University of Colorado
Department of Computer Science

Chaotic Dynamics – CSCI 4446/5446
Spring 2016

Problem Set 9

Issued:	10 March 2016
Due:	17 March 2016

Reading: *Liz's notes on Wolf's algorithm and the materials listed in the "Assigned reading for PS8-10" handout on the course webpage.*

Bibliography:

- H. Abarbanel *et al.*, "Lyapunov exponents in chaotic systems: their importance and their evaluation using observed data," *International Journal of Modern Physics B*, **5**:1347-75 (1991). There's a related paper by Abarbanel on page 229 of the Casdagli reprint collection, which is on library reserve.
- H. Abarbanel *et al.*, "Variation of Lyapunov Exponents on a Strange Attractor," *J. Nonl. Sci.* **1**:175 (1991). *local* λs.
- G. Gunaratne *et al.*, "Chaos beyond Onset: A Comparison of Theory and Experiment," *Phys. Rev. Lett.* **63**:1-4 (1989). Gives a good algorithm for finding unstable periodic orbits (UPOs).
- Predrag Cvitanovic's publications page[1] has lots of good papers on UPOs, including "Invariant Measurement of Strange Sets in Terms of Cycles," *Phys. Rev. Lett.* **61**:2729-2732 (1988), the one that defines attractors as made up of UPOs. Several other good theoretical and experimental papers on UPOs appear in chapter 9 of *Coping with Chaos* which is on library reserve.
- J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Rev. Mod. Phys* **57**:651 (1985) and J.-P. Eckmann *et al.*, "Liapunov exponents from time series," *Phys. Rev. A* **34**:4971 (1986). The first one proposed the "reverse engineer the Jacobian" approach to λ calculation; the second applied it to real data.
- U. Parlitz, "Identification of True and Spurious Lyapunov Exponents from Time Series," *Int. J. Bif. Chaos* **2**:155 (1992). Reprinted in *Coping with Chaos*, which is on library reserve.
- P. So *et al.*, "Detecting unstable periodic orbits in chaotic experimental data," *Physical Review Letters* **76**:4705-4708 (1996).

[1] `http://www.cns.gatech.edu/ predrag/papers/preprints.html`

- A. Wolf, "Quantifying chaos with Lyapunov exponents," in *Chaos*, Princeton University Press, 1986. A reasonable review article.

Problems:

1. (a) Generate a trajectory from the Lorenz system with $a = 16$, $r = 45$, $b = 4$. Use **nonadaptive** RK4 with a timestep of 0.001 from some starting point near the attractor, and generate at least 15000 points.

(b) Generate a non-chaotic trajectory by lowering the r value to 20 or less and repeating the run.

The next two problems will use these trajectories as "experimental" chaotic and non-chaotic data sets. (There is no need to turn anything in for problem 1.)

2. Implement the Wolf algorithm for determining λ_1 of an experimental data set and test it on the trajectories from problem 1. The notes listed in the assigned reading include a schematic that should help you think about the algorithm. Note that you do not have to worry about embedding here, as the data are already in full state-space form. To run the Wolf algorithm on a data set that only contained samples of one state variable, you'd first have to embed that data, using your PS8 results and making intelligent choices for m and τ. (CSCI 5446 students will do that in the next problem.)

Are your λ_1 results on these two data sets consistent with what you know about Lyapunov exponents and chaos? Why or why not?

3. *[optional for those enrolled in CSCI 4446]* Save out just the x coordinate of the data set from problem 1(a) to a file (say, `foo`) and use TISEAN's `lyap_k` command to compute λ_1 from that scalar time series. On the CSEL machines, the command will look like this:

```
lyap_k foo -m?  -M??  -d???  -o outputfile
```

...where you want the calculation done for a range of values from ? to ?? of the dimension m and ??? is the delay τ. That is, if I wanted `lyap_k` to run with $\tau = 20$ and $1 \leq m \leq 4$, I would type `lyap_k foo -m1 -M4 -d20 -o outputfile`. (If you are using the Mac version of TISEAN, the command looks like this instead: `lyap_k foo -M1,4 -d20 -o outputfile`.)

Use `mutual` on each data set, as you did in PS8, to pick τ; use the Takens theorem to pick m. Please explain your choices for these two parameters, and also compare your answer for λ_1 to the value that you obtained for this trajectory in the previous problem, using Wolf's algorithm. Discuss any discrepancies.

[totally optional for everyone, but useful for those who are interesting in actually using these techniques] Play with at least one of the other parameters of the `lyap_k` algorithm: the Theiler window `-t`, the epsilon size `-r` (and, if you wish, the number of iterations `-s`). Describe **and explain** their effects upon the results. These explanations are going to require some digging around on the TISEAN website and corroboration against the class readings and your lecture notes. Please comment on which of the results you trust more and explain why. What were the "lessons learned" from your experience with this

part of this problem?

4. Given the system derivative, the variational equation (PS7), and a linear algebra package that computes eigenvalues[2], calculating Lyapunov exponents is very easy:

$$\lambda_i := lim_{t \to \infty} \frac{1}{t} \ln |m_i(t)|$$

where $m_i(t)$ are the eigenvalues of the matrix $\{\delta_{ij}\}$ that you integrated in PS7.

(a) Calculate the λ_is for the Lorenz system, using the same initial conditions and parameter values that you used in problem 1(a) above. Approximate the "$t \to \infty$" step in the equation above by using a 10000-point integration run. Comment on your results: whether or not they match the values from Problem 3, whether or not they *should* match those values, which ones you should trust more and why, etc.

(b) *[optional for those enrolled in CSCI 4446]* Repeat the calculation for longer and shorter variational equation integration runs and comment on the results. (Hint #1: do the λs change? *Should* they, theoretically? Hint #2: watch out for ill-conditioned matrices here...)

[2] `Eigenvalues` and `eigenvals` in Mathematica and Maple, respectively.