UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2016:50

Synchronization in the Lorenz system

Max Herrgård

Examensarbete i matematik, 15 hp
Handledare: Denis Gaidashev
Examinator: Jörgen Östensson
November 2016

Department of Mathematics
Uppsala University

Abstract

The system of Lorenz differential equations has been one of the focal problems in dynamical systems. These equations are one of the simplest dynamical systems which exhibit the whole spectrum of dynamical phenomena: regularity (existence of periodic orbits), bifurcations and chaos (existence of a strange attractor).

Several versions of coupled Lorenz equations have been proposed in the literature as a model of signal reception or decoding. We study some of these coupled models, and demonstrate, both numerically, and analytically, that they allow for the so called synchronization - convergence of the solutions of the Lorenz system that models the reciever to those of the Lorenz system that models the sender. In practice, this means that the sender's signal can be decoded by the reciever.

Contents

1. Introduction

Can two chaotic systems, one transmitting and one receiving, synchronize? Is the synchronization fast enough to be usable? If an interference is added to the receiver, will it still sync properly? How much interference is tolerated? The idea for this thesis originate from Steven H. Strogatz's book Nonlinear dynamics and chaos [1].

1.1 History

In his 1963 article [2] Edward N. Lorenz came to the conclusion that long-term weather predictions are impossible mathematically. Lorenz used a system of three ordinary differential equations to approximate mass/heat convection in the atmosphere,

$$\begin{aligned}\dot{X} &= \sigma X - \sigma Y \\ \dot{Y} &= -XZ + rX - Y \\ \dot{Z} &= XY - bZ,\end{aligned}$$

an example of deterministic nonperiodic flow, and used it to simulate airflow. He found through numerical integration that for a certain range of parameters the flow trajectory of a solution in this simple system began to oscillate irregularly in a nonperiodic way. For these parameters the system had no stable fixed points or stable limit cycles and yet the trajectories stayed in a bounded region. He also noticed that the trajectories of starting points close to each other diverged quickly and in finite time.

A solution plotted in phase space shows a set that looks like a thin pair of butterfly wings, what now is called the Lorenz attractor.

1.2 The Lorenz system

The behaviour of the Lorenz system is usually varied by changing the parameter r. For $r < 1$ we have one equilibrium point, the origin, and all trajectories converge to it. At $r = 1$ we have a pitchfork bifurcation and two new equilibrium, C^+ and C^-, appear at $(x, y, z) = (\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1)$, with $x = y$. The origin becomes unstable and the stable points are surounded by an unstable manifold of the origin. At a critical value of r a subcritical Hopf bifurcation happens and C^+ and C^- loses stability. Trajectories are repelled but circling the two points, going back and forth, and this forms the Lorenz attractor [1], a strange attractor of fractal structure. It took a long time until it was proven that the Lorenz attractor really exists, and only numerical solutions was available, but in his 1998 Ph.D. thesis "The Lorenz attractor exists"[3] W. Tucker showed with a computer aided proof that it really exists.

1.3 Chaos

No definition of chaos seem fully accepted, but Devaney [4] gives the following three conditions as essential features. Let X be a metric space and $f : X \to X$ a continuous map. f is chaotic on X if

1. the periodic points of f are dense in X, meaning every point x in X is either a periodic point of f or a limit point of a periodic point,
2. f is transitive, that is for any two open sets U and V in X, $f^k(X) \cap Y \neq \emptyset$, for some non negative integer k and
3. f has sensitive dependence on initial conditions, meaning nearby trajectories separate exponentially fast.

1.4 Synchronized chaos

Synchronized chaos can arise when two identical autonomous chaotic systems are one-way linked with a common signal. The trajectories of one system will converge to the same values as the other, and the two systems will remain in step with each other. The first system, the drive, is used to send a signal to the second one, the response system. Other than that the two systems are not affecting each other in any way. The signal sent from the drive has to be choosen so the synchronization works.

Pecora and Carroll showed in their series of articles, [5], [6], [7], that the Lorenz system could be decomposed into two subsystems and and the x and y variables used as a working drive signal for synchronization.

1.5 Circuit implementation

Since the 1990s people have tried making something useful out of chaos [1]. Cuomo and Oppenheim took Pecora and Carroll's decomposed Lorenz system and combined the two parts [8], getting a receiving (response) system looking similar to the transmitting (drive) system, except $x(t)$ from the drive is used in the $\dot{y}$ and $\dot{z}$ equations of the reciever. By scaling the variables of the Lorenz system and letting $u = \frac{1}{10}x$, $v = \frac{1}{10}y$ and $w = \frac{1}{20}z$, they got a system suitable for an easy electronic implementation.

The drive and response system built was given by

$$\begin{aligned} \dot{u} &= 16(v-u) & \dot{u_r} &= 16(v_r - u_r) \\ \dot{v} &= 45.6u - v - 20uw & \dot{v_r} &= 45.6u(t) - v_r - 20u(t)w_r \\ \dot{w} &= 5uv - 4w & \dot{w_r} &= 5u(t)v_r - 4w_r, \end{aligned}$$

and they used them to transmit a song, with noise added on top, which cleared out so you could hear the song again when the response system synchronized to the transmitting part. They also illustrated that the system was showing chaotic behavior with these parameters.

In Cuomo, Oppenheim and Strogatz's expanded paper on the subject [9], they continue to look at synchronized chaos and private communications as an application. For this idea to work properly they see the need of robust and quick synchronization.

1.6 My approach

I intend to have a closer look at synchronization in the Lorenz system, and more specifically how fast it is. I will start out with Cuomo and Oppenheim's scaled system but write a computer program instead of using electronic components to show the synchronization.

