NUMERICAL CONTINUATION AND DELAY EQUATIONS: A NOVEL APPROACH FOR COMPLEX MODELS OF STRUCTURED POPULATIONS

Alessia Andò

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

Dimitri Breda*

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

Francesca Scarabel

Department of Mathematics and Statistics – University of Helsinki
P.O. Box 68 (Pietari Kalmin katu 5) FI-00014 Helsinki, Finland
and
CDLab – Computational Dynamics Laboratory

(Communicated by the associate editor name)

Abstract. Recently, many realistic models of structured populations are described through delay equations which involve quantities defined by the solutions of *external* problems. For instance, the size or survival probability of individuals may be described by ordinary differential equations, and their maturation age may be determined by a nonlinear condition. When treating these complex models with existing continuation approaches in view of analyzing stability and bifurcations, the external quantities are computed from scratch at every continuation step. As a result, the requirements from the computational point of view are often demanding. In this work we propose to improve the overall performance by investigating a suitable numerical treatment of the external problems in order to include the relevant variables into the continuation framework, thus exploiting their values computed at each previous step. We explore and test this *internal* continuation with prototype problems first. Then we apply it to a representative class of realistic models, demonstrating the superiority of the new approach in terms of computational time for a given accuracy threshold.

1. **Introduction.** In the recent literature delay equations are used to describe the evolution of structured populations [5, 11, 13, 17, 26, 28, 33]. In particular, systems of Renewal Equations (REs) coupled to Delay Differential Equations (DDEs) are employed to model consumer-resource dynamics [12, 18]. Besides the intrinsic infinite dimension due to the presence of delays, an increase of complexity in realistic applications follows from the fact that several model ingredients are not given explicitly, but rather as solutions of related *external* problems. For instance, this is the case of Ordinary Differential Equations (ODEs) whose solutions provide the evolution of some structuring variables as, e.g., the size of the individuals or their survival probability, as well as the case of nonlinear constraints which implicitly define the subdivision of the population into different classes as, e.g., juveniles and adults, making the original model state-dependent (see, e.g., [23]).

The resolution of these external problems is often the computational bottleneck when continuation techniques are applied to compute equilibria or periodic solutions, analyze their stability and detect relevant bifurcations, which are all common targets. This is the case of the approach proposed very recently in [6]. Its underlying idea is to reduce the original model to a system of ODEs, but when it comes to apply standard continuation tools for ODEs to such system the role of the external problems in determining the computational cost emerges rather clearly. Nevertheless, to the best of our knowledge, this approach is very promising, especially due to its wide applicability, and sometimes it represents the only reasonable chance to tackle the dynamical analysis of complex yet realistic models.

To overcome the aforementioned bottleneck we propose to include the resolution of the external problems into the continuation framework, calling this general strategy *internal* continuation in the sequel. Indeed, this approach eliminates the need to compute the external variables from scratch at every continuation step and hence it avoids to resort to external (yet efficient) solvers (for ODEs, for nonlinear equations, etc.). This is because the computation of these quantities at the current continuation step benefits of the information already acquired at the previous step. Of course, the above inclusion is not for free, and a careful numerical treatment of the external problems has to be considered.

In order to investigate the validity of our proposal, we focus on a class of realistic models, suitably represented by the one commonly called *Daphnia consuming algae* (for simplicity, *Daphnia* in the sequel). We describe *Daphnia* in Section 3, highlighting the external problems and the technicalities involved with it. In this work we concentrate on the continuation of its equilibria as a starting point, thus the rest of Section 3 focuses on how to approach this problem, either with the general method proposed in [6] or with more specific tools as proposed in [32]. In Section 4 we explain our internal approach in full detail. To this aim, we introduce a series of prototype models in order to simplify the original target and distinguish the role of all the technicalities in determining the final computational cost. The implementation of the standard external approach that is used for comparison is summarized in Section 5. We show the obtained numerical results in Section 6, first on the prototype problems and then on the realistic *Daphnia* model. In particular, through the latter results we demonstrate the superiority of the newly proposed approach. Some remarks close the manuscript in Section 7.

Let us remark that some work on a first use of the internal continuation on prototype problems has already been made by (some of) the authors in [4]. In particular, the current research completes the very preliminary investigation of the basic and state-dependent cases illustrated in [4] with the analysis of also the cases of external ODEs with discontinuous right-hand side and of systems of external ODEs. Moreover, and above all, it extends and applies the proposed framework to the true original target of realistic complex models.

Prior to start, we illustrate in the next section a summary of the basic aspects of numerical continuation, which we retain necessary to facilitate the overall understanding of the working context.

2. **Numerical continuation.** We summarize the key aspects of *pseudo-arclength continuation*, in order to establish the general framework in which our class of problems falls. A starting reference on the subject is [21], which inspires most of the content of this section.

Let $G : \mathbb{R}^{n+1} \to \mathbb{R}^n$ be sufficiently smooth. We are interested in solving

$$G(v) = 0. \tag{1}$$

We call $\bar{v}$ a *regular* solution if

$$G_v(\bar{v}) := \begin{pmatrix} \dfrac{\partial G_1}{\partial v_1}(\bar{v}) & \cdots & \dfrac{\partial G_1}{\partial v_{n+1}}(\bar{v}) \\ \vdots & \ddots & \vdots \\ \dfrac{\partial G_n}{\partial v_1}(\bar{v}) & \cdots & \dfrac{\partial G_n}{\partial v_{n+1}}(\bar{v}) \end{pmatrix} \in \mathbb{R}^{n \times (n+1)}$$

has maximal rank. If this is the case, then the implicit function theorem guarantees that near $\bar{v}$ there exists a unique one-dimensional continuum of solutions $v(s)$, parameterized by $s \in \mathbb{R}$, such that $s \mapsto v(s)$ is as smooth as G and $v(0) = \bar{v}$. Above s plays the role of *continuation parameter* and $v(s)$ is called a *solution branch*. Different choices of s may be possible.

The aim of numerical continuation is to approximate a solution branch $v(s)$ through a sequence of points $\{v_j\}_{j \geq 0}$ constructed by an iterated *prediction-correction* procedure as illustrated next for the initial step.

Assume to know a point v_0 on the concerned solution branch, together with the relevant tangent vector $\dot{v}_0$, normalized as $\|\dot{v}_0\|_2 = 1$, Figure 1 (top-left). Given a step of length Δs along this vector, a new point v_1 is found as the solution of the nonlinear determined system

$$\begin{cases} G(v_1) = 0 \\ (v_1 - v_0)^T \dot{v}_0 - \Delta s = 0. \end{cases} \tag{2}$$

The first n equations are obvious. The latter one, called the *pseudo-arclength condition*, is derived from the geometric interpretation of the scalar product between the vectors $v_1 - v_0$ and $\dot{v}_0$ as the length of the projection of the former into the direction of the latter (which is a unit vector).

The solution of system (2) is in general approximated by Newton's method

$$\begin{pmatrix} G_v(v_1^{(k)}) \\ \dot{v}_0^T \end{pmatrix} \Delta v_1^{(k)} = - \begin{pmatrix} G(v_1^{(k)}) \\ (v_1^{(k)} - v_0)^T \dot{v}_0 - \Delta s \end{pmatrix}, \quad k \geq 0, \tag{3}$$
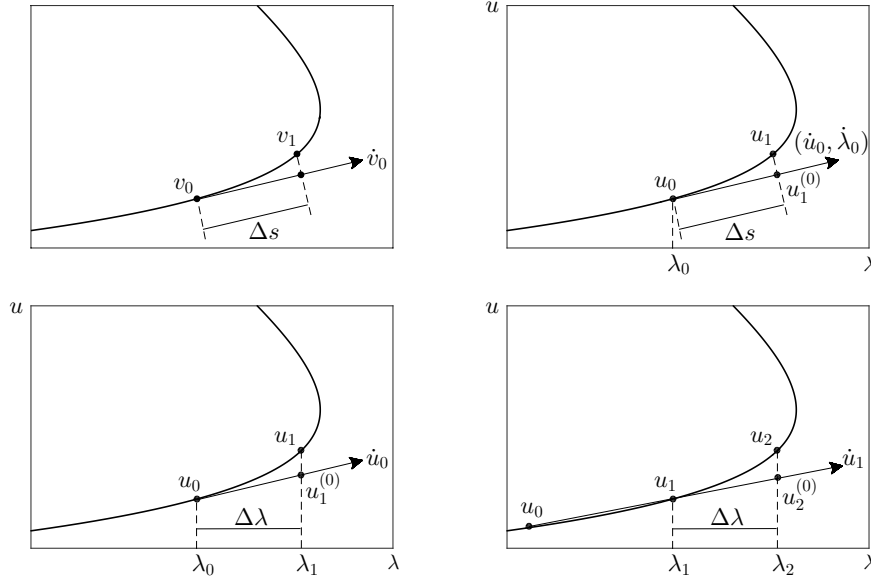
FIGURE 1. Pseudo-arclength continuation (top-left) with natural parameterization (top-right) and natural continuation with tangent (bottom-left) and secant (bottom-right) prediction.

with initial guess

$$v_1^{(0)} = v_0 + \Delta s \; \dot{v}_0 \tag{4}$$

and updates

$$v_1^{(k+1)} = v_1^{(k)} + \Delta v_1^{(k)}, \quad k \geq 0.$$

Other variants can be used to avoid the repeated calculation of the Jacobian matrix in (3). A common one in this context is the Broyden's update [8]. In any case, (4) plays the role of prediction step, while the Newton's method (3) plays the role of correction step.

