Università degli studi di Udine

Pseudospectral reduction to compute Lyapunov exponents of delay differential equations

Original

Availability:
This version is available http://hdl.handle.net/11390/1124116 since 2021-03-14T16:21:18Z

Publisher:

Published
DOI:10.3934/dcdsb.2018092

Terms of use:
The institutional repository of the University of Udine (http://air.uniud.it) is provided by ARIC services. The aim is to enable open access to all the world.

Publisher copyright

(Article begins on next page)

06 May 2024

DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES B

doi:10.3934/dcdsb.2018091

PSEUDOSPECTRAL REDUCTION TO COMPUTE LYAPUNOV EXPONENTS OF DELAY DIFFERENTIAL EQUATIONS

Dimitri Breda*

Department of Mathematics, Computer Science and Physics – University of Udine
via delle scienze 206, 33100 Udine, Italy

Sara Della Schiava

Department of Mathematics, Computer Science and Physics – University of Udine
via delle scienze 206, 33100 Udine, Italy

(Communicated by the associate editor name)

Abstract. A recent pseudospectral collocation is used to reduce a nonlinear delay differential equation to a system of ordinary differential equations. Standard methods are then applied to compute Lyapunov exponents. The validity of this simple approach is shown experimentally. Matlab codes are also included.

1. **Introduction.** We consider the question of computing Lyapunov exponents of Delay Differential Equations (DDEs). Lyapunov exponents are defined through limits in the infinite time [1]. DDEs give rise to infinite-dimensional dynamical systems [16, 23, 29]. These two aspects make the problem particularly hard.

The literature is not lacking of material on the subject. On the one hand, most of the relevant papers are of experimental flavor: more focused on applications and results, less on the often coarse techniques they apply, and always restricted to specific, yet important examples (see, e.g., [26, 39, 40]). On the other hand, extremely few papers address the question from a more numerical or computational point of view, but the methods proposed are usually difficult to replicate by non-experts and relevant algorithms and codes are rarely available or user-friendly (see, e.g., [14, 41]).

The present work aims at filling the gap between the need for off-the-shelf routines for those interested in the applications and the complexity of reliable, efficient and rigorous methods.

The approach we propose is based on reducing the DDE to a system of Ordinary Differential Equations (ODEs), a class for which definitely more is known and available about both theory and computation of Lyapunov exponents (start, e.g., from

2010 *Mathematics Subject Classification.* Primary: 37M25, 65L03, 65L07.

Key words and phrases. Lyapunov exponents, delay differential equations, pseudospectral methods.

The first author is supported by INdAM GNCS projects "Analisi numerica di sistemi dinamici infinito-dimensionali e non regolari" (2015) and "Analisi numerica di certi tipi non classici di equazioni di evoluzione" (2016).

* Corresponding author.

[20]). The reduction is based on the pseudospectral discretization recently introduced in [8]. As described in Section 2, the resulting system of ODEs is quite simple to obtain: a single equation depends on the original DDE, basically retaining the same right-hand side; the rest is linear and independent of the DDE. To compute the exponents of ODEs we choose, among others, the discrete QR technique. It is easy to understand, also with respect to the underlying concept of exponential divergence of infinitesimally closed orbits. At the same time it is not difficult to code efficiently, taking inspiration from [17].

Despite the simplicity of the combined methodology, we also furnish relevant Matlab codes in Appendix A. To the best of our knowledge, they are the first public and available codes for computing Lypaunov exponents of DDEs. We explain and use them in Section 4, where we illustrate some tests on three equations, quite simple for deliberate choice. This way we keep both presentation and application at a basic level, focusing more on the validity of the method and the use of the codes. Possible technicalities due to generalizations could distract the reader from appreciating the straightforwardness of the proposed approach (however, suitable references to treat them are given).

We postpone to the following sections, where better indicated, additional comments and detailed references on the above and other interesting aspects about pseudospectral reduction and Lyapunov exponents. Some are related to either convergence and approximation, others to more theoretical issues such as, e.g., the proper state space to deal with. We just remark from [8] that the reduction to ODEs can be extended to more general classes of functional equations. Therefore, in principle, the methodology we propose has a potentially wide range of applications.

2. **From DDEs to ODEs.** Let us summarize from [8] the pseudospectral discretization to reduce DDEs to ODEs. We restrict to the scalar case, extension to systems being straightforward.

Consider the nonlinear autonomous DDE

$$x'(t) = f(x_t), \tag{1}$$

where, for $\tau > 0$ and $X := C([-\tau, 0]; \mathbb{R})$, $f : X \to \mathbb{R}$ is smooth and $x_t \in X$ denotes $x_t(\theta) := x(t + \theta)$, $\theta \in [-\tau, 0]$. Given $\varphi \in X$, the initial value problem

$$\begin{cases} x'(t) = f(x_t), & t \geq 0, \\ x(\theta) = \varphi(\theta), & \theta \in [-\tau, 0], \end{cases} \tag{2}$$

is well-posed. The associated semigroup $\{T(t)\}_{t\geq 0}$ of nonlinear solution operators $T(t) : X \to X$ defined by

$$T(t)(\varphi) = x_t$$

is strongly continuous and has generator $\mathcal{A} : \mathcal{D}(\mathcal{A}) \subseteq X \to X$ with action

$$\mathcal{A}(\varphi) = \varphi'$$

and domain

$$\mathcal{D}(\mathcal{A}) = \{\varphi \in X \ : \ \varphi' \in X \ \text{ and } \ \varphi'(0) = f(\varphi)\}. \tag{3}$$

Eventually, standard results on semigroups provide the equivalent formulation of (2) as the abstract Cauchy problem

$$\begin{cases} u'(t) = \mathcal{A}(u(t)), & t \geq 0, \\ u(0) = \varphi, & \varphi \in \mathcal{D}(\mathcal{A}), \end{cases} \tag{4}$$

for $u(t) = x_t$, see, e.g., [25, 36].

