

See discussions, stats, and author profiles for this publication at: <http://www.researchgate.net/publication/228613270>

# Numerical Techniques for Approximating Lyapunov Exponents and Their Implementation

ARTICLE *in* JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS · JANUARY 2011

Impact Factor: 1.53 · DOI: 10.1115/1.4002088

---

CITATIONS

2

---

DOWNLOADS

46

---

VIEWS

83

3 AUTHORS, INCLUDING:



Michael Jolly

Indiana University Bloomington

57 PUBLICATIONS 807 CITATIONS

SEE PROFILE

# Numerical Techniques for Approximating Lyapunov Exponents and Their Implementation <sup>\*</sup>

Luca Dieci<sup>†</sup>

School of Mathematics  
Georgia Institute of Technology  
Atlanta, Georgia 30332

Michael S. Jolly<sup>‡</sup>

Department of Mathematics  
Indiana University  
Bloomington, Indiana 47405

Erik S. Van Vleck<sup>§</sup>

Department of Mathematics  
University of Kansas  
Lawrence, Kansas 66045

## Abstract

*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 integration for the Lyapunov exponents is carried out. 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*

---

<sup>\*</sup>This work was supported by National Science Foundation Focused Research Grant Numbers DMS-0139895, DMS-0139874, and DMS-0139824, and Grant Numbers DMS-0511533, DMS-0513438, and DMS-0812800.

<sup>†</sup>e-mail: [dieci@math.gatech.edu](mailto:dieci@math.gatech.edu)

<sup>‡</sup>e-mail: [msjolly@indiana.edu](mailto:msjolly@indiana.edu)

<sup>§</sup>Corresponding author. e-mail: [evanvleck@math.ku.edu](mailto:evanvleck@math.ku.edu)

*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.*

## INTRODUCTION

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

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

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. Let assume that the solution of (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

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

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

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

where  $A$  is a bounded and continuous function, taking values in  $\mathbb{R}^{m \times m}$ . At this point, the definitions of upper and lower Lyapunov exponents (LEs for short), and the concepts of regularity and integral separation, are exactly the same as for linear problems (see [1, 18, 17, 21] for details). Define the numbers  $\mu_j$ ,  $j = 1, \dots, m$ , as

$$\mu_j = \limsup_{t \rightarrow \infty} \frac{1}{t} \log \|Y(t)e_j\|, \quad (4)$$

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  $\mu_j$  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  $\lambda_j^s$ ,  $j = 1, \dots, m$ , for the ordered upper LEs of (3). By working with the adjoint system

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

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

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

then the system is called **regular**, and in this case the exponents are found as limits. A most important consequence of regularity is (already in [29]): “If we have a regular system with upper triangular coefficient matrix  $B(t) : \dot{y} = B(t)y$ , then, for  $j = 1, \dots, m$ , its Lyapunov exponents are given by

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

The prevalence of regularity, and the dependence of the spectrum of (2) on the initial condition  $x_0$  is addressed in the work of Oseledec, [32].

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 [42] Lyapunov exponents are computed for two traffic models to determine the level of chaos in the models. In [12] Lyapunov exponents are compared with the so-called periodicity ratio which is useful in distinguishing between chaotic, regular, and irregular behavior in nonlinear dynamical systems, while in [2] 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 [3] Lyapunov exponents are used to study vibration localization phenomena in the context of structural dynamics, in [5] they are used to study the bifurcations of the global bubble dynamics in a fluid in function of the frequency of rising bubbles, and in [25] 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 [33] 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 (1), and when the linearized

problem is regular, as follows:

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

where as usual the Lyapunov exponents are ordered:  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ . Another widely adopted use of Lyapunov exponents for nonlinear systems is in estimating the Kolmogorov (or metric) **entropy** of a systems,  $h(\mu)$  (see [33] for a definition). In fact, for this entropy one has

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

where  $\mu$  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.

## 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 (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 equation (1) must be integrated along with equation (10) or (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 [21]) as

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

In [21] (see Theorems 6.1 and 6.2) it was shown that stability spectra, in particular stable Lyapunov exponents and the Sacker-Sell spectrum  $\Sigma_{\text{ED}}$ , 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

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

Historically, it has been known since Perron [37] and Diliberto [24] that there exists a Lyapunov (bounded with bounded inverse and bounded derivative, see [1]), and orthogonal, change of variables that brings  $A$  to upper triangular form. Lyapunov [29] 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 reemerged as numerical techniques (see [4] and [41]).