Once v_1 is computed, the new tangent vector is found as

$$\begin{pmatrix} G_v(v_1) \\ \dot{v}_0^T \end{pmatrix} \dot{v}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{5}$$

with further attention to normalize the result as $\|\dot{v}_1\|_2 = 1$. Note that the equation $\dot{v}_0^T \dot{v}_1 = 1$ preserves the direction of the continuation along the branch. Moreover, observe that the matrix at the left-hand side of (5) is exactly the Jacobian of the Newton's method (3), which means that the new tangent vector can be computed by just one extra back-substitution.

It can be proved that the latter Jacobian is nonsingular at regular points. Yet the step Δs must be chosen appropriately: not too small to avoid unnecessary work, not too large to prevent convergence, or convergence to the right branch. Usually the selection of the next step is based on the convergence behavior of the Newton's method at the previous step. A first threshold is fixed on the number of iterations, say $\overline{k}$. If the number of Newton's iterations overcomes this value, the method is declared not to converge and the prediction step (4) is repeated with a

reduced length (typically halved). Otherwise the step is accepted and the length of the next prediction must be chosen. This length is left unchanged if the number of performed iterations is above a second threshold, say $\underline{k}$ ($< \overline{k}$). Otherwise, the length is increased, multiplying the current one by $c > 1$. Typical heuristic choices may be $\underline{k} = 3$, $\overline{k} = 20$, $c = 1.3$. For more detailed accounts see [3, 24].

The pseudo-arclength continuation is independent of the chosen parameterization. Often the original problem (1) is described with an explicit parameter $\lambda \in \mathbb{R}$ and hence it is rather formulated as

$$G(u, \lambda) = 0 \tag{6}$$

for $G : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and $u \in \mathbb{R}^n$ the true unknown vector. In this case in (1) we set $v = (u, \lambda)$, we talk about *natural parameterization* and we look for the solution branch $(u(\lambda), \lambda)$ or, basically, for $u(\lambda)$. Then (2) becomes

$$\begin{cases} G(u_1, \lambda_1) = 0 \\ (u_1 - u_0)^T \dot{u}_0 + (\lambda_1 - \lambda_0)\dot{\lambda}_0 - \Delta s = 0, \end{cases}$$

Figure 1 (top-right). The k-th step of the corresponding Newton's method, $k \geq 0$, reads

$$\begin{pmatrix} G_u(u_1^{(k)}, \lambda_1^{(k)}) & G_\lambda(u_1^{(k)}, \lambda_1^{(k)}) \\ \dot{u}_0^T & \dot{\lambda}_0 \end{pmatrix} \begin{pmatrix} \Delta u_1^{(k)} \\ \Delta \lambda_1^{(k)} \end{pmatrix} =$$
$$-\begin{pmatrix} G(u_1^{(k)}, \lambda_1^{(k)}) \\ (u_1^{(k)} - u_0)^T \dot{u}_0 + (\lambda_1^{(k)} - \lambda_0)\dot{\lambda}_0 - \Delta s \end{pmatrix},$$

where the notations G_u and G_λ should be clear. The relevant updates are

$$\begin{pmatrix} u_1^{(k+1)} \\ \lambda_1^{(k+1)} \end{pmatrix} = \begin{pmatrix} u_1^{(k)} \\ \lambda_1^{(k)} \end{pmatrix} + \begin{pmatrix} \Delta u_1^{(k+1)} \\ \Delta \lambda_1^{(k+1)} \end{pmatrix}, \quad k \geq 0,$$

the initial prediction is

$$\begin{pmatrix} u_1^{(0)} \\ \lambda_1^{(0)} \end{pmatrix} = \begin{pmatrix} u_0 \\ \lambda_0 \end{pmatrix} + \Delta s \begin{pmatrix} \dot{u}_0 \\ \dot{\lambda}_0 \end{pmatrix}$$

and the new tangent vector is eventually obtained as

$$\begin{pmatrix} G_u(u_1, \lambda_1) & G_\lambda(u_1, \lambda_1) \\ \dot{u}_0^T & \dot{\lambda}_0 \end{pmatrix} \begin{pmatrix} \dot{u}_1 \\ \dot{\lambda}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

by recalling to set then $\|\dot{u}_1\|_2^2 + \dot{\lambda}_1^2 = 1$.

The above general description corresponds to the standard implementation as included in celebrated continuation software like `AUTO` [1, 20, 21] and `MatCont` [2, 15, 16]. Also other continuation packages can be considered, see, e.g., `ALCON` [14], `HOMEPACK` [36] and `PITCON` [31].

A common alternative is represented by the so-called *natural continuation*, in which one predicts a new point along the tangent direction by choosing a step of length $\Delta\lambda$ along the direction of the continuation parameter, and then performs the correction by iterating along the direction orthogonal to the latter, Figure 1 (bottom-left). Although this method may fail in the presence of folds of the solution branch, it is slightly simpler to implement. A even simpler version is obtained by

substituting the prediction step along the tangent vector with that along the secant through the two preceding steps, Figure 1 (bottom-right).

3. **Target model and related continuation.** First we illustrate the *Daphnia* model of resource-consumer dynamics, a representative of the class of problems we have in mind as a target of the current research. For a reference literature see [11, 12, 13, 17, 18]. The following description is mainly inspired by [7].

At the end we report on some computational results on the numerical continuation of its equilibria, obtained by using the technique recently proposed for general REs/DDEs [6], as well as by emulating algorithms already available for *Daphnia*-like models [32] (see also [10, 12]).

We denote by $S(t)$ the (non-negative) available resource concentration at time t, whose evolution, in the absence of consumers, is given by the ODE Cauchy or Initial Value Problem (IVP)

$$\begin{cases} S'(t) = f(S(t)), & t \geq 0, \\ S(0) = S_0 \end{cases} \tag{7}$$

for given $f : [0, +\infty) \to \mathbb{R}$ sufficiently smooth and $S_0 > 0$. As commonly used in the theory of delay equations [19, 25], the function $S_t : [-a_{\max}, 0] \to [0, +\infty)$ for some $a_{\max} > 0$ describes the *history* of the resource at time t, defined as $S_t(\theta) := S(t+\theta)$ for $\theta \in [-a_{\max}, 0]$. Above $a_{\max}$ plays the role of the maximum age of the consumer individuals.

The latter are structured by their (non-negative) size $\xi(a, S_t)$, which is relevant to individuals of age a that at time t have experienced a resource history S_t. The size $\bar{\xi}(\alpha) := \bar{\xi}(\alpha; a, \psi)$ at age $\alpha \in [0, a]$ of an individual that at age a has experienced a resource history ψ is given through the IVP

$$\begin{cases} \bar{\xi}'(\alpha) = g(\bar{\xi}(\alpha), \psi(-a + \alpha)), & \alpha \in [0, a], \\ \bar{\xi}(0) = \xi_b, \end{cases} \tag{8}$$

for given size at birth $\xi_b > 0$ and growth rate $g : [\xi_b, +\infty) \times [0, +\infty) \to (0, +\infty)$ sufficiently smooth. Then $\xi(a, \psi) := \bar{\xi}(a; a, \psi)$.