The basic idea behind the pseudospectral reduction is to collocate (4) at given nodes w.r.t. the variable $\theta \in [-\tau, 0]$. For $M \geq 1$, let these nodes be the Chebyshev extrema, say $-\tau = \theta_M < \cdots < \theta_1 < \theta_0 = 0$ with

$$\theta_i = \frac{\tau}{2}\left(\cos\frac{i\pi}{M} - 1\right), \quad i = 0, 1, \ldots, M.$$

For $i = 0, 1, \ldots, M$, $t \geq 0$ and $u(t) \in \mathcal{D}(\mathcal{A})$, denote by $u_i(t)$ the approximation of $u(t)(\theta_i)$ obtained as follows. Let $U(t)$ be the Lagrange interpolation polynomial

$$U(t)(\theta) = \sum_{j=0}^{M} \ell_j(\theta) u_j(t), \quad \theta \in [-\tau, 0],$$

where

$$\ell_j(\theta) = \prod_{\substack{k=0 \\ k \neq j}}^{M} \frac{\theta - \theta_k}{\theta_j - \theta_k}, \quad j = 0, 1, \ldots, M,$$

constitute the Lagrange basis. Now require that $U(t)(\theta)$ satisfies (4) at all the nodes but for $\theta_0 = 0$, where the boundary condition in (3) is instead forced. Then

$$u_i'(t) = U'(t)(\theta_i) = \mathcal{A}(U(t))(\theta_i), \quad i = 0, 1, \ldots, M,$$

and, eventually, we get the system of $M + 1$ ODEs

$$\begin{cases} u_0'(t) = f(U(t)) \\ u_i'(t) = \displaystyle\sum_{i=0}^{M} d_{i,j} u_j(t), \quad i = 1, \ldots, M, \end{cases} \tag{5}$$

with initial conditions

$$u_i(0) = \varphi(\theta_i), \quad i = 0, 1, \ldots, M.$$

Above, $d_{i,j} := \ell_j'(\theta_i)$ are the entries of the Chebyshev differentiation matrix, see, e.g., [42] for their explicit expression.

Notice that only the first ODE in (5) depends on the original DDE, preserving the nonlinearity. All the others are linear due to the linearity of both interpolation and differentiation. They are also independent of the DDE if one scales time to get $\tau = 1$. This makes the reduction simple, and particularly attractive from the point of view of implementation. For a detailed discussion on this advantage and other aspects see [8]. Again, we just remark that, in principle, the same approach can be extended to reduce also other general classes of functional equations, e.g., neutral and retarded-advanced as well as renewal and coupled renewal-retarded, [8, 9, 11].

Now, let us underline that when f is linear and autonomous, so is (5). Then, the matrix characterizing the right-hand side corresponds to the discretization of

the infinitesimal generator first proposed in [10] to address the local stability of equilibria:

$$\mathcal{A}_M := \begin{pmatrix} f(\ell_0) & f(\ell_1) & \cdots & f(\ell_M) \\ d_{1,0} & d_{1,1} & \cdots & d_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,0} & d_{M,1} & \cdots & d_{M,M} \end{pmatrix}.$$

See [13] for an updated exposition. Otherwise, when f is nonlinear, it is immediate to verify that linearization around a given solution x^* and pseudospectral discretization commute [8, Theorem 2.5]. The result is the linear, in general nonautonomous system of ODEs

$$\overline{u}'(t) = \mathcal{A}_M(t)\overline{u}(t)$$

for $\overline{u} := (u_0, u_1, \ldots, u_M)^T$ and

$$\mathcal{A}_M(t) := \begin{pmatrix} Df(x_t^*)\ell_0 & Df(x_t^*)\ell_1 & \cdots & Df(x_t^*)\ell_M \\ d_{1,0} & d_{1,1} & \cdots & d_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,0} & d_{M,1} & \cdots & d_{M,M} \end{pmatrix}, \tag{6}$$

where $Df(x)$ is the Frechét derivative of f at $x \in X$. Time-dependence and, again, the original DDE affect only the first row.

Let us conclude this section with a glimpse of the literature. The pseudospectral technique above recalled can be recast as the collocation in θ of the hyperbolic Partial Differential Equation (PDE)

$$\begin{cases} \dfrac{\partial v}{\partial t}(t,\theta) = \dfrac{\partial v}{\partial \theta}(t,\theta), \quad t \geq 0, \quad \theta \in [-\tau, 0], \\ \dfrac{\partial v}{\partial \theta}(t,0) = f(v(t,\cdot)), \qquad\qquad t \geq 0, \\ v(0,\theta) = \varphi(\theta), \qquad\qquad \theta \in [-\tau, 0]. \end{cases} \tag{7}$$

Spectral and pseudospectral methods for PDEs have been extensively treated in, e.g., [27, 28]. (7) is equivalent to both (2) and (4) through $v(t,\theta) = u(t)(\theta) = x_t(\theta)$, see, e.g., [23, 29]. This equivalence was originally exploited in [3, 35] to reduce DDEs with a single constant delay to systems of ODEs by Runge-Kutta methods, for the purpose of numerical time-integration. Other reductions, similar in scopes, can be traced back to [30] about the use of spline functions and [2, 31] concerning splines again and Galerkin-type projections, respectively. Except for the last two, all these methods rely on the classic state space X of continuous functions; [2, 31], instead, move to an Hilbert space for the numerical aspects, see also the interesting comments in [31, p.148]. More on these different settings is left to Section 3.

