The Fractal Dimension Of Music

Perrin S. Meyer
Department of Applied Physics
Columbia University, New York, NY 10025

Abstract

Mandelbrot's fractal geometry has provided a new qualitative and quantitative approach for the understanding of the complex shapes of nature. In this paper, the fractal dimension of music waveform graphs is computed using the box-counting algorithm. It is shown that for the time scales studied, the fractal dimension of music is ~ 1.65, with very little variation for a wide range of styles of music.

1. Introduction

Fractals are geometric shapes with interesting properties that set them apart from normal Euclidean shapes. The first interesting property is that of self-similarity. This means that the geometric object is similar in both a qualitative and quantitative sense over many different scales. For example, if you zoom in on the edge of a circle, in the limit it would appear to be a line. However, as we will see, if you zoom in on a fractal such as the Mandelbrot set, It continues to be complex after literally millions of zooms. Another property of fractals is a non-integer dimension, which is related to the concept of self-similarity. The calculation of a fractal dimensions is an important way to classify objects that exhibit fractal characteristics.

In this paper, I use the box-counting algorithm to compute the fractal dimensions of music waveform graphs. The waveforms are plotted as normalized time-series data, and thus have a fractal dimension between 1 (a line) and 2 (a solid). The time scale used is 2 seconds. A wide variety of musical styles was analyzed, and the results are tabulated and discussed. A discussion then follows of the possible applications of the fractal dimension of music.

2. Background (what are fractals?)

Fractals are geometric shapes that have non-integer dimension and exhibit self-similarity [5]. A common way of generating fractals is to iterate non-linear difference equations. It is the non-linear aspect of this iteration that gives rise to the complex shapes of fractals.

One of the simplest difference equations that when iterated gives fractal behavior is the logistic growth equation. This equation arises from the differential form of a simple exponential growth equation with an added term to prevent unconstrained growth:

$$X_{n+1} = UX_n(1 - X_n)$$

This equation is iterated for U between 0 and 4. X_0 is given a starting value of 0.5, and the equation is iterated 150 times. The final 50 values of X_n is then plotted against U [Fig. 1]. We can see from the plot that until U=3 the function is stable, but then after U=3 the plot exhibits bifurcation's until finally it is goes into periods of deterministic chaos. This plot also shows self-similarity. A plot of the equation around one of its bifurcation's shows a qualitative similarity to the larger plot.

are

A now classic example of a fractal is the Mandelbrot set [1],[fig. 2]. This fractal is produced by iterating the non-linear difference equation:

$$Z_{n+1} = Z_n^2 + C$$

where Z is a complex number and C is a complex parameter. Z_0 is set equal to 0.0, and the equation is iterated. If the point is bounded, a point is plotted; if not, the point is left blank. It is traditionally plotted for C between -2 and 2. This simple equation gives rise to a very beautiful and complex fractal, and its self-similarity can be seen by zooming in on a piece of the "coastline" [fig. 3].

3. Fractal Dimension

Imagine trying to measure the coast of England. If you used a map and a ruler, you would get one estimate. If you went to England and walked around the whole island counting your steps, you would get a much greater estimate because you could walk around every bay and inlet. This brings up the concept of fractal dimension. For fractal objects that exhibit self-similarity, one gets a different estimate of "length" depending on the scale of the ruler one is using. A convenient way to look at dimension is to use the Equation:

$$N(\varepsilon) = \frac{1}{\varepsilon^{D}} \quad (1)$$

Here N(e) stands for the number of measuring sticks used to measure a certain object, e is the length of the measuring stick and D is the dimension of the object. (note that I am using e as epsilon, not e=2.7...). This equation shows that if we were to measure a certain object with a number of different size measuring sticks, and count the number needed, We could solve for the fractal dimension, D. This leads to the box-counting algorithm for the determination of fractal dimension [2].

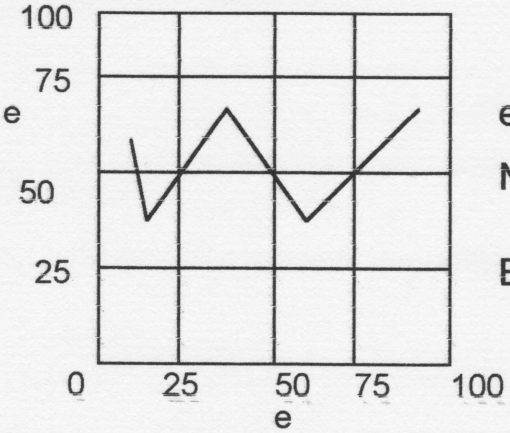
4. The Box-Counting Algorithm

If we take the log of both sides of equation (1), we get:

$$Log(N(\varepsilon)) = -D \times Log(\varepsilon) \quad (2)$$

If we plot the log(N(e)) verses log(e), the slope of the line is the negative dimension of the object. This is the basis of the box-counting algorithm [2]. Recorded music is time-series data, and can be plotted as such [fig 3]. To use different size "measuring sticks" we overlay a grid of various size spacing on the data and count the number of grid-boxes that are touched by the data. The data is normalized so it is has the same horizontal and vertical scale [fig 3,4].

fig 3



e=25

N(e)=5

Example of box-counting algorithm.

Then log(N(e)) is plotted verses log(e), and the negative slope of the line is the fractal dimension of the waveform. A straight line has a dimension of 1, and a solid has a dimension of 2, so the music waveform graph will have a dimension between 1 and 2. For example, in [fig. 5], the counts of various different grid sizes e are shown, and the line of log(N(e)) Vs, log(e) is shown giving a dimension of ~1.65. Least-squares regression is used to find the slope of the line.

5. Experimental Setup

Data acquisition and analysis was done on an IBM-PC compatible personal computer with additional hardware for music sampling. The music was sampled off a Compact Disc player at 11,025 samples per second. The analysis was done on 20,000 data points, which is approximately 2 seconds of music. The analog-to-digital board was 8-bit, which gives a dynamic range of about 48 dB. 8-bit conversion means that the analog waveform is broken into 256 discrete levels, which were then normalized [fig 4].

Analysis of the data was performed using algorithms written in the matrix language Matlab. C was used where extra speed was required, such as in the computation intensive box-counting algorithm.

6. Data and analysis

Fractal dimension calculations on a wide variety of styles of music show that the fractal dimension D is roughly invariant, for this sampling rate and resolution.

Results

Music Sample	**Dimension**
Random Noise	**1.95**
1 kHz Measured Sin Wave	**1.82**
1 Khz Generated Sin Wave	**1.81**
Grateful Dead	**1.64**
Vivaldi	**1.65**
Elliot Sharp	**1.60**
English Beat	**1.65**
Sri Lankan	**1.68**
Gin & Tonic (Jazz)	**1.69**
Chariots of Fire	**1.68**
Line	**1.01**

The fractal dimension of the music samples is centered at D=1.65 and varies between a high of 1.69 and a low if 1.60. For comparison and testing, the box-counting algorithm was also used on some synthetic data sets. The 20,000 point line [fig 6] gives the expected value of 1 to within experimental accuracy. Music is obviously not just random noise, and this is supported by the data, as a random noise sample had a much higher fractal dimension than the music samples. A 1kHz sine wave also had a higher fractal dimension fig[7].

For comparison, the power spectral density was also computed and is show in [fig 7]. The power spectral densities look similar for the music samples and the random noise, while the fractal dimension estimates are much different.

Embedded in the concept of fractal dimension is self-similarity. Self-similarly in music could come from the highly complex rhythms and melodies that exist over different time scales. This could explain the correlation of the fractal dimension for many different styles of music.

7. Applications

There are many areas of signal processing that could benefit from the inclusion of fractal dimension in their implementations.

For example, a common way to synthesize music on computers is FM modulation, which works by modulating sine waves of various frequencies with other sine waves. However, this method produces music that sounds unnatural. Fractal generating functions that produce sound with a dimension ~ 1.6 could be added to the FM modulation to enhance the realism of the sound.

?

High fidelity music in the digital domain takes up a lot of storage space. For example, 1 hour of 16-bit digital music takes up 640Mb of hard disk space. Many techniques exist to compress data, but one based on Linear Fractal Interpolation shows great promise [3]. Most algorithms in common use can compress data about 4:1, while with Linear Fractal Interpolation, Mazel was able to get a 16:1 compression ratio on certain waveforms. The great obstacle in this method is the tremendous computational burden required to compute the fractal parameters. If it can be shown that the dimension of music is invariant under much greater time scales than researched here, the computational burden would be greatly reduced.

The fractal dimension of music waveforms can also be used to set the parameters for non-linear digital filtering. Harris [4] concludes that the "technique of using the fractal dimension of input data to adapt a filter's characteristics is a viable one".

8. Conclusion

The fractal dimension of music on the time scale of 2 seconds is equal to ~1.65 and is invariant for a wide range of musical styles. This is interesting in and of itself, and it shows the need for more research to see if this invariance holds for a wider range of time scales. The potential applications of the fractal characterization are numerous. This fractal dimension characterization will hopefully be used as a supplement to more traditional means of signal analysis in order to gain more understanding of the fundamental properties of music.

References

[1] B. Mandelbrot, *The Fractal Geometry of Nature,* Freeman,San Francisco (1982).

