Appendix A
Numerical Calculation of Lyapunov Exponents

A.1 The Variational Equation

There is a large variety of numerical schemes for calculating the Lyapunov exponents. Each method possesses certain advantages and disadvantages, and a detailed comparison can be found in [13]. The most common methods can be divided into two main classes or families. Those based on the direct calculation of the distance between trajectories and those based on solving the variational equation.

The first family of methods, named as of *differences* or *direct*, derives from the definition of exponents as indicators of the divergence between points initially close enough. These schemes start by selecting two very close points, separated by a distance d_0. The system is iterated and the new separation is computed. The logarithm of the new and old separation is calculated and the scheme is repeated, averaging the obtained results.

For flows, the distance between trajectories can be easily computed as the phase space distance, but it would be worth a discussion on which norm could be used for the computation of the distance. For maps, the distance can also be calculated after a given number of iterations.

These methods imply a renormalisation process. That is, the distance should be set again to be the initial one after every certain number of steps. The reason is that when dealing with orbits within an attractor, the orbits do not diverge at a certain time scale, and may even begin to converge.

For flows, the renormalisation is done every t time units. This means the existence of a scale factor that when multiplied by the distance $d(t)$ will return the original distance d_0. The asymptotic Lyapunov exponent can be obtained from averaging the logarithm of the new and old separation after applying the scale factor. For the calculation of all exponents and not only the largest one, the normalisation process is somehow more complex [10]. Basically, the computation of the second exponent can be done by considering the evolution of a two-dimensional surface evolving

J.C. Vallejo, M.A.F. Sanjuan, *Predictability of Chaotic Dynamics*,
Springer Series in Synergetics, DOI 10.1007/978-3-319-51893-0

with $e^{(\lambda_1+\lambda_2)t}$. So, λ_2 can be computed once we have λ_1. The remaining exponents can be derived following the same idea.

One major disadvantage of the differences method is the selection of a practical initial distance d_0, and also a proper selection of the renormalisation period t. Since the strict definition requires infinitesimal deviations, to take finite initial deviations may lead to wrong results. It may happen that starting close to a limit cycle of exponent zero, the solution could be deviated towards initial conditions where the flow may converge. Conversely, when the points are as farther from the attractor as possible, there could be saturation effects. As a consequence, they cannot move away any farther, and the distance may keep roughly constant.

As a consequence of the above, nowadays it is more common to use the variational methods. The finite-time Lyapunov exponents are computed by solving the variational equation, that reflects the growth rate of the orthogonal semiaxes (equivalent to the initial deviation vectors) of one ellipse centred at the initial position as the system evolves [2]. The variational equation is essential when analysing the stability of orbits, evolution of phase space volumes under the dynamics, stable and unstable manifolds, and, obviously, Lyapunov exponents.

We define here $\boldsymbol{\Phi}(\boldsymbol{x}, t)$ the solution of the flow equation. The time evolution of a phase-space point subject to a given flow dynamics is given by flow equations $\dot{\boldsymbol{x}} = \boldsymbol{\Phi}(\boldsymbol{x}, t)$.

Without loss of generality, we can detail an example based on the equations for a three-dimensional continuous flow:

$$\begin{cases} \dot{x} = f_1(x, y, z) \\ \dot{y} = f_2(x, y, z) \\ \dot{z} = f_3(x, y, z) \end{cases} \tag{A.1}$$

with initial condition $\boldsymbol{x_0} = (x_0, y_0, z_0)$.

Once the initial condition $\boldsymbol{x_0}$ is fixed, we can integrate the flow during a given time t and the initial point will follow certain trajectory in the phase space, ending in a final point $\boldsymbol{x}$.

Imagine we add a small perturbation to $\boldsymbol{x_0}$ in, say the x-direction. Evidently, the resulting initial perturbed condition vector will evolve towards a different point $\boldsymbol{x'}$. The same can be said if we perturb the initial condition in other directions (y and z, respectively).

The slopes of the flow in each direction provide a mean to know how the perturbation will evolve. It may be kept constant, enlarged, shrinked or even both, as it happens when the perturbation points out diagonally from a saddle point.

The matrix describing these slopes is the Jacobian matrix of the flow $\boldsymbol{\Phi}$, $\boldsymbol{J}$, that describes the evolution of deformations after a finite time t. So, $\boldsymbol{J} = D_v\boldsymbol{\Phi}$, contains the differential slopes in every possible direction,

$$\boldsymbol{J} = D_v\boldsymbol{\Phi} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix}, \tag{A.2}$$

We can use the Jacobian $\boldsymbol{J}$ for analysing how the perturbations, or variations, evolve under the flow dynamics.

