The Strange Attractor of the Lorenz System

Jason Glowney

June 2, 2017

Abstract

In 1963 Lorenz published his seminal paper *Deterministic Non-periodic flow* in the Journal of Atmospheric Sciences. The philosophical ramifications of the unpredictability of phenomenon in nature noted in this work were profound and the implications have fueled an incredible development in dynamical systems. In this paper, we explore this system and its enigmatic strange attractor, by looking into the dynamics of the Lorenz equations, defining its chaotic attributes thru both an analytic and visual approach, and ultimately showing that the Lorenz system does indeed support the existence of this strange attractor.

Introduction

We explore the strange attractor of the following set of equations

$$\begin{gathered} \dot{x} = -\sigma(x - y) \\ \dot{y} = -xz + rx - y \\ \dot{z} = xy - bz. \end{gathered} \tag{1}$$

We notice the following quadratic terms in $-xz$ and xy make the above system nonlinear. Lorenz described parameter values $\sigma = 10, r = 28$ and $b = \frac{8}{3}$ for his attractor [1].

In physics and applied mathematics we are very accustomed to the idea of a conservative system, where the phase space volumes are conserved as time evolves. Dissipative systems are those that have their volumes decrease in time so the area of the final set will be less than the initial set. The idea of an attractor fits into the dissipative system where as time $\rightarrow \infty$ the set of initial conditions is drawn to some subset of the phase space. An example is the damped harmonic oscillator that will overtime come to rest. These attractor subsets of phase space can be a point in phase space (zero-dimensional), a limit cycle in phase space (one-dimensional) and other integer valued subsets.

It was David Ruelle and Floris Takens that introduced the term "strange attractor" as an attractor set in phase space that resembles that of the Lorenz system [3]. In a strange attractor, the attracting set is much different than the sets mentioned above. The subset of phase space can have a dimensional value that is not an integer and is called fractal in terms of its dimension. There is no concrete definition [2] for the term strange attractor but the following should be without controversy:

Definition *– A set Λ is called a strange attractor if*

3. *The set Λ is compact.*
3. *There exists an open set U that contains Λ and is attracted to Λ, i.e. the distance between the orbit of x and the set Λ tend to 0 as the time $\to \infty$ for all x in U.*
3. *There is a dense orbit in Λ, i.e. there is a point in Λ whose orbit visits every neighborhood of every point in Λ.*
3. *For every point in Λ and any neighborhood of that point there exists a point in the neighborhood such that the two points eventually separate to a fixed distance $\delta>0$ under the flow.*

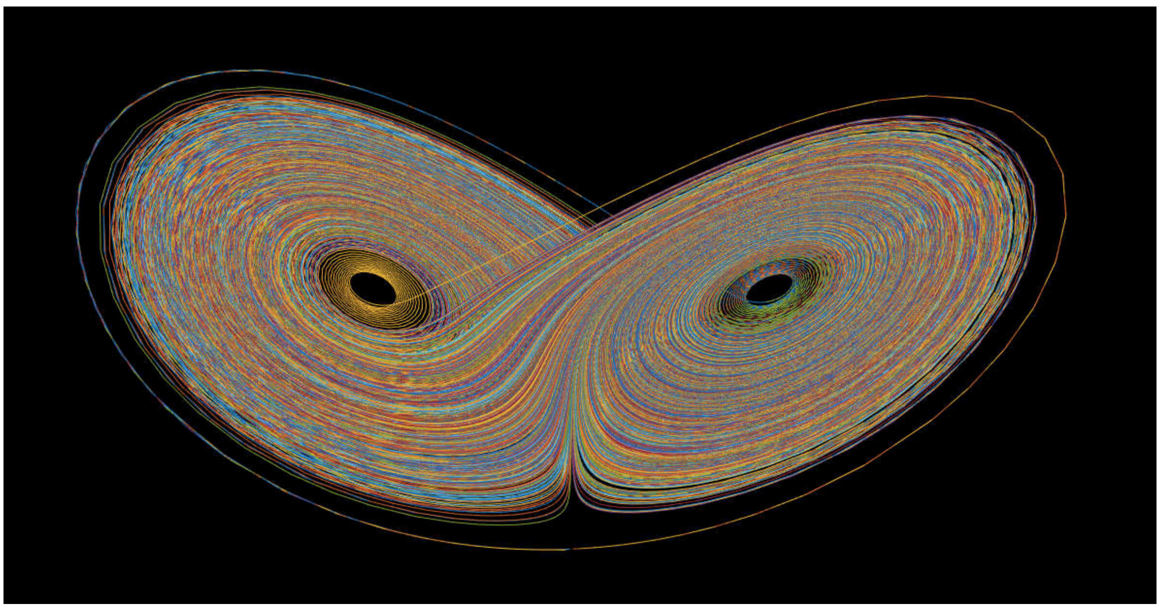


Fig. 1 Lorenz strange attractor- 10 points with random values of x from (-1,1) iterated forward 250 times in MATLAB.

Dynamics at the Equilibrium Points

The first step in understanding the aforementioned system is to determine the fixed points. Setting the above equations (1) equal to zero, we find we would have to have

$$\begin{gathered} x = y \\ x = \pm\sqrt{bz} \\ 0 = x(-z + r - 1). \end{gathered} \tag{2}$$

So, we would have the following fixed points

$$C^0(0,0,0) \text{ and } C^{\pm} = \left(\pm\sqrt{b(\sigma-1)}, \pm\sqrt{b(\sigma-1)}, \sigma-1\right). \tag{3}$$

In order to determine the stability of our fixed points we linearize the system by determining the Jacobian

$$J = \begin{pmatrix} -\sigma & \sigma & 0 \\ -z+r & -1 & -x \\ y & x & -b \end{pmatrix}. \tag{4}$$

We find the Jacobians of our fixed points are then

$$J|_{0,0,0} = \begin{pmatrix} -\sigma & \sigma & 0 \\ r & -1 & 0 \\ 0 & 0 & -b \end{pmatrix} \tag{5}$$

$$J|_{\sqrt{b(r-1)},\sqrt{b(r-1)},r-1} = \begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & -\sqrt{b(r-1)} \\ \sqrt{b(r-1)} & \sqrt{b(r-1)} & -b \end{pmatrix} \tag{6}$$

$$J|_{-\sqrt{b(r-1)},-\sqrt{b(r-1)},r-1} = \begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & \sqrt{b(r-1)} \\ -\sqrt{b(r-1)} & -\sqrt{b(r-1)} & -b \end{pmatrix}. \tag{7}$$

To find our characteristic polynomial for the origin we would compute

$$P_1(0,0,0) = \det[\lambda I - J(0,0,0)] = (\lambda+b)\left(\lambda^2 + (\sigma+1)\lambda + \sigma(1-r)\right). \tag{8}$$

We would find the eigenvalues of the Jacobian at the origin to be

$$\begin{aligned} \lambda_1 &= -b \\ \lambda_2 &= -1-\sigma-\sqrt{\sigma^2-2\sigma+4\sigma r+1} \\ \lambda_3 &= -1-\sigma+\sqrt{\sigma^2-2\sigma+4\sigma r+1}. \end{aligned} \tag{9}$$

Evaluation of these eigenvalues tells us that for values of $r < 1$ all of the eigenvalues will be stable. For values of $r > 1$ we would have one of the eigenvalues positive and this would make the origin an unstable point or saddle node.

To find our characteristic polynomial for the other two equilibrium points (C^+ and C^-) we have

$$P_{2,3}\left(\pm\sqrt{b(r-1)},\pm\sqrt{b(r-1)},r-1\right)=\det\left[\lambda I-J\left(\pm\sqrt{b(r-1)},\pm\sqrt{b(r-1)},r-1\right)\right]$$
$$=\lambda^3+(1+b+\sigma)\lambda^2+b(\sigma+r)\lambda+2b\sigma(r-1). \tag{10}$$

The above result is in the form of a cubic polynomial. If we set $\alpha=-(1+b+\sigma),\beta=b(\sigma+r)$ and $\delta=2b\sigma(r-1)$. We would then have

$$P_{2,3}=\lambda^3-\alpha\lambda^2+\beta\lambda-\delta. \tag{11}$$

We know that the root of $P_2 and\ P_3$ have negative real parts if $\alpha<0\ and\ \alpha\beta<\delta<0$. To satisfy this and have a stable equilibrium points we would have to have

$$r<\frac{\sigma(\sigma+3+b)}{(\sigma-b-1)}. \tag{12}$$

Using the parameters $\sigma=1,r=28$ and $b=\frac{8}{3}$ in our previous Jacobians we find eigenvalues at the origin are

$$\lambda_1\approx 11.83,\lambda_2\approx -22.83 \text{ and } \lambda_3\approx -2.67.$$

Which is showing that the origin is a saddle point with two stable directions and one unstable direction.

We similarly find eigenvalues for our fixed points $C^{\pm}$ are

$$\lambda_1\approx -13.85,\lambda_2\approx 0.09+10.19i \text{ and } \lambda_3\approx 0.09-10.19i.$$

We see that these fixed points are saddles as well, but with one direction that is stable and two directions that are spiraling out.

