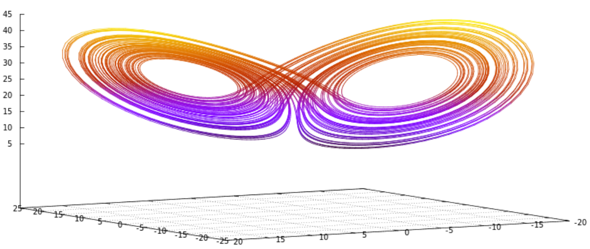
GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Institut für Theoretische Physik
Dr. Claus Heussinger / Dr. Joerg Malindretos

Simulation and Analysis of the Lorenz System

Nonlinear Dynamics and Chaos



Term paper by

Tobias Wegener
✉ Tobias.Wegener@stud.uni-goettingen.de

Computergestütztes Wissenschaftliches Rechnen II
SoSe 2013

Contents

1 Introduction

The **Lorenz system** is a coupled system of three nonlinear differential equations that were derived first by *Ed Lorenz* in 1963. He intended to find a model that is able to describe the behavior of convection currents. Because of the complexity of this issue drastic simplifications were required. The analysis of this problem leads to several aspects of **chaos theory**.

2 Theory

2.1 Lorenz Equations

The equations found by Lorenz can be written as follows:

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= 10\,(y-x)\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= -xz+rx-y\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\frac{8}{3}z\end{aligned}$$

Here r is a parameter that is linked to the **Rayleigh number**. In simplified terms the x value is related to the convection velocity, y to the temperature difference between the increasing and the decreasing flow and z to a nonlinear impact on the temperature gradient [John Argyris, 2010, S. 479 f.]. It is remarkable that the equations are symmetric in x and y since they remain the same if x and y were replaced by $-x$ and $-y$.

2.2 Phase Space, Trajectory and Attractors

The state of the above system at the time t can be illustrated by a single point in space. The point then has the components x, y and z and the space is called **phase space**. Over time the state of the system changes which corresponds to a movement of the point in the phase space along a curve that is called **trajectory** [Meschede, 2010, S. 215]. When t tends to infinity, the trajectory might approach a so-called **attractor**. That can be a geometric object like a simple set of points, a curve or a manifold. However, sometimes this attractor consists of a more complicated set that cannot be described by classical geometry because its **Hausdorff dimension** is not an integer. Those objects are therefor called **fractals**. If this applies to the attractor, it is called a **strange attractor** [Strogatz, 2001, S. 317 ff.].

2.3 Deterministic Chaos

Deterministic Chaos is characterized by the fact that long term predictions are impossible even though the next state of the system is uniquely defined by the current one. That requires a nonlinear system and **sensitive dependence on the initial conditions**. This means that minimal changes in these conditions will cause the trajectories to drift apart exponentially. But even though the trajectories will diverge very fast, they can still approach the same attractor in case it is a strange one. Then the maximum distance is limited to the diameter of the attractor and eventually saturation occurs [Meschede, 2010, S. 240 f].

3 Task Formulation

3.1 Writing the Program

First a program has to be written that integrates the differential equations by using the Runge-Kutta 4th order method. It should enable the user to enter the initial conditions as well as the simulation parameters via the command line. Furthermore, it should naturally be able to execute the simulation and write the results to a file.

When this works, the program should be extended that way that two Lorenz system objects with different initial conditions can be created and compared. Therefor a routine should be written that computes the distance between these systems as a function of the elapsed time.

3.2 Presentation and Interpretation of the Results

When the program runs properly, the results should be visualized by plotting the data. It is suggested to plot the phase space to receive an impression of the shape of the trajectory as well as to plot the time dependency of the components for more accurate considerations. Furthermore, it should be ascertained how sensitive the system reacts on small changes in the initial conditions. Therefor the distance of two different trajectories should be plotted against the time for decreasing initial distances. Ultimately, one should find appropriate ways of presentation of the remarkable results and interpret them as far as it is possible.

4 Idea and Structure of the Program

First I reflected about the required features of the program and how to implement these best.

I decided to create a class called *LorenzSystem* to bunch all information and methods that characterize such a system. These are for example the initial conditions (x_0, y_0 and z_0), the simulation parameters (r, increment h and number of steps n) and the way of how x, y and z are related (function to compute the Lorenz equations). Furthermore, a few public functions (to initialize the system, get the values of the private members and of course to simulate the system) were needed.

