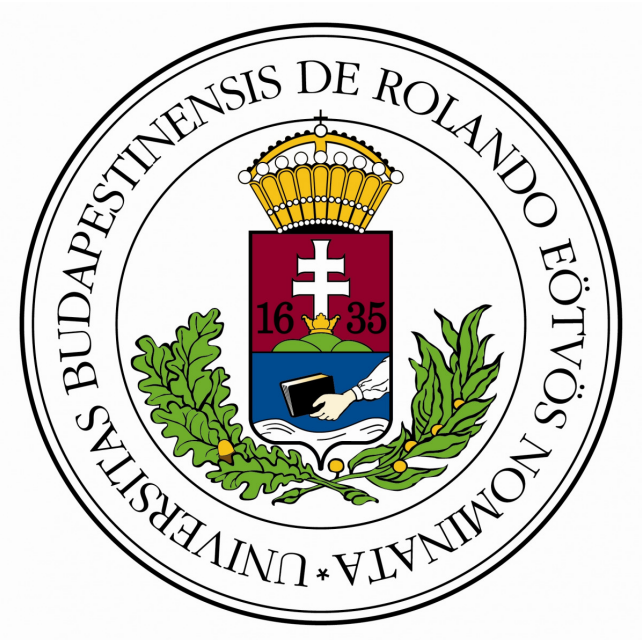
Lorenz pillangó kirajzolása, és a Lyapunov exponens

Hegyi Gáspár András

2021 Április



1. Bevezetés

A feladat a Lorenz pillangó kirajzolása, volt, amely kaotikus mozgást ír le. Az ábrát a Lyapunov exponens alapján kiszíneztem. Ez az exponens egy dinamikai rendszert jellemez, mégpedig azzal, hogy két infinitezimálisan közeli pontok mennyire távolodnak el egymástól.

2. Elméleti összefoglaló

2.1. Kaotikus rendszerek, Lorenz pillangó

A Lorenz-rendszert 1963-ban Edward Lorenz fejlesztette ki, és egy modellt írt le a légkörben való hőterjedésre. Maga a rendszer egy három ismeretlenes nem lineáris differenciálegyenletrendszer(1).

$$\frac{dx}{dt} = \sigma(y - x) \tag{1}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{2}$$

$$\frac{dz}{dt} = xy - \beta z \tag{3}$$

Ebben x,y,z a térkordináták, σ, ρ, β pedig paraméterek. Ha az utóbbiakat helyesen választjuk meg, akkor a rendszer kaotikus viselkedést fog mutatni. Lorenz a parméterekre $\rho = 28, \sigma = 10, \beta = 8/3$ értékeket használt, amelyre kaotikus megoldás jön ki. Ha a $\rho < 1$, akkor a mozgásban egy egyensúlyi pont jelenik meg az origóban. $\rho = 1$ esetben van egy pitchfork bifurkáció, vagyis ρ további növelésével még két egyensúlyi hely fog megjelenni $(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1)$, és $(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1)$. Kis ρ értékekre ezekbe az attraktorokba esik be a rendszer, a mozgása stabil. Amikor $\rho > 24.74$, akkor a két egyensúlyi pont repellor lesz, és azok körül fog folyatatni egy kaotikus mozgást. Ezek szemléltetve vannak a (4) fejezetben.

2.2. Lyapunov exponens

A Lyapunov exponens, mint már a bevezetésben is volt említve, azt írja le, hogy egy dinamikai rendszerben két infinitezimálisan közeli pont mennyire távolodik el egymástól időben(4).

$$|\delta R(t)| \approx e^{\lambda t} |\delta R_0| \tag{4}$$

Itt $\delta R(t)$ a pillanatbeli távolság a két pont között, δR_0 a kezdeti eltérés, és λ a Lyapunov exponens. Ez az exponens természetesen minden időpillanatban más és más lesz, viszont ha az idő szerinti spektrumát ábrázoljuk, akkor az értékeiből lehet következtetni a rendszer tulajdonságaira. Ha benne csak negatív értékek szereplnek, akkor a mozgás beesik egy fix pontba. Ha negatív, és legalább egy nulla szerepel benne, akkor egy periodikus mozgásról van szó, és ha pozitív értékek is, akkor kaotikus a mozgás.

