Chaotic Sensitivity Analysis using Shadow Operator Inversion

Sundar Ganesh  Stefanos Mavros

February 28, 2018
1 Introduction

Sensitivity analysis is a powerful tool that can give information about the behavior of a dynamical system under infinitesimal perturbation of its input. The sensitivity is nothing more than the partial derivative of an output parameter to an input one \( \frac{\partial p}{\partial x_i} \).

Usually the sensitivity analysis is performed with the method of the finite differences (FD) or the method of algorithmic differentiation (AD). The later comes in two flavors, tangent AD and adjoint AD where we are interested in the sensitivity of a parameter with respect to a certain input or every input variable respectively. AD has advantage over FD since sensitivities can be found with only one simulation for a given system and with machine precision accuracy. The former is particularly important for systems with many input parameters where adjoint sensitivities have to be calculated.

AD predicts incorrect sensitivities for systems that exhibit chaotic behavior. The values of the sensitivities are exploding after a certain number of time steps. In this report we investigate two ways of dealing with that problem. First we try Lea’s ensemble sensitivity analysis [1] approach and then we go through the implementation of Wang’s chaotic sensitivity analysis method [2] by applying it to a simpler system - namely the Lorenz Oscillator.

We are choosing to apply the sensitivity method to a molecular dynamics (MD) simulation. The system is chaotic so the sensitivities of time averaged quantities with respect to some input variable explode as expected when AD is used.

2 Molecular Dynamics Systems

Molecular dynamics simulations deal with the simulation of atoms and molecules where the interaction of the individual particles is performed by a classical approximation. The method has a broad spectrum of applications from biomolecule design to solid state physics. Due to the nature of the quantum mechanical interactions between the atoms, the quantum system is practically impossible to solve. Hence, the interactions are substituted by a classical potential surrounding each atom that emulates the behavior of the system. For very short distance two atoms repel each other while for long enough they attract.
2.1 Model

The molecules’ positions and momenta are given by the generalized coordinates \( \mathbf{q}(t) = (q_1(t) \ldots q_{N_p})^T \) and \( \mathbf{p}(t) = (p_1(t) \ldots p_{N_p})^T \). The dynamics of the system is described by the Hamiltonian, which is nothing more than the sum of the kinetic and the potential energy.

\[
\mathcal{H} = T + V
\]  

(1)

where the kinetic energy \( T \) is given by

\[
T = \frac{1}{2} \mathbf{p}^T \mathbf{M} \mathbf{p}
\]  

(2)

and the total potential is calculated by the summation of every possible pairwise potential \( V(||\mathbf{q}_i - \mathbf{q}_j||) \)

\[
V = \frac{1}{2} \sum_{i,j,i\neq j}^{N_p} V(||\mathbf{q}_i - \mathbf{q}_j||)
\]  

(3)

Then the equations of motion are given by the partial derivatives of the Hamiltonian with respect to the generalized coordinates

\[
\frac{d}{dt} \mathbf{q} = \frac{\partial \mathcal{H}}{\partial \mathbf{q}} = \mathbf{M}^{-1} \mathbf{p} \quad \text{and} \quad \frac{d}{dt} \mathbf{p} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\nabla V
\]  

(4)

The generalized coordinates can be represented by a state vector \( \mathbf{x}(t) = (\mathbf{q}(t) \mathbf{p}(t))^T \). The system is described by a system of first order differential equations in the form

\[
\frac{d}{dt} \begin{pmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{pmatrix} = \mathbf{J} \begin{pmatrix} d\mathcal{H}/d\mathbf{q} \\ d\mathcal{H}/d\mathbf{p} \end{pmatrix} \Rightarrow \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})
\]  

(5)

where \( \mathbf{J} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \) is the Poisson matrix.
2.2 Lennard-Jones Potential

In the literature there are a lot of different interatomic potentials that exhibit the required behavior. For the purposes of this project we choose to use the Lennard-Jones potential which is the simplest one. It requires only two fitting parameters $\epsilon$ and $\sigma$ which control the depth and the position of the minimum respectively.

\[
V_{LJ}(r) = 4\epsilon \left( \left( \frac{\sigma}{r} \right)^6 - \left( \frac{\sigma}{r} \right)^{12} \right)
\]  

(6)

![Lennard-Jones pairwise potential](image)

Figure 1: Lennard-Jones pairwise potential [3]

2.3 Parameters

For the simulation of the system, we used a 2D square computational domain measuring 10 units of length. We impose periodic boundary conditions (PBC) along each axis of the system. The molecules are initialized in a rectangular lat-
tice. 10 atoms were placed along each dimension so that the system contained 100 atoms in total. All parameters were set to unity and the initial velocities were chosen to have the same magnitude but directed along a uniformly distributed random angle in $[0, 2\pi]$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>1</td>
<td>$V_{LJ}$ parameter</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>1</td>
<td>&quot;</td>
</tr>
<tr>
<td>$q(t_0)$</td>
<td>-</td>
<td>rectangular grid vertices</td>
</tr>
<tr>
<td>$p(t_0)$</td>
<td>-</td>
<td>same magnitude, random angle</td>
</tr>
</tbody>
</table>

For simplicity we worked out a unit system based on all the parameters being unity.

