58
Lorenz and the Essence of Chaos*

> I am convinced that chaos, along with its many associated concepts - strange attractors, basin boundaries, period-doubling bifurcations and the like - can readily be understood and relished by readers who have no special mathematical or other scientific background...
> (E. Lorenz, in Foreword to *The Essence of Chaos*)

58.1 Introduction

On December 29, 1972, the meteorologist Edward Lorenz presented in a session on the Global Atmospheric Research Program at the 139th meeting of the American Association for the Advancement of Science in Washington D.C., a talk with the title *Predictability: Does the Flap of a Butterfly's Wings in Brazil Set off a Tornado in Texas?* The talk by Lorenz with its "Butterfly effect" rocketed to fame a decade later during the development of "Chaos Theory" that became a fashion in mathematics and physics during the 80s, with the pretention of explaining a variety of phenomena from turbulent fluid flow to collapsing stock markets sharing qualities of *unpredictability.* A decade earlier, "Catastrophe Theory" played a similar role, while today very few remember this intriguing subject. Of course, unpredictability or "chaos" is a phenomenon that has long been familiar to mankind. The word "chaos" comes from early Greek cosmology and signifies the complete lack of order of the Universe before the creation of Gaea and Eros (Earth and Desire).

Lorenz' question is connected to the obvious difficulty of making reasonably reliable predictions of the daily weather over longer time than a week. A weather forecast is made by numerically solving an IVP modeling the evolution of the atmosphere, including variables such as temperature, wind speed and pressure. There are many sources of errors in a weather forecast made this way: errors in the initial value, modeling errors and numerical errors, and it seems that these errors are magnified at a rate that limits the predictions, depending on the scale from a few hours in very local models to weeks in global circulation models.

Lorenz' Butterfly analogy indicates that in certain dynamical systems, very small causes may have large effects after some time. We have already met such a problem in the form of a pendulum being released starting from the unstable top position: depending on the initial perturbation the position of the pendulum will be vastly different after some time (one side or the other). In meteorology, this corresponds to a situation where the weather-man can't say if a certain low pressure will take this way or that way, and thus can't be sure if it will rain in Göteborg tomorrow or not. In his book, Lorenz gives other examples of unstable systems such as a pinball machine, where very small changes in the action of the player can change the outcome of the game completely. Of course there are many other examples from real life of "small" causes having large effects, from soccer games to the assassination of Archduke Francis Ferdinand by the Serb nationalist Gavrilo Princip in Sarajevo on June 28 1914, initiating the First World War.

58.2 The Lorenz system

Lorenz formulated an IVP of the form $\dot{u} = f(u)$ with $f : \mathbb{R}^3 \to \mathbb{R}^3$ given by

$$f(u) = (-10u_1 + 10u_2, 28u_1 - u_2 - u_1u_3, -\frac{8}{3}u_3 + u_1u_2),$$

which is the famous *Lorenz system*. Lorenz found that the solution of this system is very sensitive to perturbations. The system has some vague connection to a very simple model for fluid flow and has been given the role of explaining properties of fluid motion, such as turbulence. This was not Lorenz' original idea, who just wanted to make a connection to the apparent unpredictability and supposed sensitivity to perturbations of common meteorological models. If the seemingly very harmless and innocent Lorenz system could have unpredictable solutions, then there should be no surprise that also the weather could be unpredictable.

More precisely, Lorenz found that two solutions of the Lorenz system with very close initial data will stay close for some time but will eventually move apart completely. The Lorenz system is therefore very difficult to solve accurately using a numerical method over times longer than say 30 units. The numerical solution will stay close the the exact solution for some time, but will eventually move apart significantly. Of course there are many IVP:s sharing this property of instability. Even the simple pendulum has this property if the pendulum reaches the top position with small velocity. It is thus remarkable that the Lorenz system seemed to present some kind of surprise to the scientific world. But it did, and it has become quite popular to explain all sorts of phenomena, from turbulence to politics, by referring to the “strange attractor” supposedly being displayed in plots of solutions of the Lorenz system.