2. Synchronization of the Lorenz system

2.1 Synchronization of two Lorenz systems

The synchronization error for two scaled Lorenz systems, (2.1), transmitting and (2.2), receiving,

$$\begin{aligned}\dot{u} &= \sigma(v-u)\\ \dot{v} &= ru - v - 20uw\\ \dot{w} &= 5uv - bw\end{aligned} \tag{2.1}$$

$$\begin{aligned}\dot{u_r} &= \sigma(v_r-u_r)\\ \dot{v_r} &= ru(t) - v_r - 20u(t)w_r\\ \dot{w_r} &= 5u(t)v_r - bw_r\end{aligned} \tag{2.2}$$

is given by

$$\begin{aligned}\dot{e_1} &= \dot{u} - \dot{u_r} = \sigma(e_2 - e_1)\\ \dot{e_2} &= \dot{v} - \dot{v_r} = -e_2 - 20u(t)e_3\\ \dot{e_3} &= \dot{w} - \dot{w_r} = 5u(t)e_2 - be_3\end{aligned} \tag{2.3}$$

where e_2 and e_3,

$$\begin{aligned}\dot{e_2} &= \dot{v} - \dot{v_r} = -e_2 - 20u(t)e_3\\ \dot{e_3} &= \dot{w} - \dot{w_r} = 5u(t)e_2 - be_3,\end{aligned} \tag{2.4}$$

don't depend on e_1. u of the transmitting system (2.1) is fed into the receiving system (2.2)'s $\dot{v_r}$ and $\dot{w_r}$ equations as $u(t)$. We will quote a fundamental stability theorem for continuous dynamical systems,

Theorem 1 (Lyapunov) *Consider a system $\dot{x} = f(x)$ with a fixed point at*

x^* and suppose that we can find a continuously differentiable, real-valued function $V(x)$ with the following properties:
•$V(x^*) = 0$.
•$V(x) > 0$, *for* $x \neq x^*$.
•$\dot{V}(x) < 0$, *for* $x \neq x^*$.
Then x^* is globally asymptotically stable.

By considering the Lyapunov function

$$V(e_1, e_2, e_3, t) = \frac{1}{2}(\frac{1}{\sigma}e_1^2 + e_2^2 + 4e_3^2),$$

with

$$\dot{V}(e_1, e_2, e_3, t) = -(e_1 - \frac{1}{2}e_2)^2 - \frac{3}{4}e_2^2 - 4be_3^2,$$

we see that
•$V(0, 0, 0, t) = 0$.
•$V(e_1, e_2, e_3, t) > 0$, for $e_1, e_2, e_2 \neq 0$.
•$\dot{V}(e_1, e_2, e_3, t) < 0$, for $e_1, e_2, e_2 \neq 0$.

Lyapunov's theorem (Theorem 1) now states that system (2.3) is asymptotically stable at $(e_1, e_2, e_3) = (0, 0, 0)$, that is, the synchronization error is zero and the two Lorenz systems are synchronized.

2.2 Exponentially fast synchronization of two Lorenz systems

We begin by considering the Lyapunov function $V = \frac{1}{2}e_2^2 + 2e_3^2$, for the decoupled system (2.4) of the synchronization error system (2.3), and looks if $\dot{V} \leq -kV$, for a constant $k > 0$. We have

$$\dot{V} = e_2\dot{e_2} + 4e_3\dot{e_3} = -e_2^2 - 20u(t)e_2e_3 + 20u(t)e_2e_3 - 4be_3^2 = -e_2^2 - 4be_3^2.$$

The chaotic term $u(t)$ from the transmitting system disappears and we have the equation

$$\dot{V} \leq -kV \iff -e_2^2 - 4be_3^2 \leq -\frac{k}{2}e_2^2 - 2ke_3^2,$$

which obviously holds for $k = 2$ with $b \geq 1$. This means $\dot{V} + 2V = 0$ and multiplied by an integrating factor e^{2t}, $\dot{V}e^{2t} + 2Ve^{2t} = 0 \iff \frac{\mathrm{d}}{\mathrm{d}t}(V(t)e^{2t}) = 0$. Changing t to s and integrating this equation yields

$$\int_{t_0}^{t} \frac{\mathrm{d}}{\mathrm{d}s}(V(s)e^{2s})\mathrm{d}t = 0 \iff \left[e^{2s}V(s)\right]_{t_0}^{t} = 0 \iff$$
$$e^{2t}V(t) - e^{2t_0}V(t_0) = 0 \iff e^{2t}V(t) = e^{2t_0}V(t_0).$$

Since $e^{2t_0}V(t_0)$ is constant, say C, for a start time t_0, and $V(t)$ is a Lyapunov function we have $0 \leq V(t) = Ce^{-2t}$, so e_2 and e_3 decay exponentially fast to 0.

For $\dot{e_1}$ we begin by using the same trick multiplying with an integrating factor, in this case $e^{\sigma t}$. Since e_2 was shown above to have an upper bound of $e^{-\gamma t}$, for some $\gamma > 0$, we can use that in the equation below, which means

$$\dot{e_1} = \sigma(e_2 - e_1) \iff \frac{\mathrm{d}}{\mathrm{d}t}(e_1e^{\sigma t}) = \sigma e_2 e^{\sigma t} \Rightarrow$$
$$-\sigma Ce^{-\gamma t}e^{\sigma t} \leq \frac{\mathrm{d}}{\mathrm{d}t}(e_1e^{\sigma t}) = \sigma e_2e^{\sigma t} \leq \sigma Ce^{-\gamma t}e^{\sigma t} \iff$$
$$-\sigma Ce^{(\sigma-\gamma)t} \leq \frac{\mathrm{d}}{\mathrm{d}t}(e_1e^{\sigma t}) \leq \sigma Ce^{(\sigma-\gamma)t}.$$

Changing the variable to s and integrating from $s = 0$ to $s = t$ we get

$$-\int_0^t \sigma Ce^{(\sigma-\gamma)s}\mathrm{d}t \leq \int_0^t \frac{\mathrm{d}}{\mathrm{d}s}(e_1e^{\sigma s})\mathrm{d}t \leq \int_0^t \sigma Ce^{(\sigma-\gamma)s}\mathrm{d}t. \iff$$
$$-\frac{\sigma C}{\sigma-\gamma}e^{(\sigma-\gamma)t} - (-\frac{\sigma C}{\sigma-\gamma}e^0) = -\frac{\sigma C}{\sigma-\gamma}(e^{(\sigma-\gamma)t} - 1) \leq$$
$$e_1(t)e^{\sigma t} - e_1(0)e^0 \leq \frac{\sigma C}{\sigma-\gamma}(e^{(\sigma-\gamma)t} - 1) \iff$$
$$\left(e_1(0) - \frac{\sigma C}{\sigma-\gamma}(e^{(\sigma-\gamma)t} - 1)\right)e^{-\sigma t} \leq e_1(t) \leq \left(e_1(0) + \frac{\sigma C}{\sigma-\gamma}(e^{(\sigma-\gamma)t} - 1)\right)e^{-\sigma t}.$$

Since $e_1(0)$ and $\frac{\sigma C}{\sigma-\gamma}$ are constant, $\sigma > 0$, $e^{(\sigma-\gamma)t}e^{-\sigma t} = e^{-\gamma t}$ and $\gamma > 0$, all terms decay exponentially fast to zero as $t \to \infty$, which means that e_1 decays to zero exponentially fast. Thus all components of the synchronization error system (2.3) decays exponentially fast to $(e_1, e_2, e_3) = (0, 0, 0)$.