### 2.4 Integration

For the integration of the equations of motion, we choose the Verlet velocity scheme. The scheme has second order error and is symplectic. Hence, the energy of the system remains constant. For each step of the integrator the midpoint velocity and position has to be computed, resulting in a 3-step process.

\[
\begin{align*}
    p^{i+1/2} &= p^i - \frac{\tau}{2} \nabla V(q^i) \\
    q^{i+1/2} &= q^i + \tau M^{-1}p^{i+1/2} \\
    p^{i+1} &= p^{i+1/2} - \frac{\tau}{2} \nabla V(q^{i+1})
\end{align*}
\]

For each run of the MD simulation, we first integrated for 3000 steps in order to let the system equilibrate since a state where all of the particles sit on a rectangular grid and have the same magnitude of velocities is not in equilibrium.
3 Problems with AD for Chaotic Sensitivity Analysis

In the MD simulation, we are interested in time-averaged (statistical) quantities of the system. One time-averaged quantity can be the temperature $\langle T \rangle_\tau$ of the system which is defined over a time period $\tau$. The instantaneous temperature at time step $n$ is given by

$$T_n = \frac{1}{N_p} \sum_{i=1}^{N_p} \left( \frac{p_i^2(t_n)}{2m} \right)$$

where $N_p$ is the number of particles. The time averaged temperature after a time period $\tau$ can be computed as

$$\langle T \rangle_\tau = \frac{\sum_{n=1}^{n=\tau} T_n}{\tau/\Delta t}.$$  

Then, the sensitivity of that quantity with respect to an input parameter $p_i$ is given by

$$\frac{\partial \langle T \rangle_\tau}{\partial p_i}$$

3.1 AD Sensitivity

Using the AD method for the MD simulation, we got the sensitivity of the time-averaged temperature with respect to the Lennard-Jones $\sigma$ parameter $\partial \langle T \rangle_\tau / \partial \sigma$. We observe that after a certain number of steps, the value of the derivative goes to infinity (explodes). This behavior can be seen in figure 3.1.

This phenomenon can be explained under the chaotic properties of the MD system. That means that any state of the system is infinitely sensitive to the initial conditions. Hence, if we calculate two trajectories of the systems in the phase space, with slightly perturbed input parameters, we would see those two states following two completely different trajectories. This is nicely illustrated on figure 3.1 where two states of the system stay close together only for a certain
number of steps after initialization before they diverge. Note that the plot shows the value of the instantaneous temperature for each state since it is impossible to draw the trajectory which lies in a multidimensional space.

4 Ensemble Sensitivity

One way to tackle the problem of the AD sensitivity is to use an ensemble of same systems in order to compute the sensitivity of the statistical quantities. This method is based to the fact that in an ergodic system, like an MD simulation, an average of a quantity over a period of time is the same as the average of that over a collection of identical systems (ensemble). The method is introduced by Lea et. al [1] for the Lorenz system.

So for the sensitivity of the averaged temperature over a period time $\tau$, we can instead calculate the average over a $m$ identical systems with temperature averaged over $\tau'$ such that $\tau = m\tau'$.

$$\frac{\partial \langle T \rangle_\tau}{\partial \sigma} = \left( \frac{\partial \langle T \rangle_\tau}{\partial \sigma} \right)_m$$  (11)
Figure 3: Divergence of trajectories for small perturbation in $\sigma$
We use the same method for the MD simulation slightly differently. Instead of setting a certain number of systems, initialize them and calculate the average sensitivity, we integrate only one system where after a certain number of time steps we reset to zero the value of the sensitivity. With that approach, we actually split one system into independent chunks which can be seen collectively as an ensemble. Then the first state of each chunk is working as the initial system of that particular system.

![Figure 4: Ensemble sensitivity acquired from an ensemble of 10 systems](image)

In figure 4 it can be seen that again the derivative will blow up in case that the chunk’s size is greater than a given number of time steps (\( > 1500 \)). For that reason, we have to choose in advance the chunk size in order to get an average sensitivity which is sensible for the particular system.

