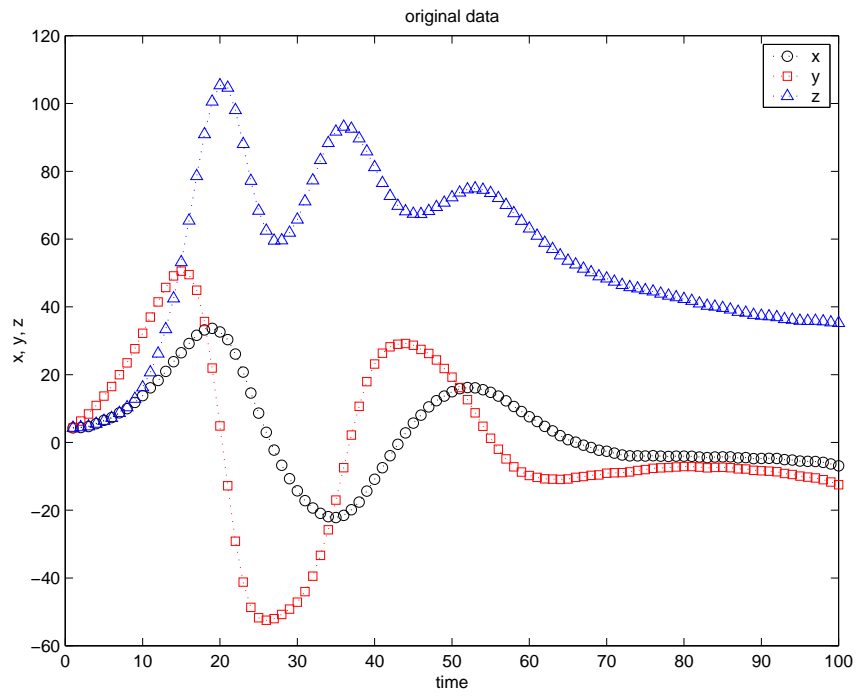


Figure 27: Original Data Set

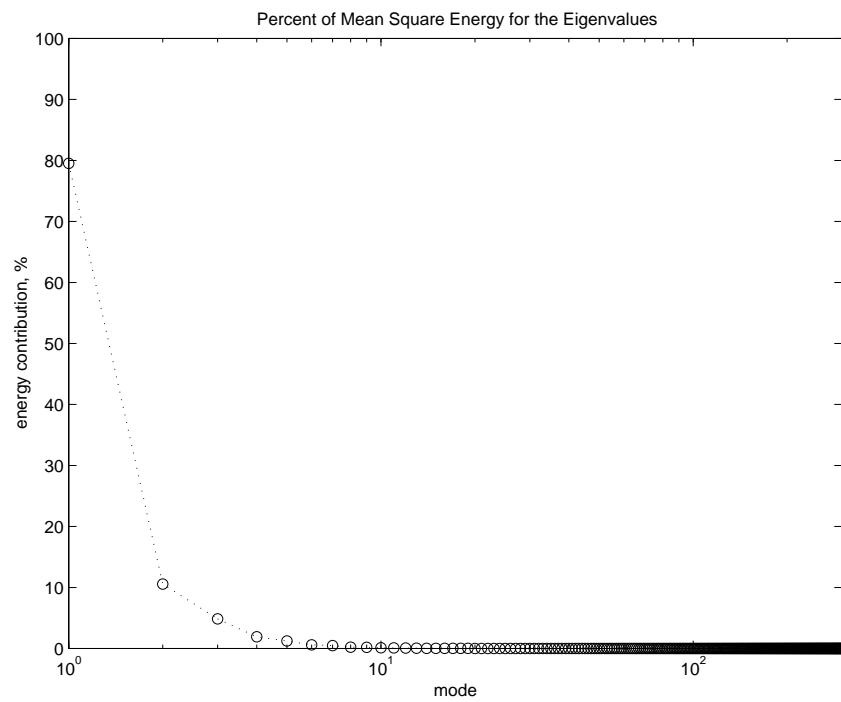


Figure 28: Percent of Mean Square Energy for the Eigenvalues

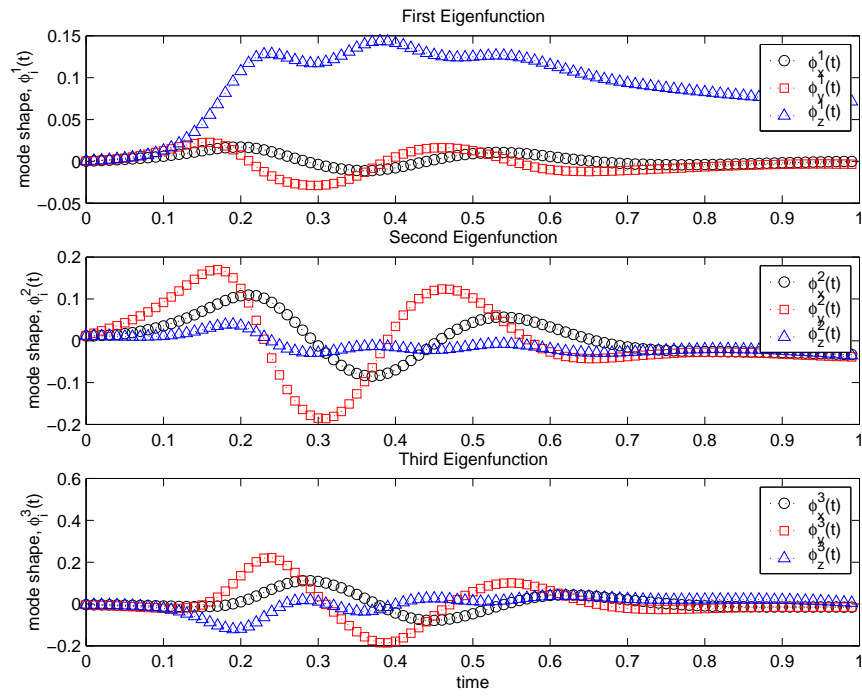


Figure 29: First Three Eigenfunctions for the x, y, and z directions

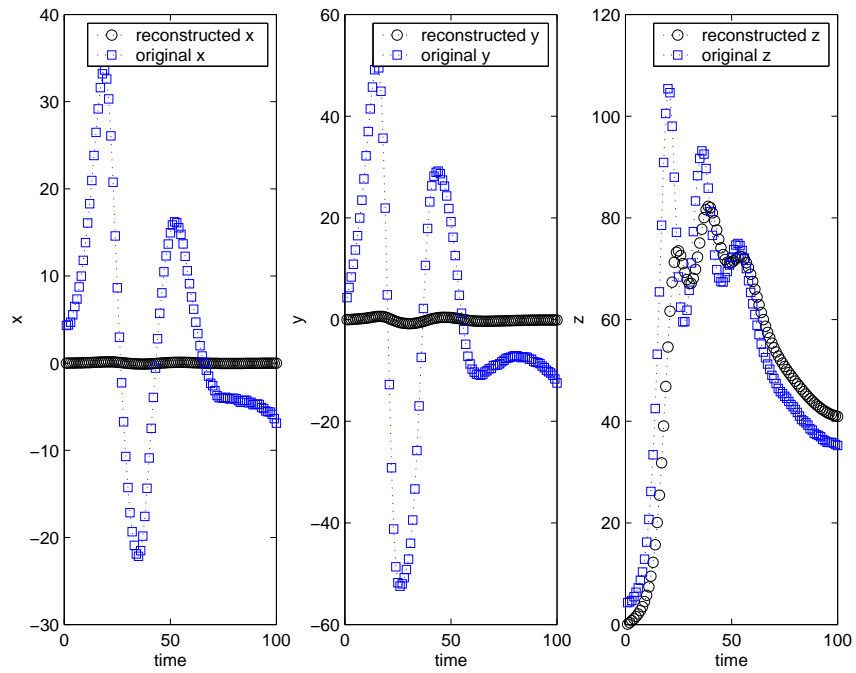


Figure 30: Reconstruction up to Mode 1

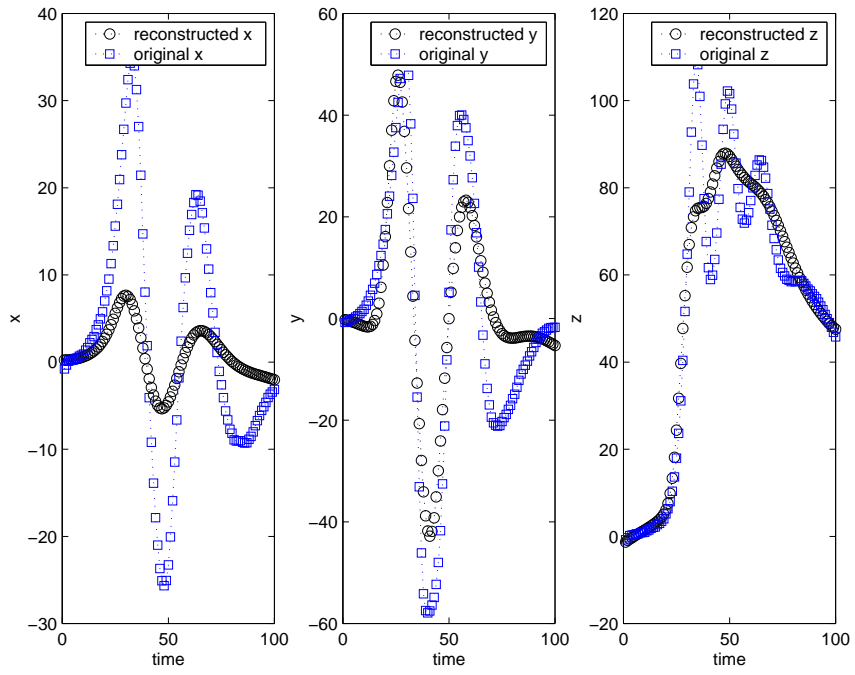


Figure 31: Reconstruction up to Mode 5

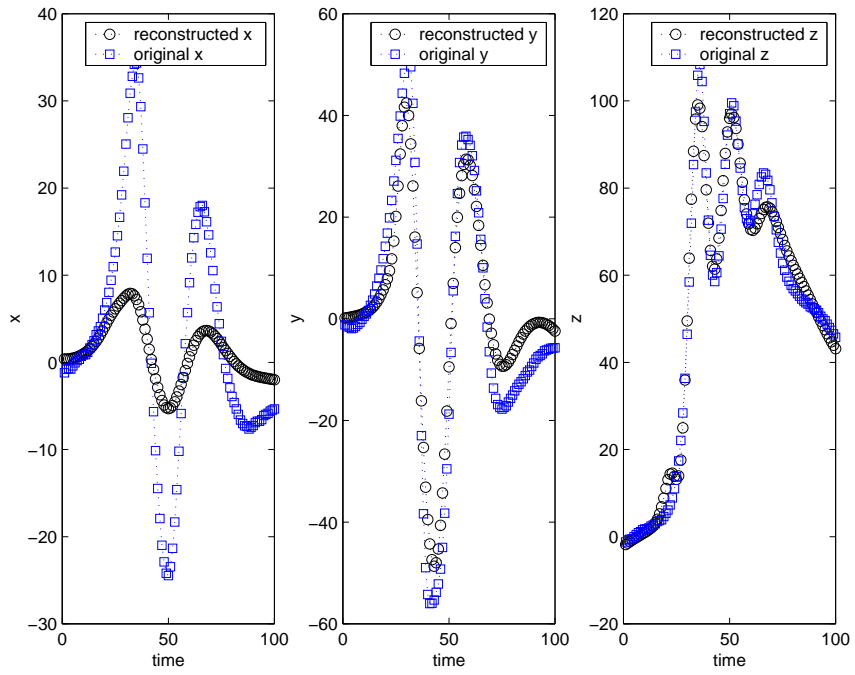


