Calculating Lyapunov Exponents and Modes for Hard Disk Systems

Joe Dinius
University of Arizona
Program in Applied Mathematics

May 17, 2009

Abstract

This paper discusses the procedure for calculating Lyapunov exponents and modes for a system of hard, colliding disks. The theoretical method is developed and implemented numerically. The numerical method's functionality is verified against simple dynamical systems and then extended to the system of hard disks. Current work performed is presented along with future work required to publish results.

Introduction

Lyapunov exponents describe the rate of volumetric growth of perturbed trajectories in phase space from their non-perturbed counterparts. For continuous orbits, these exponents can be thought of as the limit as time becomes very large of the eigenvalues of the dynamical system's Jacobian. The multiplicative ergodic theorem by Oseledec states that for ergodic systems, for which the ergodic domain is the full L-dimensional phase space, there are L orthonormal initial vectors which yield a set of L Lyapunov exponents. In addition, these Lyapunov exponents are *independent* of the initial conditions of the reference trajectory [1], [5]. A precise definition of the Lyapunov exponents is given in Definition 1.

Definition 1. The maximal *Lyapunov exponent* of a continuous dynamical system is

$$\lambda_{\max} = \lim_{t\to\infty} \frac{1}{t} \log \frac{|\delta T(t)|}{|\delta T(0)|}. \tag{1}$$

where T is the system's Jacobian, and $\delta T(0)$ and $\delta T(t)$ represent the evolution of an initial perturbation at times 0 and t, respectively. Each additional Lyapunov exponent can be calculated by taking a space tangent to the direction of maximal growth; *i.e.* a subspace of the original phase space where the direction of $e^{\lambda_{\max} t}$ is excluded and applying (1) to the reduced system.

Using this definition to calculate the exponents is cumbersome, and can lead to numerical instabilities.

Much work has already been done pertaining to the calculation of the spectra of Lyapunov exponents for dynamical systems. The most frequently sourced paper is by Benettin et. al [2]. This paper describes an algorithm utilizing orthonormalized vectors and their growth in phase space. For Hamiltonian systems, this algorithm presupposes knowledge of the systems Jacobian at each point in phase space. What happens though if you only have time series data and do not know the system's Jacobian a priori?

The literature in the field is less than descriptive as to how the spectra are calculated in practice. The purpose of this paper is to describe my work performed trying to untangle the process for calculating Lyapunov spectra and modes for hard disk systems. Future work utilizing developments from this project is described.

Theory

The basic idea behind calculating the Lyapunov spectra for any dynamical system involves the following recursive series of steps [2]:

- Begin identification loop ($p = 1$)
- Find the direction of maximal growth, given that this direction must be orthogonal to all directions identified previously.
- Pick an initial point in phase space, w_0, such that $||w_0^p|| = 1$
- Begin calculation loop ($k = 1$)
- Calculate the norm of the evolution vector $\alpha_k^p = ||J_{(k-1)s} w_{k-1}^p||$, where $J_{(k-1)s}$ is the system Jacobian at the $(k-1)$st step and s is the simulation time step
- Normalize the evolution vector and set equal to w_k^p and increment k
- End calculation loop
- Calculate Lyapunov exponent $\chi_p = \lim_{k\to\infty} \frac{1}{ks} \sum_{i=1}^{k} \ln \alpha_i^p$
- Iterate p
- End identification loop

In the process described above, $\lim_{k\to\infty} w_k^p$ is the Lyapunov mode associated with the exponent χ_p.

There needs to be a discussion of Lyapunov mode structure here so that the meaning of these modes is clear. The direction of maximal growth will change greatly from one time step to the next. Therefore, one can reasonably conclude that these modes are highly localized;

that is, they do not affect the large-scale ordering of the system. Consider now the case(s) where $\chi = 0$. Such exponents come from conservation laws for the system being investigated. These modes converge over time since there is not exponential growth in the w_k^p's. These modes show how the system orders itself over time in terms of large scale structure.