5 Chaotic Sensitivity Analysis

The other approach is Prof. Wang’s [2] chaotic sensitivity analysis. His method requires the inversion of the shadow operator for which we must know the Lyapunov vectors and Lyapunov exponents for the trajectory. Since the Lyapunov decomposition of the system and shadow operator inversion requires a significant number of steps with high computational complexity, the method is applied to a simpler system than the earlier MD simulations - namely, the Lorenz system.
5.1 Lorenz System

The Lorenz system \[4\] is probably the most famous chaotic dynamical system. It was derived by a model of atmospheric convection and it is a system of 3 ordinary differential equations with three parameters \(r\), \(\sigma\) and \(\beta\).

\[
\begin{bmatrix}
\frac{dx_1}{dt} \\
\frac{dx_2}{dt} \\
\frac{dx_3}{dt}
\end{bmatrix} =
\begin{bmatrix}
\sigma (x_2 - x_1) \\
x_1 (r - x_3) - x_2 \\
x_1 x_2 - \beta x_3
\end{bmatrix}
\] (12)

One set of parameters for which the system exhibits chaotic behavior is \((r = 28, \sigma = 10, \beta = 8/3)\).

For the Lorenz system we focus on the sensitivity of the time-averaged squared state components with respect to the parameter \(r\), \(\partial \langle x^2_i \rangle / \partial r\) where \(i = 1, 2, 3\).

5.2 Method

In this section, the chaotic sensitivity analysis algorithm described in [5] is derived. We begin with a trajectory in an \(N\)-dimensional space defined by the following equation.

\[\dot{x}(t) = f(x, s_1, ..., s_m)\] (13)

\(s_1, ..., s_m\) are \(m\)-parameters on which the trajectory depends. We are interested typically in functions of the trajectory \(J(x)\). Specifically in this work, we are interested in computing the sensitivities of time averages of \(J(x)\) with respect to the parameters \(s_j\). This is given by the following equation.

\[
\frac{d \langle J \rangle}{ds_j} = \frac{d}{ds_j} T \lim_{T \to \infty} \frac{1}{T} \int_0^T J(x(t)) dt
\]

\[
= \lim_{\epsilon \to 0} \frac{\langle J(x) \rangle (s_j + \epsilon) - \langle J(x) \rangle (s_j)}{(s_j + \epsilon) - s_j}
\] (15)

To this end, we define a perturbed trajectory \(x'(x)\) which is related to the original trajectory \(x\) and an \(N\)-dimensional vector field \(\delta x(x)\) through the following...
equation.

\[ x'(x) = x + \delta x(x) \] (16)

Given that \( x \) evolves according to equation (13), an evolution equation for \( x' \) is derived by differentiating the above equation with respect to time. We first note that expanding \( f(x) \) about \( f(x') \) has the following result, dropping out the dependence on the parameters \( s_j \) temporarily.

\[ f(x) = f(x') + \frac{\partial f}{\partial x} (-\epsilon \delta(x')) + O(\epsilon^2) \] (17)

Differentiating equation (16) with respect to time and using the above expansion gives the following.

\[
\begin{align*}
\dot{x}' &= \dot{x} + \frac{\partial \delta x(x)}{\partial x} \dot{x} \\
&= f(x) + \epsilon \frac{\partial \delta x(x)}{\partial x} f(x) \\
&= f(x') - \epsilon \frac{\partial f}{\partial x} \delta(x') + \epsilon \frac{\partial \delta x(x)}{\partial x} f(x') + O(\epsilon^2) \\
&= f(x') + \epsilon \left[ S_f \delta x(x') \right] + O(\epsilon^2)
\end{align*}
\] (18)

In the last step, the Shadow Operator \( S_f : \delta x \implies \delta f \) is defined. A given \( \delta x \) relates uniquely to a particular \( \delta f \). \( S_f \) is called the shadow operator because for small \( \epsilon \), the trajectory of \( x' \) which is determined by a particular \( (\delta x, \delta f) \) pair “shadows” the original trajectory \( x \). It is now to be noted that \( f(x, s_1, \ldots, s_m) \) has the following Taylor expansion about a particular parameter \( s_j \).

\[ f(x, s_1, \ldots, s_j + \epsilon, \ldots, s_m) = f(x, s_1, \ldots, s_j, \ldots, s_m) + \epsilon \frac{\partial f}{\partial s_j} + O(\epsilon^2) \] (19)