The Lorenz system in component form reads:

$$\begin{cases} \dot{u_1} = -10u_1 + 10u_2, \\ \dot{u_2} = 28u_1 - u_2 - u_1u_3, \\ \dot{u_3} = -\frac{8}{3}u_3 + u_1u_2, \\ u_1(0) = u_{01},\ u_2(0) = u_{02},\ u_3(0) = u_{03}, \end{cases} \tag{58.1}$$

and u_0 is a given initial condition. The system (58.1) has three equilibrium points $\bar{u}$ with $f(\bar{u}) = 0$: $\bar{u} = (0,0,0)$ and $\bar{u} = (\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$. The equilibrium point $\bar{u} = (0,0,0)$ is unstable with the corresponding Jacobian $f'(\bar{u})$ having one positive (unstable) eigenvalue and two negative (stable) eigenvalues. The equilibrium points $(\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$ are slightly unstable with the corresponding Jacobians having one negative (stable) eigenvalue and two eigenvalues with very small positive real part (slightly unstable) and also an imaginary part. More precisely, the eigenvalues at the two non-zero equilibrium points are $\lambda_1 \approx -13.9$ and $\lambda_{2,3} \approx .0939 \pm 10.1i$.

In Fig. 58.1, we present two views of a solution $u(t)$ that starts at $u(0) = (1,0,0)$ computed to time 30 with an error tolerance of $TOL = 0.5$ using an adaptive IVP-solver of the form presented in Chapter *Adaptive IVP-solvers*. We can think of $u(t) = (x(t), y(t), z(t))$ as the position at time t of a particle that moves according to the equation $\dot{u} = f(u)$. In Fig. 58.1 we thus plot the trajectory or path followed by the particle as the particle moves with increasing time. The plotted trajectory is typical: the particle is kicked away from the unstable point $(0,0,0)$ and moves towards one of the non-zero equilibrium points. It then slowly orbits away from that point and at some time decides to cross over towards the other non-zero equilibrium point, again slowly orbiting away from that point and coming back again, orbiting out, crossing over, and so on. This pattern of some orbits around one non-zero equilibrium point followed by a transition to the other non-zero equilibrium point is repeated with a seemingly random number of revolutions around each equilibrium point.

As noted by Lorenz, a close inspection of the trajectory in Fig. 58.1 reveals quite a bit of structure in the behavior of the solution. From the path of the trajectory, it seems that, roughly speaking, there are two flat “lobes” in which the orbits around the non-zero equilibrium points are

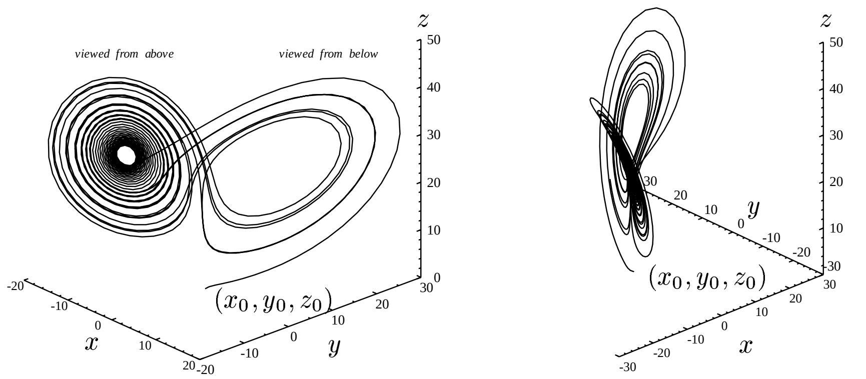


FIGURE 58.1. Two views of a numerical trajectory of the Lorenz system over the time interval $[0, 30]$ starting at $(1, 0, 0)$ computed with absolute error tolerance 0.5.

located. In each lobe, the spiraling segments of the trajectory seem to be grouped in "bands" that are made up of parts of the trajectory that are spiralling out from the equilibrium point and parts of the trajectory that have just crossed over from the other lobe. Only the trajectories in the outer band switch to the other equilibrium point. This causes a sharp separation between trajectories located in the outer band and those located in the next band inside as the trajectories approach the z-axis. We refer to this as *cutting* through the action of a "razor" separating the trajectories in the outer band. The trajectories in the outer band expand in width as they approach the other equilibrium point, with trajectories near the outside of the band ending up nearer to the fixed point. We refer to this as *expansion* and *flipping* respectively. The position of initial approach of the trajectory to an equilibrium point determines the number of orbits the trajectory makes in that lobe before returning to the other equilibrium point. Finally, we see that the orbits in one band come close to the next outer band after one revolution, this repeats with every band of the trajectory, until eventually they all end up in the outer band and leave towards the other equilibrium point. We refer to this as *interlacing*. In short, we can describe the dynamics of the Lorenz system as a never-ending process of cutting, expansion, flipping, and interlacing.