Evolution of Orbits on the Strange Attractor

Orbits on the Lorenz system will be confined inside of a sphere denoted as

$$x^2+y^2+\left(z-(\sigma+r)\right)^2=C^2 \tag{13}$$

with large enough C. So, we would find that this sphere in phase space binds orbits as all of the equilibrium points are saddle points. The orbits would not be able to settle down to a steady state as a result.

To get a better understanding of how orbits progress on the Lorenz attractor we will plot the evolution using MATLAB. We will start with a point $(x_0,y_0,z_0)=(1,1,1)$

which is close to the origin. We see below in **Fig. 2a** that our point starts and approaches our C^+ equilibrium point, orbits once and then progresses to our C^- equilibrium point. In **Fig. 2b** we see the orbit path comes closer to the C^- equilibrium point and orbits the equilibrium several times before making its way back to C^+. In **Figs. 2c** and **2d** we see the continued evolution where more orbits accumulate around both C^+ and C^- and form the two wings of the strange attractor. We can take note that these orbits are aperiodic, in that the periods of making a cycle across either or both wings will change continually, exposing its chaotic nature.

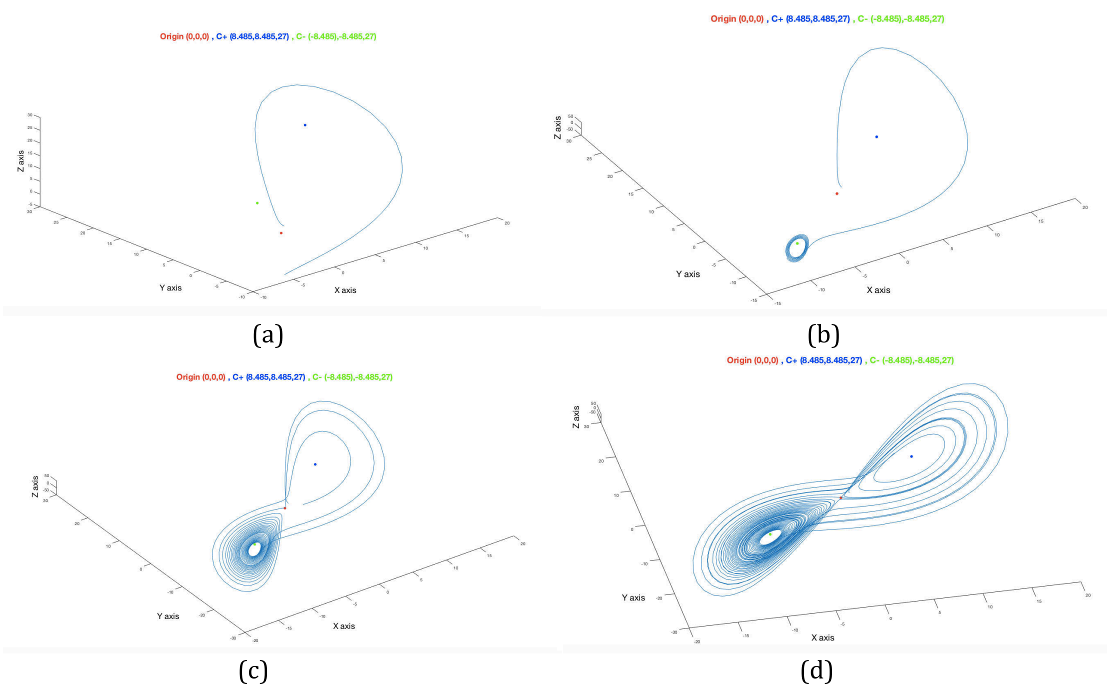


Fig. 2 Evolution of an orbit on the Lorenz attractor

In addition to the circular orbits noted above, the orbits tend to spiral out on the wings of the attractor until they hop over to the next wing. The factor that seems to play the role with when this hop occurs, is how far out on the wing the orbit is. Lorenz studied this phenomenon by looking at the dynamics of the maximal z values for orbits. In his origin paper, he obtained a sequence of the z values and wound up plotting z_i vs. z_{i+1} to further investigate how the position of an orbit affects the successive orbit [1]. We have re-created this plot in MATLAB using the peaks command to gather maximal z values and then plotted the z_i vs z_{i+1} sequence as seen in **Fig. 3**.

What we can surmise is that there appears to be a near perfect relationship between the successive maximal z-values. We see that this near 2-to-1 symmetry between the two wings, is due to the fact that the maximal value can stem from either spiral.

The slope of the one-dimensional graph is also greater than |1| at all values and is thus expanding continually.

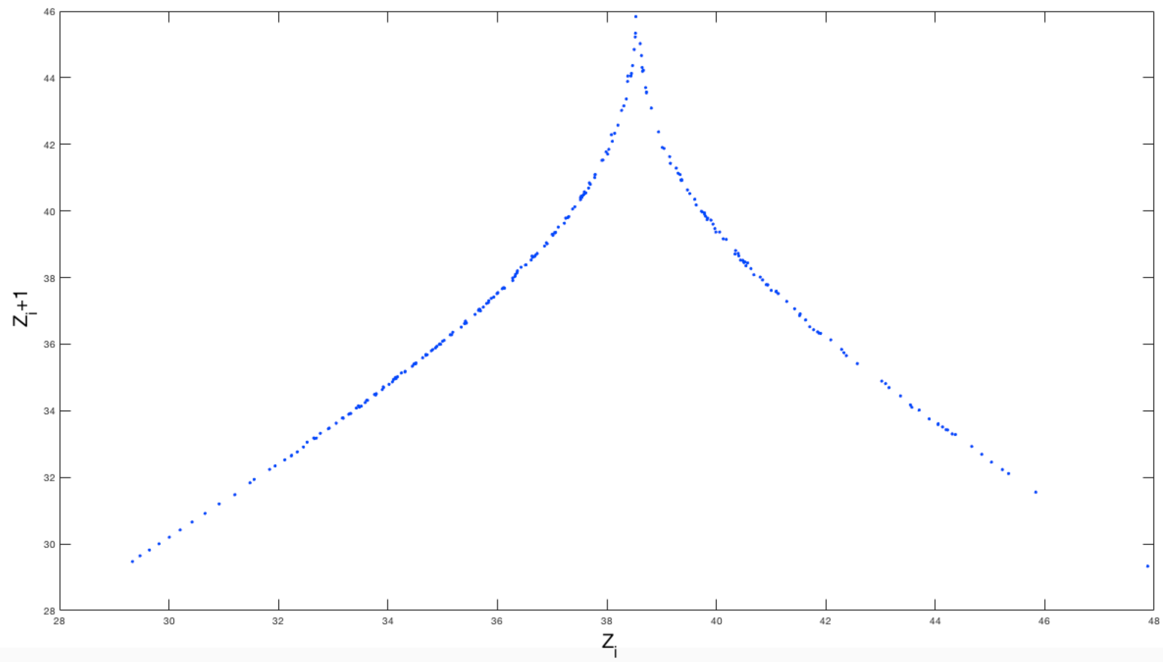


Fig. 3 Maximal z dynamics of Lorenz system

The Lorenz System's Sensitivity to Initial Conditions, Lyapunov Exponents and Fractal Dimension

During Lorenz's study of his system he was carrying out calculations to 6 digits [2]. Printers at the time were slow, and in an effort to speed things up he was only printing out to 3 decimal digits. It was at one point during a computation that he wanted to take a look at a particular solution. He fed the printed truncated numbers with 3 decimal digits into the numerical integration algorithm. It then came as quite the surprise to him, that gradually the solution would diverge from the original 6 decimal digit numbered solution and eventually yielded completely different results. What Lorenz had stumbled upon is a hallmark of chaotic systems and their sensitivity dependence on initial conditions.

To demonstrate this, we have taken 2 points that only differ by 0.000001 in their x value. We have progressed over 200 iterations using ODE45 in MATLAB (4^{th} order Runge-Kutta) to look for divergence in the values of each point. In **Fig. 4** we see the end location of each of the points on the strange attractor as marked by a red diamond for point (1,1,1) and a blue diamond for point (0.999999,1,1). We can see that these points have diverged from one another and are now on opposite wings of the attractor.