By setting \( \delta f = \partial f / \partial s_j \) in equation (18), a \( \delta x \) is uniquely determined by inverting the shadow operator \( S_f \) as \( \delta x = S_f^{-1} \delta f \). This causes \( x' \) to evolve according to a slightly perturbed equation from \( x \), as shown below. The second order and
higher terms are ignored here.

\[ \dot{x'} = f(x') + \epsilon \frac{\partial f}{\partial s_j}(x') \]

\[ = f(x') + (\epsilon S_f \delta x)(x') \]  

Equation (15) then has the following form.

\[ \frac{d}{ds_j} \langle J \rangle = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left[ \lim_{T \to \infty} \frac{\int_{0}^{T} J(x') dt - \int_{0}^{T} J(x) dt}{T} \right] \]

\[ = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} \left[ \frac{J(x') - J(x)}{\epsilon} \right] dt \]

\[ = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} \left[ \frac{\partial J(x)}{\partial x} \delta x \right] dt \]  

\[ \Rightarrow \frac{d}{ds_j} \langle J \rangle = \left\langle \frac{\partial J(x)}{\partial x} \delta x \right\rangle \]  

In summary, this means that computing the average of the directional derivative of \( J(x) \) in the direction \( \delta x \) over the entire trajectory gives the required sensitivity derivative when \( \delta x \) is fixed according to the following.

\[ \delta x = S_f^{-1} \delta f = S_f^{-1} \frac{\partial f}{\partial s_j} \]  

5.3 Shadow Operator Inversion

For the inversion of the shadow operator \( S_f \) we have to perform the Lyapunov spectrum decomposition method to get the Lyapunov vectors (LV) \( \phi_i \) and Lyapunov exponents (LE) \( \lambda_i \) of the system which will follow the relation

\[ \frac{d}{dt} \phi_i(x) = \frac{\partial f}{\partial x} \phi_i(x) - \lambda_i \phi_i(x) \]  

Then both \( \delta f \) and \( \delta x \) can be decomposed to the Lyapunov vector basis

\[ \delta x(x) = \sum_{i=1}^{n} \alpha_i^x(x) \phi_i(x) \quad \delta f(x) = \sum_{i=1}^{n} \alpha_i^f(x) \phi_i(x) \]  

The shadow operator can be now written like
\[
S_f = -\frac{\delta f}{\delta x} \alpha_i^\tau(x) \phi_i(x) + \frac{d}{dt} \alpha_i^\tau(x) \phi_i(x)
\]

\[
= -\alpha_i^\tau(x) \frac{\delta f}{\delta x} \phi_i(x) + \frac{d\alpha_i^\tau(x)}{dt} \phi_i(x) + \alpha_i^\tau(x) \frac{d\phi_i(x)}{dt}
\]  \hspace{1cm} (26)

and by using equation (25) we get

\[
S_f = \left( \frac{d\alpha_i^\tau(x)}{dt} - \lambda_i \alpha_i^\tau(x) \right) \phi_i(x)
\]  \hspace{1cm} (27)

Finally, the perturbation \( \delta f \) is given by

\[
\delta f = \sum_{i=1}^{n} S_f = \sum_{i=1}^{n} \left( \frac{d\alpha_i^\tau(x)}{dt} - \lambda_i \alpha_i^\tau(x) \right) \phi_i(x) = \sum_{i=1}^{n} \alpha_i^f(x) \phi_i(x)
\]  \hspace{1cm} (28)

So the inversion of \( S_f \) requires the solution of a system of independent ordinary differential equations for every projection component of \( \delta f \) and \( \delta x \) in the Lyapunov basis.

\[
\frac{d\alpha_i^\tau(x)}{dt} = \alpha_i^f(x) + \lambda_i \alpha_i^\tau(x)
\]  \hspace{1cm} (29)

### 6 Lyapunov Vectors and Exponents

In this section, the definition and derivation of Lyapunov Vectors (LV) and Lyapunov Exponents (LE) for a given chaotic system are discussed. Firstly, the meaning of LVs in the context of linearized disturbances is given. Following this, a methodology for the derivation of LVs from singular vectors from tangent space propagators and of LEs from the local growth rates of these LVs is also provided. Since the derivation of the LVs and LEs for a given trajectory was a self-contained task in itself, separate methodology, test case description and results sections are presented in this chapter, separate from the global versions of these sections, for the verification of the methodology. It is important to note that although the procedure of [6] and [5] is cited by Wang as the method to obtain the LVs and LEs, Wang does not follow this method in the test case of
the Lorenz Oscillator and hence, the LVs predicted in this study are different than the ones predicted in [2].

