Numerical Techniques for Approximating Lyapunov Exponents and Their Implementation

Luca Dieci
School of Mathematics,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: dieci@math.gatech.edu

Michael S. Jolly
Department of Mathematics,
Indiana University,
Bloomington, IN 47405
e-mail: msjolly@indiana.edu

Erik S. Van Vleck[1]
Department of Mathematics,
University of Kansas,
Lawrence, KS 66045
e-mail: evanvleck@math.ku.edu

The algorithms behind a toolbox for approximating Lyapunov exponents of nonlinear differential systems by QR methods are described. The basic solvers perform integration of the trajectory and approximation of the Lyapunov exponents simultaneously. That is, they integrate for the trajectory at the same time, and with the same underlying schemes, as is carried out for integration of the Lyapunov exponents. Separate computational procedures solve small systems for which the Jacobian matrix can be computed and stored, and for large systems for which the Jacobian cannot be stored, and may not even be explicitly known. If it is known, the user has the option to provide the action of the Jacobian on a vector. An alternative strategy is also presented in which one may want to approximate the trajectory with a specialized solver, linearize around the computed trajectory, and then carry out the approximation of the Lyapunov exponents using techniques for linear problems. [DOI: 10.1115/1.4002088]

1 Introduction

Lyapunov exponents (LEs) are concerned with the asymptotic growth behavior of solutions of the linearized dynamics about a specific trajectory. Consider the m-dimensional nonlinear system

$$\dot{x} = f(x), \quad x(0) = x_0 \tag{1}$$

to be integrated for $t \geq 0$ (of course, it may also be that we need to integrate backward in time, $t \leq 0$). Here, f is assumed to be a smooth function. We assume that the solution of Eq. (1) exists for all $t \geq 0$ and is bounded; call $\phi^t(x_0)$ the solution. Lyapunov exponents are associated with the linearized problem

$$\dot{y} = f_x(\phi^t(x_0))y \tag{2}$$

and the formal definitions of the linear case now apply. That is, set $A(t)=f_x(\phi^t(x_0))$, and write Eq. (2) as the m-dimensional linear system

$$\dot{y} = A(t)y, \quad t \geq 0 \tag{3}$$

where A is a bounded and continuous function, taking values in $IR^{m\times m}$. At this point, the definitions of upper and lower Lyapunov exponents (LEs), and the concepts of regularity and integral separation, are exactly the same as for linear problems (see Refs. [1–4] for details). Define the numbers μ_j, $j=1,\ldots,m$, as

$$\mu_j = \lim\sup_{t\to\infty} \frac{1}{t}\log\|Y(t)e_j\| \tag{4}$$

where the e_j's are the standard unit vectors and $Y(t)$ is the solution of

$$\dot{Y} = A(t)Y, \quad Y(0) = Y_0 \text{ invertible}$$

When the sum of these numbers μ_j is minimized as we vary over all possible ICs (initial conditions) Y_0, the numbers are called the upper LEs of the system, and the columns of the optimal ICs are said to form a *normal basis*. We will write λ_j^s, $j=1,\ldots,m$, for the ordered upper LEs of Eq. (3). By working with the adjoint system

$$\dot{Z} = -A^T(t)Z$$

we analogously can define (lower) LEs, λ_j^i, $j=1,\ldots,m$, which again we consider ordered. The λ_j^i and λ_j^s then make up the end points of the so-called Lyapunov spectral intervals. In the case in which $\lambda_j^i=\lambda_j^s=\lambda_j$, for all $j=1,\ldots,m$, and

$$\lim_{t\to\infty}\frac{1}{t}\log(\det(Y(t))) = \sum_{j=1}^{m}\lambda_j$$

then the system is called regular, and in this case the exponents are found as limits. An important consequence of regularity is as follows (already in Ref. [5]): If we have a regular system with upper triangular coefficient matrix $B(t)$: $\dot{y}=B(t)y$, then, for $j=1,\ldots,m$, its Lyapunov exponents are given by

$$\lambda_j = \lim_{t\to\infty}\frac{1}{t}\int_0^t B_{jj}(s)ds \tag{5}$$

The prevalence of regularity and the dependence of the spectrum of Eq. (2) on the initial condition x_0 are addressed in the work of Oseledec [6].

The aim of this paper is to explain how a toolbox comprised of procedures for small systems, LESNLS, and large systems, LESNLL, can be used to approximate Lyapunov exponents. In particular, in this paper, we describe the algorithms that are implemented in these robust, reliable, computational procedures. The need for reliable codes to approximate Lyapunov exponents and related quantities is apparent in many application areas. In Ref. [7], Lyapunov exponents are computed for two traffic models to determine the level of chaos in the models. In Ref. [8], Lyapunov exponents are compared with the so-called periodicity ratio, which is useful in distinguishing between chaotic, regular, and irregular behaviors in nonlinear dynamical systems, while in Ref. [9], Lyapunov exponents are employed as a metric for characterizing dynamic patterns to differentiate between noise and order or chaos. The applications of Lyapunov exponents mentioned above are routinely encountered in engineering studies. In addition, in Ref. [10] Lyapunov exponents are used to study vibration localization phenomena in the context of structural dynamics, in Ref.

[1] Corresponding author.

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received May 27, 2009; final manuscript received April 20, 2010; published online September 27, 2010. Assoc. Editor: Harry Dankowicz.

[11] they are used to study the bifurcations of the global bubble dynamics in a fluid as a function of the frequency of rising bubbles, and in Ref. [12] they are used to study complexity of reaction/diffusion and hydrodynamic processes of chemical/biological engineering relevance.