3. **Lyapunov exponents and the discrete QR method.** There is an extensive literature on the computation of Lyapunov exponents of DDEs, mostly of an experimental flavor. We refer the reader to the introduction of [14] as a starting point, to [7] for a minimal theoretical basis and to [26] for maybe the most considered paper on the subject. Avoiding repetitions, we could summarize by saying that, until very recently, [14] represented to the best of our knowledge the only method with a rigorous and complete proof of convergence. This is no longer true potentially, since the authors of [15] propose a modern reduction of DDEs to ODEs by Galerkin-type projections via so-called Koornvinder orthogonal polynomials [32]. They show the

reliability of the technique in approximating strange attractors and related statistical features. Therefore, we believe that computing the Lyapunov exponents by applying standard tools for ODEs to their reduced system is as reasonable as the method we presently investigate, although they do not mention such an approach. The main difference is that we keep on working on the classic Banach state space of continuous functions – where DDEs are naturally posed – instead of the Hilbert setting required in [15] to legitimate orthogonal projections. The same setting is required even in [14], indeed, and more comments are given towards the end of the section. Now we prefer instead to briefly account for the available methods for computing Lyapunov exponents of ODEs, since there fall back either [15] or the collocation presented in Section 2.

If the literature on Lyapunov exponents of DDEs is ample, just guess about ODEs. Not to get lost, beyond the original thesis of Lyapunov [33], the most complete, modern reference from a theoretical point of view is maybe [1]. One usually considers linear and nonautonomous systems, resorting to linearization around a reference trajectory in the attractor in the nonlinear case. Lyapunov exponents can be recovered from a time limit as $t \to +\infty$ of the diagonal elements of the matrix function at the right-hand side, after the system has been reduced to triangular form through orthogonalization. From a computational point of view, due to stability considerations, this orthogonalization has to be performed along a sequence of not too-large time steps. In so doing one prevents the alignment along the direction of highest growth (i.e., largest exponent). This procedure is at the base of the most used class of computational techniques, namely QR methods, first discussed in the pioneering works [5, 6]. Here we summarize from [17] the discrete version of the QR method for computing the Lyapunov exponents of

$$x'(t) = A(t)x(t), \tag{8}$$

where $A : [0, +\infty) \to \mathbb{R}^{n\times n}$ is continuous and bounded. In the following a QR factorization of a nonsingular matrix is intended as the unique one with positive diagonal elements. For a starting reference on theory and computation of Lyapunov exponents of ODEs see however [20].

Prior to go on we remark that the simple key of this work consists in replacing A in (8) with $\mathcal{A}_M$ given in (6).

Let $X(t)$ be the fundamental matrix solution of (8) exiting from a given nonsingular matrix X_0 prescribed at time 0 without loss of generality. Choose any strictly increasing sequence of time instants $\{t_k\}$ with $t_0 = 0$ and construct the iterative QR factorization

$$X(t_k) = Q_k R_k \tag{9}$$

as follows. Starting from $X_0 = Q_0 R_0$, at each step $j = 1, \dots, k$ solve

$$\begin{cases} \Psi'(t, t_{j-1}) = A(t)\Psi(t, t_{j-1}), \quad t \in [t_{j-1}, t_j], \\ \Psi(t_{j-1}, t_{j-1}) = Q_{j-1} \end{cases} \tag{10}$$

and factorize the solution at t_j as

$$\Psi(t_j, t_{j-1}) = Q_j R_{j,j-1}. \tag{11}$$

If $S(t,s) = X(t)X(s)^{-1}$ is the state transition matrix, then

$$\begin{aligned} X(t_k) &= S(t_k,t_{k-1})\cdots S(t_2,t_1)S(t_1,t_0)Q_0R_0 \\ &= S(t_k,t_{k-1})\cdots S(t_2,t_1)\Psi(t_1,t_0)R_0 \\ &= S(t_k,t_{k-1})\cdots S(t_2,t_1)Q_1R_{1,0}R_0 \\ &= S(t_k,t_{k-1})\cdots \Psi(t_2,t_1)R_{1,0}R_0 \\ &= S(t_k,t_{k-1})\cdots Q_2R_{2,1}R_{1,0}R_0 \\ &\cdots \\ &= Q_kR_{k,k-1}\cdots R_{1,0}R_0. \end{aligned} \tag{12}$$

Uniqueness of the QR factorization and (9) give

$$R_k = \left(\prod_{j=1}^{k} R_{j,j-1}\right) R_0.$$

Eventually, (upper) Lyapunov exponents are recovered as

$$\lambda_i = \limsup_{k\to\infty} \frac{1}{t_k}\sum_{j=1}^{k} \ln[R_{j,j-1}]_{i,i}, \quad i = 1,\dots,n, \tag{13}$$

where $[R_{j,j-1}]_{i,i}$ stands for the i-th diagonal entry of the j-th triangular factor $R_{j,j-1}$. Lower exponents come either as $\liminf$ or as upper exponents of the adjoint system. Summarizing, each step requires the solution of the n initial value problems (10) and the QR factorization (11).

For comparison, let us comment on the method in [14], which is based on lifting (12) to infinite dimension. Of course, for a QR factorization to make sense in infinite dimension, the state space must be changed from the Banach space X of continuous functions with uniform norm to the Hilbert space $H := \mathbb{R}\times L^2([-\tau,0);\mathbb{R})$ with metric induced by the proper inner product. See [7, 16] for a discussion on this setting and the former for the definition of the relevant Lyapunov exponents. The work [14] proposes to discretize the evolution family $S(t,s)$ in H via generalized Fourier projection, going back to finite dimension with no need to solve the initial value problems (10), (11) being replaced by

$$\tilde{S}(t_j,t_{j-1})Q_{j-1} = Q_jR_{j,j-1}$$

for $\tilde{S}$ approximating S. A rigorous convergence analysis is also provided there. The key fact is that one does not usually compute state transition matrices for ODEs (but rather solve them), while for DDEs an efficient way to approximate evolution families was already available in [12], for the sake of stability of periodic problems. The approach in [14] was indeed inspired from the latter. The pseudospectral collocation technique in [12] instead, is not appropriate in the context of Lyapunov exponents, since based on X where orthogonal factorization is nonsense. This is why here we take the other way around, i.e., we first reduce to finite dimension, allowing to change from uniform to euclidean metric with proper scalar product. Summarizing, one could choose between two strategies: lifting finite-dimensional methods to solve infinite-dimensional problems or reduce infinite-dimensional problems and solve by finite-dimensional methods. While [14] can be ascribed to the first class, the method proposed here belongs to the second. As such, it represents the natural

attempt to address the study of chaotic dynamics by following the originating motivation of [8], there focused on equilibria and their bifurcation, and later applied to periodic orbits of renewal equations in [9]. We conclude by observing that most of the computational techniques proposed in the literature for computing exponents of DDEs belong to the second category above mentioned, see, e.g., [26, 40].