Similarly, $\mathcal{F}(a, S_t) \in [0, 1]$ denotes the survival probability of a consumer individual that at time t has age a and has experienced a resource history S_t. The survival probability $\bar{\mathcal{F}}(\alpha) := \bar{\mathcal{F}}(\alpha; a, \psi)$ at age $\alpha \in [0, a]$ of an individual that at age a has experienced a resource history ψ is given through the IVP

$$\begin{cases} \bar{\mathcal{F}}'(\alpha) = -\mu(\bar{\xi}(\alpha), \psi(-a + \alpha))\bar{\mathcal{F}}(\alpha), & \alpha \in [0, a], \\ \bar{\mathcal{F}}(0) = 1 \end{cases} \tag{9}$$

for a given mortality rate $\mu : [\xi_b, +\infty) \times [0, +\infty) \to (0, +\infty)$ sufficiently smooth. Then $\mathcal{F}(a, \psi) := \bar{\mathcal{F}}(a; a, \psi)$.

The reproduction and ingestion rates of a consumer individual that at time t has age a and size $\xi(a, S_t)$ are denoted by $\beta(\xi(a, S_t), S(t))$ and $\gamma(\xi(a, S_t), S(t))$, respectively, with given $\beta, \gamma : [\xi_b, +\infty) \times [0, +\infty) \to \mathbb{R}$ sufficiently smooth.

Consumers are assumed to be partitioned between juveniles and adults, the separation being determined by a given maturation size ξ_A ($> \xi_b$). Juveniles are not able to reproduce, so that $\beta(\xi, S) > 0$ only for $\xi \geq \xi_A$. The maturation age a_A is implicitly given through the maturation condition

$$\xi(a_A, \psi) = \xi_A, \tag{10}$$

so that $a_A = a_A(\psi)$ depends on the resource history, making the problem *state-dependent*. Correspondingly, the rates β, g, γ and μ are assumed to be piecewise sufficiently smooth functions with respect to their first argument on $[\xi_b, +\infty)$, with breaking point ξ_A.

Eventually, the dynamics at the population level is modeled as a system of a RE coupled to a DDE. As for the RE, the consumer birth rate $b(t)$ at time t is obtained by integrating with respect to the age the contribution of the individuals that have age a at time t, i.e.,

$$b(t) = \int_{a_A(S_t)}^{a_{\max}} \beta(\xi(a, S_t), S(t))\mathcal{F}(a, S_t)b(t - a)\,\mathrm{d}a. \tag{11}$$

The same reasoning leads to the DDE regulating the dynamics of the resource by subtracting from the right-hand side of the ODE in (7) the total ingestion of food:

$$S'(t) = f(S(t)) - \int_0^{a_{\max}} \gamma(\xi(a, S_t), S(t))\mathcal{F}(a, S_t)b(t - a)\,\mathrm{d}a. \tag{12}$$

For completeness, as motivated in [17], the system of (11) and (12) is equipped with initial histories $b_0 \in L^1([-a_{\max}, 0]; \mathbb{R})$ and $S_0 \in C([-a_{\max}, 0]; \mathbb{R})$ (the choice of the state spaces comes from the fact that b represents a rate, while S a density).

In conclusion, the target model consists of a coupled RE/DDE with integral right-hand sides, whose integration kernels are not given explicitly but rather through the solutions of external IVPs. Moreover, there can be discontinuities due to the different classes of the population, distinguished by their structuring variable and, in turn, by their age. Eventually, the breaking point between these classes is unknown in general, depending it on the state itself of the system through the solution of a nonlinear equation.

Remark 1. We stress that the model (11)–(12) with (8)–(10) can be obtained indirectly by integration along characteristic of an underlying Partial Differential Equation (PDE) of transport type. Indeed, by defining $n(t, \xi)$ as the density of individuals with size ξ at time t, the model can be described by the PDE

$$\partial_t n(t, \xi) + \partial_\xi[g(\xi, S(t))n(t, \xi)] = -\mu(\xi, S(t))n(t, \xi)$$

with boundary condition

$$g(\xi_b, S(t))n(t, \xi_b) = \int_{\xi_b}^\infty \beta(\xi, S(t))n(t, \xi)\,\mathrm{d}\xi.$$

The RE (11) is then obtained by defining $b(t) := g(\xi_b, S(t))n(\xi_b, t)$. One of the advantages of the RE (11) is that it allows to treat variables defined on the one-dimensional time domain. For a detailed derivation of the RE from the PDE we refer to [11, Chapter 6].

In the present work we are interested in computing the equilibria and in following their variation with respect to some model parameters as a first step in the dynamical analysis. Note, however, that the prototype problems we propose in Section 4 are suitable to investigate more general solutions (e.g., periodic ones) and thus, in this sense, the relevant internal strategy is presented in its most general version.

It is not difficult to see that the *Daphnia* model described above can have *trivial* equilibria, i.e., constant solutions $(b, S) \equiv (0, \bar{S})$ for $\bar{S}$ any zero of f in (12), as well as *nontrivial* equilibria $(b, S) \equiv (\bar{b}, \bar{S})$, where $\bar{S}$ satisfies

$$\int_{\bar{a}_A}^{a_{\max}} \beta(\xi(a, \bar{S}), \bar{S})\mathcal{F}(a, \bar{S}) \, \mathrm{d}a = 1 \tag{13}$$

for $\bar{a}_A := a_A(\bar{S})$ and $\bar{b}$ is consequently given by

$$\bar{b} = \frac{f(\bar{S})}{\int_0^{a_{\max}} \gamma(\xi(a, \bar{S}), \bar{S})\mathcal{F}(a, \bar{S}) \, \mathrm{d}a}.$$

We focus only on the nontrivial equilibria, hence on (13) (indeed, the computation of the trivial ones does not pose any special difficulties, except for those due to the specific form of f). In particular, the objective is to continue a solution branch of $\bar{S}$ with respect to a selected model parameter. Most of the model parameters are hidden in (13), appearing only in the definitions of the various rates defining the model. In this regard we refer to [7] for the current choices and, for the reader's convenience, we recall from there all the necessary quantities in Table 1.

resource intrinsic rate of change	$f(S) = a_1 S(1 - S/C)$
consumer growth rate	$g(\xi, S) = \gamma_g (\xi_m f_r(S) - \xi)$
consumer mortality rate	$\mu(\xi, S) = \mu$
consumer adults reproduction rate	$\beta(\xi, S) = r_m f_r(S)\xi^2$
consumer ingestion rate	$\gamma(\xi, S) = \nu_S f_r(S)\xi^2$
Holling type II functional response	$f_r(S) := \sigma S/(1 + \sigma S)$
size at birth	$\xi_b = 0.8$
size at maturation	$\xi_A = 2.5$
maximum size	$\xi_m = 6.0$
growth time constant	$\gamma_g = 0.15$
functional response shape parameter	$\sigma = 7.0$
maximum feeding rate	$\nu_S = 1.8$
maximum reproduction rate	$r_m = 0.1$
mortality rate parameter	$\mu = $ varying
environment carrying capacity	$C = 25$
flow-through rate	$a_1 = 0.5$
maximum age	$a_{\max} = 70$

TABLE 1. Rates (top) and parameters (bottom) of the considered *Daphnia* model.

In Figure 2 we show the results of the continuation of $\bar{S}$ with respect to the mortality rate parameter μ. Two curves are shown in the left panel, obtained as explained next.

The solid curve with circles is the result of the approach in [6], implemented in Matlab. The system of (11) and (12) is first reduced to a system of ODEs via a pseudospectral discretization. Then the latter is given in input to `MatCont`, which performs the continuation of the desired equilibrium. The quadrature of the integrals in (11) and (12) (by Clenshaw-Curtis quadrature [34], see Section 4.1), the resolution of the external IVPs (8) and (9) (by `ode45`) and the determination of the maturation age through (10) (by the event locator of `ode45`, see below) are all performed automatically inside the routine constructing the right-hand side of the
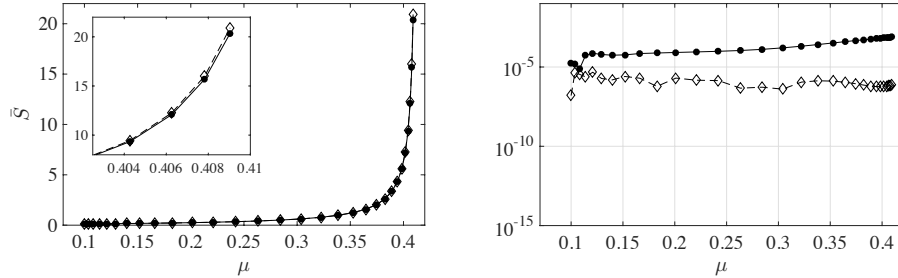
FIGURE 2. Equilibrium branch $\bar{S}(\mu)$ with zoom (left) and relevant residual (right) of the *Daphnia* model, computed with [6] (solid line with circles) and [32] (dashed line with diamonds). See text for more details.

reduced system of ODEs. As such, all these calculations are repeated from scratch at every continuation step done by `MatCont`, with no possibility of exploiting the information available from the previous step.