2.3 Interference in the receiver

Suppose we introduce an interference to the receiving system. $f(u_r, v_r, w_r, t)$ is added to $\dot{u_r}$, $g(u_r, v_r, w_r, t)$ is added to $\dot{v_r}$ and $h(u_r, v_r, w_r, t)$ is added to $\dot{w_r}$, where $|f, g, h| << |u_r, v_r, w_r|$, so that the new receiving system is given by

$$\begin{aligned}
\dot{u_r} &= \sigma(v_r - u_r) + f(u_r, v_r, w_r, t) \\
\dot{v_r} &= ru(t) - v_r - 20u(t)w_r + g(u_r, v_r, w_r, t) \\
\dot{w_r} &= 5u(t)v_r - bw_r + h(u_r, v_r, w_r, t).
\end{aligned}$$

This means we get a slightly altered system for the synchronization error,

$$\begin{aligned}
\dot{e_1} &= \sigma(e_2 - e_1) - f(u_r, v_r, w_r, t) \\
\dot{e_2} &= -e_2 - 20u(t)e_3 - g(u_r, v_r, w_r, t) \\
\dot{e_3} &= 5u(t)e_2 - be_3 - h(u_r, v_r, w_r, t).
\end{aligned}$$

2.4 Synchronization with interference in the receiver

We still have the Lyapunov function

$$V(e_1, e_2, e_3, t) = \frac{1}{2}(\frac{1}{\sigma}e_1^2 + e_2^2 + 4e_3^2),$$

with

$$\dot{V}(e_1, e_2, e_3, t) = \frac{1}{\sigma}e_1\dot{e_1} + e_2\dot{e_2} + 4e_3\dot{e_3} = -(e_1 - \frac{1}{2}e_2)^2 - \frac{3}{4}e_2^2 - 4be_3^2 - \frac{1}{\sigma}e_1 f - e_2 g - 4e_3 h.$$

Thus we have

• $V(0,0,0,t) = 0$.

• $V(e_1,e_2,e_3,t) > 0$, for $e_1,e_2,e_3 \neq 0$.

For $\dot{V}(e_1,e_2,e_3,t) < 0$, it is sufficient to have $e_1 f > 0$, $e_2 g > 0$ and $e_3 h > 0$. An example of a pertubation which results in stability is

$$\begin{aligned} f &= e_1^{2k+1}\tilde{f}, \\ g &= e_2^{2m+1}\tilde{g}, \\ h &= e_3^{2n+1}\tilde{h}, \end{aligned} \tag{2.5}$$

with $k,m,n \in \mathbb{N}$ and $\tilde{f},\tilde{g},\tilde{h} > 0$ this gives us

$$\begin{aligned} &\dot{V}(e_1,e_2,e_3,t) = \frac{1}{\sigma}e_1\dot{e_1} + e_2\dot{e_2} + 4e_3\dot{e_3} = \\ &-(e_1 - \frac{1}{2}e_2)^2 - \frac{3}{4}e_2^2 - 4be_3^2 - \frac{1}{\sigma}e_1^{2k+2}\tilde{f} - e_2^{2m+2}\tilde{g} - 4e_3^{2n+2}\tilde{h} \end{aligned}$$

and we have

• $\dot{V}(e_1,e_2,e_3,t) < 0$, for $e_1,e_2,e_3 \neq 0$, and the system is asymptotically stable for interference of this type (f, g and h).

2.5 Exponential synchronization with interference in the receiver

We begin again by showing that e_2 and e_3 decay exponenitally by first showing that $\dot{V} \leq -kV$. V is clearly a Lyapunov function. We have $V = \frac{1}{2}e_2^2 + 2e_3^2$ and

$$\begin{aligned} &\dot{V} = e_2\dot{e_2} + 4e_3\dot{e_3} = -e_2^2 - 20u(t)e_2e_3 - e_2 g + 20u(t)e_2e_3 - 4be_3^2 - 4e_3 h = \\ &-e_2^2 - 4be_3^2 - e_2 g - 4e_3 h. \end{aligned}$$

This gives us

$$\dot{V} \le -kV \iff -e_2^2 - 4be_3^2 - e_2 g - 4e_3 h \le -\frac{k}{2}e_2^2 - 2ke_3^2 \iff$$
$$-e_2^2 - 4be_3^2 - e_2 e_2^{2m+1}\tilde{g} - 4e_3 e_3^{2n+1}\tilde{h} \le -\frac{k}{2}e_2^2 - 2ke_3^2 \iff$$
$$-e_2^2(1 + e_2^{2m}\tilde{g}) - 4e_3^2(b + e_3^{2n}\tilde{h}) \le -\frac{k}{2}e_2^2 - 2ke_3^2,$$

which is true if $1 + e_2^{2m}\tilde{g} \ge \frac{k}{2}$ and $4b + 4e_3^{2n}\tilde{h} \ge 2k$, which it is for $k = 2$ since $\tilde{g}, \tilde{h} > 0$ (and we set $b \ge 1$ earlier). This means $\dot{V} \le -2V$, and since V is a Lyapunov function, we can integrate like before and have $0 \le V(t) \le Ce^{-2t}$, so e_2 and e_3 have an upper bound of $e^{-\gamma t}, \gamma > 0$ and thus decay exponentially fast to 0 with an interference $g(u_r, v_r, w_r, t)$ and $h(u_r, v_r, w_r, t)$.