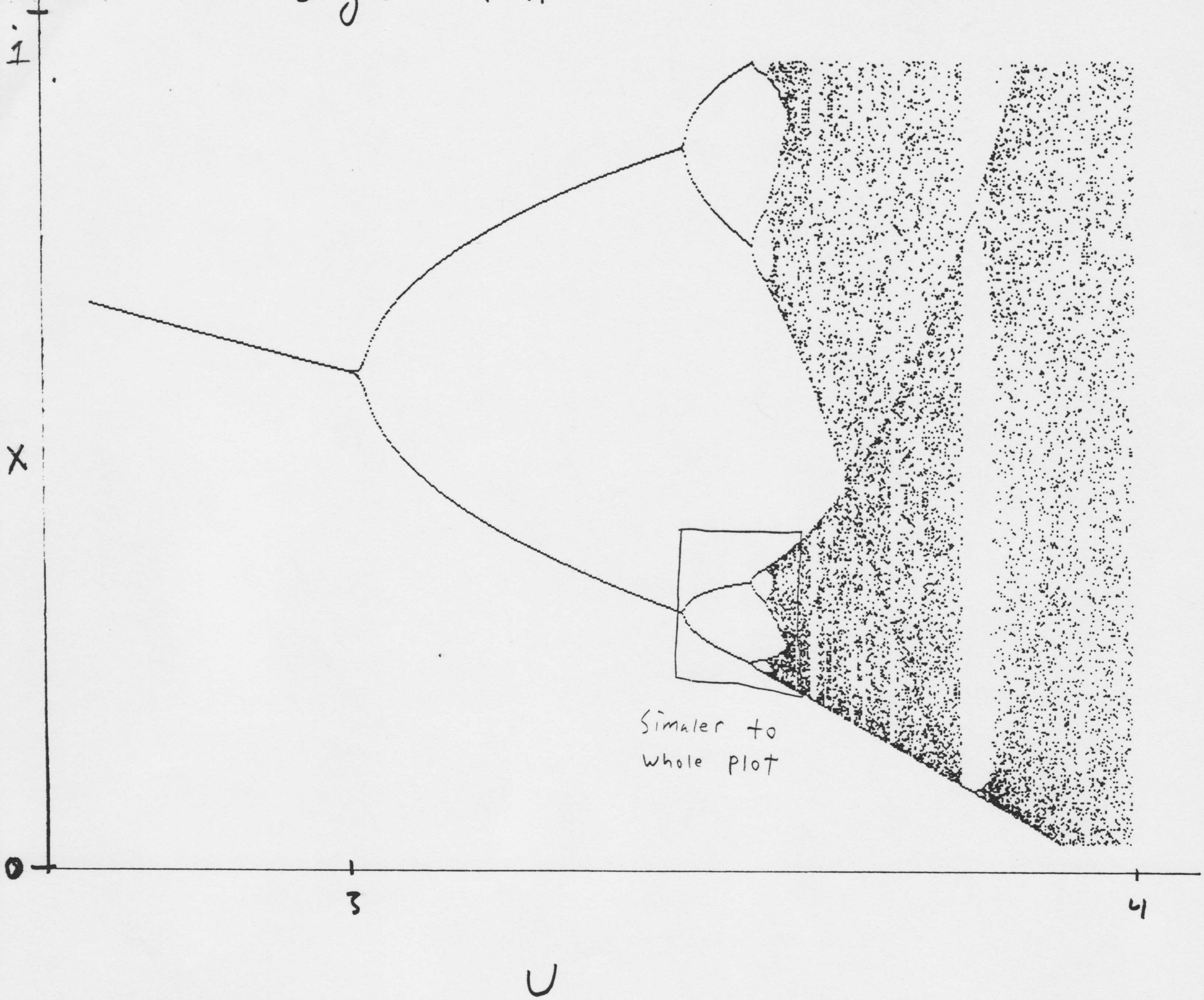
[2] C. Pickover, "Fractal Characterization of Speech Waveform Graphs" in *Computers and Graphics* (1986) vol. 10 pp. 51-61.

[3] D. Mazel and M. Hayes, "Fractal Modeling of Time-Series Data", in *Proc. 23rd Annual Asilomar Conf. on Signals,Systems and Computers* (1989) Maple Press, San Francisco, pp. 182-186.

[4] F. Harris and D. King, "Adaptive Filters Using Fractal Dimension of Data", in *Proc. 24rd Annual Asilomar Conf. on Signals,Systems and Computers* (1990) Maple Press, San Francisco, pp. 278-282.

[5] M. Barnsley, *Fractals Everywhere*, Academic Press, Inc. New York,(1988).

Logistic Map



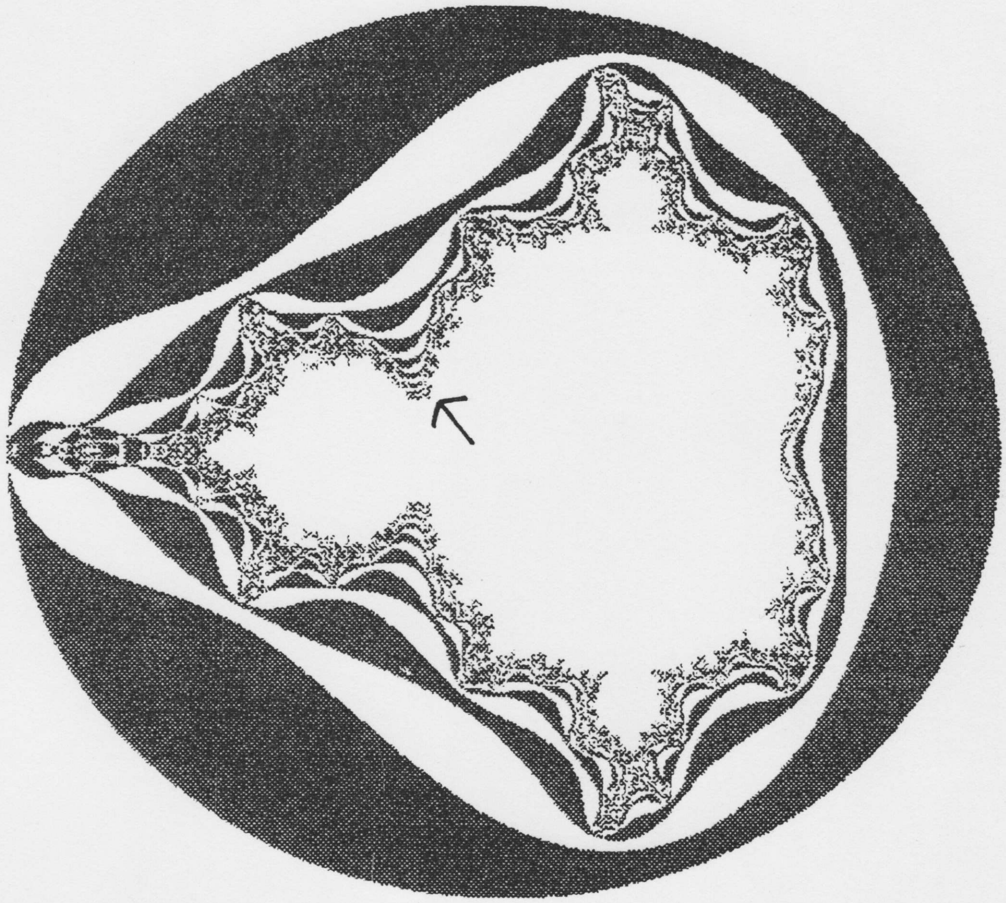
$$X_{n+1} = U X_n (1 - X_n)$$

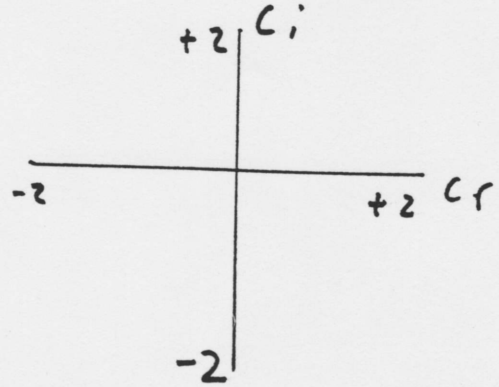
$X_0 = .5$

150 Iterations

[Fig. 1]