The variations in each direction $[\delta_x]$, $[\delta_y]$ and $[\delta_z]$ are defined as vectors that will track the perturbation along each direction, as follows:

$$[\delta_x] = \begin{bmatrix} \delta_{xx} \\ \delta_{xy} \\ \delta_{xz} \end{bmatrix}, \tag{A.3}$$

$$[\delta_y] = \begin{bmatrix} \delta_{yx} \\ \delta_{yy} \\ \delta_{yz} \end{bmatrix}, \tag{A.4}$$

$$[\delta_z] = \begin{bmatrix} \delta_{zx} \\ \delta_{zy} \\ \delta_{zz} \end{bmatrix}. \tag{A.5}$$

The variation $[\delta]$ is then defined as the nine-component tensor,

$$[\delta] = \begin{bmatrix} \delta_{xx} & \delta_{yx} & \delta_{zx} \\ \delta_{xy} & \delta_{yy} & \delta_{zy} \\ \delta_{xz} & \delta_{yz} & \delta_{zz} \end{bmatrix}, \tag{A.6}$$

and the variational equation is then

$$[\dot{\delta}] = \boldsymbol{J}[\delta] = D_v \boldsymbol{\Phi}[\delta]. \tag{A.7}$$

Equivalently,

$$[\dot{\delta}] = \begin{bmatrix} \dot{\delta}_{xx} & \dot{\delta}_{yx} & \dot{\delta}_{zx} \\ \dot{\delta}_{xy} & \dot{\delta}_{yy} & \dot{\delta}_{zy} \\ \dot{\delta}_{xz} & \dot{\delta}_{yz} & \dot{\delta}_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial z} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial z} \\ \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial y} & \frac{\partial f_z}{\partial z} \end{bmatrix} \begin{bmatrix} \delta_{xx} & \delta_{yx} & \delta_{zx} \\ \delta_{xy} & \delta_{yy} & \delta_{zy} \\ \delta_{xz} & \delta_{yz} & \delta_{zz} \end{bmatrix}. \tag{A.8}$$

So, in practical terms, the variational equation following the same pattern that the original equation flow is encoded by writing

$$[\dot{\delta}_x] = \begin{bmatrix} \dot{\delta}_{xx} \\ \dot{\delta}_{xy} \\ \dot{\delta}_{xz} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial z} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial z} \\ \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial y} & \frac{\partial f_z}{\partial z} \end{bmatrix} \begin{bmatrix} \dot{\delta}_{xx} \\ \dot{\delta}_{xy} \\ \dot{\delta}_{xz} \end{bmatrix}, \tag{A.9}$$

for solving the evolution of the three-dimensional variation δ_x. Following the same approach we have

$$[\dot{\delta_y}] = \begin{bmatrix} \dot{\delta_{yx}} \\ \dot{\delta_{yy}} \\ \dot{\delta_{yz}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial z} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial z} \\ \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial y} & \frac{\partial f_z}{\partial z} \end{bmatrix} \begin{bmatrix} \dot{\delta_{yx}} \\ \dot{\delta_{yy}} \\ \dot{\delta_{yz}} \end{bmatrix}, \tag{A.10}$$

for solving the evolution of the three-dimensional variation δ_y, and similarly,

$$[\dot{\delta_z}] = \begin{bmatrix} \dot{\delta_{zx}} \\ \dot{\delta_{zy}} \\ \dot{\delta_{zz}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial z} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial z} \\ \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial y} & \frac{\partial f_z}{\partial z} \end{bmatrix} \begin{bmatrix} \dot{\delta_{zx}} \\ \dot{\delta_{zy}} \\ \dot{\delta_{zz}} \end{bmatrix}, \tag{A.11}$$

for solving the evolution of the three-dimensional variation δ_z.

Therefore, we need to solve the variational equation and the system equation at the same time, working with the so-called augmented vector. For an n-dimensional system, this means to deal with a vector of $(n + n^2)$ variables. The first n variables correspond to n components of the "physical" n-dimensional vector, $\boldsymbol{x}$, and the following n^2 variables are required for solving the evolution of the n-variations, each one being an n-dimensional vector.

A.2 Selection of Initial Perturbations

By solving at the same time the flow equation and the fundamental equation of the flow, that is, the distortion tensor evolution, we can follow the evolution of the vectors, or axes, along the trajectory, and in turn, their growth rate. This method is described in [4] and [1].

The key point is that solving the variational equation implies solving the flow equations. So, for the augmented vector it must be selected a suitable initial condition, visualised as the initial axes lengths and directions of an initial deviation vector.

This is a key issue when dealing with finite integrations, because depending on this selection, the evolution of the distortion vectors will be different.

Obviously, there are several choices for the initial orientation of the ellipse axes. Due to the dependence on the finite integration time interval used in Eq. (2.11), every orientation will lead to different exponents [18].

A first simple choice may be the identity matrix, but this option does not seem to reflect any property of the flow. To allow the flow to point this initial selection to the most growing directions, and to obtain proper averaged indexes, we should integrate during long-times.