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 [32], 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 (see [31, 30]) and Bylov and Izobov (see [7]) gave conditions under which the LEs are stable, and further proved that these conditions are *generic*, see [34, p. 21], in the class of linear systems with continuous bounded coefficients. The key assumption needed is **integral separation**: “A fundamental matrix solution (written columnwise)  $Y(t) = [Y_1(t), \dots, Y_m(t)]$  is integrally separated if for  $j = 1, \dots, m - 1$ , there exist  $a > 0$  and  $d > 0$  such that

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

for all  $t, s : t \geq s$ ”. In the cited works, it is proved that “If the system (3) has different characteristic exponents  $\lambda_1 > \dots > \lambda_m$ , 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 [1] and the references therein, and the complementary works of Palmer [35, 36],

Integral separation also plays a central role in quantifying the error in the stability spectra when using  $QR$  techniques. The results in [19] 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 [20], [22], and [40] 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 [13] 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 is made up by just  $n$  columns and thus  $Y : t \in \mathbb{R} \rightarrow \mathbb{R}^{m \times n}$ . 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  $Y_0 = \begin{pmatrix} I_n \\ 0 \end{pmatrix}$ , for the success of the QR methods it is important that the initial condition matrix  $Y_0 \in \mathbb{R}^{m \times n}$  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 [4]. In particular, almost every initial condition  $Y_0$  will result in a fundamental matrix solution such that  $\lambda_1 \geq \dots \geq \lambda_n$ .

**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 (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 methods (see [14, 8]), as well as several others which automatically maintain orthogonality, see [9, 15, 16]. However, our extensive practical experience with orthogonality-preserving methods has lead us to favor so-called projection techniques, whereby a possibly non orthogonal 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, and controlling, the local errors during integration, and choose the stepsizes adaptively based upon 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*, and 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 lead us to the creation of our codes, **LESNLS** and **LESNLL**. These codes are the naturally outcome of a 10 year experience on computation of LEs, and the codes represent a stable and efficient mean to approximate 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.

## Discrete QR method

The method requires the QR factorization of  $Y(t_{k+1}) \in \mathbb{R}^{m \times n}$ , a matrix solution of (3) with linearly independent columns. Let  $t_0 = 0$ , and  $Y_0 = Q_0 R_0$ . Then, for  $j = 0, \dots, k$ ,

progressively 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 (3) for  $t \geq t_j$ , and  $Z_{j+1} : t \in [t_j, t_{j+1}] \rightarrow \mathbb{R}^{m \times n}$  is the solution of

$$\begin{cases} \dot{Y}_{j+1} = f_x(\phi^t(x_0))Z_{j+1}, & t_j \leq t \leq t_{j+1} \\ Z_{j+1}(t_j) = Q_j, \end{cases} \quad (10)$$

and update the QR factorization as

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

so that

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

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

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

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

## 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  $R$ :

$$\dot{R} = B(t)R, \quad R(0) = R_0, \quad B(t) := Q^T f_x(\phi^t(x_0))Q - S, \quad (14)$$

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

since  $S$  is skew-symmetric,

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

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 (14) to obtain the differential equation for  $Q$ :

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

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

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

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 (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.

## IMPLEMENTATIONS

The implementation of the QR methods for nonlinear systems is similar to what is done for linear systems (see [23]). 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 (10) and (16-17). 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 steps** strategy discussed later in this paper.

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

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

where the  $\nu_i$ 's are defined as, see (13),

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

or equivalently as, see (17),

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

## 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 (10) or (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 1-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 Runge-Kutta (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.

### 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 3/8th rule; again, see Sections 3.1, 3.2, and 3.3 of [23]. 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 Chapter 7 of [23].

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 exemplify how the error on the trajectory is monitored, errors in the approximate exponents and/or the orthogonal factor are handled similarly (and see [23]). Monitoring of the error on the trajectory is done in the way explained in [26], with a mixed error control with respect to a scalar tolerance in the sup-norm.