6.1 Definition

LVs are vectors that characterize the growth of infinitesimal disturbances on a particular trajectory. They are defined at all points along the trajectory. LEs are the long-time averaged local growth rates of the corresponding LVs. They give a notion of how the corresponding LV grows over a long period of time. Both LEs and LVs give important insight into mechanisms behind instabilities [5]. For example, a given perturbation can be resolved into its components along each of the LVs. Components along LVs with negative LEs will eventually die out over a long-time whereas components with positive LVs will eventually explode. A perturbation will eventually turn towards the LV with the largest positive LE when moving forward in time.

An example of this is given in Figure 5. Initially orthonormal perturbations all eventually begin turning towards the LV with the largest positive LE when
propagating forward in time. A variation of this methodology is used to find the first \( n \) largest and smallest LVs by Wolfe et al [6]. In their work, LVs are calculated in a norm-independent and efficient manner by exploiting the dependence of the LVs on the asymptotic forward and backward singular vectors of the propagator of the tangent space equation of the trajectory. The detailed mathematical formulation is given in the following section.

6.2 Formulation

Consider an \( N \)-dimensional trajectory \( \mathbf{x}(t) \) governed by the following equation.

\[
\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}) \tag{30}
\]

The tangent space equation corresponding to this equation governs the evolution of perturbations \( \mathbf{y}(t) \) of the system at a particular point \( \mathbf{x}(t) \).

\[
\dot{\mathbf{y}}(t) = \mathbf{A}[\mathbf{x}(t)]\mathbf{y}(t) \tag{31}
\]

where \( \mathbf{A}[\mathbf{x}(t)] = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \)

The propagator \( \mathcal{L}(t_2, t_1) \) of the tangent space equation evolves solutions \( \mathbf{y}(t_1) \) of (31) to \( \mathbf{y}(t_2) \). It is given by

\[
\mathcal{L}(t_2, t_1) = \mathbf{Z}(t_2)\mathbf{Z}(t_1)^{-1} \tag{32}
\]

where \( \mathbf{Z}(t) \) is matrix whose columns are linearly independent solutions of equation (31). It is known as a fundamental solution matrix.

The methodology of Wolfe et al relies on computing the singular value decomposition (SVD) of the propagator matrix defined by equation (32). This SVD results in a set of left and right singular vectors with the following properties.

\[
\sigma_j \xi_j(t_2; t_1, t_2) = \mathcal{L}(t_2, t_1)\xi_j(t_1; t_1, t_2) \tag{33}
\]

\[
\sigma_j \xi_j(t_1; t_1, t_2) = \mathcal{L}^\dagger(t_2, t_1)\xi_j(t_2; t_1, t_2) \tag{34}
\]

Here, \( \xi_j(t_2; t_1, t_2) \) and \( \xi_j(t_1; t_1, t_2) \) are the left and right singular vectors of the matrix \( \mathcal{L}(t_2, t_1) \). \( \sigma_j \) is their corresponding singular value. The singular vectors
\(\xi_j(t; t_1, t_2), t \in [t_1, t_2]\) of the propagator matrix have a very important property that makes them useful in the computation of LVs. Namely, they are vectors that maximize the growth of perturbations initialized at \(t_1\) and optimized at \(t_2\) in the \(L_2\)-norm. Mathematically speaking, the leading singular vectors are quantities that maximize the quantity \(J\) given by

\[ J_j = \frac{\|\xi(t_2; t_1, t_2)\|^2}{\|\xi(t_1; t_1, t_2)\|^2} = \frac{\xi(t_1; t_1, t_2)^T \mathcal{L}(t_2, t_1) \mathcal{L}(t_2, t_1) \xi(t_1; t_1, t_2)}{\xi(t_1; t_1, t_2)^T \xi(t_1; t_1, t_2)} \]  

(35)

Of importance in the derivation of LVs at a particular point in the trajectory \(x(t)\) are the asymptotic forward and backward singular vectors defined according to the following equations.

\[ \hat{\eta}_j(t) = \lim_{t_1 \to \infty} \xi_j(t; t_1, t) \]  

(36)

\[ \hat{\xi}_j(t) = \lim_{t_2 \to \infty} \xi_j(t; t, t_2) \]  

(37)

These are obtained from the propagators \(\mathcal{L}(t, t_1)\) and \(\mathcal{L}(t_2, t)\). The asymptotic singular vectors \(\hat{\eta}(t)\) and \(\hat{\xi}(t)\) both form complete basis sets in which the LVs can be expressed. Based on the arguments of [6] on the dependence of the \(n\)th LV \(\phi_n(t)\) on \(\hat{\eta}_j\) and \(\hat{\xi}_j\) has the following form.