Besides giving a measure on the asymptotic growth behavior of solutions of the linearized system, for nonlinear systems, Lyapunov exponents provide also other interesting information, which is routinely used in practical experiments of engineering relevance. For the concepts below, we refer to Ref. [13] for physical insight and development. For example, the most commonly used criterion for chaotic dynamics in dissipative systems is the existence of a positive Lyapunov exponent. Also, a widely adopted approximation for the dimension of an attractor is the so-called Kaplan–Yorke dimension, $\dim_{KY}$, also called Lyapunov dimension. Recall that this is defined for dissipative dynamical systems Eq. (1), and when the linearized problem is regular, as follows:

$$\text{Let}\quad k{:}\sum_{i=1}^{k}\lambda_i \geq 0 \quad \text{but} \quad \sum_{i=1}^{k+1}\lambda_i < 0 \quad \text{then} \quad \dim_{KY} = k + \frac{\sum_{i=1}^{k}\lambda_i}{|\lambda_{k+1}|} \tag{6}$$

where as usual the Lyapunov exponents are ordered: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Another widely adopted use of Lyapunov exponents for nonlinear systems is in estimating the Kolmogorov (or metric) entropy of a systems, $h(\mu)$ (see Ref. [13] for a definition). In fact, for this entropy one has

$$h(\mu) \leq \sum_{\lambda_i > 0} \lambda_i \tag{7}$$

where μ refers to the invariant measure on the attractor, and the summation is with respect to the positive Lyapunov exponents.

Remark. Observe that in the applications above, it is sufficient to approximate only a subset of the LEs, namely, only the largest (to detect chaos), or all the positive ones (to estimate the entropy), or all those giving a positive sum (to estimate the dimension). In other words, only the n largest Lyapunov exponents, $n \leq m$, need to be approximated.

2 *QR* Approaches

There are two broad classes of methods for approximating Lyapunov exponents by a change of variables to triangular form: *discrete* and *continuous* *QR* methods. These are essentially identical for the linear and nonlinear cases, using Eqs. (3) and (2), respectively. However, there are important practical algorithmic aspects in the nonlinear case, especially related to avoiding explicitly forming the Jacobian f_x, and of course the original differential Eq. (1) must be integrated along with Eq. (10) or Eq. (16) below. Subsequently, we will comment on these aspects.

The basic idea is to transform the coefficient matrix, $f_x(\phi^t(x_0))$ or $A(t)$, to upper triangular form, $B(t)$, using an orthogonal time-varying change of variables. Once we have transformed the problem to triangular form, stable upper Lyapunov exponents (and similarly for the lower exponents) may be extracted from the diagonal elements of the triangular coefficient matrix $B(t)$ (see Theorem 6.1 in Ref. [4]) as

$$\lambda_i^s = \limsup_{t\to\infty} \frac{1}{t}\int_0^t B_{ii}(s)ds \tag{8}$$

In Ref. [4] (see Theorems 6.1 and 6.2), it was shown that stability spectra, in particular, stable Lyapunov exponents and the Sacker–Sell spectrum Σ_{ED}, may be extracted from the diagonal of the transformed, upper triangular coefficient matrix. In particular, for a triangular system with coefficient matrix $B(t)$, $\Sigma_{\text{ED}} = \cup_i[\alpha_i, \beta_i]$, where

$$\alpha_i = \inf_{t_0} \liminf_{t\to\infty} \frac{1}{t}\int_{t_0}^{t_0+t} B_{ii}(s)ds, \qquad \beta_i = \sup_{t_0} \limsup_{t\to\infty} \frac{1}{t}\int_{t_0}^{t_0+t} B_{ii}(s)ds$$

Historically, it has been known since Perron [14] and Diliberto [15] that there exists a Lyapunov (bounded with bounded inverse and bounded derivative, see Ref. [1]), and orthogonal, change of variables that brings A to upper triangular form. Lyapunov [5] showed in his thesis that for regular systems the Lyapunov exponents may be extracted as the limit of the time average of the diagonal elements of the upper triangular coefficient matrix. More recently these *QR* type methods have re-emerged as numerical techniques (see Refs. [16,17]).

Nearly all numerical works of which we are aware (see references at the end) assume that the system is regular. Although a convenient and quite reasonable assumption in many practical situations, see Ref. [6], regularity is not sufficient to guarantee stability of the Lyapunov exponents, which is what we need to have in order to pursue computational procedures for their approximation. Stability for the LEs means that small perturbations in the function of coefficients, A, produce small changes in the LEs. Millionschikov [18,19] and Bylov and Izobov [20] gave conditions under which the LEs are stable and further proved that these conditions are *generic*, see Ref. [21], in the class of linear systems with continuous bounded coefficients. The key assumption needed is integral separation: "A fundamental matrix solution (written columnwise) $Y(t) = [Y_1(t), \ldots, Y_m(t)]$ is integrally separated if for $j = 1, \ldots, m-1$, there exist $a > 0$ and $d > 0$ such that

$$\frac{\|Y_j(t)\|}{\|Y_j(s)\|} \cdot \frac{\|Y_{j+1}(s)\|}{\|Y_{j+1}(t)\|} \geq de^{a(t-s)} \tag{9}$$

for all $t, s{:}\, t \geq s$." In Ref. [20] (see also Ref. [1], Theorem 5.4.8), it is proved that "If the system (3) has different characteristic exponents $\lambda_1 > \cdots > \lambda_m$, then they are stable if and only if there exists a fundamental matrix solution with integrally separated columns." For a good introduction to integral separation and some of its consequences, see Ref. [1] and the references therein, and the complementary works of Palmer [22,23],

Integral separation also plays a central role in quantifying the error in the stability spectra when using *QR* techniques. The results in Ref. [24] provide a so-called backward error analysis where it is shown that with respect to the exact orthogonal factor, one is computing a perturbed triangular problem. In Refs. [25–27], this backward error analysis is used as a starting point to provide a quantitative perturbation theory or error analysis for approximation of stability spectra by *QR* methods. These bounds depend on the local error tolerance used to integrate for Q and on structural properties of the problem itself (essentially the strength of the integral separation as compared with the size of the off diagonal terms in the upper triangular factor). An extension to some infinite dimensional problems was obtained in Ref. [28] by, in essence, combining a C^1 inertial manifold result, shadowing, and perturbation theory.