Another suitable option is to have the axes pointing to the local expanding/contracting directions, given by the eigenvectors of the Jacobian matrix. At local time scales the eigenvalues will provide insight on the stability of the point. Furthermore, these finite-time exponents can trace the stable and unstable manifolds, the latter with a time backwards integration [5, 7]. Note that in turn, the angle of both manifolds provides the nonhyperbolic nature of the system.

Another way of doing it could be to point the axes towards the direction which may have grown the most under the linearized dynamics, or to point them to the globally fastest growing direction.

Another possibility is to select the initial deviation vectors, by using the singular vectors linked to the Singular Value Decomposition (SVD) of the Jacobian matrix. The SVD takes into account that every matrix can be written in form of the product of three matrices. The first one is one formed by the left singular vectors (gene coefficient vectors, visualised as a "hanger" matrix). The second one is a diagonal matrix containing the so-called singular-values (mode amplitudes, visualised as a "stretcher" matrix). The third one is formed by the right singular vectors (expression level vectors, visualised as an "aligner" matrix).

The options listed above provide interesting insights on the behaviour of the dynamics of the flow at local scales when integrated during small finite time intervals. They are selected for pointing to directions that are already known "a priori" to express these local properties. As the integrations are larger, the local properties are washed out and the deviation vectors will end following the averaged global properties of the flow.

The axes will tend towards the fastest growing direction, may be at exponential rate, making their computation difficult to tackle. Because of that, the most commonly used methods use a Gram–Schmidt orthonormalisation process [3, 16]. By annotating the vector magnitudes before the normalization, we can calculate all Lyapunov exponents as defined in Sect. 2.1.

However, sometimes we are not merely interested in the final asymptotic values, but conversely, in looking for properties of the flow before the asymptotic values are returned. This is achieved by using the finite-time Lyapunov exponents, as described in Sects. 2.3 and 2.4.

It is worth to note that by selecting "a priori" directions, we may be already favouring the evolution towards the most growing directions. Therefore, some information about the time scales taken by the system for such an evolution may be lost.

As a consequence, there is also another interesting choice for the initial axes of the ellipse, that is to arbitrarily set them coincident with a random set of orthogonal vectors. This is the option selected along this book, used in, for instance, [14, 15].

With this selection, as the flow evolves, the axes get orientated from the arbitrary chosen direction as per the flow dynamics. Because this initial orientation will not favour any initial privileged direction, the growth rates of the axes of the ellipse will depend naturally on the flow time scales, and will begin to point to globally growing directions once a necessary finite time interval elapses.

Obviously, as the finite time interval grows above these time scales, the asymptotic regime will begin to appear. This algorithm returns the nearly asymptotic Lyapunov values ordered from the largest to the smallest when large enough time intervals are used. Conversely, there will be just a simple relationship among the exponents when using very local time scales [15]. But when using intermediate interval sizes, the returned values will characterise a given orbit.

A.3 Other Methods

The two methods described earlier are the most commonly used. But it is worthy to mention the existence of additional algorithms and methods. We refer the interested reader to follow, for instance, [13] and [9].

There are the so-called QR methods, based on the factorisation of the matrix resulting from the QR decomposition of the Jacobian, writing it as the product of an orthogonal matrix Q and an upper triangular matrix R. These methods seem to be more adequate for the computation of LCEs, but not for the computation of FTLEs, because they introduce certain errors that are only cleared as the integration time increases. There are also methods based on the SVD decomposition mentioned in the previous section. In general, both methods could not be so convenient in nonhyperbolic systems. In these systems, the FTLEs can accumulate around zero and the decomposition to have an almost degenerate spectrum. Certainly, there are corrections to the QR method that can be applied to degenerated cases. And indeed, there are modifications for proper handling of Hamiltonian systems, incorporating their symplectic nature [11]. Nevertheless, some of these modifications are only effective in systems with one or two degrees of freedom, and they are not very suitable for systems with a higher number of degrees of freedom. In any case, every method has certain advantages and disadvantages, and some additional description can be found in [13] and [12].

A final remark could be given regarding the computation of Lyapunov exponents in time series. In these cases some specific methods are required [6, 17]. If we do not have an appropriate knowledge of the fundamental equations of the system, having only experimental data available, these methods can be classified into two families: direct methods or tangent space methods.

The direct methods are based on searching for time series in the neighbourhood of the initial point, and rely on computing the necessary comparisons. The tangent space methods perform the computation by predicting the Jacobian using the available time series. See [8] and the references therein.

A.4 Practical Implementation for Building the Finite-Time Distributions

If we make a partition of the whole integration time along one orbit into a series of time intervals of size Δt, then it is possible to compute the finite-time Lyapunov exponents $\chi(\Delta t)$ for each interval.