4. **Implementation and results.** Three Matlab codes are listed in Appendix A:
   - `dqr`, implementing the discrete QR method for computing Lyapunov exponents of linear nonautonomus ODEs like (8), as described in Section 3 and based on [17];
   - `solveDDE_MG`, implementing the time-integration of nonlinear DDEs like (1), based on [38];
   - `lrhs_MG`, implementing the construction of the linear(ized) nonautonomous matrix (6), based on Section 2 and [8].

The first code is a Matlab version of the Fortran 77 routine LESLIS freely available at [22]. As the original one, it solves the initial value problems (10) by using the celebrated DOPRI54 embedded Runge-Kutta pair [24] to keep the error on the exponents below a given tolerance by automatic selection of the stepsizes $t_j - t_{j-1}$. The QR factorization is obtained via Matlab built-in `qr`, just notice that lines 39 and 57 are a rude, yet trivial way to realize the required uniqueness. For more on computational aspects of orthogonal factors in this context see [21]. The limit (13) is truncated at a given large time specified in input, say T. The latter two codes are written for the specific example of the Mackey-Glass equation, described below. The first one simply resorts to Matlab `dde23`. The second one follows Section 2 and uses the auxiliary routine `difmat` taken from [42] to construct the Chebyshev differentiation matrix.

We believe that the first code is useful for those accustomed to the Matlab environment rather than to less high-level frameworks. The other two may be of help to those who wish to repeat the following numerical experiments or test their own problems. It is enough indeed to alter the auxiliary function `rhs` in `solveDDE_MG` and lines $25 - 26$ in `lrhs_MG`, beyond straightforward modifications of parameters input. Results are obtained by computing a reference trajectory x^* for a given choice of parameters with `solveDDE_MG`, constructing the matrix (6) with `lrhs_MG` and eventually approximating the Laypunov exponents with `dqr`. An instance of explicit call is given next.

The remaining of the section is devoted to run the proposed methodology on three test DDEs, two scalar and a system, all with a single delay. We have no reason to doubt that the method could work properly also for more general equations and systems with either multiple and distributed delays. For extensions of the pseudospectral reduction in these directions it is enough to consider the linear case, for which we refer to [13], also for an efficient piecewise approach. Moreover, we have already mentioned in the Introduction and in Section 2 the possibility to extend the reduction to other classes of functional equations. In this sense, a first successful attempt to compute Lyapunov exponents of renewal equations is reported in [9].

Here we choose to sacrifice extensions and generality in favor of simplicity. We focus on demonstrating the validity of the approach as a manageable strategy to

obtain Lyapunov exponents, unveiling some of its features experimentally. A rigorous proof of convergence of the computed exponents to the exact ones is not available, and out of the reach of the present work. In fact, even the convergence of the approximating system of ODEs to the original DDE is still a subject of current research. More comments can be found in [8], especially w.r.t the prominent role of the Chebyshev differentiation matrix D_M and the stability of its time exponential $\exp(D_M t)$ as $M \to \infty$ (see also [28] in this respect).

The first example is the linear autonomous DDE

$$x'(t) = -x(t) + 2ex(t-1). \tag{14}$$

Taken from [14], it is invoked for the lack of DDEs with known and exact exponents. For (14), instead, the exponents coincide with the real part of the characteristic roots. The recent literature abounds of efficient and reliable methods to approximate the latter. Moreover, it is not difficult to show that the rightmost root of (14) is 1. In Figure 1 (left) we plot in $\mathbb{C}$ the first five characteristic roots ($\times$, with nonnegative imaginary part) computed to machine precision with the code `eigAM` accompanying [13], together with the corresponding Lyapunov exponents (along the real axis) approximated with the proposed technique ($\bullet$) and with the method in [14] ($\circ$), both for $M = 20$. The truncation time was set to $T = 10^5$, while `dqr` was modified to use the exact solution of (10) through the matrix exponential in order to save useless computations (recall that (14) is autonomous). In Figure 1 (right) we plot the absolute error of the largest exponent w.r.t. the leading real part 1 (leftmost curves, solid $\bullet$ for the current method, dashed $\circ$ for the method in [14]). We plot also the absolute errors of the remaining exponents w.r.t. the real part of the relevant characteristic roots, assumed as reference values. The spectrally accurate behavior (see, e.g., [42]) typical of pseudospectral methods as applied to smooth functions emerges evident. Notice, moreover, how the error curves move to the right as the exponents decrease, signature that the error constants increase in general proportionally to $|\lambda|$ in both cases. Eventually, the method in [14] seems to give lower errors in general (made exception for the leading exponent at larger values of M). This behavior is not surprising: the method in [14] is developed specifically for DDEs, while the current one is more general w.r.t. the class of equations it can tackle. This major generality (and the subsequent simplicity) seems thus to trade the accuracy off.

A lower barrier for the error at about 10^{-5} is also evident from Figure 1 (right). It is essentially due to the truncation of the limit (13) at a finite time T. Some insight is given by the results illustrated in Figure 2, where the error on the largest exponent is plotted against T for varying $M = 10, 15, 20$ (solid $\bullet$). Similar data for $M = 20$ from Figure 3 of [14] are included for a direct comparison (dashed $\circ$), confirming the expected linear decay as discussed in the latter. For a deeper analysis of the consequences of this truncation we refer also to [18, 19].