As we previously observed, often only the n most dominant (outermost to the right) spectral intervals are needed (and n can be much smaller than m). In these cases, the matrix Y_0 of initial conditions consists of n columns and thus $Y{:}\, t \in I\!R \to I\!R^{m\times n}$. With this in mind, we will henceforth restrict to the case in which n LEs are desired for an m-dimensional linear system, $m \geq n$. Although often it suffices to take $Y_0 = \binom{I_n}{0}$, for the success of the *QR* methods it is important that the initial condition matrix $Y_0 \in I\!R^{m\times n}$ be chosen at random. In essence, this is needed to guarantee that all possible growth behavior is represented in the growth of the columns of $Y(t)$ and that the n most dominant exponents will emerge, ordered, during integration. A more mathematical justification for choosing random initial condition is in Ref. [4]. In particular, almost every initial condition Y_0 will result in a fundamental matrix solution such that $\lambda_1 \geq \cdots \geq \lambda_n$.

Remark. There has been interest recently in techniques for preserving the orthogonality of approximate solutions to matrix differential equations with orthogonal solutions, such as in the case of QR factorization of a matrix solution of Eq. (3). Preserving orthogonality is important when finding Lyapunov exponents since an orthogonal transformation preserves the sum of the Lyapunov exponents. Techniques based on a continuous version of the Gram–Schmidt process are a natural idea, but they have not proven to be reliable numerically due to the loss of orthogonality when the differential equations that describe the continuous Gram–Schmidt process are approximated numerically. Still, a host of successful techniques have emerged in the last 10–15 years. Some parallel the algorithmic development of symplectic techniques for Hamiltonian systems and maintain automatically orthogonality of the computed solution: among these are Gauss–Runge–Kutta (RK) methods (see Refs. [29,30]), as well as several others, which automatically maintain orthogonality, see Refs. [31–33]. However, our extensive practical experience with orthogonality-preserving methods has led us to favor so-called projection techniques, whereby a possibly nonorthogonal solution is projected onto an orthogonal one, without loss of accuracy. The toolbox described here builds upon this algorithmic development and includes local error estimates, which make our computational procedures more efficient. We want to stress the importance of estimating, controlling the local errors during integration, and choosing the stepsizes adaptively based on the local error estimate. Not only this is needed for efficiency (as it is generally the case) but it also puts us on safe ground insofar as the validity of the previously mentioned backward error analysis results; these tell us that we are finding the LEs of a problem close to the original one, where the measure of closeness depends on the local errors committed during integration for the orthogonal factor Q. The result of our theoretical effort, and also of extensive testing of different algorithmic choices, has led us to the creation of our codes, LESNLS and LESNLL. These codes are the natural outcome of a 10 year experience in computing LEs; they represent a stable and efficient means of approximating Lyapunov exponents. It is our hope that they will be of use to the audience of this journal, particularly to those readers interested in engineering applications where stability information is required.

2.1 Discrete *QR* method. The method requires the QR factorization of $Y(t_{k+1}) \in I\!R^{m\times n}$, a matrix solution of Eq. (3) with linearly independent columns. Let $t_0=0$ and $Y_0=Q_0R_0$. Then, for $j=0,\ldots,k$, recursively define $Z_{j+1}(t)=Y(t,t_j)Q_j$, where $Y(t_j,t_j)=I$ and $Y(t,t_j)$ is a solution to Eq. (3) for $t\geq t_j$, and $Z_{j+1}:t\in[t_j,t_{j+1}]\to I\!R^{m\times n}$ is the solution of

$$\dot{Y}_{j+1} = f_x(\phi^t(x_0))Z_{j+1}, \quad t_j \leq t \leq t_{j+1}$$

$$Z_{j+1}(t_j) = Q_j \tag{10}$$

and update the QR factorization as

$$Z_{j+1}(t_{j+1}) = Q_{j+1}R_{j+1} \tag{11}$$

so that

$$Y(t_{k+1}) = Q_{k+1}[R_{k+1}R_k\cdots R_1R_0] \tag{12}$$

is the QR factorization of $Y(t_{k+1})$: $Q_{k+1}\in I\!R^{m\times n}$ and $\Pi^0_{j=k+1}R_j \in I\!R^{n\times n}$, the so-called reduced QR decomposition of $Y(t_{k+1})$. The LEs are obtained from the relation

$$\limsup_{k\to\infty}\frac{1}{k}\sum_{j=0}^{k}\log(R_j)_{ii}, \quad i=1,\ldots,n \tag{13}$$

which is equivalent to the definition (8) since $\dot{R}=BR$.

2.2 Continuous *QR* Method. Here one derives—and integrates—the differential equations governing the evolution of the Q and R factors in the QR factorization of Y. Differentiating the relation $Y=QR$, one gets $\dot{Q}R+Q\dot{R}=f_x(\phi^t(x_0))QR$, and multiplying by Q^T on the left, one gets the equation for R:

$$\dot{R} = B(t)R, \quad R(0)=R_0, \quad B(t) \coloneqq Q^Tf_x(\phi^t(x_0))Q - S \tag{14}$$

where $S\coloneqq Q^T\dot{Q}$. Since $Q^TQ=I$, S is skew symmetric, and since B is upper triangular, from Eq. (14) we have that the strict lower triangular parts of S and $Q^Tf_x(\phi^t(x_0))Q$ agree. Thus, since S is skew symmetric,

$$S_{ij} = \begin{cases} (Q^T(t)f_x(\phi^t(x_0))Q(t))_{ij}, & i>j \\ 0, & i=j \\ -S_{ji}, & i<j \end{cases} \tag{15}$$

Next, multiply $\dot{Q}R+Q\dot{R}=f_x(\phi^t(x_0))QR$ on the right by R^{-1} (R^{-1} exists since R is a fundamental matrix solution of $\dot{R}=BR$), and use Eq. (14) to obtain the differential equation for Q:

$$\dot{Q} = (I-QQ^T)f_x(\phi^t(x_0))Q + QS, \quad Q(0)=Q_0 \tag{16}$$

Note that if $m=n$, then $QQ^T=I$ and the equation for Q reduces to $\dot{Q}=QS$. To approximate the exponents, from Eq. (14), one uses (cf. Eq. (13))

$$\limsup_{t\to\infty}\frac{1}{t}\int_0^t (Q^T(s)f_x(x(s))Q(s))_{ii}ds, \quad i=1,\ldots,n \tag{17}$$

since $B_{ii}=(Q^T(s)f_x(x(s))Q(s))_{ii}$, the diagonal of the upper triangular coefficient matrix function. In practice, a quadrature rule is used to approximate the integral in Eq. (17).