2.3. Numerikus megvalósítás

A differenciálegyenletrendszer megoldását Euler módszerrel végeztem, vagyis az egyenletek alapján a kezdőfeltételekből indulva lépettem a rendszert akkora lépésközzel, hogy pontos megoldás jöjjön ki.

A Lyapunov exponenst a (4) egyenlet átrendezéséből számoltam ki.

3. A program

Ebben a fejezetben a megírt kódot részletezem.

A teljes kód két nagyobb részből áll, egy c++ Lorenz osztályból, melynek létrehozáskor megadhatjuk a β, σ, ρ paramétereket, a teljes időintervallumot amin megoldjuk a mozgásegyenleteket, a lépés nagyságát (ezt érdemes <0.01-re venni, hogy ne szállson el a végtelenbe egy idő után),és az utolsó három paramétere a kezdőfeltételek. Az osztálynak két metódusa van, az egyik a *data_export*, aminek meg kell adni a kimeneteli fájl nevét, és a megoldást kiírja belé három oszlopban, x,y,z sorrendben. A másik a *Lyapunov*, amelynek egy másik Lorenz objektum a bemenete. Érdemes úgy megcsinálni őket, hogy a kezdőfeltételeik közel legyenek. Ez kiszámolja a Lyapunov exponenst a mozgás során, és eltárolja egy vector változóba őket. Példa a használatára:

```
#include <iostream>
#include <math.h>
#include <vector>
#include "Butterfly.h"

using namespace std;

int main()
{

    Lorenz pillango1(10,28,8/3,100,0.005,0,1,0);
    Lorenz pillango2(10,28,8/3,100,0.005,0.0001,0.9999,0.0001);

    pillango1.data_export("output1.txt");


    vector<double> res=pillango1.Lyapunov(pillango2);

    return 0;
}
```

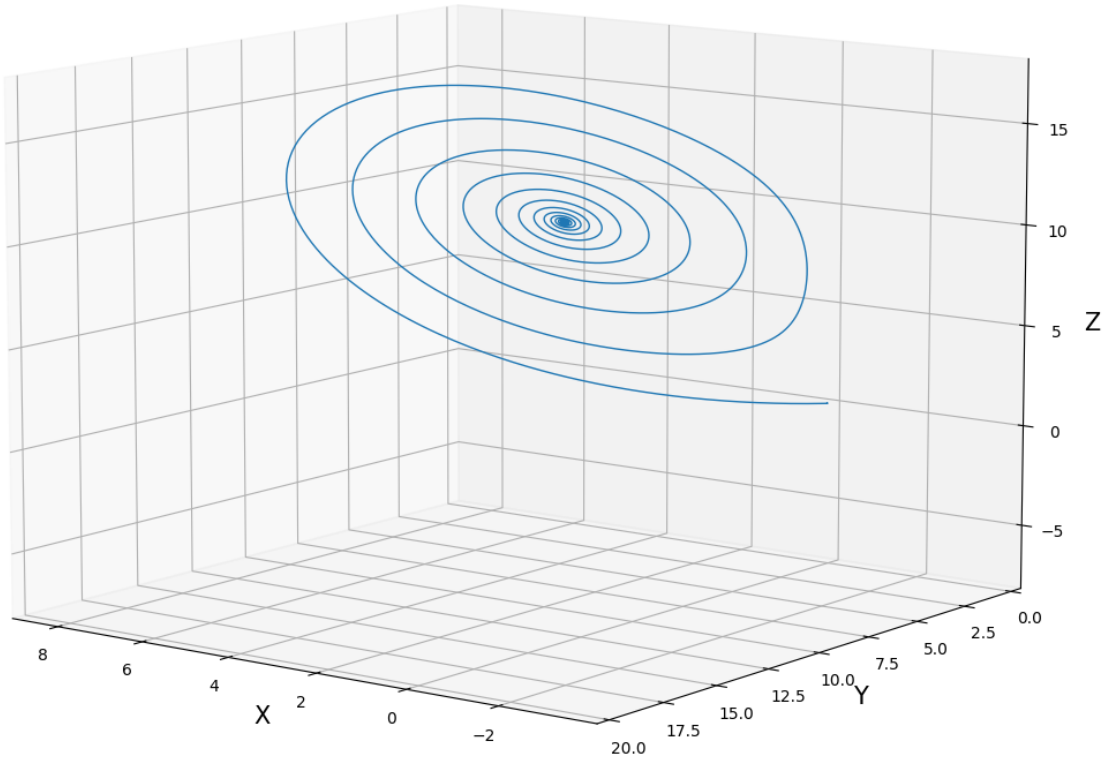
A másik rész egy python Visualize osztály, amellyel a c++ által kiszámolt mozgás időbeli fejlődését egyszerűen lehet szimulálni. Ennek létrehozásakor elég az adatokat megadni olyan formátumban mint a c++ kimenete, egy dim változót, hogy 3 vagy 2 dimenzióban szeretnénk megjeleníteni. Ha két dimenzióban jelenítjük meg, megadhatjuk azt is, hogy melyik síkmetszetet rajzolja ki, ezt a plane változóval tehetjük pl.: plane='xz'. Ezután már csak meg kell hívnunk a draw metódust és megkapjuk a képet egy ablakban. Példa a használatára:

```
import csv
import matplotlib.pyplot as plt
from vis_class_test import Visualize
import numpy as np


def get_data(file_name):
    with open(file_name) as f:
        reader=csv.reader(f)
        data=np.array(list(reader))

    data=data.astype(float)
    return data

data1=get_data('output1.txt')
data2=get_data('output2.txt')

data=[data1,data2]

sim=Visualize(data,dim=3)
```