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

- Estimate the worse error componentwise with respect to the desired tolerance:

$$\mathbf{err} = \max_{1 \leq i \leq m} \left( |x_{j+1}^{(i)} - \hat{x}_{j+1}^{(i)}| / [(1 + \max(|x_j^{(i)}|, |x_{j+1}^{(i)}|)) \text{TOLT}] \right). \quad (21)$$

- Now estimate

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

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

- Subject to the restrictions that  $h/5 \leq h_{\text{new}} \leq 5h$ , if  $\mathbf{err} \leq 1$  the step is successful and accepted, otherwise is 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 [23]. This way, the integration for the exponents conceptually proceed 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, or
- we need to approximate  $n \ll m$  Lyapunov exponents, or
- storage is at a premium and we cannot afford memory for a  $(m, m)$  matrix,

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

*What's in a name?* **LESNLS** refers to: **L**yapunov **E**xponent**S** **N**on**L**inear **S**mall. 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: **L**yapunov **E**xponent**S** **N**on**L**inear **L**arge, 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 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**.

### **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 gives the approximation

$$\begin{aligned} Y_{j+1} &= Q_j + h_j \sum_{l=1}^s b_l K_l, \quad K_l = f_x(x_{j,l})Y_{j,l}, \\ Y_{j,l} &= Q_j + h_j \sum_{p=1}^{l-1} a_{jp} K_p, \quad l = 1, 2, \dots, s. \end{aligned}$$

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 (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

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

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  $\eta$  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  $O(\sqrt{\text{EPS}})$ .

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

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

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.

## 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 (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** ([6]) 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 (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  $x_j, x_{j+1} \in \mathbb{R}^m$  and the vector field defined by  $f(x)$  [basically the same as that used in **Step 1**], determine coefficients  $a_j, b_j, a_{j+1}, b_{j+1} \in \mathbb{R}^m$  such that

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

the cubic Hermite interpolant such that  $\tilde{x}(t_j) = x_j$ ,  $\tilde{x}(t_{j+1}) = x_{j+1}$ ,  $\tilde{x}'(t_j) = f(x_j)$ , and  $\tilde{x}'(t_{j+1}) = f(x_{j+1})$ . 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 (1) are often wiped out in the context of approximation of 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.

## Examples

Several examples are available at the web sites

[www.math.gatech.edu/~dieci/software-les.html](http://www.math.gatech.edu/~dieci/software-les.html)

and

[www.math.ku.edu/~evanvleck/software-les.html](http://www.math.ku.edu/~evanvleck/software-les.html)

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 [27, 28] with values  $x_j$  of  $m$  points equally spaced around a circle of constant latitude:

$$\dot{x}_k = (x_{k+1} - x_{k-2})x_{k-1} - x_k + 8, \quad k = 0, 1, \dots, m-1, \pmod{m}. \quad (23)$$

In the results reported in [27, 28] 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 factor, 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 problems **LESNLL** to be approximately twice as fast as that for small problems **LESNLS** when computing all Lyapunov exponents. In addition, we obtained values of the Kaplan-Yorke dimension  $\dim_{KY} \approx 27.06$  and found that