Remark. We remark here that in exact arithmetic if no approximations are made when approximating the differential equations, then the discrete and continuous QR methods are equivalent. The numerical performance will depend on the particular problem, but in general one expects the continuous QR method to be more efficient since it does not rely directly on resolving exponential functions.

3 Implementations

The implementation of the QR methods for nonlinear systems is similar to what is done for linear systems (see Ref. [34]). In the nonlinear context, the new issue is that integration for the trajectory must be also carried out in order to be able to obtain the linearized system (2) and hence to be able to integrate the relevant equations of discrete or continuous QR methods; see Eqs. (10), (16), and (17). There are two ways in which this can be done: the all-at-once strategy of LESNLS and LESNLL, which is reviewed next, and the three-step strategy discussed later in this paper.

The overall goal is to compute approximations—for given values of t—to the quantities $\lambda_i(t)$, $i=1,\ldots,n$, herein defined as

$$\lambda_i(t) = \frac{1}{t}\nu_i(t), \quad i=1,\ldots,n \tag{18}$$

where the ν_is are defined as, see Eq. (13),

$$\nu_i(t) = \log R_{ii}(t), \quad i=1,\ldots,n, \tag{19}$$

or equivalently as, see Eq. (17),

$$\nu_i(t) = \int_0^t B_{ii}(s)ds = \int_0^t (Q^T(s)f_x(x(s))Q(s))_{ii}ds, \quad i=1,\ldots,n \tag{20}$$

3.1 Integration of the Nonlinear Problem. For nonlinear problems, the system

$$\dot{x} = f(x), \quad x(0) = x_0$$

must also be integrated to generate the linearized dynamics as in Eq. (10) or Eq. (16). Suppose that this has been carried out so that a sequence $x_j \approx x(t_j)$ of approximate values to the solution is known, and moreover, there is access to the vector field f. There are two basic points of view one may adopt.

(i) One can use any code, which does the one-step integration for x and returns the starting value x_j, the obtained approximation x_{j+1} at the new value $t_{j+1} = t_j + h_j$, and possibly the vector fields f_j and f_{j+1} evaluated at these approximations. At this point, x (and f_x) may be approximated locally by interpolation, and the linear procedures LESLIS-LESLIL used. This three-step strategy is examined later in this paper. The obvious advantage of this strategy is that one can integrate as desired (or appropriate) for the nonlinear system. For example, if the problem has Hamiltonian character, one might consider a symplectic integrator. The disadvantage is that this approach requires more user intervention.

(ii) The nonlinear system and the equations for the QR methods may be integrated together. This is what the computational procedures LESNLS and LESNLL do; they use the same RK scheme for the trajectory as used for the QR methods. The advantage is the overall simplicity. The remainder of this section reviews the basic schemes upon which these are based and discuss some aspects of their implementation.

3.2 Basic RK Schemes of LESNLS-LESNLL. The m-dimensional differential system (1) is integrated with one of the two embedded explicit RK schemes used for the linear codes, the Dormand–Prince pair, and the pair based on the RK 3/8 rule; again, see Sections 3.1, 3.2, and 3.3 of Ref. [34]. As there, these pairs are referred to as DP and RK38, which are RK pairs of order (5,4) and (4,3), respectively. For the values of the RK coefficients in DP and RK38, we refer to Chap. 7 of Ref. [34].

At this point, it is possible to perform error control on the trajectory to choose the next stepsize (an option in LESNLS-LESNLL). The next stepsize may be chosen according to several measures of error: that on the trajectory, as well as the error on the step for the approximate exponents and/or the orthogonal factor Q. We illustrate how the error on the trajectory is monitored. Errors in the approximate exponents and/or the orthogonal factor are handled similarly (see Ref. [34]). Monitoring of the error on the trajectory is done in the way explained in Ref. [35], with a mixed-error control with respect to a scalar tolerance in the sup-norm.

Let h be the current stepsize, and h_{new} the new stepsize to be chosen. Choose h_{new} as follows.

(a) Estimate the worse error componentwise with respect to the desired tolerance:

$$\text{err} = \max_{1 \le i \le m} (|x_{j+1}^{(i)} - \hat{x}_{j+1}^{(i)}|/[(1 + \max(|(x_j^{(i)}\ |,|(x_{j+1}^{(i)}\ |))\text{TOLT}]) \tag{21}$$

(b) Now estimate

$$h_{\text{new}} = \text{safe}|h(1/\text{err})^l| \tag{22}$$

where "safe" is a safety factor usually set at 0.8, and $l = 1/5$ for DP5 and $l = 1/4$ for RK38.

(c) Subject to the restrictions that $h/5 \le h_{\text{new}} \le 5h$, if err ≤ 1 the step is successful and accepted, otherwise rejected.

All of the stage values are saved so that after the step for the trajectory is completed, integration of the equations of the QR methods can be performed. Finally, there are important options linked to the use (or lack thereof) of the Jacobian.

For problems of small size, and if the Jacobian is easy to obtain, the user should use LESNLS, whereby the Jacobian is explicitly formed as a (m,m) matrix in the subroutine indicated in Ref. [34]. This way, the integration for the exponents conceptually proceeds precisely as in the linear code LESLIS, with the Jacobian playing the role of the coefficients' function $A(t)$. On the other hand, whenever one of the following (not exclusive) situations holds true:

- the Jacobian is not easy to write down conveniently
- we need to approximate $n \ll m$ Lyapunov exponents
- storage is at a premium and we cannot afford memory for a (m,m) matrix

then one should (must) use LESNLL, described next.