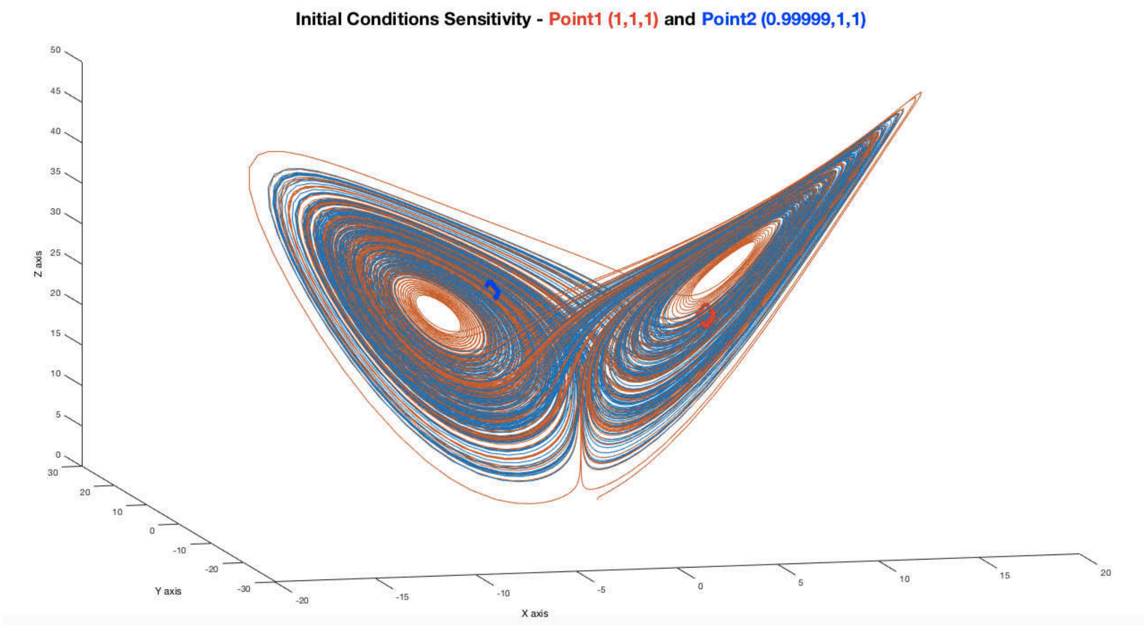


Fig. 4 Sensitivity to Initial Conditions

Plotting values (**Fig. 5**) for x for both points over time shows very well this hallmark of chaos. The points have been colored coded by red and blue. For the initial iterations, the values of each point thru the numerical integrations line up quite well, with the color blue obscuring the red color of the point 1 that lay beneath it. At around time-step 2000 we see a drastic divergence of both points from one another and it becomes readily apparent that the points are on very different orbital paths at this point in time.

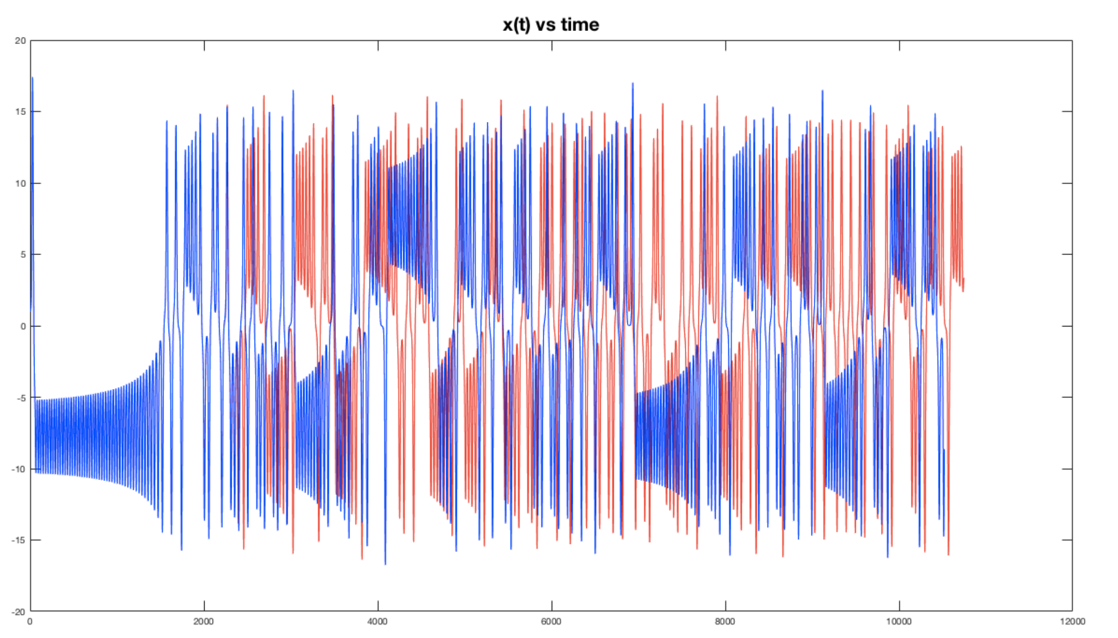


Fig. 5 Divergence of initial conditions in Lorenz system

In dynamical systems, we use the Lyapunov exponents (λ) to assess a system for chaos. It is defined as follows for one-dimensional maps

$$e^{\lambda n} = \frac{\Delta x_n}{\Delta x_0} \approx \frac{dx_n}{dx_0} \tag{14}$$

and

$$\lambda = \lim_{n \to \infty} \frac{1}{n} ln \left| \frac{dx_n}{dx_0} \right|. \tag{15}$$

Here Δx_0 and Δx_n are the respective differences between the 2 initial conditions and the difference between the *n-th* iterates. Indications of chaos in terms of Lyapunov exponents would be to have at least one positive exponent and the larger the more sensitive dependence exists.

Lyanpunov exponents can be very hard to solve analytically even in the 1-dimensional case. For the Lorenz system, we have used a MATLAB program that appears in "Dynamical Systems and Applications using MATLAB" by Stephen Lynch. In this program, he used the protocol developed by Wolf, Swift, Swinney and Vastano in their paper "Determining Lyapunov Exponents from a Time Series". In their method, they monitor the long-term growth rate of small elements in the attractor to find the Lyapunov Exponents [7].

Running the MATLAB program for the Lorenz system we find after 500 iterations we have Lypanunov exponent values of 0.883672, 0.000840 and -14.547618. Seeing the first is both positive and large is an indication of the sensitivity dependence in the Lorenz system. Below in **Fig. 6,** I have plotted the values of the Lyapunov exponents over the iterative time-span and we can see our large positive exponent plotted in the color blue.

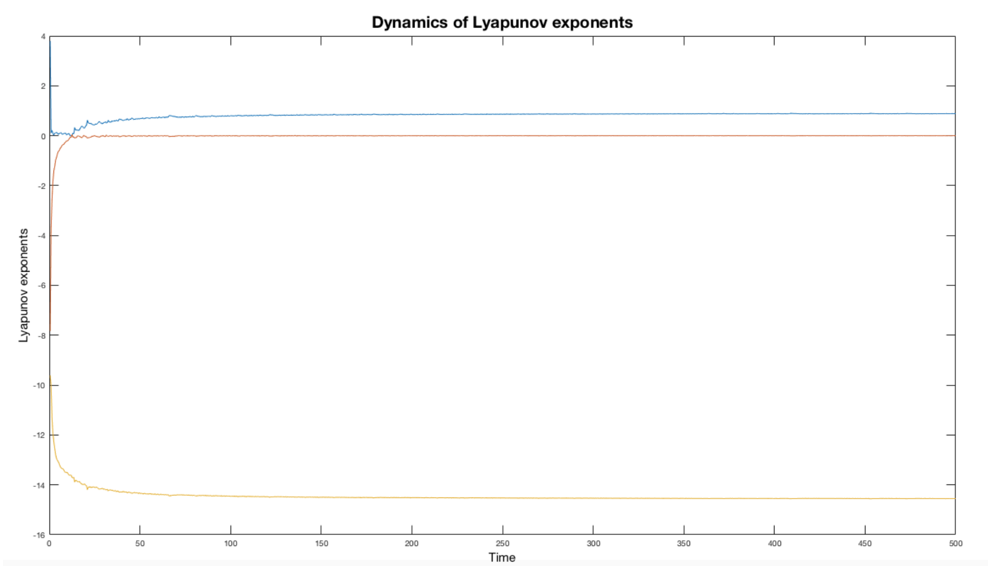


Fig. 6 Plot of Lyapunov Exponents of the Lorenz system using a Time Series

Another characteristic of chaotic systems is the fact that the dimensions of the system will tend to have a value that is a non-integer. Using MATLAB code written by Chris King from The University of Auckland we are able to indeed verify that the Lorenz system's dimension value is fractal and not an integer value. The technique involved using a box counting method as such

$$D_0 = \lim_{\epsilon \to 0} \frac{ln\hat{N}(\epsilon)}{\ln\left(\frac{1}{\epsilon}\right)}. \tag{16}$$

Where ϵ is the side length of an N-dimension object and $\hat{N}(\epsilon)$ is the number of cubes required to cover the set.

Running the MATLAB code and extending it to forty thousand iterations we arrive at a fractal dimensional value of 2.0487 for our Lorenz system. This is close to the proposed value of 2.060. As seen in **Table 1** we see the fractal dimension seems to level off with iterations higher than 40,000.

