Electronic Journal of Differential Equations, Vol. 2017 (2017), No. 68, pp. 1–25.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu

NONLINEAR DIFFERENTIAL EQUATIONS WITH DEVIATING ARGUMENTS AND APPROXIMATIONS VIA A PARKER-SOCHACKI APPROACH

VINCENZO M. ISAIA

Abstract. The Parker-Sochacki method has been successful in generating approximations for a wide variety of ODEs, and even PDEs of evolution type, by achieving an autonomous polynomial vector field and implementing the Picard iteration. The intent of this article is to extend PSM to a large family of differential equations with deviating arguments. Results will be given for problems with delays which are linear in time and state independent, and also have constant initial data and nonlinear differential equations which are retarded, neutral or advanced. The goal of the proofs is to motivate a numerically efficient DDE solver. In addition, an explicit a priori error estimate that does not require derivatives of the vector field is presented. The non-constant initial data cases and the state dependent delay cases are discussed formally.

1. Introduction

In 1964, Fehlberg [6] produced a technical report for NASA that touched upon the benefits of auxiliary variables in solving ODEs. This idea appeared to go unnoticed and in 1989 Parker and Sochacki noticed the benefit of a polynomial environment for the Picard iteration, whose integral form both preserves the polynomial environment as well as computes the appropriate Taylor polynomial for the analytic solution on its domain.

The method of Parker and Sochacki was introduced in [9] and the structure of the class of ODEs was studied in [3]. The method was extended to PDEs in [10] and an explicit a priori error estimate, which do not rely on derivatives of the vector field in the ODE case, was given in [13] , see also [11]. The method was dubbed the PS method in [12], here the acronym PSM is used. This method as applied to ODEs will be reviewed in Section 2, along with the presentation of an array interpretation that will be applicable in the deviating argument case and an example which converts a non-polynomial and non-autonomous vector field to an autonomous, polynomial one.

After establishing an existence and uniqueness result via the method of successive approximations in Section 3, the central intent of this article is to adapt PSM to certain nonlinear differential equations with deviating arguments, referred to as DDEs, whose initial data is polynomial. This approach, dubbed dPSM, is presented

2010 *Mathematics Subject Classification.* 34K07, 34K40, 65L03.
Key words and phrases. Delay differential equations; lag; PSM method; method of steps; method of successive approximation; deviating argument.

Submitted April 1, 2016. Published March 8, 2017.

in Section 4. In addition, two examples are presented, one that converts a DDE's vector field to an autonomous, polynomial one and another that highlights a special case of how information can propagate.

In the case of linear in time and state independent delays, structures for the approximation to the solution across time can be established, and this is done in Section 5. The purpose of highlighting these structures is so they can be leveraged computationally. To understand the structure across iterations, which can also provide computational benefit, the array approach from Section 2 for PSM is developed in Section 6 for dPSM.

Two more examples are also given in Section 6, one that looks more carefully at converting vector fields, since this process is not unique, along with an example that shows how ''lag' dependence appears and propagates in the coefficients of the polynomial series approximation. The proof of the structure across iterations as well as explicit a priori error bounds that do not rely on higher order derivatives of the vector field. A formal discussion occurs in Section 8 concerning the use of dPSM for state dependent delays.

2. PSM overview

Given an initial value problem or IVP, based on a scalar ODE, the introduction of a polynomial vector field to the method of successive approximation allows the Picard iteration to be viewed as more than just a theoretical tool. If the Picard iteration can be shown to converge, establishing existence and uniqueness of a solution to the IVP, then a polynomial vector field will preserve the polynomial form of the initial data, which is constant, for all iterations, and a structure arises that allows the coefficients of the polynomial to be computed efficiently. In addition to requiring a polynomial form, the vector field also needs to be autonomous and the problem begun at time $t_0 = 0$, which can be achieved when $t_0 \neq 0$ via the change $\underline{t} = t - t_0$ since $\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}\underline{t}}$. Picard's then generates a Maclaurin polynomial, and shifting the time variable recovers the required Taylor polynomial of the solution.

An autonomous polynomial vector field requires the introduction of auxiliary variables that replace non-polynomial terms with polynomial terms, where nonlinearity and non-autonomous terms are exchanged for a larger system size. It can be shown that the analytic functions that arise as (a component of the) solutions to an ODE with a polynomial vector field, occupies a large portion of the class of all analytic functions, although they are not equal, see [3]. The solution of an ODE whose vector field can be transformed into a polynomial one, is dubbed *projectively polynomial*. The vector fields under consideration in this article are those which have projectively polynomial components.

This example takes a non-polynomial vector field, which is non-autonomous, begins at an arbitrary starting time $t_0 \neq 0$ and converts the vector field to one suitable for *PSM*, which used auxiliary variables to achieve the proper form for the vector field and initial time, and uses Picard's and the inherent structure to compute coefficients efficiently.

Example 2.1. Suppose $u' = \cos(u) + \cos(t)$ and $u(t_0) = u_0$ with $t_0 \neq 0$. Then introduce $V = \cos(u)$ and $T = \cos(t)$ so that $u' = V + T$ and also $W = \sin(u)$, so that $V' = -Wu' = -W(V + T)$ and $W' = Vu' = V(V + T)$. To get T', introduce $S = \sin t$ so that $T' = -S$ and $S' = T$.

For data given at $t_0 \neq 0$, Picard's iteration produces polynomials in powers of t. This does not align with the Taylor polynomial that would be in powers of $t - t_0$. To facilitate, introduce $\underline{t} = t - t_0$ and $u(t) = U(\underline{t})$ implies the method of successive approximations will now generate a Maclaurin polynomial for each component, and the Taylor polynomial for the original problem's solution is obtained by substituting $t - t_0$ for τ in the first component U. Then $u' = \cos u + \cos t$, with $u(t_0) = u_0$ is equivalent to the system

$$x_4(t) = \begin{cases} U' = V + T, \quad U(0) = u_0 \\ V' = -W(V + T), \quad V(0) = \cos(u_0) \\ W' = V(V + T), \quad W(0) = \sin(u_0) \\ T' = -S, \quad T(0) = \cos(t_0) \\ S' = T, \quad S(0) = \sin(t_0) \end{cases}$$

which has an autonomous polynomial vector field, indeed in this case, quadratic. It is possible to reduce any polynomial vector field of arbitrary degree to one that is at most degree two [3]. In addition, this reference contains a construction of an analytic function that is not projectively polynomial, i.e., its vector field cannot be transformed as per this example.

Note that if the original ODE were higher order, then in addition, the standard change of variable would have been applied to covert it to a first order system and this change produces polynomial terms as well. In addition, there exists an explicit a priori error estimate that does not involve any higher order derivatives of the original vector field, see [13].

A subtle and relevant point is that the initial data poses no issue since u_0 and t_0 are constants, which is always the case for ODEs. So, $\cos(u_0)$, $\cos(t_0)$ etc. are polynomials, in fact constants. This would not be the case, for example, if u_0 was not constant, but rather a polynomial of degree greater than zero, as is the case with some DDE problems. This will be addressed again in the example at the beginning of Section 4.

Let $\mathbf{u}_k(t) = \mathbf{\Psi} + \sum_{i=1}^{d_k} \mathbf{a}_{ki} t^i$ be PSM's approximation to an IVP with initial data Ψ. It was shown in [9] that PSM leaves invariant, in all subsequent iterations, coefficients for powers of the argument t smaller than the current number of iterations, i.e. $\mathbf{a}_{ki} = \mathbf{a}_{k+n,i}$ for any $n \geq 0$ if $i \leq k$. Then only t^{k+1} needs to have its coefficient computed; prior powers of t will have the same coefficient, while the coefficients on larger powers of t change in later iterations. A consequence discussed below is that these terms will not be computed until a later iteration. Note that only powers of t less than or equal to k are capable of producing t^{k+1} after integration.

For example, the array in Figure 2 would represent the case of a vector field consisting of a multiplication of two components. A general vector field would be a linear combination of terms, each of which could be represented by such an array. The linear combination will not disturb what is found in this particular example. Assume the current approximations are the quadratics $a_1 + b_1 t + c_1 t^2$ and $a_2 + b_2 t + c_2 t^2$. The empty row and column are included for visual purposes to accommodate the next power. The underlined terms are not be computed by this integration.

To see why powers are invariant with subsequent iteration, note that the constant terms a_1 and a_2 are fixed by the initial data for all iterations, so all entries involving

	a_1	$b_1 t$	$c_1 t^2$
a_2	$a_1 a_2 t$	$\frac{b_1 a_2}{2} t^2$	$\frac{c_1 a_2}{3} t^3$
$b_2 t$	$\frac{a_1 b_2}{2} t^2$	$\frac{b_1 b_2}{3} t^3$	$\underline{t}^4$
$c_2 t^2$	$\frac{a_1 c_2}{3} t^3$	$\underline{t}^4$	$\underline{t}^5$

FIGURE 1. Integration of vector field array - ODE

only these two coefficients are fixed for all iterations. In addition, the ordering implies each off diagonal corresponds to a unique power of t. Hence, these two points imply that the t^1 terms, namely b_1 and b_2, are fixed. This idea is extended via induction to each component [9].

To see why the underlined terms aren't computed, note that only the off diagonals above and including the main off diagonal are complete: in the example above, the t^4 off diagonal still needs the t^3 times the constant terms integrated before all the t^4 terms have been produced in the array. So the underlined t^4 terms are not be used in the vector field during the next integration. But they are included in integrations after that, when t^4 becomes the main off diagonal for the array.

Computationally, one can specialize the integration to only compute terms with powers one larger than the current degree. The approximation's degree then only increases by one per iteration. This power's coefficient is written as a scalar multiple of the convolution $\sum a_j b_{d-j}$ for an approximation of degree d. For convenience, the discussion here assumes only the relevant powers are computed, and hence, the presentation uses the language that powers greater than the current iteration will not be computed.