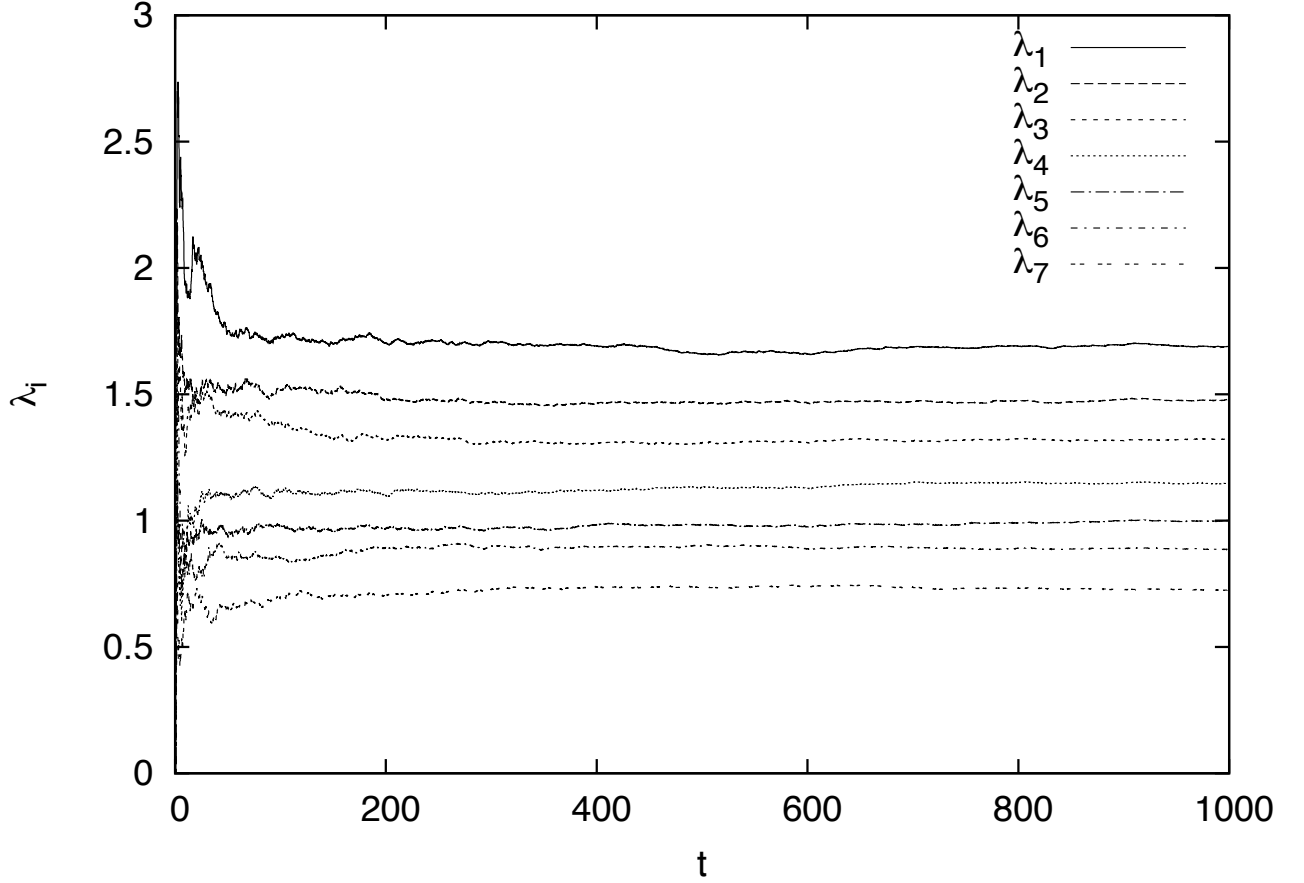


Figure 1: First seven approximate Lyapunov exponents versus time for the model of atmospheric variability (23).

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

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

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

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

(24) can be written as

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

This allows us to compare to the Lyapunov exponents reported in [11] and [13]. All computations were performed for  $\xi = 0.02991$ ,  $\vartheta = 133.73454$ , one of the parameter values considered in [11, 13, 39].

We consider a spatial discretization using standard Galerkin truncation with 16 or 32 modes (see [13] 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** [6] to approximate the solution of (1).

We report on numerical experiments to solve for the first four Lyapunov exponents over intervals of length 10 and 100. In Table 1-3,  $T$  denote the final time,  $M$  the number of modes,  $N$  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 [11, 13]. For  $M=16$  modes, the discrete and continuous QR approach 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.

## 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

$T$	$M$	$N$	TOL	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\sum_{j=1}^4 \lambda_j$	CPU
10	16	4	1.E-5	87.715079	0.10104269	-762.91721	-1139.7821	-1814.88	1
100	16	4	1.E-5	84.543528	-0.0036706406	-760.01825	-1136.9618	-1812.44	10.0
10	16	4	1.E-8	84.186532	0.0062248129	-759.54664	-1136.8065	-1812.16	2.8
100	16	4	1.E-8	83.917786	-0.0048723886	-759.47680	-1136.2823	-1811.85	28.2
10	32	4	1.E-5	85.075006	-0.28543797	-757.78116	-1128.0010	-1801.00	63.2

Table 1: The discrete QR method applied to the Kuramoto-Sivashinsky equation using the Dormand-Prince pair.