The second example is the celebrated Mackey-Glass equation [34]:

$$x'(t) = \frac{ax(t-\tau)}{1-[x(t-\tau)]^c} - bx(t). \tag{15}$$

In Table 1 we collect the first six exponents for $a = 0.2$, $b = 0.1$, $c = 10$ and $\tau = 50$ computed with $M = 20$, comparing with the values from [14, 39]. Truncation time and tolerance in `dqr` were set to 10^5 and 10^{-6}, respectively. The accordance of the

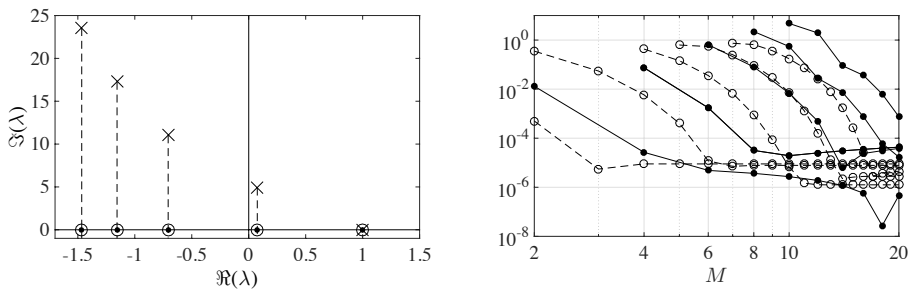

FIGURE 1. First five rightmost characteristic roots ($\times$) of (14) and first five dominant Lyapunov exponents computed with the current method ($\bullet$) and with the method in [14] ($\circ$), both for $M = 20$ and $T = 10^5$ (left); relevant absolute errors for increasing M (right): current method (solid $\bullet$) and method in [14] (dashed $\circ$).

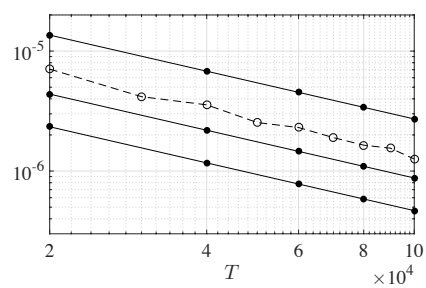

FIGURE 2. Absolute error with respect to 1 of the largest exponent of (14) plotted against the final truncation time T, computed with the current method for varying $M = 10, 15, 20$ (solid $\bullet$, top-to-bottom) and with the method in [14] (dashed $\circ$) for $M = 20$.

results is limited to few digits, but this is not surprising in the context of Lyapunov exponents, given the role of the truncation of the limit already encountered above.

5.85×10^{-3}	5.76×10^{-3}	5.83×10^{-3}
3.29×10^{-3}	3.02×10^{-3}	3.15×10^{-3}
0.53×10^{-3}	0.65×10^{-3}	0.01×10^{-3}
-0.92×10^{-3}	-0.85×10^{-3}	-0.29×10^{-3}
-5.17×10^{-3}	-4.78×10^{-3}	-5.08×10^{-3}
-9.56×10^{-3}	-9.85×10^{-3}	-9.78×10^{-3}

TABLE 1. First six exponents of (15) for $a = 0.2$, $b = 0.1$, $c = 10$ and $\tau = 50$ computed with $M = 20$ (first column), from [14] (second column) and from [39] (third column); the reference solution corresponds to the initial function of constant value $\varphi \equiv 2$ in (2).

It is also interesting to validate the approach by plotting some attractors, see Figure 3 and confront with Figure 2 in [26]. The plane of projection has coordinates

$x(t-\tau)$ and $x(t)$, respectively approximated by $u_M(t)$ and $u_0(t)$ obtained by solving (5) with Matlab `ode23s` for $M = 9$: notice how 10 ODEs are sufficient to get a rather accurate representation. In this sense, the pseudospectral reduction could be a valid alternative to more accomplished methods for initial value problems for DDEs [4].

To conclude with the Mackey-Glass equation, an example call to the Matlab codes follows, w.r.t. the data in Table 1:

```
>> sol=solveDDE_MG(.2,.1,10,50,2,1e5);
>> [lambda,t]=dqr(@(t)lrhs_MG(t,sol,.2,.1,10,50,20),1e5,1e-6);
```

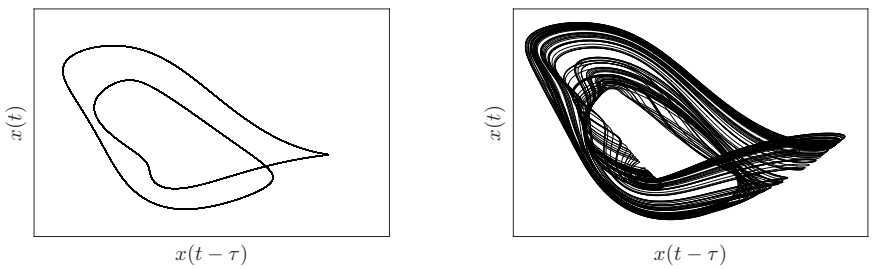

FIGURE 3. Projection of the attractor of (15) for $a = 0.2$, $b = 0.1$, $c = 10$ and $\tau = 14$ (left), $\tau = 17$ (right).