Figure 32: Reconstruction up to Mode 10

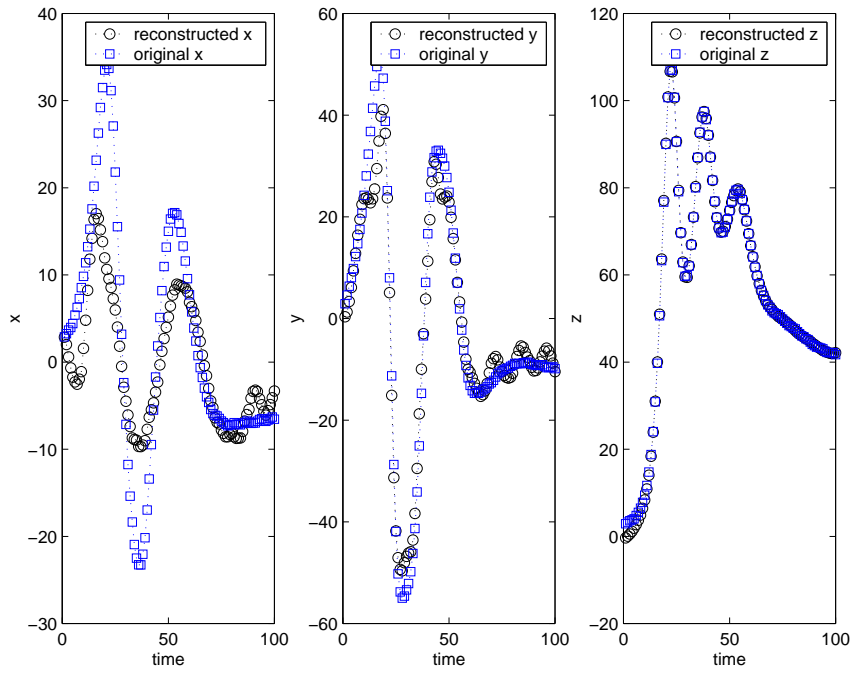


Figure 33: Reconstruction up to Mode 45

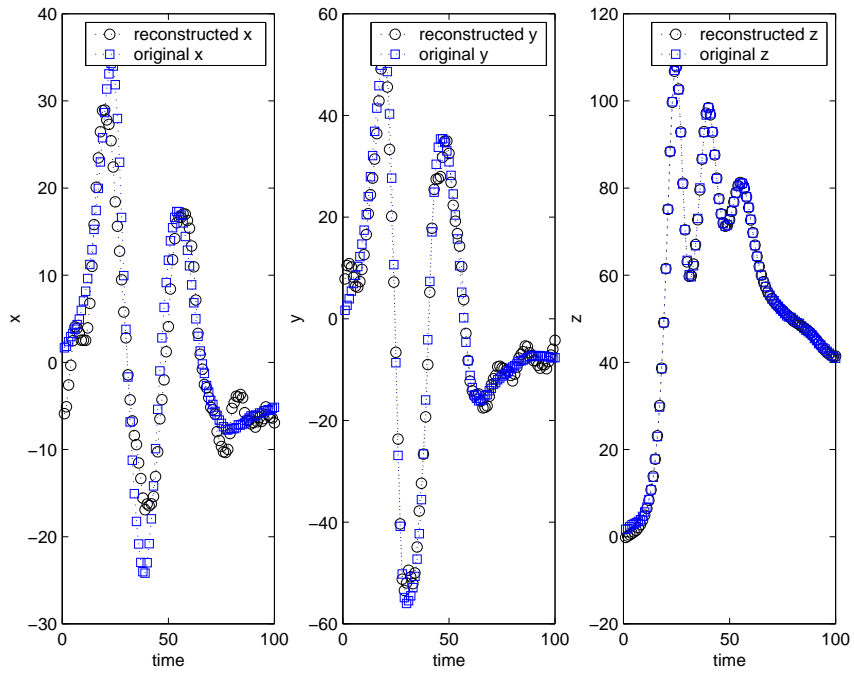


Figure 34: Reconstruction up to Mode 50

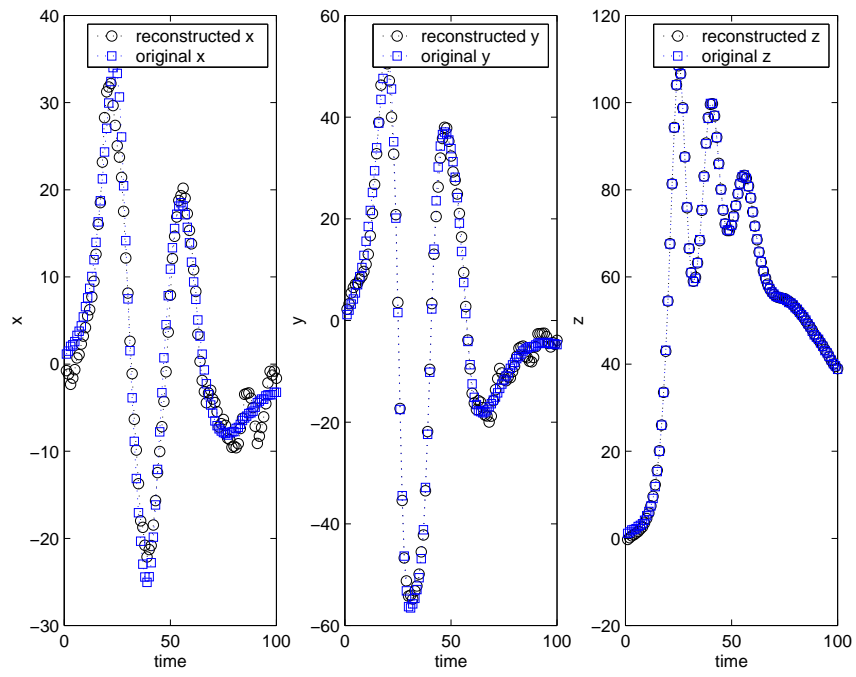


Figure 35: Reconstruction up to Mode 55

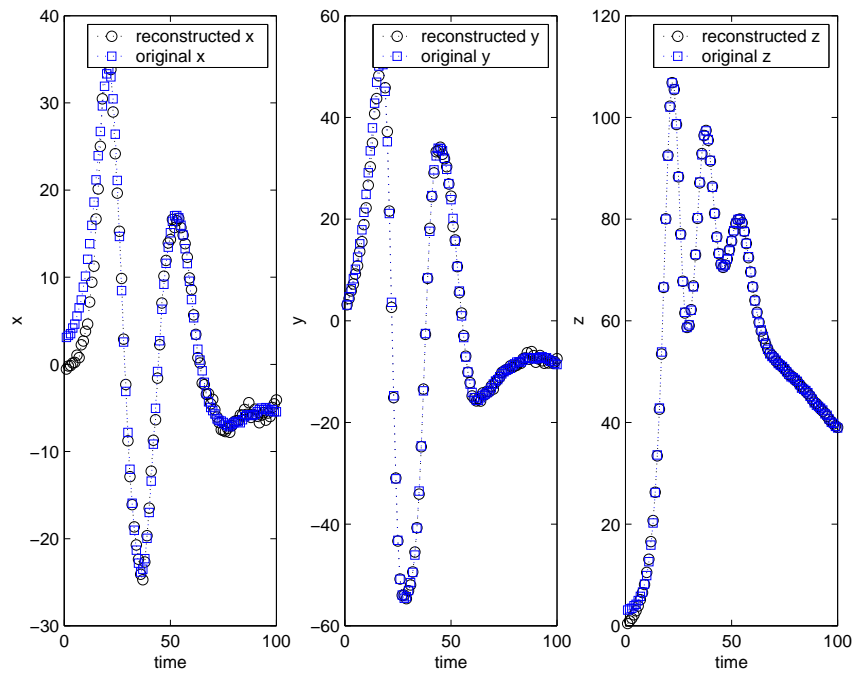


Figure 36: Reconstruction up to Mode 100

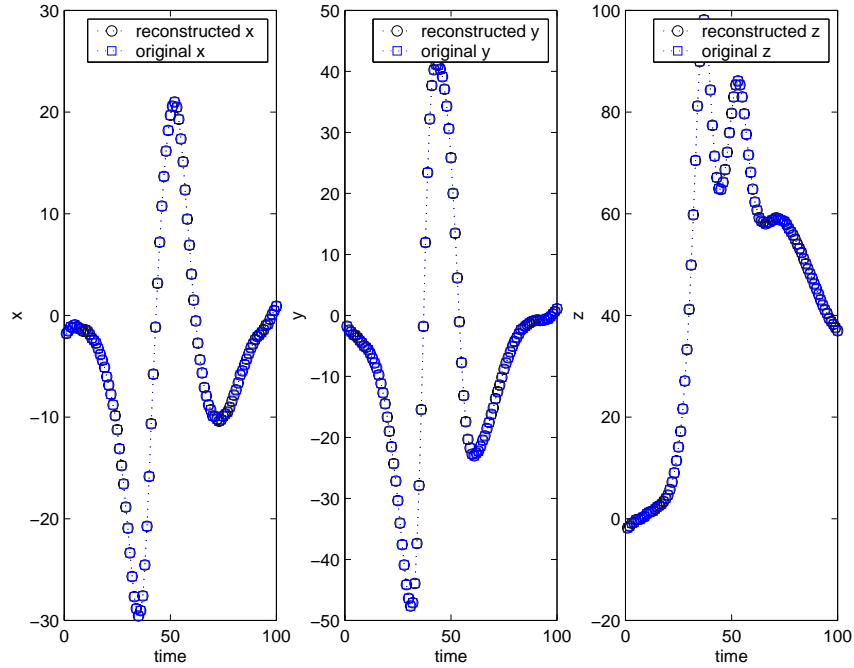


Figure 37: Reconstruction up to Mode 300

An interesting phenomenon that occurred was that while the y and z directions were reconstructed fairly accurately at lower mode numbers, the x direction wasn't. It wasn't until mode 50 that the x direction was reconstructed in a manner that was close to the original data. Similarly, 50 is also the value selected for the constant r in the Lorenz model. This is the constant which determines if the solution is periodic or chaotic. It was then changed to 40 to determine if the reconstruction in any way depended on this value. Figures 38 - 40 show that before reaching mode 40, the x direction is not reconstructed well, while after mode 40, it is reconstructed closer to the original value. Other values were selected for r and the same phenomenon occurred. Therefore, it was concluded that the constant r plays a role in the reconstruction of the Lorenz model. The values of the coefficients were then plotted against the mode number and it showed that there was energy located at these higher modes. Figures 41 and 42 show the coefficients for the case with $r = 40$ and $r = 50$ respectively.