$T$	$M$	$N$	TOL	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\sum_{j=1}^4 \lambda_j$	CPU
10	16	4	1.E-5	80.787364	-0.0811046	-756.33708	-1136.1482	-1811.77	1.03
100	16	4	1.E-5	83.655799	-0.0002499180	-759.12671	-1136.8852	-1812.36	10.2
10	16	4	1.E-8	84.057142	-0.110254327	-759.33644	-1136.8114	-1812.20	2.9
100	16	4	1.E-8	83.982066	-0.0019134863	-759.46494	-1136.7057	-1812.19	28.4
10	32	4	1.E-5	84.961902	0.32904663	-758.65851	-1124.8598	-1798.23	72.9

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	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\sum_{j=1}^4 \lambda_j$	CPU
10	16	4	1.E-5	85.132041	0.1193929	-760.43735	-1141.4009	-1816.59	1.8
100	16	4	1.E-5	83.030609	-0.045702566	-758.47894	-1137.2718	-1812.67	18.5
10	16	4	1.E-8	83.555912	-0.07864597	-759.21634	-1137.0916	-1812.83	4.8
100	16	4	1.E-8	84.244061	0.0069730059	-759.69436	-1137.0974	-1812.54	48.8
10	32	4	1.E-5	88.068104	-0.16851648	-760.07970	-1129.7976	-1801.98	77.2

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

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.

## References

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

- [2] D. Arasteh, “Measures of Order in Dynamic Systems”, *J. Comput. Nonlinear Dynam.* **3** (2008), 031002.
- [3] O. Bendiksen, “Localization phenomena in structural dynamics”, *Chaos, Solitons & Fractals* **11-10** (2000), pp. 1621–1660.
- [4] G. Benettin, L. Galgani, A. Giorgilli and J.-M. Strelcyn, “Lyapunov Exponents for Smooth Dynamical Systems and for Hamiltonian Systems; A Method for Computing All of Them. Part 1: Theory ”, and “ ... Part 2: Numerical Applications ”, *Meccanica* **15** (1980), pp. 9–20, 21–30.
- [5] P. Blomgren, A. Palacios, B. Zhu, S. Daw, C. Finney, J. Halow and S. Pannala, “Bifurcation analysis of bubble dynamics in fluidized beds”, *CHAOS* **17** (2007), 013130.
- [6] P. Brown and A. Hindmarsh, “Matrix free methods for stiff systems of ODEs”, *SIAM J. Numer. Anal.* **23** (1986), pp. 610–638.
- [7] B.F. Bylov and N.A. Izobov, “Necessary and sufficient conditions for stability of characteristic exponents of a linear system”, *Differentsial’nye Uravneniya* **5** (1969), pp. 1794–1903.
- [8] M. P. Calvo, A. Iserles, and A. Zanna, “Numerical solution of isospectral flows”, *Math. Comp.* **66** (1997), pp. 1461–1486.
- [9] M.T. Chu, “On the continuous realization of iterative processes”, *SIAM Review* **30** (1988), pp. 375–387.

- [10] P. Constantin and C. Foias, “Global Lyapunov exponents, Kaplan-Yorke formulas and the dimension of the attractors for 2D Navier-Stokes equations”, *Comm. Pure Appl. Math.* **38** (1985), pp. 1–27.
- [11] F. Christiansen, P. Cvitanović, and V. Putkaradze, “Spatiotemporal chaos in terms of unstable recurrent patterns”, *Nonlinearity*, **10** (1997), pp. 55–70.
- [12] L. Dai, “Implementation of Periodicity Ratio in Analyzing Nonlinear Dynamic Systems: A Comparison With Lyapunov Exponent”, *J. Comput. Nonlinear Dynam.* **3** (2008), 011006.
- [13] L. Dieci, M. S. Jolly, M. S., R. Rosa, E. S. Van Vleck, “Error in approximation of Lyapunov exponents on inertial manifolds: the Kuramoto-Sivashinsky equation”, *Discrete Contin. Dyn. Syst. Ser. B* **9** (2008), pp. 555–580.
- [14] L. Dieci, R. D. Russell, and E. S. Van Vleck. On the computation of Lyapunov exponents for continuous dynamical systems. *SIAM J. Numer. Anal.*, 34:402–423, 1997.
- [15] L. Dieci and E. S. Van Vleck. Computation of a few Lyapunov exponents for continuous and discrete dynamical systems. *Applied Numerical Math.*, 17:275–291, 1995.
- [16] L. Dieci and E. S. Van Vleck, “Computation of orthonormal factors for fundamental solution matrices”, *Numer. Math.* **83** (1999), pp. 599–620.
- [17] L. Dieci and E. S. Van Vleck, “Lyapunov and other spectra: a survey”, In D. Estep and S. Tavener, editors, *Preservation of Stability under Discretization*. SIAM, Philadelphia, 2002.