The literature on Lypaunov exponents for DDEs is not much rich on quantitative results to compare with. Nevertheless, we include a last example to test the behavior of the pseudospectral reduction on systems of DDEs. In particular, we consider from [14] the two Rössler oscillators symmetrically coupled

$$\begin{cases} x_1'(t) &= -x_2(t) - x_3(t), \\ x_2'(t) &= x_1(t) + ax_2(t) + \epsilon[x_5(t-\tau) - x_2(t)], \\ x_3'(t) &= b + x_3(t)[x_1(t) - c], \\ x_4'(t) &= -x_5(t) - x_6(t), \\ x_5'(t) &= x_4(t) + ax_5(t) + \epsilon[x_2(t-\tau) - x_5(t)], \\ x_6'(t) &= b + x_6(t)[x_4(t) - c], \end{cases} \tag{16}$$

taken originally from [37]. In Table 2 we compare the first three exponents of (16) with $a = b = 0.1$, $c = 14$ and $\epsilon = 0.5$ and varying coupling delay τ, computed (on purpose) with a quite rough approximation obtained with the proposed method for $M = 5$, $T = 10^3$ and the usual tolerance 10^{-6} in `dqr`. Notice that the dimension of the approximating system of ODEs is $n(M + 1) = 36$, hence already not properly small. The reference trajectory x^* is computed by `dde23` with starting time 0 and initial function φ in (2) of constant value a (pseudo)random vector in $\mathbb{R}^6$. The results are compared with those in Figure 6 of [14], computed with the method proposed therein with $M = 20$ and $T = 10^4$. The latter is basically much more accurate, but the size of the underlying matrix discretization is $n(M + 1) = 126$, almost 4 times the one of the current method. The correspondence is nevertheless good, since signs and magnitudes of the exponents are preserved. The analogous of

Figure 6 in [14] obtained with the current method would be almost indistinguishable at eye, or within the same range of variation of the original one in [37].

-2.20×10^{-2}	-6.74×10^{-2}	-1.51×10^{-2}
-2.21×10^{-2}	-6.77×10^{-2}	-1.49×10^{-2}
-3.65×10^{-1}	-1.19×10^{-1}	-1.21×10^{-1}
-2.28×10^{-2}	-6.81×10^{-2}	-1.55×10^{-2}
-2.30×10^{-2}	-6.89×10^{-2}	-1.57×10^{-2}
-3.59×10^{-1}	-1.11×10^{-1}	-1.14×10^{-1}

TABLE 2. First three exponents of (16) for $a = b = 0.1$, $c = 14$ and $\epsilon = 0.5$ and varying coupling delay $\tau = 1$ (first column), 1.5 (second column) and 2 (third column), computed with the current method for $M = 5$ and $T = 10^3$ (first three rows) and with the method in [14] for $M = 20$ and $T = 10^4$ (second three rows); the reference solution of (2) corresponds to the initial function of constant value a (pseudo)random vector in $\mathbb{R}^6$.

As a general conclusion, especially from the results on the nonlinear examples, we remark that the proposed method can represent a valid alternative, particularly useful for non-experts necessitating practical computations. Let us notice, indeed, that the experimental results available in the literature of Lyapunov exponents for DDEs are usually furnished with limited precision, which seems enough for most applications. On the other hand, codes implementing most methods proposed in the literature are practically unavailable. Then the pseudospectral reduction might result more advantageous than these techniques, in that it is easily implementable, adaptable to other classes of functional equations and rather reliable (at least experimentally), although we are aware that a rigorous convergence is not proved yet, as already commented.

REFERENCES

[1] L. Y. Adrianova, *Introduction to linear systems of differential equations*, no. 146 in Transl. Math. Monographs, AMS, Providence, 1995.

[2] H. T. Banks and F. Kappel, Spline approximations for functional differential equations, *J. Diff. Equations*, **34** (1979), 496–522.

[3] A. Bellen and S. Maset, Numerical solution of constant coefficient linear delay differential equations as abstract cauchy problems, *Numer. Math.*, **84** (2000), 351–374.

[4] A. Bellen and M. Zennaro, *Numerical methods for delay differential equations*, Numerical Mathemathics and Scientifing Computing series, Oxford University Press, 2003.

[5] 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, *Meccanica*, **15** (1980), 9–20.

[6] 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 2: Numerical applications, *Meccanica*, **15** (1980), 21–30.

[7] D. Breda, Nonautonomous delay differential equations in Hilbert spaces and Lyapunov exponents, *Diff. Int. Equations*, **23** (2010), 935–956.

[8] D. Breda, O. Diekmann, M. Gyllenberg, F. Scarabel and R. Vermiglio, Pseudospectral discretization of nonlinear delay equations: new prospects for numerical bifurcation analysis, *SIAM J. Appl. Dyn. Sys.*, **15** (2016), 1–23.

[9] D. Breda, O. Diekmann, D. Liessi and F. Scarabel, Numerical bifurcation analysis of a class of nonlinear renewal equations, *Electron. J. Qual. Theory Differ. Equ.*, **65** (2016), 1–24.

[10] D. Breda, S. Maset and R. Vermiglio, Pseudospectral differencing methods for characteristic roots of delay differential equations, *SIAM J. Sci. Comput.*, **27** (2005), 482–495.

[11] D. Breda, S. Maset and R. Vermiglio, Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions, *Appl. Numer. Math.*, **56** (2006), 318–331.

[12] D. Breda, S. Maset and R. Vermiglio, Approximation of eigenvalues of evolution operators for linear retarded functional differential equations, *SIAM J. Numer. Anal.*, **50** (2012), 1456–1483.

[13] D. Breda, S. Maset and R. Vermiglio, *Stability of linear delay differential equations – A numerical approach with MATLAB*, Springer Briefs in Control, Automation and Robotics, Springer, New York, 2015.

[14] D. Breda and E. S. Van Vleck, Approximating Lyapunov exponents and Sacker-Sell spectrum for retarded functional differential equations, *Numer. Math.*, **126** (2014), 225–257.

[15] M. D. Chekroun, M. Ghil, H. Liu and S. Wang, Low-dimensional Galerkin approximations of nonlinear delay differential euqations, *Discrete Contin. Dyn. S.*, **36** (2016), 4133–4177.