3. DDEs and convergence of Picard iteration

The family of differential equations with deviating arguments that are considered is

$$\begin{aligned} D^{L_1} u(t) &= f\big(t, D_{\mathbf{L}_1} u(t), D_{\mathbf{L}_2} u(\Delta(t))\big), \quad t > t_0 \\ u(t) &= \Psi(t), \quad t \in [a, t_0] \end{aligned} \tag{3.1}$$

with L_1, $L_2 \in \mathbb{N}$, $t_0 \in \mathbb{R}$ and D being the differential operator with the subscripted version indicating an ordered list of derivatives of u. The entries of the subscript specify which derivatives, for example, $D_{[0,2,4]} u(t)$ would indicate f has a $u(t)$, $u''(t)$ and $u''''(t)$ dependence. For convenience, take $\mathbf{L}_1 = [0, \dots, L_1]$ and let f absorb unused derivatives. Similarly, take $\mathbf{L}_2 = [0, \dots, L_2]$, with $L_2 \in \mathbb{N}$ independent of L_1.

Following [5] and [8], if $L_1 > L_2$ the equation is considered to be *retarded*, if $L_1 = L_2$, then the equation is considered to be *neutral*, while if $L_1 < L_2$ the equation is considered to be *advanced*. In practice, one would specify a particular $\mathbf{L}_1 \subseteq [0, \dots, L_1]$ and $\mathbf{L}_2 \subseteq [0, \dots, L_2]$ to use. It is interesting to note that the approach developed here does not need to be altered if the equation is retarded, neutral or advanced.

The delay structure $\Delta(t)$ as a function of t is assumed to be continuous and strictly increasing with $\Delta(t) < t$. Otherwise, the delay structure is left as general as possible until it becomes necessary to keep track of powers of the delay argument,

in which case only the linear structure $\Delta(t) = \sigma t - \tau$ with some $0 < \sigma \leq 1$ and $\tau \in \mathbb{R}^+$ are addressed.

A state dependent delay structure is also examined, once the role of the delay structure in the solution is realized, but this will be formal. Only one delay structure is addressed in this article, although an extension of this approach to a single problem with several delay structures is possible. In general, the delay structure $\Delta(t)$ determines the value of a. The specifics of this will appear later in this section.

The initial data needs to be polynomial to adapt PSM to the DDE case. If a problem has analytic initial data, for example see [2], one could consider approximating via a Taylor polynomial. However, two major proofs here will be relegated to constant initial data. The difficulties encountered for polynomial initial data are pointed out after the result for the constant case is established. The polynomial initial data case will be addressed in [7].

The standard change of variables converts (3.1) to a first order system, so that the Picard iteration is applied to

$$\begin{aligned} D\mathbf{u}(t) &= \mathbf{f}\big(t, \mathbf{u}(t), \mathbf{u}(\Delta(t))\big), \quad t > t_0 \\ \mathbf{u}(t) &= \mathbf{\Psi}(t), \quad t \in [a, t_0] \end{aligned} \tag{3.2}$$

where $1 \leq l \leq \max L \equiv \{L_1, L_2\}$ and $\mathbf{u} = (u^l)_{l=0}^L$ is such that $u^l = D^l u$ and $\mathbf{\Psi} = (\Psi^l)_{l=0}^L$, hence, $\Psi^l = D^l \Psi$. For constant Ψ, note that $D^l \Psi = 0$ for $l > 0$. Thus, the approximation method to come may also be applied to systems of higher order DDEs of the form (3.1).

It can be shown that if the method of successive approximations is applied to this first order system, then the sequence of approximations is uniformly convergent and and its limit solves (3.2) uniquely even if the vector field is not polynomial or autonomous. This proves the existence, uniqueness and continuity of a solution to (3.1). The proof from [4] is now adapted to the deviating argument case, and a general, but state independent, delay structure is assumed.

Let $\mathbf{f}$ in (3.2) be continuous with respect to t and Lipschitz continuous in the remaining variables on $[t_0, T^*] \times [-U, U]^{2L}$, with U, $T^* < \infty$. Such vector fields are called *admissible*. Denote by C_1^l the Lipschitz constant of admissible $\mathbf{f}$ with respect to component $u^l(t)$ of $\mathbf{u}(t)$ and denote by C_2^l the Lipschitz constant of admissible $\mathbf{f}$ with respect to component $u^l(t)$ of $\mathbf{u}(\Delta(t))$. Define $C_f \equiv \sum_{l=0}^L C_1^l + C_2^l$ and $M_f \equiv \max_l \max_{[t_0, T^*]} \mathbf{f}$. General initial data is assumed in the following proposition.

Proposition 3.1. *For admissible vector fields* $\mathbf{f}$*, let* $T \equiv \min\{T^*, UM_f^{-1}\}$ *and let* $\mathbf{\Psi}$ *in* (3.2) *be analytic and such that* $\max_l \max_{[a,T^*]} |\mathbf{\Psi}(t) - \mathbf{\Psi}(t_0)| < \infty$ *and for* $t > t_0$*, let* $\mathbf{u}_0(t) = \mathbf{\Psi}(t)$*. For each* $k \geq 0$ *define*

$$\mathbf{u}_{k+1}(t) = \mathbf{u}_k(\tau_0) + \int_{t_0}^t \mathbf{f}\big(s, \mathbf{u}_k(s), \mathbf{u}_k(\Delta(s))\big)\, \mathrm{d}s \tag{3.3}$$

then $\mathbf{u}_k(t)$ *converges uniformly to some* $\mathbf{u}_*(t)$*, which solves* (3.2) *uniquely over* $[t_0, T]$*.*

Proof. Analyticity of Ψ implies $\mathbf{u}_0 \in C^1([t_0, T])$. Applying induction, let $\mathbf{u}_k \in C^1([t_0, T])$. Since each component of $\mathbf{f}$ is continuous, the Fundamental Theorem of Calculus implies $\mathbf{u}_{k+1} \in C^1([t_0, T])$, so $\mathbf{u}_k \in C^1([t_0, T])$ for $k \in \mathbb{Z}^+$.

Denote u_k^l as a component of $\mathbf{u}_k(t)$ and denote Ψ^l and f^l as components of $\mathbf{\Psi}$ and $\mathbf{f}$, respectively. Denote $f_k^l(t) = f^l(t, \mathbf{u}_k(t), \mathbf{u}_k(\Delta(t))$. One has for every $1 \leq l \leq L$

that

$$|u_1^l - u_0^l|(t) = \left|\Psi^l(t_0) + \int_{t_0}^t f_0^l(s)\,\mathrm{d}s - \Psi^l(t)\right| \leq M_\Psi + M_f(t-t_0)$$

which follows since $|\Psi^l(t) - \Psi^l(t_0)| \leq M_\Psi \equiv \max_l \max_{[a,T]} |\mathbf{\Psi}(t) - \mathbf{\Psi}(t_0)| < \infty$ by hypothesis.

Consider $|u_1^l - u_0^l|(\Delta(t))$. If $\Delta(t) \leq t_0$, then

$$|u_1^l - u_0^l|(\Delta(t)) = |\Psi^l(\Delta(t)) - \Psi^l(\Delta(t))| = 0$$

otherwise, if $\Delta(t) > t_0$, then

$$|u_1^l - u_0^l|(\Delta(t)) \leq \left|\Psi^l(t_0) + \int_{t_0}^{\Delta(t)} f_0^l(s)\,\mathrm{d}s - \Psi^l(\Delta(t))\right| \leq M_\Psi + M_f(t-t_0)$$

since $\Delta(t) < t$ by hypothesis. So both $|u_1^l - u_0^l|(t)$ and $|u_1^l - u_0^l|(\Delta(t))$ are bound by $M_\Psi + M_f(t-t_0)$ for every $1 \leq l \leq L$.

From the Lipschitz continuity of $\mathbf{f}$ and the previous bound, one has

$$|u_2^l - u_1^l|(t) \leq \int_{t_0}^t |f_1^l - f_0^l|(s)\,\mathrm{d}s \leq \int_{t_0}^t C_f\big(M_\Psi + M_f(s-t_0)\big)\,\mathrm{d}s \tag{3.4}$$

so $|u_2^l - u_1^l|(t) \leq C_f\left(\frac{1}{2}M_f(t-t_0)^2 + M_\Psi(t-t_0)\right)$ for every $1 \leq l \leq L$. Moving to the delayed terms, if $\Delta(t) < t_0$ then $|u_2^l - u_1^l|(\Delta(t)) = 0$ as before, otherwise

$$|u_2^l - u_1^l|(\Delta(t)) \leq \int_{t_0}^{\Delta(t)} |f_1^l - f_0^l|(s)\,\mathrm{d}s \leq \int_{t_0}^t |f_1^l - f_0^l|(s)\,\mathrm{d}s$$

and so $|u_2^l - u_1^l|(\Delta(t))$ along with $|u_2^l - u_1^l|(t)$ are both bound by $C_f\big(\frac{1}{2}M_f(t-t_0)^2 + M_\Psi(t-t_0)\big)$. Continuing in this fashion shows that

$$\begin{aligned} |u_{k+1}^l - u_k^l|(t) &\leq \int_{t_0}^t |f_k^l - f_{k-1}^l|(s)\,\mathrm{d}s \\ &\leq \int_{t_0}^t C_f^k\Big(M_\Psi \frac{(t-t_0)^{k-1}}{k-1!} + M_f\frac{(t-t_0)^k}{k!}\Big)\,\mathrm{d}s \end{aligned} \tag{3.5}$$

subsequently, one has

$$|u_{k+1}^l - u_k^l|(t) \leq C_f^k\Big(M_\Psi\frac{(t-t_0)^k}{k!} + M_f\frac{(t-t_0)^{k+1}}{k+1!}\Big)$$

which implies that for every component $1 \leq l \leq L$,

$$\sum_{k=0}^\infty |u_{k+1}^l - u_k^l|(t) \leq (M_\Psi + M_f(t-t_0))\sum_{k=0}^\infty C_f^k\frac{(t-t_0)^k}{k!}$$

and one can conclude that for each component u_k^l,

$$u_{k+1}^l = \sum_k u_{k+1}^l - u_k^l \leq (M_\Psi + M_f(t-t_0))\,e^{C_f(t-t_0)}$$

and so $u_k^l \to u_*^l$ uniformly by Weierstrass M-test with u_*^l continuous. Hence (3.3) holds in the limit, which is equivalent to $\mathbf{u}_* = (u_*^l)_{l=1}^L$ solving (3.2) since the uniform convergence allows the limit to be exchanged with the integral. Using the bounds developed above and adapting the argument from [8], it is straightforward to show that $\mathbf{u}_*$ is unique. □

The method of successive approximation is a global approximation, in that it updates $\mathbf{u}_k$ over $[\tau_0, T]$. However, such a global update is computationally difficult with a DDE. On the other hand, the method of steps, see [1] for example, is computationally friendly but at the expense of being a local approximation. For the method of steps, the approach is to reduce (3.1) to a sequence of ODEs: given t_0, then prior to a certain point in time, to be determined, terms like $\mathbf{u}(\Delta(t))$ are found in terms of Ψ, which is known, so that (3.2) defaults to an ODE, although it may have non-constant coefficients and/or non-homogeneous terms.