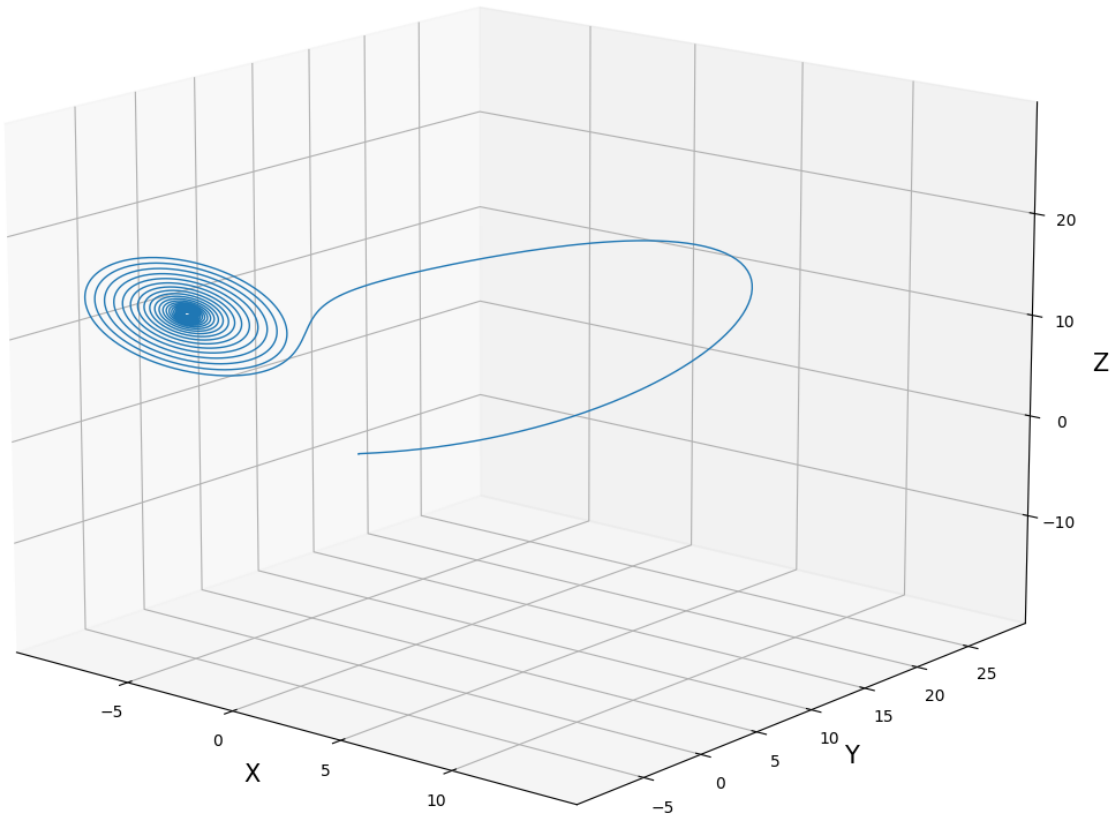
```
sim.axs.set_xlabel('X',size=15)
sim.axs.set_ylabel('Y',size=15)
sim.axs.set_zlabel('Z',size=15)


sim.draw()
```