What is in a name? LESNLS refers to Lyapunov exponents nonlinear small. The name reflects the task of approximating Lyapunov exponents of a nonlinear system whose Jacobian matrix f_x is known and is small enough that it can be stored. LESNLL refers to Lyapunov exponents nonlinear large, in that it is concerned with approximating Lyapunov exponents of a nonlinear system whose Jacobian matrix f_x is not explicitly required, as is often desirable for large systems.

It should be said that whenever an approximation of the full set of m Lyapunov exponents is desired, and it is possible to compute (possibly, symbolically) and store the Jacobian, then it is usually preferable to use LESNLS.

The major difference between LESNLS and LESNLL is the way the Jacobian f_x is provided by the user. In LESNLS, the user must provide a subroutine to compute f_x at any t. In LESNLL, the user can provide a subroutine where the action $f_x(\cdot)v$ is given (here, v is a generic vector) or can bypass explicitly providing this routine by letting LESNLL generate the approximation internally. In any case, the difference in the way the Jacobian is specified is the only difference that the user will see between LESNLS and LESNLL.

3.3 LESNLL: Bypassing the Jacobian. The crux of LESNLL is to proceed using direct matrix vector products, the action of the Jacobian on a certain vector. This way of proceeding is especially effective when only a few exponents are needed, that is, when $n \ll m$.

To illustrate, consider a typical explicit RK step needed for the discrete QR method. On the step between t_j and t_{j+1}, the linear system to integrate is

$$\dot{Y} = f_x(\phi^t(x_0))Y, \quad Y(t_j) = Q_j$$

The stage values of the trajectory give the approximation

$$Y_{j+1} = Q_j + h_j \sum_{l=1}^{s} b_l K_l, \quad K_l = f_x(x_{j,l})Y_{j,l}$$

$$Y_{j,l} = Q_j + h_j \sum_{p=1}^{l-1} a_{jp} K_p, \quad l = 1,2, \ldots, s$$

Observe that the Jacobian is needed only through the actions $f_x(x_{j,l})Y_{j,l}$, that is, only the action of the Jacobian (evaluated at some value $\bar{x}$) on several vectors; say, the action $f_x(\bar{x})v$ is required. In LESNLL, there are two options for the user to approximate this action.

(i) If the coding "Jacobian × vector" is easy to do, then the user should define this action explicitly in a subroutine. For example, in evolving Burgers equation

$$u_t = u_{yy} - uu_y, \quad u(t,y) = u(t,y + 2\pi)$$

by a spectral method, the term uu_y is represented by a bilinear function $B(x,x)$ where the vector x consists of

Fourier coefficients. The contribution of B to $f_x(\phi^t(x_0))v$ is

$$B(\phi^t(x_0),v)+B(v,\phi^t(x_0))$$

and this is easy to code. Incidentally, this type of nonlinear term occurs also when dealing with several important PDEs, such as the Kuramoto–Sivashinsky equation (see Eq. (24)) and the Navier–Stokes equations.

(ii) If the action Jacobian × vector is not easily available, then this action is approximated internally by the package using discrete directional derivatives (this is done internally in LESNLL and requires no user intervention).

What LESNLL does is the following (standard) approximation:

$$f_x(\bar{x})v \approx [f(\bar{x}+\eta v)-f(\bar{x})]/\eta$$

where LESNLL chooses

$$\eta=\max(1,\|f(\bar{x})\|_2)\sqrt{\mathrm{EPS}}$$

and EPS is the machine precision. It should be understood that this way of proceeding now gives an inexact Jacobian action, and one cannot get a better accuracy than $O(\eta)$ for this evaluation. This is fine, if η is (less than or equal to) the order of the required error tolerances on the exponents; otherwise, there is a limitation on the obtainable accuracy, which practically is of $O(\sqrt{\mathrm{EPS}})$.

The situation for the continuous QR method is very similar, and LESNLL proceeds in a similar way. This is because in this case (see Eq. (16)), the Jacobian is needed only through actions of the form

$$f_x(\bar{x})Q, \quad Q\in IR^{m\times n}$$

and thus again one need only to evaluate terms like $f_x(\bar{x})v$. These are handled as above.

Remark. The Jacobian-free methods are a considerable help insofar as the evaluation of the various right-hand-sides in the equations of the QR methods. However, the outstanding computational expense of these methods remains that of the QR factorizations, which is $O(mn^2)$ flops per step. This limits all approaches to medium-sized problems or to very small values of n. The largest experiments we have performed have been on a problem with $(m,n)\approx(1000,100)$, and these were very demanding computations.

4 Three-Step Method: Using a Nonlinear Solver and LESLIS-LESLIL

As previously mentioned, an alternative to the all-at-once strategy of LESNLS-LESNLL, is a three-step strategy. On a step from t_j to $t_{j+1}=t_j+h_j$, one does the following:

1. Integrate for Eq. (1) with any desired scheme.
2. Rebuild via Hermite interpolation an approximation to the trajectory on the step.
3. Use the linear codes LESLIS or LESLIL with Jacobian evaluated at the interpolant.

The rationale for this strategy is that it may be convenient (or even necessary) to integrate for the trajectory with a better suited code than an explicit RK code. For example, the solver VODPK ([36]) can be highly effective for the integration of stiff problems. For Hamiltonian systems, the integration for the trajectory might be done with a symplectic integrator. Of course, any nonlinear solver for Eq. (1) could be used as needed.

Step 1. Integration of the nonlinear DEs.
  OUTPUT: Current time and approximation.
Step 2. LESLIS or LESLIL
  INPUT: Current time and approximation.
  OUTPUT: Initial time and number of LEs, followed by current time step, integral of diagonal elements over the current step.
Step 3. Obtain spectral information.
  INPUT: Output of step 2.
  OUTPUT: Approximate LEs, Sacker-Sell spectrum, Lyapunov dimension, integral separation information, etc.