The dashed curve with diamonds is, instead, the result of a natural continuation with secant prediction, recall Figure 1 (bottom-right), always implementd in Matlab. Following [32], the correction step is made by using the Broyden's update, integrals and external IVPs are solved simultaneously via the embedded Runge-Kutta pair `DOPRI54` (see, e.g., [22]: basically, it is also behind `ode45`) and the maturation age is directly obtained as an output of the latter, which is indeed capable of *event detection*, i.e., it automatically detects when (10) is satisfied during the integration of (8). As for the previous approach, integrals, IVPs and maturation age are computed from scratch at every continuation step, independently of the same quantities computed at the previous step.

In the following we refer to the two approaches described above by simply citing the works of reference, i.e., [6] and [32], respectively. Let us note that other works exist on the numerical equilibrium analysis of the *Daphnia* model, see, e.g., [12] and the references therein. Yet, from the numerical point of view, [32] is largely inspired from [12], so that the applied numerical techniques are basically the same. A similar comment holds also with regards to the software `PSPManalysis` [10].

Back to Figure 2 (left), the two curves are practically indistinguishable, but for the greatest values of μ as highlighted in the zoom in the inner panel. The same can be said for the marked points, which represent the performed continuation steps. Let us remark that it is difficult to set all the parameters regulating the automatic choice of the length of the prediction step in order to make a reasonable and fair comparison (e.g., in `MatCont`, as used with [6], it is not possible to directly control $\underline{k}$, $\bar{k}$ or c, but just the minimum or maximum length of the steps). The same can be said about fixing the tolerances or the discretization parameters of all the concerned tools (`MatCont`, pseudospectral reduction, `ode45`, `DOPRI54`, Broyden's update, etc.). Nevertheless, we did our best to avoid favoring any of the two methods as much as we could. To this aim, the lengths along μ of the secant prediction steps as used with [32] are chosen to be equal to those obtained with [6]. Doing this way the two computed branches cover the same range for μ (viz. $[0.1, 0.4090]$) with the same number of steps (viz. 30). Moreover, as explained next, the resulting errors have similar magnitude.

Figure 2 (right) shows in fact the residual obtained along the branch with the two methods. By *residual* we mean the absolute difference between 1 and the integral at the left-hand side of (13). Indeed, the latter can be evaluated analytically thanks to the choices in Table 1, while the input values for $\bar{S}$ are those computed with either [6] (solid line with circles) or [32] (dashed line with diamonds). As anticipated, the outcome is comparable, even though a slightly increasing trend emerges for [6], which we believe is due to the effect of μ on the error of the pseudospectral reduction as used in [6]. Note that the range of the vertical axis is kept as large in order to facilitate the comparison with the method proposed in this work, as reported later on in Section 6.5. Eventually, it is worth to remark that the error as measured here is the result of several sources, qualitatively and quantitatively different (quadrature, IVPs, maturation condition, correction procedures, `MatCont` inner tolerances), so that we refrain from drawing sharper conclusions, unless differences of several orders of magnitude were the case (unlike the current situation indeed).

Taking into account the aforementioned premises, the computational time needed to trace the equilibrium branch amounts to 155.88 s with [6] and 59.32 s with [32] (both run on a MacBook Pro 2.3GHz Intel Core i7 16GB, the same hardware used to perform also the tests in Section 6). As anticipated in the Introduction, these data furnish enough motivation to try to improve the continuation strategy for complex models like the *Daphnia* one. This is the main contribution of the present work and we illustrate it in the following sections.

4. **Prototype problems and internal continuation.** With the aim of decreasing the overall computational cost of the techniques available in the literature to continue the equilibria of the complex models of interest for this work, we propose a new strategy whose underlying idea is to take advantage of the information acquired at the previous continuation step when computing a new point of the branch. With reference to the class of problems exemplified by the *Daphnia* model, this can be achieved by including into the continuation framework the solution of the external IVPs (8) and (9) as well as the solution of the maturation condition (10).

In this sense we talk about *internal* continuation, as opposed to the standard *external* continuation methods, as those employed at the end of Section 3.

In the sequel we keep on focusing on (13) even though, as already anticipated, the following analysis is valid for continuing also solutions other than equilibria.

In order to test our proposal, we first separate the peculiar difficulties of the *Daphnia* model, represented by (13). To this aim, in the following sections we formulate a series of prototype problems, each of which tackles one single challenge of the original model at a time, namely the presence of an external ODE, a state-dependent maturation age, possible discontinuities among juveniles and adults and, finally, systems of external ODEs. In each of these sections we also describe how the problem is addressed in the framework of the newly proposed internal continuation.

Then in Section 5 we summarize the features of the external continuation used to compare the results on these prototype problems. All the computational tests are reported in Section 6, together with the final comparison with [6, 32] on the *Daphnia* model. We anticipate that the specific instances of prototype problems used in Section 6 are constructed to obtain a known analytic expression of the solution branch, with the aim of measuring the true error due to the applied continuation.

First to proceed, note that the notation used in the following sections is completely unrelated to that used so far for the *Daphnia* model, but the context should be clear enough to avoid confusion. In particular, the final objective is that of determining a quantity $x \in \mathbb{R}$ as a function of a varying parameter $\lambda \in \mathbb{R}$, defined implicitly through an integral condition. With reference to the case of *Daphnia*, x should be read as $\bar{S}$, λ as the mortality parameter μ (recall Table 1) and the integral condition as (13).

4.1. **The basic case.** We concentrate on the treatment of the integral in (13) and on the fact that the integrand is defined through the solution of external IVPs. In this respect we simplify the problem considering just a single ODE (systems are dealt with in Section 4.4). Moreover, the integration extrema are kept fixed in order to avoid state-dependency (argument of Section 4.2).

The prototype problem is thus represented by the continuation of the curve $x(\lambda)$ defined by

$$\int_0^1 f(a, x, \lambda) \, da = 0, \tag{14}$$

where $f(a, x, \lambda) := \varphi(a; a, x, \lambda)$ and $\varphi(\alpha; a, x, \lambda) \in \mathbb{R}$ is the solution of

$$\begin{cases} \varphi'(\alpha; a, x, \lambda) = g(\varphi(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [0, a], \\ \varphi(0; a, x, \lambda) = \varphi_0 \end{cases} \tag{15}$$

for given $g : \mathbb{R} \times [0, 1] \times \mathbb{R}^2 \to \mathbb{R}$ sufficiently smooth and $\varphi_0 \in \mathbb{R}$.

Note that, biologically, it should be $\varphi(\alpha; a, x, \lambda) \geq 0$ as well as $\varphi_0 > 0$. As far as only prototype problems are concerned, we feel free to relax these and similar constraints, recalling that in the end we want to obtain exact expressions of the solution branches.

We need to choose how to approximate both the integral in (14) and the solution of the IVP (15) needed to compute the relevant values of the integrand function.

As for the former, we opt for the Clenshaw-Curtis quadrature [34], i.e.,

$$\int_0^1 f(a, x, \lambda) \, da \approx \sum_{j=0}^N w_j f(a_j, x, \lambda),$$

where N is a fixed positive integer, $0 = a_0 < \cdots < a_N = 1$ are the Chebyshev extrema in $[0, 1]$ and $w_0, \ldots, w_N$ are the corresponding quadrature weights. The same quadrature is used for the forthcoming prototypes as well. Moreover, recall from Section 3 that it is also adopted for treating the *Daphnia* problem with [6].