[16] F. Curtain and H. Zwart, *An Introduction to Infinite-Dimensional Linear Systems Theory*, vol. 21 of Texts in Applied Mathematics, Springer-Verlag, New York, 1995.

[17] L. Dieci, M. S. Jolly and E. S. Van Vleck, Numerical techniques for approximating Lyapunov exponents and their implementation, *J. Comput. Nonlinear Dynam.*, **6** (2010), 011003–1–7.

[18] 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** (1997), 402–423.

[19] L. Dieci and E. S. Van Vleck, Computation of few Lyapunov exponents for continuous and discrete dynamical systems, *Appl. Numer. Math.*, **17** (1995), 275–291.

[20] L. Dieci and E. S. Van Vleck, Lyapunov spectral intervals: theory and computation, *SIAM J. Numer. Anal.*, **40** (2002), 516–542.

[21] L. Dieci and E. S. Van Vleck, Orthonormal integrators based on Householder and Givens transformations, *Future Gener. Comp. Sy.*, **19** (2003), 363–373.

[22] L. Dieci and E. S. Van Vleck, LESLIS and LESLIL: codes for approximating Lyapunov exponents of linear systems, 2004, http://www.math.gatech.edu/~dieci/software-les.html.

[23] O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel and H. O. Walther, *Delay Equations - Functional, Complex and Nonlinear Analysis*, no. 110 in Applied Mathematical Sciences, Springer Verlag, New York, 1995.

[24] J. R. Dormand and P. J. Prince, A family of embedded Runge-Kutta formulae, *J. Comput. Appl. Math.*, **6** (1980), 19–26.

[25] K. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, no. 194 in Graduate Texts in Mathematics, Springer-Verlag, New York, 1999.

[26] D. Farmer, Chaotic attractors of an infinite-dimensional dynamical system, *Physica D*, **4** (1982), 605–617.

[27] D. Gottlieb, The stability of pseudospectral-Chebyshev methods, *Math. Comp.*, **36** (1981), 107–118.

[28] D. Gottlieb, M. Y. Hussaini and S. A. Orszag, Theory and applications of spectral methods, in *Spectral methods for partial differential equations*, SIAM, Philadelphia, Hampton, Va., 1984, 1–54.

[29] J. K. Hale and S. M. Verduyn Lunel, *Introduction to functional differential equations*, 2nd edition, no. 99 in Applied Mathematical Sciences, Springer Verlag, New York, 1993.

[30] K. Ito and F. Kappel, A uniformly differentiable approximation scheme for delay systems using splines, *Appl. Math. Opt.*, **23** (1991), 217–262.

[31] F. Kappel, *Semigroups and delay equations*, no. 152 (Trieste, 1984) in Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, 1986.

[32] T. H. Koornvinder, Orthogonal polynomials with weight functions $(1-x)^\alpha(1+x)^\beta + M\delta(x+1) + N\delta(x-1)$, *Canad. Math. Bull.*, **27** (1984), 205–214.

[33] A. M. Lyapunov, *The general problem of the stability of motion*, no. 7 in Collected Works II (in Russian), Kharkov Mathematical Society, 1892, Republished by the University of Tolouse 1908 and Princeton University Press 1949 (in French), in Int. J. Control 55, 531-773 and as a book by Taylor & Francis, London, 1992 (in English).

[34] M. C. Mackey and L. Glass, Oscillations and chaos in physiological control systems, *Science*, **197** (1977), 287–289.

[35] S. Maset, Numerical solution of retarded functional differential equations as abstract Cauchy problems, *J. Comput. Appl. Math.*, **161** (2003), 259–282.
[36] A. Pazy, *Semigroups of linear operators and applications to partial differential equations*, no. 44 in Applied Mathematical Sciences, Springer Verlag, New York, 1983.
[37] A. Prasad, Amplitude Death in coupled chaotic oscillators, *Phys. Rev. E*, **72** (2005), 056204–10pp.
[38] L. F. Shampine and S. Thompson, Solving DDEs in MATLAB, *Appl. Numer. Math.*, **37** (2001), 441–458.
[39] D. E. Sigeti, Exponential decay of power spectra at high frequency and positive Lyapunov exponents, *Physica D*, **82** (1995), 136–153.
[40] J. C. Sprott, A simple chaotic delay differential equation, *Phys. Lett. A*, **366** (2007), 397–402.
[41] A. Stefanski, A. Dabrowski and T. Kapitaniak, Evaluation of the largest Lyapunov exponent in dynamical systems with time delay, *Chaos, Solitons and Fract.*, **23** (2005), 1651–1659.
[42] L. N. Trefethen, *Spectral methods in MATLAB*, Software - Environment - Tools series, SIAM, Philadelphia, 2000.