The distribution is built by integrating the augmented vector under the flow dynamics up to a selected Δt interval. We fix the initial point of the orbit, as desired, and as initial perturbation an arbitrary set of orthonormal vectors, as described in the previous Sect. A.2. We will keep it for later use.

A small note could be raised here related to this orthonormalisation process. A widely used algorithm is the Gram–Schmidt process. This has been considered as inherently numerically unstable, that is, very sensitive to round-off errors. We can alleviate this by avoiding divisions by small numbers and column pivoting appropriately. We should aim to more numerically stable processes by using some of the available modified versions of the Gram–Schmidt algorithms or by using Householder transformations.

When integrating, at each integration step, we propagate the variations, calculating how the log of their norms evolve following Eq. (2.11), and Sect. 2.4.

When the finite time value is reached, we save the value of the calculated finite-time exponent $\chi(\Delta t)$, and start the cycle again, resetting the augmented vector. The new initial condition is the current point of the trajectory, when we have stopped. The new initial perturbation will be the one we selected previously. This is done to assure that we are comparing how the same perturbation evolves along the trajectory points.

We repeat the above process until the total integration time is reached. This integration time could be as long as needed, embracing any transient period, and going farther, or stopping the integration once the transient has ended.

This distribution of finite-time Lyapunov exponents can be normalized dividing it by the total number of intervals, thus obtaining a probability density function $P(\chi)$, that gives the probability of getting a given value χ between $[\chi, \chi + d\chi]$.

There is a huge amount of available languages and mathematical packages for solving the dynamical flow equation, the variational equation, and computing the finite-time or asymptotic infinite Lyapunov exponents. Some of them have already implemented many of the required algorithms, and the final choice for using a given language or package, commercial or free is up to the user.

References

1. Alligood, K.T., Sauer, T.D., Yorke, J.A.: Chaos. An Introduction to Dynamical Systems, p. 383. Springer, New York (1996)
2. Alligood, K.T., Sauer, T.D., Yorke, J.A.: Chaos. An Introduction to Dynamical Systems, p. 195. Springer, New York (1996)
3. Bay, J.S.: Fundamentals of Linear State Space Systems. McGraw-Hill, Boston (1999)
4. Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.M.: Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Meccanica **9**, 20 (1980)
5. Doerner, R., Hubinger, B., Martiensen, W., Grossmann, S., Thomae, S.: Stable manifolds and predictability of dynamical systems. Chaos Solitons Fractals **10**, 1759 (1999)
6. Eckmann, J.P., Oliffson, S., Ruelle, D., Ciliberto, S.: Lyapunov exponents from time series. Phys. Rev. **34**, 4971 (1986)
7. Joseph, B., Legras, B.: On the relation between kinematic boundaries, stirring and barriers. J. Atmos. Sci. **59**, 1198 (2002)
8. McCue, L.S., Troesch, A.W.: Use of Lyapunov exponents to predict chaotic vessel motions. In: Contemporary Ideas on Ship Stability and Capsizing in Waves. Fluid Mechanics and Its Applications, 1, vol. 97, Part 5, pp. 415–432. Springer, Dordrecht (2011)
9. Okushima, T.: New method for computing finite-time Lyapunov exponents. Phys. Rev. Lett. **91**, 25 (2003)
10. Parker, T.S., Chua, L.O.: Practical Numerical Algorithms for Chaotic Systems. Springer, New York (1989)
11. Partovi, H.: Reduced tangent dynamics and Lyapunov spectrum for Hamiltonian system. Phys. Rev. Lett. **82**, 3424 (1999)
12. Ramasubramanian, K., Sriram, M.S.: Lyapunov spectra of Hamiltonian systems using reduced tangent dynamics. Phys. Rev. E **62**, 4850 (2000)
13. Ramasubramanian, K., Sriram, M.S.: A comparative study of computation of Lyapunov spectra with different algorithms. Physica D **139**, 72 (2000)
14. Vallejo, J.C., Sanjuan, M.A.F.: The forecast of predictability for computed orbits in galactic models. Mon. Not. R. Astron. Soc. **447**, 3797 (2015)
15. Vallejo, J.C., Viana, R., Sanjuan, M.A.F.: Local predictibility and non hyperbolicity through finite Lyapunov exponents distributions in two-degrees-of-freedom Hamiltonian systems. Phys. Rev. E **78**, 066204 (2008)
16. Wolf, A.: Quantifying chaos with Lyapunov exponents. In: Chaos, Chap. 13. Princeton University Press, Princeton (1986)
17. Wolf, A., Swift, J., Swinney, H., Vastano, J.: Determining Lyapunov exponents from a time series. Physica D **16**, 285 (1985)
18. Ziehmann, C., Smith, L.A., Kurths, J.: Localized Lyapunov exponents and the prediction of predictability. Phys. Lett. A **271**, 237 (2000)