Iterations	Fractal Dimension
10,000	2.007549262452206
20,000	2.025916649222897
30,000	2.041486175275203
40,000	2.048716730162061
50,000	2.048586419469994

Table 1. Number of Iterations and Lorenz system Fractal Dimension results using dlor.m

Evolution of the Lorenz System with Variation of the Rayleigh Constant

While looking into the dynamics of the equilibrium points we were able to see that the transition from $r = 1$ represented a junction of interest. We also saw that for our equilibrium points at $\left(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1\right)$ and $(-\sqrt{b(r-1)}, -\sqrt{b(r-1)}, r-1)$ a change in stability occurred from stable to unstable at the value of

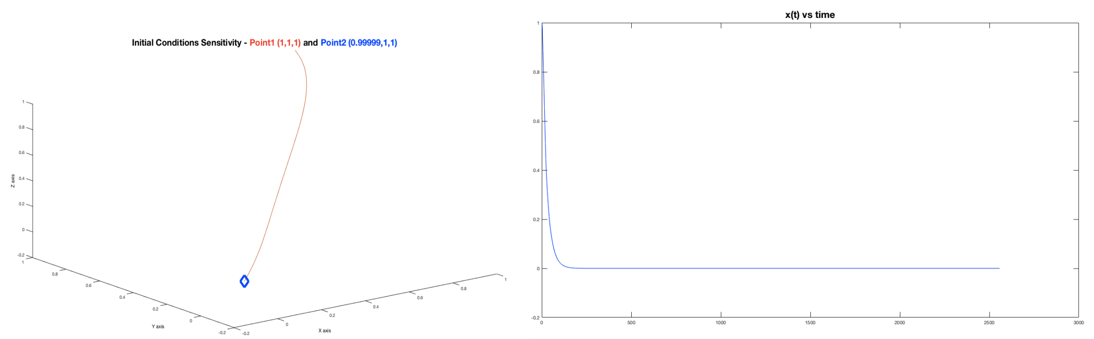
$$r = \frac{\sigma(\sigma + 3 + b)}{(\sigma - b - 1)}. \tag{17}$$

We will use the values, as did Lorenz, for σ and b as 10 and 8/3. Plugging these into the above equation we find that

$$r = \frac{10(10 + 3 + \frac{8}{3})}{(10 - \frac{8}{3} - 1)} = r_h \approx 24.7368.$$

This value we will call r_h and it denotes a Hopf bifurcation where C^+ and C^- lose their stability and this is where the dynamics of a strange attractor begin.

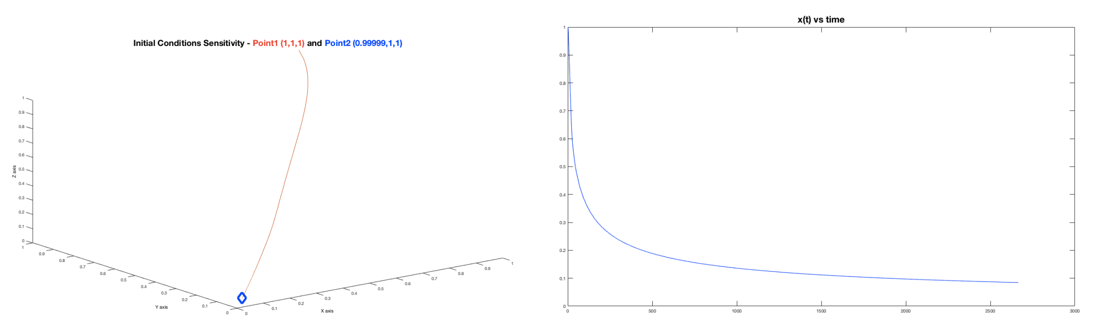
To help the reader visualize the Lorenz system with different values for the Rayleigh constant *r,* we created in MATLAB plots of the Lorenz system with *r*-values of 0.5, 1, 10, 23, and 24.2 to study the dynamics. Recall that similar plots for *r*=28 have been show previously. We have plotted 2 points in each system that are extremely close to one another to also help visualize sensitivity dependence of initial conditions. Point values are (1,1,1) and (0.999999,1,1).



(a) (b)

Fig. 7 (a) Lorenz system plot *r* =0.5, (b) *x(t)* vs. time for r=0.5.

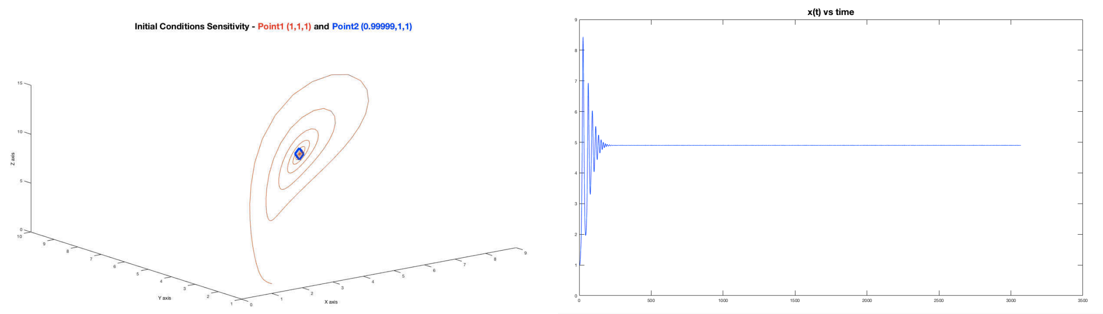
In **Fig. 7** we see no divergence on the path of the 2 points. The origin is stable by our previous dynamics assessment and we see in the plots that the orbits rapidly approach an attracting point.



(a) (b)

Fig. 8(a) Lorenz system plot *r* =1, (b) *x(t)* vs. time for r=1.

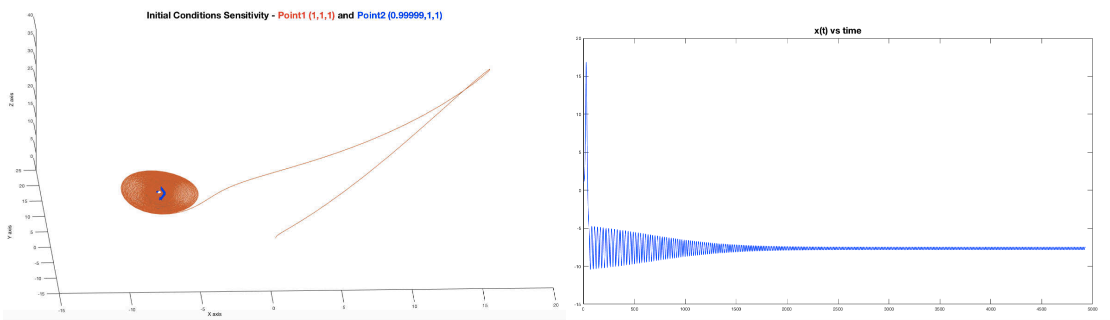
In **Fig. 8** we see no divergence on the path of the 2 points. The origin is in transition from being stable to unstable and we see the points approach at infinity in a curved trajectory as seen in **Fig. 8b**.



(a) (b)

Fig. 9(a) Lorenz system plot *r*=10, (b) *x(t)* vs. time for r=10.

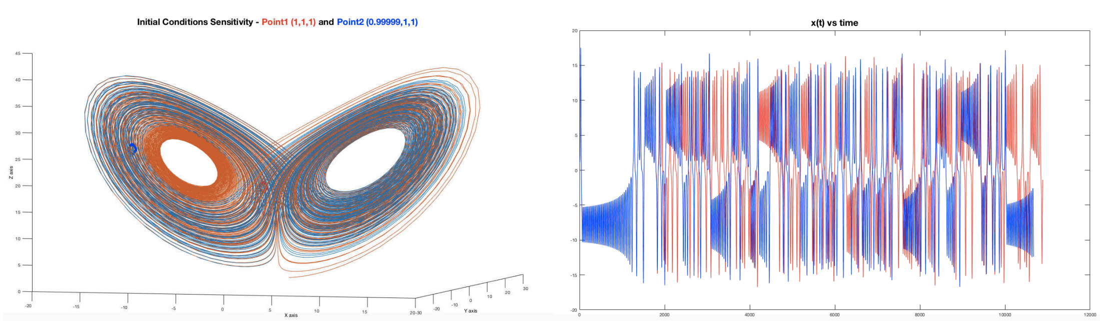
In **Fig. 9** we see no appreciable divergence on the path of the 2 points. The origin has become an attractor and converges to a singular point. In **Fig. 9b** we see the initial oscillatory cycle of the point as it approaches the singular attractor equilibrium.



(a) (b)

Fig. 10 (a) Lorenz system plot *r* =23, (b) *x(t)* vs. time for r=23.

In **Fig. 10** we see no appreciable divergence on the path of the 2 points. We see that a limit cycle has formed. The oscillatory orbits are greater in number and more tightly spaced. We see in **Fig. 10b** small oscillations persist as time goes to infinity.



(a) (b)

Fig. 11(a) Lorenz system plot *r* =24.2, (b) *x(t)* vs. time for r=24.2.

In **Fig. 11** we see significant divergence on the path of the 2 points. We see that the butterfly has formed with the points orbiting both wings. We see in **Fig. 11b** that the points have diverged and are displaying chaotic behavior. At this stage, the system has become a chaotic attractor.

