SIAM J. Applied Dynamical Systems
Vol. 20, No. 3, pp. 1514–1543

Numerical Computation of Periodic Orbits and Isochrons for State-Dependent Delay Perturbation of an ODE in the Plane*

Joan Gimeno†, Jiaqi Yang‡, and Rafael de la Llave‡

Abstract. We present algorithms and their implementation to compute limit cycles and their isochrons for state-dependent delay differential equations (SDDEs) which are perturbed from a planar ordinary differential equation (ODE) with a limit cycle. Note that the space of solutions of an SDDE is infinite dimensional. We compute a two parameter family of solutions of the SDDE which converges to the solutions of the ODE as the perturbation goes to zero in a neighborhood of the limit cycle. The method we use formulates functional equations among periodic functions (or functions converging exponentially to periodic). The functional equations express that the functions solve the SDDE. Therefore, rather than evolving initial data and finding solutions of a certain shape, we consider spaces of functions with the desired shape and require that they are solutions. The mathematical theory of these invariance equations is developed in a companion paper [J. Yang, J. Gimeno, and R. de la Llave, *SIAM J. Math. Anal.*, 53 (2021), pp. 4031–4067], which provides proofs of a posteriori theorems. They show that if there is a sufficiently approximate solution (with respect to some explicit condition numbers), then there is a true solution close to the approximate one. Since the numerical methods produce an approximate solution, and provide estimates of the condition numbers, we can make sure that the numerical solutions we consider approximate true solutions. In this paper, we choose a systematic way to approximate functions by a finite set of numbers (Taylor–Fourier series) and develop a toolkit of algorithms that implement the operators—notably composition—that enter into the theory. We also present several results obtained by running the implementations in some representative cases.

Key words. state-dependent delay, perturbation theory, parameterization method

AMS subject classifications. 37M05, 65T50, 37M15

DOI. 10.1137/20M1336965

1. Introduction.

Many phenomena in nature and technology are described by limit cycles, and by now there is an extensive mathematical theory of them [30, 1].

These limit cycles often arise in feedback loops between effects that pump and remove energy in ways that depend on the state of the system. When the feedback happens instantaneously, these phenomena are modeled by an ordinary differential equation (ODE). Neverthe-

*Received by the editors May 11, 2020; accepted for publication (in revised form) by J. Sieber June 7, 2021; published electronically September 7, 2021.
https://doi.org/10.1137/20M1336965
Funding: The second and third authors were partially supported by NSF grant DMS-1800241. The first author acknowledges financial support from Spanish grants MDM-2014-0445, PGC2018-100699-B-I00 (MCIU/AEI/FEDER, UE), Catalan grant 2017 SGR 1374, and Italian grant MIUR-PRIN 20178CJA2B "New Frontiers of Celestial Mechanics: Theory and Applications." This research was also funded by H2020-MCA-RISE 734577 which supported visits of the first author to the Georgia Institute of Technology and of the second author to the University of Barcelona.
†Department of Mathematics, University of Rome Tor Vergata, Via della Ricerca Scientifica 1, 00133 Rome, Italy (joan@maia.ub.es).
‡School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 USA (jyang373@gatech.edu, rafael.delallave@math.gatech.edu).

less, in many real phenomena, the feedback takes time to start acting; see, e.g., [3, 14]. In such cases, the delay differential equations (DDEs) play a role. When the delay term depends on the state itself, we end up with state-dependent delay differential equations (SDDEs). Such delays are documented to be important in several areas of science and technology (e.g., in electrodynamics, population dynamics, neuroscience, circuits, manufacturing, etc.; see [21] for a relatively recent survey documenting many areas where SDDEs are important models).

Note that, from the mathematical point of view, adding even a small delay term in the ODE model is a very singular perturbation since the nature of the problem changes drastically. Notably, the natural phase spaces in delay equations are infinite dimensional (there is some discussion about which are the most natural ones) rather than the finite dimensional phase spaces of ODEs.

One would heuristically expect that, if the delay term is a small quantity, there are solutions of the delay problem that resemble the solutions of the unperturbed ODE. Due to the singular perturbation nature of the problem, justifying this intuitive idea is a nontrivial mathematical task. Of course, besides the finite dimensional set of solutions that resemble the solutions of the ODE, one expects many other solutions, which may be very different.

The recent rigorous paper [37] describes a formalism to study the effect of introducing a delay to an ODE in the plane with a limit cycle. The paper shows that, in some appropriate sense, the solutions of the ODE persist. The method is, in fact, constructive since it is based on showing that the iterations of an explicit operator converge.

The goal of this paper is to present algorithms and implementation details and provide some source code as supplementary material (M133696_01.zip [local/web 368KB]) for the mathematical arguments developed in [37]. The solutions we compute—and which resemble the solutions of the ODE—capture the full dynamics of the SDDE in the sense that the solutions of the SDDE in a neighborhood converge to this finite dimensional solution family very quickly.

One of the novelties of the method in [37] consists in bypassing the evolution operator and formulating the existence of periodic orbits (and the solutions converging to them) as the solutions in a class of functions with periodicity.

On the contrary, if one studies periodic orbits as fixed points of an evolution operator, one needs to study the smooth dependence of the solutions on the initial data and on parameters, which is a delicate question for general solutions (see [21]). The approach developed in [37] does not need to study such a dependency in order to obtain the persistence of the limit cycle under the perturbation. In fact, the smooth dependence on parameters in [37] is a corollary rather than a requirement.

Our approach may be generalized to other forms of history dependence using the concept of extendable differentiability of the right-hand side as proposed in [28].

The companion paper [37] establishes an a posteriori theorem which states that given a sufficiently approximate solution of the invariance equation, there is a true solution close to it. To be more precise, an approximate solution is sufficiently approximate if the error is smaller than an explicit expression involving several properties of the approximate solution (commonly called condition numbers).

The numerical methods developed and run here produce an approximate solution and obtain estimates on the condition numbers. So, we can be quite confident that the solutions

produced by our numerical methods correspond to true solutions. The a posteriori results justify, in fact, that the approximate solution is independent of the method for which it has been produced.

The algorithms consist in specifying discretizations for all the functional analysis steps in [37]. We do not present rigorous estimates on the effects of discretizations (they are in principle applications of standard estimates), but we present analysis of running times. We have implemented the algorithms above and report the results of running them in some representative examples. In our examples, one can indeed obtain very accurate solutions in a few minutes using a currently available standard laptop.

In addition to the numerical approximations, it is also customary in applied mathematics to produce approximate solutions using formal asymptotic expansions. For the problem at hand, the paper [10] develops formal asymptotic expansions of the periodic solutions in powers of the term in the delay.

The expansions in [10] are readily computable with the methods presented here. They can be taken as starting points for the fixed point method in [37]. Moreover, our a posteriori results in [37] show that these expansions are asymptotic in a very strong sense.

1.1. Organization of the paper.

The paper is organized in an increasing level of details trying to guide the reader from the general steps of the algorithms to the more specialized and most difficult steps.

First, we detail in section 2 an overview of the method developed in [37]. In particular, we first introduce the unperturbed problem in section 2.1 in order to move to the perturbed problem in section 2.2. That will lead to the explicit expression of the invariance equation in section 2.3 and the periodicity and normalization conditions in sections 2.7 and 2.8, respectively.

The algorithms that allow us to solve the invariance equation introduced in section 2.3 are fully detailed in section 3.

The numerical composition of periodic mappings as well as its computational complexity needs special care. Hence, section 4 explains in detail such a process in a Fourier representation.

In section 5 we report the results in some representative examples.

Finally, we present conclusions in section 6, where we introduce general remarks on the novel results in [37].

Our results have as input the outputs of the unperturbed case. They can be obtained from standard ODE techniques. For completeness, we summarize in Appendix A the steps and add practical comments of the parameterization method strategy described in [23].

As our numerical representation for periodic orbit is going to be one-dimensional Fourier expansion, we add Appendix B to summarize possibly well-known results of this kind of representation and how they are managed and packed from a programming point of view.

2. Overview of the problem and the method.

2.1. The parameterization method for limit cycles and their isochrons in ODEs.

Our numerical starting point is the main result in [23], which we recall informally (omitting precisions on regularity, domains of definition, etc.).

Given an analytic ordinary differential equation (ODE) in the plane

$$\dot{x} = X_0(x) \tag{2.1}$$

with a (stable) limit cycle, there are an analytic local diffeomorphism K, in particular a local change of variables, defined from $\mathbb{T} \times [-1, 1]$ to $\mathbb{R}^2$, a frequency $\omega_0 > 0$, and a rate $\lambda_0 < 0$ such that

$$X_0 \circ K(\theta, s) = (\omega_0 \partial_\theta + \lambda_0 s \partial_s) K(\theta, s) = DK(\theta, s) \begin{pmatrix} \omega_0 \\ \lambda_0 s \end{pmatrix}. \tag{2.2}$$

Hence, if θ and s satisfy the very simple ODE

$$\begin{aligned} \dot{\theta}(t) &= \omega_0, \\ \dot{s}(t) &= \lambda_0 s(t), \end{aligned} \tag{2.3}$$

then

$$x(t) = K(\theta(t), s(t))$$

is a solution of (2.1) in a neighborhood of the limit cycle.

Therefore, the paper [23] trades finding all the solutions near the limit cycle of (2.1) for finding K, ω_0, and λ_0 solving (2.2). The paper also develops efficient algorithms for the study of (2.2), the so-called invariance equation. We provide Appendix A which reproduces the algorithm, adding some practical comments from our implementation.

The key idea of the formalism in [37] consists in accommodating the delay by just changing (2.2). We will obtain a modified functional equation, which involves nonlocal terms that reflect the delay in time. This equation was treated in [37]. Hence, we will produce a two-dimensional family of solutions of the delay problem which resemble the solutions of the unperturbed problem (2.1).

The solutions we construct for the SDDE are analogues of the limit cycle as well as the solutions that converge to the limit cycle exponentially fast (notice that for the simple ODE, these are all the solutions with initial data in a neighborhood of the limit cycle).

The set $I_{\theta_0} = \{K(\theta_0, s_0) \colon s_0 \in [-1, 1]\}$ is called in the biology literature the *"isochron"* of θ_0 because the orbit of a point in I_{θ_0} converges to the limit cycle with a phase θ_0. See [36].

2.1.1. Isochron term. The theory of normally hyperbolic manifolds shows that the isochrons are the same as the stable manifolds of points [18] (see also [11] for generalizations beyond normal hyperbolicity). Therefore, in the ODE case, isochrons and stable manifolds can be used interchangeably.

However, the theory of delay equations in [19, Chapter 10] (specially Theorem 3.2) produces stable (or strong stable) manifolds in the (infinite dimensional) phase space. Thus, the stable manifolds produced there are infinite dimensional.

The solutions we construct are finite dimensional families. To avoid confusion with the version of the stable manifolds in [19], we prefer to maintain the name isochrons to refer to the solutions we construct. Thus, the isochrons that we can construct for the cases in [19] are subsets of the (infinite dimensional) manifolds there. Note that, since the evolution operator is compact, most of the eigenvalues of the evolution are very small in modulus, in particular,

smaller than our choice of λ, so that the solutions in the stable manifold converge to the space of solutions produced here in a very fast way.

As a matter of fact, our isochrons are slow manifolds—they correspond to the least stable eigenvalues. In applications, the isochrons will be the most observable solutions since they correspond to the modes that decrease the slowest so that any solution will converge to the isochron much faster than the isochron converges to the limit cycle (an analogue of what happens in ODEs in a stable node).

2.2. The perturbed problem. We consider now a perturbation of (2.1) of the form

$$\begin{aligned}\dot{x}(t) =& X\big(x(t), \varepsilon x(t - r(x(t)))\big)\\ \coloneqq& X(x(t), 0) + \varepsilon P(x(t), x(t - r(x(t))), \varepsilon),\end{aligned} \tag{2.4}$$

where $0 < \varepsilon \ll 1$, $X(x(t), 0) = X_0(x(t))$, $\varepsilon P(x(t), x(t-r(x(t))), \varepsilon) \coloneqq X\big(x(t), \varepsilon x(t-r(x(t)))\big) - X(x(t), 0)$, and the function r is defined in a subset of $\mathbb{R}^2$, with positive values and as smooth as we need, hence bounded in compact sets.