Define $\tau_0 \equiv t_0$ and denote by τ_1 the point in time prior to which $\mathbf{u}(\Delta(t))$ can be found in terms of Ψ. By standard theory, if the vector field is Lipschitz continuous, then the unique solution that ensues would be valid over $[\tau_0, \tau_1]$. Denote this solution by $\mathbf{u}_1(t)$. In order for the delay term $\mathbf{u}(\Delta(t))$ in the vector field to be known, $\Delta(t)$ must be less than or equal to τ_0 over $[\tau_0, \tau_1]$. Using the fact that Δ is strictly increasing and solving $\Delta(t) = \tau_0$ for t would determine τ_1. In addition, $\Delta(\tau_0)$ represents the farthest back in time information is needed for the vector field, and so $\Delta(\tau_0) = \tau_{-1} = a$.

With the problem solved over $[\tau_0, \tau_1]$, consider τ_m defined as the solution to $\Delta(t) = \tau_{m-1}$. Define $\mathcal{T} \equiv \{\tau_m : \Delta(\tau_m) = \tau_{m-1}, m \in \mathbb{Z}_0^+\}$ and let $\mathbf{u}_m(t)$ be the unique solution to (3.2) over $[\tau_m, \tau_{m+1}]$. Consider solving (3.2) over $[\tau_{m+1}, \tau_{m+2}]$ to get $\mathbf{u}_{m+1}$. Then the delay terms $D_{\mathbf{L}_2}\mathbf{u}_{m+1}(\Delta(t))$ for $t > \tau_{m+1}$ would involve $\mathbf{u}_m(t)$, which is known. Denote the unique solution by $\mathbf{u}_{m+1}(t)$. Hence

$$\mathbf{u}(t) = \{\mathbf{u}_m(t) \text{ over } [\tau_m, \tau_{m+1}] \text{ for } m \in \mathbb{Z}^+\} \tag{3.6}$$

solves (3.2). The overlap in endpoints for the τ-steps is not an issue due to the continuity of the solution to (3.2) as per Proposition 3.1.

The method of steps is represented by the intervals $[\tau_m, \tau_{m+1}]$, or *τ-steps*, and computationally this approach is restrictive in the sense that it updates the solution one τ-step at a time: the solution needs to be known in $[\tau_{m-1}, \tau_m]$ before it can be determined in $[\tau_m, \tau_{m+1}]$.

Anticipating the Picard iteration, which will cause the delay structure to be composed with itself, notice that $\Delta : [\tau_m, \tau_{m+1}] \to [\tau_{m-1}, \tau_m]$ satisfies

$$\Delta_m(t) \equiv \Delta \circ \cdots \circ \Delta(t) \tag{3.7}$$

with $\Delta_1(t) = \Delta(t)$ and $m > 0$. Define $\Delta_0(t) \equiv t = t - \tau_0$ and $\Delta_{-1}(t) \equiv t - \tau_{-1}$, where this last definition is for notational convenience. Then $\Delta_m(\tau_m) = 0$ for all $m \geq -1$. Note that the linear in time and state independent delay

$$\Delta(t) = \sigma t + \tau_{-1} \tag{3.8}$$

with $0 < \sigma \leq 1$ and *lag* $\tau_{-1} < 0$, includes the constant lag case: $\Delta(t) = t - \tau$ for some fixed $\tau > 0$. In this case, one has $\tau_m = m\tau$ for all $m \geq 0$, while $\Delta_m(t)$ reduces to $t - m\tau$.

4. dPSM and examples

The PSM philosophy of creating a polynomial environment on which Picard iteration thrives is now adapted to (3.2). It is important to note that a change of variable for the delay terms can come for free, because they may not require an evolution line in the system. Once $u(t)$ evolves, one additional functional evaluation will give the evolution of $u(\Delta(t))$, and taking derivatives will also evolve $u'(\Delta(t))$,

$u''(\Delta(t))$ etc. as needed because the Picard iteration explicitly depends on previously computed information. This contributes to making the distinction between retarded, neutral and advanced unnecessary as far as implementing this approach is concerned.

Example 4.1. Ignoring the initial information, suppose $u'(t) = \cos(u(t)) + u(t-\tau) + u'(t-\tau)$ with $t \geq t_0 \neq 0$. Following the example in Section 2, introduce $V = \cos(U)$ and $W = \sin(U)$ along with $\underline{t} = t - t_0$, $u(t) = U(\underline{t})$, $R = u(t-\tau)$ and $S = u'(t-\tau)$, so that $U' = V + R + S$, $V' = -WU' = -W(V + R + S)$ and $W' = VU' = V(V + R + S)$. Then $u' = \cos u + u(t-\tau) + u'(t-\tau)$ is equivalent to the autonomous polynomial system

$$x_4(t) = \begin{cases} U' = V + R + S \\ V' = -W(V + R + S) \\ W' = V(V + R + S) \end{cases} \tag{4.1}$$

The explicit nature of successive approximations allows the update of R via $R' = S$ to be replaced with a function evaluation from U's evolution, $R(\underline{t}) = u(t-\tau) = U(\underline{t} - \tau)$. The system may be handled in its current foem, since the vector fields are evaluated at the previous approximation, so only R's (and S's) initial information is needed.

There may be a price to pay for the V and W change of variable: if the initial data is not constant then $\cos(U)$ and $\sin(U)$ will not be polynomials. This could be handled by using an appropriate Maclaurin polynomial, where the error could possibly be controlled through the choice of the polynomial's degree. Computationally, there is also a price to pay for non-constant initial data in general.

Parallel to [9], the vector field needs to be autonomous, and evolution needs to begin at $t_0 = 0$. A change of variable as per the previous example will account for this, and the explicit t dependence can be dropped from $\mathbf{f}$'s argument with no other change in notation. The variable $\underline{t}$, as per the last example, is relabeled as t and the set $\mathcal{T}$ is then determined using this new time variable. Invoking the notation for the method of steps, the IVP (3.2) becomes, with $t_0 = \tau_0 = 0$,

$$\begin{gathered} \mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t), \mathbf{u}(\Delta(t))), \quad t > \tau_0 \\ \mathbf{u}(t) = \mathbf{\Psi}(t), \quad t \in [\tau_{-1}, \tau_0] \end{gathered}$$

where $\mathbf{u}$ now includes not only derivatives of u but the auxiliary variables necessary to make the vector field be polynomial, autonomous and have $t_0 = 0$. Although an ordering for the vector is important in practice, there is no reliance on ordering in the proofs, hence one is not specified.

To compute the Picard iteration over $[\tau_0, T]$, the method of steps is used and a very nice computational structure appears once the initial data $\mathbf{\Psi}$ is extended over the entire interval $[\tau_0, T]$. One has

$$\mathbf{u}_{k+1}(t) = \mathbf{u}_k(\tau_0) + \int_{\tau_0}^{t} \mathbf{f}(\mathbf{u}_k(s), \mathbf{u}_k(\Delta(s)))\, \mathrm{d}s \tag{4.2}$$

for $t \in [\tau_0, T]$ and $k \geq 0$. Combining with the method of steps notation, let $\mathbf{u}_k(t) = \mathbf{u}_{km}(t)$ over $[\tau_m, \tau_{m+1}]$ with $m \in \mathbb{Z}_0^+$. Then (4.2) becomes *dPSM*, or the

delayed Parker-Sochacki method, as stated.

$$\mathbf{u}_{k+1,\underline{m}}(t) = \mathbf{\Psi}(\tau_0) + \sum_{m=0}^{\underline{m}-1} \int_{\tau_m}^{\tau_{m+1}} \mathbf{f}_{km}(s)\,\mathrm{d}s + \int_{\tau_{\underline{m}}}^{t} \mathbf{f}_{k,\underline{m}}(s)\,\mathrm{d}s \tag{4.3}$$

The iterative step of (4.3) assumes $\mathbf{f}_{k,\underline{m}}(s) \equiv \mathbf{f}(\mathbf{u}_{k,\underline{m}}(s), \mathbf{u}_{k,\underline{m}-1}(\Delta(s)))$. In the next section, it is shown that this calculation can be modified and performed for only $\underline{m} = k+1$, and it will 'contain' $\mathbf{u}_{k,\underline{m}}(t)$ for each $0 \le \underline{m} < k+1$. A few comments about notation: some indices are comma separated for clarity, and to streamline notation in later sections, m is changed to $\underline{m}$ so that m can be used as a local index for $\underline{m}$.