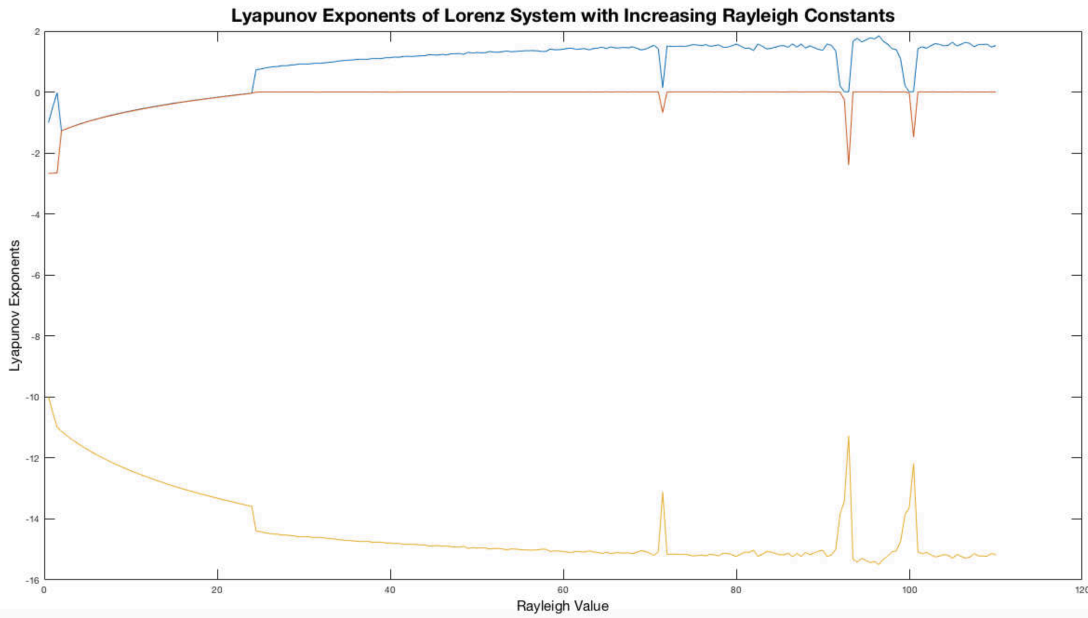


Fig. 12 Lyapunov exponents of Lorenz system with increasing Rayleigh values

In **Fig. 12** above, we plot the values of Lyapunov exponents for *r*-values from 0 to 110. We can see postive Lyapunov exponents starting at about *r*=24, heralding chaotic onset. We can see for *r*-values up to 110 we have 3 interesting areas of fluctuation in the value of the positive Lyapunov exponents; approximately *r*= 72,92 and 100. Many times, these changes can be areas of interesting dynamics. If we take the example of the dip in *r*-value at around 100, this is indeed a curious transition point of the Lorenz system. At an *r*-value of 99.96 the system becomes a T(3,2) torus knot as plotted below in **Fig. 13**.

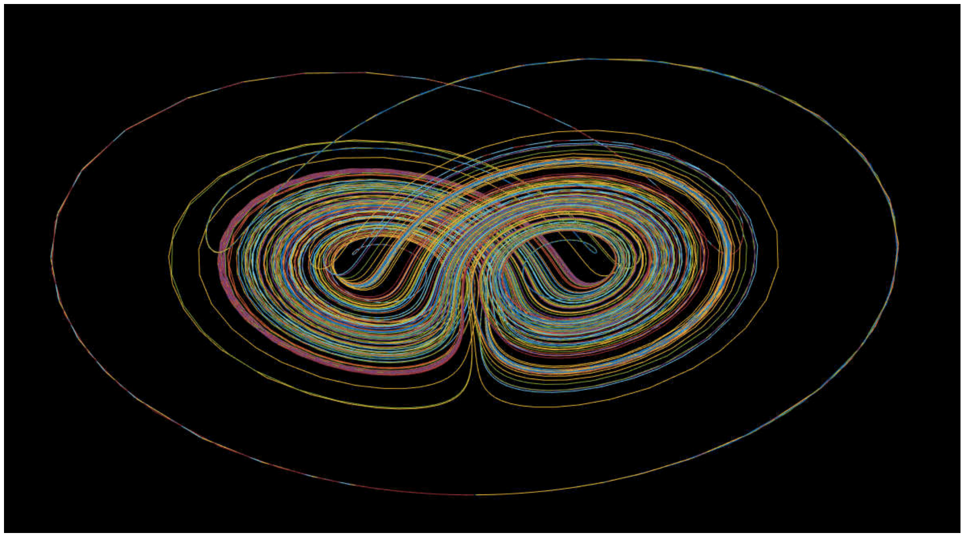


Fig. 13 – A T(3,2) torus knot for *r*=99.96

The Pitfalls of Numeric Integration and Proof the Lorenz Attractor Exists

Discrete numerical methods with finite time steps are the most practical technique to study chaotic systems. This holds in particular for the Lorenz system, as there exists no analytic solutions to the nonlinear system of equations at this time. The choice of numerical integration technique can yield different results as well as the fact that these numerical methods are time-step dependent. This time-step dependence is due to two main factors: first is that these chaotic differential equations are unstable and small errors can be amplified exponentially in the vicinity of an unstable manifold and second is that stable and unstable manifolds of singular points can form virtual separatrices which can jump thru trajectories and open up the possibility of profound numerical errors.

To study numerical integration techniques and the differing results we used MATLAB to iterate a singular point at $(x_0, y_0 z_0) = (1,1,1)$, we used 3 different and popular ODE solvers *(ODE45, ODE23, ODE113),* and we set equal the step sizes for each using a tspan=0:0.001:100. What became readily apparent is that we quickly noted divergence of the solutions for the singular point as it iterated forward. In **Fig. 14** we see the endpoints of each respective singular point are in very different locations of the strange attractor. We document the numeric values of these endpoints and find them to be

$$p_n(ODE45) = (-10.18, -15.78, 20.03)$$
$$p_n(ODE23) = (14.78,10.62, 39.23)$$
$$p_n(ODE113) = (1.01,1.97, 17.53).$$

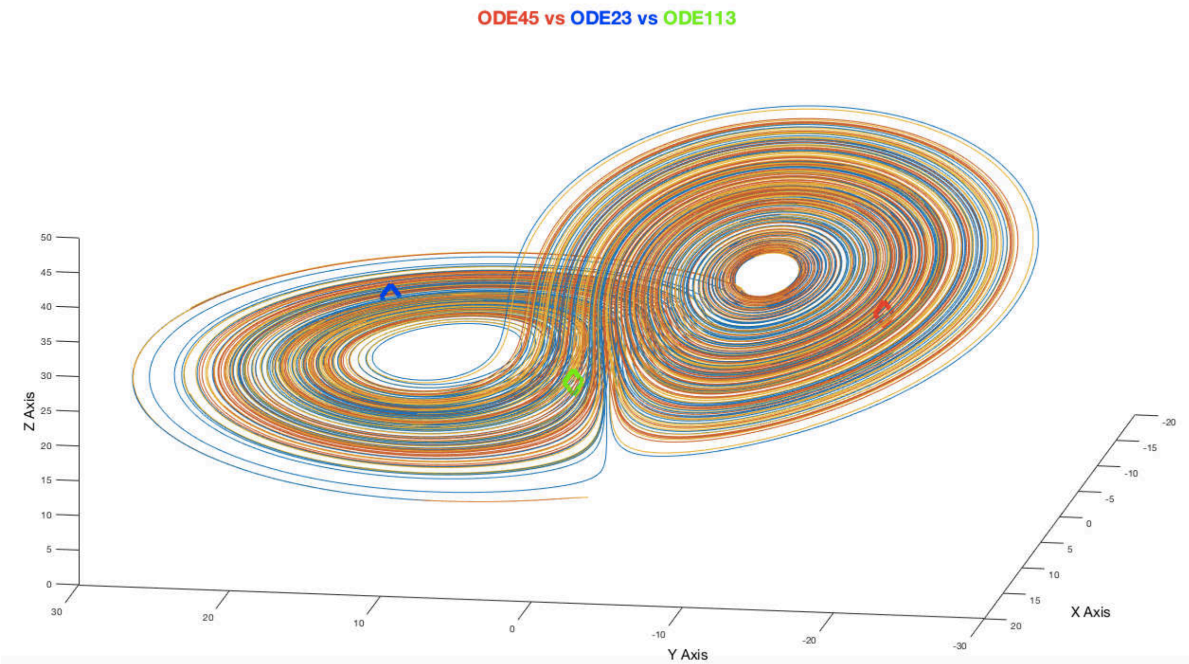


Fig. 14 Endpoints of singular points with differing numerical integration techniques.

In **Fig. 15** we plot the position of the x-value vs. time and with close inspection we see a very quick divergence in ODE45/ODE113 vs. ODE23, with ODE23 showing tighter orbits vs. the other two ODE solver solutions. At about step $(1.5)^4$ we see evidence of ODE45 and ODE113 solutions jumping to the other wing of the strange attractor while ODE23 remains on the initial wing. By the time we are at step $(3.0)^4$ we can see complete divergence of the respective solutions.