58.3 The accuracy of the computations

The first task is to measure the reliability of the computed error bound based on an a posteriori error estimate of the form presented in Chapter *Adaptive IVP-solvers.* Since we do not have the exact solution, we perform the following experiment: Using the initial data $(0, 1, 0)$, we compute twice using residual tolerances 10^{-5} and 10^{-9} and approximate the error in the less accurate computation by taking the difference between the values of the less accurate and more accurate computations. In Fig. 58.2, we plot the computed error bound and the approximate error. The error bound predicts the size of the error quite well in spite of the sensitivity of the solution to perturbations. Similar results are obtained for a variety of initial data.

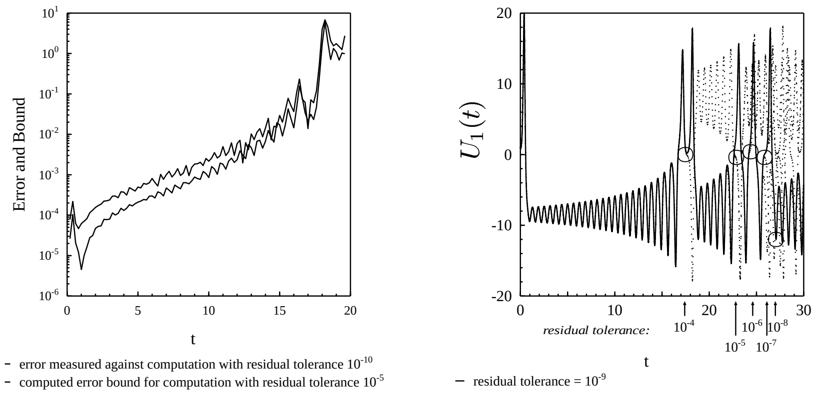
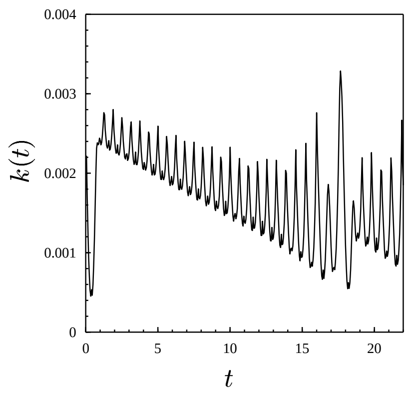


FIGURE 58.2. On the left, we plot the results of the reliability test for the computed error bound with initial condition is $(0, 1, 0)$. On the right, we plot the effect of changing the residual tolerance on the accuracy in $U_1(t)$ component with initial condition is $(1, 0, 0)$.

To give some idea of the behavior of the error control, we plot the step sizes used in a computation with absolute error tolerance 0.75 in Fig. 58.3. The step sizes vary roughly by a factor of 6 over the interval of computation. In Fig. 58.3, we also plot the product of the time step and the residual for this computation. We note that these values are kept within 10% of a constant value. With more computational work, the size of the variations can be reduced, which produces a more smoothly-varying error bound.



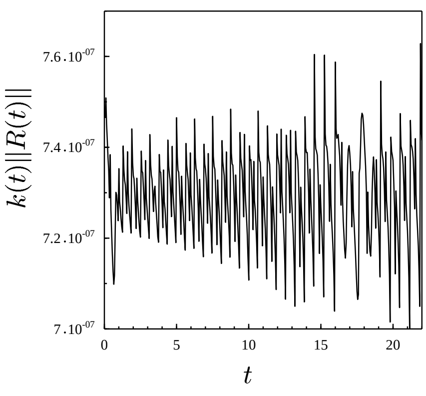


FIGURE 58.3. Time steps and residual $\times$ time step versus time for a computation beginning at $(1, 0, 0)$ with absolute error tolerance 0.75.

