

Determining the Fractal Dimension of Crumpled Paper using Chi-Squared Data Analysis and Model Testing

By Larry Hui

Author Bio:

Larry Hui is a current high school student at Crescent School located in Canada who would like to pursue Mechanical Engineering with a concentration in Aerospace and Aeronautical Engineering. He is a beginner researcher who, under the guidance of Dr. Carey Witkov, has just recently completed one paper regarding the fractal dimensions of crumpled paper using chi-squared model testing. Outside of his academic interests, he is a holder of a private pilot's licence and working on both his night and instrument ratings. He likes to read short fiction and classics and spend time playing classical music on the trumpet and the piano.

Abstract

Most real objects in the natural world are fractals (i.e., non-smooth). Fractal dimensions are non-integer and therefore, by example, may lie between that of a plane (two-dimensional) and a solid object (three-dimensional). Determining the dimension of fractal objects can help describe objects for which suitable geometric representations may be unavailable. The purpose of this study is to determine the fractal dimension of crumpled paper using chi-squared data analysis and model testing. This offers several improvements over linear regression including better parameter estimation, better estimation of parameter uncertainties and model testing.

Crumpled paper does not fully occupy the space it is embedded in, so its fractal dimension is estimated to be between that of a plane and a solid sphere (i.e., between 2 and 3). Data were obtained from photographs of crumpled paper of various masses as the independent variable and the diameter of each crumpled ball was repeatedly measured for the dependent variable. Chi-squared data analysis of a two-parameter linear model gave a best-fit slope estimate of the fractal dimension as 2.17, and chi-squared model testing showed that the model should not be rejected.

Keywords: Fractal, Geometry, Dimension, Chi-Squared, Regression, Estimation, Paper, Crumpled

Nomenclature

χ^2	Chi-squared test statistic
m	Relative mass of crumpled paper ball
ρ	Density of the crumpled paper ball
R	Radius of the crumpled paper ball
n	Fractal dimension of crumpled paper ball
A_{Best}	Best fit slope parameter
B_{Best}	Best fit y-intercept parameter
$P(x)$	Probability density function of the Gaussian
$P(\chi^2)$	Probability density function of the chi-squared for continuous variables
SE	Standard error of the mean
N	Number of data points on the plane
σ	Standard deviation
μ	Population mean
$\bar{y}$	Mean radii
$\bar{x}$	Mean relative mass
$SE_{\bar{y}}$	Standard error of the mean radii

Section I: Introduction to the Study

The problem in adequately describing shapes, forms of curves and surfaces has been a contentious and thought-provoking matter for hundreds of years. Euclidean geometry fails to adequately represent many real-world objects, for example, the coastline of a country is neither straight nor elliptical. Imagine attempting to describe rough or uneven terrain in terms of classical geometry—it cannot easily be done; if at all. However, in 1979, the French polymath Benoit Mandelbrot proposed the interesting idea of using a single number as a measure of roughness, a statistical complexity showing how complicated a self-similar object is, which he called the “fractal dimension” of an object. This idea was used to alleviate the vagueness caused by the absence of any suitable

geometric representations (Mandelbrot, 1980). A fractal dimension characterizes curves and surfaces in terms of their complexity by treating dimension as a continuum and not a discrete value.

Normally, dimensions belong to the positive set of integers, Z^+ ; for example, the 1st dimension is a line, the 2nd dimension is a plane, and the 3rd dimension is a real-world object (Devaney, 1995). However, fractal dimensions are non-integer and may lie between the aforementioned examples. Fractal dimension, therefore, is a measure of space-filling and can be related to other physical properties, e.g., rigidity, which is relevant to crumpled paper.

Although knowledge of fractal properties has been around since the beginning of the 20th century, Mandelbrot was the first to recognize their applications outside of mathematics and related them to economics, chaos theory, statistical physics and more (Mirowski, 1990). This paper aims to obtain the fractal dimension of crumpled paper using a chi-squared algorithm written in Python. The main result of this paper is that chi-squared analysis improves on ordinary linear regression for determining fractal dimensions by including measurement uncertainties and model testing in the algorithm.