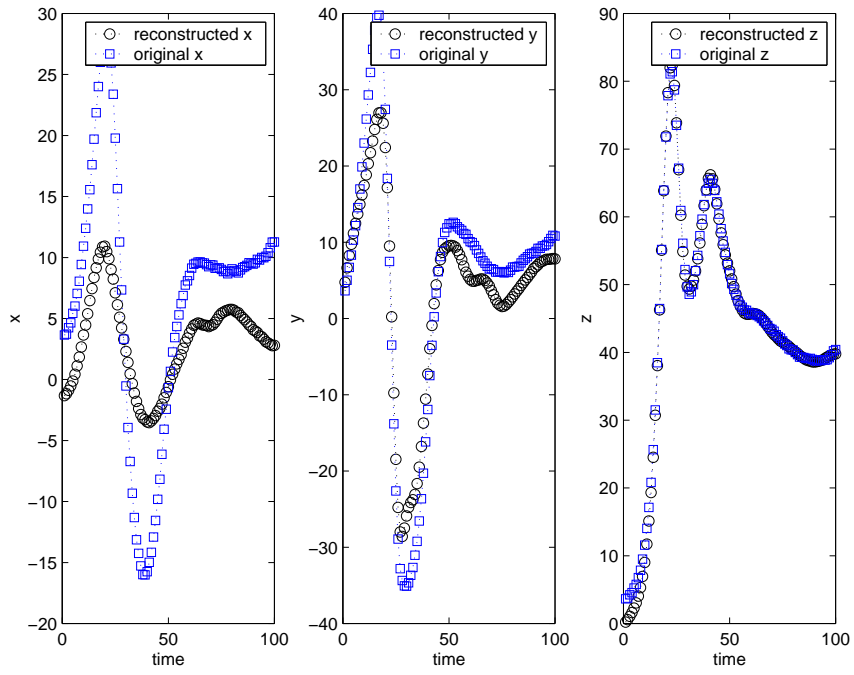


Figure 38: Reconstruction up to Mode 30 with $r = 40$

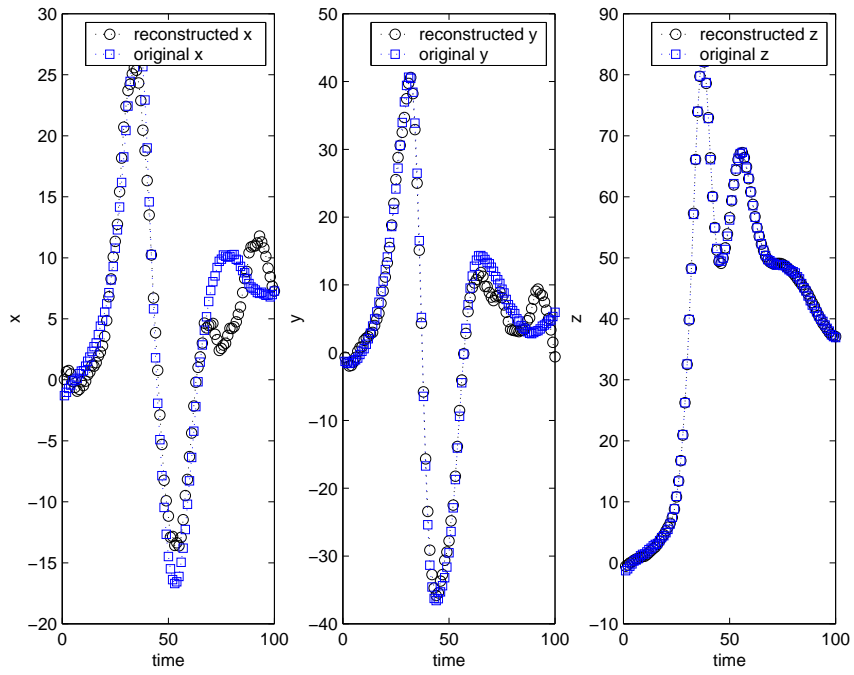


Figure 39: Reconstruction up to Mode 40 with $r = 40$

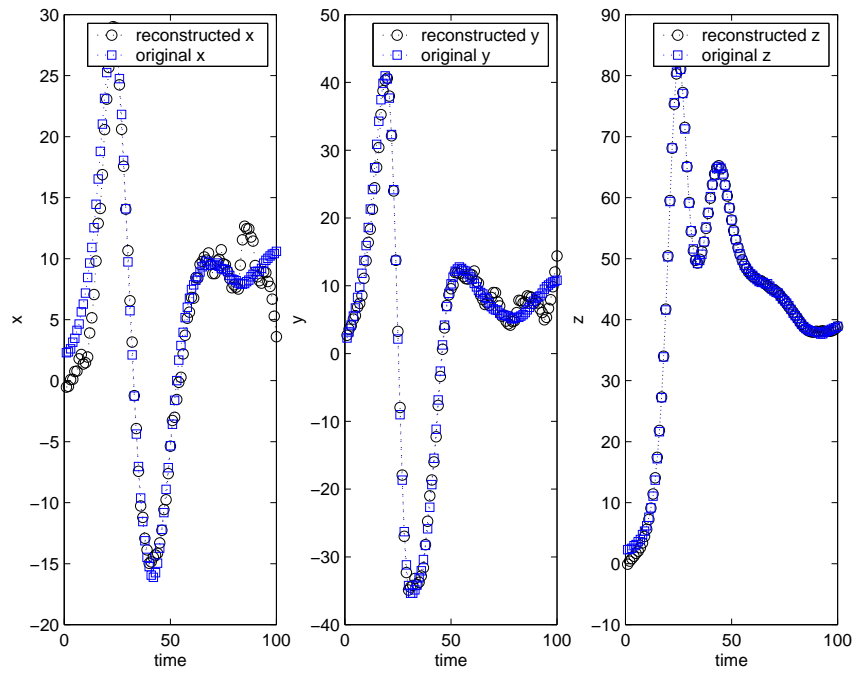


Figure 40: Reconstruction up to Mode 50 with $r = 40$

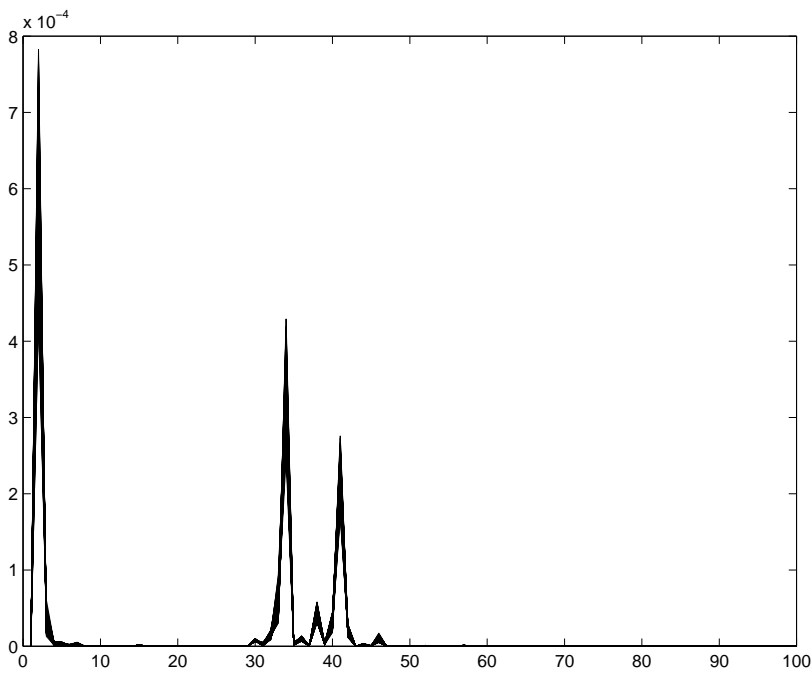


Figure 41: Coefficients for $r = 40$