For e_1 we will use the squeeze theorem to show it decays exponentially fast to 0. We have $\dot{e_1} = \sigma(e_2 - e_1) - f(u_r, v_r, w_r, t) \Rightarrow$

$$\dot{e_1} = \sigma(e_2 - e_1) - e_1^{2k+1}\tilde{f}, \tag{2.6}$$

$k \in \mathbb{N}, \tilde{f} > 0$. By omitting the e_2-term we get a differential equation with a solution that has a smaller absolute value than (2.6), $\dot{e_1} = -\sigma e_1 - e_1^{2k+1}\tilde{f} \Rightarrow \dot{e_1} + \sigma e_1 = -e_1^{2k+1}\tilde{f}$, which we write with $y_1 = e_1$,

$$y_1' + \sigma y_1 = -y_1^{2k+1}\tilde{f} \tag{2.7}$$

We have a Bernoulli differential equation, $y' + P(t)y = Q(t)y^n$, with $y = y_1, P(t) = \sigma, Q(t) = \tilde{f}$ and $n = 2k + 1$. Assume that $y_1 \ne 0$. First we multiply both sides of equation (2.7) with $y_1^{-n} = y_1^{-(2k+1)}$ and get $y_1' y_1^{-(2k+1)} + \sigma y_1 y_1^{-(2k+1)} = -y_1^{2k+1}\tilde{f} y_1^{-(2k+1)} \Rightarrow$

$$y_1' y_1^{-(2k+1)} + \sigma y_1^{-2k} = -\tilde{f}. \tag{2.8}$$

Let

$$\nu = y_1^{1-n} = y_1^{1-(2k+1)} = y_1^{-2k},$$
$$\nu' = -2ky_1^{-2k-1}y_1' = -2ky_1^{-(2k+1)}y_1'.$$

Substituting ν into equation (2.8) gives $\frac{\nu'}{-2k} + \sigma\nu = -\tilde{f}$, and by multiplying with $-2k$ we have the linear equation

$$\nu' - 2k\sigma\nu = 2k\tilde{f} \tag{2.9}$$

to solve. By multiplying equation (2.9) with the integrating factor $e^{\int -2k\sigma \mathrm{d}t} = e^{-2k\sigma t}$, we have

$$\nu' e^{-2k\sigma t} - 2k\sigma\nu e^{-2k\sigma t} = 2k\tilde{f}e^{-2k\sigma t} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}(e^{-2k\sigma t}\nu) = 2k\tilde{f}e^{-2k\sigma t}.$$

Integrating both sides and dividing by $e^{-2k\sigma t}$ gets us

$$\frac{1}{e^{-2k\sigma t}}\int \left(\frac{\mathrm{d}}{\mathrm{d}t}(e^{-2k\sigma t}\nu)\right)\mathrm{d}t = \frac{1}{e^{-2k\sigma t}}\int 2k\tilde{f}e^{-2k\sigma t}\mathrm{d}t \Rightarrow$$
$$\nu = \frac{2k}{e^{-2k\sigma t}}\int \tilde{f}e^{-2k\sigma t}\mathrm{d}t.$$

By substituting back $\nu = y_1^{-2k}$ we have

$$y_1^{-2k} = \frac{2k}{e^{-2k\sigma t}}\int \tilde{f}e^{-2k\sigma t}\mathrm{d}t \Rightarrow y_1 = \left(\frac{2k}{e^{-2k\sigma t}}\int \tilde{f}e^{-2k\sigma t}\mathrm{d}t\right)^{-\frac{1}{2k}} \Rightarrow$$

$$y_1 = e^{-\sigma t}\left(2k\int \tilde{f}e^{-2k\sigma t}\mathrm{d}t\right)^{-\frac{1}{2k}}. \tag{2.10}$$

Having $\tilde{f} = e^{\beta t}$ in (2.10) gives us

$$y_1 = e^{-\sigma t}\left(2k\int e^{(\beta-2k\sigma)t}\mathrm{d}t\right)^{-\frac{1}{2k}} = e^{-\sigma t}\left(\frac{2k}{\beta - 2k\sigma}\left(e^{(\beta-2k\sigma)t} + D\right)\right)^{-\frac{1}{2k}} =$$
$$\left(e^{2k\sigma t}\frac{2k}{\beta - 2k\sigma}\left(e^{(\beta-2k\sigma)t} + D\right)\right)^{-\frac{1}{2k}} = \frac{\left(\frac{2k}{\beta-2k\sigma}\right)^{-\frac{1}{2k}}}{(e^{\beta t} + De^{2k\sigma t})^{\frac{1}{2k}}}.$$

where D is an arbitrary integration constant, $\beta < 2k\sigma$ and thus y_1 goes to zero exponentially fast as $t \to \infty$ if $\beta > 0$.

The authors of [10] state that if the coefficients $a(t), b(t), c(t)$ and $d(t)$ of the Abel differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a(t) + b(t)y + c(t)y^{\alpha-1} + d(t)y^{\alpha} \tag{2.11}$$

with $\alpha > 1, \alpha \in \mathbb{R}$ satisfies the conditions

$$c(t) = d(t)e^{\int b(t)\mathrm{d}t}\left[K_1 - (1-\alpha)k_1 \times \int e^{(\alpha-1)\int b(t)\mathrm{d}t} d(t)\mathrm{d}t\right]^{\frac{1}{1-\alpha}}, \tag{2.12}$$

with $\alpha \neq 1$, $k_1 \in \mathbb{R}, k_1 \neq 0$, K_1 an arbitrary constant of integration, and

$$a(t) = k_2 \frac{c^{\alpha}(t)}{d^{\alpha-1}(t)} = -\frac{k_2}{k_1} e^{\int b(t)\mathrm{d}t} \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{c(t)}{d(t)e^{\int b(t)\mathrm{d}t}}\right], \tag{2.13}$$

where $k_2 \in \mathbb{R}, k_2 \neq 0$, then (2.11) is exactly integrable and it's general solution is given by

$$y(t) = \frac{c(t)}{d(t)} s(t), \tag{2.14}$$

where $s(t)$ is a solution of

$$\left|\frac{d(t)e^{\int b(t)\mathrm{d}t}}{c(t)}\right| = K^{-1} e^{F[s(t),k_1,k_2]}, \tag{2.15}$$