Mandelbrot Set

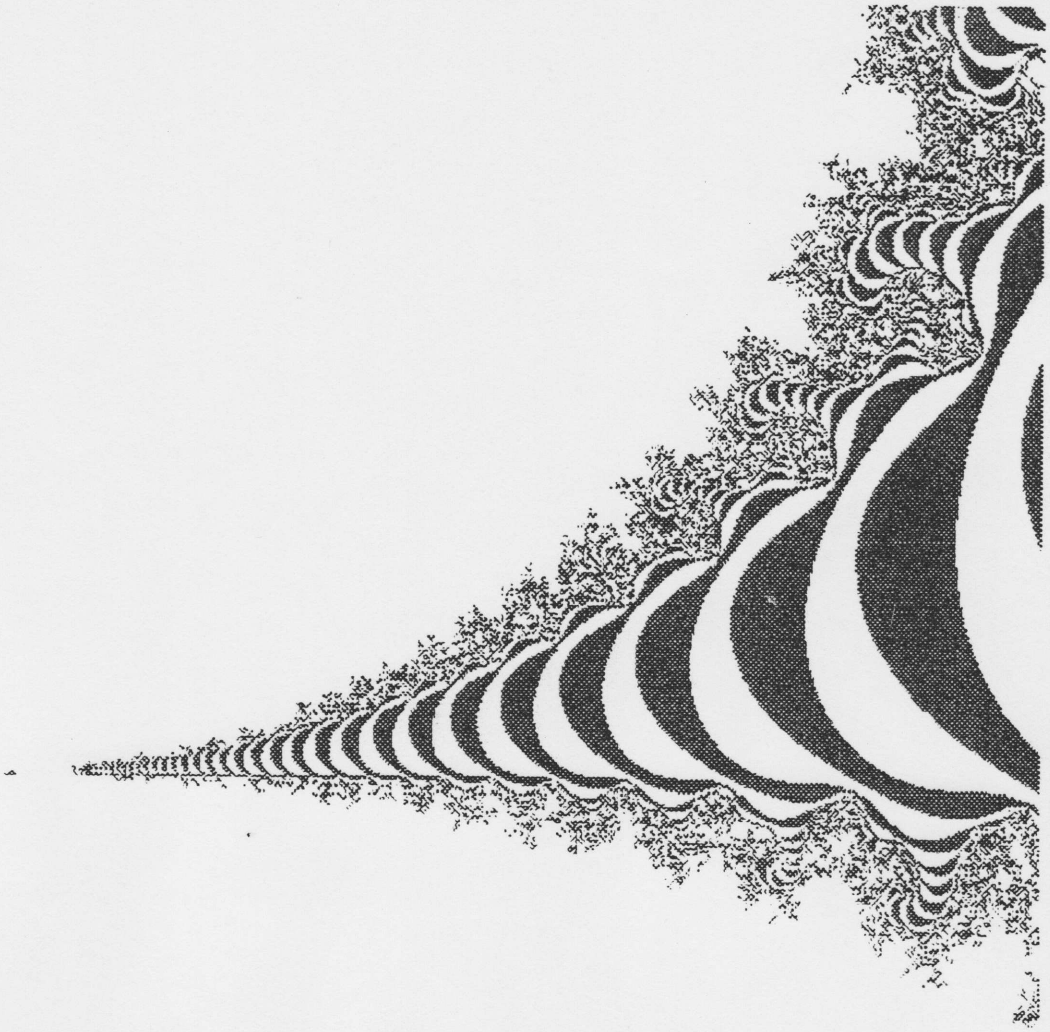




$$Z_{n+1} = Z_n^2 + C$$

$$Z_0 = 0$$

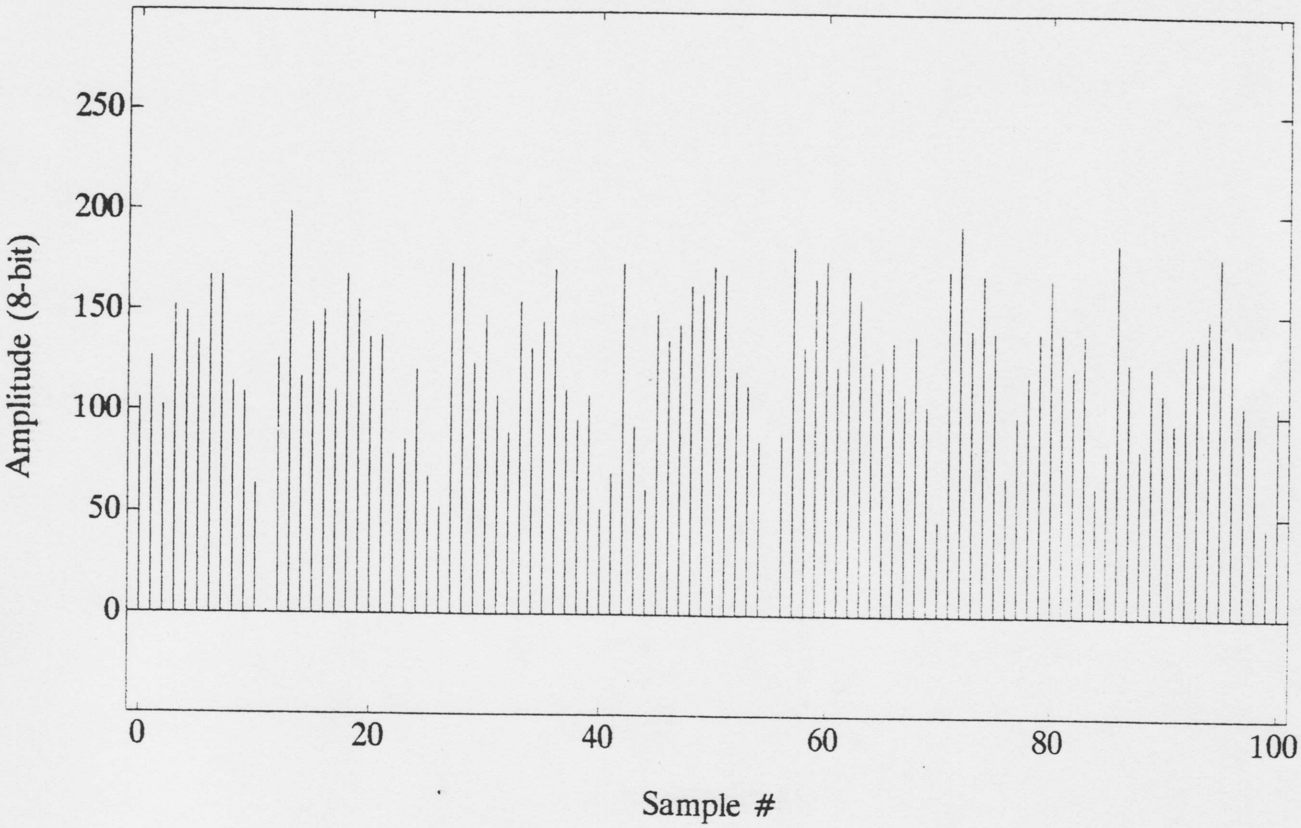
[Fig. 2]



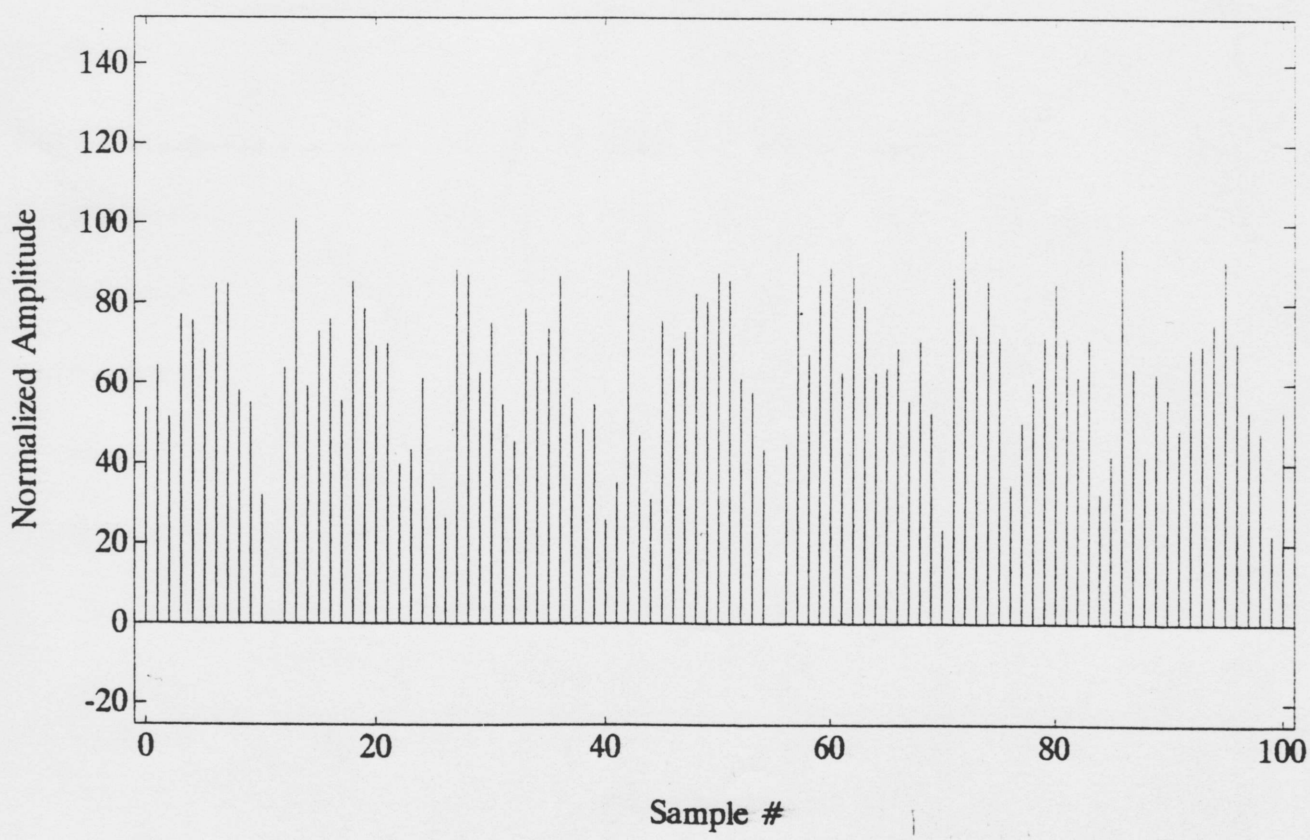
Zooming in on
Mandelbrot set

[Fig. 2B]