Equation (2.4) is an SDDE for $\varepsilon \neq 0$. For typographical reasons we will denote $\widetilde{x}(t) \coloneqq x(t - r(x(t)))$, and then (2.4) can just be written as

$$\dot{x}(t) = X_0(x(t)) + \varepsilon P(x(t), \widetilde{x}(t), \varepsilon).$$

Notice that because of the perturbative nature of the companion paper [37], the function r in (2.4) can be rather general. That is, r does not need to be restricted to strictly positive values, which leads to applications to advanced or mixed differential equations. The reason is because in the proof of existence, only bounds of r and some of its first derivatives in a neighborhood of the limit cycle are used.

The map r can also depend on parameters, and the same results in the companion theoretical paper prove the smooth dependency on the solutions with respect to those parameters.

We point out that we are not considering an explicit dependency on time in the mapping r. In such a case, we believe that slight modifications in the theoretical paper may be required.

In the current paper we also do not explicitly address the case that r has domain in a subset of $C^k([-h, h], \mathbb{R}^2)$ for some $h > 0$ and some integer $k \geqslant 0$. Theoretically, it would require a deep discussion of the smoothness condition of r as well as a small range of ε to fulfill the convergence conditions of the corresponding a posteriori theorem. Therefore it would be harder to numerically get the convergence.

2.3. The invariance equation in the perturbed problem. Let $\tilde{\mathbb{T}}$ be the universal cover of the one-dimensional torus $\mathbb{T}$, and let us consider a map $K\colon \tilde{\mathbb{T}} \times [-1, 1] \to \mathbb{R}^2$, the frequency as ω_0, and the rate as λ_0, which solve (2.2). They correspond to the case $\varepsilon = 0$ in (2.4).

In analogy with the ODE case, we want to find a $W(\theta, s)$ with periodicity in the first variable and numbers ω and λ such that for all θ and s,

$$x(t) = K \circ W(\theta + \omega t, s e^{\lambda t}) \tag{2.5}$$

is a solution of (2.4).

The mapping W gives us a parameterization of the limit cycle with its isochrons via the change of coordinates $K \circ W(\theta, s)$. That is, the limit cycle will be represented by the set

$\{K \circ W(\theta, 0) \colon \theta \in \mathbb{T}\}$, and the isochron associated to the angle θ in $\mathbb{T}$ will be $\{K \circ W(\theta, s) \colon s \in [-s_0, s_0]\}$, where s_0 denotes a region of validity in s in the solution (2.5). This region will depend on the perturbative parameter.

Note that heuristically (and it is also shown in [37]) W is close to the identity map, and ω and λ are close to the values in the unperturbed case. Hence, we will produce a two-dimensional family of solutions of the delayed equation (2.4) which resemble the solutions of the ODE.

Remark 2.1 (number of solutions of (2.5)). Since the phase space of the delay equations is infinite dimensional, there are many more solutions of (2.4) with different λ, and possibly they are complex numbers.

Imposing that the tuple (W, ω, λ) is such that (2.5) is a solution of (2.4) and knowing that the tuple (K, ω_0, λ_0) is also a solution of (2.2) but for $\varepsilon = 0$, then

$$DK \circ W DW = DK \circ W \begin{pmatrix} \omega_0 \\ \lambda_0 W_2 \end{pmatrix} + \varepsilon P(K \circ W, K \circ \widetilde{W}, \varepsilon), \tag{2.6}$$

where W_2 refers to the second component of W.

Now, since K is a local diffeomorphism, it also acts as a change of variable. In particular, we can premultiply (2.6) by $(DK \circ W)^{-1}$ to get the functional equation, whose unknowns are $W \equiv (W_1, W_2)$, ω, and λ:

$$(\omega \partial_\theta + \lambda s \partial_s) W(\theta, s) = \begin{pmatrix} \omega_0 \\ \lambda_0 W_2(\theta, s) \end{pmatrix} + \varepsilon Y\big(W(\theta, s), \widetilde{W}(\theta, s), \varepsilon\big), \tag{2.7}$$

where we use the shorthand

$$\widetilde{W}(\theta, s) \coloneqq W\big(\theta - \omega r(K \circ W), s e^{-\lambda r (K \circ W)}\big),$$
$$Y\big(W(\theta, s), \widetilde{W}(\theta, s), \varepsilon\big) \coloneqq (DK \circ W(\theta, s))^{-1} P\big(K \circ W(\theta, s), K \circ \widetilde{W}(\theta, s), \varepsilon\big).$$

Equation (2.7) is called the invariance equation, and it will be the center of our attention. Let us start by making some preliminary remarks about it.

We have ignored the precise definition of the domain of the function W. We need the range of W to be contained in the domain of K. Note also that it is not clear that the domain of the right-hand side can match the domain of the left-hand side of (2.7). As it turns out, this will not matter much for our treatment provided that ε is small enough (see [37] for a detailed discussion).

From the point of view of analysis, one of the main difficulties of (2.7) is that it involves a function composed with itself (hence the operator is not really differentiable). Also the term $\widetilde{W}$ does not have the same domain as W. We refer the reader to [37] for a deeper discussion of the composition domain.

Similar problems appear in the treatment of center manifolds [29, 9], and indeed, in [37], there are only results for finite differentiable solutions, and the solutions obtained may depend on cut-offs and extensions taken to solve the problem.[1]

[1] On the other hand, the coefficients of the expansion in powers of s are unique and do not depend on cut-offs and extensions.

Based on the experience with center manifolds, we believe, indeed, that the solutions could only be finitely differentiable and that there are different solutions of the invariance equation (depending on the extensions considered).

Remark 2.2. In the language of ergodic theory, for those familiar with it, the results of [37] can be described as saying that there is a *factor* in the (infinite dimensional) phase space of the SDDE which is a two-dimensional flow with dynamics close to the dynamics of the ODE.

In this paper, we will compute numerical approximations of the map giving the semiconjugacy as well as the new dynamics of such a *factor*.

2.4. Format of solution for the invariance equation (2.7). It is shown in [37] that, for small ε, one can construct smooth solutions of (2.7) of the form

$$W(\theta, s) = W^0(\theta) + W^1(\theta)s + \sum_{j=2}^{n} W^j(\theta)s^j + W^{>}(\theta, s), \tag{2.8}$$

where $W^j \colon \mathbb{T} \to \mathbb{T} \times \mathbb{R}$ and $W^{>} \colon \mathbb{T} \times [-s_0, s_0] \to \mathbb{T} \times \mathbb{R}$ with $W^{>}(\theta, s) = O(s^{n+1})$ and for some $s_0 > 0$.

As we will see in more detail later, if one substitutes (2.8) into (2.7) and matches powers in s, one gets a finite set of recursive equations for the coefficients W^j of the expansion (and for ω, λ). We will deal with these equations in detail later. Note that this will require a discretization of W^j, which are only functions of the angle θ.

2.5. The equations for terms of the expansion of W. Assume for the moment that W^0 and W^1 have already been computed. Then we can substitute the expansion (2.8) of W in powers of s into the invariance equation (2.7). Matching the coefficients of the powers of s on both sides, we obtain a hierarchy of equations for W^j, $j = 2, 3, \ldots$.

The equations for W^j involve just $W^0, \ldots, W^{j-1}$. Hence, they can be studied recursively. In [37] it is shown that if we know $W^0, \ldots, W^{j-1}$, it is possible to find W^j in a unique way, and, hence, we can proceed to solve the equations recursively. In this paper, we show that there are precise algorithms to compute these recursions. We also report results of implementation in some cases.

Note that W^j, $j = 0, \ldots, n$, in (2.8) are functions only of θ. The function $W^{>}$ depends on both θ and s but vanishes at high order in s and does not enter in the equations for $j = 0, \ldots, n$.

As it frequently happens in perturbative expansions, the low order equations are special. The equation for W^0—which gives the periodic solution that continues the limit cycle—also determines ω. The equation for W^1 also determines λ. The equations for W^j, $j = 2, \ldots, n$, are all similar and involve solving the same operator (with different terms).

In this paper, we will only consider the computation of the W^j, $j = 0, 1, \ldots$. The term $W^{>}$ is estimated in [37], and it is not only high order in s but also actually small in rather large balls.

Note that even if the W^j are unique (up to the parameters in (2.9)), the $W^{>}$ depends on properties of the extension considered. This is, of course, very reminiscent of what happens in the theory of center manifolds [32]. For numerical studies of expansions of center manifolds

we refer the reader to [6, 24, 31] and detailed estimates of the truncation in [8]. The numerical considerations about the effect of the truncation apply with minor changes to our case.

2.6. Uniqueness of the invariance equation. The equation (2.7) (as well as (2.2)) is underdetermined. That means if W, ω, and λ solve (2.7), then $W_{\sigma,\eta}$, ω, and λ also solve the same equation with

$$W_{\sigma,\eta}(\theta, s) = W(\theta + \sigma, \eta s). \tag{2.9}$$

The parameters σ and η correspond respectively to choosing a different origin in the angle coordinate θ and a different scale of the parameter s.

Even if all these solutions in (2.9) are mathematically equivalent, we anticipate that choosing a different η can change the reliability of the numerical algorithms.

In [37], it is shown that the solutions in the family (2.9) are locally unique. That is, all the solutions of the invariance equation (2.7) are included in (2.9). Hence the numerically computed approximate solutions of the invariance equation identify a unique solution, which is unambiguous. This uniqueness is crucial to compare different numerical runs as well as to obtain smooth dependence on parameters. Equations (2.12) and (2.13) introduce normalization conditions that specify the parameters in (2.9).

The uniqueness in [37] is somewhat subtle. The limit cycle is unique, as are the formal Taylor expansions of isochrons (their parameterizations are unique once we fix origins of coordinates and scales). On the other hand, the full isochrons are unique only when one specifies a cut-off. Similar effects happen in the study of center manifolds [32].

From the numerical point of view, we only compute the limit cycle and a finite Taylor expansion of the isochrons. The error of the remainder of the Taylor expansion is indeed very small (much smaller than other sources of numerical error, which are already small).

2.7. Periodicity conditions. From the point of view of implementation in computers, it is convenient to think of the functions K and W in (2.7) which involve angle variables (and which range on angles) as real functions with boundary conditions (in mathematical language this is described as taking lifts). Hence, we take

$$\begin{aligned} K(\theta + 1, s) &= K(\theta, s), \\ W(\theta + 1, s) &= W(\theta, s) + \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{2.10}$$

Notice that we are normalizing the angles in (2.10) to run between 0 and 1 rather than in $[0, 2\pi)$.

The periodicity conditions in (2.10) indicate the second component of W is periodic (it describes a radial coordinate) in θ, while the first component increases by 1 when θ increases by 1 (it describes an angle). Thus, the circle described by increasing θ makes the angle in the coordinate go around, so that it is a noncontractible circle in the angle.

For the expansion of W in powers of s as in (2.8), the periodicity conditions amount to

$$W^0(\theta + 1) = W^0(\theta) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad W^j(\theta + 1) = W^j(\theta) \quad \text{for } j \geqslant 1. \tag{2.11}$$

In the numerical analysis, there are many well-known ways to discretize periodic functions. We will use Fourier series, but there are also other alternatives such as periodic splines.

In general, for functions Ψ with $\Psi(\theta+1) = \Psi(\theta)+1$, we define $\widetilde{\Psi}(\theta) = \Psi(\theta) - \theta$ which is a periodic function, i.e., for all θ, $\widetilde{\Psi}(\theta+1) = \widetilde{\Psi}(\theta)$. Then we will discretize $\widetilde{\Psi}$ and rewrite the functional equations so that this is the only unknown.

2.8. Normalization of the solutions. As indicated in the discussion, the invariance equation has two obvious sources of indeterminacy: One is the choice of the origin of the variable θ (the σ in (2.9)), and the other is the choice of the scale of the variable s (the η in (2.9)). In [37] it is shown that these are the only indeterminacies for the solution up to any order n and that once we fix them, we can get any other solution by applying (2.9).

A convenient way to fix the origin of θ is to require

$$\int_0^1 \left[\partial_\theta W_1^0(\theta,0)\, W_1(\theta,0)\right] d\theta = a, \tag{2.12}$$

where W^0 is an initial approximation and a is a real number, typically close to 1. This normalization is easy to compute and is rather sensitive since, when we move in the family (2.9), the derivative with respect to the shift is a positive number.

The normalization of the origin of coordinates has no numerical consequences except for the possibility of comparing the solutions in different runs. The solutions corresponding to different normalizations have very similar properties. The numerical algorithm, Algorithm 3.1, in its step 5 leads to a small drift in the normalization in each iteration, but it is guaranteed to converge to one of the solutions in (2.9).