To integrate the Lorenz equations I used the Runge-Kutta 4th order method which works as follows:

$$\begin{aligned}
x_{i+1} &= x_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\
k_1 &= F\left(t_i, x_i\right) \\
k_2 &= F\left(t_i + \frac{h}{2},\, x_i + k_1\,\frac{h}{2}\right) \\
k_3 &= F\left(t_i + \frac{h}{2},\, x_i + k_2\,\frac{h}{2}\right) \\
k_4 &= F\left(t_i + h,\; x_i + \, k_3\right)
\end{aligned}$$

where x_i is the current position, x_{i+1} the position after one more time increment and k_1 - k_4 are estimators for the average slope in this interval. The function F represents the derivative of x with respect to t. The current state of a system must be given to be able to apply this method. The slope of the function between the positions x_i and the x_{i+1} is estimated by four different values that will finally be averaged (weighted). The product of this value and the increment h added to the value of x_i should then be an appropriate estimation for x_{i+1}. [Yang, 2000, S. 197]

On top of the first class I created a second one called *LorenzPair* that owns among others two objects of the type *LorenzSystem*. It should cluster those methods that are required to compare the trajectories of these systems. First I had to decide which initial positions should be chosen for them. I did not want to predetermine the initial positions arbitrarily because I think for numerical studies it is beneficial to generate initial values randomly unless there are restrictions. So I wrote a function that sets the initial position of a *LorenzSystem* object to random values (though in a limited interval). In contrast to that I defined r to be 28 by default as that seems to be a particularly suitable value (for the distance analysis).

A consequence of the random choice of the initial positions was that the halving of the distance was slightly more difficult. I decided to realize that by holding the position of one system constant while the other system would be placed in the center of the previous initial positions.

I do not want to go into more details at this place because they are described fairly extensively in the source code which is attached to this document.

5 Interpretation of the Results

First I want to clarify the following: The results I found do not need to be valid in general. It must be considered that it is impossible to verify an assumption for all possible values and combinations so maybe it was pure chance to find some correlations and regularities. But with an appropriate number of tests the probability (that the founded results are valid) should be fairly reasonable.

5.1 The Lorenz Attractor

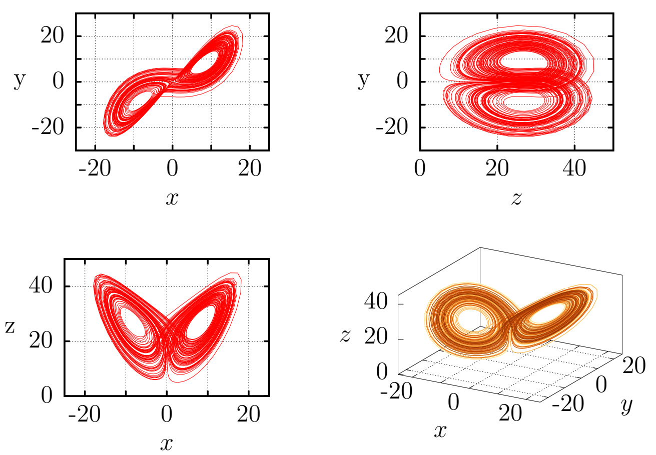


Figure 1: Multi-sided view of a Lorenz system trajectory with the initial position $x_0 = y_0 = z_0 = 1$ and the parameter $r = 28$. The trajectory approaches an object that is reminiscent of butterfly wings or a twisted eight.

As the program was finished, I simulated a Lorenz system exemplarily for the initial position $x_0 = y_0 = z_0 = 1$ and set the parameter r to 28 as suggested. To illustrate the shape of the computed values I decided to plot the trajectory in the phase space as a 3D plot. In doing so I received a spatially limited object that looks more or less regular and resembles a pair of butterfly wings or a twisted eight (see fig. 1).

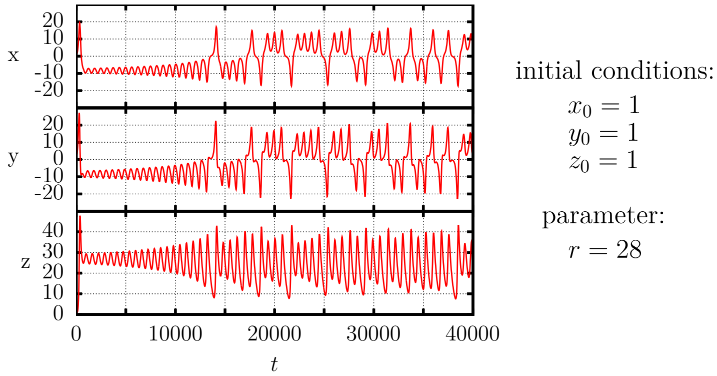


Figure 2: Time dependency of x, y and z. After an initial transient the x and y components oscillate irregular alternating around a positive and a negative value whereas the z component oscillates always around zero.

Then I plotted x, y and z against the elapsed time as presented in fig. 2. By this, the long term behavior of each component can be analyzed more accurately It becomes clear that all of them behave irregularly concerning the time: After an initial transient the x and y components start to oscillate alternating around a positive and a negative value (of similar/equal modulus). The shift of the oscillation from the positive to the negative range and vice versa seems to be fairly irregular. The same applies to the amplitude. The z component behaves in a comparable way even though it always oscillates around zero.