$K \neq 0$ an arbitrary integration constant, with

$$F[s(t), k_1, k_2] = k_1 \int \frac{\mathrm{d}s}{s^{\alpha} + s^{\alpha-1} + k_1 s + k_2}. \tag{2.16}$$

We add a term $c(t)y_2^{2k}$ and change sign of $-e_1^{2k+1}$ to positive so that (2.6) for $\dot{e_1}$ would be an Abel differential equation (2.11) with $e_1 = y_2$, $a(t) = \sigma e_2 = \sigma C e^{-\gamma t}$, $b(t) = -\sigma$ and $d(t) = \tilde{f}$ and we get

$$y_2' = \sigma C e^{-\gamma t} - \sigma y_2 + c(t)y_2^{2k} + \tilde{f} y_2^{2k+1}, \tag{2.17}$$

a differential equation with a solution that has a bigger absolute value than (2.6), if $c(t)$ is chosen such that $c(t) > 0$. From (2.12) with $\alpha = 2k+1$ we get

$$c(t) = \tilde{f}e^{-\sigma t}\left[K_1 - (1-(2k+1))k_1 \times \int e^{(2k+1-1)\int -\sigma \mathrm{d}t}\tilde{f}\mathrm{d}t\right]^{\frac{1}{1-(2k+1)}} \Rightarrow$$
$$c(t) = \tilde{f}e^{-\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t}\tilde{f}\mathrm{d}t\right]^{-\frac{1}{2k}}, \tag{2.18}$$

which means $c(t) > 0$ since $\tilde{f} > 0$. (2.13) must be satisfied,

$$\sigma C e^{-\gamma t} = k_2 \frac{\left(\tilde{f}e^{-\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t}\tilde{f}\mathrm{d}t\right]^{-\frac{1}{2k}}\right)^{2k+1}}{\tilde{f}^{2k}} =$$
$$= -\frac{k_2}{k_1}e^{-\sigma t}\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\tilde{f}e^{-\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t}\tilde{f}\mathrm{d}t\right]^{-\frac{1}{2k}}}{\tilde{f}e^{-\sigma t}}\right] \Rightarrow$$

$$\sigma C e^{-\gamma t} = k_2 \tilde{f} e^{-(2k+1)\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t}\tilde{f}\mathrm{d}t\right]^{-\frac{2k+1}{2k}} =$$
$$= -\frac{k_2}{k_1}e^{-\sigma t}\frac{\mathrm{d}}{\mathrm{d}t}\left(\left[K_1 + 2kk_1 \int e^{-2k\sigma t}\tilde{f}\mathrm{d}t\right]^{-\frac{1}{2k}}\right). \tag{2.19}$$

Continuing with $\tilde{f} = e^{\beta t}, \beta > 0$, inserted into (2.18) gives us

$$c(t) = e^{\beta t}e^{-\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t}e^{\beta t}\mathrm{d}t\right]^{-\frac{1}{2k}} =$$
$$= e^{(\beta-\sigma)t}\left[K_1 + 2kk_1 \int e^{(\beta-2k\sigma)t}\mathrm{d}t\right]^{-\frac{1}{2k}}$$

and inserted into (2.19) it must satisfy

$$\sigma C e^{-\gamma t} = k_2 e^{\beta t} e^{-(2k+1)\sigma t}\left[K_1 + 2kk_1 \int e^{-2k\sigma t} e^{\beta t} \mathrm{d}t\right]^{-\frac{2k+1}{2k}} =$$

$$= k_2 e^{(\beta-(2k+1)\sigma)t}\left[K_1 + 2kk_1 \int e^{(\beta-2k\sigma)t} \mathrm{d}t\right]^{-\frac{2k+1}{2k}}.$$

Setting the integration constant $K_1 = 0$ gets us

$$c(t) = e^{(\beta-\sigma)t}\left[2kk_1 \int e^{(\beta-2k\sigma)t} \mathrm{d}t\right]^{-\frac{1}{2k}} \Rightarrow$$

$$= \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}} e^{\left((\beta-\sigma)-\frac{\beta-2k\sigma}{2k}\right)t} = \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}} e^{\left(\beta-\frac{\beta}{2k}\right)t}$$

and

$$\sigma C e^{-\gamma t} = k_2 e^{(\beta-(2k+1)\sigma)t}\left[2kk_1 \int e^{(\beta-2k\sigma)t} \mathrm{d}t\right]^{-\frac{2k+1}{2k}} =$$

$$= k_2 \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{2k+1}{2k}} e^{\left((\beta-(2k+1)\sigma)+\frac{-(\beta-2k\sigma)(2k+1)}{2k}\right)t} =$$

$$= k_2 \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{2k+1}{2k}} e^{-\frac{\beta}{2k}t},$$

which is satisfied for $k_2 \left(\frac{2kk_1}{\beta-2k\sigma}\right)^{-\frac{2k+1}{2k}} = \sigma C$ and $\frac{\beta}{2k} = \gamma$. This means the solution of (2.17) is given by (2.14) with $d(t) = \tilde{f} = e^{\beta t}$,

$$y_2 = \frac{\left(\frac{2kk_1}{\beta-2k\sigma}\right)^{-\frac{1}{2k}} e^{\left(\beta-\frac{\beta}{2k}\right)t}}{e^{\beta t}} s(t) = \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}} e^{-\frac{\beta}{2k}t} s(t).$$

Combining (2.15) and (2.16) with $\alpha = 2k+1$ we get to solution to $s(t)$ from

$$\left| \frac{e^{\beta t} e^{-\sigma t}}{\left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}} e^{\left(\beta - \frac{\beta}{2k}\right)t}} \right| = K^{-1} e^{k_1 \int \frac{\mathrm{ds}}{s^{2k+1}+s^{2k}+k_1 s+k_2}} \Rightarrow$$

$$\left| \frac{e^{\left(\frac{\beta}{2k} - \sigma\right)t}}{\left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}}} \right| = \frac{1}{K} e^{k_1 \int \frac{\mathrm{ds}}{s^{2k+1}+s^{2k}+k_1 s+k_2}}.$$