If an admissible vector field has at least one delay term in each component, i.e. for each component f^l of $\mathbf{f}$, the vector of partial derivatives with respect to to the delay terms $f^l_{\mathbf{u}(\Delta(t))} \neq \mathbf{0}$, the vector field $\mathbf{f}$ is called *fully delayed*. For the case of a fully delayed vector field and the linear delay in (3.8), it can be established that the following piecewise structure represents how information propagates for any component l of $\mathbf{u}_k$, which happens to be the fastest possible propagation,

$$u^l_k(t) = \begin{cases} \Psi(\tau_0) + p_{k0}(\Delta_0(t)) & t \in [\tau_0, \tau_1] \\ \Psi(\tau_0) + p_{k0}(\Delta_0(t)) + p_{k1}(\Delta(t)) & t \in [\tau_1, \tau_2] \\ \Psi(\tau_0) + p_{k0}(\Delta_0(t)) + p_{k1}(\Delta(t)) + p_{k2}(\Delta_2(t)) & t \in [\tau_2, \tau_3] \\ \dots & \\ \Psi(\tau_0) + p_{k0}(\Delta_0(t)) + \dots + p_{k,k-1}(\Delta_{k-1}(t)) & t \in [\tau_{k-1}, \tau_k] \\ \Psi(\tau_0) + p_{k0}(\Delta_0(t)) + \dots + p_{k,k-1}(\Delta_{k-1}(t)) & t \in [\tau_k, T] \end{cases} \tag{4.4}$$

where p_{km} is a polynomial whose coefficients depend on the iteration k and which delay structure $\Delta_m(t)$ is in the argument, but not the τ-step that contains t. In particular, in each τ-step, a new delay structure appears, and earlier delay structures have invariant coefficients, so that the polynomial $\mathbf{p}_{km}$ has coefficients that depend on m in its argument Δ_m but not on m where $t \in [\tau_m, \tau_{m+1}]$. However, the τ-step does affect the appearance of the delay structure, in that Δ_m first appears when $t \in [\tau_m, \tau_{m+1}]$. In compact notation,

$$u^l_{k,\underline{m}}(t) = \Big(\sum_{m=-1}^{\underline{m}_k} p^l_{km}(\Delta_m(t)), \quad t \in [\tau_{\underline{m}}, \tau_{\underline{m}+1}] \text{ with } p^l_{km}(z) = \sum_{i=m+1}^{d} a^i_{km} z^i \Big)$$

for $\underline{m}_k \equiv \min\{\underline{m}, k-1\}$ where the constant term $\Psi(\tau_0)$ is hidden in the $\Delta_{-1}(t)$ terms, with $d = 0$ if $\underline{m} = -1$. Theorem 5.3 in the next section will establish this structure. In addition, the polynomials also exhibit an invariance across iterations, but this is a little more complicated than the PSM version and will be handled in Theorem 7.1.

In (4.4), note that the functional form in $[\tau_k, T]$ is the same as the functional form in $[\tau_{k-1}, \tau_k]$. All the currently computed information has propagated to $[\tau_{k-1}, \tau_k]$ and its replication in $[\tau_k, T]$ is just a mimic of extending the initial data to $[\tau_0, T]$. This is helpful because of the invariance across τ-steps and iterations will show these terms belong in the solution eventually and thus they provide a good forward approximation in regions where information has yet to propagate, thereby extending the 'local' method of steps.

For general delay structures, when (4.3) is enacted on a fully delayed vector field, there is a nonzero finite speed of propagation of the initial information $\mathbf{\Psi}$

into $[\tau_0, T]$ with each iteration, in particular, one τ-step per iteration. This will manifest in an element $\tau_*^k \in \mathcal{T}$ with m^* such that $\tau_*^k = \tau_{m^*}$, for which the solution will have a different functional dependence in the τ-step $[\tau_{m^*}, \tau_{m^*+1}]$ than it did in $[\tau_{m^*-1}, \tau_{m^*}]$, and m^* will be the largest such m for which this is true.

Call this largest element of $\mathcal{T}$ for which a change in functional form occurs across that element the *extension boundary*. It will also imply that it is the smallest τ_m for which $\mathbf{u}_{km}$ has the same functional dependence as $\mathbf{u}_{k,m+1}$. Here is an example that shows a glimpse of extension boundary behavior when the vector field is not fully delayed.

Example 4.2. Consider $x' = y$, $y' = z$ and $z' = x(t-\tau)$, each with constant initial data of unity, and note that the right hand side of x and y lack delay terms, and as such the vector field for this problem is not fully delayed. Using (4.3) one can compute $x_5(t)$, $y_5(t)$, and $z_5(t)$.

$$x_5(t) = \begin{cases} x_{50}(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} & t \in [\tau_0, \tau_1] \\ x_{51}(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-\tau)^5}{5!} & t \in [\tau_1, \tau_2] \\ x_{52}(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-\tau)^5}{5!} & t \in [\tau_2, T] \end{cases}$$

$$y_5(t) = \begin{cases} y_{50}(t) = 1 + t + \frac{t^2}{2!} & t \in [\tau_0, \tau_1] \\ y_{51}(t) = 1 + t + \frac{t^2}{2!} + \frac{(t-\tau)^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-\tau)^5}{5!} & t \in [\tau_1, \tau_2] \\ y_{52}(t) = 1 + t + \frac{t^2}{2!} + \frac{(t-\tau)^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-\tau)^5}{5!} & t \in [\tau_2, T] \end{cases}$$

$$z_5(t) = \begin{cases} z_{50}(t) = 1 + t & t \in [\tau_0, \tau_1] \\ z_{51}(t) = 1 + t + \frac{(t-\tau)^2}{2!} + \frac{(t-\tau)^3}{3!} + \frac{(t-\tau)^4}{4!} & t \in [\tau_1, \tau_2] \\ z_{52}(t) = 1 + t + \frac{(t-\tau)^2}{2!} + \frac{(t-\tau)^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-2\tau)^5}{5!} & t \in [\tau_2, T] \end{cases}$$

A partial calculation of $z_5(t)$ will be given, based on

$$x_4(t) = \begin{cases} x_{40}(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} & t \in [\tau_0, \tau_1] \\ x_{41}(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{(t-\tau)^4}{4!} & t \in [\tau_1, T] \end{cases}$$

which will be computed over $[\tau_2, \tau_3] = [2\tau, 3\tau]$, i.e. $z_{52}(t)$,

$$\begin{aligned} z_{52}(t) &= 1 + \int_0^t x_4(s-\tau)\,\mathrm{d}s \\ &= 1 + \int_0^\tau \Psi(s-\tau)\,\mathrm{d}s + \int_\tau^{2\tau} x_{40}(s-\tau)\,\mathrm{d}s + \int_{2\tau}^t x_{41}(s-\tau)\,\mathrm{d}s \\ &= 1 + \int_0^\tau 1\,\mathrm{d}s + \int_\tau^t 1 + (s-\tau) + \frac{(s-\tau)^2}{2!} + \frac{(s-\tau)^3}{3!}\,\mathrm{d}s \\ &\quad + \int_{2\tau}^t 1 + (s-\tau) + \frac{(s-\tau)^2}{2!} + \frac{(s-\tau)^3}{3!} + \frac{(s-2\tau)^4}{4!}\,\mathrm{d}s \\ &\overset{*}{=} 1 + \int_0^t 1\,\mathrm{d}s + \int_\tau^t (s-\tau) + \frac{(s-\tau)^2}{2!} + \frac{(s-\tau)^3}{3}\,\mathrm{d}s + \int_{2\tau}^t \frac{(s-2\tau)^4}{4!}\,\mathrm{d}s \\ &= 1 + t + \frac{(t-\tau)^2}{2!} + \frac{(t-\tau)^3}{3!} + \frac{(t-\tau)^4}{4!} + \frac{(t-2\tau)^5}{5!}\,. \end{aligned}$$

It will be shown in the next section that the regrouping of integrals that occurs in the fourth equality above ($\overset{*}{=}$) will occur in general. After proving the speed of

the extension boundary with respect to k for a fully delayed vector field, a formal discussion of this example's extension boundary propagation will be given in the next section.

5. dPSM Results - I

The speed with respect to iterations k will be proved for a general state independent delay and for a fully delayed vector field. Note that the following result is independent of the initial data, constant or non-constant.

Theorem 5.1. *Assuming $\mathbf{u}_k$ does not solve (3.2) and $\mathbf{f}$ is a fully delayed vector field, then extension boundary for $\mathbf{u}_k$ is $\tau_*^k = \tau_{k-1}$, $k \geq 1$, i.e. $\mathbf{u}_{k,k-1}(t)$ has the same functional form as $\mathbf{u}_{km}(t)$ for every $m > k-1$, but not the same functional form as $\mathbf{u}_{k,k-2}(t)$.*

Proof. Since the data begins at τ_{-1}, there is a change in functional form from non-existent to Ψ as time crosses over τ_{-1} for $\mathbf{u}_0$, and this is the largest element of $\mathcal{T}$ for which this is true. If $\mathbf{\Psi}$ solves (3.2), then iterating further is not necessary, hence the assumption that $\mathbf{u}_k$ doesn't solve (3.2). Otherwise, $\tau_*^1 = \tau_0$ for $\mathbf{u}_1$ since

$$\mathbf{u}_{10}(t) = \mathbf{\Psi}(\tau_0) + \int_{\tau_0}^{t} \mathbf{f}(\mathbf{\Psi}(s), \mathbf{\Psi}(\Delta(s)))\,\mathrm{d}s \neq \mathbf{\Psi}(t)$$

and there is a change in functional form from Ψ to $\mathbf{u}_{10}$ across $t = \tau_0$.