5.2 Variation of the Parameter r

Then I wanted to find out how a variation of the parameter r affects the trajectory shape. First I set r to some negative numbers. The result was that the trajectory crashed more or less straight into the origin no matter which initial conditions I chose. So that might be a fix point for a certain r range.

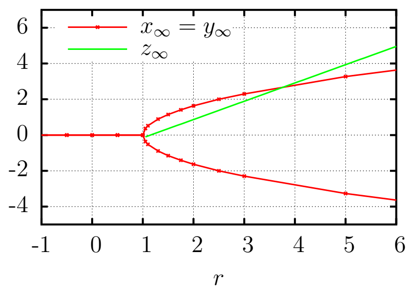


Figure 3: Fix point positions $(x_\infty, y_\infty, z_\infty)$ as a function of r. For $r < 1$ there is only one fix point (the origin). If r becomes greater than one, the fix point divides into two ones that move apart for rising r. The z value increases linearly whereas the x and y values seem to be connected with r by some quadratic relationship.

Subsequently I set r to small positive numbers to delimitate the extent of that range. I recognized that the behavior changed for values greater than one: The fix point started to move apart from the origin with increasing r. On top of that there appeared to be a second fix point for the same value of r which could be hit by modifying the initial position. The places of these fix points turned out to be symmetric: They are on the same z-level but have a reversed sign in the x and y component. But that makes absolutely sense since the Lorenz equations are symmetric in x and y (see sec. 2).

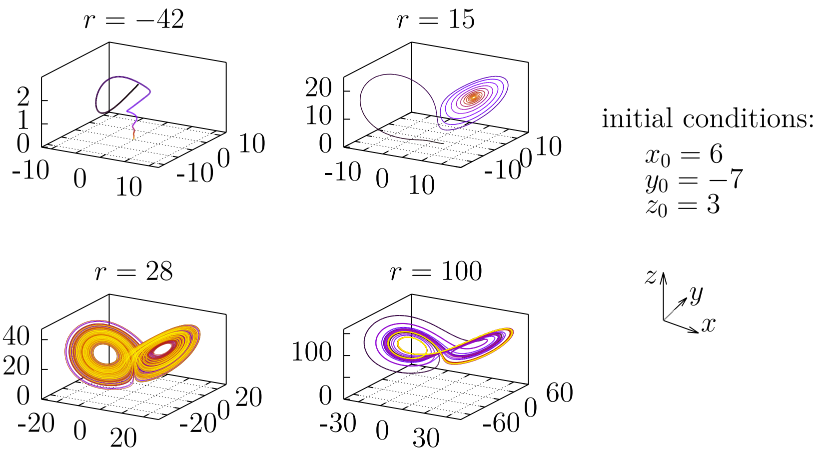


Figure 4: Shape of the attractor depending on the choice of r: For small r (< 1) the trajectory crashes quite fast into the origin. If r is chosen little bigger, it edges its way towards one of two fix points depending on the initial position. For still bigger values of r (roughly > 24) the trajectory approaches a strange attractor whereas its form can still vary for some r ranges. The attractors are indicated by the yellow/orange color.

For values of r laying roughly in the interval from 15 to 24 I found that the convergence of the trajectory's course becomes more slowly. First the trajectory seems to remain on a closed curve like an ellipse. But then it behaves according to one of two different patterns depending on the initial position: It eventually starts to form either a spiral-shaped curve approaching the center of the ellipse (which is a fix point) or one that tends outwards. In the second case the trajectory will break away from the ellipse after some orbits and start to rotate around an opposite one. That means that the trajectory will approach the Lorenz attractor.

When r increases in the stated interval, those trajectories with initial positions close to the origin will approach fairly long one of the fix points. In contrast to this, those trajectories which were quite fast off the origin at the beginning already start to approach the Lorenz attractor for smaller values of r.

If r is big enough (roughly > 24), almost all initial conditions seem to lead to the same strange attractor and one receives objects similar to the one in fig. 1.

Finally I set r to some values between 24 and 200 and received a strange attractor most often even though there were some more or less small intervals in which I came across periodic (non chaotic) results (e.g. for $r = 100$ in fig. 4).

5.3 Sensitivity on the Initial Conditions

Finally I wanted to find out how sensitive the system reacts on minimal changes in the initial conditions. For this purpose I computed and plotted the distance of two trajectories (with different initial positions) as a function of t. I set r to 28 as this value leads to chaotic behavior and the Lorenz attractor. Besides, I reduced the initial distance to the half length a couple of times. In the course of this I found the following things:

1. Both trajectories converge on the same attractor. So the maximum distance of two trajectories (for large values of t) is limited to the diameter of this attractor. That also implies that if the initial distance has a order of magnitude similar to the attractor, its development is not quite that interesting.