A. Inquiry Statement, Research Approach, and Hypothesis

Using chi-squared analysis, we can determine if the derived family of models should be rejected or not. If the model is not rejected, how do we determine the fractal dimension of crumpled paper? Based on the fact that the crumpled paper does not fully occupy the space it is embedded in, a general hypothesis we could assume is that the fractal dimension of crumpled paper should be between 2 (a plane) and 3 (a solid object). Indeed, the fractal dimension that was found (2.2) is between 2 and 3.

The research approach used for the chi-squared analysis is as follows: (i) derive the model, perform the experiment, (ii) determine the best-fit parameters, (iii) test if the best-fit is a good fit, (iv) find the uncertainties on the parameters, (v) test if the model should be rejected, and lastly, (vi) determine

if there should be a revised and improved model. The following chi-squared analysis in Eq. 1 was used to perform the model testing using chi-squared for continuous variables as shown below:

$$\chi^2 = \sum_{i=1}^n \frac{(y_i - y_m)^2}{\sigma_i^2} \quad (1)$$

While this resembles a weighted least squares, where the contribution of each data point is inversely weighted by its uncertainty, weighted least squares, like ordinary least squares, do parameter estimation but not model testing. Model testing usually involves the use of ad hoc methods (e.g., correlation coefficients, F-tests, confidence intervals, etc.) whereas chi-squared analysis combines parameter estimation and model testing in one methodology (Witkov & Zengel, 2019). The chi-squared test was not done by hand but instead done in a program written in Python. An explanation of what the program does and how it computes the chi-square value will be explained later in this paper. So we ask the question: why use chi-squared analysis rather than a linear regression?

Using chi-squared to test two quantitative variables is not seen as often as two categorical variables since the chi-squared test usually involves testing models which include the number of Observed (O) and Expected (E) occurrences of certain event(s) as evident in Eq. 2.

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

This is not, however, the form of chi-squared used for testing continuous models. Chi-squared is used to assess our model because it can be shown that it provides optimal estimates of model parameters for normally distributed errors; in this case, these are—slope A and y-intercept B . In using chi-squared, we should be asking the question, given the model, how probable is the data? This leads to the definition of chi-squared, which, when minimized, provides a maximum likelihood estimate of the best slope and y-intercept. As shown in **Appendix A**, the best-fit parameters are those that minimize chi-squared. This is because the minimum value of chi-squared corresponds to the maximum likelihood that, given the model, the data would be generated.

The most important feature of chi-squared curve-fitting versus Gomes' linear regression method (Gomes, 1987) is that chi-squared, by its definition, includes measurement uncertainties (note that the square of the standard error appears in the denominator of chi-squared). For example, a good fit to the data could be a bad fit to the model. A good model fit requires that the minimum value of 2 should be of the order of N , the number of data points. This ensures that the weighted sum of squares between the data and model is within one standard error (Witkov & Zengel, 2019). By including uncertainties, chi-squared is able to combine parameter estimation and model testing in one consistent methodology.

B. System Model and Derivation

For solid objects, we assume a power-law relationship between mass and radius of the crumpled paper, but allow for fractional exponents as the crumpled paper is a fractal object. In the following power-law relation, m is the mass of the paper, R is the radius, ρ is the density, and n is the fractal dimension:

$$m = \rho R^n \quad (3)$$

To solve for the fractal dimension of the crumpled paper, we need to solve Eq. 3 for n . Taking logarithms on both sides of the equation, we obtain:

$$\log \log m = \log \log \rho + n \log \log R \quad (4)$$

Rearranging for $\log R$, note that the equation is a linear function in the form $y = m(A)x + B$.

$$\log \log R = \frac{1}{n} \log \log m - \frac{1}{n} \log \log \rho \quad (5)$$

Eq. 5 is now in the form of the following two-parameter linear model

$$y = A_{Best} x + B_{Best}, \text{ where } n = \frac{1}{A_{Best}} \quad (6)$$

While the mass of the crumpled paper is a function of its radius, it is convenient to express the model in Eq. 5 using radius ($\log \log R$) as the dependent variable (y) and mass ($\log \log M$) as the independent variable (x).