Appendix A. Matlab codes. The following codes are freely available at http://users.dimi.uniud.it/~dimitri.breda/research/software/.

```
function [lambda,t]=dqr(A,T,TOL)
%DQR Discrete QR method for Lyapunov exponents of linear ODEs.
%   [lambda,t]=dqr(A,T,TOL) computes the Lyapunov exponents of
%   the linear ODE x'(t)=A(t)x(t) with the discrete QR method
%   according to [1].
%   INPUT:
%                 A = matrix function (inline)
%                 T = truncation time
%               TOL = tolerance for error control
%   OUTPUT:
%       lambda(:,i) = exponents at the i-th time step
%                 t = sequence of time steps
%   CALL:
%       >>A=@(t)matrix expression of t;
%       >>[lambda,t]=dqr(A,1e3,1e-4)
%   REFERENCES:
%       [1] L. Dieci, M.S. Jolly and E. Van Vleck, "Numerical
%           techniques for approximating Lyapunov exponents and
%           their implementation", J. Comput. Nonlinear Dynam.
%           6(1):011003-1-7, 2010.

%DOPRI54
s=7;
Ark=[0,0,0,0,0,0,0;
     1/5,0,0,0,0,0,0;
     3/40,9/40,0,0,0,0,0;
     44/45,-56/15,32/9,0,0,0,0;
     19372/6561,-25360/2187,64448/6561,-212/729,0,0,0;
     9017/3168,-355/33,46732/5247,49/176,-5103/18656,0,0;
     35/384,0,500/1113,125/192,-2187/6784,11/84,0];
brk=[5179/57600,0,7571/16695,393/640,-92097/339200,187/2100,...
     1/40];
crk=[0;1/5;3/10;4/5;8/9;1;1];
brkhat=Ark(s,1:s-1);
%inizialization
n=size(A(0),1);
I=eye(n);
t=0;
[Q,R]=qr(rand(n));
Q=Q*diag(diag(R)./abs(diag(R)));
```

```
h=.01;
lambda(:,1)=zeros(n,1);
rejects=0;
steps=0;
while t(end)+h<=T
    %IVP
    PSI(1:n,:)=Q;
    F(1:n,:)=A(t(end)+crk(1)*h)*Q;
    for i=2:s
        PSI(n*(i-1)+1:n*i,:)=Q+h*kron(Ark(i,1:i-1),I)*...
            F(1:n*(i-1),:);
        F(n*(i-1)+1:n*i,:)=A(t(end)+...
            crk(i)*h)*PSI(n*(i-1)+1:n*i,:);
    end
    Psi=Q+h*kron(brk,I)*F;
    Psihat=Q+h*kron(brkhat,I)*F(1:n*(s-1),:);
    %QR
    [Qnew,R]=qr(Psi);
    Qnew=Qnew*diag(diag(R)./abs(diag(R)));
    R=abs(diag(R));
    [~,Rhat]=qr(Psihat);
    Rhat=abs(diag(Rhat));
    %error
    e=max(abs(R-Rhat)./((1+R).*TOL));
    hnew=.8*h*(1/e^.2);
    %update
    if e<=1 %accept step
        t=[t,t(end)+h];
        h=min(hnew,5*h);
        Q=Qnew;
        lambda=[lambda,(t(end-1)*lambda(:,end)+log(R))/t(end)];
        %avoid accumulation of lambda and t if not
        %interested in monitoring the time behavior
        steps=steps+1;
    else h=max(hnew,h/5); %reject step
        rejects=rejects+1;
    end
end
end
```

```
function sol=solveDDE_MG(a,b,c,tau,x0,T)
%SOLVEDDE_MG solution of the Mackey-Glass equation by dde23.
%sol=solveDDE_MG(a,b,c,tau,x0,T) approximates the solution of the
%    Mackey-Glass equation
%
%    x'(t)=a*x(t-tau)/(1+x(t-tau)^c)-b*x(t)
%
%    exiting from x0 at time 0 up to time T by using dde23 [1].
%    INPUT:
%    a,b,c,tau = parameters
%            x0 = initial data (constant)
%             T = final time
%    OUTPUT:
%           sol = approximated solution
%    CALL:
%        >>sol=solveDDE_MG(.2,.1,10,50,2,1000);
%    REFERENCES:
%        [1] L.F. Shampine and S. Thompson, "Solving DDEs in
%            MATLAB", Appl. Numer. Math., 37:441-458, 2001.
```

```
sol=dde23(@rhs,tau,x0,[-tau,T],[],a,b,c);

    function f=rhs(t,x,z,a,b,c)
        f=a*z/(1+z^c)-b*x;
    end
end
```

```
function A=lrhs_MG(t,sol,a,b,c,tau,M)
%LRHS_MG linearized Mackey-Glass DDE reduced to ODE.
%   A=lrhs_MG(t,sol,a,b,c,tau,M) computes the right-hand side
%   matrix of the system of M+1 ODEs approximating the
%   Mackey-Glass equation linearized around the reference
%   solution sol according to [1] and evaluated at time t.
%   INPUT:
%           t = time
%         sol = reference solution
%   a,b,c,tau = parameters
%           M = degree of approximation
%   OUTPUT:
%           A = right-hand side matrix
%   CALL:
%       >>A=lrhs_MG(150,sol,.2,.1,10,50,10);
%   REFERENCES:
%       [1] Breda D., Diekmann O., Gyllenberg M., Scarabel F. and
%           Vermiglio R.. "Pseudospectral discretization of
%           nonlinear delay equations: new prospects for
%           numerical bifurcation analysis", SIAM J. Appl. Dyn.
%           Syst., 15(1):1-23, 2016.

theta=tau*(cos((0:M)*pi/M)-1)/2;
A=difmat(theta);
z=deval(sol,t-tau);
A(1,:)=[-b,zeros(1,M-1),a*(1+(1-c)*z^c)/((1+z^c)^2)];

    function D=difmat(x)
        %DIFMAT differentiation matrix on Chebyshev II nodes.
        % D=DIFMAT(x) returns the differentiation matrix D
        % relevant to the M+1 Chebyshev II nodes x according
        % to [2].
        %  INPUT:
        %      x = vector of Chebyshev II nodes
        %  OUTPUT:
        %      D = differentiation matrix
        %  CALL:
        %      >>M=10;
        %      >>D=difmat(cos((0:M)*pi/M));
        %  REFERENCES:
        %      [2] L.N. Trefethen, "Spectral methods in Matlab",
        %          SIAM, 2000.

        MM=length(x)-1;
        if MM==0
            D=0;
            return
        end
        C=[2;ones(MM-1,1);2].*(-1).^(0:MM)';
        X=repmat(x',1,MM+1);
        dX=X-X';
```

```
            D=(C*(1./C)')./(dX+(eye(MM+1)));
            D=D-diag(sum(D'));
        end
    end
```

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: dimitri.breda@uniud.it
E-mail address: dellaschiava.sara@spes.uniud.it