4. Kiértékelés

A Lorenz-pillangót kirajzoltam több ρ értékre is, látszik, hogy kisebb értékekre beesik valamelyik egyensúlyi pontba(1)(2).

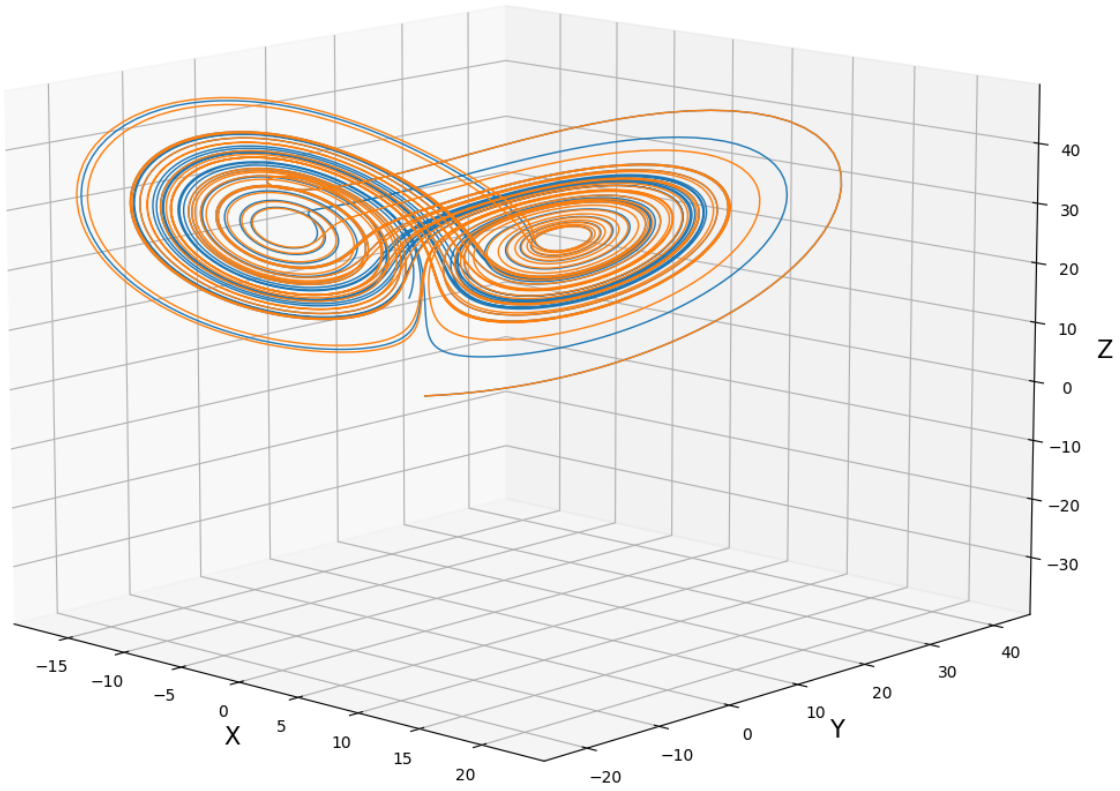


1. ábra. A rendszer időbeli fejlődése $\rho = 10, \beta = 8/3, \sigma = 10$ esetben.