Section II: Experimentation and Data Analysis

To perform this experiment, we used the crumpled paper image in Fig. 1 taken from Yale University's Fractals website, which involves two square sheets each of 8 by 11-inch paper with relative masses of 1, $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$, respectively (Yale, n.d). The raw data was extracted using the online geometry software GeoGebra on a photo of four crumpled balls of different masses. Using GeoGebra, we estimated the diameters of the crumpled paper by taking three measurements: the longest diameter, the shortest diameter, and one in between to reduce sampling variability.

Figure 1 Paper balls of relative mass used for experiment.



Table 1: Raw Data

Crumpled Ball	Relative Mass	Radii (cm)
1	1.0	2.8, 3.2, 3.4
2	0.5	2.2, 2.5, 2.6
3	0.25	1.4, 1.7, 2.0
4	0.125	0.9, 1.3, 1.3

Table 2: Radii, $\bar{x}$, $\bar{y}$, and $SE_{\bar{y}}$

Crumpled Ball	$\bar{x}$	Radii (cm)	$\bar{y}$	$SE_{\bar{y}}$
1	1.0	2.8, 3.2, 3.4	3.1	0.06
2	0.5	2.2, 2.5, 2.6	2.4	0.05
3	0.25	1.4, 1.7, 2.0	1.7	0.10
4	0.125	0.9, 1.3, 1.3	1.2	0.11

In Table 1 and Table 2, N is the number of data points which is equal to the number of independent variables: 4. $\bar{y}$ is the mean radii, $\bar{x}$ is the mean mass, and $SE_{\bar{y}}$ is the standard error of the mean radii $\bar{y}$.

A. Probabilistic Basis for Chi-Square Model Testing

Looking at the probability density function (*pdf*) of the normal distribution, we get the following equation $P(x)$, σ represents the standard deviation, and μ represents the mean:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (7)$$

The normal distribution can thus be represented as a model for the 2-parameter linear equation:

$$P(x) \propto e^{-\sum_{i=1}^n \frac{(y_i - y_{model_i})^2}{2\sigma^2}} \quad (8)$$

Eq. 8 is based on the difference of each data point y_i and the corresponding model value y_{model_i} where the model is the two-parameter linear model $y = Ax + B$. After transforming the normal distribution into a model, we now have a probabilistic basis for chi-squared model testing.

The Central Limit Theorem allows us to construct a normal distribution centered around each data point. We can then ask, given the model, what is the probability of generating all the data? Starting with the probability of generating one point of data and using the probability rule for a sequence of independent events, $PA \cap B = PA \times P(B)$, we obtain for the probability of generating all the data:

$$P(\text{all data}) = e^{-\frac{(y_1 - y_{model_1})^2}{2\sigma_1^2}} \times e^{-\frac{(y_2 - y_{model_2})^2}{2\sigma_2^2}} \dots \times e^{-\frac{(y_n - y_{model_n})^2}{2\sigma_n^2}} = e^{-\sum_{i=1}^n \frac{(y_i - y_{model_i})^2}{2\sigma_i^2}} \quad (9)$$

Defining chi-squared $\chi^2 = \sum_{i=1}^n \frac{(y_i - y_{model_i})^2}{\sigma_i^2}$ to simplify Eq. 9, we get the following result:

$$P(\text{Generating Data} \mid \text{The Model}) \propto e^{-\frac{\chi^2}{2}} \quad (10)$$

This means that the probability of generating the data will be greatest, given the model, when χ^2 is least. The normalized probability density function for chi-squared (**Appendix C**) that will be used later to find probability contours is given by:

$$P(\chi^2) = \frac{1}{2} e^{-\chi^2/2} \quad (11)$$

B. Description of the Chi-Squared Script

The Python script does chi-squared curve fitting and tests the two-parameter linear model $y = Ax + B$. Three arrays must be populated to use the script. These are array x , which is the array of $\bar{x}$ (mean masses), array y , which is the array of $\bar{y}$ (mean radii), and $yerr$, which is the array of the standard errors of the mean radii (error bars) where σ represents the standard deviation and n represents the number of samples shown in Eq. 12.

$$SE = \frac{\sigma}{\sqrt{n}} \quad (12)$$

N represents the total number of data points (i.e., the number of values for the independent variable). σ_r is the standard error of the radii; chi-squared is the sum of the squared differences between data and model, weighted inversely by the square of measurement standard errors.