With $K = \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}}$ it means

$$\left(\frac{\beta}{2k} - \sigma\right) t = k_1 \int \frac{\mathrm{ds}}{s^{2k+1}+s^{2k}+k_1 s+k_2} = k_1 \int \frac{s'\mathrm{dt}}{s^{2k+1}+s^{2k}+k_1 s+k_2} \Rightarrow$$

$$s' = \frac{1}{k_1}\left(\frac{\beta}{2k} - \sigma\right)\left(s^{2k+1}+s^{2k}+k_1 s+k_2\right). \tag{2.20}$$

We have two cases, $s(t) \geq 0$ and $s(t) < 0$. If $s(t) \geq 0 \Rightarrow k_1, k_2 > 0$, (2.20) is < 0 since $\beta < 2k\sigma \Rightarrow \frac{\beta}{2k} < \sigma$ and $s(t)$ can't be an exponential function $Ce^{\gamma t}$ since positive exponential functions don't have negative derivatives. If $s(t) < 0$ we have for large t

$$s' \approx \frac{1}{k_1}\left(\frac{\beta}{2k} - \sigma\right) s^{2k+1},$$

which means there exists a $T > 0$ where $s' > 0$ and $s(t) > -Ce^{\gamma t}$ after this T. Thus these two cases show that there exists no $\gamma > 0$ and $C > 0$ such that $|s(t)| > Ce^{\gamma t}$ and

$$y_2 = \left(\frac{2kk_1}{\beta - 2k\sigma}\right)^{-\frac{1}{2k}} e^{-\frac{\beta}{2k}t} s(t)$$

is dominated by the $e^{-\frac{\beta}{2k}t}$ term, and y_2 decays exponentially fast to zero as $t \to \infty$.

This means $|y_1| \leq e_1 \leq |y_2|$ where $y_1 \approx e^{-\frac{\beta}{2k}t}$ and $y_2 \approx e^{-\frac{\beta}{2k}t}$ and thus $e_1 \to 0$ when $t \to \infty$ exponentially fast by the squeeze theorem.

3. Simulation

3.1 Implementation

The simulation program is a simple one written in C. The source code can be seen in appendix (A).

It keeps two Lorenz systems, the scaled Cuomo/Oppenheim variant, with one transmitter, (2.1), and one receiver, (2.2), with interference, (2.5), and each system is solved with the euler method,

$$\begin{aligned} u(t_{n+1}) &= u(t_n) + step \cdot \dot{u} \\ v(t_{n+1}) &= v(t_n) + step \cdot \dot{v} \\ w(t_{n+1}) &= w(t_n) + step \cdot \dot{w}, \end{aligned}$$

$$\begin{aligned} u_r(t_{n+1}) &= u_r(t_n) + step \cdot \dot{u_r} \\ v_r(t_{n+1}) &= v_r(t_n) + step \cdot \dot{v_r} \\ w_r(t_{n+1}) &= w_r(t_n) + step \cdot \dot{w}_r, \end{aligned}$$

where t_n is some time and t_{n+1} is the next time after a step of time $step = 0.001$.

Interference $f = e_1^{2k+1}\tilde{f}$ has $\tilde{f} = e^{2t}$, $g = e_2^{2m+1}\tilde{g}$ has $\tilde{g} = u_r^2 + v_r^2 + w_r^2$ and $h = e_3^{2n+1}\tilde{h}$ has $\tilde{h} = u_r^2 + v_r^2 + w_r^2$.

The program runs until the synchronization error (2.3), e_1, e_2 and e_3 is smaller than the set cut off, 10^{-6}, or until the maximum iterations, 1000000, is over. It stops if any of the variables u, v, w, u_r, v_r or w_r goes to infinity. On finish it prints the exponential speed of decrease of the system. The speed is calculated by dividing the exponent of the cut off expressed as an exponential with base e, in the case with $10^{-6} = e^{ln(10^{-6})} = e^{-13.815511}$, that is, -13.815511, with t, the step size times the number of iterations.

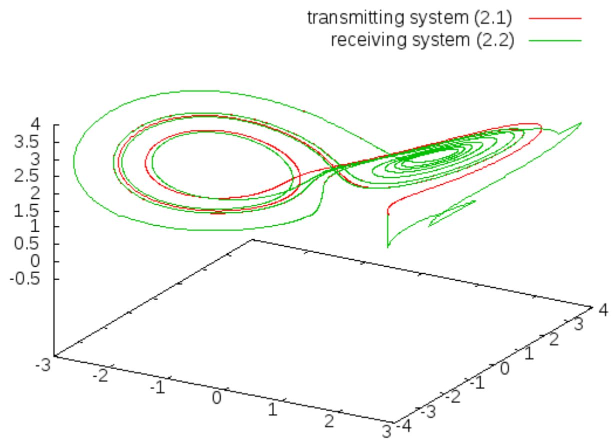

Figure 3.1: Plot of simulation (3.1) odd power run.

3.2 Results

Initial error	Even	Odd
small (3.1)	fails to synchronize	$e^{-2.705740t}$
medium (3.2)	fails to synchronize	$e^{-2.644117t}$
large (3.3)	fails to synchronize	$e^{-2.405207t}$

Table 3.1: Simulations and their synchronization speed.

We have performed several simulations as seen in table 3.1, to show the synchronization really is exponential, which also demonstrates that the synchronization with interference of the right kind works, that is odd powers in f, g and h and a small initial error.