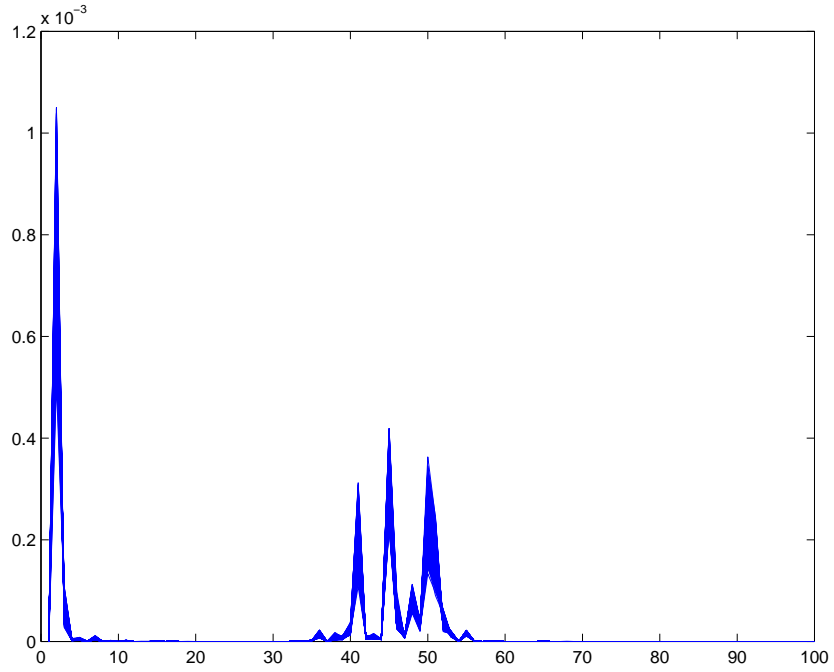


Figure 42: Coefficients for $r = 50$

What this reconstruction illustrated was that there is a large amount of energy surrounding the mode number that is equal to the chaotic parameter r . This large amount of energy affects the reconstruction in the x direction significantly more than the y and z directions.