As for the IVP, we need the value of $f(a_j, x, \lambda)$ for each $j = 1, \ldots, N$ given x and λ, and hence those of $\varphi(\alpha; a_j, x, \lambda)$ at $\alpha = a_j$ ($j = 0$ is excluded with the above choice of quadrature nodes because $f(a_0, x, \lambda) = \varphi_0$ is given). To this aim, we use polynomial collocation and thus look for an n-degree polynomial $p^{(j)}(\alpha) := p(\alpha; a_j, x, \lambda)$ such that

$$\begin{cases} p^{(j)\prime}(\alpha_i^{(j)}) = g(p^{(j)}(\alpha_i^{(j)}), a_j, x, \lambda), & i = 1, \ldots, n, \\ p^{(j)}(\alpha_0^{(j)}) = \varphi_0. \end{cases}$$

for given collocation points $0 = \alpha_0^{(j)} < \cdots < \alpha_n^{(j)} = a_j$, $j = 1, \ldots, N$.

With respect to (6) let then

$$u := (p^{(1)}(\alpha_1^{(1)}), \ldots, p^{(1)}(\alpha_n^{(1)}), \ldots, p^{(N)}(\alpha_1^{(N)}), \ldots, p^{(N)}(\alpha_n^{(N)}), x)^T \in R^{nN+1}, \ (16)$$

so that the collocation variables $p^{(j)}(\alpha_i^{(j)})$, $i = 1, \ldots, n$, $j = 1, \ldots, N$, are included in the continuation framework (again, $i = 0$ is excluded since $p^{(j)}(\alpha_0^{(j)}) = \varphi_0$ for every $j = 1, \ldots, N$).

Then G in (6) is given componentwise by

$$\begin{cases} p^{(j)'}(\alpha_i^{(j)}) - g(p^{(j)}(\alpha_i^{(j)}), a_j, x, \lambda), & i = 1, \ldots, n, \ j = 1, \ldots, N \\ \sum_{j=0}^{N} w_j p^{(j)}(\alpha_n^{(j)}). \end{cases}$$

Note that the first group of equations does not concern any collocation variable $p^{(k)}(\alpha_i^{(k)})$ for $k \neq j$. This means that the resulting Jacobian matrix will have a bordered block diagonal structure, as shown in Figure 3.
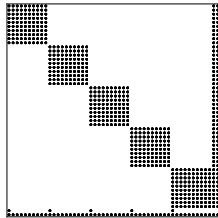

FIGURE 3. An example of bordered block diagonal structure of the Jacobian matrix for $n = 10$ (determining the size of the diagonal blocks) and $N = 5$ (determining the number of the diagonal blocks).

Some comments are worthy with regards to the numerical solution of (15) to compare internal and external continuation. Recall that we need this solution at the end of the integration interval $[0, a_j]$ for every quadrature node a_j (a_0 excluded). The internal continuation uses collocation, which has a classical drawback when applied to solve IVPs. Indeed, this leads to solve nonlinear equations, requiring for a suitable initial guess to start the chosen iterative solver. Clearly, this problem is immediately solved by including the collocation variables into the continuation framework, since the mentioned initial guess is directly provided by the same quantities as computed at the previous continuation step (or, better, at the current prediction step). As far as the external continuation is concerned, instead, one typically chooses standard initial value solvers (like, e.g., `ode45` in Matlab), which means that the numerical solution is based just on the initial value φ_0, independently of what computed at the preceding continuation step, thus repeating all the calculations from scratch. This should be the main source of computational advantage of the internal strategy we propose with respect to traditional external ones.

Nevertheless, there is a price to pay, which is represented by the increased dimension of the continuation problem, namely $O(nN)$ for the internal approach as opposed to $O(N)$ for the external one (quadrature is anyway necessary for both). This is in part lightened by the structure of the Jacobian matrix mentioned above,

but the situation worsens as the number of external ODEs increases (see Section 4.4).

Eventually, let us underline that in the case of equilibria, e.g., for (13), the family of ODEs (15) parameterized by $a \in [0, a_{\max}]$ actually reduces to the single ODE corresponding to (8) for $\psi = \bar{S}$ constant and $\alpha = a \in [0, a_{\max}]$ since g does not depend on a anymore. We still use the full formulation (15) in view of the extension of the method to, for instance, periodic solutions, for which such a simplification is not possible.

4.2. **State-dependent problems.** With respect to the previous section, we put back the problem of state-dependency, assuming that one extremum of integration in (14) depends on the unknown x.

The prototype problem is thus represented by the continuation of the curve $x(\lambda)$ defined by

$$\int_{\bar{a}}^{1} f(a, x, \lambda) \, \mathrm{d}a = 0, \tag{17}$$

where $\bar{a} = \bar{a}(x, \lambda)$ is implicitly defined by

$$f(\bar{a}, x, \lambda) = \bar{\varphi} \tag{18}$$

for given $\bar{\varphi}$, the rest being unchanged with respect to Section 4.1. The focus here is only in the addition of condition (18) to the problem. In Section 4.3 we address also the possible change of the right-hand side of the external ODE across $\bar{a}$.

The problem now includes the extra (nonlinear) equation (18) and, consequently, the internal strategy suggests to consider just an extra continuation variable, which is $\bar{a}$. Hence (16) becomes

$$u := (p^{(1)}(\alpha_1^{(1)}), \ldots, p^{(1)}(\alpha_n^{(1)}), \ldots, p^{(N)}(\alpha_1^{(N)}), \ldots, p^{(N)}(\alpha_n^{(N)}), x, \bar{a})^T \in \mathbb{R}^{nN+2}.$$

As it happens for the solution of the external IVP, also the computation of the value of $\bar{a}$ takes advantage of the proposed internal strategy, indeed the value computed at the previous continuation step is used to start the iterative solver at the current step. Let us remark once more that in the external case, in principle, an initial guess to solve (18) is never available to the chosen external solver, fact that may also prevent the convergence to a solution. Nevertheless, IVP solvers with event location (as those used in Section 3) avoid this problem, too.

4.3. **External ODE with discontinuous right-hand side.** In realistic models, the right-hand side of the external ODE may change across $\bar{a}$. This can be, e.g., the case of *Daphnia*, where the right-hand side of the ODE in (8) defining the size represents the growth rate of the consumer population, which in general is different between juveniles and adults. The same can obviously hold for (9), too.

To accommodate this feature we consider that f in (17) is defined through

$$\begin{cases} \varphi'(\alpha; a, x, \lambda) = g_1(\varphi(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [0, a], \\ \varphi(0; a, x, \lambda) = \varphi_0 \end{cases}$$

as long as $a \leq \bar{a}$, while for $a > \bar{a}$ it is defined through

$$
\begin{cases}
\varphi'(\alpha; a, x, \lambda) = g_2(\varphi(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [\bar{a}, a], \\
\varphi(\bar{a}; a, x, \lambda) = \bar{\varphi}, \\
\varphi'(\alpha; a, x, \lambda) = g_1(\varphi(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [0, \bar{a}], \\
\varphi(0; a, x, \lambda) = \varphi_0.
\end{cases}
$$

Of course, the functions g_1 and g_2 might be different.

In the case $a > \bar{a}$, the collocation solution is a continuous piecewise polynomial in $[0, a]$, with $\bar{a}$ the only breaking point. In principle, different numbers of collocation nodes can be used in the two intervals, even though in the tests we perform in Section 6 this number is kept the same for simplicity.

As a side remark, let us note that multi-point boundary-value solvers can be suitably adopted in the presence of discontinuities, such as, e.g., the `bvp` toolbox of `coco` [9].

4.4. **Systems of external ODEs.** Finally, as a last prototype, we go back to the case with no state-dependency treated in Section 4.1, but now we consider the presence of more external ODEs determining the integrand. Although the internal approach is identical, but for the dimension of the continuation problem increased to $O(nNd)$, where d is the number of considered external ODEs, we retain worth describing it for experimental purposes ($d = 2$ below). Indeed, as we will see in Section 6.4, the internal continuation may not always result advantageous in terms of speed-up.

To exemplify we continue the curve $x(\lambda)$ defined by

$$
\int_0^1 f(a, x, \lambda) c(a, x, \lambda) \, \mathrm{d}a = 0, \tag{19}
$$

where $f(a, x, \lambda) := \varphi(a; a, x, \lambda)$, $c(a, x, \lambda) := \gamma(a; a, x, \lambda)$ and the solution pair $(\varphi(\alpha; a, x, \lambda), \gamma(\alpha; a, x, \lambda)) \in \mathbb{R}^2$ is relevant to the system

$$
\begin{cases}
\varphi'(\alpha; a, x, \lambda) = g(\varphi(\alpha; a, x, \lambda), \gamma(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [0, a], \\
\gamma'(\alpha; a, x, \lambda) = h(\varphi(\alpha; a, x, \lambda), \gamma(\alpha; a, x, \lambda), a, x, \lambda), & \alpha \in [0, a], \\
\varphi(0; a, x, \lambda) = \varphi_0, \\
\gamma(0; a, x, \lambda) = \gamma_0
\end{cases} \tag{20}
$$

for given $g, h : \mathbb{R}^2 \times [0, 1] \times \mathbb{R}^2 \to \mathbb{R}$ and $\varphi_0, \gamma_0 \in \mathbb{R}$. Note that, for our goals, the integrand in (19) may in principle be defined as any function of $f(a, x, \lambda)$ and $c(a, x, \lambda)$ having a simple definition (to strive for an exact solution), such as a linear combination.