In (3.1) start values for the run with small initial error can be seen. Only u and u_r differ.

$$\begin{aligned} u(0) &= 1.111 & u_r(0) &= 1.1 & e_1(0) &= 0.011 \\ v(0) &= 0.1 & v_r(0) &= 0.1 & e_2(0) &= 0 \\ w(0) &= 1 & w_r(0) &= 1, & e_3(0) &= 0, \end{aligned} \tag{3.1}$$

with interference $f = e_1^{1333878787837}\tilde{f}$, $g = e_2^{811}\tilde{g}$ and $h = e_3^{133333476767}\tilde{h}$ for the

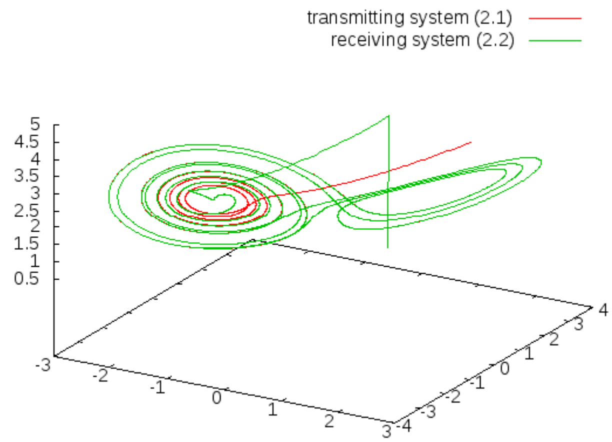

Figure 3.2: Plot of simulation (3.2) odd power run.

odd run and $f = e_1^2\tilde{f}$, $g = e_2^2\tilde{g}$ and $h = e_3^2\tilde{h}$ for the even run. The even run took 106 iterations to fail as e_2 went to infinity. The odd run begins to fail if the power to e_2 is increased to 813. Figure 3.1 shows a plot of the successful synchronization for simulation (3.1).

(3.2) shows the start values of the run with medium initial error,

$$\begin{aligned} u(0) &= 2.111 & u_r(0) &= 1.1 & e_1(0) &= 1.011 \\ v(0) &= 1.1 & v_r(0) &= 0.1 & e_2(0) &= 1 \\ w(0) &= 4 & w_r(0) &= 1, & e_3(0) &= 3, \end{aligned} \tag{3.2}$$

with interference $f = e_1^{67}\tilde{f}$, $g = e_2^{63}\tilde{g}$ and $h = e_3^7\tilde{h}$ for the odd run and $f = e_1^2\tilde{f}$, $g = e_2^2\tilde{g}$ and $h = e_3^2\tilde{h}$ for the even run. The even run took 48 iterations to fail as e_2 went to inifinty. Figure 3.2 shows a plot of the successful synchronization for simulation (3.2).

(3.3) shows the start values of the run with large initial error,

$$\begin{aligned} u(0) &= 5.111 & u_r(0) &= -2.1 & e_1(0) &= 7.211 \\ v(0) &= -2.1 & v_r(0) &= 3.1 & e_2(0) &= -5.2 \\ w(0) &= 0 & w_r(0) &= 3, & e_3(0) &= -3, \end{aligned} \tag{3.3}$$

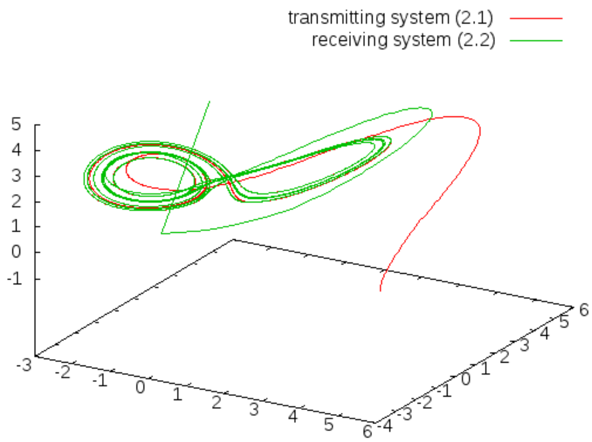

Figure 3.3: Plot of simulation (3.3) odd power run.

with interference $f = e_1^3 \tilde{f}$, $g = e_2^3 \tilde{g}$ and $h = e_3^3 \cdot \tilde{h}$ for the odd run and $f = e_1^2 \tilde{f}$, $g = e_2^2 \tilde{g}$ and $h = e_3^2 \tilde{h}$ for the even run. The even run took 10 iterations to fail as e_2 went to inifinty. If the powers in f, g and h was raised to 5 in the odd run the receiving system diverged to infinity quickly there too. Figure 3.3 shows a plot of the successful synchronization for simulation (3.3).

3.3 Conclusions

Run	Interference
small (3.1) even	$f = e_1^2 e^{2t},\ g = e_2^2(u_r^2 + v_r^2 + w_r^2), h = e_3^2(u_r^2 + v_r^2 + w_r^2)$
small (3.1) odd	$f = e_1^{1333878787837} e^{2t},\ g = e_2^{811}(u_r^2 + v_r^2 + w_r^2), h = e_3^{133333476767}(u_r^2 + v_r^2 + w_r^2)$
medium (3.2) even	$f = e_1^2 e^{2t},\ g = e_2^2(u_r^2 + v_r^2 + w_r^2), h = e_3^2(u_r^2 + v_r^2 + w_r^2)$
medium (3.2) odd	$f = e_1^{67} e^{2t},\ g = e_2^{63}(u_r^2 + v_r^2 + w_r^2), h = e_3^7(u_r^2 + v_r^2 + w_r^2)$
large (3.3) even	$f = e_1^2 e^{2t},\ g = e_2^2(u_r^2 + v_r^2 + w_r^2), h = e_3^2(u_r^2 + v_r^2 + w_r^2)$
large (3.3) odd	$f = e_1^3 e^{2t},\ g = e_2^3(u_r^2 + v_r^2 + w_r^2), h = e_3^3(u_r^2 + v_r^2 + w_r^2)$

Table 3.2: Chosen interference for the simulations.

Speed of synchronization with interference depends on a power of the error. The interference used in the simulations is shown in table 3.2. The synchronization fails if the initial error is too large. In the analysis we assumed the interference was much smaller than the value of the variable, however this is not always the case in practice. We showed that the synchronization is of exponential speed if the interference function is raised to an odd power. We showed that this holds for any odd power but simulations demonstrated that the larger the initial error is the smaller the odd power has to be for it to work. In the even runs the synchronization failed even for low powers.

References

[1] Steven H. Strogatz, Nonlinear dynamics and chaos, 2nd edition, 2015.

[2] Edward N. Lorenz, "Deterministic nonperiodic flow", J. Atmospheric Sci., vol 20, pp. 130-141, Mar. 1963.

[3] Warwick Tucker, "The Lorenz attractor exists", 1998.