5 Matlab Script

This is the Matlab script that was used to create the figures.

```
% x' = -ax + ay
% y' = bx - y - xz
% z' = -cz + xy

close all clear clc

step=0.01; t=1000; x=[0,1,0]; a=[10,15,8/3];
%a for chaotic [10,50,8/3], a for periodic [10,15,8/3]

%without white noise
%r=zeros(3,t+1);
%r(:,1) = x';

%for i=1:t
%r(:,i+1) = r(:,i) + step.*[-a(1).*r(1,i) + a(1).*r(2,i);
    a(2).*r(1,i) - (r(2,i) + r(1,i).*r(3,i));(-a(3))*r(3,i) +
    r(1,i).*r(2,i)];
%end;

%with white noise
% initialize

rand('state',sum(100*clock)); x = 10*(rand(size(t))-0.5);

r = zeros(3,t); r(:,1) = x';

% Solve for r
for i = 1:t-1, r(:,i+1) = r(:,i) + 0.5*rand(size(t)) +
step.*[-a(1).*r(1,i) + a(1).*r(2,i); a(2).*r(1,i) - (r(2,i) +
r(1,i).*r(3,i)); (-a(3))*r(3,i) + r(1,i).*r(2,i)]; end

figure plot3(r(1,:),r(2,:),r(3,:))
title('3D Phase Plot of the
periodic solution to the Lorentz model')
xlabel('x')
ylabel('y')
zlabel('z')
grid on figure

subplot(3,1,1), plot(r(1,:))
title('Time histories for the x, y,
and z directions for the periodic solution')
%xlabel('time (sec)')
ylabel('x') subplot(3,1,2), plot(r(2,:))
%xlabel('time (sec)')
ylabel('y') subplot(3,1,3), plot(r(3,:))
xlabel('time (sec)')
ylabel('z')

figure subplot(3,1,1) h=plot(r(1,:),r(3,:))
set(h,'EraseMode','xor','MarkerSize',18)
title('Poincare maps for
the periodic solution in the xy, xz, and yz planes')
xlabel('x')
ylabel('y')

subplot(3,1,2) n=plot(r(1,:),r(2,:))
set(n,'EraseMode','xor','MarkerSize',18)
xlabel('x')
ylabel('z')
```

```

subplot(3,1,3) p=plot(r(2,:),r(3,:))
set(p,'EraseMode','xor','MarkerSize',18)
xlabel('y')
ylabel('z')

%cross and auto correlation:
figure xyc = xcorr(r(1,:),r(2,:));
subplot(3,1,1), plot(xyc)
xlabel('lag time (sec)')
title('Cross correlations for the periodic solution')
ylabel('cross correlation xy')
yzc = xcorr(r(2,:),r(3,:));
subplot(3,1,2), plot(yzc)
xlabel('lag time (sec)')
ylabel('cross correlation yz')
zxc = xcorr(r(1,:),r(3,:));
subplot(3,1,3),plot(zxc)
xlabel('lag time (sec)')
ylabel('crosscorrelation xz')

figure subplot(3,1,1) [ac,lags] = xcorr(r(1,:),50,'coeff');
plot(lags,ac)
xlabel('lag time (sec)')
ylabel('autocorrelation x') title('Autocorrelation for the
periodic solution')
subplot(3,1,2)
[ac,lags]=xcorr(r(2,:),50,'coeff');
plot(lags,ac)
xlabel('lag time (sec)')
ylabel('autocorrelation y') subplot(3,1,3)
[ac,lags]=xcorr(r(3,:),50,'coeff'); plot(lags,ac)
xlabel('lagtime(sec)')
ylabel('autocorrelation z')

%mean and mean square
Mx = mean(r(1,:)); My = mean(r(2,:)); Mz = mean(r(3,:)); Mx My Mz
Mx2 = mean((r(1,:)).^2); My2 = mean((r(2,:)).^2); Mz2 =
mean((r(3,:)).^2); Mx2 My2 Mz2

%power spectral density
figure subplot(3,1,1) PSD(r(1,:))
%spectrum(r(1,:))
xlabel('frequency')
ylabel('Power Spectral Density x')
title('Power Spectral Density
for x, y, and z for periodic solution')
subplot(3,1,2) PSD(r(2,:))
%spectrum(r(2,:))
xlabel('frequency')
ylabel('Power Spectral Density y')
subplot(3,1,3) PSD(r(3,:))
%spectrum(r(3,:))
xlabel('frequency')
ylabel('Power Spectral Density z')

%moments, n=6, n=3, n=20, respectively
Mx6 = 1*3*5*(Mx2-Mx^2)^(6/2);
Mx3 = 0*2*(Mx2-Mx^2)^(3/2);
Mx20=1*3*5*7*9*11*13*15*17*19*(Mx2-Mx^2)^(20/2);
My6=1*3*5*(My2-My^2)^(6/2);
My3 = 0*2*(My2-My^2)^(3/2);
My20=1*3*5*7*9*11*13*15*17*19*(My2-My^2)^(20/2);
Mz6=1*3*5*(Mz2-Mz^2)^(6/2);

```

```
Mz3 = 0*2*(Mz2-Mz^2)^(3/2);  
Mz20=1*3*5*7*9*11*13*15*17*19*(Mz2-Mz^2)^(20/2);
```

```
Mx6  
My6  
Mz6  
Mx3  
My3  
Mz3  
Mx20  
My20  
Mz20
```