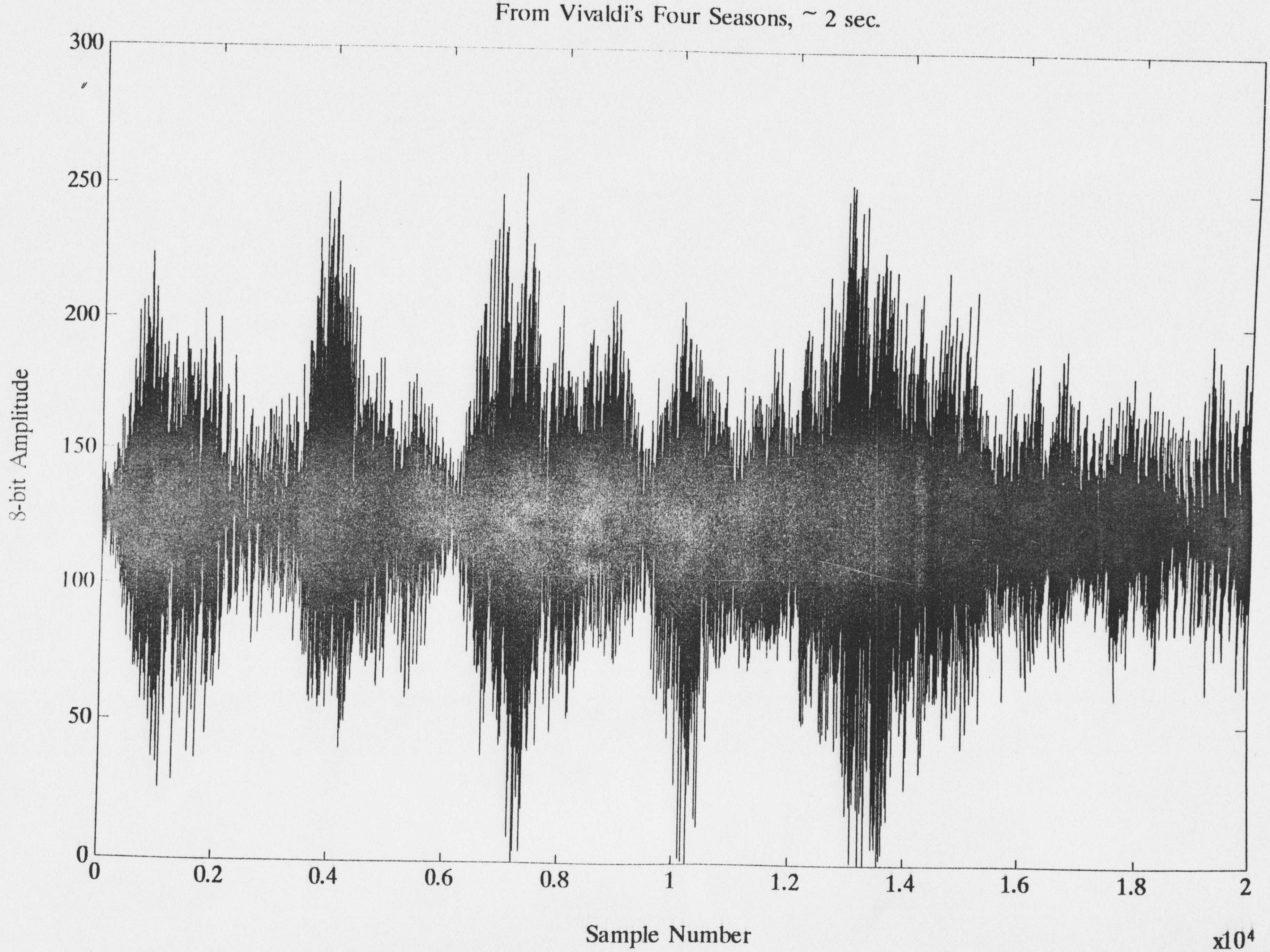
100 samples from Vivaldi's Four Seasons



100 samples from Vivaldi's Four Seasons, Normalized to 100

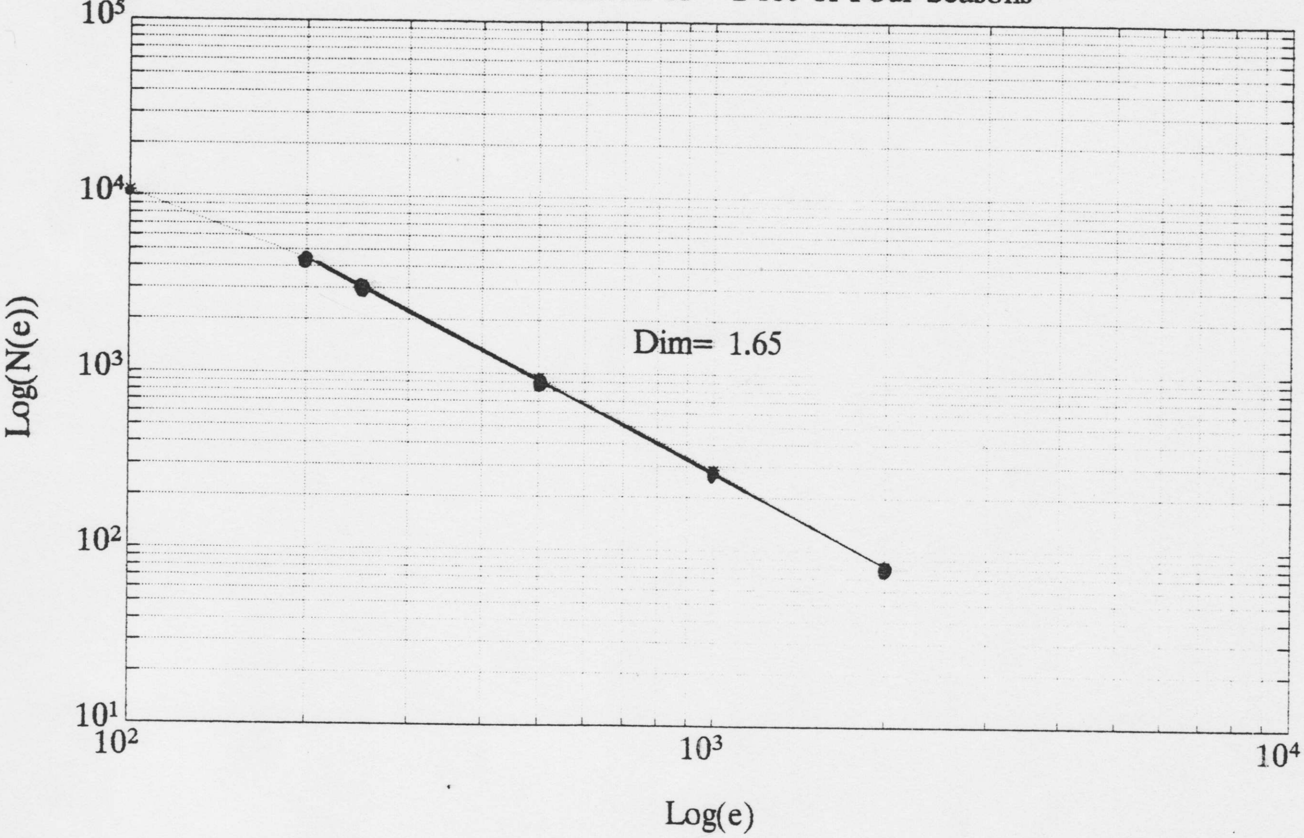


[Fig. 4A]


From Vivaldi's Four Seasons, ~ 2 sec.
8-bit Amplitude
300
250
200
150
100
50
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Sample Number
x10⁴

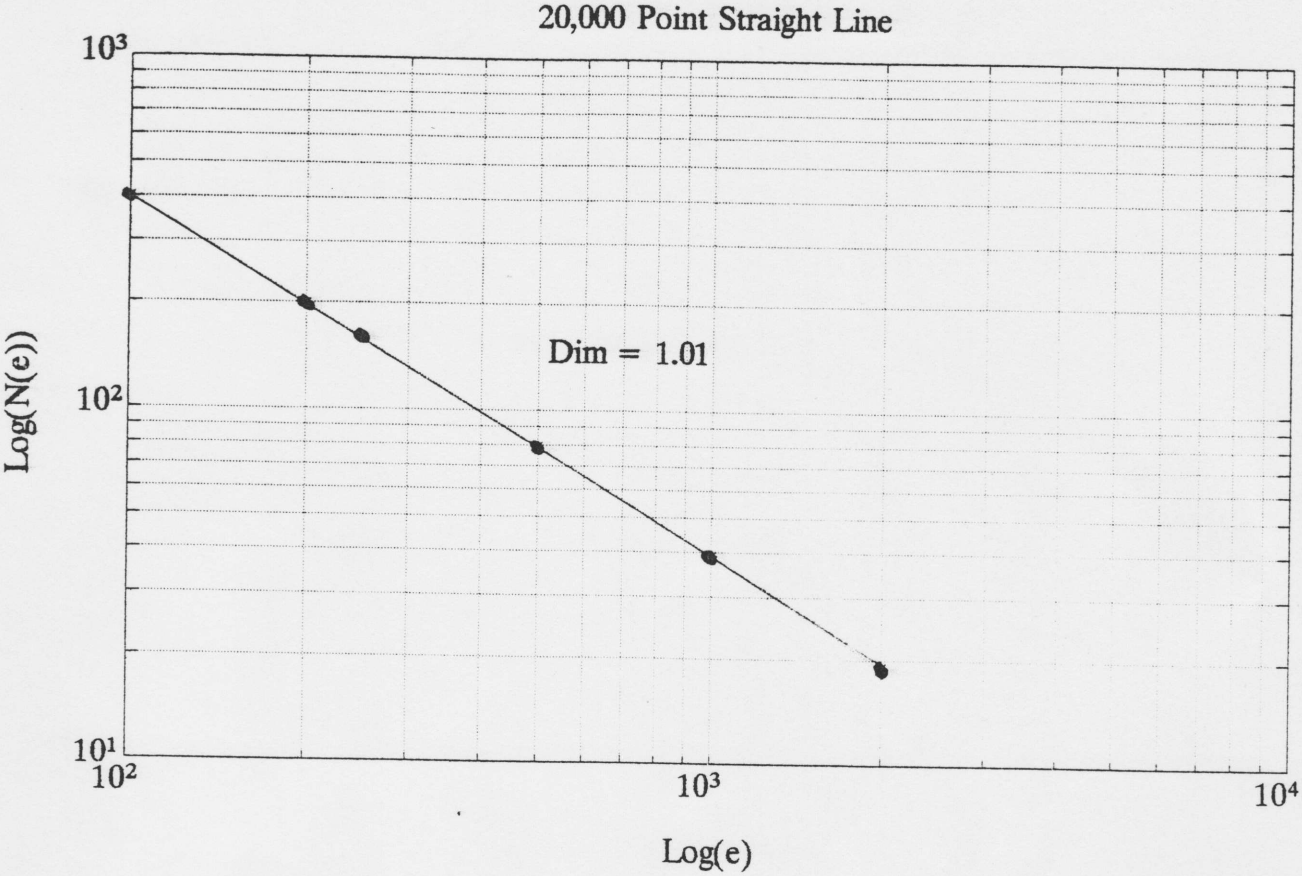
[Fig. 4B]