The second normalization is just a choice of the eigenvector of an operator. We have found it convenient to take

$$\int_0^1 \partial_s W_2(\theta,0)\, d\theta = \rho \tag{2.13}$$

with a real $\rho \neq 0$.

We anticipate that changing the value of ρ is equivalent to changing s into bs, where b is commonly named *scaling factor*.

All the choices of ρ are mathematically equivalent—they amount to setting the scale of the parameter s. The choice of this normalization, however, affects the numerical accuracy dramatically. Notice that if we change s into bs, the coefficients $W^j(\theta)$ in (2.8) change into $W^j(\theta)b^j$. Thus, different choices of b may lead the Taylor coefficients to be very large or very small, which makes the computations with them very susceptible to round-off error. It is numerically advantageous to choose the scale in such a way that the Taylor coefficients have a comparable size. In our problem, we are also going to use the scaling to ensure that the second component of W lies in the domain of K, and then $K \circ W$ is well defined.

In practice, we run the calculations twice. First we do a preliminary calculation whose only purpose is to compute an approximation of the scale that makes the coefficients remain more or less the same size. Then a more definitive calculation can be run. The latter running is more numerically reliable.

Remark 2.3. In standard implementation of the Newton method for fixed points of a functional, say Ψ, the fact that the space of solutions is two-dimensional leads $D\Psi - \mathrm{Id}$ to have a two-dimensional kernel and be noninvertible.

In our case, we will develop a very explicit and fast algorithm that produces an approximate linear right inverse. This linear right inverse leads to convergence to an element of the family (2.9).

3. Computation of (W, ω, λ)—perturbed case.

The main result in the paper [37] states that if ε in (2.7) is small enough, and a periodicity condition like (2.12) and a normalization like (2.13) are considered, then there exists a unique tuple (W, ω, λ) satisfying (2.7), (2.12), and (2.13).

The formulation of that result is done in an a posteriori format which ensures the existence of a true solution once an approximate enough solution is provided as initial guess for the iterative scheme.

Moreover, it also gives the Lipschitz dependence of the solution on parameters which allows us to consider a continuation approach.

We refer the reader to [37] for a precise formulation of the result involving choices of norms to measure the error in the approximate solutions. In the following sections, we formulate in an algorithmic way the steps to follow to converge to the new limit cycle and its isochrons.

3.1. Fixed point approach.

We compute all the coefficients $W^j(\theta)$ of the truncated expression $W(\theta, s)$ in (2.8) order by order. The zero and first orders require special attention due to the fact that the values ω and λ are obtained in (2.7) by matching coefficients of s^0 and s^1, respectively. The condition that allows us to obtain ω comes from the periodicity condition (2.11). The mapping W^0 is not a periodic function. But we can use it to get a periodic one defined by $\hat{W}^0(\theta) \coloneqq W^0(\theta) - \binom{\theta}{0}$. The condition for λ is given by the normalization condition (2.13). As in the unperturbed case, we are allowed to use a scaling factor. The use of such a scaling factor allows us to set the value of ρ in (2.13) equal to 1.

Algorithm 3.1 sketches the fixed point procedure to get ω and W^0 whose periodicity condition is ensured in step 5. In this case the initial condition will be ω_0 (the value for $\varepsilon = 0$) for ω and $\binom{\theta}{0}$ for $W^0(\theta)$ since $W(\theta, s)$ is close to the identity.

Algorithm 3.1 (s^0 case). Let $\widetilde{W^0}(\theta) \coloneqq W^0\big(\theta - \omega r \circ K(W^0(\theta))\big)$.

- ⋆ `Input:` $\dot{x} = X(x) + \varepsilon P(x, \tilde{x}, \varepsilon)$, $0 < \varepsilon \ll 1$, $K(\theta, s) = \sum_{j=0}^{m-1} K^j(\theta)(b_0 s)^j$, $b_0 > 0$, $\omega_0 > 0$, $\lambda_0 < 0$, and a tolerance *tol*.
- ⋆ `Output:` $\hat{W}^0 \colon \mathbb{T} \to \mathbb{R}^2$ and $\omega > 0$.

1. $\hat{W}^0(\theta) \leftarrow 0$ and $\omega \leftarrow \omega_0$.
2. $W^0(\theta) \leftarrow \begin{pmatrix} \theta \\ 0 \end{pmatrix} + \hat{W}^0(\theta)$.
3. Solve $DK \circ W^0(\theta)\eta(\theta) = \varepsilon P(K \circ W^0(\theta), K \circ \widetilde{W^0}(\theta), \varepsilon)$. Let $\eta \equiv (\eta_1, \eta_2)$.
4. $\alpha \leftarrow \int_0^1 \eta_1(\theta)\, d\theta$ and $\omega \leftarrow \omega_0 + \alpha$.

5. Solve $\omega\partial_\theta \hat{W}_1^0(\theta) = \eta_1(\theta) - \alpha$ imposing $\int_0^1 \hat{W}_1^0(\theta)\, d\theta = 0$.

6. Solve $(\omega\partial_\theta - \lambda_0)\hat{W}_2^0(\theta) = \eta_2(\theta)$.

7. Iterate from step 2 to step 6 until convergence in W^0 and ω with tolerance *tol*.

Algorithm 3.2 sketches the steps to compute (W^1, λ) and W^n for $n \geqslant 2$. The initial guesses are λ_0 for λ, $\binom{0}{1}$ for W^1, and $\binom{0}{0}$ for W^n.

Algorithm 3.2 (s^1 case and s^n case with $n \geqslant 2$). Let $\widetilde{W}(\theta, s) \coloneqq W\big(\theta - \omega r \circ K(W(\theta, s)), se^{-\lambda r \circ K(W(\theta,s))}\big)$.

- ⋆ `Input:` $\dot{x} = X(x) + \varepsilon P(x, \tilde{x}, \varepsilon)$, $0 < \varepsilon \ll 1$, $K(\theta, s) = \sum_{j=0}^{m-1} K^j(\theta)(b_0 s)^j$, $b_0 > 0$, $\omega_0 > 0$, $\lambda_0 < 0$, $\hat{W}^0(\theta)$, $W^j(\theta)$ for $0 < j < n$, $b > 0$, $\omega > 0$, and a tolerance *tol*.
- ⋆ `Output:` either $W^1 \colon \mathbb{T} \to \mathbb{T} \times \mathbb{R}$ and $\lambda < 0$ or $W^n \colon \mathbb{T} \to \mathbb{T} \times \mathbb{R}$.

1. $W^n(\theta) \leftarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

s^1 If $n = 1$, $W^1(\theta) \leftarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\lambda \leftarrow \lambda_0$.

2. $W(\theta, s) \leftarrow \begin{pmatrix} \theta \\ 0 \end{pmatrix} + \hat{W}^0(\theta) + \sum_{j=1}^{n} W^j(\theta)(bs)^j$.

3. $Y(W(\theta, s)) \leftarrow DK \circ W(\theta, s)^{-1} P(K \circ W(\theta, s), K \circ \widetilde{W}(\theta, s), \varepsilon)$.

4. $\eta(\theta) \leftarrow \varepsilon \frac{\partial^n Y}{\partial s^n}(W(\theta, s))_{|s=0}$. Let $\eta \equiv (\eta_1, \eta_2)$.

s^1 If $n = 1$, then $\lambda \leftarrow \lambda_0 + \int_0^1 \eta_2(\theta)\, d\theta$.

5. Solve $(\omega\partial_\theta + n\lambda)W_1^n(\theta) = \eta_1(\theta)$.

6. Solve $(\omega\partial_\theta + n\lambda - \lambda_0)W_2^n(\theta) = \eta_2(\theta)$.

7. Iterate from step 2 to step 6 until convergence with tolerance *tol*. Then undo the scaling b.

Both Algorithms 3.1 and 3.2 have nontrivial parts, such as the effective computation of $\widetilde{W}$, the numerical composition of K with W and also with $\widetilde{W}$ (see section 4), the effective computation of step 4 in Algorithm 3.2, and the choice of the scaling factor (see section 3.3). On the other hand, there are steps in which we can use the same methods as in the unperturbed case (see Appendix A), such as the solution of linear systems like step 3 in Algorithm 3.2 via Lemma 3.4 or the solutions of the cohomological equations by Proposition 3.3.

In the next sections we address each of these parts, and they have been successfully implemented and provided as self-contained supplementary material (M133696_01.zip [local/web 368KB]).

3.2. Stopping criterion. Algorithms 3.1 and 3.2 require us to stop when the prescribed tolerances have been reached respectively. Alternatively, one can stop when the invariance equation is satisfied up to the given tolerance.

3.3. Scaling factor for orders $n \geqslant 1$. As in the unperturbed case, if $W(\theta, s)$ is a solution, then $W(\theta + \theta_0, bs)$ will be a solution, too, for any θ_0 and b. A difference with the $\varepsilon = 0$ case is that now $K \circ W$ and $K \circ \widetilde{W}$ are required to be well defined. That means the second components of W and $\widetilde{W}$ must lie in $[-1, 1]$. Stronger conditions are

$$p(s) \coloneqq \sum_{j \geqslant 0} \|W_2^j(\theta)\| |s|^j \leqslant 1 \qquad \text{and} \qquad \widetilde{p}(s) \coloneqq \sum_{j \geqslant 0} \|\widetilde{W_2^j}(\theta)\| |s|^j \leqslant 1.$$

In the iterative scheme of Algorithm 3.2, these series become finite sums, and a condition for the value $b > 0$ is led by the upper-bound $\min\{s^*, \widetilde{s}^*\}$ where $s^* > 0$ is the value so that $p(s^*) = 1$ and, similarly, $\widetilde{s}^* > 0$ is the value verifying $\widetilde{p}(\widetilde{s}^*) = 1$. Notice that the solutions s^* and $\widetilde{s}^*$ exist because $\|W_2^0(\theta)\| < 1$, $\|\widetilde{W_2^0}(\theta)\| < 1$, and the polynomials have positive derivative.

3.4. Solutions of the cohomology equations in Fourier representation. Under the Fourier representation (see section B) we can solve the cohomological equations in steps 5 and 6 in Algorithm 3.1 as well as in steps 5 and 6 in Algorithm 3.2.

Proposition 3.3 (Fourier version, [23]). *Let $E(\theta, s) = \sum_{j,k} E_{jk} e^{2\pi \boldsymbol{i} k\theta} s^j$.*

- *If $E_{00} = 0$, then $(\omega \partial_\theta + \lambda s \partial_s) u(\theta, s) = E(\theta, s)$ has solution $u(\theta, s) = \sum_{j,k} u_{jk} e^{2\pi \boldsymbol{i} k\theta} s^j$ and*

$$u_{jk} = \begin{cases} \frac{E_{jk}}{\lambda j + 2\pi \boldsymbol{i} \omega k} & \text{if } (j,k) \neq (0,0), \\ \alpha & \text{otherwise} \end{cases}$$

  for all real α. Imposing $\int_0^1 u(\theta, 0)\, d\theta = 0$, then $\alpha = 0$.
- *If $E_{10} = 0$, then $(\omega \partial_\theta + \lambda s \partial_s - \lambda) u(\theta, s) = E(\theta, s)$ has solution $u(\theta, s) = \sum_{j,k} u_{jk} e^{2\pi \boldsymbol{i} k\theta} s^j$ and*

$$u_{jk} = \begin{cases} \frac{E_{jk}}{\lambda (j-1) + 2\pi \boldsymbol{i} \omega k} & \text{if } (j,k) \neq (1,0), \\ \alpha & \text{otherwise} \end{cases}$$

  for all real α. Imposing $\int_0^1 \partial_s u(\theta, 0)\, d\theta = 0$, then $\alpha = 0$.

The paper [23] also presents a solution in terms of integrals. Those integral formulas for the solution are independent of the discretization and work for discretizations such as Fourier series, splines, and collocation methods. Indeed, the integral formulas are very efficient for discretizations in splines or in collocation methods, which could be preferable in some regimes where the limit cycles are bursting. In this paper we will not use them since we will discretize functions in Fourier series and for this discretization, the methods described in Proposition 3.3 are more efficient.

3.5. Treatment of step 3 in Algorithm 3.2. To solve the linear system in step 3 of Algorithm 3.2, we can use Lemma 3.4, whose proof is a direct power matching.