58.4 Computability of the Lorenz system

Encouraged by these results we decrease the tolerance or, equivalently, the time step, and try to compute an accurate solution to the Lorenz system on an even longer time interval. Using the cG(1) method as described in Chapter *Adaptive IVP-solvers*, we compute solutions with smaller and smaller time steps, $k = 0.01$, $k = 0.001$ and $k = 0.0001$, and expect to produce more and more accurate solutions. We plot the U_1-component of the solution in Fig. 58.4 where we also indicate the points at which the solutions are no longer accurate. We see that even with $300,000$ time steps the solution is not accurate beyond $t = 26$. Decreasing the time step with a factor 10 or 100 will take us only a little further, but the computation will take 10 or 100 times longer. We conclude that it is difficult to compute the solution to the Lorenz system over long time intervals.

To examine in detail the computability of the Lorenz system we return to the error estimate that we derived for the error $e(t)$ of the cG(1) method:

$$\|e(t)\| \leq S_c(T) \max_{0 \leq t \leq T} \|k(t)R(t)\|. \tag{58.2}$$

Remember that the stability factor $S_c(T)$ for the Lorenz system is defined in terms of the solution to the linearized dual problem as

$$S_c(T) = \max_{\varphi_0 \in \mathbb{R}^3} \frac{\int_0^T \|\dot{\varphi}(t)\| \, dt}{\|\varphi_0\|}.$$

Judging by the error estimate we should be able to reach as far as we want if only the time step $k(t)$ and the residual $R(t)$ are small enough. However,

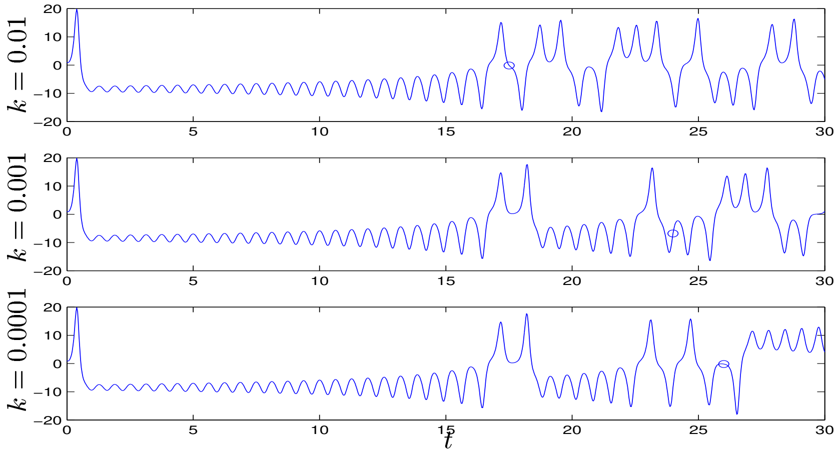


FIGURE 58.4. The U_1-component of the cG(1) solution for different time steps. The small circles indicate the points at which the solutions are no longer accurate.

a little more careful analysis reveals an additional error contribution, which is often ignored. Including also this term into our error estimate, we find:

$$\|e(t)\| \leq S_c(T) \max_{0\leq t\leq T} \|k(t)R(t)\| + S_0(T) \max_{0\leq t\leq T} \epsilon/k(t), \tag{58.3}$$

where ϵ is the *machine precision* of the computer, i.e. the smallest number for which $1+\epsilon \neq 1$ (in computer arithmetic) and $S_0(T)$ is a new stability factor. For a standard computer (in 2002) with so-called double-precision arithmetic, the machine precision is $\epsilon \approx 10^{-16}$. The stability factor $S_0(T)$ is defined in terms of the dual solution as

$$S_0(T) = \max_{\varphi_0 \in \mathbb{R}^3} \frac{\int_0^T \|\varphi(t)\|\, dt}{\|\varphi_0\|}.$$

The additional term in our refined error estimate (58.3) accounts for the round-off error made at every time step in the computation; when the new value $U(t_n)$ for the cG(1) solution is computed in every time step, it is unavoidable that we make a round-off error of size ϵ. As we shall see, it is the second term that sets the limit for the computability of the Lorenz system; the second term in (58.3) can be large even though the first term is small.