The process for computing chi-squared in this script takes advantage of the fact that $\min \chi^2$, A_{Best} , and B_{Best} can be obtained in closed-form for linear models. The reason is that chi-squared for two parameters is a paraboloid whose minimum can be obtained by an equation rather than a search.

The python script outputs two plots. The first plot is a log-log data plot whose slope is the fractal dimension. Error bars represent standard errors as defined in Eq. 12. The second plot is a contour plot in the A-B parameter plane and displays two contours.

The inner contour represents the range of parameter values that, given the model with parameter values within this contour, there is a 68% likelihood that the data would have been generated. The outer contour corresponds to a 95% likelihood. The likelihood of the two parameters in the experiment follow a 2probability density function for two measurements as stated in Eq. 11. Thus, to calculate the contours we need to find χ^2_{68} and χ^2_{95} . To obtain the value χ^2 that encloses 95% of the likelihood, we integrate the probability density function in Eq. 11:

$$0.95 = \int_0^{\chi^2_{95}} P(\chi^2) d(\chi^2) = \int_0^{\chi^2_{95}} \frac{1}{2} e^{-\chi^2/2} d(\chi^2) \quad (13)$$

So, we can solve $\chi^2_{68} = 2.3\sigma$, and $\chi^2_{95} = 6.2\sigma$. Now, we can build the contour with $\chi^2_{\min} + 2.3\sigma$ and $\chi^2_{95} + 6.2\sigma$ which is highlighted in the code. This is especially helpful if we have a parameter reference value. If that reference value is outside the yellow line (2σ), then we should reject the model.

Section III: Results, Discussion, and Conclusions

From the outset, the main goal of this paper was to investigate the fractal dimension of crumpled paper using chi-squared instead of ordinary least-squares. Although least-squares generates feasible results, chi-squared provides a more accurate insight into the fractal dimension including better parameter estimation and model testing criteria to reject the model. Fig. 4 below shows the results generated by the chi-squared script that does curve fitting to the 2-parameter linear model.

The results generated include A_{Best} , the fractal dimension (the reciprocal of A_{Best}), B_{Best} , $\min \chi^2$, N value, and the good fit $\min \chi^2$ range. **Fig. 4** shows that a fractal dimension of 2.17 is obtained.

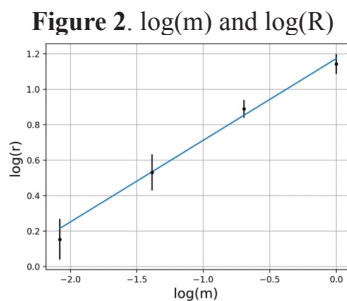


Figure 3. min χ^2 contour

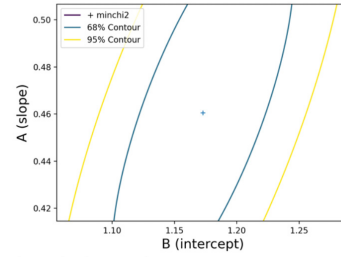


Figure 4. Chi-squared analysis results

```
CHI-SQUARED ANALYSIS RESULTS
A_best = 0.46
fractal dimension = 2.17
B_best = 1.17
minchi2 = 1.11
N = 4
SigmaR = [0.17638342 0.12018504 0.17320508 0.13333333]
Y Error = [0.05629258 0.04939111 0.10188534 0.11428571]
minchi2 good fit range = [ 1.17 , 6.83 ]
chi2 for 68% Contour = 3.41
chi2 for 95% Contour = 7.31
```

This makes sense as fractal dimensions cannot be whole integers and the crumpled paper has dimensionality between a plane $n = 2$, and a sphere $n = 3$. This is because the paper used in **Fig. 1** is likely more rigid than standard sheets of printing paper. The min 2 is 1.47 and the min 2 within a good fit range is [1.17, 6.83], meaning that it falls within the range. Referring to **Fig. 2**, the error bars or variance for the radii for the smaller ball r_4 is much larger than the rest of them. This is most likely due to measurement error in GeoGebra as zooming in on bigger images is easy and zooming in on smaller images creates inaccuracies. However, because our fractal dimension is 2.17 and the min χ^2 value of 1.47 falls within the good fit range from 1.17 to 6.83, so we should not reject this model. The good fit range was determined using the following:

$$- GFR = N - \sqrt{2N} \text{ and } + GFR = N + \sqrt{2N} \text{ since } \chi^2_{\min} \approx N \quad (14)$$