Lemma 3.4. *Let $A(\theta,s)x(\theta,s)=b(\theta,s)$ be a linear system of equations for each given (θ,s), explicitly,*

$$\left(\sum_{k\geqslant 0} A_k(\theta)s^k\right)\sum_{k\geqslant 0}\boldsymbol{x}_k(\theta)s^k=\sum_{k\geqslant 0}\boldsymbol{b}_k(\theta)s^k.$$

Then, the coefficients $\boldsymbol{x}_k(\theta)$ are obtained recursively by solving

$$A_0(\theta)\boldsymbol{x}_k(\theta)=\boldsymbol{b}_k(\theta)-\sum_{j=1}^{k}A_j(\theta)\boldsymbol{x}_{k-j}(\theta),$$

which can be done provided that $A_0(\theta)$ is invertible and that one knows how to multiply and add periodic functions of θ.

3.6. Use of polynomials for elementary operations. We also recall that composition in the left of a polynomial with an exponential, trigonometric functions, powers, logarithms (or any function that satisfies an easy differential equation) can be done very efficiently using algorithms that are reviewed in [20], which go back to [27].

We present here the case of the exponential which can be used in Algorithm 3.2 for the computation of $\widetilde{W}$.

If P is a given polynomial—or a power series—with coefficients P_j, we see that $E(s)=\exp P(s)$ satisfies

$$\frac{d}{ds}E(s)=E(s)\frac{d}{ds}P(s),$$

with Taylor coefficients E_j at $s=0$. Equating like powers on both sides, this leads to $E_0=\exp P(0)$ and the recursion

$$E_j=\frac{1}{j}\sum_{k=0}^{j-1}(j-k)P_{j-k}E_k,\qquad j\geqslant 1.$$

Note that this can also be done if the coefficients of P are periodic functions of θ (or polynomials in other variables). In modern languages supporting overloading or arithmetic functions, all this can be done in an automatic manner.

Note that if the polynomial has degree n_s, the computation up to degree n_s takes $\Theta(n_s^2)$ operations of multiplications of the coefficients.

4. Numerical composition of periodic maps. The goal of this section is to deeply discuss how we can numerically compute $\widetilde{W}$ and the compositions of K with $W(\theta,s)$ and $\widetilde{W}(\theta,s)$ only having a numerical representation (or approximation) of K and W in Algorithms 3.1 and 3.2.

There are a variety of methods that can be employed to numerically get the composition of a periodic mapping with another (or the same) mapping. Some of these methods depend strongly on the representation of the periodic mapping, and others depend only on specific parts of the algorithm.

We start the discussion from the general methods to those that strongly depend on the numerical representation. One expects that the general ones will have a bigger numerical complexity or they will be less accurate.

Before starting to discuss the algorithms, it is important to stress again that for functions of two variables $(\theta, s) \in \mathbb{T} \times [-1, 1]$, there are two complementary ways of looking at them. We can think of them as functions that, given θ, produce a polynomial in s—this polynomial valued function will be periodic in θ—or we can think of them as polynomials in s taking values in spaces of periodic functions (of the variable θ). Of course, the periodic functions that appear in our interpretation can be discretized either by the values in a grid of points or by the Fourier transform.

Each of these—equivalent!—interpretations will be useful in some algorithms. In the second interpretation, we can "overload" algorithms for standard polynomials to work with polynomials whose coefficients are periodic functions (in particular, Horner schemes). In the first interpretation, we can easily parallelize algorithms for polynomials for each of the values of θ using the grid discretization of periodic functions.

Possibly the hardest part of Algorithms 3.1 and 3.2 are the compositions between K with W and K with $\widetilde{W}$. Due to step 4 of Algorithm 3.2 the composition should be done so that the output is still a polynomial in s with coefficients that are periodic functions of θ. In our implementation, we use the automatic differentiation approach [20, 17].

If $W(\theta, s) = (W_1(\theta, s), W_2(\theta, s))$ is a function of two variables taking values in $\mathbb{R}^2$, then

$$K \circ W(\theta, s) = \sum_{j=0}^{m-1} K^j(W_1(\theta, s)) \, (b_0 W_2(\theta, s))^j \,, \tag{4.1}$$

which can be evaluated with $m-1$ polynomial products and $m-1$ polynomial sums using a Horner scheme, once we have computed $K^j \circ W_1(\theta, s)$.

The problem of composing a periodic function with a periodic polynomial in s—to produce a polynomial in s taking values in the space of periodic functions—is what we consider now. In particular, we are going to discuss three different approaches and their computational complexities.

The first is the most general one, and it is based on a dynamic programming technique. It assumes some given information to build a table from which the composition can be extracted. In this case the numerical representation is in the part that is assumed to be given.

The second one exploits the Fourier representation in the inputs of the dynamic programming to provide the final full complexity of the composition.

Finally, the third approach also uses the Fourier representation, but rather than using the dynamic programming technique, it uses the recurrences in automatic differentiation for the sine and cosine functions.

4.1. Composition via dynamic programming. The most general method considers S a periodic function, the K^j in (4.1), and $q(s) = \sum_{j=0}^{k} q_j s^j$ a polynomial of a fixed order $k \geqslant 0$, where the q_j are periodic functions of θ that we consider discretized by their values in a grid (the W_1 in (4.1)).

We want to compute the polynomial $p \coloneqq S \circ q$ up to order k. Assume that $\frac{d^j}{d\theta^j} S(q_0)$ for $0 \leqslant j \leqslant k$ are given as input and that they have been previously computed in a bounded computational cost. These inputs in a computer strongly depend on the numerical representation of the periodic function S. In later sections we will consider the Fourier series as a representation which will lead to two different algorithms.

Table 1
Composition of a function with a polynomial.

	$S(q_0)$	$\frac{d}{d\theta}S(q_0)$	$\frac{d^2}{d\theta^2}S(q_0)$	$\cdots$	$\frac{d^{k-1}}{d\theta^{k-1}}S(q_0)$	$\frac{d^k}{d\theta^k}S(q_0)$
p_0	1	0				
p_1	0	$\frac{d}{ds}q(s)$	0			
p_2	0	$\frac{1}{2}\square$	$\frac{1}{2}\square$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	0	
p_{k-1}	0	$\frac{1}{k-1}\square$	$\frac{1}{k-1}\square$	$\cdots$	$\frac{1}{k-1}\square$	0
p_k	0	$\frac{1}{k}\square$	$\frac{1}{k}\square$	$\cdots$	$\frac{1}{k}\square$	$\frac{1}{k}\square$

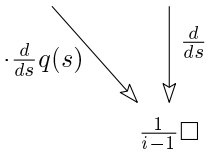

Figure 1. *Generation rule for $i = 2, \dots, k+1$ Table 1 entries.*

The chain rule gives us a procedure to compute the coefficients of $p(s) = \sum_{j=0}^{k} p_j s^j$. Indeed, one can build a table, whose entries are polynomials in s, like in Table 1 and following the generation rule in Figure 1.

The inputs of Table 1 are $a_{i,1} = 0$ for $i \neq 1$ and $a_{2,2} = \frac{d}{ds}q(s)$. Then the entries a_{ij} with $2 \leqslant j \leqslant i \leqslant k+1$ are given by

$$a_{ij}(s) = \frac{1}{i-1}\left(\frac{d}{ds}a_{i-1,j}(s) + a_{i-1,j-1}(s)\frac{d}{ds}q(s)\right). \tag{4.2}$$

Thus, the coefficients of $p(s)$ are $p_j = \sum_{l=0}^{k} a_{jl}(0)\frac{d^l}{d\theta^l}S(q_0)$ for $0 \leqslant j \leqslant k$.

Note that it is enough to store in memory k entries of Table 1 to compute all the coefficients p_j.

Moreover, for each entry in the ith row with $i = 2, \dots, k+1$, one only needs to consider polynomials of degree $k+1-i$. Overall the memory required is at most $\frac{1}{2}k(k+1)$. The number of arithmetic operations following the rule (4.2) is given by Proposition 4.1.

Proposition 4.1. *Let S be a real-periodic function, and let $q(s)$ be a real polynomial of degree k. Assume $\frac{d^j}{d\theta^j}S(q(0))$ for $j = 0, \dots, k$. The polynomial $S \circ q$ can be computed using Table 1 with $\frac{1}{2}k(k+1)$ units of memory and $\Theta(k^4)$ multiplications and additions.*

Proof. Note that $k(k+1)$ multiplications and $(k+1)^2$ additions are needed to perform the product of two polynomials of degree k. Also k multiplications are needed to perform

the derivative of a polynomial of degree k multiplied by a scalar. To bound the number of operations we must consider three different situations of Table 1.

1. The column $a_{3..k,2}$: $\sum_{i=1}^{k-2}(k-i+1) = \frac{1}{2}(k^2+k-6)$ multiplications.
2. The diagonal $a_{3..k,3..k}$:
   - $\sum_{j=1}^{k-2}(k-j-1)(k-j+1)+1 = \frac{1}{6}(2k^3-3k^2+k-6)$ multiplications.
   - $\sum_{j=1}^{k-2}(k-j-1)^2+1 = \frac{1}{6}(2k^3-9k^2+19k-18)$ additions.
3. The rest:
   - $\sum_{j=1}^{k-2}\sum_{i=j+1}^{k-2}(k-i-1)(k-i+1)+(k-i-2)+1 = \frac{1}{12}(7k^4-56k^3+71k^2+38k-24)$ multiplications.
   - $\sum_{j=1}^{k-2}\sum_{i=j+1}^{k-2}(k-i-1)^2+(k-i)+1 = \frac{1}{12}(5k^4-36k^3+85k^2-102k+72)$ additions.

Overall, there are $\frac{7}{12}k^4+\Theta(k^3)$ multiplications and $\frac{5}{12}k^4+\Theta(k^3)$ additions. ∎

The next theorem, Theorem 4.2, summarizes the previous explanations and provides the complexities to numerically compute $K \circ W$ in (4.1). It assumes that $\frac{d^i}{d\theta^i}S(q_0)$ of Table 1, which are the $\frac{d^i}{d\theta^i}K^j(W_1(\theta,0))$ in $K\circ W$, are given as input. These inputs are the only elements in Table 1 that depend on the numerical representation of the periodic functions (i.e., the K^j in $K \circ W$), and they make the result in Theorem 4.2 independent of how periodic functions are represented.

Theorem 4.2. *For a fixed θ, the computational complexity to compute the compositions of $K(\theta,s) = \sum_{j=0}^{m-1}K^j(\theta)(b_0 s)^j$ with $W(\theta,s) = \sum_{j=0}^{k-1}W^j(\theta)(bs)^j$ and $\widetilde{W}(\theta,s)$ using Table 1 is $\Theta(mk^4)$ and space $\Omega(k^2)$ assuming $\frac{d^i}{d\theta^i}K^j(W_1^0(\theta))$ as input for $i=0,\dots,k-1$.*

Remark 4.3. In general, if n_θ denotes the mesh size of the variable θ, we will have $k \leqslant m \ll n_\theta$. That is, the mesh size will be much larger than the degree (in s) of $K(\theta,s)$. That means that the parallelization in n_θ will be more advantageous.

Theorem 4.2 has an important assumption involving $\frac{d^i}{d\theta^i}K^j(W_1^0(\theta))$ which can have a big impact in the complexity of $K\circ W(\theta,s)$. However, such an impact strongly depends on the numerical representation of K^j, and it will be discussed in the Fourier representation case.

4.2. Composition in Fourier. Theorem 4.2 reduces the problem of computing $K\circ W(\theta,s)$ in (4.1) to the problem of computing composition of a periodic function with another one.

In the case of a Fourier representation (see Appendix B) of an arbitrary mapping $S\colon \mathbb{T}\to\mathbb{R}$ (the K^j's in (4.1)), such a composition between Fourier truncated series may require one to know the values not in the standard equispaced mesh $\{k/n_\theta\}_{k=0}^{n_\theta-1}$ of θ, which hampers the use of the fast Fourier transform (FFT). Indeed, the FFT states a fast way to biject $\{S(k/n_\theta)\}_{k=0}^{n_\theta-1}\subset\mathbb{R}$ to $\{\widehat{S}^j\}_{j=0}^{n_\theta-1}\subset\mathbb{C}$ such that

$$S(k/n_\theta) = \sum_{j=0}^{n_\theta-1}\widehat{S}^j e^{2\pi \boldsymbol{i} jk/n_\theta} \quad \text{and} \quad \widehat{S}^j = \frac{1}{n_\theta}\sum_{k=0}^{n_\theta-1}S(k/n_\theta)e^{-2\pi \boldsymbol{i} jk/n_\theta}. \tag{4.3}$$

It assumes the mesh of θ to be equispaced. However, the $S(q_0)$ may require evaluating S outside the mesh.