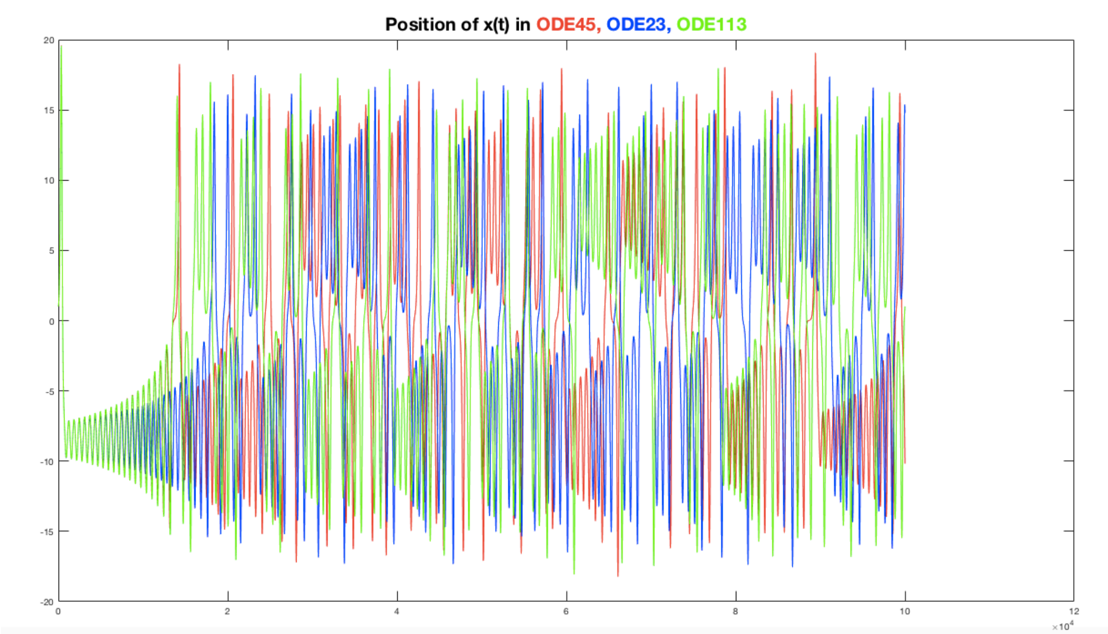


Fig. 15 Position of x-value for solvers: *ODE45* (red), *ODE23* (blue) and *ODE113* (green)

With the pitfalls of computer assisted numerical methods in mind there was a concern over the true existence of the Lorenz attractor. It was ultimately Warwick Tucker in his doctoral thesis in 2002 that proved the existence of the Lorenz attractor in a computer assisted proof [5].

A background of the problems comes with the notation of numerical analysis indicating the existence of the forward invariant open set U, which contains the origin, but being bounded away from the equilibrium points we have described in $C^{\pm}$. We let φ denote the flow of the Lorenz system of equations and we can define the maximal invariant set

$$\mathcal{A} = \bigcap_{t \geq 0} \varphi(U, t). \tag{18}$$

As the flow on the attractor is dissipative, the attracting set $\mathcal{A}$ must have zero volume. The set $\mathcal{A}$ must also contain the unstable manifold $W^u(0)$, which spirals around the equilibrium points $C^{\pm}$ in complex and non-periodic fashion and this set appears to have strong hyperbolic structure.

Being that it was difficult to glean exact information from the differential equations of the attracting set $\mathcal{A}$, Guckenheimer and Williams in 1979 introduced a geometric model of the Lorenz attractor as a tool to study the system. In their model, the origin plays an important role as being the only equilibrium that point orbits seem to flow close to. It is near the origin that the point flows seem to behave in a relative linear fashion akin to our linearization of the dynamics we performed earlier. There are two manifolds that are in the vicinity of the origin, a two-dimensional stable manifold $W^s(0)$ consisting of points that converge on the origin and a one-dimensional unstable manifold $W^u(0)$ consisting of points that stem from the origin when followed backwards. Thru analysis of the negative eigenvalues of the stable manifold, the point flows are pulled toward the origin in a cusp-like path and then pulled away along the direction of the unstable manifold.

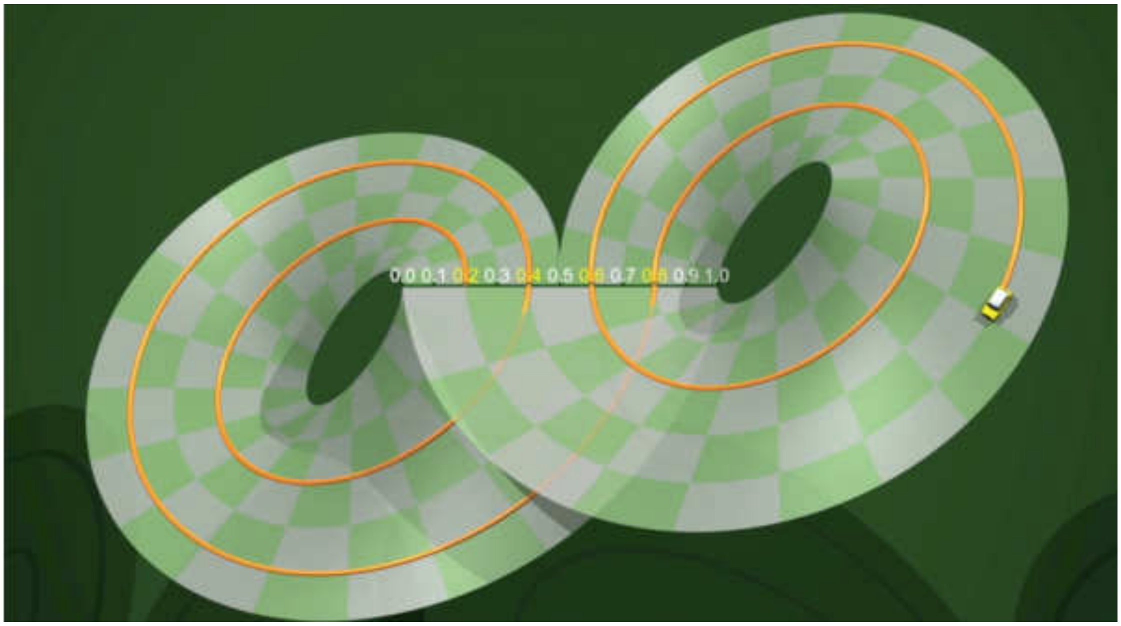


Fig. 16 – Aperiodic nature of orbits thru the "starting line" at intersection of manifolds $W^s(0)$ and $W^u(0)$.

In **Fig. 16,** from the chaos-math.org website, we see an excellent image that helps the reader to understand the non-periodic dynamics of the geometric Lorenz model. In it we see orbital trajectories that pass thru the origin as described above. A "starting line" is drawn along the area of the intersection of the folded halves of the attractor and labeled from 0.0 to 1.0. A point flow approaches the "starting line" and hits at the point 0.2 on the line. The next orital pass hits at twice this value and as such, passes at 0.4. For the subsequent pass, our point will hit the "starting line" at twice this value in 0.8, but as this is past the half way point on the line that is 0.5, it jumps to the other wing of the attractor. The "starting line" behaves as modulus 1, so the next crossing of the flow will intersect at 0.6 on the "starting line", and stay on the same wing with the successive orbit. To put this into an abstract mathematical form we can express this as

$$P_{n+1} = [2P_n]\ modulo\ 1. \tag{19}$$

Where P_{n+1} is the next position where the point crosses the "starting line" and P_n is the position from the prior orbit's crossing.

It was Warwick Tucker that ultimately proved the existence of the Lorenz strange attractor. He was able to overcome the obstacles of numerical approaches to the problem by first introducing a local change of coordinates and then was able to prove the following properties on the return map R as follows [6]:

- *There exists a compact set $N \subset \Sigma$ such that $N \backslash \Gamma$ is forward invariant under R, i.e., $R(N \backslash \Gamma) \subset int(N)$. This ensures that the flow has an attracting set $\mathcal{A}$ with a large basin of attraction. We can then form a cross-section of the attracting set: $\mathcal{A} \cap \Sigma = \bigcap_{n=0}^{\infty} R^n(N) = \Lambda$.*

- *On N, there exists a cone field $\mathcal{C}$, which is mapped strictly into itself by DR, i.e., for all $x \in N, DR(x) \cdot \mathcal{C}(x) \subset \mathcal{C}\big(R(x)\big)$. The cones of $\mathcal{C}$ are centered along two curves with approximate Λ, and each cone has an opening of at least* 5 *degrees.*

- *The tangent vectors in $\mathcal{C}$ are eventually expanded under the action of DR: there exists $C > 0$ and $\lambda > 1$ $such\ that\ for\ all\ v \in \mathcal{C}(x), x \in N, we\ have$ $|DR^n(x)v| \geq \mathcal{C}\lambda^n|v|, n \geq 0$. In fact, the expansion is strong enough to ensure that R is topologically transitive on Λ.*

His proof was thereafter broken down into two main parts: a global one that involves finding enclosure to ODE solutions, and a local one that is based on normal form theory. The details of the proof are beyond the scope of this paper but I encourage the audience to read Tucker's "A Rigorous ODE Solver and Smale's 14th Problem" if interested.

With the preceding Tucker was able to prove that the Lorenz equations did indeed support the strange attractor as surmised by Lorenz in 1963.

Conclusion

The Lorenz system marked the start of the study of chaotic nonlinear dynamics. The ideal deterministic systems of Newton and Laplace, where the future can be determined from the present state with enough computational power, have been put to the test. Minimal changes in states can cause exponential changes as the time evolves in the system. The strange attractor of the Lorenz system is as beautiful as it is profound. In a finite volume, there lies an aperiodic infinite fractal set that takes the shape of the butterfly as the systems flows evolve. The study of chaotic systems continues to evolve rapidly and is taking mathematics and physics in exiting new territories.

References

1. Lorenz, E. N., “Deterministic Non-Periodic Flow”, J. Atmos. Sci., 1963, Vol. 20, pp. 130-141

2. Ekola, T., “A Numerical Study of the Lorenz and Lorenz-Stenflo Systems”, Stockholm: Matematik, 2002, 150 p

3. Ruelle, David; Takens, Floris, “On the Nature of Turbulence”, *Communications in Mathematical Physics, 1971*. **20** (3): 167–192.