Fractal Dimension of ~ 2 sec of Four Seasons

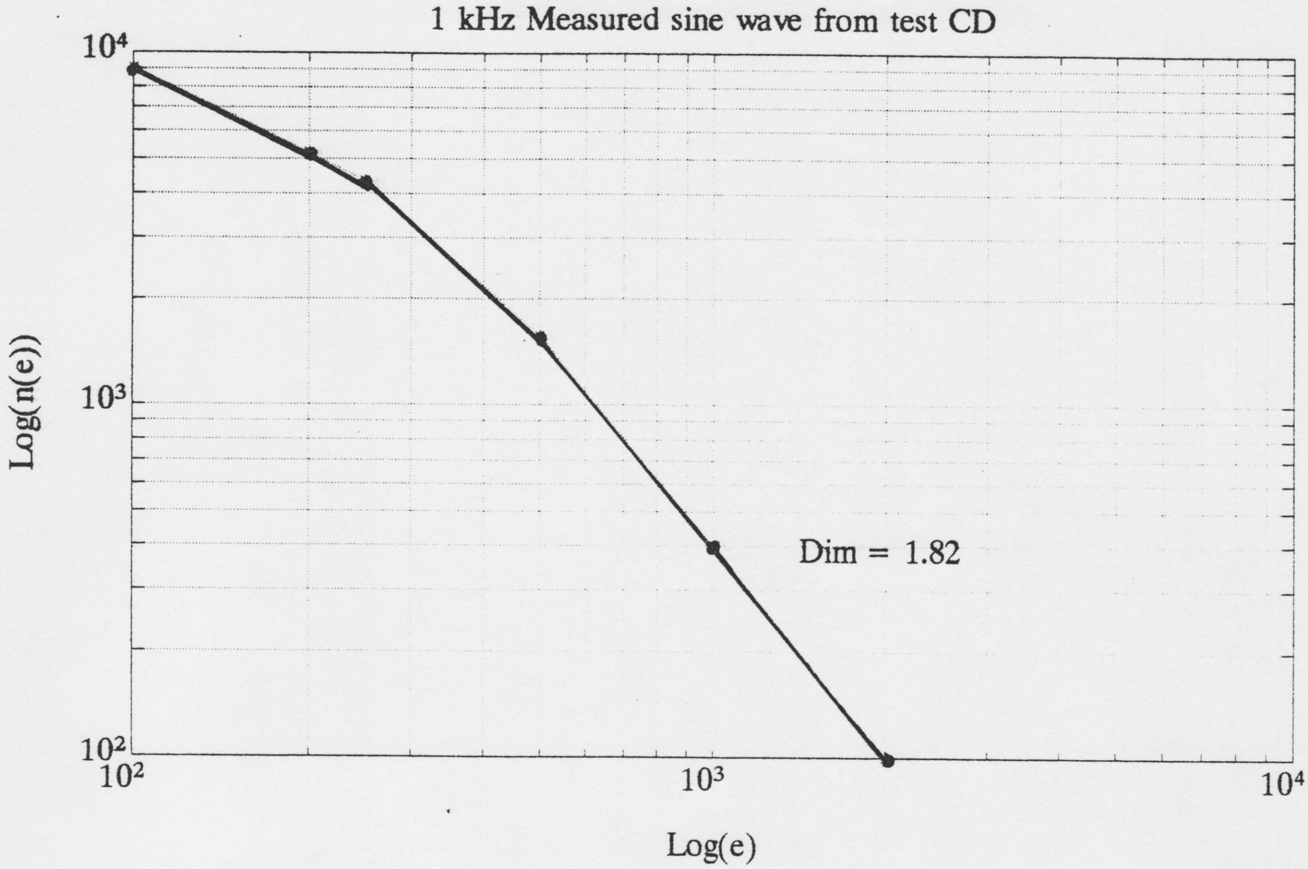


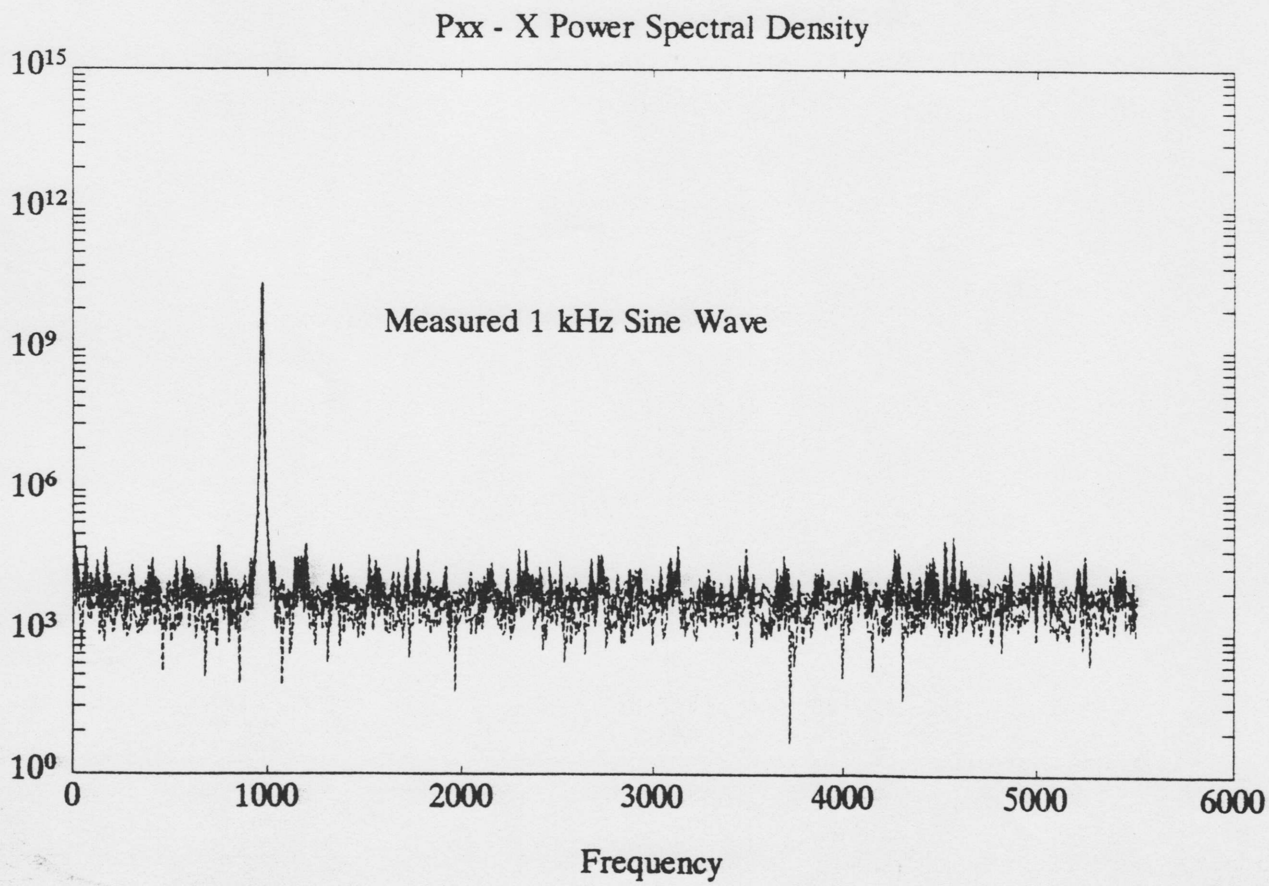
e	N(e)
100	10720
200	4461
250	3131
1000	288
2000	83

[Fig. 5]

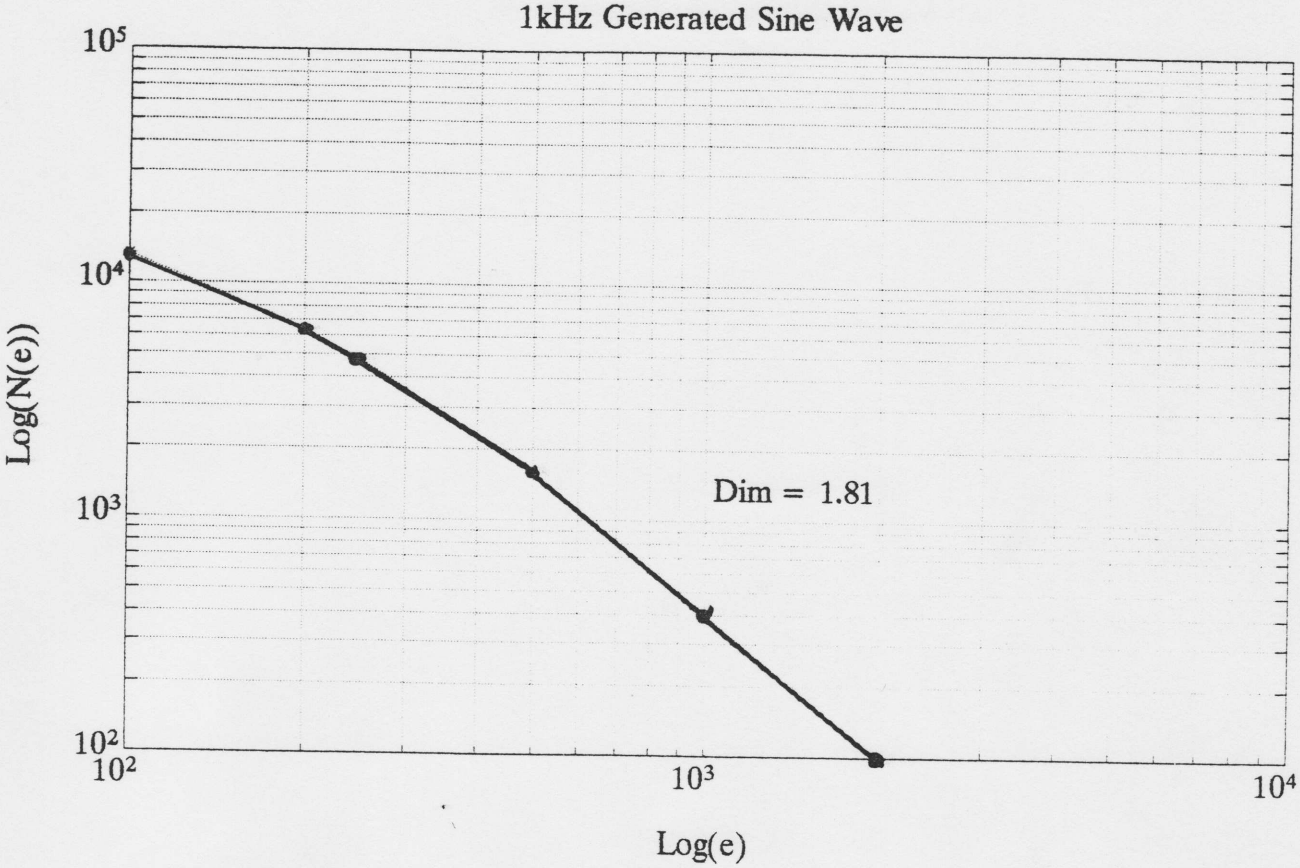

20,000 Point Straight Line
10³
10²
10¹
Log(N(e))
Dim = 1.01
10²
10³
10⁴
Log(e)