A direct composition of real Fourier series requires a computational complexity $\Theta(n_\theta^2)$. However, nowadays recent algorithms with a $\Theta(n_\theta \log n_\theta)$ complexity efficiently solve this

possible bottleneck in the performance of our algorithms. See, for instance, the NFFT3 in [25] or FINUFFT in [5, 4]. The package NFFT3 allows us to express $S\colon \mathbb{T} \to \mathbb{R}$ with the same coefficients as in (B.1) and perform its evaluation in an even number of nonequispaced nodes $(\theta_k)_{k=0}^{n_\theta-1} \subset \mathbb{T}$ by

$$S(\theta_k) = \sum_{j=0}^{n_\theta-1} \widehat{S}^j e^{-2\pi \boldsymbol{i}(j-n_\theta/2)(\theta_k-1/2)}. \tag{4.4}$$

The corrections of θ_k in (4.4) are necessary because NFFT3 considers $\mathbb{T} \simeq [-1/2, 1/2)$ rather than the other standard equispaced discretization in $[0, 1)$. NFFT3 uses some window functions for a first approximation as a cut-off in the frequency domain and also for a second approximation as a cut-off in the time domain. This package takes these approximations under control (by bounds) to ensure the solution is a good approximation. Joining these results with Proposition 4.1 we can rewrite Theorem 4.2 as follows.

Theorem 4.4. *The computational complexity to compute in Algorithm 3.2 the compositions of* $K(\theta, s) = \sum_{j=0}^{m-1} K^j(\theta)(b_0 s)^j$ *with the maps* $W(\theta, s) = \sum_{j=0}^{k-1} W^j(\theta)(bs)^j$ *and* $\widetilde{W}(\theta, s) = \sum_{j=0}^{k-1} \widetilde{W}^j(\theta)(bs)^j$ *using Table 1 and* NFFT3*, and assuming that* K^j*,* W^j*, and* $\widetilde{W}^j$ *are expressed with* n_θ *Fourier coefficients, is* $\Theta(mk^4 n_\theta + mkn_\theta \log n_\theta)$*. The space complexity is* $\Omega(kn_\theta + k^2)$*.*

Remark 4.5. Remark 4.3 also applies to Theorem 4.4 in terms of the parallelization of n_θ due to the fact that in general $k \leqslant m \ll n_\theta$. However, in the parallelism case, the space complexity increases to $\Omega(kn_\theta + k^2 n_p)$ with n_p the number of processes, although the part corresponding to kn_θ can be shared memory.

In particular, the NFFT3 can also be used for the zero order W^0 of Algorithm 3.1 giving in that case the same complexity as in Theorem 4.4 but with $k = 1$.

4.3. Automatic differentiation in Fourier. Theorem 4.2 tells us that the composition $K \circ W(\theta, s)$ can numerically be done independently of the periodic mapping representation. Nevertheless, differentiation is a notoriously ill-posed problem due to the lack of information in the discretized problem. Thus, Table 1 is a good option when no advantage of the computer periodic representation exists or $k \ll m$.

Using the representation (B.3), we can use the Taylor expansion of the sine and cosine by recurrence [27, 20]. That is, if $q(s)$ is a polynomial, then $\sin q(s)$ and $\cos q(s)$ are given by $s_0 = \sin q_0$, $c_0 = \cos q_0$, and for $j \geqslant 1$,

$$s_j = \frac{1}{j} \sum_{k=0}^{j-1} (j-k) q_{j-k} c_k, \qquad c_j = -\frac{1}{j} \sum_{k=0}^{j-1} (j-k) q_{j-k} s_k. \tag{4.5}$$

Therefore the computational cost to obtain the sine and cosine of a polynomial is linear with respect to its degree.

Theorem 4.6 says that the composition of K with W or $\widetilde{W}$ is, rather than $\Theta(mk^4 n_\theta + mkn_\theta \log n_\theta)$ as in Theorem 4.4, just $\Theta(mkn_\theta^2)$. Therefore if $k \ll m$ and n_θ is large, the approach given by Theorem 4.4 has a better complexity, although Theorem 4.6 will be more stable for larger k.

Theorem 4.6. *The computational complexity to compute in Algorithm 3.2 the compositions of* $K(\theta,s)=\sum_{j=0}^{m-1}K^j(\theta)(b_0 s)^j$ *with the maps* $W(\theta,s)=\sum_{j=0}^{k-1}W^j(\theta)(bs)^j$ *and* $\widetilde{W}(\theta,s)=\sum_{j=0}^{k-1}\widetilde{W}^j(\theta)(bs)^j$ *using automatic differentiation and assuming that* K^j, W^j, *and* $\widetilde{W}^j$ *are expressed with* n_θ *Fourier coefficients is* $\Theta(mkn_\theta^2)$.

5. Numerical results. The van der Pol oscillator [35] is an oscillator with a nonlinear damping governed by a second order differential equation.

As an example, we consider the state-dependent perturbation of the van der Pol oscillator like in [22], which has the form

$$\begin{aligned}\dot{x}(t)&=y(t),\\ \dot{y}(t)&=\mu(1-x(t)^2)y(t)-x(t)+\varepsilon x(t-r(x(t))),\end{aligned}\tag{5.1}$$

with $\mu>0$ and $0<\varepsilon\ll 1$. For the delay function $r(x(t))$ we are going to consider two cases: a pure state-dependent delay case $r(x(t))=0.006e^{x(t)}$ or just a constant delay case $r(x(t))=0.006$.

The first step consists in computing the change of coordinate K, the frequency ω_0 of the limit cycle, and its stability value $\lambda_0<0$ for $\varepsilon=0$. By standard methods of computing periodic orbits and their first variational equations, we compute the limit cycles close to $(x,y)=(2,0)$ for different values of μ. Table 2 shows the values of ω_0 and λ_0 for each of those values of the parameter μ.

Table 2

Values of ω_0 *and* λ_0 *for different values of the parameter* μ *in (5.1) with* $\varepsilon=0$.

μ	ω_0	λ_0
0.25	0.1585366857025485	−0.2509741760777654
0.5	0.1567232109993800	−0.5077310891698608
1	0.1500760842377394	−1.0593769948418550
1.5	0.1409170454968141	−1.6837946490433340

The computation of $K(\theta,s)$, following Algorithm A.1, up to order 16 in s and with a Fourier mesh size of 1024, allows us to plot the isochrons in Figure 2.

In the case of ODEs, the isochrons computed by evaluating the expansion can be globalized by integration of the ODE (5.1) forward and backward in time; see [23]. In the case of the SDDE, $\varepsilon\neq 0$, propagating backwards is not possible. We hope that this limitation can be overcome, but this will require some new rigorous developments and more algorithms. We think that this is a very interesting problem.

A relevant indicator for engineers is the power spectrum, i.e., the square of the modulus of the complex Fourier coefficients. In Figure 3 we illustrate the power spectrum for K^0, since K^0 is the one that is commonly observed in a circuit system.

Due to the quadratic convergence of Algorithm A.1 (see [23]), the computations of Table 2 and Figure 2 are performed in less than one minute in a currently available standard laptop. However, we notice that for values of $\mu>1.5$ the method may not converge for the unperturbed case. The scaling factor and the Fourier mesh size need to be smaller due to spikes, especially for the high orders in s, i.e., $K^j(\theta)$ for large j. This is an inherent drawback of the numerical

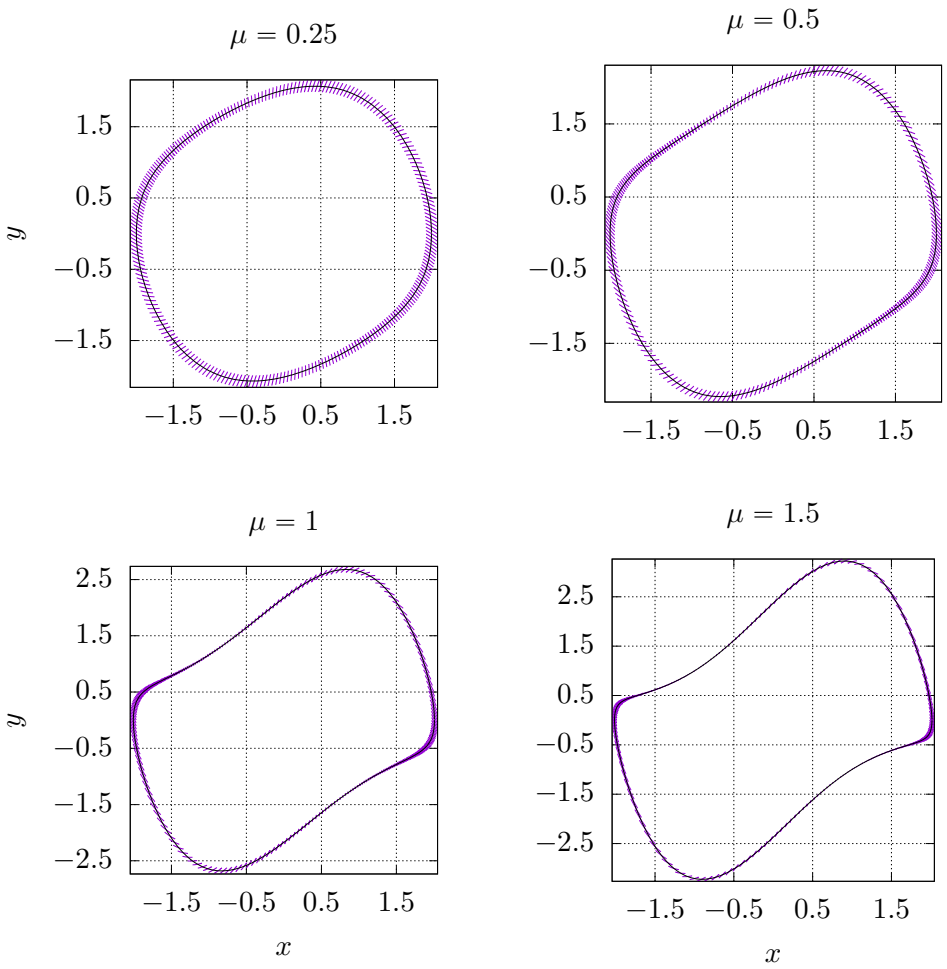

Figure 2. *Limit cycles and their isochrons for different values of the parameter μ in the unperturbed case,* (5.1).

representation of periodic functions by Fourier series, which may become very important for some parameter values.

5.1. Perturbed case. Let us analyze the case of $\mu = 1.5$ for two different types of delay functions: a constant one $r(x(t)) = 0.006$ and a state-dependent one $r(x(t)) = 0.006e^{x(t)}$.

The two cases have some advantages to be exploited. For instance, in the constant case $\widetilde{W}(\theta, s) = W(\theta - \omega\beta, se^{-\lambda\beta})$ is easier to compute than in the state-dependent case. Since in both cases W and $\widetilde{W}$ must be composed by K, the use of automatic differentiation for step 4 in Algorithm 3.2 is still needed. In particular, for Algorithm 3.1 and the composition via Theorem 4.4, the NFFT3 can be used to perform the numerical composition of K with W and $\widetilde{W}$.

The first steps of our method get ω and λ. These values are summarized respectively in Tables 3 and 4 for the parameter value $\mu = 1.5$. They were computed fixing a tolerance for

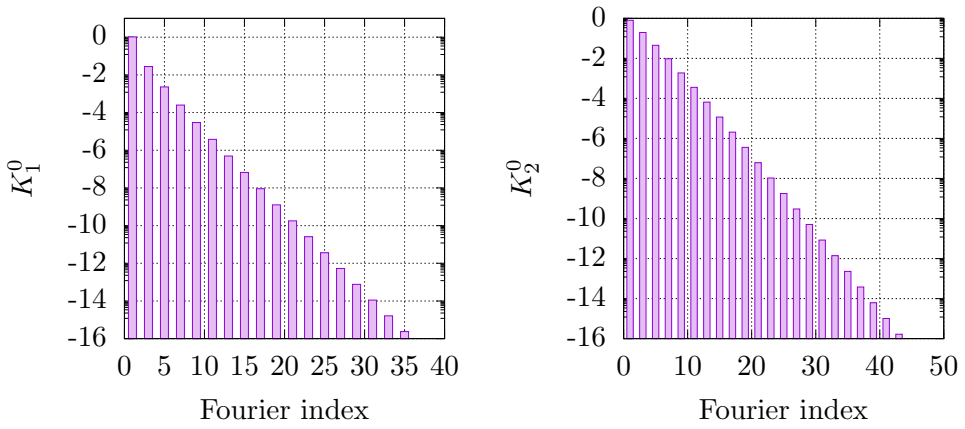

Figure 3. *Logscale of the power spectrum of* $K^0 \equiv (K_1^0, K_2^0)$ *for* $\mu = 1.5$ *and* $\varepsilon = 0$ *in* (5.1).

the stopping criterion of 10^{-10} in double-precision. Because the result is perturbative, these values are close to those in Table 2 but are further apart as ε increases. Moreover, we report a speed factor around 2.25 using the NFFT3 with respect to a direct implementation of the Fourier composition.