\[ \phi_n = \sum_{i=n}^N \langle \phi_n, \hat{\xi}_i \rangle \hat{\xi}_i = \sum_{j=1}^n \langle \phi_n, \hat{\eta}_j \rangle \hat{\eta}_j \]  

(38)

This can be rewritten into a matrix equation for the projection components \(\langle \phi_n, \eta_j \rangle\) or \(\langle \phi_n, \xi_i \rangle\). The equation for the coefficients \(\langle \phi_n, \eta_j \rangle\) of the leading \(n\) LVs has the form

\[ D^{(n)} y^{(n)} = 0, \]  

(39)

where \(D^{(n)}_{kj} = \sum_{i=1}^{n-1} \langle \hat{\eta}_k, \hat{\xi}_i \rangle \langle \hat{\xi}_i, \hat{\eta}_j \rangle, k, j \leq n \)  

(40)

and \(y^{(n)}_j = \langle \hat{\eta}_j, \phi_n \rangle, j = 1, 2, ..., n \)  

(41)

and can also be derived similarly for the trailing \(n\) LVs. This amounts to finding the null space basis of this matrix. The matrix so derived always has rank \(n - 1\) and hence, the null space basis consists of just one normalized vector. Once the projection coefficients are derived by finding this vector, the normal-
Figure 6: Analysis Trajectory

ized Lyapunov vector can be reconstructed from the asymptotic singular vectors.

Local Lyapunov Exponents (LLE) are defined according to the following equation.

$$\text{LLE}_j := \frac{1}{\|\phi_n\|_2} \frac{d\|\phi_n\|_2}{dt}$$  \hspace{1cm} (42)

Global LEs $\lambda_j$ are then obtained by time averaging each LLE over a long enough analysis time.

$$\lambda_j = \langle \text{LLE}_j \rangle_t$$  \hspace{1cm} (43)

6.3 Algorithm Description

The following is the step by step algorithm used to obtain the LVs and LEs from a given trajectory.

1. A trajectory from time $t = -T_B$ to $t = T_A + T_B$ is calculated. The analysis time over which averaging is carried out is $t \in [0, T_A]$, as shown in Figure 6. $T_B$ is chosen to be larger than $1/|\lambda_i|$ where $|\lambda_i|$ is the smallest expected LE.

2. For a given analysis point $t_a$ in the trajectory, a window of time is selected at $t_1 = t_a - \Delta t$ and $t_2 = t_a + \Delta t$. $\Delta t$ is an interval large enough to ensure convergence of the asymptotic singular vectors up to an order $10^{-3}$. Alternatively, instead of fixing a value, the window can be iteratively expanded, checking convergence at each step.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_A$</td>
<td>10.0</td>
</tr>
<tr>
<td>$T_B$</td>
<td>10.0</td>
</tr>
<tr>
<td>Time step</td>
<td>0.01</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>$0.5T_B$</td>
</tr>
</tbody>
</table>

Table 1: Parameter values for simulation

3. A set of $N$ disturbances is initialized at the times $t_1$ and $t_a$. Disturbance $y_i(t)$ is given by $[0,\ldots,1,\ldots,0]$ with a 1 in the $i^{th}$ position. Hence, the set of disturbances is orthonormal.

4. Each disturbance is propagated forward from $t_1$ to $t_a$ and $t_a$ to $t_2$ according to equation (31). From these two sets of $N$ orthonormal solutions to equation (31), the propagators required for equations (36) and (37) are constructed

$$L(t_a,t_1) = Z_f(t_a)Z_f^{-1}(t_1)$$  \hspace{1cm} (44)

$$L(t_2,t_a) = Z_i(t_2)Z_i^{-1}(t_a)$$  \hspace{1cm} (45)

where $Z_f(t)$ is a matrix whose columns are the perturbations initialized at $t_1$ and $Z_i(t)$ is a matrix whose columns are the perturbations initialized at $t_a$.

5. $\hat{\eta}(t_a)$ and $\hat{\xi}(t_a)$ are obtained as the left and right singular vectors of the propagators $L(t_a,t_1)$ and $L(t_2,t_a)$ respectively.

6. Having obtained $\hat{\eta}$ and $\hat{\xi}$, the matrix equation 39 is set up and solved to obtain projection coefficients. The coefficients are then used to reconstruct each LV from the asymptotic singular vector basis.

7. After computing $\phi_n(t)$ for all $t \in [0, T_A]$, the LLEs corresponding to each $\phi_n(t)$ is computed through the trajectory using a discrete approximation for the time derivative. These are then averaged over the whole analysis trajectory to obtain the global LLEs.