Looking at **Fig. 3**, to find out what min χ^2 is, we simply find the intersection of A_{Best} and B_{Best} as both are quadratic functions in a one-dimensional plot, with χ^2 the y-axis and $A(\text{slope})$ or $B(\text{intercept})$ as the x-axis, shown in **Fig. 2** and **3**. Knowing this, χ^2 will always appear at the vertex of the parabola where $x = A_{\text{Best}}$ or B_{Best}

There is a slant to the shape of the paraboloid since we expect there to be a high correlation coefficient value. If there was no slant, the change in the y -intercept would not affect the slope which is not the case. Furthermore, looking at all the points mentioned above, the model should not be rejected.

A. Conclusion, Applications, Limitations

In conclusion, chi-squared analysis was used for optimal parameter estimation of the fractal dimension and model testing. In doing so, a reasonable value for the fractal dimension, 2.17, was obtained. Looking back at the model, we could also find B_{Best} represented by ρ as density. However, we are only interested in A_{Best} as the reciprocal of A_{Best} being the fractal dimension. Using the chi-squared code has helped simplify the process a lot rather than doing the chi-squared analysis by hand as well. The code has some drawbacks including flexibility of the code only on linear models. A potential improvement would be to make a more universal chi-squared test that could apply to multiple situations. Error in testing was reduced by taking three measurements all of different lengths. However, using GeoGebra did have some downsides as it was not as exact as we were hoping for. By actually taking the mass of the balls and using a caliper, for example, we could get more precise results. Another way of increasing the precision of obtaining an accurate fractal dimension is to increase the number of sample measurements taken. Looking at the bigger picture, however, fractals and especially chi-squared model testing can provide a unique solution to many problems and should be considered whenever practical, especially in new fields of research. Furthering this study, a potentially interesting approach to solving this problem could be increasing the number of parameters or using a non-linear model to express the fractal dimension.

Acknowledgements

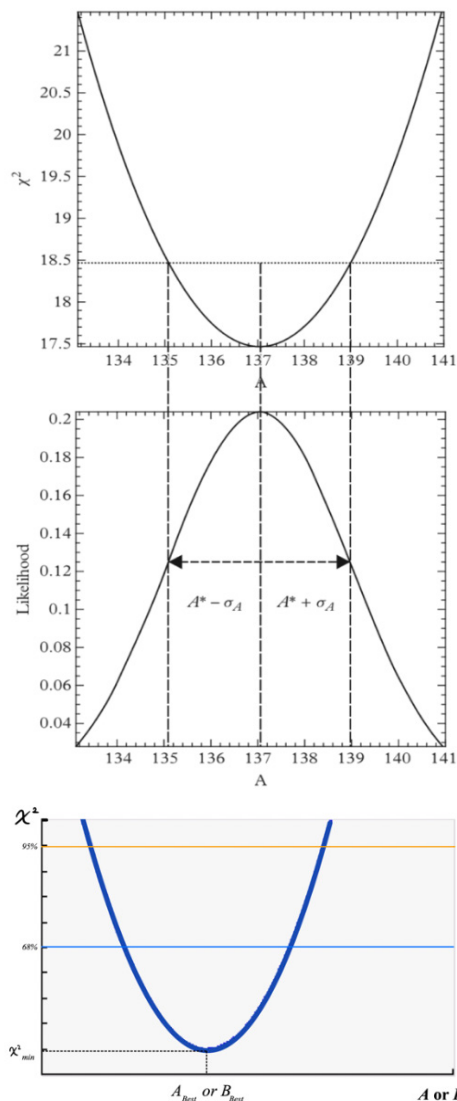
The Python code was written by Dr. Carey Witkov (former Preceptor) and Keith Zengel, Preceptor. Harvard University, Physics Department, Cambridge, MA. I thank and acknowledge assistance

from Eric Arsenault and Dr. Carey Witkov for making comments to improve the readability of the manuscript and for providing their expertise in the field.