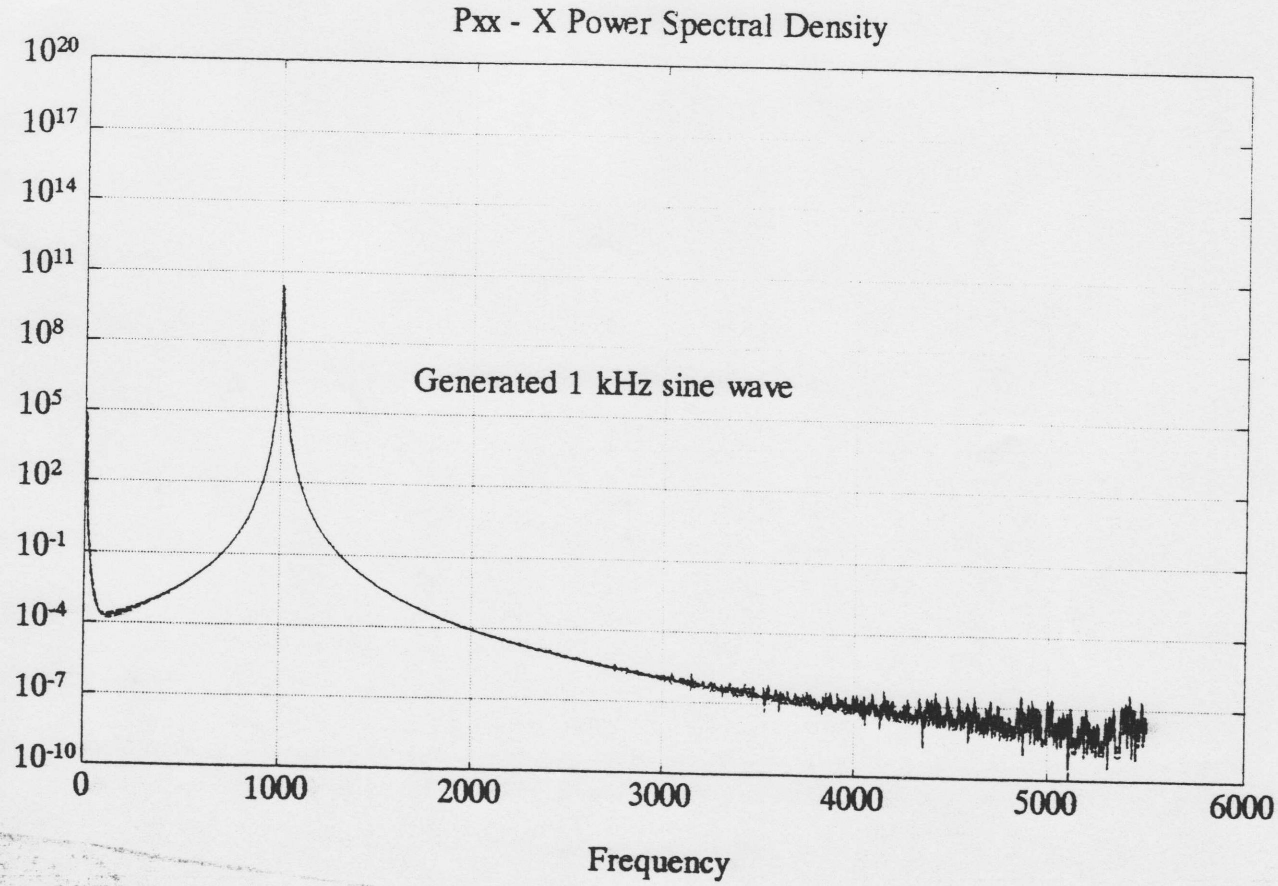
[Fig. 6]


1 kHz Measured sine wave from test CD
10⁴
10³
10²
Log(n(e))
Dim = 1.82
10²
10³
10⁴
Log(e)

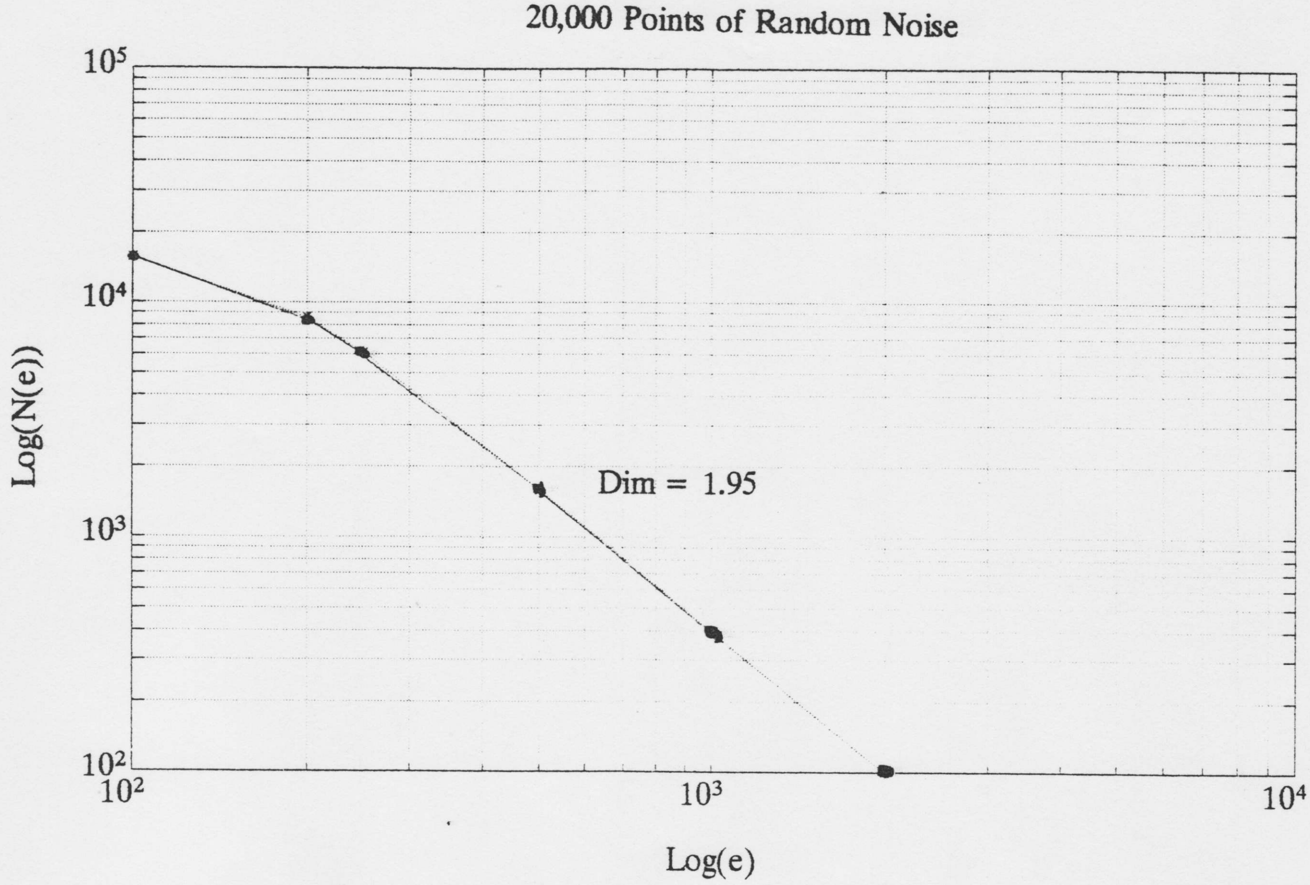

Pxx - X Power Spectral Density
10¹⁵
10¹²
10⁹
10⁶
10³
10⁰
Measured 1 kHz Sine Wave
0
1000
2000
3000
4000
5000
6000
Frequency

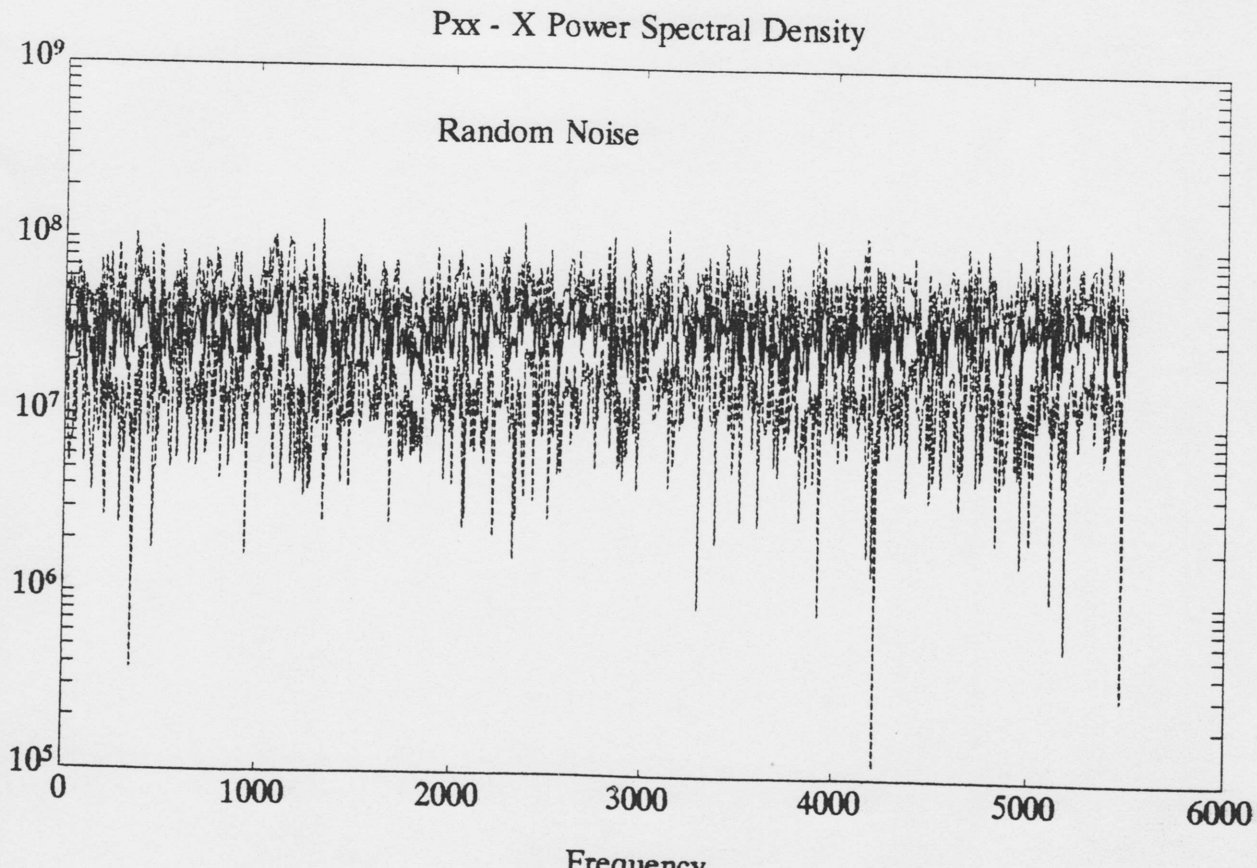
[Fig. 7A]