It is important here to discuss why the above "algorithm" will give the correct behavior. From a numerics standpoint, using the Gram-Schmidt orthonormalization process guarantees we are always manipulating objects with norm 1. Therefore, we are assured numerical stability; since vector norms remain constant throughout each iteration. The algorithm converges, but does it converge to the right value? Convergence and accuracy of the particular algorithm presented is developed in great detail in [2]. Considerations for how to choose an appropriate time step to use in the algorithm are also presented in [2]. The argument hinges on the fact that there exists a uniform bound on the α_k^p's for all k above; any time step that is not too large ensures proper behavior of the algorithm. Therefore, any reasonable choice of time step (say, anywhere from 1 to 5 units) will be adequate to evaluate the Lyapunov exponents numerically.

Work Completed

Note. All numerical results presented in this paper use the algorithm developed by Wolf et. al [4]. This algorithm uses the same process as prescribed in [2]. A numerical comparison of the two algorithms is presented in this section which shows that the two algorithms give the same results. The reason for comparison to [2] is that it is the most highly sourced paper pertaining to the calculation of the Lyapunov spectra for hard disk systems.

It is important to begin by looking at the calculation of Lyapunov exponents for a simple dynamical system. The natural choice is the Lorenz attractor, described by the ODEs

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned}$$

For this paper, the values $\sigma = 10, \rho = 28, \beta = \frac{8}{3}$ are used.

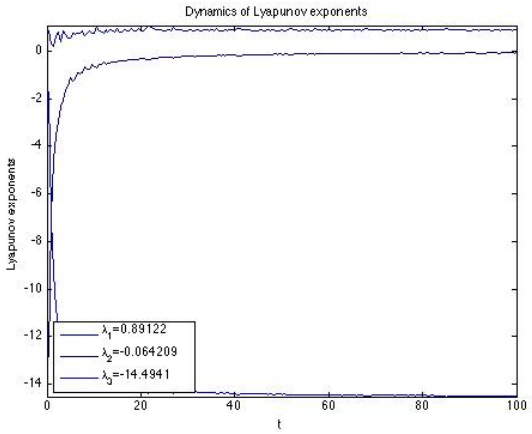

Figure 1: Lyapunov Exponents of the Lorenz System

A rough analytic calculation shows $\lambda_1 \approx .9, \lambda_2 = 0, \lambda_3 \approx -14.3$ [1]. The values in Figure 1 at $t = 100$ show close agreement with those calculated by hand. The physical interpretation of $\lambda = 0$ is very straightforward. A λ of 0 means that there is a direction in phase space which is unchanging in time, a *conserved quantity*. In the case of the Lorenz system, there are no source terms; therefore the energy of the system is conserved.

There is a very interesting side note to present here concerning the connection of Lyapunov exponents to chaos in continuous dynamical systems. A continuous dynamical system is *chaotic* if at least one Lyapunov exponent is positive. An interesting question regarding chaos in the Lorenz system was presented by and answered in [7]: How do global properties (*i.e.* chaotic or not) of the Lorenz system change with the parameter ρ in the equations for the orbit?

[1] For a description of how to calculate these exponents by hand, see [6] Pg. 665-669

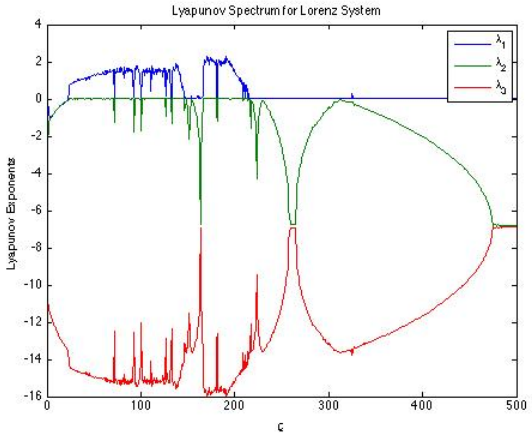

Figure 2: Lyapunov Exponents of the Lorenz System with $\rho \in [.5, 500]$

Figure 2 shows a reproduction of a plot from [7] using Wolf's algorithm. The reason for the dips is explained in [7]; the Matlab integrator RK45 jumps through fine details of the Lorenz orbit. The problem can be fixed by setting a stricter tolerance or smaller stepsize used in the algorithm; however, results are accurate enough for the current discussion.