The difficulty of computing accurate solutions to the Lorenz system becomes obvious if we plot the size of the stability factors. In Fig. 58.5 we plot the size of the stability factor $S_0(T)$ associated with round-off errors as function of the final time T. Notice the logarithmic scale in this figure. A simple approximation of the growth of this stability factor is

$$S_0(T) \approx 10^{T/3},$$

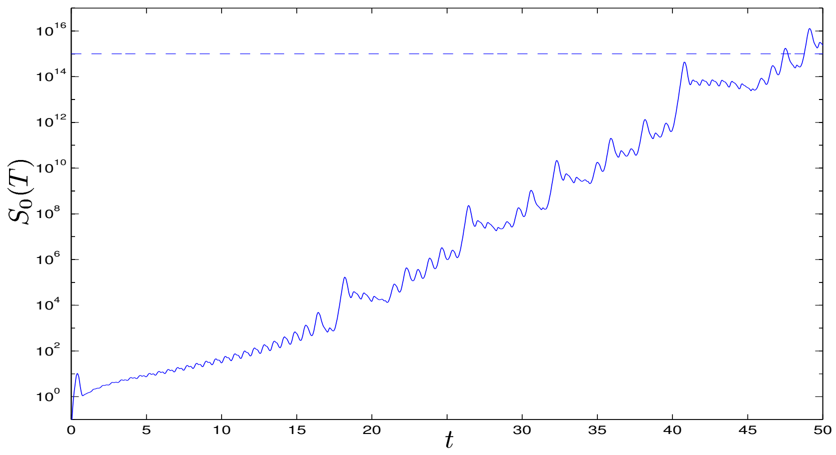


FIGURE 58.5. The growth of the stability factor $S_0(T)$ for the Lorenz problem.

and so the round-off error grows as $E_r = 10^{T/3} \cdot 10^{-16}/k = 10^{T/3-16}/k$. Notice how the error grows larger if we decrease the time step! This is natural (although unusual), since with a smaller time step we will have to take a larger number of time steps and thus make a larger number of round-off errors. To make the influence from round-off errors small we specify a large time step, say $k = 0.1$, for which the round-off error now grows as $10^{T/3-15}$. At time $T = 3 \cdot 15 = 45$ the accumulated round-off error is then $E_r = 1$, which means that we cannot expect to compute much beyond time $T = 45$, since then the round-off error will dominate anyway. Using the cG(1) method, we will not even reach $T = 45$, since we have to use a time step much smaller than $k = 0.1$ (as seen in Fig. 58.4) to make the first term in the error estimate small.

58.5 The Lorenz Challenge

From the previous discussion it is now clear that the mysterious unpredictability and "chaotic" behaviour of the Lorenz system only means that the stability factors grow quickly, making it difficult to compute accurate solutions over long time intervals. The obvious challenge is now, using the method of choice, *to compute an accurate solution to the Lorenz system over as long a time interval* $[0, T]$ *as possible.*

We saw in the previous section that brute force is not the way to go. It is not enough to use a very fast computer with very small and very many time steps. Using the cG(1) method we cannot reach much further than $T = 30$, no matter how small time steps we use since then the accumulated round-off error will grow large. A solution to this problem would be if we could design a method, similar to cG(1), which can be used with larger time steps than what is possible with cG(1). As one can expect, there exist corresponding methods cG(2), cG(3) and so on, which can be used with larger time steps. It can be proved that for these cG(q) methods, the error grows as k^{2q}, i.e. we have so called *a priori* error estimates of the type