2. If however the initial distance is very small compared to the size of the attractor, the trajectories first seem to have an identical course. But after a certain time they start to diverge exponentially fast (on a large scale) even though smaller-scale oscillations superpose this increase in the distance. These small oscillations are the results of the irregular shape of the trajectories while approaching the strange attractor.

Due to these observations I plotted the logarithm of the distance against the elapsed time. The results for some different initial distances are presented in fig. 5 and confirm the findings: After a certain time the logarithmized curve exhibits an almost linear increase (with more or less small oscillations) before eventually saturation arises. It is interesting to note that the slope of the linear part seems to be independent of the initial distance. So for smaller distances saturation will be achieved later.

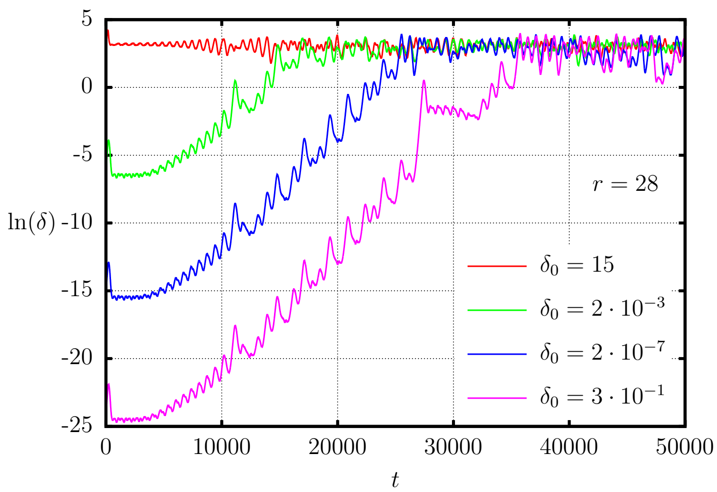


Figure 5: Sensitivity on initial conditions: Development of the distance of two trajectories with various initial distances δ_0. They will start to diverge exponentially fast after a certain time, no matter how close they were to each other at the beginning. As the maximum distance is limited to the diameter of the attractor, eventually saturation occurs.

According to [Strogatz, 2001, S. 321 f.] the average slope of the curve in the linear interval is called **Liapunov exponent**. This value indicates a time horizon beyond which prediction breaks down. In real studies it is impossible to determine the initial conditions exactly, so for chaotic systems one cannot predict future states precisely. If the time t becomes big enough, the predicted behavior will differ completely from the real behavior.

So the system reacts extremely sensitive on small changes in the initial conditions. For a better illustration of this I created one more plot (see fig. 6). Therefor I simulated the Lorenz system for a couple of initial positions that were fairly close to each other (average distance of around 10^{-3}). Then I plotted the distribution of the the resulting trajectory positions for several values of t. It becomes clear that they first remain close to each other but after some time diverge exponentially fast so that they are scattered over the whole attractor for $t > 30000$.

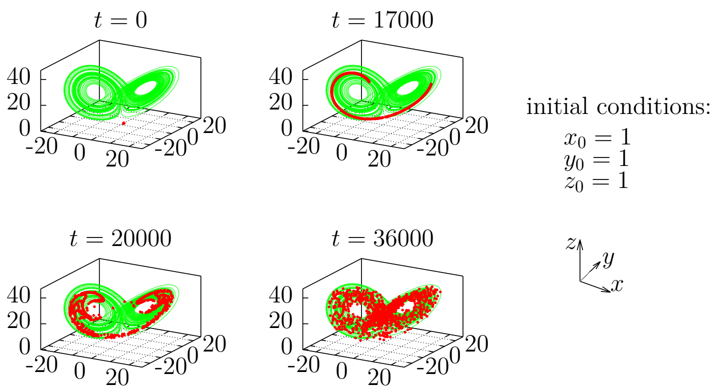


Figure 6: Divergence of trajectories with small initial distance. A red point indicates the position of one trajectory in the phase spase at the stated time t. Although the initial distances were very small (around 10^{-3}) the states in the phase space quickly spread on the whole attractor which is indicated by the green color.

References

Maria Haase Rudolf Friedrich John Argyris, Gunter Faust. *Die Erforschung des Chaos*. Springer, 2 edition, 2010.

Dieter Meschede. *Gerthsen Physik*. Springer-Verlag Berlin Heidelberg, 24 edition, 2010.

Steven H. Strogatz. *Nonlinear Dynamics and Chaos*. Perseus Books Group, 2001.

Daoqi Yang. *C++ And Object-Oriented Numeric Computing for Scientists and Engineers*. Springer, 2000.