Basic idea. In step 2, build globally defined approximation of trajectory. For example, given times $t_j<t_{j+1}$ and corresponding $x_j,x_{j+1}\in IR^m$ and the vector field defined by $f(x)$ (basically the same as that used in step 1), determine coefficients $a_j,b_j,a_{j+1},b_{j+1}\in IR^m$ such that

$$\bar{x}(t)=(a_j\cdot t+b_j)\cdot L_0^2(t)+(a_{j+1}\cdot t+b_{j+1})\cdot L_1^2(t)$$

the cubic Hermite interpolant such that $\bar{x}(t_j)=x_j$, $\bar{x}(t_{j+1})=x_{j+1}$, $\bar{x}'(t_j)=f(x_j)$, and $\bar{x}'(t_{j+1})=f(x_{j+1})$. Here $L_0(t),L_1(t)$ are linear scalar functions, the Lagrange interpolants corresponding to t_j,t_{j+1}.

Remark. For stiff problems, the benefits of a potentially less expensive integration of Eq. (1) are often wiped out in the context of approximating the exponents. In part, this is because the dominant expense is still that associated with the various QR factorizations, but there is more. In fact, often the stepsize sequence chosen by the stiff solver is inadequate to resolve the approximation of the exponents. The reason for this is not yet clear.

5 Examples

Several examples are available at websites.[2] Various drivers and test problems are available including examples of the all-at-once strategy and the three-step strategy with both stiff and symplectic integrators.

Example 1. The first example is a model of atmospheric variability [37,38] with values x_j of m points equally spaced around a circle of constant latitude:

$$\dot{x}_k=(x_{k+1}-x_{k-2})x_{k-1}-x_k+8, \quad k=0,1,\ldots,m-1 \quad (\mathrm{mod}\ m) \tag{23}$$

In the results reported in Refs. [37,38] for $m=40$ there are 13 positive LEs, the Kaplan–Yorke dimension $\dim_{KY}\approx 27$, and $\lambda_{14}=0$. As a concrete example, we considered the initial condition $x(0)=e_2$, the second unit basis vector, and solved for all LEs using LESNLS and LESNLL. We employed error tolerances on the trajectory, orthogonal factorization, and Lyapunov exponents of size 10^{-4} and the RK38 scheme. Over integration times up to 10^4, we found the computational procedure for large problem LESNLL to be approximately twice as fast as that for small problem LESNLS when computing all Lyapunov exponents. In addition, we obtained values of the Kaplan–Yorke dimension $\dim_{KY}\approx 27.06$ and found that the approximate values of the Lyapunov exponents quickly converge to relatively uniform values. We plot here the approximate LEs as a function of time for the first seven LEs in Fig. 1.

Example 2. As a second example consider the Kuramoto–Sivashinsky equation in the form

$$\frac{\partial\tilde{u}}{\partial\tau}+4\frac{\partial^4\tilde{u}}{\partial y^4}+\vartheta\left[\frac{\partial^2\tilde{u}}{\partial y^2}+\tilde{u}\frac{\partial\tilde{u}}{\partial y}\right]=0 \tag{24}$$

with $\tilde{u}(y,t)=\tilde{u}(y+2\pi,t)$ and $\tilde{u}(y,t)=-\tilde{u}(-y,t)$. With the change of variables

[2] www.math.gatech.edu/~dieci/software-les.html and www.math.ku.edu/~evanvleck/software-les.html

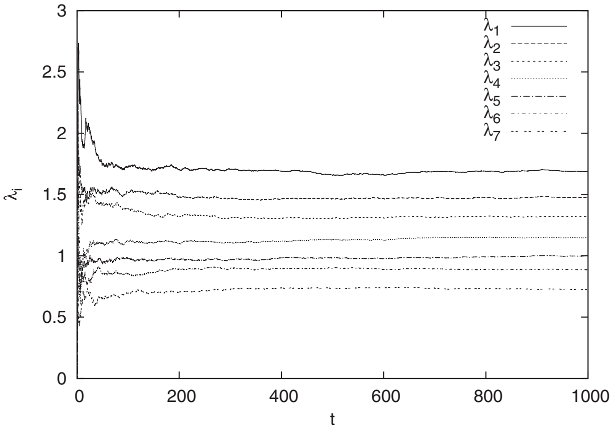

Fig. 1 First seven approximate Lyapunov exponents versus time for the model of atmospheric variability (23).

$$-2w(s,y)=\tilde{u}(\xi s/4,y), \quad \xi=\frac{4}{\vartheta}$$

Eq. (24) can be written as

$$w_s=(w^2)_y-w_{yy}-\xi w_{yyyy} \tag{25}$$

This allows us to compare with the Lyapunov exponents reported in Refs. [39,28]. All computations were performed for $\xi=0.02991$, $\vartheta=133.73454$, one of the parameter values considered in Refs. [39,28,40].

We consider a spatial discretization using standard Galerkin truncation with 16 or 32 modes (see Ref. [28] for more details). We employ LESNLL using the DP integrator for both the discrete (Table 1) and the continuous *QR* (Table 2) and the three- step strategy (Table 3) using LESLIL (with the DP integrator and the continuous *QR* method) and the stiff solver VODPK [36] to approximate the solution of Eq. (1).

We report on numerical experiments to solve for the first four Lyapunov exponents over intervals of lengths 10 and 100. In Tables 1–3, *T* denotes the final time, *M* denotes the number of modes, *N* denotes the number of exponents approximated, and TOL is the value of the tolerances that were employed. We report on the approximate Lyapunov exponents obtained and record normalized CPU times. The values of the LEs obtained are consistent across the different methods and with previous results [39,28]. For $M=16$ modes, the discrete and continuous *QR* approaches take essentially the same amount of CPU time and are considerably faster than the three-step strategy. For $M=32$, the discrete *QR* method is faster than all others.