6.4 Test Case Description

As mentioned earlier, the test case used for testing this methodology is the Lorenz Oscillator [4]. The Lorenz Oscillator is predicted to have the LLEs...
\[ \lambda_1 = 0.91, \lambda_2 = 0.0 \text{ and } \lambda_3 = -14.58 [6]. \] The list of parameters used for the algorithm can be found in Table 1. Explicit 4\textsuperscript{th} order Runge Kutta integration is used for time stepping. The initial conditions are set according to [2] as \[ \mathbf{x}(t = -2T_B) = (-8.67139571762, 4.98065219709, 25) \] to ensure that for the duration \( t \in [-T_B, T_A + T_B] \), the trajectory lies on the oscillator.

All 3 LVs are explicitly computed at every time point. Since this is an expensive process, it is also possible to compute the LVs at only some intermediate points in the trajectory and propagate these LVs forward according to equation 31 for all the other points [5]. However, one should be careful not to integrate for too long since after a certain time, due to inevitable numerical errors, all three vectors begin turning towards the leading LV. To compute the local growth rates, one set of LVs is picked at every 25\textsuperscript{th} time step and propagates them forward for the next 24 time steps. Based on how the LVs evolve, equation 42 is used to compute the LLE corresponding to each LV and equation 43 is used to compute the global LE corresponding to each LV.

### 6.4.1 Results

In Fig. 7, 8 and 9, the \( x_1 \) and \( x_3 \) components of LV1, LV2 and LV3 are plotted respectively on top of the trajectory. At every 25\textsuperscript{th} point, the newly computed LVs are taken and propagated forward according to equation 31. This is why the rapid changes in magnitude are visible. Infinitesimal perturbations initialized in these directions at a given time would evolve according to the direction and growth rate of the arrows in each of these plots.

Figure 10 shows the evolution of the LLEs computed according to equation 42. Their time averaged values are plotted on top of each curve as a dotted line. These time averaged values give the value of the LE corresponding to each of the LVs. They are as follows - \( \lambda_1 = 0.90, \lambda_2 = 0.04 \) and \( \lambda = -14.53 \). These are very close to the values reported by [6].

In conclusion, the methodology proposed here after reading through Wolfe \textit{et al} and Norwood \textit{et al} was tested and found to predict closely the global LEs.
Figure 7: LV1 plotted on trajectory

Figure 8: LV2 plotted on trajectory

Figure 9: LV3 plotted on trajectory

Figure 10: LLEs (solid) and LEs (dotted)
The LEs and LVs obtained by this process will be used in Wang’s algorithm for chaotic sensitivity analysis.

7 Back to Sensitivity Analysis

Following the calculation of LVs and global LEs, the perturbation $\delta f$ is projected onto the LV vector basis as follows.

$$\delta f(x) = \sum_{i=1}^{N} a_{i}^{f}(x)\phi_{i}(x)$$

$$\Rightarrow \langle \phi_{j}, \delta f(x) \rangle = \sum_{i=1}^{N} a_{i}^{f}(x) \langle \phi_{j}(x), \phi_{i}(x) \rangle \quad (46)$$

The matrix equation (46) is then inverted at all points along the trajectory to give the projection coefficients $a_{i}^{f}$ at all points. Figure 11 indicates the evolution of the three projection coefficients for the case $\delta f = \partial f / \partial r$ for the Lorenz Oscillator.

The next stage of the process is to perform the shadow operator inversion to
obtain the projection coefficients $a_i^x$. According to the algorithm in [2], the coefficients $a_i^f$ and $a_i^x$ are related according to the equation

$$\frac{da_i^x(x)}{dt} = a_i^f(x) + \lambda_i a_i^x(x)$$ (47)

For $\lambda_i > 0$, equation (47) is solved backward in time from an arbitrary initial condition far forward. For $\lambda_i < 0$, it is solved forward in time from an arbitrary initial condition far back in time. In either case, $a_i^x$ converges to the correct value. For $\lambda_i = 0$, the following equation is solved instead.

$$\frac{da_i^x(x)}{dt} = a_i^f(x) + \eta$$ (48)

Here, $\eta = \langle a_i^f(x) \rangle$. Wang predicted a value of $\eta \approx -2.78$ for the zero LE $\lambda_2$. Our implementation predicted a value of $\eta \approx 2.73$. The value is very close and differs by a sign. This is to be expected because the Lyapunov vectors are known only up to a normalization factor and hence, the LV2 predicted by this study could differ from Wang’s LV2 by a sign. Equation (48) gives the evolution $a_i^x(x)$ up to a constant factor. This factor is chosen in such a way that $\langle a_i^x(x) \rangle = 0$ in accordance with the algorithm.