6 Karhunen-Loeve Expansion Matlab Script

This is the Matlab script that was used to perform the Karhunen-Loeve Expansion and to create all the plots.

```
% Housekeeping
clear clc close all

% Define the size of the time step, dt
dt = 0.01;

% Define the number of iterations, N
N = 100; % Keep M*N below 1000!!

%a for chaotic [10,50,8/3], a for periodic [10,15,8/3]
a = [10,50,8/3];

% number of variables
M = 3;

% number of times we are averaging the kernel
numLoops = 3000;

% Initialize
sumRij( 1:M*N , 1:M*N) = 0;

% Begin looping through initial conditions
for initCond = 1:numLoops,

% initialize
rand('state',sum(100*clock)); x =10*(rand(size(N))-0.5);

r = zeros(3,N); r(1:3,1:N) = x';

% Solve for r
for i = 1:N-1, r(:,i+1) = r(:,i) + 0.5*rand(size(N)) +
dt.*[-a(1).*r(1,i) + a(1).*r(2,i); a(2).*r(1,i) - (r(2,i) +
r(1,i)).*r(3,i)]; (-a(3))*r(3,i) + r(1,i).*r(2,i)]; end

% convert the MxN matrix to a vector thats MNx1
xVec = reshape(r,M*N,1);

sumRij = sumRij + (xVec * xVec')/numLoops;

% end looping through the initial conditions
end

% Construct weights for integration
weights(1:3,1) = (dt).^(0.5);
weights(1:3,2:N-1) = (dt).^(0.5);
weights(1:3,N) = (dt).^(0.5);
W=reshape(weights,M*N,1)*(reshape(weights,M*N,1))';

% Combine the data vector with the weights AND
% Solve the eigenvalue problem
[V, D] = eig(sumRij.*W);

% clean up the eigenvalue and eigenvector ordering
eVals = flipud(diag(D));
eVecs = fliplr(V);

% take a look at the data
figure semilogx([1:1:M*N], eVals,'ko:');
axis([1 M*N 0 eVals(1)]);
title('Mean Square Energy for the Eigenvalues')
xlabel('mode');
ylabel('mean square energy');
```

```

figure semilogx([1:1:M*N], eVals/sum(eVals)*100,'ko:');
axis([1
M*N 0 100]);
title('Percent of Mean Square Energy for the
Eigenvalues')
xlabel('mode');
ylabel('energy contribution, %');

% use reshape to pull the three components out of the eigenfunctions

% plot up the first eigenfuction
eVecComps = reshape(squeeze(eVecs(:,1)),M,N);
figure
subplot(3,1,1), plot([0:dt:dt*N-dt],eVecComps(1,:), 'ko:');
hold on
plot([0:dt:dt*N-dt],eVecComps(2,:), 'rs:');
plot([0:dt:dt*N-dt],eVecComps(3,:), 'b^:');
hold off title('First
Eigenfunction')
xlabel('time')
ylabel('mode shape, \phi_i^1(t)')
legend('\phi_x^1(t)', '\phi_y^1(t)', '\phi_z^1(t)', 1);

% plot up the second eigenfuction
eVecComps = reshape(squeeze(eVecs(:,2)),M,N);
%figure
subplot(3,1,2), plot([0:dt:dt*N-dt],eVecComps(1,:), 'ko:');
hold on
plot([0:dt:dt*N-dt],eVecComps(2,:), 'rs:');
plot([0:dt:dt*N-dt],eVecComps(3,:), 'b^:');
hold off title('Second
Eigenfunction')
xlabel('time')
ylabel('mode shape, \phi_i^2(t)')
legend('\phi_x^2(t)', '\phi_y^2(t)', '\phi_z^2(t)', 1);

% plot up the third eigenfuction
eVecComps = reshape(squeeze(eVecs(:,3)),M,N);
%figure
subplot(3,1,3), plot([0:dt:dt*N-dt],eVecComps(1,:), 'ko:');
hold on
plot([0:dt:dt*N-dt],eVecComps(2,:), 'rs:');
plot([0:dt:dt*N-dt],eVecComps(3,:), 'b^:');
hold off title('Third
Eigenfunction')
xlabel('time')
ylabel('mode shape, \phi_i^3(t)')
legend('\phi_x^3(t)', '\phi_y^3(t)', '\phi_z^3(t)', 1);

% Plot up r vector for each of the components

figure plot(r(1,:), 'ko:')
hold on plot(r(2,:), 'rs:')
plot(r(3,:), 'b^:')
title('original data') xlabel('time')
ylabel('x, y, z')
legend('x', 'y', 'z', 1)

% reconstruct the x, y, and z components
% define how many modes you want to reconstruct with (d)

d = 300;

% reconstruct the x component

new(1:N) = 0; new1(1:N) = 0; new2(1:N) = 0;

```

```

for i=1:d
    eVecComps = reshape(squeeze(eVecs(:,i)),M,N);
    eVecComps = eVecComps./((weights).^2);
    coeff(i) = (1/(N*N))*r(1,:)*(eVecComps(1,:))';    %(1/(M*N))*r(1,:)*eVecComps(1,:);
    new = new + coeff(i) * eVecComps(1,:);
end

for i=1:d
    eVecComps = reshape(squeeze(eVecs(:,i)),M,N);
    eVecComps = eVecComps./((weights).^2);
    coeff1(i) = (1/(N*N))*r(2,:)*(eVecComps(2,:))';    %(1/(M*N))*r(1,:)*eVecComps(1,:);
    new1 = new1 + coeff1(i) * eVecComps(2,:);
end

for i=1:d
    eVecComps = reshape(squeeze(eVecs(:,i)),M,N);
    eVecComps = eVecComps./((weights).^2);
    coeff2(i) = (1/(N*N))*r(3,:)*(eVecComps(3,:))';    %(1/(M*N))*r(1,:)*eVecComps(1,:);
    new2 = new2 + coeff2(i) * eVecComps(3,:);
end

%new = new/(max(new));
%r(1,:) = r(1,)/(max(r(1,:)));

figure subplot(1,2,1),plot(new);
title('reconstructed x');
axis('tight') subplot(1,2,2),plot(r(1,:))';
title('original x')
axis('tight')

figure subplot(1,2,1),plot(new1);
title('reconstructed y');
axis('tight') subplot(1,2,2),plot(r(2,:))';
title('original y')
axis('tight')

figure subplot(1,2,1),plot(new2);
title('reconstructed z');
axis('tight') subplot(1,2,2),plot(r(3,:))';
title('original z')
axis('tight')

figure title('original and reconstructed x, y, and z quantities')
subplot(1,3,1),plot((new),'ko:')
hold on plot(r(1,:),'bs:')
xlabel('time')
ylabel('x')

legend('reconstructed x ','original x',1)

%figure
subplot(1,3,2),plot((new1),'ko:')
hold on plot(r(2,:),'bs:')
xlabel('time')
ylabel('y')
%title('original and reconstructed y quantities')
legend('reconstructed y','original y',1)

%figure
subplot(1,3,3),plot((new2),'ko:')
hold on plot(r(3,:),'bs:')
xlabel('time')
ylabel('z')
%title('original and reconstructed z quantities')
legend('reconstructed z','original z',1)

figure plot(coeff.^2) axis([0 300 0 0.6])
title('POD Coefficients')

```

vs. Mode')