6 Discussion

We have developed a toolbox for approximation of Lyapunov exponents for nonlinear differential equations. Among the benefits of the techniques described is the ability to use explicit Jacobians, the action of the Jacobian, or a finite difference approximation of the Jacobian depending on the information available and on whether the problem is of large or small dimension. The differential equations to be approximated are solved with techniques that preserve the orthogonal structure of the factor *Q*, which guarantees that the sum of the Lyapunov exponents is preserved. Further avenues of research include coupling the toolbox with a recently developed error analysis as well as application to problems of interest in science and engineering [41,42].

Acknowledgment

This work was supported by National Science Foundation Focused Research Grant Nos. DMS-0139895, DMS-0139874, and DMS-0139824, and Grant Nos. DMS-0511533, DMS-0513438, and DMS-0812800.

Table 1 The discrete *QR* method applied to the Kuramoto–Sivashinsky equation using the Dormand–Prince pair

T	M	N	TOL	λ_1	λ_2	λ_3	λ_4	$\Sigma_{j=1}^4\lambda_j$	CPU
10	16	4	1×10^{-5}	87.715079	0.10104269	−762.91721	−1139.7821	−1814.88	1
100	16	4	1×10^{-5}	84.543528	−0.0036706406	−760.01825	−1136.9618	−1812.44	10.0
10	16	4	1×10^{-8}	84.186532	0.0062248129	−759.54664	−1136.8065	−1812.16	2.8
100	16	4	1×10^{-8}	83.917786	−0.0048723886	−759.47680	−1136.2823	−1811.85	28.2
10	32	4	1×10^{-5}	85.075006	−0.28543797	−757.78116	−1128.0010	−1801.00	63.2

Table 2 The continuous *QR* method applied to the Kuramoto–Sivashinsky equation using the Dormand–Prince pair and the projected method

T	M	N	TOL	λ_1	λ_2	λ_3	λ_4	$\Sigma_{j=1}^4\lambda_j$	CPU
10	16	4	1×10^{-5}	80.787364	−0.0811046	−756.33708	−1136.1482	−1811.77	1.03
100	16	4	1×10^{-5}	83.655799	−0.0002499180	−759.12671	−1136.8852	−1812.36	10.2
10	16	4	1×10^{-8}	84.057142	−0.110254327	−759.33644	−1136.8114	−1812.20	2.9
100	16	4	1×10^{-8}	83.982066	−0.0019134863	−759.46494	−1136.7057	−1812.19	28.4
10	32	4	1×10^{-5}	84.961902	0.32904663	−758.65851	−1124.8598	−1798.23	72.9

Table 3 The three-step strategy applied to the Kuramoto–Sivashinsky equation with VODPK used to the approximate the original nonlinear equation and the Dormand–Prince pair and the projected method to approximate the orthogonal factor

T	M	N	TOL	λ_1	λ_2	λ_3	λ_4	$\Sigma_{j=1}^4\lambda_j$	CPU
10	16	4	1×10^{-5}	85.132041	0.1193929	−760.43735	−1141.4009	−1816.59	1.8
100	16	4	1×10^{-5}	83.030609	−0.045702566	−758.47894	−1137.2718	−1812.67	18.5
10	16	4	1×10^{-8}	83.555912	−0.07864597	−759.21634	−1137.0916	−1812.83	4.8
100	16	4	1×10^{-8}	84.244061	0.0069730059	−759.69436	−1137.0974	−1812.54	48.8
10	32	4	1×10^{-5}	88.068104	−0.16851648	−760.07970	−1129.7976	−1801.98	77.2

References

[1] Adrianova, L. Ya., 1995, *Introduction to Linear Systems of Differential Equations*, Translations of Mathematical Monographs, AMS, Providence, RI, Vol. 146.

[2] Dieci, L., and Van Vleck, E. S., 2002, "Lyapunov Spectral Intervals: Theory and Computation," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **40**, pp. 516–542.

[3] Dieci, L., and Van Vleck, E. S., 2002, "Lyapunov and Other Spectra: A Survey," *Preservation of Stability Under Discretization*, D. Estep and S. Tavener, eds., SIAM, Philadelphia, PA.

[4] Dieci, L., and Van Vleck, E. S., 2007, "Lyapunov and Sacker-Sell Spectral Intervals," J. Dyn. Differ. Equ., **19**, pp. 265–293.

[5] Lyapunov, A., 1992, "The General Problem of the Stability of Motion," Int. J. Control, **55**, pp. 521–590.

[6] Oseledec, V. I., 1998, "A Multiplicative Ergodic Theorem. Lyapunov Characteristic Numbers for Dynamical Systems," Trans. Mosc. Math. Soc., **19**, pp. 197–231.

[7] Xu, M., and Gao, Z., 2008, "Nonlinear Analysis of Road Traffic Flows in Discrete Dynamical System," ASME J. Comput. Nonlinear Dyn., **3**, p. 021206.

[8] Dai, L., 2008, "Implementation of Periodicity Ratio in Analyzing Nonlinear Dynamic Systems: A Comparison With Lyapunov Exponent," ASME J. Comput. Nonlinear Dyn., **3**, p. 011006.

[9] Arasteh, D., 2008, "Measures of Order in Dynamic Systems," ASME J. Comput. Nonlinear Dyn., **3**, p. 031002.

[10] Bendiksen, O., 2000, "Localization Phenomena in Structural Dynamics," Chaos, Solitons Fractals, **11**, pp. 1621–1660.

[11] Blomgren, P., Palacios, A., Zhu, B., Daw, S., Finney, C., Halow, J., and Pannala, S., 2007, "Bifurcation Analysis of Bubble Dynamics in Fluidized Beds," Chaos, **17**, p. 013120.

[12] Elnashaie, E., and Grace, J., 2007, "Complexity, Bifurcation and Chaos in Natural and Man-Made Lumped and Distributed Systems," Chem. Eng. Sci., **62**, pp. 3295–3325.