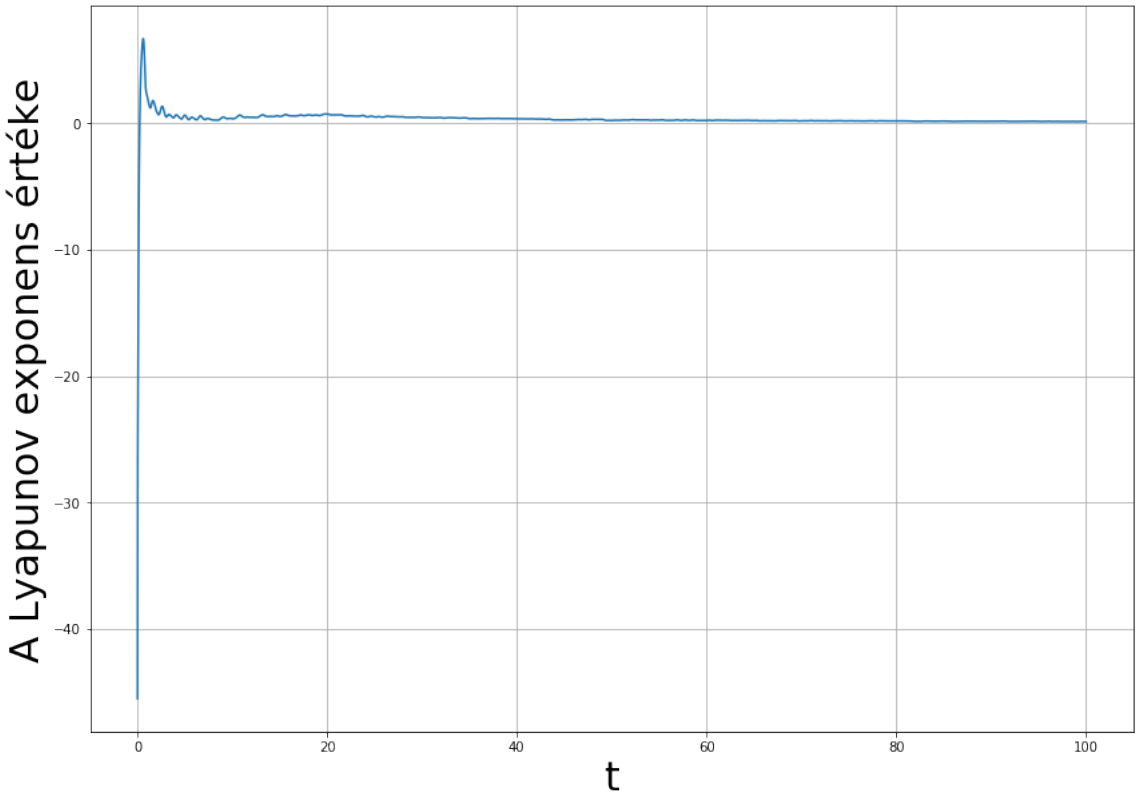
2. ábra. A rendszer időbeli fejlődése $\rho = 14, \beta = 8/3, \sigma = 10$ esetben.

Ezen kívül ábrázoltam két infinitezimálisan közeli pontok mozgását. Ezek az elején nagyon együtt voltak, viszont egy idő után más pályákon mentek(3).



3. ábra. Két infinitezimálisan közel induló pont időbeli fejlődése. Ami kék vég látszódik a grafikonon nem a kezdőpontja, hanem a jelenlegi időpillanatban ott tartott a mozgása a kékkel jelölt pontnak. Mindkettő a látszódó narancssárga végből indult ki (vagyis ott nagyon közel).

Ezeknek a pontoknak ábrázoltam a Lyapunov exponensét is az idő szerint(4).



4. ábra. A Lyapunov exponens az idő függvényében ábrázolva.