4. Gukenheimer, J.: Williams R.F., “Structural Stability of Lorenz Attractors”, Publ. Math. IHES, 1979, Vol. 50, pp. 307-320

5. Tucker,W., “A Rigorous ODE Solver and Smale’s 14th Problem”, Foundations of Computational Mathematics, 2002, Vol. 2, No. 1, pp. 53-117

6. Tucker, Warwick. “The Lorenz Attractor Exists – An Auto-Validated Proof.” (2002).

7. Wolf, A.; Swift, J.; Swinney, H.; Vastano, J., “Determining Lyapunov Exponents from A Time Series”,1985, Physica 16D, pp. 285-317

8. Grebogi, C.; Ott, E.: Yorke, J.; “Chaos, Strange Attractors, and Fractal Basin Boundaries in Nonlinear Dynamics”, Science, 1987, Vol. 238, pp. 632-638

Appendix – MATLAB Code

LorenzSA.m

```
%Lorenz Strange Attractor – Program to make nice plot of the system
%Jason Glowney

%Constant Values
sigma=10;beta=8/3;rho=28;

%Random value generator from -1 to 1
r=-1+(1+1)*rand(10,1);
%Number of Interations
iter=250;
%Setting up 10 points with the random x values
point1=[r(1) 0 0];
point2=[r(2) 0 0];
point3=[r(3) 0 0];
point4=[r(4) 0 0];
point5=[r(5) 0 0];
point6=[r(6) 0 0];
point7=[r(7) 0 0];
point8=[r(8) 0 0];
point9=[r(9) 0 0];
point10=[r(10) 0 0];


%Differential Equations:
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -beta*a(3) + a(1)*a(2)];

%Point Iterations
[t1,a1] = ode45(f,[0 iter],point1);
[t2,a2] = ode45(f,[0 iter],point2);
[t3,a3] = ode45(f,[0 iter],point3);
[t4,a4] = ode45(f,[0 iter],point4);
[t5,a5] = ode45(f,[0 iter],point5);
[t6,a6] = ode45(f,[0 iter],point6);
[t7,a7] = ode45(f,[0 iter],point7);
[t8,a8] = ode45(f,[0 iter],point8);
[t9,a9] = ode45(f,[0 iter],point9);
[t10,a10] = ode45(f,[0 iter],point10);

%Plotting vectors
figure
plot3(a1(:,1),a1(:,2),a1(:,3)) %plotting point 1
hold on;
plot3(a2(:,1),a2(:,2),a2(:,3)) %plotting point 1
plot3(a3(:,1),a3(:,2),a3(:,3)) %plotting point 2
plot3(a4(:,1),a4(:,2),a4(:,3)) %plotting point 3
plot3(a5(:,1),a5(:,2),a5(:,3)) %plotting point 4
plot3(a6(:,1),a6(:,2),a6(:,3)) %plotting point 5
plot3(a7(:,1),a7(:,2),a7(:,3)) %plotting point 6
plot3(a8(:,1),a8(:,2),a8(:,3)) %plotting point 7
plot3(a9(:,1),a9(:,2),a9(:,3)) %plotting point 8
plot3(a10(:,1),a10(:,2),a10(:,3)) %plotting point 9
set(gcf,'color','black');
set(gca,'color','black');
box off;
axis off;
```

Lorenzattractor.m

```
%Lorenz Strange Attractor – program to study dynamics of the system
%user can change variables as desired in Constant Values
%Jason Glowney
clear all;
format long;
%Constant Values
sigma=10;beta=8/3;rho=24.2;
point1=[1 1 1];point2=[0.99999 1 1];
iter=200;

%Differential Equations:
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -beta*a(3) + a(1)*a(2)];

%Point Iterations
```

```
[t1,a1] = ode45(f,[0 iter],point1);%first point
[t2,a2] = ode45(f,[0 iter],point2);%'second point

%Plotting vectors
%ODE45 Plot
figure
plot3(a1(:,1),a1(:,2),a1(:,3)) %plotting point 1
hold on;
plot3(a2(:,1),a2(:,2),a2(:,3)) %plotting point 2
set(gcf,'color','white');
set(gca,'color','white');
% box off;
% axis off;
title('\color{black}Initial Conditions Sensitivity - \color{red}Point1 (1,1,1) \color{black}and
\color{blue}Point2 (0.99999,1,1)','FontSize',20)
plot3(a1(end,1),a1(end,2),a1(end,3),'rd','MarkerSize',20,'LineWidth',5)%ODE45 endpoint
plot3(a2(end,1),a2(end,2),a2(end,3),'bd','MarkerSize',20,'LineWidth',5)%ODE23 endpoint
xlabel('\color{black}X axis');ylabel('\color{black}Y axis');zlabel('\color{black}Z axis')


[sa1,n1]=size(a1);
[sa2,n2]=size(a2);

for i1=1:sa1
     d45point1(i1,:)=sqrt((a1(i1,1)-0)^2+(a1(i1,2)-0)^2+(a1(i1,3)-0)^2);
end
for i2=1:sa2
     d45point2(i2,:)=sqrt((a2(i2,1)-0)^2+(a2(i2,2)-0)^2+(a2(i2,3)-0)^2);
end
[ax1,nx1]=size(a1);
[ax2,nx2]=size(a2);

%Difference for x(t) point1 vs point2
for x1=1:ax1
     point1x(x1,1)=a1(x1,1);
end
for x2=1:ax2
     point2x(x2,1)=a2(x2,1);
end

%Difference for y(t) point1 vs point2
[ay1,ny1]=size(a1);
[ay2,ny2]=size(a2);

for y1=1:ay1
     point1y(y1,1)=a1(y1,2);
end
for y2=1:ay2
     point2y(y2,1)=a2(y2,2);
end

%Difference for z(t) point1 vs point2
[az1,nz1]=size(a1);
[az2,nz2]=size(a2);

for z1=1:az1
     point1z(z1,1)=a1(z1,3);
end
for z2=1:az2
     point2z(z2,1)=a2(z2,3);
end

%Plots
% figure
% plot(d45point1(:,:),'r');
% hold on
% plot(d45point2(:,:),'b');
% title('\color{black}Initial Conditions Sensitivity - \color{red}Point1
\color{blue}Point2','FontSize',20)

figure
plot(point1x(:,1),'r');
hold on
plot(point2x(:,1),'b');
title('\color{black}x(t) vs time','FontSize',20)
set(gcf,'color','white');
set(gca,'color','white');
```

```
%To Calculate Lyapunov Exponents for Lorenz System with (x,y,z)=[1,1,1]
%time = 2000
[T,Res]=lyapunov(3,@lorenz_ext,@ode45,0,0.5,500,[1 1 1],10);
figure
plot(T,Res,'MarkerSize',5);
title('\color{black}Dynamics of Lyapunov exponents','FontSize',20);
xlabel('\color{black}Time','FontSize',15); ylabel('\color{black}Lyapunov
exponents','FontSize',15);
set(gcf,'color','white');
set(gca,'color','white');
```

Programs_14d.m

```
% Programs 14d - Lyapunov exponents of the Lorenz system.
% Chapter 14 - Three-Dimensional Autonomous Systems and Chaos.
% Copyright Springer 2014. Stephen Lynch.

% Special thanks to Vasiliy Govorukhin for allowing me to use his M-files.
% For continuous and discrete systems see the Lyapunov Exponents Toolbox of
% Steve Siu at the mathworks/matlabcentral/fileexchange.

% Reference.
% A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov Exponents from
a Time Series," Physica D,
% Vol. 16, pp. 285-317, 1985.
% You must read the above paper to understand how the program works.

% Lyapunov exponents for the Lorenz system below are:
% L_1 = 0.9022, L_2 = 0.0003, L_3 = -14.5691 when tend=10,000.

function [Texp,Lexp]=lyapunov(n,rhs_ext_fcn,fcn_integrator,tstart,stept,tend,ystart,ioutp);

n=3;rhs_ext_fcn=@lorenz_ext;fcn_integrator=@ode45;
tstart=0;stept=0.5;tend=300;
ystart=[1 1 1];ioutp=10;
n1=n; n2=n1*(n1+1);

%  Number of steps.
nit = round((tend-tstart)/stept);

% Memory allocation.
y=zeros(n2,1); cum=zeros(n1,1); y0=y;
gsc=cum; znorm=cum;

% Initial values.
y(1:n)=ystart(:);

for i=1:n1 y((n1+1)*i)=1.0; end;

t=tstart;

% Main loop.
for ITERLYAP=1:nit
% Solutuion of extended ODE system.
  [T,Y] = feval(fcn_integrator,rhs_ext_fcn,[t t+stept],y);
  t=t+stept;
  y=Y(size(Y,1),:);

  for i=1:n1
      for j=1:n1 y0(n1*i+j)=y(n1*j+i); end;
  end;

% Construct new orthonormal basis by Gram-Schmidt.

  znorm(1)=0.0;
  for j=1:n1 znorm(1)=znorm(1)+y0(n1*j+1)^2; end;

  znorm(1)=sqrt(znorm(1));

  for j=1:n1 y0(n1*j+1)=y0(n1*j+1)/znorm(1); end;

  for j=2:n1
      for k=1:(j-1)
          gsc(k)=0.0;
          for l=1:n1 gsc(k)=gsc(k)+y0(n1*l+j)*y0(n1*l+k); end;
      end;
```

```
      for k=1:n1
          for l=1:(j-1)
              y0(n1*k+j)=y0(n1*k+j)-gsc(l)*y0(n1*k+l);
          end;
      end;

      znorm(j)=0.0;
      for k=1:n1 znorm(j)=znorm(j)+y0(n1*k+j)^2; end;
      znorm(j)=sqrt(znorm(j));

      for k=1:n1 y0(n1*k+j)=y0(n1*k+j)/znorm(j); end;
  end;

%  Update running vector magnitudes.

  for k=1:n1 cum(k)=cum(k)+log(znorm(k)); end;

%  Normalize exponent.

  for k=1:n1
      lp(k)=cum(k)/(t-tstart);
  end;

% Output modification.

  if ITERLYAP==1
     Lexp=lp;
     Texp=t;
  else
     Lexp=[Lexp; lp];
     Texp=[Texp; t];
  end;

  for i=1:n1
      for j=1:n1
          y(n1*j+i)=y0(n1*i+j);
      end;
  end;

end;

% Show the Lyapunov exponent values on the graph.
str1=num2str(Lexp(nit,1));str2=num2str(Lexp(nit,2));str3=num2str(Lexp(nit,3));
plot(Texp,Lexp);
title('Dynamics of Lyapunov Exponents');
text(235,1.5,'\lambda_1=','Fontsize',10);
text(250,1.5,str1);
text(235,-1,'\lambda_2=','Fontsize',10);
text(250,-1,str2);
text(235,-13.8,'\lambda_3=','Fontsize',10);
text(250,-13.8,str3);
xlabel('Time'); ylabel('Lyapunov Exponents');
% End of plot

function f=lorenz_ext(t,X);
%
% Values of parameters.
SIGMA = 10; R = 28; BETA = 8/3;

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

Y= [X(4), X(7), X(10);
    X(5), X(8), X(11);
    X(6), X(9), X(12)];

f=zeros(9,1);

%Lorenz equation.
f(1)=SIGMA*(y-x);
f(2)=-x*z+R*x-y;
f(3)=x*y-BETA*z;

%Linearized system.
 Jac=[-SIGMA, SIGMA,     0;
         R-z,    -1,    -x;
           y,     x, -BETA];
```

```
%Variational equation.
f(4:12)=Jac*Y;

%Output data must be a column vector.

% End of Programs 14d.
```