[13] Ott, E., 1993, *Chaos in Dynamical Systems*, Cambridge University Press, New York.

[14] Perron, O., 1930, "Die Ordnungszahlen Linearer Differentialgleichungssystemen," Math. Z., **31**, pp. 748–766.

[15] Diliberto, S. P., 1950, "On Systems of Ordinary Differential Equations," *Contributions to the Theory of Nonlinear Oscillations* (Ann. of Math. Studies 20), Princeton University Press, Princeton, NJ, pp. 1–38.

[16] Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J.-M., 1980, "Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems; a Method for Computing All of Them. Part 1: Theory," Meccanica, **15**, pp. 9–20.

[17] Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A., 1985, "Determining Lyapunov Exponents From a Time Series," Physica D, **16**, pp. 285–317.

[18] Millionshchikov, V. M., 1969, "Systems With Integral Division Are Everywhere Dense in the Set of All Linear Systems of Differential Equations," Differentsial'nye Uravneniya, **5**, pp. 1167–1170.

[19] Millionshchikov, V. M., 1969, "Structurally Stable Properties of Linear Systems of Differential Equations," Differentsial'nye Uravneniya, **5**, pp. 1775–1784.

[20] Bylov, B. F., and Izobov, N. A., 1969, "Necessary and Sufficient Conditions for Stability of Characteristic Exponents of a Linear System," Differentsial'nye Uravneniya, **5**, pp. 1794–1903.

[21] Palmer, K. J., 1979, "The Structurally Stable Systems on the Half-Line Are Those With Exponential Dichotomy," J. Differ. Equations, **33**, pp. 16–25.

[22] Palmer, K. J., 1982, "Exponential Dichotomy, Integral Separation and Diagonalizability of Linear Systems of Ordinary Differential Equations," J. Differ. Equations, **43**, pp. 184–203.

[23] Palmer, K. J., 1982, "Exponential Separation, Exponential Dichotomy and Spectral Theory for Linear Systems of Ordinary Differential Equations," J. Differ. Equations, **46**, pp. 324–345.

[24] Dieci, L., and Van Vleck, E. S., 2005, "On the Error in Computing Lyapunov Exponents by *QR* Methods," Numer. Math., **101**, pp. 619–642.

[25] Dieci, L., and Van Vleck, E. S., 2006, "Perturbation Theory for Approximation of Lyapunov Exponents by *QR* Methods," J. Dyn. Differ. Equ., **18**, pp. 815–840.

[26] Dieci, L., and Van Vleck, E. S., 2008, "On the Error in *QR* Integration," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **46**, pp. 1166–1189.

[27] Van Vleck, E. S., 2010, "On the Error in the Product *QR* Decomposition," SIAM J. Matrix Anal. Appl., **31**, pp. 1775–1791.

[28] Dieci, L., Jolly, M. S., M. S. R. Rosa, and E. S. Van Vleck, 2008 "Error in Approximation of Lyapunov Exponents on Inertial Manifolds: The Kuramoto-Sivashinsky Equation," Discrete Contin. Dyn. Syst., Ser. B, **9**, pp. 555–580.

[29] Dieci, L., Russell, R. D., and Van Vleck, E. S., 1997, "On the Computation of Lyapunov Exponents for Continuous Dynamical Systems," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **34**, pp. 402–423.

[30] Calvo, M. P., Iserles, A., and Zanna, A., 1997, "Numerical Solution of Isospectral Flows," Math. Comput., **66**, pp. 1461–1487.

[31] Chu, M. T., 1988, "On the Continuous Realization of Iterative Processes," SIAM Rev., **30**, pp. 375–387.

[32] Dieci, L., and Van Vleck, E. S., 1995, "Computation of a Few Lyapunov Exponents for Continuous and Discrete Dynamical Systems," Appl. Numer. Math., **17**, pp. 275–291.

[33] Dieci, L., and Van Vleck, E. S., 1999, "Computation of Orthonormal Factors for Fundamental Solution Matrices," Numer. Math., **83**, pp. 599–620.

[34] Dieci, L., and Van Vleck, E. S., "LESLIS and LESLIL: Codes for Approximating Lyapunov Exponents of Linear Systems," see http://www.math.gatech.edu/~dieci/software-les.html

[35] Hairer, E., Nœrsett, S. P., and Wanner, G., 1993, *Solving Ordinary Differential Equations I*, 2nd ed., Springer-Verlag, Berlin, Heidelberg.

[36] Brown, P., and Hindmarsh, A., 1986, "Matrix Free Methods for Stiff Systems of ODEs," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **23**, pp. 610–638.

[37] Lorenz, E., 1995, "Predictability. A Problem Partly Solved," Proceedings on Predictability, ECMWF, Sept. 4–8, pp. 1–18.

[38] Lorenz, E., and Emmanuel, K., 1998, "Optimal Sites for Supplementary Weather Observations: Simulations With a Small Model," J. Atmos. Sci., **55**, pp. 399–414.

[39] Christiansen, F., Cvitanović, P., and Putkaradze, V., 1997, "Spatiotemporal Chaos in Terms of Unstable Recurrent Patterns," Nonlinearity, **10**, pp. 55–70.

[40] Smyrlis, Y. S., and Papageorgiou, D. T., 1991, "Predicting Chaos for Infinite-Dimensional Dynamical Systems: The Kuramoto-Sivashinsky Equation, a Case Study," Proc. Natl. Acad. Sci. U.S.A., **88**, pp. 11129–11132.

[41] Constantin, P., and Foias, C., 1985, "Global Lyapunov Exponents, Kaplan-Yorke Formulas and the Dimension of the Attractors for 2D Navier-Stokes Equations," Commun. Pure Appl. Math., **38**, pp. 1–27.

[42] Ruelle, D., 1989, *Chaotic Evolution and Strange Attractors*, Cambridge University Press, Cambridge.