By extending the given data in $[\tau_{-1}, \tau_0]$ to each τ-step $[\tau_m, \tau_{m+1}]$ with $m \in \mathbb{Z}_0^+$, the integral in (4.2) for $\mathbf{u}_1$ can be determined in each subsequent $[\tau_{m+1}, \tau_{m+2}]$ as well via (4.3)

$$\mathbf{u}_{1\underline{m}}(t) = \mathbf{u}_{1,\underline{m}-1}(\tau_{\underline{m}}) + \int_{\tau_{\underline{m}}}^{t} \mathbf{f}\big(\mathbf{u}_{0,\underline{m}}(s), \mathbf{u}_{0,\underline{m}-1}(\Delta(s))\big)\,\mathrm{d}s \tag{5.1}$$

Since $\mathbf{u}_{0,\underline{m}} = \mathbf{\Psi}(t)$ for each $\underline{m} \geq 0$, then (5.1) has the same integrand for $m \geq 0$, hence $\mathbf{u}_{1,\underline{m}}$ has the same functional form as $\mathbf{u}_{10}$ for any $\underline{m} \geq 0$. Technically, $\mathbf{u}_{1,\underline{m}}$ and $\mathbf{u}_{10}$ cannot be equated when $\underline{m} \geq 1$, even though they have the same functional form, because of the differing domains: $[\tau_{\underline{m}}, \tau_{\underline{m}+1}]$ versus $[\tau_0, \tau_1]$.

By hypothesis, if $\tau_*^k = \tau_{k-1}$ for $\mathbf{u}_k$ then consider (4.3) with $\underline{m} = k-1$

$$\mathbf{u}_{k+1,k-1}(t) = \mathbf{u}_{k+1,k-2}(\tau_{k-1}) + \int_{\tau_{k-1}}^{t} \mathbf{f}\left(\mathbf{u}_{k,k-1}(s), \mathbf{u}_{k,k-2}(\Delta(s))\right)\,\mathrm{d}s\,. \tag{5.2}$$

When considering $\mathbf{u}_{k+1,k}$, this does not have the same integrand as $\mathbf{u}_{k+1,k-1}$,

$$\mathbf{u}_{k+1,k}(t) = \mathbf{u}_{k+1,k-1}(\tau_k) + \int_{\tau_k}^{t} \mathbf{f}\left(\mathbf{u}_{k,k}(s), \mathbf{u}_{k,k-1}(\Delta(s))\right)\,\mathrm{d}s \tag{5.3}$$

since the extension boundary is at τ_{k-1} for $\mathbf{u}_k$ implying that $\mathbf{u}_{k,k-1}$ and $\mathbf{u}_{k,k}$ have the same functional form, but that $\mathbf{u}_{k,k-1}$ and $\mathbf{u}_{k,k-2}$ will not. Since the partials of $\mathbf{f}$ with respect to to the delay terms are not all zero by hypothesis, and $\mathbf{u}_k$ does not solve (3.2), then (5.2) has a different integrand, and hence functional form than (5.3). The integral for $\mathbf{u}_{k+1,k+1}$ looks like

$$\mathbf{u}_{k+1,k+1}(t) = \mathbf{u}_{k+1,k}(\tau_{k+1}) + \int_{\tau_{k+1}}^{t} \mathbf{f}\left(\mathbf{u}_{k,k+1}(s), \mathbf{u}_{k,k}(\Delta(s))\right)\,\mathrm{d}s$$

and since the extension boundary for $\mathbf{u}_k$ is at τ_{k-1} for $\mathbf{u}_k$, this is the same integrand as in (5.3). Hence there is a change in functional form across $t = \tau_k$ but not across $t = \tau_{k+1}$ and so the extension boundary has moved to τ_k for $\mathbf{u}_{k+1}$. □

Example 4.2 (cont.) Because of the lack of a delay term in each component's right hand side, it can be seen easily that the information has not reached $[4\tau, 5\tau]$ as would have been predicted by Theorem 5.1. Suppress iteration dependence and label the extension boundary for component x_k by τ_*^x and similarly for components y_k and z_k. The movement of the extension boundary can be described heuristically as follows: assuming Ψ does not solve the DDE, upon computing the first iteration, $\tau_*^x = \tau_*^y = \tau_*^z = \tau_0$ and all three move as predicted by Theorem 5.1.

However, computing the second iteration, both x and y are governed by ODEs rather than DDEs, and so the information in $[0, \tau]$ is updated, but lack of a delay term in the vector field does not create new delay terms in $[\tau, 2\tau]$, hence $\tau_*^x = \tau_*^y = \tau_0$. In contrast, z does have a delay term in its vector field and so, the second iteration does produce new delay terms in $[\tau, 2\tau]$ and so $\tau_*^z = \tau_1$ as Theorem 5.1 would indicate for a fully delayed vector field.

Now, these new delay terms in z during the computation of the third iteration produce new delay terms for y in $[\tau, 2\tau]$ since the vector field for y is z dependent, and so $\tau_*^y = \tau$. However, x, whose vector field is y dependent, will have to wait one more iteration to see a change due to the new delay terms created in z during the second iteration, and only its info in $[0, \tau]$ updates. Because x has not changed, one finds that z will not see new delay terms and $\tau_*^z = \tau$.

During the fourth iteration, the new delay terms in z from the second iteration finally reach x and $\tau_*^x = \tau$, while both y and z only update their previous information and $\tau_*^y = \tau_*^z = \tau$. Hence, after the initial movement in the first iteration, it requires three iterations to have all components' extension boundaries move again. □

The speed of the extension boundary with respect to k is independent of the form of $\Delta(t)$. The choice of $\Delta(t)$ of the form given in (3.8) is only necessary to make sorting terms arising from the integration manageable in a general setting. Given a specific polynomial form for $\Delta(t)$, one may be able to extend the proofs here to the specific case, upon knowing what terms to track. On the flip side, with $\Delta(t)$ given by (3.8), this will show that the only special treatment time dependent delay structures require are how to sort their terms.

The following lemma's result is not surprising, but it helps track arbitrary nonlinear interaction terms in the vector field. Beyond that, the ramifications of the result are important from a computational aspect. Integration in the following form, which occurs eventually in all nonlinear problems, implies a specific power for a specific delay structure $\Delta_m(t)$ contributes a range of powers in the result.

Lemma 5.2. *Given α, $\beta \in \mathbb{Z}^+$ and $\Delta(t)$ as in (3.8) and $\Delta_m(t)$ in (3.7), let*

$$I(t) = \int_{\tau_m}^{t} \Delta_m^{\alpha}(s)\Delta_{\underline{m}}^{\beta}(s)\,\mathrm{d}s \tag{5.4}$$

with $m > \underline{m}$ so that $\tau_m > \tau_{\underline{m}}$. Then $I(t)$ is a polynomial with argument $\Delta_m(t)$ which can be written in the form

$$\sum_{i=\alpha+1}^{\alpha+\beta+1} c_i \Delta_m^i(t) \tag{5.5}$$

with c_i constant with respect to t.

Proof. Let $\tau \equiv -\tau_{-1}$ and some straightforward computation shows that $\tau_m = \tau\sum_{q=1}^{m}\sigma^{-q}$ and $\Delta_m(t) = \sigma^m(t-\tau_m)$. Rather than Taylor expand the integrand, consider the substitution given by $S = \Delta_m(s),\ \mathrm{d}S = \sigma^m\,\mathrm{d}s$ in (5.4). Rewriting

$$\tau_m - \tau_{\underline{m}} = \tau\sum_{q=\underline{m}+1}^{m}\sigma^{-q} \equiv \gamma_{m,\underline{m}}\tau$$

and using the binomial expansion on $(\sigma^{\underline{m}-m}S + \sigma^{\underline{m}}\gamma_{m,\underline{m}}\tau)^\beta$, we have

$$I(t) = (\sigma^{\beta(\underline{m}-m)-m})\sum_{i=0}^{\beta}\binom{\beta}{i}\frac{(\gamma_{m,\underline{m}}\tau)^{\beta-i}}{i+\alpha+1}(\sigma^m(t-\tau_m))^{i+\alpha+1}$$

and note that the powers of τ are independent of α. Upon shifting the index, letting

$$c_i = (\sigma^{\beta(\underline{m}-m)-m})\sum_{i=\alpha+1}^{\alpha+\beta+1}\binom{\beta}{i-\alpha-1}\frac{(\gamma_{m,\underline{m}}\tau)^{\alpha+\beta+1-i}}{i}$$

and exchanging $\sigma^m(t-\tau_m)$ with $\Delta_m(t)$, one recovers (5.5). □

For the constant lag case, $\Delta_m(t) = t - m\tau$ and $\gamma_{m,\underline{m}} = m - \underline{m}$. Nonlinear interaction in the vector field will cause (5.4) and so the coefficients in the power series for the approximation will be lag dependent. These lag dependent coefficients will also occur when the initial data in (3.2) is not constant.

The next theorem will show that there is a particular structure to the Picard iterations with respect to the τ-steps during a fixed iteration. This is simpler to demonstrate than the structure with respect to iterations k.

Theorem 5.3. *If $\mathbf{u}_{0,\underline{m}}(t) = \mathbf{\Psi}$, i.e. constant initial data, for $\underline{m} \in \mathbb{Z}$ with $\underline{m} \geq -1$, then define $\mathbf{u}_{k,\underline{m}}(t)$ by (4.3) for $k \geq 1$ with $\Delta(t)$ as in (3.8). If $t \in [\tau_{\underline{m}}, \tau_{\underline{m}+1}] \subset [\tau_0, T]$, $\mathbf{f}$ is fully delayed and defining $\mathbf{a}^0_{k,-1} \equiv \Psi(\tau_0)$, then $\mathbf{u}_{k,\underline{m}}(t)$ has the form for $k \geq 0$*

$$\mathbf{u}_{k,\underline{m}}(t) = \sum_{m=-1}^{\underline{m}_k}\mathbf{p}_{km}(t) \tag{5.6}$$

where $\underline{m}_k \equiv \min\{\underline{m}, k-1\}$ and $\Delta_0(t) = t - \tau_0$ and $\Delta_{-1}(t) = t - \tau_{-1}$. In addition, $\mathbf{p}_{km}$ is a polynomial of the form

$$\mathbf{p}_{km}(t) = \sum_{i=m+1}^{d}\mathbf{a}^i_{km}\Delta^i_m(t) \tag{5.7}$$

with $d \in \mathbb{Z}^+$, $d \geq k$ if $m \geq 0$, $d = 0$ if $m = -1$, and the domain of $\mathbf{p}_{km}$ is $[\tau_m, T]$.