Table 3

Values of ω *for different values of* ε *in* (5.1) *with* $\mu = 1.5$ *obtained by Algorithm* 3.1. ω_s *corresponds to the state-dependent delay, and* ω_c *to the constant delay.*

ε	ω_s	ω_c
10^{-4}	0.140908673246532	0.140908547470887
10^{-3}	0.140833302396846	0.140832045466042
10^{-2}	0.140077545298062	0.140065058638519

Table 4

Values of λ *for different values of* ε *in* (5.1) *with* $\mu = 1.5$ *obtained by Algorithm* 3.2. λ_s *corresponds to the state-dependent delay, and* λ_c *to the constant delay.*

ε	λ_s	λ_c
10^{-4}	−1.6838123845562083	−1.6838091880373793
10^{-3}	−1.6839721186835845	−1.6839401491442914
10^{-2}	−1.6855808865357260	−1.6852607528946115

Figure 4 shows, for different values of ε in (5.1), the logarithmic error of the invariance equation for each of the different orders $j \geqslant 0$, that is, the finite system of invariance equations obtained after plugging $W(\theta, s) = \sum W^j(\theta) s^j$ into (2.7) and matching terms of the same order. The state-dependent case needs smaller values of ε to satisfy the invariance equation, while the constant delay case admits larger values of ε which can be deduced from the inequalities in [37].

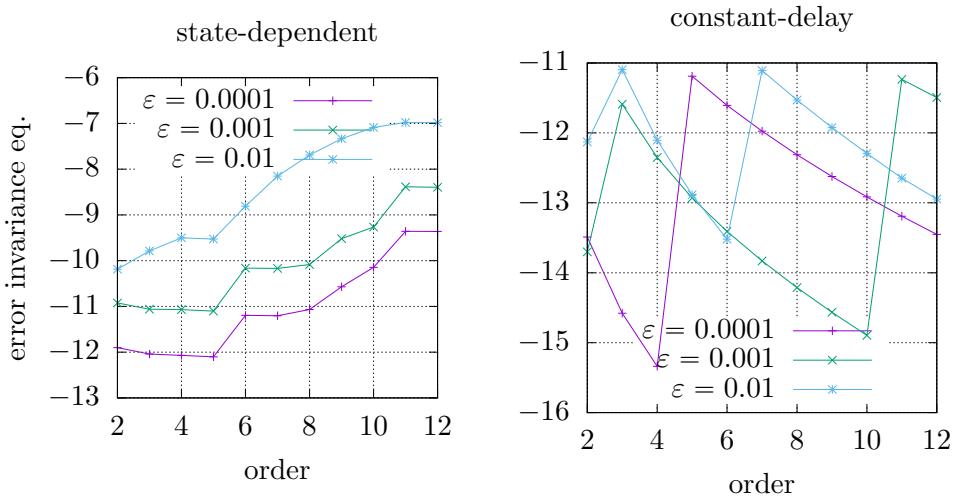

Figure 4. Log10 *scale of the* 2*-norm of the error in the invariance equation.*

Figure 5 shows the difference between the isochrons for the perturbed and unperturbed cases. As one expects from the theorems in [37], the error is smaller as the perturbation parameter value ε becomes smaller.

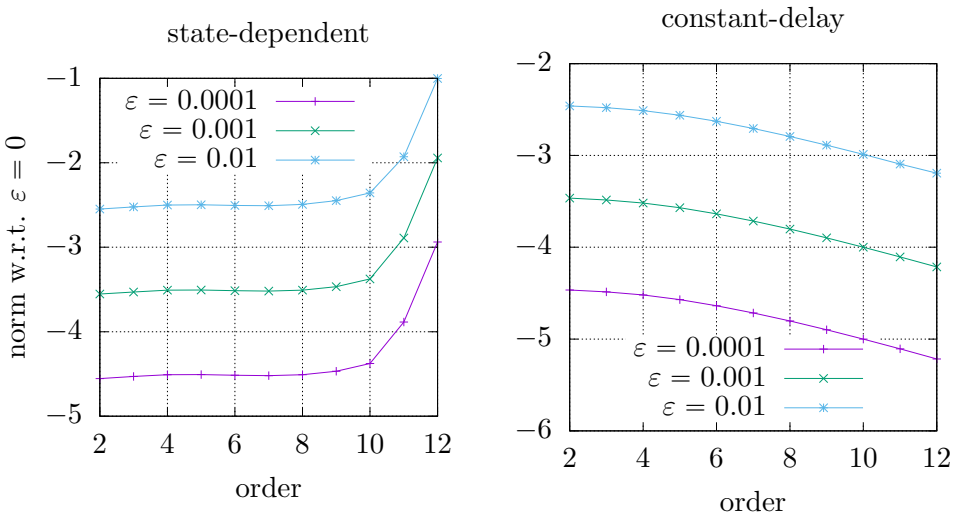

Figure 5. Log10 *scale of the* 2*-norm of the difference between the perturbed and unperturbed cases, that is,* $\|K^j - (K \circ W)^j\|$.

An important point in Algorithm 3.2 is the well-definedness of the composition of K with W and $\widetilde{W}$. Because the state-dependent delays consider many more situations than just the constant delay, one expects that a potentially smaller scaling factor compared to the constant delay will be needed for computation of big orders. Figure 6 shows that if ε is large, the scaling factor will need to be small. We also see that for the constant case, it is enough to

use a constant scaling factor, and for the state-dependent case, the scaling factor decreases drastically in the first orders.

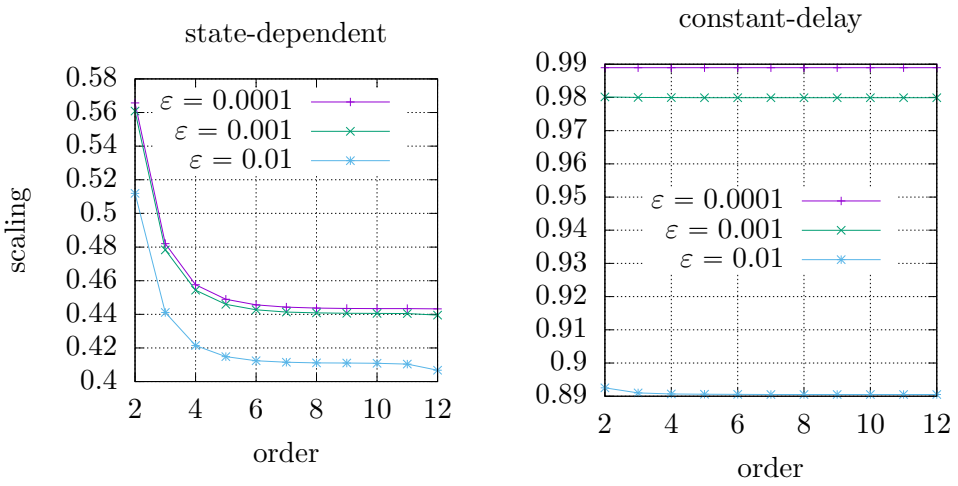

Figure 6. *Scaling factor to ensure that the composition of K with W and with $\widetilde{W}$ in Algorithm 3.2 are well defined.*

To illustrate the physical observation Figures 7 and 8 show the power spectra of the limit cycles after the perturbations. More concretely, Figure 7 displays the power spectrum of $(K \circ W)^0$ for the pure state-dependent delay case and $\varepsilon = 0.01$. In contrast with Figure 3, we observe that for the even indexes these figures have nonzero values in the double-precision arithmetic sense. On the other hand, Figure 8 shows that these nonzero values in the even indexes are not present in the constant delay case. The power spectrum for the case $\varepsilon > 0$ changes from that when $\varepsilon = 0$ as ε increases.

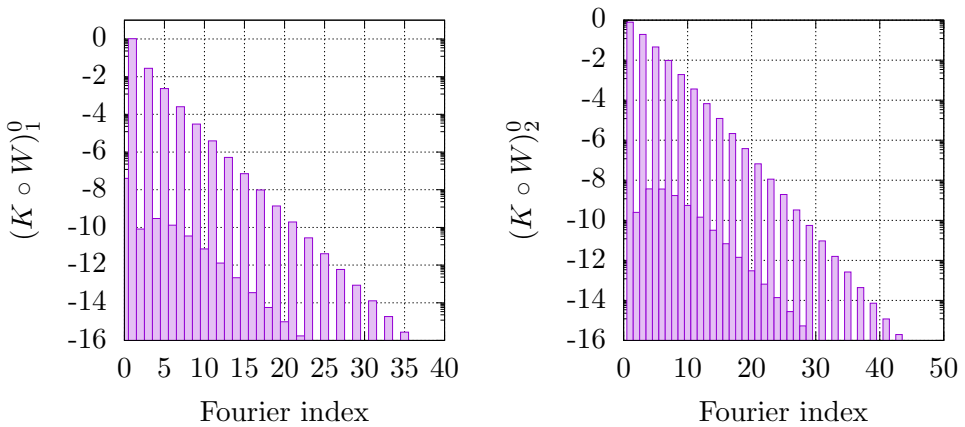

Figure 7. Log10 *scale of the power spectrum of $(K \circ W)^0$ for $\mu = 1.5$, $\varepsilon = 0.01$, and the state-dependent delay $r(x(t)) = 0.006e^{x(t)}$ in (5.1).*

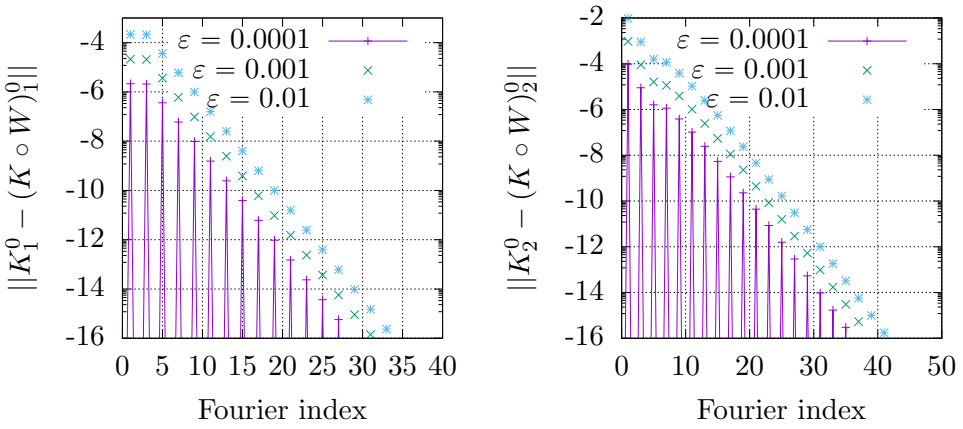

Figure 8. *Log*10 *scale of the difference between the power spectrum of* K^0 *and the power spectrum of* $(K \circ W)^0$ *for* $\mu = 1.5$*, different values of* ε*, and constant delay* $r = 0.006$ *in* (5.1).

6. Conclusions. We have presented Algorithms 3.1 and 3.2 to compute limit cycles and their isochrons for an SDDE coming from perturbing a planar ODE with a limit cycle.

As input of those algorithms we use a parameterization of the limit cycle and its isochrons of the ODE, which acts as a change of variables in a neighborhood of the unperturbed limit cycle.

To fix the solution, we use normalization conditions in the zeroth and first orders of the Taylor–Fourier representation of the solution in the perturbed system.

We have also discussed the numerical composition between Taylor–Fourier expansions which is the hardest step in those algorithms.

Finally, we have shown numerical experiments, and we provide a code sample as supplementary material (M133696_01.zip [local/web 368KB]), allowing one to adapt the code for other examples.

We observed that for the unperturbed case, the scaling factor as well as the mesh size play an important role to get the parameterization. In the perturbed system, the scaling factor also allows us to ensure the well-definedness of the composition, which causes the scaling factor to be smaller, especially for the nonconstant delay expression.