Finally, with respect to (16), we decide to group the collocation unknowns as block-vectors with blocks of size d, so that the structure of (16) remains unchanged but for the range of p which is now in $\mathbb{R}^d$. This choice does not influence neither the overall structure of the Jacobian matrix of the relevant Newton's method (recall, e.g., Figure 3), just each entry is now a block of size $d \times d$.

5. **External continuation.** As already observed, there exist several continuation-based software tools for solving our class of problems using the standard external approach. For the purpose of comparing with the internal continuation strategy, we emulate the behavior of those tools using the correspondent Python routines. In particular, we use pseudo-arclength continuation with tangent prediction and Newton's correction.

The IVPs are solved by the Python IVP solver `scipy.integrate.odeint` from the `scipy` package [27], based on the `LSODA` solver from the FORTRAN library `odepack`, which is able to switch automatically between non-stiff problems (solved using the implicit Adams formula) and stiff ones (solved using backward differentiation formulas), according to the method proposed in [29].

For the solution of the nonlinear equation representing the maturation condition, we use `scipy.optimize.fsolve`, which is based on the `HYBRD` and the `HYBRJ` solvers from the FORTRAN library `minpack`, which implement a variant of Powell's hybrid method [30]. The need for an external solver of nonlinear equations comes from the fact that `odeint`, unlike `ode45`, is not capable of event detection.

We remark that Python is an arbitrary choice, in fact, everything could be implemented just as well using other software.

6. **Numerical tests.** In the following Sections 6.1 to 6.4 we show the results of numerical simulations using our internal continuation approach on the prototype models described in Section 4. We compare its performance with the external continuation described in Section 5. For both approaches the tolerance of the Newton's corrections is set to 10^{-13}. As already anticipated, all the tests are run on a MacBook Pro 2.3GHz Intel Core i7 16GB.

Finally, in Section 6.5 we test the performance of the internal continuation on the *Daphnia* model, and compare with the results relevant to Figure 2 as described at the end of Section 3.

6.1. **The basic case.** With reference to (15) in Section 4.1, we consider

$$g(\varphi(\alpha; a, x, \lambda), a, x, \lambda) = \lambda\varphi(\alpha; a, x, \lambda) + 2xe^{-\lambda a}, \qquad (21)$$

for which the solution branch defined by (14) reads

$$x(\lambda) = \frac{\varphi_0}{2} \cdot \frac{\lambda(1 - e^{\lambda})}{\lambda + e^{-\lambda} - 1}. \qquad (22)$$

By knowing the latter we can evaluate the true error on the continuation curve by varying the number of collocation and quadrature nodes, while running a fixed number of continuation steps (precisely 10) in order to reach the same final value of the continuation parameter with both the internal and external approaches (this holds for all the tests below).

Figure 4 (top) shows the error obtained on the true curve (22) when using (21) as right-hand side of (15) and $\varphi_0 = 1$, with increasing number n of collocation nodes and fixing $N = 10$ quadrature nodes. The error decays *spectrally* as n increases in the internal case (line with circles). Indeed, this is the expected behavior of collocation, because our problem is smooth [35]. The horizontal lines are the result of the external continuation where the tolerances of `odeint` and `fsolve` are both set to 10^{-8}, 5×10^{-10} and 10^{-13} respectively. For each of those values there is (at least) a number n of collocation points for which the internal continuation performs

better in terms of both time and error. The diamond markers in Figure 4 (bottom) show that this holds for $n = 7$, 8 and 11 respectively.
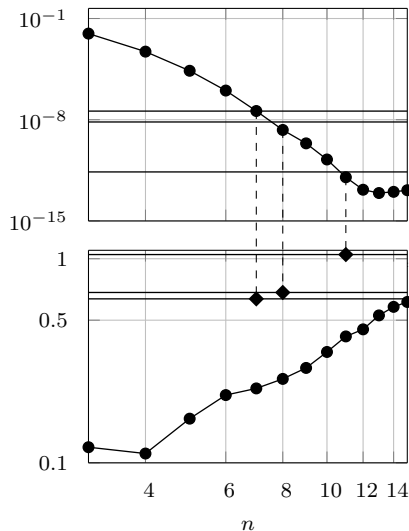

FIGURE 4. Internal (lines with circles) versus external (horizontal lines) continuation for (21): error on the true curve (22) (top) and elapsed time (bottom, s) using n collocation points and $N = 10$ quadrature nodes. See text for more details.

The simulations above were replicated using different values for the number N of quadrature nodes, up to 100, and always gave, qualitatively speaking, the same results. A partial evidence of this is shown in Figure 5, obtained fixing $n = 12$ collocation nodes and varying the number N of quadrature nodes. Figure 5 (top) shows the error obtained on the true curve (22), while Figure 5 (bottom) compares the elapsed time. Lines marked with circles refer to the internal continuation, while the ones with squares refer to the external continuation with external tolerances fixed to 10^{-13}: the internal continuation performs better in terms of both time and error.

6.2. **State-dependent problems.** With reference to Section 4.1 and Section 4.2, we show the results obtained by choosing

$$g(\varphi, a, x, \lambda) = -(a+1)x(\lambda^2 + 2)\left(\frac{\varphi - \varphi_0}{x(\lambda^2 + 2)(a+1)} + 2\right)^2 \tag{23}$$

in (15) and

$$\bar{\varphi} = \varphi_0 - 1 \tag{24}$$

in (18).

Figure 6 (top) shows the error obtained on the true curve

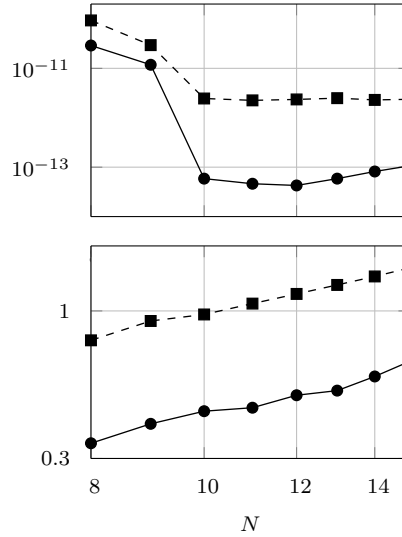$$x(\lambda) = \frac{2\varphi_0 - 1}{\lambda^2 + 2} \tag{25}$$

FIGURE 5. Internal (lines with circles) versus external continuation (lines with squares) for (21): error on the true curve (22) (top) and elapsed time (bottom, s) using $n = 12$ collocation points and N quadrature nodes. See text for more details.

when using (23), (24) and $\varphi_0 = 1.5$, with increasing number n of collocation nodes and fixing $N = 10$ quadrature nodes. Again, the horizontal lines are the result of external continuation where the tolerances of `odeint` and `fsolve` are both set to 10^{-8}, 5×10^{-10} and 10^{-13} respectively. For each of those values there is (at least) a number n of collocation points for which the internal continuation performs better in terms of both time and error. The diamond markers in Figure 6 (bottom) show that this holds for $n = 13$, 12 and 17 respectively. As a side remark, note that the external continuation takes slightly less time for 5×10^{-10} than for 10^{-8}: this can be caused by the automatic error control of either `odeint` or `fsolve`.

6.3. **External ODE with discontinuous right-hand side.** With reference to Section 4.3, we show the results obtained by choosing

$$g_1(\varphi, a, x, \lambda) = -x(\lambda^2 + 1)\left(\frac{\varphi - \varphi_0}{x(\lambda^2 + 1)} - \frac{1}{2}\right)^2, \quad \alpha \in [0, \bar{a}], \qquad (26)$$

$$g_2(\varphi, a, x, \lambda) = -(a+1)x(\lambda^2 + 1)\left(\frac{\varphi - \varphi_0}{x(\lambda^2 + 1)(a + 1)} + \frac{1}{\bar{a} + 1}\right)^2, \quad \alpha \in [\bar{a}, a], \quad (27)$$

and

$$\bar{\varphi} = \frac{1}{2}\varphi_0 \qquad (28)$$

in (18) of Section 4.2.