[4] R. L. Devaney, An introduction to dynamical chaotic systems, 1989.

[5] Louis M. Pecora and Thomas L. Carroll, "Synchronization in chaotic systems", Phys. Rev. Lett., vol. 64, pp. 821-824, Feb. 1990.

[6] Thomas L. Carroll and Louis M. Pecora, "Synchronizing chaotic circuits", IEEE Trans. Circuits Systs., vol. 38, pp. 453-456, Apr. 1991.

[7] Louis M. Pecora and Thomas L. Carroll, "Driving systems with chaotic signals", Phys. Rev. A, vol. 44, pp. 2374-2383, Aug. 1991.

[8] Kevin M. Cuomo and Alan V. Oppenheim, "Circuit Implementation of Synchronized Chaos with Applications to Communications", Phys. Rev. Lett., vol. 71, pp. 65-68, Jul. 1993.

[9] Kevin M. Cuomo, Alan V. Oppenheim and Steven H. Strogatz, "Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications", IEEE Trans. Circuits Systs., vol. 40, pp. 626-632, Oct. 1993.

[10] Tiberiu Harko, Francisco S. N. Lobo and M. K. Mak, "A Chiellini Type Integrability Condition for the Generalized First Kind Abel Differential Equation", Universal Journal of Applied Mathematics, 1, pp. 101-104, 2013.

A. lorenz.c

```
/*
 * Run simulations on two synchronized Lorenz systems.
 * Compile with cc  lm lorenz.c. Debug prints with cc  lm  DDEBUG lorenz.c
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* start values */
double u = 5.111;
double v =  2.1;
double w = 0;
double ur =  2.1;
double vr = 3.1;
double wr = 3;
double s = 16; /* sigma */
double r = 45.6;
double b = 4;
double e_1, e_2, e_3;
double f, g, h; /* set to powers of e1 3 multiplied with ft, gt and ht */
double ft, gt, ht; /* set in error_and_interference() */
double k = 2; /* interference, f = pow(e_1, k+1) * ft; */
double m = 2; /* interference, g = pow(e_2, m+1) * gt; */
double n = 2; /* interference, h = pow(e_3, n+1) * ht; */
double zero = 1e 6; /* considered zero */
double tu, tv, tw, tur, tvr, twr; /* temporary u, v, w, ur, vr, wr*/
double step = 0.001; /* time step */
double total_time = 0;
int max_iterations = 1000000;
FILE *file;

void error_and_interference();

/*
 * Use the Euler method to get two solutions to two Lorenz systems.
 * The second system is supposed to synchronize with the first. Stop
 * when error (difference between the two systems) is zero, some value
 * went to infinity or when maximum iterations are reached.
 */
int main(int argc, char *argv[]) {
        double zero_exp = 1;
        int flag = 1; /* not touched if for loop runs all iterations */
        int i;
```

```
error_and_interference();

file = fopen("data", "w");
if (file == NULL) {
        printf("Could not open write  to file\n");
        exit(1);
}
fprintf(file, "%f %f %f %f %f %f\n", u, v, w, ur, vr, wr);

for (i = 0; i < max_iterations; i++) {
        total_time += step;
        tu = u + step * (s * (v    u));
        tv = v + step * (r * u    v    20 * u * w);
        tw = w + step * (5 * u * v    b * w);

        tur = ur + step * (s * (vr   ur) + f);
        tvr = vr + step * (r * u    vr    20 * u * wr + g);
        twr = twr + step * (5 * u * vr    b * wr + h);

        u = tu;
        v = tv;
        w = tw;

        ur = tur;
        vr = tvr;
        wr = twr;

        error_and_interference();

        #ifdef DEBUG
        fprintf(file, "%f %f %f %f %f %f %f %f %f\n", u, v, w, ur, vr, wr,
                e_1, e_2, e_3);
        #else
        fprintf(file, "%f %f %f %f %f %f\n", u, v, w, ur, vr, wr);
        #endif /* DEBUG */

        /* finished if error is small enough */
        if (fabs(e_1) < zero && fabs(e_2) < zero
                && fabs(e_3) < zero) {
                flag = 0;
                break;
        }

        /* failure: check if u,v,w,ur,vr or wr went to + infinity */
        if (isinf(u) != 0 || isinf(v) != 0 || isinf(w) != 0 || isinf(ur) != 0
                || isinf(vr) != 0 || isinf(wr) != 0) {
                flag = 2;
                break;
        }
```

```
        }

        /* set the exponential considered zero at choosen cut off */
        zero_exp = log(zero);

        fclose(file);

        #ifdef DEBUG
        printf("%s%f\n", "Error tolerance considered zero: ", zero);
        printf("%s%f%s\n", "Error tolerance considered zero: exp(", zero_exp, ")");
        #endif /* DEBUG */

        if (flag == 1) {
                printf("%s %d %s\n", "Zero error not reached in", i, "iterations.");
        } else if (flag == 0) {
                printf("%s%f%s\n", "Zero error reached in: exp(",
                zero_exp/((double)i*step), "t)");
                #ifdef DEBUG
                printf("%d %s\n", i, "iterations");
                printf("%f %s\n", (double)i*step, "time");
                #endif /* DEBUG */
        } else {
                printf("%s %d %s\n", "Zero error not reached in", i,
                        "iterations (error to inf).");
        }

        #ifdef DEBUG
        printf("%s%f %s%f %s%f\n", "e_1: ", e_1, "e_2: ", e_2, "e_3: ", e_3);
        printf("%s%f %s%f %s%f\n", "u: ", u, "v: ", v, "w: ", w);
        printf("%s%f %s%f %s%f\n", "ur: ", ur, "vr: ", vr, "wr: ", wr);
        #endif /* DEBUG */

        return(0);
}

void error_and_interference() {
        e_1 = u    ur;
        e_2 = v    vr;
        e_3 = w    wr;

        ft = exp(2*total_time);
        gt = pow(ur, 2) + pow(vr, 2) + pow(wr, 2);
        ht = pow(ur, 2) + pow(vr, 2) + pow(wr, 2);

        f = pow(e_1, k+1) * ft;
        g = pow(e_2, m+1) * gt;
        h = pow(e_3, n+1) * ht;
}
```