An important point is that the only $\underline{m}$ dependence in $\mathbf{u}_{k,\underline{m}}$ occurs from $\underline{m}_k$.

Proof of Theorem 5.3. Clearly, starting with polynomial (constant) initial data and enacting (4.3) implies $\mathbf{u}_k$ is polynomial for any polynomial vector field and any $k \geq 0$. Hence the focus is on the specific forms of the polynomials in (5.6) given by (5.7).

If the the initial data $\mathbf{\Psi}$ is constant, then (5.1) yields for all $\underline{m} \geq 0$

$$\mathbf{u}_{1\underline{m}}(t) = \mathbf{\Psi}(\tau_0) + \mathbf{z}(t-\tau_0) = \mathbf{\Psi}(\tau_0) + \mathbf{p}_{10}(t)$$

with $\mathbf{a}_{10}^1 = \mathbf{z}$ for every $[\tau_{\underline{m}}, \tau_{\underline{m}+1}]$, where $\mathbf{z} = \mathbf{f}(\mathbf{\Psi}, \mathbf{\Psi})$ is constant since Ψ constant. It follows that both (5.6) and (5.7) hold with $d = 1$ when $\underline{m} \geq 0$ and $d = 0$ when $\underline{m} = -1$ for each τ-step of $\mathbf{u}_1(t)$.

Some convenient notation will now be introduced. Since $\mathbf{f}$ is polynomial, its full Taylor series is a finite sum with respect to any center. Denote the Taylor series for $\mathbf{f}(a + c, b + d)$ without its constant term by $\mathbf{f}^*(a, b; c, d)$, which represents a power series of the form

$$\mathbf{f}^*(a, b; c, d) \equiv \sum_{(i,j)\neq (0,0)}^{n} \sum^{n} C_{ij}(a, b)c^i d^j$$

for some set of constants C_{ij} which depend on a and b. In addition, we denote

$$\mathbf{f}_{k,\underline{m}}(s) \equiv \mathbf{f}\left(\mathbf{u}_{k,\underline{m}}(s), \mathbf{u}_{k,\underline{m}-1}(\Delta(s))\right)$$

as well as

$$\mathbf{f}^*_{k,\underline{m}-1}(s) \equiv \mathbf{f}^*\left(\mathbf{u}^*_{k,\underline{m}-1}(s), \mathbf{u}^*_{k,\underline{m}-2}(\Delta(s)); \mathbf{p}_{k,\underline{m}}(s), \mathbf{p}_{k,\underline{m}-1}(\Delta(s))\right)$$

Using (5.6) and (5.7) as hypothesis for $\mathbf{u}_k$, consider $\mathbf{u}_{k+1}$ during any τ-step $[\tau_{\underline{m}}, \tau_{\underline{m}+1}]$ with $\underline{m} \leq k - 1$, so that (4.3) implies

$$\mathbf{u}_{k+1,\underline{m}}(t) = \mathbf{\Psi}(\tau_0) + \sum_{m=0}^{\underline{m}-1} \int_{\tau_m}^{\tau_{m+1}} \mathbf{f}_{km}(s)\,\mathrm{d}s + \int_{\tau_{\underline{m}}}^{t} \mathbf{f}_{k,\underline{m}}(s)\,\mathrm{d}s$$

and after a Taylor expansion around the point $\left(\mathbf{u}_{k,\underline{m}-1}(s), \mathbf{u}_{k,\underline{m}-2}(\Delta(s))\right)$, we have

$$\begin{aligned}\mathbf{u}_{k+1,\underline{m}}(t) &= \mathbf{\Psi}(\tau_0) + \sum_{m=0}^{\underline{m}-1} \int_{\tau_m}^{\tau_{m+1}} \mathbf{f}_{km}(s)\,\mathrm{d}s + \int_{\tau_{\underline{m}}}^{t} \mathbf{f}_{k,\underline{m}-1}(s) + \mathbf{f}^*_{k,\underline{m}-1}(s)\,\mathrm{d}s \\ &\equiv I_\sigma(t) + I_{\underline{m}}(t)\end{aligned}$$

Note that the first term in the integrand of $I_{\underline{m}}(t)$ has precisely the same functional form as the $m = \underline{m} - 1$ term in the integrand of $I_\sigma(t)$, so these integrals may be combined to run over $[\tau_{\underline{m}-1}, t]$ with $t > \tau_{\underline{m}}$. This demonstrates that the solution inherits the previous τ-step's functional form in the next τ-step, $[\tau_{\underline{m}}, \tau_{\underline{m}+1}]$, which includes delay structures from $\Delta_0(t)$ up to $\Delta_{m-1}(t)$.

Using Lemma 5.2 on the integration of $\mathbf{f}^*{}_{k,\underline{m}-1}$, which has arguments of $\Delta_{\underline{m}}(t)$ and $\Delta_{\underline{m}-1}(\Delta(t)) = \Delta_{\underline{m}}(t)$ respectively, which implies (5.6) holds by induction. When $m = k$, this also introduces a new delay structure given by $\Delta_k(t)$. In particular, this new delay structure's minimum power is the minimum power on $\Delta_{k-1}(t)$ in $\mathbf{p}_{k,k-1}$ plus one. Then (5.7) holds for the new delay terms $m = k$, so that (5.7) holds for all k by induction. □

If the initial data is not constant, then it is possible that the constant term associated with the vector field terms may change across τ-steps. Hence, because there is no initial invariance across τ-steps in the constant term, there can be no invariance with respect to τ in general for any coefficients. This does not preclude using dPSM, it just makes it 'trickier'.

6. Iteration structure - formal

6.1. **Basic array setup.** For DDEs, the PSM invariance result with respect to iterations is true if the vector field is linear, however, it requires a modification for the nonlinear case, because of the interaction between different delay structures.

In particular, lower order terms will have their coefficients changed during each iteration. But upon considering these coefficients as functions of τ, it will be that the coefficients on τ^s for fixed $s \leq k$ are invariant as k increases. In addition, tracking powers of $\Delta_m(t)$, one can ascertain whether all terms of a given power are computed or not by the integration.

To aid in the proof for the deviating argument case, it is convenient to note that a polynomial vector field associated with (3.2) can be reduced to a quadratic vector field.

Lemma 6.1. *If* (3.1) *is equivalent to* (3.2) *with* $\mathbf{f}$ *polynomial and solution* $\mathbf{u}$*, then there exists an* $\underline{\mathbf{f}}$ *such that degree* $\underline{\mathbf{f}}$ *is two and a component of which solves* (3.1)*.*

Proof. See [3], noting that deviating arguments may be relabeled in a similar fashion as non-deviating arguments and a change of variable for the deviating arguments may not contribute to the size of the vector field since an evolution line may not be required. □

This result is not necessary but invoking its result streamlines the upcoming proof of the structure between iterations by only having to consider sums of single multiplications in the vector field. Each multiplication can be represented by an array with monomials from the current approximations representing a row or column. The entries of the array would then represent the integration of the row and column monomial multiplication.

Example 6.2. Let $y' = (y(t-\tau))^r$ for constant τ with $t_0 = 0$. The vector field is not polynomial unless $r \in \mathbb{Z}^+$, so assume otherwise. To recast the vector field for dPSM, let $u = (y(t-\tau))^r$ which implies that $u' = r(y(t-\tau))^{r-1}y'(t-\tau) = ru(y(t-\tau))^{-1}y'(t-\tau)$. Note that the derivative term can be computed during Picard's iteration from the current approximation for y (after delaying it and taking a derivative), hence it does not require its own evolution line. This can be rewritten in terms of letters already in use by noting that $(y(t-\tau))' = y'(t-\tau) = (y(t-2\tau))^r = u(t-\tau) \equiv u_*$ upon using the DDE. The u' line can now be updated to $u' = ruu_*y_*^{-1}$, where the asterisk will denote in general delaying the current argument by τ.

The y_*^{-1} term needs to be in polynomial form still, so to this end, let $v = y_*^{-1}$, so that $v' = -1y_*^{-2}y_*' = -v^2y_*' = -v^2u_*$ and the u' line becomes polynomial as well. With these change of variables in place, the original DDE can be written as a two component system

$$\begin{aligned} u' &= ruvu_* \\ v' &= -v^2u_* \end{aligned} \tag{6.1}$$

and then integrating u will recover the original variable y. However, this system is cubic. A quadratic one can be made to appear, at the price of moving to a three component system, with the addition of $w = vu_*$ which implies $w' = -w^2 + vu_*'$. this updates (6.1) to the following system

$$\begin{aligned} u' &= ruw \\ v' &= -vw \\ w' &= -w^2 + vu_*' \end{aligned} \tag{6.2}$$

which is quadratic. Note that if one were to compute by hand, (6.1) is preferable to (6.2), but for the proofs, (6.2) is preferred.

Using Theorem 5.3, let $\mathbf{u}_{k,\underline{m}}(t) = \sum_{m=0}^{m} \mathbf{p}_{km}(t)$. It is convenient to arrange each component of $\mathbf{u}_k$ as an ordered list rather than an array across i and m. To this end, given u_k^l, suppress the component dependence, and consider $\mathbf{U}_k^l = [P_{k0}(t), \dots, P_{k,k-1}(t)]$ where $P_{k,\underline{m}}(t) = [\Delta_{\underline{m}}(t), \dots, \Delta_{\underline{m}}^d(t)]$, where coefficient information will be suppressed, to focus on tracking powers of $\Delta_m(t)$.

Because there will be two components to track in the multiplications of a quadratic vector field, the indices m and $\underline{m}$ will now be used as independent indices tracking the delay structures. Let $u_k^{l_1}$ generate $\mathbf{U}_k^{l_1} = (P_{km})_i$ and $u_k^{l_2}$ generate $\mathbf{U}_k^{l_2} = (Q_{k,\underline{m}})_j$, an array may be used to represent the quadratic terms, which can be thought of in block form $\mathbf{U}_k^{l_1}\mathbf{U}_k^{l_2} = (P_{km})_i(Q_{k,\underline{m}})_j$. In each entry of the array, tabulate the integration of the corresponding terms $(P_{km})_i(Q_{k,\underline{m}})_j$ by listing the result in the form $\Delta_m^i(t)$. The entries of these arrays would then be scalar multiplied and summed over components to achieve the integration of the entire vector field.