Lorenzerror.m

```
% Lorenz Attractor Numerical Differences with varius ODE solvers
% Jason Glowney
clear all;
format long;
% Constant Values
sigma=10;beta=8/3;rho=28;
point1=[1 1 1];
tspan=0:0.001:100;

% Differential Equations:
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -beta*a(3) + a(1)*a(2)];

% Point Iterations
[t1,a1] = ode45(f,tspan,point1);%point with ODE45
[t2,a2] = ode23(f,tspan,point1);%point with ODE23
[t3,a3] = ode113(f,tspan,point1);%point with ODE113

%Setup vector for each numerical solver based on size
[ax1,nx1]=size(a1);
[ax2,nx2]=size(a2);
[ax3,nx3]=size(a3);

%Difference for x(t) points for different solvers
for x1=1:ax1
     point1x(x1,1)=a1(x1,1);
end
for x2=1:ax2
     point2x(x2,1)=a2(x2,1);
end
for x3=1:ax3
     point3x(x3,1)=a3(x3,1);
end

% Plotting Strange Attractor with Endpoints
figure
plot3(a1(:,1),a1(:,2),a1(:,3)) %plotting ODE45 point
hold on;
plot3(a2(:,1),a2(:,2),a2(:,3)) %plotting ODE23 point
plot3(a3(:,1),a3(:,2),a3(:,3)) %plotting ODE113 point
plot3(a1(end,1),a1(end,2),a1(end,3),'rd','MarkerSize',20,'LineWidth',5)%ODE45 endpoint
plot3(a2(end,1),a2(end,2),a2(end,3),'bd','MarkerSize',20,'LineWidth',5)%ODE23 endpoint
plot3(a3(end,1),a3(end,2),a3(end,3),'gd','MarkerSize',20,'LineWidth',5)%DOE113 endpoint
set(gcf,'color','white');
set(gca,'color','white');
% box off;
% axis off;
title('\color{red}ODE45 vs \color{blue}ODE23 vs \color{green}ODE113','FontSize',20)
xlabel('\color{black}X Axis','FontSize',15); ylabel('\color{black}Y
Axis','FontSize',15);zlabel('\color{black}Z Axis','FontSize',15)

%Plotting x(t)values for various ODE Solvers
figure
plot(point1x(:,:),'r');
hold on
plot(point2x(:,:),'b');
plot(point3x(:,:),'g');
title('\color{black}Position of x(t) in \color{red}ODE45, \color{blue}ODE23,
\color{green}ODE113','FontSize',20)
set(gcf,'color','white');
set(gca,'color','white');

PointODE45=[a1(end,1);a1(end,2);a1(end,3)]
PointODE23=[a2(end,1);a2(end,2);a2(end,3)]
PointODE113=[a3(end,1);a3(end,2);a3(end,3)]
```

Lorenzzplot.m

```
% Lorenz Attractor Z Dynamics
```

```
% Jason Glowney
format long;
% Constant Values
sigma=10;beta=8/3;rho=28;
point1=[1 1 1];
iter=200;

% Differential Equations:
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -beta*a(3) + a(1)*a(2)];

% Point Iterations
[t1,a1] = ode45(f,[0 iter],point1);%point with ODE45

zpeaks=findpeaks(a1(:,3));

figure
plot(zpeaks(1:end-1),zpeaks(2:end),'b.')
xlabel('\color{black} Z_i','FontSize',20)
ylabel('\color{black} Z_i+1', 'FontSize',20)
set(gcf,'color','white');
set(gca,'color','white');

plot3(a1(:,1),a1(:,2),a1(:,3)) %plotting point 1
figure
plot3(a1(1:50,1),a1(1:50,2),a1(1:50))
```

LorenzSingle.m

```
% Lorenz Attractor Single Point Progression – to study point flows on the system by plotting
%different iterative intervals
% Jason Glowney
format long;
% Constant Values
Origin=[0,0,0];
Cplus=[sqrt(8/3*(27)),sqrt(8/3*(27)),27]
Cminus=[-sqrt(8/3*(27)),-sqrt(8/3*(27)),27]
sigma=10;beta=8/3;rho=28;
point1=[1 1 1];
iter=200;

% Differential Equations:
f = @(t,a) [-sigma*a(1) + sigma*a(2); rho*a(1) - a(2) - a(1)*a(3); -beta*a(3) + a(1)*a(2)];

% Point Iterations
[t1,a1] = ode45(f,[0 iter],point1);%point with ODE45


figure
plot3(Origin(1,1),Origin(1,2),Origin(1,3),'r.','MarkerSize',15)
hold on
plot3(Cplus(1,1),Cplus(1,2),Cplus(1,3),'b.','MarkerSize',15)
plot3(Cminus(1,1),Cminus(1,2),Cminus(1,3),'g.','MarkerSize',15)
plot3(a1(1:2000,1),a1(1:2000,2),a1(1:2000))
title('\color{red}Origin (0,0,0) \color{blue}, C+ (8.485,8.485,27) \color{green}, C- (-8.485),-
8.485,27)','FontSize',20)
xlabel('\color{black} X axis','FontSize',20)
ylabel('\color{black} Y axis', 'FontSize',20)
zlabel('\color{black} Z axis', 'FontSize',20)
set(gcf,'color','white');
set(gca,'color','white');
```

Dlor.m

```
%Program to determine fractal dimension of the Lorenz strange attractor
function ratio=d2lor(iter)
%iter=10000; gives d2=2.0006 a little below the 2.06 actual
dt=0.02;
[x y z]=lorenzo(iter,dt);
d=3;
l2=iter-d;
vec=[x;y;z];
% Now find the maximum and minimum distance apart
% of pairs of points in our d-dimensional space
max = 0;
min = 0;
```

```
for i=1:l2
    for j=1:l2
        sum = 0;
        for k = 1:d
            sum = sum + (vec(k,i) - vec(k,j)).^2;
        end
        sum = sqrt(sum);
        if sum > max
            max = sum;
        end
        if i==1 && j==2
            min = sum;
        else
            if (sum < min) && (sum>0)
                min = sum;
            end
        end
    end
end

% Now add up how many pairs of points are distance apart
% closer than epsilon and return epsilon and count in
% the array ratio for later graphical analysis with Excel
scales = 18;
start = 1;
ratio = zeros(2,scales);
n=start;
epsilon = 1/(2^n);
while epsilon*max>2*min && n<scales
    count = 0;
    for i=1:l2
        for j=1:l2
            sum = 0;
            for k = 1:d
            sum = sum + (vec(k,i) - vec(k,j)).^2;
            end
            sum = sqrt(sum);
            if sum < epsilon*max
                count = count + 1;
            end
        end
    end
    ratio(1,n) = epsilon;
    ratio(2,n) = count;
    n=n+1;
    epsilon = 1/(1.5^n);
end
[p q]=size(ratio);
loglog(ratio(1,:),ratio(2,:));
polyfit(log(ratio(1,floor(q/3):ceil(2*q/3))),log(ratio(2,floor(q/3):ceil(2*q/3))),1)
```