- [18] L. Dieci, E. S. Van Vleck, “Lyapunov spectral intervals: theory and computation”, *SIAM J. Numer. Anal.* **40** (2002), pp. 516–542.
- [19] L. Dieci, E. S. Van Vleck, “On the error in computing Lyapunov exponents by QR methods”, *Numer. Math.* **101** (2005), pp. 619–642.
- [20] L. Dieci, E. S. Van Vleck, “Perturbation theory for approximation of Lyapunov exponents by QR methods”, *J. Dynam. Differential Equations* **18** (2006), pp. 815–840.
- [21] L. Dieci, E. S. Van Vleck, “Lyapunov and Sacker-Sell spectral intervals”, *J. Dynam. Differential Equations* **19** (2007), pp. 265–293.
- [22] L. Dieci, E. S. Van Vleck, “On the error in QR integration”, *SIAM J. Numer. Anal.* **46** (2008), pp. 1166–1189.
- [23] L. Dieci and E. S. Van Vleck. “**LESLIS** and **LESLIL**: Codes For Approximating Lyapunov Exponents of linear systems”, See: <http://www.math.gatech.edu/~dieci/software-les.html>
- [24] S.P. Diliberto, “On Systems of Ordinary Differential Equations,” in *Contributions to the Theory of Nonlinear Oscillations* (Ann. of Math. Studies 20), Princeton Univ. Press, Princeton (1950), pp. 1–38.
- [25] E. Elnashaie and J. Grace, “Complexity, bifurcation and chaos in natural and man-made lumped and distributed systems”, *Chemical Engineering Science* **62-13** (2007), pp. 3295–3325.

- [26] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*. Springer-Verlag, Berlin-Heidelberg, (1993) Second edition.
- [27] E. Lorenz, “Predictability. A problem partly solved”, in Proceedings on predictability held at ECMWF on 4-8 September 1995, pp. 1-18. [Available from ECMWF, Shinfield Park, Reading, Berkshire RG29AX, e.kooij@ecmwf.int] (1996), pp. 1–18.
- [28] E. Lorenz and K. Emmanuel, “Optimal sites for supplementary weather observations: Simulations with a small model”, *J. Atmos. Sci.* **55** (1998), pp. 399–414.
- [29] A. Lyapunov, “Problém général de la stabilité du mouvement”, *Int. J. Control* **53** (1992), pp. 531–773.
- [30] V. M. Millionshchikov. Structurally stable properties of linear systems of differential equations. *Differents. Uravneniya*, 5:1775–1784, 1969.
- [31] V. M. Millionshchikov. Systems with integral division are everywhere dense in the set of all linear systems of differential equations. *Differents. Uravneniya*, 5:1167–1170, 1969.
- [32] V. I. Oseledec, “A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems”, *Trans. Moscow Mathem. Society* **19** (1998), 197.
- [33] E. Ott, *Chaos in dynamical systems*. Cambridge University Press, New York, (1993).
- [34] K. J. Palmer, “The structurally stable systems on the half-line are those with exponential dichotomy,” *J. Diff. Eqn.* **33** (1979), pp. 16–25.
- [35] K. J. Palmer, “Exponential dichotomy, integral separation and diagonalizability of linear systems of ordinary differential equations,” *J. Diff. Eqn.* **43** (1982), pp. 184–203.

- [36] K. J. Palmer, “Exponential separation, exponential dichotomy and spectral theory for linear systems of ordinary differential equations,” *J. Diff. Eqn.* **46** (1982), pp. 324–345.
- [37] O. Perron, “Die Ordnungszahlen Linearer Differentialgleichungssystemen,” *Math. Zeits.* **31** (1930), pp. 748–766.
- [38] D. Ruelle, *Chaotic evolution and strange attractors*. Cambridge University Press, Cambridge, (1989).
- [39] Y. S. Smyrlis and D. T. Papageorgiou, *Predicting chaos for infinite-dimensional dynamical systems: the Kuramoto-Sivashinsky equation, a case study*, Proc. Nat. Acad. Sci. U.S.A., **88** (1991), pp. 11129–11132.
- [40] E. S. Van Vleck, “On the Error in the Product QR Decomposition”, *to appear in SIAM J. Matr. Anal. Applic.*, (2010).
- [41] A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano, “Determining Lyapunov Exponents from a Time Series, ” *Physica D* **16** (1985), pp. 285–317.
- [42] M. Xu and Z. Gao, “Nonlinear Analysis of Road Traffic Flows in Discrete Dynamical System”, *J. Comput. Nonlinear Dynam.* **3** (2008), 021206.