Figure 7 (top) shows the error obtained on the true curve

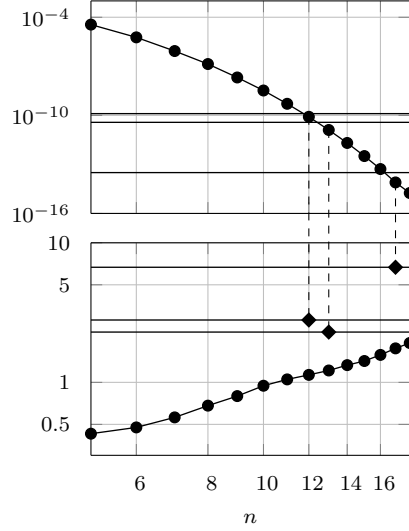$$x(\lambda) = \frac{3\varphi_0}{\lambda^2 + 1} \qquad (29)$$

FIGURE 6. Internal (lines with circles) versus external (horizontal lines) continuation for (23) and (24): error on the true curve (25) (top) and elapsed time (bottom, s) using n collocation points and $N = 10$ quadrature nodes. See text for more details.

when using (26), (27), (28) and $\varphi_0 = 1$, with increasing number n of collocation nodes and fixing $N = 10$ quadrature nodes. The horizontal lines are the result of external continuation where the tolerances of `odeint` and `fsolve` are both set to $10^{-8}, 5 \times 10^{-10}$ and 10^{-13} respectively. For each of those values there is (at least) a number n of collocation points for which the internal continuation performs better in terms of both time and error. The diamond markers in Figure 7 (bottom) show that this holds for $n = 8$, 10 and 11 respectively.

6.4. **Systems of external ODEs.** With reference to Section 4.4, we show the results obtained by choosing

$$g(\varphi, \gamma, a, x, \lambda) =$$
$$-(a + 1)x(\lambda^2 + 1) \left(\frac{\varphi - \varphi_0}{x(\lambda^2 + 1)(a + 1)} + 1 \right) \left(\frac{\gamma - \gamma_0}{(\lambda^2 + 1)(a + 1)} + 1 \right) \quad (30)$$

and

$$h(\varphi, \gamma, a, x, \lambda) =$$
$$-(a + 1)(\lambda^2 + 1) \left(\frac{\varphi - \varphi_0}{x(\lambda^2 + 1)(a + 1)} + 1 \right) \left(\frac{\gamma - \gamma_0}{(\lambda^2 + 1)(a + 1)} + 1 \right) \quad (31)$$

in (20).

Figure 8 (top) shows the error obtained on the true curve

$$x(\lambda) = \frac{3\varphi_0}{\lambda^2 + 1} \cdot \frac{\lambda^2 + 1 - 2\varphi_0}{2(\lambda^2 + 1) - 3\gamma_0} \quad (32)$$

when using (30), (31) and $\varphi_0 = \gamma_0 = 1$, with increasing number n of collocation nodes and fixing $N = 10$ quadrature nodes. The horizontal lines are the result of external continuation where the tolerances of `odeint` and `fsolve` are both set
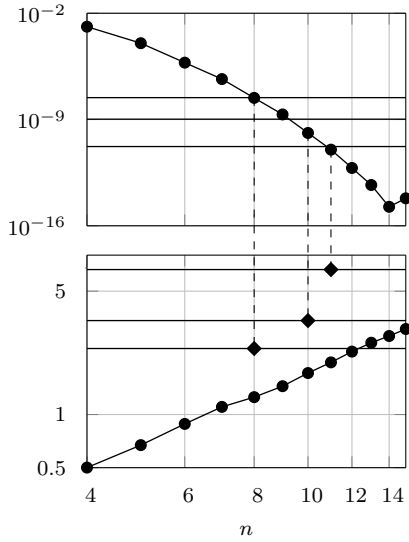
FIGURE 7. Internal (lines with circles) versus external (horizontal lines) continuation for (26), (27) and (28): error on the true curve (29) (top) and elapsed time (bottom, s) using n collocation points and $N = 10$ quadrature nodes. See text for more details.

to 10^{-8}, 5×10^{-10} and 10^{-13} respectively. Unlike the previous scalar cases, the external continuation seems to perform slightly better in terms of both time and error. The diamond markers in Figure 8 (bottom) show that this holds for $n = 12$, 13 and 18 respectively (mind anyway the time scale in the bottom panel). Indeed, the explanation resides in the increased dimension of the continuation problem for the internal approach, as already explained in Section 4.1 and further commented on in Section 7. Nevertheless, other experiments (not reported here) show that there are also cases slightly in favor of the internal approach, especially when stricter tolerances are used.

6.5. **Daphnia.** We finally compare the results of the continuation of the branch $\bar{S}(\mu)$ obtained with [6] and [32] in Section 3 (recall Table 1 and Figure 2) with those obtained with the internal continuation. As for the latter, the initial step and the parameters of the automatic step selection are chosen in order to cover (almost) the same range for μ as in Figure 2, always with 30 steps (viz. $\underline{k} = 2$, $\overline{k} = 20$ and $c = 1.3$). Moreover, the tolerance of the Newton's corrections is set to 10^{-6}. We perform three runs, which differ only by the number of quadrature and collocation nodes, respectively $n = N = 10$, 15 and 20.

In Figure 9 we superpose to Figure 2 the results thus obtained. In particular, in the left panel we add only the curve obtained with $n = N = 20$ (dash-dot line with stars), anticipating the correctness of the results. In the right panel, instead, we add the residual for all the three choices of $n = N$ (again dash-dot lines with stars): lines with smaller residual correspond to larger values of $n = N$. Note a slightly more prominent increasing trend with respect to μ than for [6] (recall the relevant comment in Section 3).
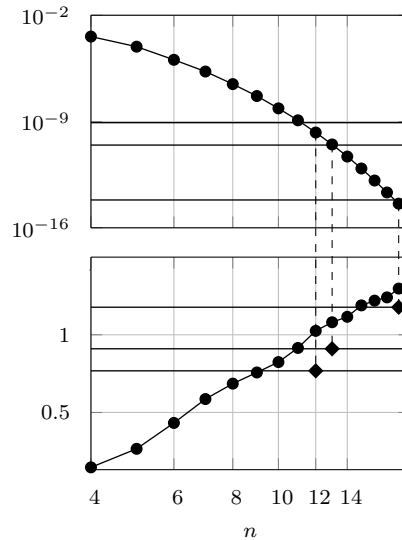
FIGURE 8. Internal (lines with circles) versus external (horizontal lines) continuation for (30) and (31): error on the true curve (32) (top) and elapsed time (bottom, s) using n collocation points and $N = 10$ quadrature nodes. See text for more details.
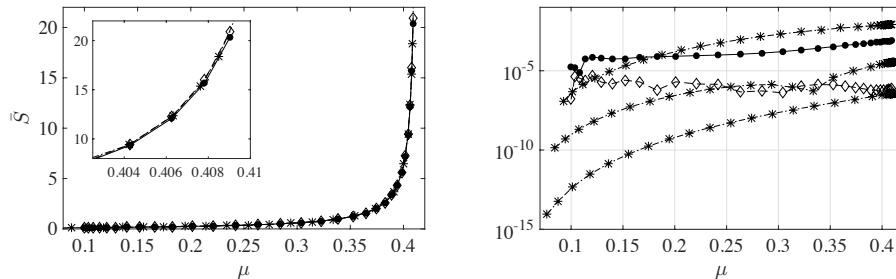

FIGURE 9. Equilibrium branch $\bar{S}(\mu)$ with zoom (left) and relevant residual (right) of the *Daphnia* model, computed with the internal continuation (dash-dot line with stars), superposed to Figure 2 for comparison. See text for more details.