1kHz Generated Sine Wave
Log(N(e))
Dim = 1.81
Log(e)

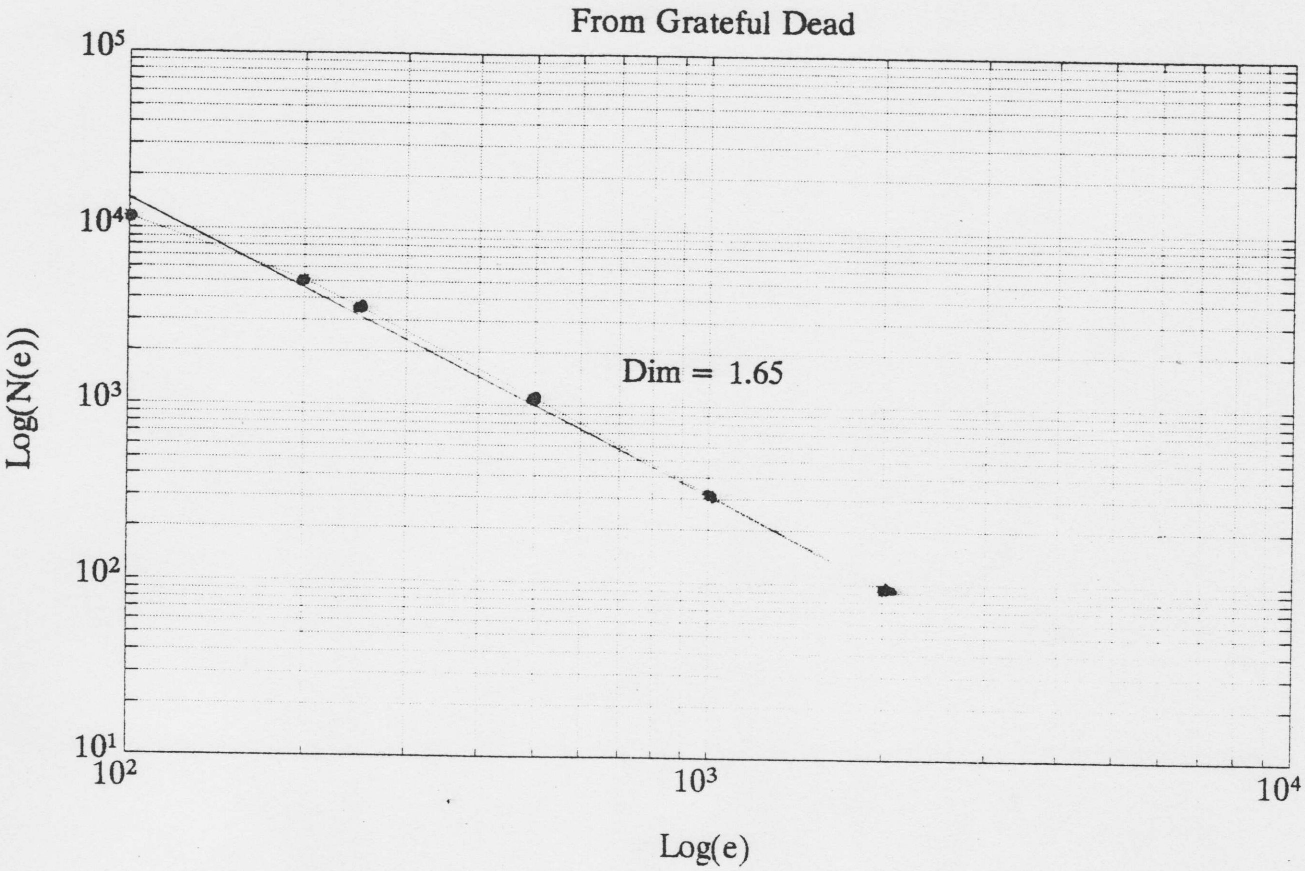

Pxx - X Power Spectral Density
Generated 1 kHz sine wave
Frequency

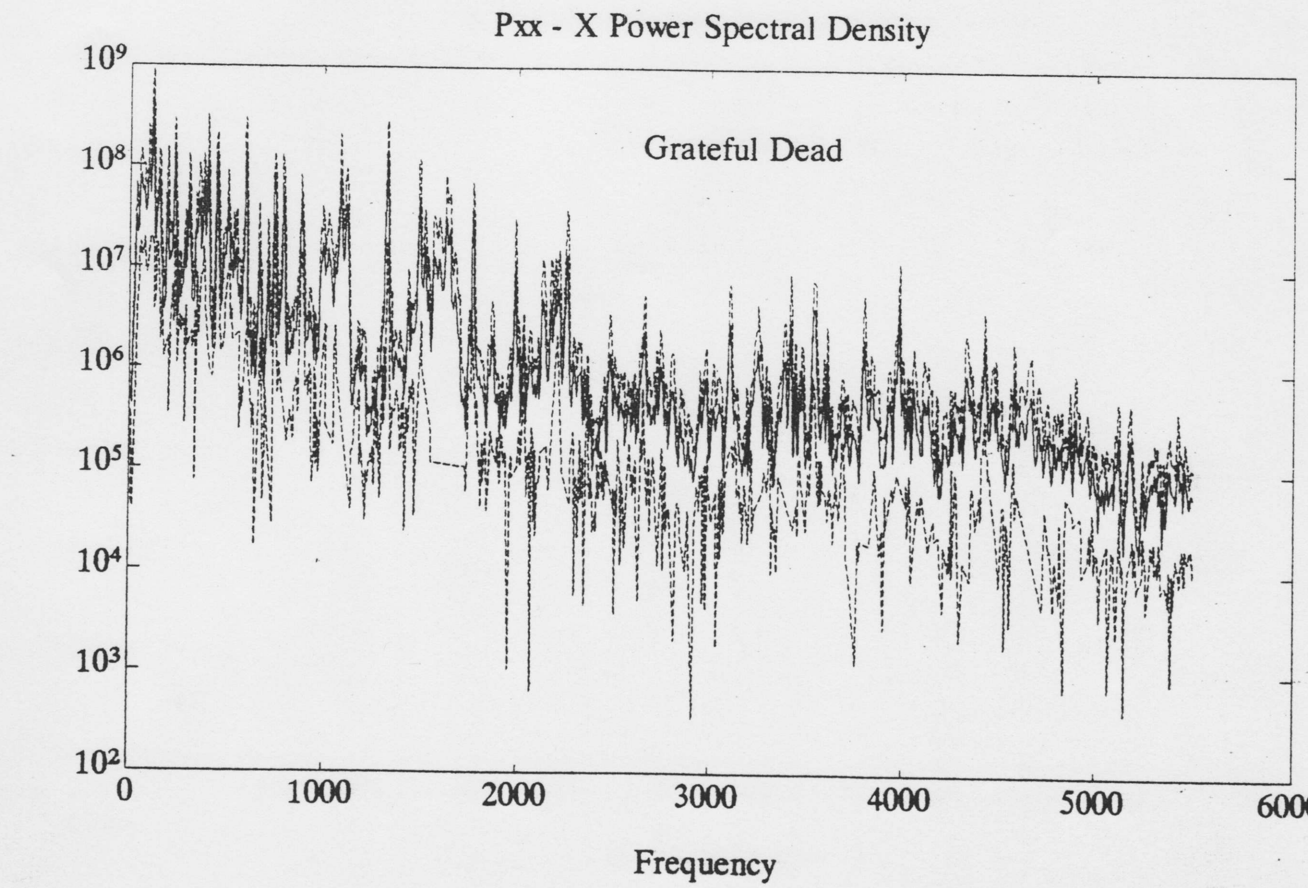
[Fig. 7B]


20,000 Points of Random Noise
10^5
10^4
10^3
10^2
Log(N(e))
Dim = 1.95
10^2
10^3
10^4
Log(e)