References

- Carey Witkov, K. Z. (2019). *Chi-Squared Data Analysis and Model Testing for Beginners*. Oxford University Press.
- Carey Witkov, K. Z. (2020). *Chi-Squared script in Python public software release*. Retrieved from Github: <https://github.com/witkov/chi-squared>
- Crumpled Paper Lab*. (n.d.). Retrieved from Yale Mathematics: https://users.math.yale.edu/public_html/People/frame/Fractals/Labs/CrumpledPaperLab
- Devaney, R. L. (1995, April 2). *Fractal Dimension*. Retrieved from Boston University Mathematics and Statistics: <https://math.bu.edu/DYSYS/chaos-game/node6.html>
- Gomes, M. (1987). Fractal Geometry in Crumpled Paper Balls. *American Journal of Physics*.
- Gomes, M. (1987). Paper Crushes Fractally. *Journal of Physics A: Mathematical and General*.
- Mandelbrot, B. B. (1981). Fractals and the Geometry of Nature. *Yearbook of Science*.
- Mirowski, P. (1990). From Mandelbrot to Chaos in Economic Theory. *Southern Economic Journal*.

APPENDIX A: Gaussian vs. Chi-Squared Distribution (Witkov & Zengel, 2019)



APPENDIX B: Chi-Square Curve Fitting for 2-Parameters

```
# 2dchi2.py version 5/12/20
# This script does chi-square curve fitting to the
# 2-parameter linear model  $y = Ax + B$ 
# SOFTWARE DEPENDENCIES: Numpy, Matplotlib
# -----
```

```
# Copyright (C) 2020 Carey Witkov and Keith Zengel
# This program is free software: you can redistribute it
# and/or modify it under
# the terms of the GNU General Public License as
# published by the Free Software
# Foundation; either version 3 of the License, or (at
# your option) any later version.
```

```
# This program is distributed in the hope that it will be
# useful, but WITHOUT ANY WARRANTY;
# without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE.
# See the GNU General Public License for more
# details.
```

```
# You should have received a copy of the GNU
# General Public License along with this program.
# If not, see https://www.gnu.org/licenses/.
```

```
# -----
```

```
import matplotlib.pyplot as plt
from numpy import *
import numpy as np
```

```
# relative mass of crumpled paper ball
m = np.array([1, 0.5, 0.25, 0.125])
```

```
# Radii of crumpled paper balls
```

```
r1 = np.array([3.4, 3.2, 2.8])
```

```
r2 = np.array([2.5, 2.2, 2.6])
```

```
r3 = np.array([2, 1.7, 1.4])
```

```
r4 = np.array([1.3, 1.3, 0.9])
```

```
r = np.array([mean(r1), mean(r2), mean(r3),
mean(r4)])
```

```
# number of data points
```

```
N = len(r)
```

```
# number of samples for each data point
```

```
n = len(r1)
```

```
sigma_r = np.array([np.std(r1, ddof=1), np.std(r2,
ddof=1), np.std(r3, ddof=1), np.std(r4, ddof=1)]/
np.sqrt(n));
```

```
# x,y data arrays and y-error array
```

```
x = log(m);
```

```
y = log(r);
```

```
yerr = sigma_r/r;
```

```

# calculate sums needed to obtain chi-square
Syy=sum(y**2/yerr**2)
Sxx=sum(x**2/yerr**2)
S0=sum(1/yerr**2)
Sxy=sum((y*x)/yerr**2)
Sy=sum(y/yerr**2)
Sx=sum(x/yerr**2)

A_best = (S0*Sxy - Sx*Sy)/(S0*Sxx - (Sx*Sx))
B_best = (Sy*Sxx - Sx*Sxy)/(S0*Sxx - (Sx*Sx))
minchi2 = Syy+(S0*(Sxy*Sxy) - 2*Sx*Sy*Sxy +
Sxx*(Sy*Sy))/((Sx*Sx) - (S0*Sxx))
sigmaA = 1/sqrt(Sxx)
sigmaB = 1/sqrt(S0)

# create parameter grid for the minchi2
a = np.linspace(Abest-
0.2*Abest,Abest+0.2*Abest,500)
b = np.linspace(Bbest-
0.2*Bbest,Bbest+0.2*Bbest,500)
A,B = np.meshgrid(a,b)

# calculate chi-square over parameter grid (scan and
find using a matrix type calculation)
chi2 =(Syy) + (A**2)*(Sxx) + (B**2)*(S0) -
2*A*Sxy - 2*B*Sy + 2*A*B*Sx

# plot data with error bars
plt.figure()
plt.grid(True)
plt.errorbar(x,y,yerr,linestyle='None',fmt='k')
plt.xlabel('log(m)', fontsize=16)
plt.ylabel('log(r)', fontsize=16)

# plt.title('y vs x data and line-of-best-fit',
fontsize=16)
plt.plot(x,Abest*x+Bbest)

# plot chi-square in A-b parameter plane with 68% and
95% contours
plt.figure()
levels=[minchi2,minchi2+2.3,minchi2+6.2]
Z=plt.contour(B,A,chi2,levels)
plt.ylim(Abest-0.1*Abest,Abest+0.1*Abest)
plt.xlim(Bbest-0.1*Bbest,Bbest+0.1*Bbest)
plt.plot(Bbest,Abest,'+')
plt.ylabel('A (slope)', fontsize=16)
plt.xlabel('B (intercept)', fontsize=16)

# add legend
artists, labels = Z.legend_elements()
plt.legend(handles=artists, labels=["+ minchi2", "68%
Contour", "95% Contour"], loc='upper left')
plt.show()

# display results
print("CHI-SQUARED ANALYSIS RESULTS")
Abest_float = "{}

# display results
print("CHI-SQUARED ANALYSIS RESULTS")
Abest_float = "{:.2f}".format(Abest)
FractalDimension = "{:.2f}".format(1/Abest)
print("A_best =", Abest_float)
print("fractal dimension =", FractalDimension)
Bbest_float = "{:.2f}".format(Bbest)
print("B_best =", Bbest_float)
minchi2_float = "{:.2f}".format(minchi2)
print("minchi2 =", minchi2_float)
print("N =", N)
print("SigmaR =", sigma_r)
print("Y Error =", yerr)
N_min_float = "{:.2f}".format(N-math.sqrt(2*N))
N_max_float = "{:.2f}".format(N+math.sqrt(2*N))
print("minchi2 good fit range = [", N_min_float, ",",
N_max_float, "]")
contour68 = "{:.2f}".format(levels[1])
print("chi2 for 68% Contour =", contour68)
contour95 = "{:.2f}".format(levels[2])
print("chi2 for 95% Contour =", contour95)