It is interesting to note that, for varying values of ρ, the system goes from having one positive Lyapunov exponent, $0 < \rho \leq 150$ and $170 < \rho \leq 210$; meaning that the Lorenz system is chaotic in these regions. For $\rho > 210$, the Lyapunov spectra structure as calculated is very smooth and does not have a positive exponent. An additional plot showing chaos in the Lorenz system is presented [7].

To get to the end goal of calculating a Lyapunov spectrum for a system of rigid disks (and hopefully beyond), the approach of using Wolf's algorithm must be verified. After that, application to the system of colliding disks will be less of a leap logically.

In [2], an example of a Hamiltonian system is presented for which the Lyapunov spectrum is calculated using their algorithm. The Hamiltonian of the system is

$$H(q,p) = \sum_{j=1}^{3} \frac{\omega_i}{2}(q_i^2 + p_i^2) + q_1^2 q_2 + q_1^2 q_3.$$

This system resembles the standard harmonic oscillator with additional (nonlinear) terms. Benettin et. al [2] proposes different regimes for this system. As a verification effort, results using Wolf's algorithm are compared to numerical results from [2].

First, calculate the state transition matrix for use in the integration routine. To do this calculation, use Hamilton's equations of motion

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}$$

to get the state transition matrix

$$\Phi = \begin{pmatrix} 0 & 0 & 0 & \omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_3 \\ -\omega_1 & -2q_1 & -2q_1 & 0 & 0 & 0 \\ -q_1 & -\omega_2 & 0 & 0 & 0 & 0 \\ -q_1 & 0 & -\omega_3 & 0 & 0 & 0 \end{pmatrix}.$$

Propagate states using

$$\dot{u} = \Phi u$$

and integrate.

To integrate the equations, the RK45 built in Matlab integrator is used. The results for the Lyapunov spectrum are shown below. Figure 3 shows results given the same setup for

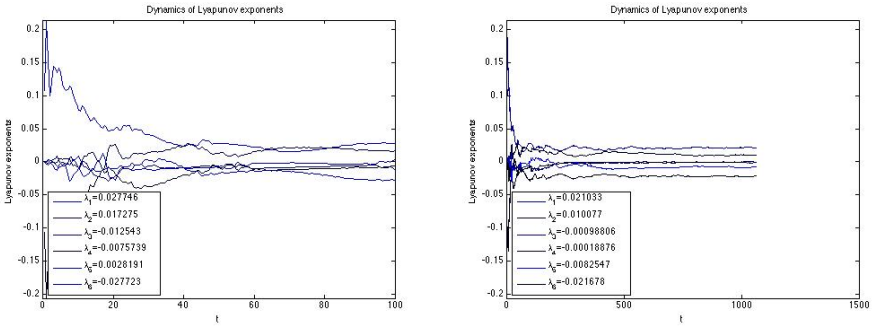

Figure 3: Wolf's Algorithm Results for Benettin Hamiltonian system

the test case $E = (.01, .01, .07)$.