We also showed that the perturbative parameter ε affects the convergence of the algorithm. To compute high orders of W, the ε needs to be smaller in the nonconstant delay. This makes sense with the existence proofs in [37]. They are based on a contraction argument of an operator, which is reproduced in the steps of the algorithms used in this paper. The bounds to ensure the contraction of the operators depend on the norms of the delay function as well as its derivatives. Therefore, if the delay is not constant, those quantities will cause ε to be small enough to ensure the contraction in the operator.

The algorithms we have presented in this paper come from the companion paper [37]. Avoiding hard mathematical notation, there are some implications and consequences of this new technique.

The statements in [37] are formulated for SDDEs (i.e., DDEs whose delays depend on the

state) in an a posteriori format which does not depend on time explicitly. The result is then directly applicable to constant, state-dependent, and even implicitly defined delays.

Although the results therein have been detailed for the case of SDDEs, they also apply without major modifications to advanced or even mixed differential equations. That means that the perturbation added to the ODE can, at a current time, require the information at future or past times.

It is important to stress that our approach does not provide a full parameterization of the foliation in the phase space of the perturbed model. Indeed, we bypass that discussion by just providing a partial dynamics near the limit cycle closest to the dynamics of the unperturbed model. We decided to keep this distinction, calling it isochron (a term often used in biology for the foliation of a limit cycle in an ODE scenario).

The companion paper [37] also includes a theory for different regularities of the differential equation, but in the current numerical paper, we only formulated results for analytic systems. Since we specified which derivatives appear in the calculations, it is clear that the algorithms for the delay-perturbed case also apply for problems with finite differentiability.

The theoretical result can also be the starting point for computer-assisted proofs (CAPs). The study of periodic orbit by CAPs independent of the evolution operator has also been considered in the literature, e.g., in [26], which deals with several delays. We recently learned of the paper [34], which produces CAPs of analytic periodic solutions. In the case of SDDEs considered here, analytic solutions are not expected to exist and the framework in [34] would need to be adjusted; see [16].

Finally, we want to emphasize that our algorithms as well as the algorithm in [23] can be generalized to higher dimensions with a more tedious notation. It will require changes in the numerical composition, and its computational complexities will have a more difficult expression. The good point is that there is still a chance to take advantage of the parallelism with higher dimensional isochrons.

Appendix A. Computation of (K, ω_0, λ_0)—unperturbed case. For completeness, we quote Algorithm 4.4 in [23], adding some practical comments. That algorithm allows us to numerically compute ω_0, λ_0, and $K\colon \tilde{\mathbb{T}} \times [-1,1] \to \mathbb{R}^2$ in (2.7). We note that the algorithm has quadratic convergence as it was proved in [23].

Algorithm A.1. Quasi-Newton method.

- ⋆ `Input:` $\dot{x} = X(x)$ in $\mathbb{R}^2$, $K(\theta, s) = K^0(\theta) + K^1(\theta) b_0 s$, $\omega_0 > 0$, $\lambda_0 \in \mathbb{R}$, scaling factor $b_0 > 0$, and a tolerance *tol*.
- ⋆ `Output:` $K(\theta, s) = \sum_{j=0}^{m-1} K^j(\theta)(b_0 s)^j$, ω_0 and λ_0 such that $\|E\| \ll 1$.

1. $E \leftarrow X \circ K - (\omega_0 \partial_\theta + \lambda_0 s \partial_s) K$.
2. Solve $DK\tilde{E} = E$ and denote $\tilde{E} \equiv (\tilde{E}_1, \tilde{E}_2)$.
3. $\sigma \leftarrow \int_0^1 \tilde{E}_1(\theta, 0)\, d\theta$ and $\eta \leftarrow \int_0^1 \partial_s \tilde{E}_2(\theta, 0)\, d\theta$.
4. $E_1 \leftarrow \tilde{E}_1 - \sigma$ and $E_2 \leftarrow \tilde{E}_2 - \eta s$.

5. Solve $(\omega_0\partial_\theta + \lambda_0 s\partial_s)S_1 = E_1$ imposing

$$\int_0^1 S_1(\theta, 0)\, d\theta = 0. \tag{A.1}$$

6. Solve $(\omega_0\partial_\theta + \lambda_0 s\partial_s)S_2 - \lambda_0 S_2 = E_2$ imposing

$$\int_0^1 \partial_s S_2(\theta, 0)\, d\theta = 0. \tag{A.2}$$

7. $S \equiv (S_1, S_2)$.

8. Update: $K \leftarrow K + DKS$, $\omega_0 \leftarrow \omega_0 + \sigma$, and $\lambda_0 \leftarrow \lambda_0 + \eta$.

9. Iterate from step 1 to step 8 until convergence with tolerance *tol* in K, ω, and λ. Then undo the scaling b_0.

Algorithm A.1 requires some practical considerations:

i. *Initial guess.* $K^0\colon \tilde{\mathbb{T}} \to \mathbb{R}^2$ will be a parameterization of the periodic orbit of the ODE with frequency ω_0. It can be obtained, for instance, by a Poincaré section method, continuation of integrable systems, or Lindstedt series. An approximation for $K^1\colon \tilde{\mathbb{T}} \to \mathbb{R}^2$ and λ can be obtained by solving the variational equation

$$\begin{aligned} DX \circ K^0(\theta)U(\theta) &= \omega_0 \frac{d}{d\theta}U(\theta), \\ U(0) &= Id_2. \end{aligned}$$

Hence if $(e^{\lambda_0/\omega_0}, K^1(0))$ is the eigenpair of $U(1)$ such that $\lambda_0 < 0$, then $K^1(\theta) = U(\theta)K^1(0)e^{-\lambda_0\theta/\omega_0}$.

ii. *Stopping criteria.* As with any Newton method, a possible condition to stop the iteration can be when either $\|E\|$ or $\max\{\|DKS\|, |\sigma|, |\eta|\}$ is smaller than a given tolerance.
Note that the a posteriori theorems in [23] give a criterion of smallness on the error depending on properties of the function K. If these criteria are satisfied, one can ensure that there is a true solution close to the numerical one.

iii. *Uniqueness.* Note that in steps 5 and 6, which involve solving the cohomology equations, the solutions are determined only up to adding constants in the zeroth or first order terms. We have adopted the conventions (A.1), (A.2). These conventions make the solution operator linear (which matches well the standard theory of Nash–Moser methods since it is easy to estimate the norm of the solutions).
As it is shown in [23], the algorithm converges quadratically fast to a solution, but since the problem is underdetermined, we have to be careful when comparing solutions of different discretization. In [23] there is discussion of the uniqueness, but for our purposes in this paper, any of the solutions will work. The uniqueness of the solutions considered in this paper is discussed in section 2.8.

iv. *Convergence.* Even though Algorithm A.1 is a quasi-Newton method, [23] proved that it still has quadratic convergence.

Note that it is remarkable that we can implement a Newton-like method without having to store—much less invert—any large matrix. Note also that we can get a Newton method even if the derivative of the operator in the fixed point equation has eigenvalues 1. See Remark 2.3.

v. *Cohomological equations.* The most delicate steps of the above algorithm are steps 5 and 6, which are often called *cohomology equations*. These steps involve solving PDEs, whereas the other steps are much simpler. In the case of a Fourier representation (see Appendix B), the cohomological equations can be addressed by using Proposition 3.3.

vi. *Linear system.* Step 2 can be addressed by Lemma 3.4.

Appendix B. Fourier discretization of periodic functions. As mentioned before, the key part of Algorithm A.1 is to solve the equations in steps 5 and 6. Their numerical resolution will be particularly efficient when the functions are discretized in Fourier–Taylor series. This is the only discretization considered in this paper for which we provide a deep discussion.

Recall that a function $S\colon \mathbb{R} \to \mathbb{R}$ is called periodic when $S(\theta+1) = S(\theta)$ for all θ.

To get a computer representation of a periodic function, we can either take a mesh in θ, i.e., $(\theta_k)_{k=0}^{n_\theta-1}$, and store the values of S at the points $\check{S} = (\check{S}_k)_{k=0}^{n_\theta-1} \in \mathbb{R}^{n_\theta}$ with $\check{S}_k = S(\theta_k)$, or we can take advantage of the periodicity and represent it in a trigonometric basis.

The discrete Fourier transform (DFT), and also its inverse, allows us to switch between the two representations above. If we fix a mesh of points of size n_θ uniformly distributed in $[0,1)$, i.e., $\theta_k = k/n_\theta$, the DFT is

$$\widehat{S} = (\widehat{S}_k)_{k=0}^{n_\theta-1} \in \mathbb{C}^{n_\theta}$$

so that

$$\check{S}_k = \sum_{j=0}^{n_\theta-1} \widehat{S}_j e^{2\pi \boldsymbol{i} jk/n_\theta}, \tag{B.1}$$

or equivalently,

$$\widehat{S}_k = \frac{1}{n_\theta} \sum_{j=0}^{n_\theta-1} \check{S}_j e^{-2\pi \boldsymbol{i} jk/n_\theta}. \tag{B.2}$$

In the case of a real valued function, $\widehat{S}_0$ is real and the complex numbers $\widehat{S}$ satisfy Hermitian symmetry, i.e., $\widehat{S}_k = \widehat{S}^*_{n_\theta-k}$ (denoting by * the complex conjugate), which implies $\widehat{S}_{n_\theta/2}$ is real when n_θ is even. Then, we define real numbers $(a_0; a_k, b_k)_{k=1}^{\lceil n_\theta/2\rceil-1}$ if n_θ is odd (here $\lceil\cdot\rceil$ denotes the ceil function); otherwise $(a_0, a_{n_\theta/2}; a_k, b_k)_{k=1}^{n_\theta/2-1}$ are defined by

$$a_0 = 2\widehat{S}_0,\ a_{n_\theta/2} = 2\widehat{S}_{n_\theta/2},\ a_k = 2\operatorname{Re}\widehat{S}_k, \text{ and } b_k = -2\operatorname{Im}\widehat{S}_k$$

with $1 \leqslant k < \lceil n_\theta/2\rceil$.

Thus, S can be approximated by

$$S(\theta) = \frac{a_0}{2} + \frac{a_{n_\theta/2}}{2}\cos(\pi n_\theta \theta) + \sum_{k=1}^{\lceil n_\theta/2\rceil-1} a_k \cos(2\pi k\theta) + b_k \sin(2\pi k\theta), \tag{B.3}$$

where the coefficient $a_{n_\theta/2}$ only appears when n_θ is even, and it refers to the aliasing notion in signal theory.

Therefore (B.3) is equivalent to (B.1), but rather than $2n_\theta$ real numbers, only half of them are needed.

Henceforth, all real periodic functions S can be represented in a computer by an array of length n_θ whose values are either the values of S on a grid or the Fourier coefficients. These two representations are for all practical purposes equivalent since there is a well-known algorithm FFT which allows us to go from one to the other in $\Theta(n_\theta \log n_\theta)$ operations. The FFT has very efficient implementations so that the theoretical estimates on time are realistic (we can use FFTW3 [15], which optimizes the use of the hardware).

We can also think of functions of two variables $W(\theta, s)$ where one variable θ is periodic and the other variable s is a real variable. In the numerical implementations, the variable s will be discretized as a polynomial. Thus $W(\theta, s)$ can be thought as a function of θ taking values in polynomials of length n_s. Hence, a function of two variables with periodicity as above will be discretized by an array $n_\theta \times n_s$. The meaning could be that it is a polynomial for each value of θ in a mesh or that it a polynomial whose coefficients are Fourier coefficients. Alternatively, we could think of $W(\theta, s)$ as a polynomial in s taking values in a space of periodic functions.

This mixed representation of Fourier series in one variable and of power series in another variable is often called Fourier–Taylor series and has been used in celestial mechanics for a long time, dating back to [7] or earlier. We note that modern computer languages allow one to overload the arithmetic operations among different types in a simple way.

It is important to note that all the operations in Algorithm A.1 are fast either in the Fourier representation or in the values of a mesh representation. For example, the product of two functions or the composition on the left with a known function is fast in the representation by values in a mesh. More importantly for us, as we will see, the solution of cohomology equations is fast in the Fourier representation. On the other hand, there are other steps of Algorithm A.1, such as adding, that are fast in both representations.

Similar consideration of the efficiency of the steps will apply to the algorithms needed to solve our problem. The main novelty of the algorithms in this paper compared with those of [23] is that we will need to compose some of the unknown functions (in [23] the unknowns are only composed on the left with a known function). The algorithms we use to deal with composition were presented in section 4. The composition operator was the most delicate numerical aspect, which was to be expected, since it was also the most delicate step in the analysis in [37]. The composition operator is analytically subtle. A study which gives examples that results are sharp is in [12]. See also [2].