```

APPENDIX C: Derivation of Probability Density Function

The derivation of the probability density function (*pdf*) for two measurements x and y follows. We need to take the combined multiple probability density function for both x and y

$$C(x, y) = C(x)C(y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}}$$

Since we are only interested in contours of equal probability in the function $CC(x, y)$ which are defined as $\chi_x^2 + \chi_y^2$ which is equal to some constant τ :

$$\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} = \chi_x^2 + \chi_y^2 = \tau$$

So we are interested in how the probability changes when jumping from one contour of equal probability to another. We write this probability in terms of

$$\chi = \sqrt{\chi_x^2 + \chi_y^2}$$

Using polar coordinates, we change x and y to χ_x and χ_y respectively:

$$\iint G(x)G(y) = \iint G(x)G(y) \frac{dx}{d\chi_x} d\chi_x \frac{dy}{d\chi_y} d\chi_y = \iint \frac{1}{2\pi} e^{-\frac{\chi_x^2}{2} - \frac{\chi_y^2}{2}} d\chi_x d\chi_y$$

We introduce an angular coordinate, ϕ , to convert cartesian coordinates to polar coordinates so we can rewrite the double integral as:

$$\iint \frac{1}{2\pi} e^{-\frac{\chi^2}{2}} \chi d(\chi) d(\phi) = \int e^{-\chi^2/2} \chi d(\chi)$$

Changing the variables from $\chi \rightarrow \chi^2$ and $d\chi = d\chi^2/(2\chi)$

$$P(\chi^2) = \frac{1}{2} e^{-\chi^2/2}$$