Figures 12 and 13 show the convergent behaviour of $a_1^x$ and $a_3^x$ for different initial condition, integrated backward and forward. This behavior is as predicted by Wang [2]. Figure 14 shows the evolution of all three projection coefficients of $\delta x$ in the analysis time interval of the study. $\delta x$ was reconstructed using equation
(25) and the sensitivity $\partial \langle J \rangle / \partial r$ was computed using equation (22). However, the predicted sensitivities did not match the values predicted by Wang as of the completion of this report.

8 Conclusions

The chaotic sensitivity analysis algorithm proposed by Wang [2], using the procedure of Wolfe [6] and Norwood [5] to compute Lyapunov vectors and exponents, is implemented for the Lorenz Oscillator [4]. The goal of this implementation was to understand the specifics of the algorithm and to explore the feasibility of its extension to MD systems. Based on the results of the implementation, the following section covers the recommendations on extension to MD systems.
8.1 Recommendations on Extension to MD

Firstly, it is to be noted that the algorithm for Lyapunov vector calculation and for Lyapunov spectrum decomposition is extremely expensive. The algorithm is of complexity $O(N^3 N_t)$. Hence, even for very simple MD systems, the algorithm could take an unpractical amount of time. Secondly, the boundary condition formalism is yet to be developed for the tangent space equation for MD systems. In the Lorenz attractor, the phase space trajectory is allowed to start from anywhere but will still eventually end up on the Lorenz attractor. However, for even simple MD systems, boundary conditions such as periodicity or thermal reflection introduce complexity into the problem. Thirdly, since the Lyapunov exponents of the system are not \textit{a priori} known, some degree of experimentation with the time scales $T_A$ and $T_B$ is required before convergent asymptotic singular vectors can be obtained. Lastly, since MD systems often have symmetries present in them, it is highly plausible that degenerate Lyapunov exponents exist. Unfortunately, the procedure of Wolfe \textit{et al.} is not guaranteed to convergence in the case of degeneracies.

Although extremely expensive to compute, Lyapunov exponents and vectors provide an extremely detailed view into instability mechanisms of a given system. For example, in [5], it was reported that local spikes in the Lyapunov vector growth rates were correlated with the change of regime of the oscillator from “cold” to “hot”. Lyapunov vectors also have seen widespread use in instability prediction in meteorological studies.

8.2 Summary of Results

1. It was shown that automatic differentiation produced diverging sensitivities in chaotic systems. In particular, a molecular dynamics system was simulated and the sensitivity of the time-averaged temperature with respect to the parameters of the interaction potential was calculated. This quantity was found to diverge.

2. Ensemble-average calculations were also computed and found to produce diverging sensitivities. However, the method was deemed to be too expensive due to the number of ensembles needed to produce a small degree of
3. To remedy this, the technique proposed by Wang [2] based on shadow operator inversion was investigated. The technique was implemented for the Lorenz Oscillator.

4. The technique required the computation of Lyapunov vectors and exponents along a given trajectory using the asymptotic singular vector method proposed by Wolfe et al. Lyapunov vectors and Local Lyapunov exponents were successfully computed. Global Lyapunov exponents were calculated based on time averaging the local values and found to closely match the values reported by Wolfe et al. and Wang.

5. The code infrastructure for computing the Lyapunov spectrum decomposition of the perturbation of $\delta f$, the shadow operator inversion and the reconstruction of the trajectory perturbation $\delta x$ has been put in place. The correct value of $\eta = \langle a_f^2(\mathbf{x}) \rangle$ is predicted up to a sign flip for the zero Lyapunov vector. Correct trends were predicted for the non-zero Lyapunov vectors - namely that the solution converged to the same value independent of initial condition when integrating forward/backward depending on the sign of the Lyapunov exponent.

6. However, as of the time of writing this report, the produced sensitivities did not yet match with the values reported by Wang.

9 Future Scope

We propose the following as the future scope for this work. The implementation of shadow operator inversion is yet to produce the correct sensitivity values reported by Wang [2]. Following this, the algorithm could be implemented for a simple molecular dynamics system of small size. Also, a newer more efficient version of this algorithm - the Non-Intrusive Least Squares Shadowing (NILSS) method, was released recently [8]. One can also look into the implementation of this algorithm. Lastly, the successfully implemented algorithm could also be used to study other types of chaotic systems.
References