Remark B.1. Fourier series are extremely efficient for smooth functions which do not have very pronounced spikes. For rather smooth functions—a situation that appears often in practice—it seems that Fourier–Taylor series is better than other methods.

It should be noted, however, that in several models of interest in electronics and neuroscience, the solutions move slowly during a large fraction of the period, but there is a fast movement for a short time (bursting). In these situations, the Fourier scheme has the disadvantage that the coefficients decrease slowly and that the discretization method does not allow for putting more effort into describing the solutions during the times when they are

indeed changing fast. Hence, the Fourier methods become unpractical when the limit cycles are bursting. In such cases, one can use other methods of discretization. In this paper, we will not discuss alternative numerical methods, but we note that the theoretical estimates of [37] remain valid independent of the method of discretization. We hope to come back to implementing the toolkit of operations of this paper in other discretizations.

Remark B.2. One of the confusing practical aspects of the actual implementation is that the coefficients of the Fourier arrays are often stored in a complicated order to optimize the operations and the access during the FFT.

For example, concerning the coefficients a_k's and b_k's in (B.3), in FFTW3, the function `fftw_plan_r2r_1d` uses the following order of the Fourier coefficients in a real array $(v_0, \dots, v_{n_\theta - 1})$:

$$\begin{aligned} v_0 &= a_0, \\ v_k &= 2a_k \text{ and } v_{n_\theta - k} = -2b_k \quad \text{for } 1 \leqslant k < \lceil n_\theta/2 \rceil, \\ v_{n_\theta/2} &= a_{n_\theta/2}, \end{aligned}$$

where the index $n_\theta/2$ is taken into consideration if and only if n_θ is even. Another standard order in other packages is just $(a_0, a_{n_\theta/2}; a_k, b_k)$ in sequential order or $(a_0; a_k, b_k)$ if n_θ is odd.

To measure errors and size of functions represented by Fourier series, we have found it useful to deal with weighted norms involving the Fourier coefficients:

$$\begin{aligned} \|S\|_{w\ell^1,n} &= 2(n_\theta/2)^n |\widehat{S}_{n_\theta/2}| + \sum_{k=1}^{\lceil n_\theta/2 \rceil - 1} ((n_\theta - k)^n + k^n)|\widehat{S}_k| \\ &= (n_\theta/2)^n |a_{n_\theta/2}| + \frac{1}{2} \sum_{k=1}^{\lceil n_\theta/2 \rceil - 1} ((n_\theta - k)^n + k^n)(a_k^2 + b_k^2)^{1/2}, \end{aligned}$$

where, again, the term for $n_\theta/2$ only appears if n_θ is even.

The smoothness of S can be measured by the speed of decay of the Fourier coefficients, and indeed, the above norms give useful regularity classes that have been studied by harmonic analysts.

Remark B.3. The relation of the above regularity classes with the most common C^m is not straightforward, as is well known by Harmonic analysts, [33].

The Riemann–Lebesgue lemma tells us that if S is continuous and periodic, $\widehat{S}_k \to 0$ as $k \to \infty$, and in general if S is m times differentiable, then $|\widehat{S}_k||k|^m$ tends to zero. In particular, $|\widehat{S}_k| \leqslant C/|k|^m$ for some constant $C > 0$.

In the other direction, from $|\widehat{S}_k| \leqslant C/|k|^m$ we cannot deduce that $S \in C^m$.

One has to use more complicated methods. In [13] it was found that one could find a practical method based on the Littlewood–Paley theorem (see [33]) which states that the function S is in α-Hölder space with $\alpha \in \mathbb{R}_+$ if and only if for each $\eta \geqslant 0$ there is a constant $C > 0$ such that for all $t > 0$,

$$\left\| \left(\frac{\partial}{\partial t} \right)^\eta e^{-t\sqrt{-\Delta_\theta}} \right\|_{L^\infty(\mathbb{T})} \leqslant C t^{\alpha - \eta}.$$

The above formula is easy to implement if one has the Fourier coefficients, as is the case in our algorithms.

Acknowledgments. JG thanks the School of Mathematics, Georgia Institute of Technology, for the hospitality received in Spring 2018, Spring 2019, and Fall 2019. The authors thank the editor and referees for their discussions and the suggested improvements in the paper.

REFERENCES

[1] A. A. Andronov, A. A. Vitt, and S. È. Khaĭkin, *Theory of Oscillators*, translated from the Russian by F. Immirzi, reprint of the 1966 translation, Dover Publications, New York, 1987.

[2] J. Appell and P. P. Zabrejko, *Nonlinear Superposition Operators*, Cambridge Tracts in Math. 95, Cambridge University Press, Cambridge, 1990, https://doi.org/10.1017/CBO9780511897450.

[3] T. Arecchi, G. Giacomelli, A. Lapucci, and R. Meucci, *Dynamics of a CO_2 laser with delayed feedback: The short-delay regime*, Phys. Rev. A, 43 (1991), pp. 4997–5004, https://doi.org/10.1103/PhysRevA.43.4997.

[4] A. H. Barnett, *Aliasing Error of the* $\exp(\beta\sqrt{1-z^2})$ *Kernel in the Nonuniform Fast Fourier Transform*, preprint, https://arxiv.org/abs/2001.09405, 2020.

[5] A. H. Barnett, J. F. Magland, and L. af Klinteberg, *A Parallel Non-uniform Fast Fourier Transform Library Based on an "Exponential of Semicircle" Kernel*, preprint, https://arxiv.org/abs/1808.06736, 2019.

[6] W.-J. Beyn and W. Kleß, *Numerical Taylor expansions of invariant manifolds in large dynamical systems*, Numer. Math., 80 (1998), pp. 1–38, https://doi.org/10.1007/s002110050357.

[7] R. Broucke and K. Garthwhite, *A programming system for analytical series expansions on a computer*, Celestial Mechanics, 1 (1969), pp. 271–284, https://doi.org/10.1007/BF01228844.

[8] M. J. Capiński and P. Roldán, *Existence of a center manifold in a practical domain around L_1 in the restricted three-body problem*, SIAM J. Appl. Dyn. Syst., 11 (2012), pp. 285–318, https://doi.org/10.1137/100810381.

[9] J. Carr, *Applications of Centre Manifold Theory*, Applied Mathematical Sciences 35, Springer-Verlag, New York, Berlin, 1981.

[10] A. Casal, L. Corsi, and R. de la Llave, *Expansions in the delay of quasi-periodic solutions for state dependent delay equations*, J. Phys. A, 53 (2020), pp. 235202, https://doi.org/10.1088/1751-8121/ab7b9e.

[11] C. Chicone and W. Liu, *Asymptotic phase revisited*, J. Differential Equations, 204 (2004), pp. 227–246, https://doi.org/10.1016/j.jde.2004.03.011.

[12] R. de la Llave and R. Obaya, *Regularity of the composition operator in spaces of Hölder functions*, Discrete Contin. Dynam. Systems, 5 (1999), pp. 157–184, https://doi.org/10.3934/dcds.1999.5.157.

[13] R. de la Llave and N. P. Petrov, *Regularity of conjugacies between critical circle maps: An experimental study*, Experiment. Math., 11 (2002), pp. 219–241.

[14] A. Di Garbo, S. Euzzor, J.-M. Ginoux, F. T. Arecchi, and R. Meucci, *Delayed dynamics in an electronic relaxation oscillator*, Phys. Rev. E, 100 (2019), 032224, https://doi.org/10.1103/PhysRevE.100.032224.

[15] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, Proc. IEEE, 93 (2005), pp. 216–231.

[16] J. Gimeno, J.-P. Lessard, J. Mireles-James, and J. Yang, *Computer Assisted Proofs for Persistence of Periodic Orbits in Functional Differential Equations Close to ODE*, work in progress, 2021.

[17] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed., SIAM, Philadelphia, 2008, https://doi.org/10.1137/1.9780898717761.

[18] J. Guckenheimer, *Isochrons and phaseless sets*, J. Math. Biol., 1 (1975), pp. 259–273, https://doi.org/10.1007/BF01273747.

[19] J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*, Appl. Math. Sci. 99, Springer-Verlag, New York, 1993, https://doi.org/10.1007/978-1-4612-4342-7.

[20] À. Haro, M. Canadell, J.-L. Figueras, A. Luque, and J.-M. Mondelo, *The Parameterization Method for Invariant Manifolds. From Rigorous Results to Effective Computations*, Appl. Math. Sci. 195, Springer, Cham, 2016, https://doi.org/10.1007/978-3-319-29662-3.

[21] F. Hartung, T. Krisztin, H.-O. Walther, and J. Wu, *Functional differential equations with state-dependent delays: Theory and applications*, in Handbook of Differential Equations: Ordinary Differential Equations. Vol. III, Handb. Differ. Equ., Elsevier/North-Holland, Amsterdam, 2006, pp. 435–545, https://doi.org/10.1016/S1874-5725(06)80009-X.

[22] A. Hou and S. Guo, *Stability and Hopf bifurcation in van der Pol oscillators with state-dependent delayed feedback*, Nonlinear Dynam., 79 (2015), pp. 2407–2419, https://doi.org/10.1007/s11071-014-1821-3.

[23] G. Huguet and R. de la Llave, *Computation of limit cycles and their isochrons: Fast algorithms and their convergence*, SIAM J. Appl. Dyn. Syst., 12 (2013), pp. 1763–1802, https://doi.org/10.1137/120901210.

[24] À. Jorba, *A methodology for the numerical computation of normal forms, centre manifolds and first integrals of Hamiltonian systems*, Experiment. Math., 8 (1999), pp. 155–195.

[25] J. Keiner, S. Kunis, and D. Potts, *Using NFFT 3—a software library for various nonequispaced fast Fourier transforms*, ACM Trans. Math. Software, 36 (2009), 19, https://doi.org/10.1145/1555386.1555388.

[26] G. Kiss and J.-P. Lessard, *Computational fixed-point theory for differential delay equations with multiple time lags*, J. Differential Equations, 252 (2012), pp. 3093–3115, https://doi.org/10.1016/j.jde.2011.11.020.

[27] D. E. Knuth, *The Art of Computer Programming. Vol. 2. Seminumerical Algorithms*, 2nd ed., Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, Reading, MA, 1981.

[28] T. Krisztin, *A local unstable manifold for differential equations with state-dependent delay*, Discrete Contin. Dyn. Syst., 9 (2003), pp. 993–1028, https://doi.org/10.3934/dcds.2003.9.993.

[29] O. E. Lanford, III, *Bifurcation of periodic solutions into invariant tori: The work of Ruelle and Takens*, in Nonlinear Problems in the Physical Sciences and Biology: Proceedings of a Battelle Summer Institute, Lecture Notes in Math. 322, I. Stakgold, D. D. Joseph, and D. H. Sattinger, eds., Springer-Verlag, Berlin, 1973, pp. 159–192.

[30] N. Minorsky, *Nonlinear Oscillations*, D. Van Nostrand, Princeton, Toronto, London, New York, 1962.

[31] C. Pötzsche and M. Rasmussen, *Taylor approximation of integral manifolds*, J. Dynam. Differential Equations, 18 (2006), pp. 427–460, https://doi.org/10.1007/s10884-006-9011-8.

[32] J. Sijbrand, *Properties of center manifolds*, Trans. Amer. Math. Soc., 289 (1985), pp. 431–469, https://doi.org/10.2307/2000247.

[33] E. M. Stein, *Singular Integrals and Differentiability Properties of Functions*, Princeton Mathematical Series No. 30, Princeton University Press, Princeton, NJ, 1970.

[34] J. B. van den Berg, C. Groothede, and J.-P. Lessard, *A general method for computer-assisted proofs of periodic solutions in delay differential problems*, J. Dynam. Differential Equations, (2020), https://doi.org/10.1007/s10884-020-09908-6.

[35] B. van der Pol, *A theory of the amplitude of free and forced triode vibrations*, Radio Review, 1 (1920), pp. 701–710, 754–762.

[36] A. T. Winfree, *Patterns of phase compromise in biological cycles*, J. Math. Biol., 1 (1974/75), pp. 73–95, https://doi.org/10.1007/BF02339491.

[37] J. Yang, J. Gimeno, and R. de la Llave, *Parameterization method for state-dependent delay perturbation of an ordinary differential equation*, SIAM J. Math. Anal., 53 (2021), pp. 4031–4067, https://doi.org/10.1137/20M1311430.