As far as the computational time is concerned, in relation to the maximal residual, the outcome reported in Table 2 clearly demonstrates the superiority of the internal continuation with respect to either [6] or [32]. Let us observe, anyway, that the internal continuation is implemented in Python, whereas [6] and [32] are implemented in Matlab (for the former there is no alternative due to `MatCont`, for the latter the relevant codes are available only in Matlab): in neither case the different language is responsible for such an evident speed-up. Moreover, recall that [32] is implemented with secant prediction and Broyden's update, both choices favoring the latter with respect to the internal approach for avoiding the computation of the Jacobian.

method	computational time	maximal residual
[6]	155.88 s	7.6652×10^{-4}
[32]	59.32 s	4.6768×10^{-6}
internal continuation with $n = N = 10$	1.73 s	8.1723×10^{-3}
internal continuation with $n = N = 15$	4.40 s	3.6517×10^{-5}
internal continuation with $n = N = 20$	9.18 s	3.6854×10^{-7}

TABLE 2. Computational time and maximal residual for the continuation of the *Daphnia* model.

7. **Concluding remarks.** The analysis of stability and bifurcation is a major target of population dynamics. When complex models are concerned, the approach recently proposed in [6] represents undoubtedly a promising solution in terms of generality and applicability, whether not the only one. Nevertheless, several experiments (run by the authors even outside the current work) have shown that the requirements in terms of computational time are often severe. The present work aims at investigating possible remedies to this drawback, and the proposed internal continuation strategy seems to be a valid candidate as shown by the numerical tests. The implementation of such approach comes, however, at the expense of loss of generality, since it relies on exploiting the features of the specific model (e.g., external IVPs, maturation condition, etc.) to take advantage of the continuation framework. Note, anyway, that this new method is also superior to that proposed in [32] (and [10, 12]), which suffers from the same lack of generality.

Let us note, however, that the current analysis concentrates on saving computational time with respect to the approach of [6], for the reasons mentioned above. Nevertheless, other potential benefits can be taken into account, like avoiding the use of time steppers for ODEs which may cause discontinuities, failure in choosing an optimal pivoting strategy and prevent the use of parallelization and vectorization of the relevant codes.

Besides the aspects just summarized, which might be more evident in the case of more difficult problems, we plan to further investigate the internal continuation by also testing models with more than two external ODEs (see, e.g., [32]). Indeed, the advantage of our approach seems to be milder when the number of external ODEs increases, due to the correspondent increase in the dimension of the continuation problem as observed in Section 6.4. Specifically, a possible improvement can be realized by suitably exploiting the structure of the Jacobian of the correction steps (recall Figure 3).

The present research is restricted to the analysis of equilibria as a starting point. Our ultimate goal is to extend the internal continuation thus proposed to the detection of bifurcations and to the continuation of periodic orbits. Computationally speaking, the latter is a even more challenging problem. On a side note, we point out that the usual computation of periodic orbits inside a continuation framework is indeed an instance of *internal* approach.

REFERENCES

[1] AUTO, http://indy.cs.concordia.ca/auto/.

[2] MatCont, https://sourceforge.net/projects/matcont/.

[3] E. L. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*, vol. 45 of Classics in Applied Mathematics, SIAM, Philadelphia, 2003.

[4] A. Andò and D. Breda, Collocation techniques for structured populations modeled by delay equations, SEPA SIMAI series, Springer, To appear.

[5] D. Breda, O. Diekmann, W. de Graaf, A. Pugliese and R. Vermiglio, On the formulation of epidemic models (an appraisal of Kermack and McKendrick), *J. Biol. Dyn.*, **6** (2012), 103–117.

[6] 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.

[7] D. Breda, P. Getto, J. Sánchez Sanz and R. Vermiglio, Computing the eigenvalues of realistic Daphnia models by pseudospectral methods, *SIAM J. Sci. Comput.*, **37** (2015), 2607–2629.

[8] C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Math. Comp.*, **19** (1965), 577–593.

[9] H. Dankowicz and F. Schilder, *Recipes for continuation*, vol. 11 of Computational Science and Engineering, SIAM, Philadelphia, 2013.

[10] A. M. de Roos, PSPManalysis, https://cran.r-project.org/package=PSPManalysis.

[11] A. M. de Roos, A gentle introduction to models of physiologically structured populations, in *Structured-population models in marine, terrestrial and freshwater systems* (eds. S. Tuljapurkar and H. Caswell), Chapman and Hall, New York, 1997, 119–204.

[12] A. M. de Roos, O. Diekmann, P. Getto and M. A. Kirkilionis, Numerical equilibrium analysis for structured consumer resource models, *B. Math. Biol.*, **72** (2010), 259–297.

[13] A. M. de Roos, J. A. J. Metz, E. Evers and A. Leipoldt, A size-dependent predator prey interaction: who pursues whom?, *J. Math. Biol.*, **28** (1990), 609–643.

[14] P. Deuflhard, B. Fiedler and P. Kunkel, Efficient numerical pathfollowing beyond critical points, *SIAM J. Numer. Anal.*, **24** (1987), 912–927.

[15] A. Dhooge, W. J. F. Govaerts and Y. A. Kuznetsov, MatCont: A MATLAB package for numerical bifurcation analysis of ODEs, *ACM T. Math. Software*, **29** (2003), 141–164.

[16] A. Dhooge, W. J. F. Govaerts, Y. A. Kuznetsov, H. G. E. Meijer and B. Sautois, New features of the software MatCont for bifurcation analysis of dynamical systems, *Math. Comput. Model. Dyn. Syst.*, **14** (2008), 147–175.

[17] O. Diekmann, P. Getto and M. Gyllenberg, Stability and bifurcation analysis of Volterra functional equations in the light of suns and stars, *SIAM J. Math. Anal.*, **39** (2008), 1023–1069.

[18] O. Diekmann, M. Gyllenberg, J. A. J. Metz, S. Nakaoka and A. M. de Roos, *Daphnia* revisited: local stability and bifurcation theory for physiologically structured population models explained by way of an example, *J. Math. Biol.*, **61** (2010), 277–318.

[19] 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.

[20] E. Doedel, AUTO: a program for the automatic bifurcation analysis of autonomous systems, in *Proceedings of the Tenth Manitoba Conference on Numerical Mathematics and Computing*, vol. I, Winnipeg, Man., 1981, 265–284.

[21] E. Doedel, Lecture notes on numerical analysis of nonlinear equations, in *Numerical continuation methods for dynamical systems* (eds. H. M. Osinga, B. Krauskopf and J. Galán-Vioque), Understanding Complex Systems, Springer, 2007, 1–49.

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

[23] P. Getto, M. Gyllenberg, Y. Nakata and F. Scarabel, Stability analysis of a state-dependent delay differential equation for cell maturation: Analytical and numerical methods, *J. Math. Biol.*, URL https://doi.org/10.1007/s00285-019-01357-0.

[24] W. J. F. Govaerts, *Numerical methods for bifurcations of dynamical equilibria*, SIAM, Philadelphia, 2000.

[25] 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.

[26] H. Inaba, *Age-Structured Population Dynamics in Demography and Epidemiology*, Springer, New York, 2017.

[27] E. Jones, T. Oliphant, P. Peterson et al., SciPy: Open source scientific tools for Python, 2001–, URL http://www.scipy.org/.

[28] Y. Kuang, *Delay Differential Equations: With Applications in Population Dynamics*, no. 191 in Dynamics in Science and Engineering, Academic Press, New York, 1993.

[29] L. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, *SIAM J. Sci. Stat. Comput.*, **4** (1983), 136–148.

[30] M. J. D. Powell, A hybrid method for nonlinear equations, in *Numerical Methods for Nonlinear Algebraic Equations* (ed. P. Rabinowitz), Gordon and Breach, 1970.

[31] W. C. Rheinboldt and J. V. Burkardt, Algorithm 596: a program for a locally parameterized continuation process, *ACM Trans. Math. Softw.*, **9** (1983), 236–241.

[32] J. Sánchez Sanz and P. Getto, Numerical bifurcation analysis of physiologically structured populations: Consumer-resource, cannibalistic and trophic models, *B. Math. Biol.*, **78** (2016), 1546–84.

[33] H. L. Smith, *An introduction to delay differential equations with applications to the life sciences*, no. 57 in Texts in Applied Mathematics, Springer, New York, 2011.

[34] L. N. Trefethen, *Spectral methods in MATLAB*, Software - Environment - Tools series, SIAM, Philadelphia, 2000.

[35] L. N. Trefethen, *Approximation theory and approximation practice*, no. 128 in Other Titles in Applied Mathematics, SIAM, Philadelphia, 2013.

[36] L. T. Watson, S. C. Billups and A. Morgan, Homepack: A suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Softw.*, **13** (1987), 281–310.

E-mail address: ando.alessia@spes.uniud.it
E-mail address: dimitri.breda@uniud.it
E-mail address: francesca.scarabel@helsinki.fi