

# Chaotic Dynamics - CSCI 5446

## Problem Set 9

Santhanakrishnan Ramani

April 6, 2016

### Problem 1

Generated the data sets for the Lorenz System with  $a=16$ ,  $r=45$ ,  $b=4$ ,  $h=0.001$ ,  $n=15000$  from an initial condition  $(-13,-12,52)$  for a chaotic trajectory and changed the value of  $r=20$  to generate a non-chaotic trajectory keeping all other values same.

### Problem 2

Running the wolf algorithm on the data set generated in prob1, found the value of  $\lambda_1$  for the chaotic trajectory to be equal to 1.07 and for the non-chaotic trajectory it was -0.3, and the values are consistent with what we know about lyapunov exponents, as  $\lambda < 0$  if it has an attractor as they converge to the fixed point and  $\lambda > 0$  if the system is chaotic.

### Problem 3

By running mutual on the x coordinate data from prob1a, I found out  $\tau = 108$  or 0.108 secs, then ran false nearest neighbor on the same data and found out the embedding dimension  $m = 3$ , and according to taken's theorem  $m > 2d$ , so  $m = 7$  since  $d = 3$  for Lorenz system. Then ran the following `lyap_k` command on the data set.

*lyap\_k data.txt - m3 - M7 - d108 - o out.dat*

The lyapunov exponent obtained using this was equal to 1.18, this is slightly greater than the value obtained for the chaotic trajectory using wolf's algorithm in the previous problem. This could be possibly because of the delay coordinate embedding, as it doesn't preserve the geometric shape of the dynamical system.

### Problem 4

- (a) The  $\lambda_i$ s for the Lorenz system, using the same initial conditions and parameter values that I used in prob1a by using a 10000-point integration run is  $[1.3481, -2.4463, 0.0709]$ . The largest positive lyapunov exponent 1.3481 is closest to the one obtained in Prob3, and yes they should match with them. I trust the value obtained using Kantz `lyap_k` algorithm more as it takes in consideration all the factors and we use approximation while calculating Jacobian in variational equation which can have some effect on the lyapunov exponent values obtained.
- (b) I ran the algorithm for 3000-point integration run and obtained  $\lambda_i = [1.3431, -11.1498, 0.0117]$  and a 50000-point integration run and obtained  $\lambda_i = [1.4598, 0.7675, 0.6923]$ . We could see that there is some effect on the lyapunov exponents (1.3431 for 3000-point run, 1.4598 for 50000-point run) depending on how long you run the algorithm. The possible reason for this could be because of the numerical approximation being made at every step of the calculation and the matrices become ill-conditioned in longer run, and yes it is expected to change.