The result of Theorem 5.3 allows the block array when $m = k$ to be used for integration, and this result will contain the information for the cases when $m < k$ by simply ignoring the $\Delta_{\underline{m}}(t)$ terms for which $\underline{m} > m$. Assume the constant lag case for the delay, and as an example, let component $u_k^{l_1} = a_1 + b_1 t + c_1 t^2 + d_1(t-\tau)^2$ and $u_k^{l_2} = a_2 + b_2 t + c_2 t^2 + d_2(t-\tau)^2$. The array for a nonlinear interaction term in a vector field, upon suppressing coefficients, is given in Figure 2, which would be used to compute the next approximation $\mathbf{u}_{k+1}$. Ordering by exponent in each block shows that equal powers of t appear on the off diagonals in each of the main diagonal $P_{km}Q_{k,\underline{m}}$ blocks.

	1	t	t^2	$(t-\tau)^2$
1	t	$\frac{1}{2}t^2$	$\frac{1}{3}t^3$	$\frac{1}{3}(t-\tau)^3$
t	$\frac{1}{2}t^2$	$\frac{1}{3}t^3$	$\underline{t^4}$	wing
t^2	$\frac{1}{3}t^3$	$\underline{t^4}$	$\underline{t^5}$	wing
$(t-\tau)^2$	$\frac{1}{3}(t-\tau)^3$	wing	wing	$\underline{(t-\tau)^5}$

$$\begin{bmatrix} P_{k0}Q_{k0}\ (3\times 3) & P_{k0}Q_{k1}\ (3\times 1) \\ P_{k1}Q_{k0}\ (1\times 3) & P_{k1}Q_{k1}\ (1\times 1) \end{bmatrix}$$

FIGURE 2. Integration of vector field array - DDE

The underlined terms in the array are of higher power than the next power of t, which happens to be t^3 in the first main diagonal block, and will not be computed. This is due to $\Psi(t)$ being known so that the $P_{k0}Q_{k0}$ block represents a PSM problem, hence the PSM result applies, i.e. integration which produces t^k terms will be retained for $\mathbf{u}_k$'s integration to obtain $\mathbf{u}_{k+1}$.

These arrays expand in two ways each iteration. In PSM, the arrays expand by adding a row and column, which represents the increase in degree of the approximation by one. In dPSM, each block in the array adds a row and column for the

same reason, while overall the array also adds a new block row and block column, which corresponds to the newest delay term $\Delta_k(t)$.

6.2. **Wing terms.** Associated with each main diagonal block $P_{km}Q_{km}$ with $m \neq 0$ fixed, are off diagonal blocks, $P_{km}Q_{k,\underline{m}}$ and $P_{k,\underline{m}}Q^{km}$ with $0 \leq \underline{m} < m$. Ignoring the constant term multiplications, the remaining terms, referred to as *wing* terms, are a complication since Lemma 5.2 implies that an integration will produce terms with the same delay as the main diagonal block m, hence their contribution has to factor into when terms should be computed.

In particular, because of the range of powers produced by Lemma 5.2, wing terms allow the coefficients on lower powers of $\Delta_m(t)$ to change after each iteration, which is contrary to the PSM result. However, this coefficient change is predictable and it occurs through the development of a lag dependence in the coefficients, which will be polynomial with argument $\gamma_{m,\underline{m}}\tau = \tau\sum_{r=\underline{m}}^{m}\sigma^{-r}$, for the delay structure given by (3.8) with τ_{-1} relabeled as τ. Looking at just the coefficients, then a version of the PSM result holds, where the only change in coefficients in the next iteration is the inclusion of the next power of τ.

To see this, suppose the coefficients are τ dependent. The result of Lemma 5.2 on the wing terms shows specifically that they are a linear combination of powers of $\Delta_m(t)$ with powers from $\beta+1$ to $\alpha+\beta+1$ and coefficients with powers of τ from τ^β to τ^0, respectively. As an example, consider an arbitrary $P_{k1}Q_{k0}$ block, and assume $(a_1(t-\tau)+b_1(t-\tau)^2+c_1(t-\tau)^3)$ times $(a_2t+b_2t^2+c_2t^3)$. Suppressing coefficients and noting $\gamma_{m,\underline{m}} = (m-\underline{m})\tau$ for the constant lag case, the following array is constructed in Figure 3, using the notation $(\alpha,\beta) \equiv \tau^\alpha(t-\tau)^\beta$ and the underline for terms not to be computed in the next integration.

	$(t-\tau)$	$(t-\tau)^2$	$(t-\tau)^3$
t^0	$(0,2)$	$(0,3)$	$(0,4)$
t	$(1,2)+(0,3)$	$(1,3)+(0,4)$	$(1,4)+\underline{(0,5)}$
t^2	$(2,2)+(1,3)+(0,4)$	$(2,3)+(1,4)+\underline{(0,5)}$	$\underline{(2,4)}+\underline{(1,5)}+\underline{(0,6)}$
t^3	$(3,2)+(2,3)$ $+(1,4)+\underline{(0,5)}$	$\underline{(3,3)}+\underline{(2,4)}$ $+\underline{(1,5)}+\underline{(0,6)}$	$\underline{(3,4)}+\underline{(2,5)}$ $+\underline{(1,6)}+\underline{(0,7)}$

FIGURE 3. Integration of vector field array - wing terms and τ dependence

In this array, note that terms with the same power on both $t-\tau$ as well as τ run along the off diagonals, but not over the entire off diagonal. In addition, for a fixed m and based on Lemma 5.2, array entries are $\underline{m}$ invariant, so the only change in wing terms in different $\underline{m}$ blocks are the starting and ending indices for the rows and columns and the actual value of the coefficient on the power of τ under consideration. Neither of these changes across $\underline{m}$ will play a role in the invariance theorem.

6.3. **More Examples.** Before the theorem for invariance with respect to iterations is established, one more example is presented, which shows what can occur under specific conditions on the vector field.

Example 6.3. Consider the DDE $y'(t) = y^2(t) + (y(t-\tau))^2$ with initial data 1 over $[-\tau, 0]$. Using (4.3) as in Example 3 when $z_{52}(t)$ was computed, one can show that

$$y_2(t) = \begin{cases} y_{20}(t) = 1 + 2t + 2t^2 & t \in [0, \tau] \\ y_{21}(t) = 1 + 2t + 2t^2 + 2(t-\tau)^2 & t \in [\tau, T] \end{cases}$$

and using $y_{21}(t)$ and the results of all the theorems, one can quickly compute $y_3(t)$ via substituting $y_{21}(t)$ into the DDE and integrating each term involving τ_m from τ_m up to t. Using a double underline to indicate the newly computed terms, i.e. the non-underlined terms are $y_2(t)$, we have

$$\begin{aligned} y_{32}(t) = 1 + 2t + 2t^2 + \underline{\underline{\frac{8}{3}t^3}} + 2(t-\tau)^2 + \underline{\underline{\big(4 + \frac{8}{3}\tau + \frac{8}{3}\tau^2\big)(t-\tau)^3}} \\ + \underline{\underline{\big(\frac{4}{3} + \frac{8}{3}\tau + \frac{8}{3}\tau^2\big)(t-2\tau)^3}} \end{aligned} \tag{6.3}$$

where the coefficients that have developed a τ dependence have retained terms up to τ^2.

The theorem to follow uses the relation $s_k^* = k + m - i$ to determine the largest power of τ to retain on the term of the form $\Delta_m^{i+1}(t)$. Using $k = 2$, $m = 2$, which is the minimum power of $\Delta_m(t)$ that produces the wing term, which happens to be m and considering the $(t-\tau)^3$ term, so that $i+1 = 3$, then $s_k^* = 2+2-2 = 2$, which coincides with the τ^2 that was retained. In this example, the highest power of τ generated happened to be 2, so there weren't higher powers to retain. However, if there had been, this would imply more terms of that power still need to be computed.

Further, if the coefficient on $(t-\tau)^3$ in $y_{42}(t)$, i.e. same τ-step, next iterate, were computed it would be $\big(4 + \frac{8}{3}\tau + \frac{8}{3}\tau^2 + \tau^3\big)$ because only powers up to and including $s_k^* = k + m - i = 3 + 2 - 2 = 3$ should be computed, and the coefficients on τ^2 and τ are invariant.

This will now be contrasted against the following example. Let $y'(t) = (y(t-\tau))^2$ with the same initial data. Again using (4.3) as in Example 3, one can compute

$$y_2(t) = \begin{cases} y_{20}(t) = 1 + t & t \in [0, \tau] \\ y_{21}(t) = 1 + t + (t-\tau)^2 + \underline{\underline{\frac{1}{3}(t-\tau)^3}} & t \in [\tau, T], \end{cases}$$

where the cubic term has been underlined because normally more cubic terms would still need to be computed, so this term shouldn't be computed in this iteration. But for this vector field, those cubic terms will have zero coefficients, and so this underlined term represents all cubic terms, and will be retained during this iteration. In addition,

$$\begin{aligned} y_{32}(t) = 1 + t + (t-\tau)^2 + \frac{1}{3}(t-\tau)^3 \\ + \big(\frac{2}{3} + \frac{2}{3}\tau\big)(t-2\tau)^3 + \underline{\underline{\big(\frac{2}{3} + \frac{1}{6}\tau\big)(t-2\tau)^4}} \\ + \underline{\underline{\frac{1}{3}(t-2\tau)^5}} + \underline{\underline{\frac{1}{9}(t-2\tau)^6}} + \underline{\underline{\frac{1}{63}(t-2\tau)^7}} \end{aligned} \tag{6.4}$$

where the underlines are used for the same reason as before. Note that looking at the $(t-2\tau)^3$ term, $k = 2$, $m = 2$ and $i+1 = 3$, so $k+m-i = 2$ which does not match

the term τ^1. Here, constant initial data and the lack of t arguments in the vector field produce only $t-\tau$ terms, and none of higher power. Hence the assumptions that went into forming $k+m-i$ above need to be revisited, in particular, k should be replaced by $k-1$ here.