Figure 4 displays the corresponding log-log plot of the data for test case $E = (.01, .01, .07)$ The data for $p = 1, 2$ looks very good. For smaller exponent, the method seems to dance

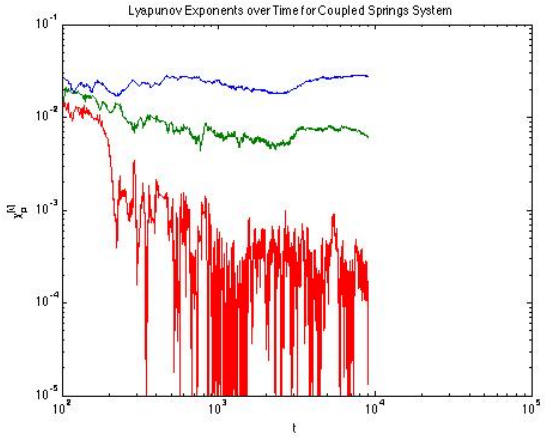

Figure 4: Benettin Hamiltonian System with $E = (.01, .01, .07)$

around the 0-line resulting in a large fluctuation. However, in the sense of average, the magnitude looks very close to that of Benettin et al. From physical considerations, (the system has a conserved quantity and is time-reversible), we know that there will be two Lyapunov exponents converging to 0. Therefore, we expect given numerical sensitivity, there will be fluctuation about the $y = 0$ line in the calculation of the third Lyapunov exponent [2]. Now, to apply these results to the desired problem, a model must be coded to represent the physics of colliding hard disks.

Creating the model seemed incredibly simple at the outset, however the devil is in the details. Incomplete documentation exists describing the steps to follow for the representation of a multiple hard disk system with periodic boundary conditions. In order to facilitate the calculation of the Lyapunov exponents of a hard disk system, the following steps were necessary to setup the physics of the problem [1].

1. Identify equations of motion: these are described to some detail in the paper by Posch in the book by Springer. The basic EOM for free flight (no collisions) for n disks are

$$\begin{aligned} \mathbf{q}_{i+1}^{(n)} &= \mathbf{q}_i^{(n)} + \frac{1}{m}\mathbf{p}_i^{(n)}\Delta t \\ \mathbf{p}_{i+1}^{(n)} &= \mathbf{p}_i^{(n)}. \end{aligned}$$

After collision, the momentum changes according to the collision rule for hard disks

[2]Maybe a little more discussion is needed here. The calculation of Lyapunov exponents includes both positive and negative terms. For the data presented in Figure 4, the calculation of λ_3 fluctuates between positive and negative values close to $\lambda = 0$; hence, the fluctuations in the log-log plot.

(of diameter σ):

$$\begin{aligned}
\mathbf{q}_{i+1}^{(j)} &= \mathbf{q}_i^{(j)} \text{ for one frame} \\
\mathbf{q}_{i+1}^{(k)} &= \mathbf{q}_i^{(k)} \text{ for one frame} \\
\mathbf{p}_{i+1}^{(j)} &= \mathbf{p}_i^{(j)} + \frac{1}{\sigma^2}(\mathbf{p}\cdot\mathbf{q})\mathbf{q} \\
\mathbf{p}_{i+1}^{(k)} &= \mathbf{p}_i^{(k)} - \frac{1}{\sigma^2}(\mathbf{p}\cdot\mathbf{q})\mathbf{q}
\end{aligned}$$

where $\mathbf{p} = \mathbf{p}_i^{(j)} - \mathbf{p}_i^{(k)}$ and $\mathbf{q} = \mathbf{q}_i^{(j)} - \mathbf{q}_i^{(k)}$. How to determine which particle to apply these rules to is the tricky part. I had to make especially sure that the code correctly determined which particle was doing the colliding; this ensured that both (1) energy and (2) momentum were conserved.