$$\|e(T)\| \leq C(T)k^{2q},$$

where $C(T)$ is a constant (unknown!) depending on the exact solution $u(t)$. We say that the cG(q) method is of *order* $2q$. The standard cG(1) method is thus a second order method. (This is in agreement with (58.2) since one factor $k(t)$ is hidden inside $R(t)$.) With a higher order method, i.e. $q > 1$, we can thus obtain a smaller error with the same time step, which makes it possible to compute the solution with larger time steps. This in turn implies that with a higher order method, we can keep the round-off error smaller and thus reach further than what is possible with the cG(1) method. In Fig. 58.6 we plot the U_1-components of solutions to the Lorenz system, computed with time step $k = 0.1$, with a sequence of higher order methods. We see that with high enough order, the solutions agree to a point just beyond $T = 45$ as we predicted; the first term in our error estimate (58.3) has been reduced by increasing the order of the method and so the second term dominates. It is possible to reach beyond time $T = 50$, perhaps to $T = 100$, but to do this we have to go from double-precision arithmetic to quadruple-precision.

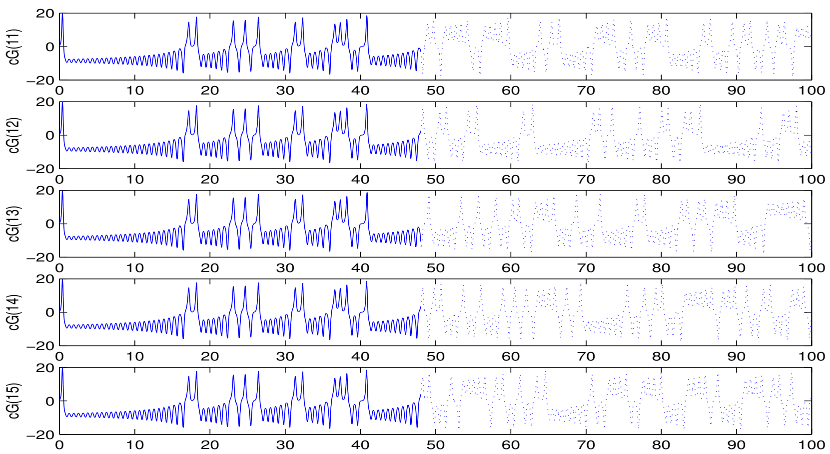


FIGURE 58.6. The U_1-component of cG(q) solutions for $q = 11, 12, 13, 14, 15$ with time step $k = 0.1$. Dashed lines indicate where the solution is no longer accurate.

Chapter 58 Problems

58.1. Verify that the three equilibrium points given in the text satisfy $f(u) = 0$. Linearise the system around these equilibrium points, i.e. compute the eigenvalues (and eigenvectors) for the Jacobian of f at the three equilibrium points.

58.2. Compute a solution to the Lorenz system and plot the orbit $(x(t), y(t), z(t)) = u(t)$ for $t \in [0, T]$. Do you agree with the description of the dynamics of the Lorenz system as never-ending process of cutting, expansion, flipping, and interlacing?

58.3. Repeat the experiment outlined in Section 58.4, i.e. compute solutions to the Lorenz system using the cG(1) method with a sequence of smaller and smaller time steps and examine the accuracy of the solutions (by comparing them to each other). Can you reach beyond $T = 25$?

58.4. Try the same experiment as in the previous problem but now with the lower order methods explicit Euler and implicit Euler. How far do you reach now?

58.5. Implement a simple version of the fourth-order cG(2) method given by

$$\begin{aligned} U(t_{n-1/2}) &= U(t_{n-1}) + \int_{t_{n-1}}^{t^n} f(U(t), t) \cdot (5 - 6(t - t_{n-1})/k_n)/4 \, dt, \\ U(t_n) &= U(t_{n-1}) + \int_{t_{n-1}}^{t^n} f(U(t), t) \, dt, \end{aligned}$$

where $U(t)$ is the quadratic polynomial on $[t_{n-1}, t_n]$ determined by the three values $U(t_{n-1})$, $U(t_{n-1/2})$ and $U(t_n)$. How much further can you reach with this method?

58.6. Give a motivation for the additional term in the refined error estimate (58.3), starting from the estimate containing errors caused by using the wrong initial condition as presented in Chapter *Adaptive IVP-solvers*.

58.7. Take on the Lorenz Challenge, i.e. compute an accurate solution over $[0, T]$ with T as large as possible. No rules, all is allowed!