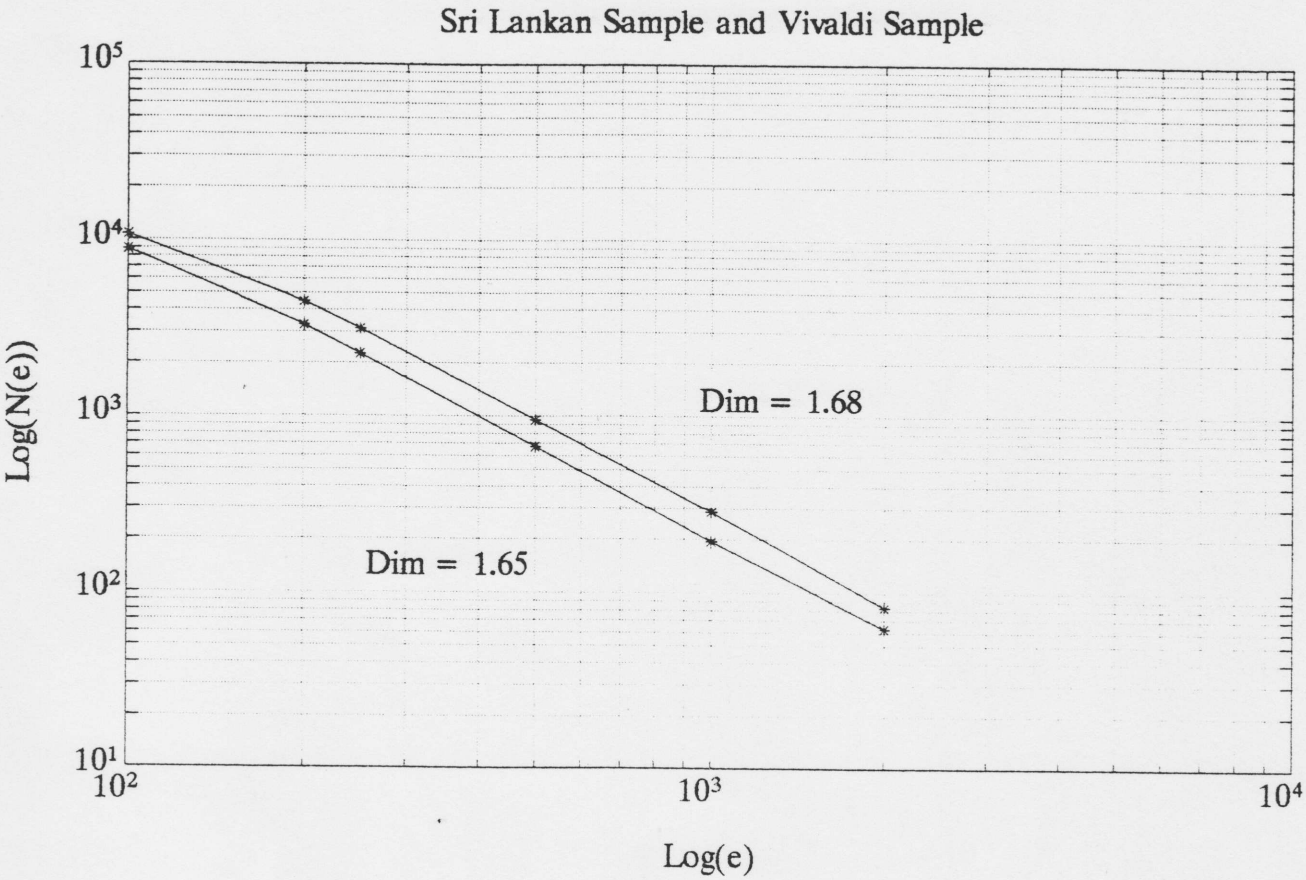

Pxx - X Power Spectral Density
Random Noise
10^9
10^8
10^7
10^6
10^5
0
1000
2000
3000
4000
5000
6000
Frequency

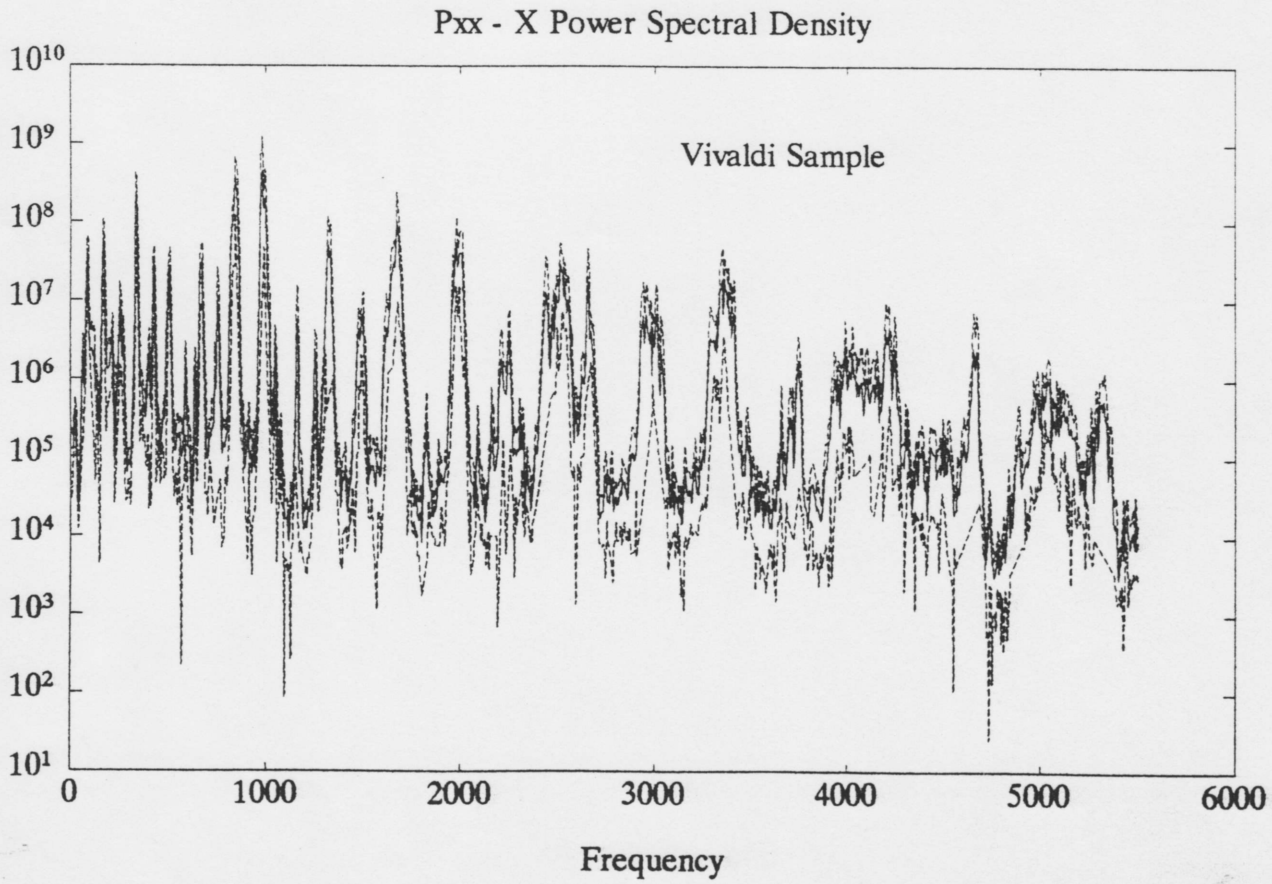
[Fig. 7C]


From Grateful Dead
Dim = 1.65
Log(N(e))
Log(e)
10^5
10^4
10^3
10^2
10^1
10^2
10^3
10^4

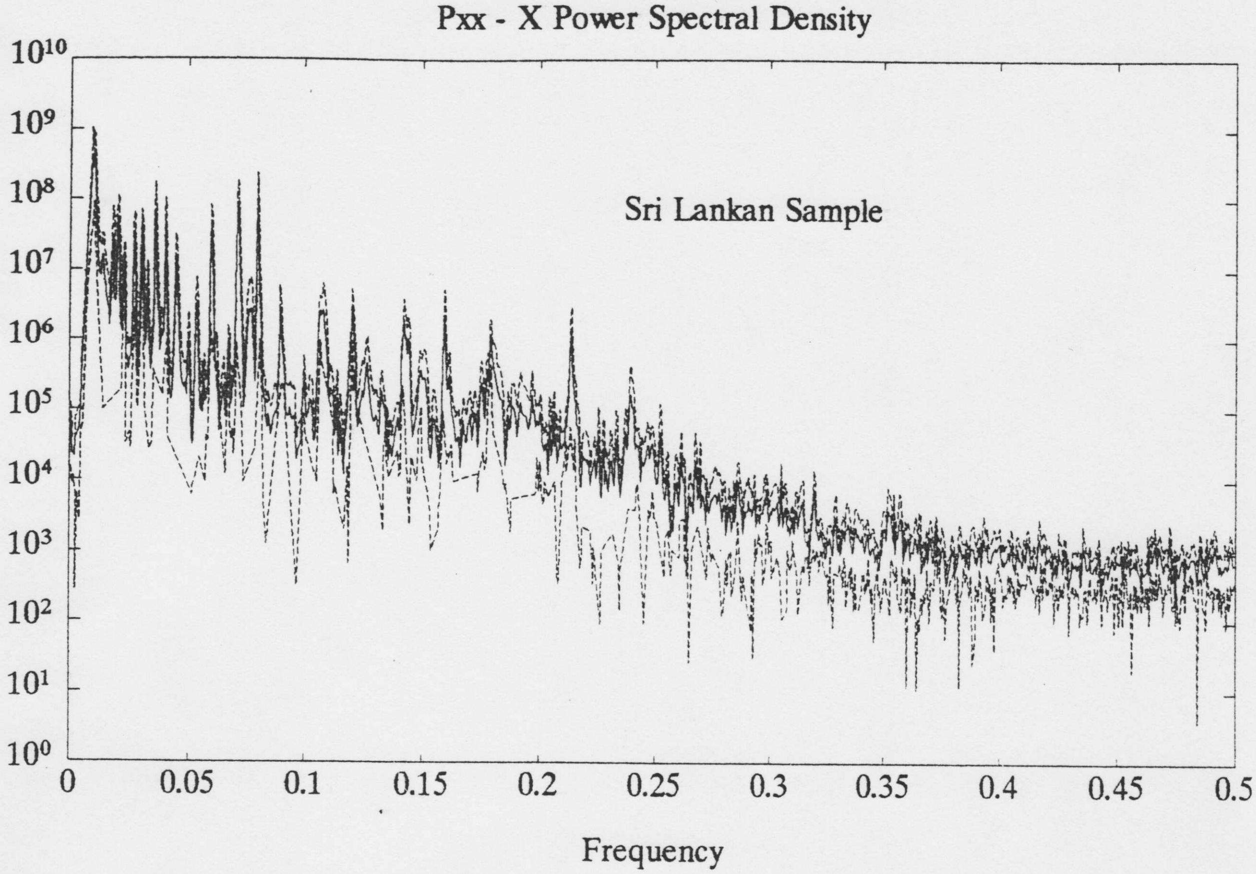

Pxx - X Power Spectral Density
Grateful Dead
Frequency
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
0
1000
2000
3000
4000
5000
6000

[Fig. 7D]


Sri Lankan Sample and Vivaldi Sample
Log(N(e))
Log(e)
Dim = 1.68
Dim = 1.65
10^5
10^4
10^3
10^2
10^1
10^2
10^3
10^4


Pxx - X Power Spectral Density
Vivaldi Sample
Frequency
10^10
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
0
1000
2000
3000
4000
5000
6000

[Fig. 7E]


Pxx - X Power Spectral Density
Sri Lankan Sample
10^10
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Frequency

[Fig 7.F]