2. Collision paths play an important part in this study. Particles can either collide with one another along the "traditional" collision path (shortest distance between particles) or they can collide "outside" of the box; that is, they can meet somewhere in the box after the periodic boundary condition on particle displacement modded the position back inside the box. Therefore, it becomes a delicate balancing of calculating geometric differences between two particles that can collide. Coding the boundary conditions and collision rule was fairly straightforward, although rather difficult to debug. First, identify what is the proper collision path based on the individual particle momenta. This step is nontrivial, if the collision path needs to be rescaled back inside the box, the line-of-sight (LOS) vector must be extrapolated to box's end and the total path length between particle 1 and particle 2 must be calculated. Next, identify which particle is colliding with which. Finally, modify vector signs to ensure momentum-energy conservation condition and proper geometry is captured. See code for full description.

After capturing the physics, debugging was performed. This step was pretty arduous, as it was difficult to identify proper debugging steps to fully test code functionality. However, a debug matrix was developed and tested utilizing a simple 1D geometric reduction of the 2D problem. All tests passed proper collision time calculation and energy-momentum conservation; i.e. code was behaving as expected. Table 1 shows the debug matrix.

A little explanation of Table 1 is needed. Two particles are placed in a box (three, actually, since required by code, but only two are interacting) at fixed positions $(x_1, y_1) = (0.5447, 0.5447)$ and $(x_2, y_2) = (2.7386, 0.5477)$. The values in the table for p_1 and p_2 represent the x-component of the momentum. All motion/collisions occur exclusively in the x-plane. The tests performed from Table 1 demonstrated that the collision logic was behaving properly and that the periodic boundary conditions kept the particles in the box.

At the writing of this paper, the model code is not functioning as desired. Given time constraints, the decision was made to take Dr. Lega's existing data streams and apply Wolf's algorithm to evaluate the Lyapunov modes and exponents.

Logic Exercised	Test Case	Result
proj $(dv) > 0$	$p_1 = .1, p_2 = .3$	pass
proj $(dv) > 0$	$p_1 = -.1, p_2 = .3$	pass
proj $(dv) > 0$	$p_1 = -.3, p_2 = .1$	pass
proj $(dv) > 0$	$p_1 = -.2, p_2 = 0$	pass
proj $(dv) > 0$	$p_1 = -.2, p_2 = -.1$	pass
proj $(dv) \leq 0$	$p_1 = .1, p_2 = -.2$	pass
proj $(dv) \leq 0$	$p_1 = -.1, p_2 = -.2$	pass
proj $(dv) \leq 0$	$p_1 = .2, p_2 = -.1$	pass
proj $(dv) \leq 0$	$p_1 = .2, p_2 = .1$	pass

Table 1: Code debug/checkout

Conclusions and Future Work

Steps were presented showing the calculation process for computing Lyapunov exponents and modes of dynamical systems. Code was developed utilizing Wolf's algorithm, verified against simple dynamical systems, and work was begun to extend the algorithm to the problem of hard disks.

To publish, work needs to be performed to reconstruct the Jacobian given experimental data for disk systems with different collision rules. Current work I have completed shows that I am very close to completing the code for application to experimental data; however much verification to previous results (Eckmann et. al [3]) needs to be completed. Code to reproduce experimental results presented in this paper is available at `http://math.arizona.edu/~jdinius`.

References

[1] H.A. Posch, R. Hirschl, in *Hard ball systems and the Lorentz gas*, edited by D. Szász (Springer-Verlag, Berlin, 2000), p. 279.

[2] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, Meccannica **15**, 9 (1980); Meccanica **15**, 21 (1980).

[3] J.-P. Eckmann, C. Forster, H.A. Posch, E. Zabey, J. Stat. Phys. **118**, 813 (2005).

[4] A. Wolf, J.-B. Swift, H.-L. Swinney, J.-A Vastano, Physica **16D**, 285 (1985).

[5] V.-I Oseledec, Trans. Moscow Math Soc. **19**, 197 (1968)

[6] H.-O Peitgen, H. Jürgens, D. Saupe, *Chaos and Fractals: New Frontiers of Science* (Springer-Verlag, New York, 2004)

[7] J. Frøyland, K.-H Alfsen, Physical Review A **29**, 2929 (1984).