7. dPSM results - II

The next theorem is essentially the invariance of coefficients idea from Theorem 4.1 but with respect to k rather than with respect to m with consideration of a τ dependence. The theorem holds if Ψ is not constant, but in light of Theorem 5.3, only in a fixed τ-step.

The fastest propagation occurs when the vector fields have an additional condition placed on them. A *mixing* vector field is fully delayed and has in each component, either a term of the form $u(t)v(\Delta(t))$, otherwise terms of the form $u(t)v(t)+w(\Delta(t))x(\Delta(t))$, where u,v,w and x are any components, some or all of which may be the same. Note that the first vector field in Example 5 falls into this category, while the second one does not.

Theorem 7.1. *Using $\Delta(t)$ as in* (3.8) *and mixing $\mathbf{f}$, then the solution to* (3.2) *has the form* (5.6) *and* (5.7)*, where the coefficients satisfy*

$$\mathbf{a}^i_{km}=\sum_{s=0}^{s_k}\underline{\mathbf{a}}^{i,s}_{km}\tau^s$$

for some $s_k>0$. In addition, for fixed k and fixed $i\le k$ and every $-1\le m\le \underline{m}_k$, defined in Theorem 5.3, we have

$$\underline{\mathbf{a}}^{i,s}_{k+n,m}=\underline{\mathbf{a}}^{i,s}_{km} \tag{7.1}$$

for each $s\le s^*\equiv k+m-i$ and every $n>0$.

Proof. Consider fusing the ordered lists $[\mathbf{u}_k(t),\mathbf{u}_k(\Delta(t))]$ so that the delay terms in the vector field can be distinguished via the component index: $u^l_k(t)$ represents the $(l-L-1)$th derivative of $\mathbf{u}_k$ evaluated at $\Delta(t)$ if $L<l\le 2L$. Given a solution to (3.2), consider the quadratic vector field for the $(k+1)$th iteration and suppressing time dependence, we have

$$Du^l_{k+1}=\sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}c^l_{l_1l_2}u^{l_1}_k u^{l_2}_k$$

The indices m and $\underline{m}$ will now denote independent indices for the delay structure in each component. There is then the need to relabel $\underline{m}$, the index for the τ-step, which is now denoted by n. For $n<k$, (4.3) produces $\{u^l_{k,n}\}_{l_1=1}^{2L}$, polynomials of the form (5.6) and (5.7).

$$\begin{aligned}u^l_{k+1,n}=u^l_{kn}(\tau_0)&+\sum_{m=0}^{n-1}\int_{\tau_m}^{\tau_{m+1}}\sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}c^l_{l_1l_2}u^{l_1}_{km}u^{l_2}_{km}\,\mathrm{d}s\\&+\int_{\tau_n}^{t}c^l_{l_1l_2}u^{l_1}_{kn}u^{l_2}_{kn}\,\mathrm{d}s\end{aligned} \tag{7.2}$$

Decompose further with

$$u^l_{kn}=\sum_{m=-1}^{n_k}p^l_{km}=\sum_{m=-1}^{n_k}\Big(\sum_{i=i^*}^{k-1}a^{il}_{km}\Delta^i_m+\sum_{i=k}^{d}a^{il}_{km}\Delta^i_m\Big) \tag{7.3}$$

where d is the largest power produced by integrating the vector field and $i^*=n+1$ if $l\le L$ and $i^*=n$ if $l>L$. In particular, if $n=-1$, then $d=0$ and both sums collapse to constant terms. Integrands in (7.2) are a linear combination of function multiplications. Substituting (7.3) into (7.2) shows these multiplications each have the following form:

$$\begin{aligned}
&\sum_{m=-1}^{n_k}\sum_{\underline{m}=-1}^{\underline{n}_k}\Big(\sum_{i=i^*}^{k-1}a_{km}^{il_1}\Delta_m^i\sum_{\underline{i}=i^*}^{k-1}a_{k,\underline{m}}^{\underline{i}l_2}\Delta_{\underline{m}}^{\underline{i}}\Big)\\
&+\sum_{m=-1}^{n_k}\sum_{\underline{m}=-1}^{\underline{n}_k}\Big(\sum_{i=k}^{d}a_{km}^{il_1}\Delta_m^i\sum_{\underline{i}=k}^{d}a_{k,\underline{m}}^{\underline{i}l_2}\Delta_{\underline{m}}^{\underline{i}}\Big)\\
&+\sum_{m=-1}^{n_k}\sum_{\underline{m}=-1}^{\underline{n}_k}\Big(\sum_{i=k}^{d}a_{km}^{il_1}\Delta_m^i\sum_{\underline{i}=i^*}^{k-1}a_{k,\underline{m}}^{\underline{i}l_2}\Delta_{\underline{m}}^{\underline{i}}\Big)\\
&+\sum_{m=-1}^{n_k}\sum_{\underline{m}=-1}^{\underline{n}_k}\Big(\sum_{i=i^*}^{k-1}a_{km}^{il_1}\Delta_m^i\sum_{\underline{i}=k}^{d}a_{k,\underline{m}}^{\underline{i}l_2}\Delta_{\underline{m}}^{\underline{i}}\Big)\\
&\equiv\sum_1+\sum_2+\sum_3+\sum_4
\end{aligned}\tag{7.4}$$

where for simplicity a_{km}^{il} is taken to be zero if $m=0$ and $l>L$.

Intuitively, each individual l_1 and l_2 will produce a block array of the form $(P_{km}^{l_1})_i(P_{k,\underline{m}}^{l_2})_{\underline{i}}$. In this sense, m and $\underline{m}$ are indices for the block form of the array, and i and $\underline{i}$ are local indices inside a particular block.

The multiplication of the sums in the terms of $\sum_1$ demonstrate that the previous iteration's powers of $\Delta_m(t)$ contribute to the next iteration. The multiplication in the terms of $\sum_2$ produces terms with powers of $\Delta_m(t)$ at least $2k+1$ after integrating and should not be computed in the next integration since terms of power $2k,2k-1,\dots,k+1$ are not yet present in the approximations to contribute. This also applies to powers above k in $\sum_1$.

The multiplications from $\sum_3$ and $\sum_4$ are wing terms, and depending on whether $m>\underline{m}$ or $m<\underline{m}$, one set will contribute $\Delta_m^i(t)$ with $i\ge k$ and will not be computed, while the other set due to Lemma 5.2, will contribute $\Delta_m^i(t)$ with $i=m+1,\dots,k$. This implies that coefficients for lower powers in general are not invariant with respect to iterations, however, they can be shown invariant with respect to iterations when considered as polynomials of τ.

To see this, denote the coefficient on Δ_m^{i+1}, with $i\le k$, in the l component of $\mathbf{u}_k$ by $a_{km}^{i+1,l}$. This coefficient may be a polynomial in τ, and is determined by summing up the coefficients on all Δ_m^{i+1} terms in the arrays for each (l_1,l_2). Since i is considered fixed, introduce I, J as independent indices for powers of Δ_m. In each array, these sums have the form

$$\begin{aligned}
&a_{k+1,m}^{i+1,l}\\
&=\sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}\Big[(i+1)^{-1}\sum_{I+J}\sum_{=i}a_{km}^{Il_1}a_{km}^{Jl_2}\\
&\quad+\sum_{\underline{m}=-1}^{m-1}\Big(\sum_{J=m}^{i}\sum_{I=i-J}^{k}a_{k,\underline{m}}^{Il_1}a_{km}^{Jl_2}\gamma\tau^{IJ}+\sum_{I=m}^{i}\sum_{J=i-I}^{k}a_{km}^{Il_1}a_{k,\underline{m}}^{Jl_2}\gamma\tau^{IJ}\Big)\Big]
\end{aligned}\tag{7.5}$$

where $\gamma\tau^{IJ} \equiv \gamma_{m\underline{m}}\tau^{I+J-i}$. Note that $\gamma_{m,\underline{m}}$ is constant with respect to τ and independent of k. The terms in the first line come out of the appropriate main diagonal block and the second line has all the wing terms. Note also that the indices have been relabeled so that m is the larger index, while $\underline{m}$ is the smaller one

Consider now a future integration for the same term $\Delta_m^{i+1}(t)$ using $\mathbf{u}_{k+n}$,

$$\begin{aligned}
a_{k+n+1,m}^{i+1,l} = \sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}\Big[&(i+1)^{-1}\sum_{I+J}\sum_{=i} a_{k+n,m}^{Il_1}a_{k+n,m}^{Jl_2} \\
&+ \sum_{\underline{m}=-1}^{m}\Big(\sum_{J=m}^{i}\sum_{I=i-J}^{k+n} a_{k+n,\underline{m}}^{Il_1}a_{k+n,m}^{Jl_2}\gamma\tau^{IJ} \\
&+ \sum_{I=m}^{i}\sum_{J=i-I}^{k+n} a_{k+n,m}^{Il_1}a_{k+n,\underline{m}}^{Jl_2}\gamma\tau^{IJ}\Big)\Big]
\end{aligned}$$

and upon breaking off the powers above k, we have

$$\begin{aligned}
&a_{k+n+1,m}^{i+1,l} \\
&= \sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}\Big[(i+1)^{-1}\sum_{I+J}\sum_{=i} a_{k+n,m}^{Il_1}a_{k+n,m}^{Jl_2} \\
&\quad + \sum_{\underline{m}=-1}^{m}\Big(\sum_{J=m}^{i}\sum_{I=i-J}^{k} a_{k+n,\underline{m}}^{Il_2}a_{k+n,m}^{Jl_2}\gamma\tau^{IJ} + \sum_{I=m}^{i}\sum_{J=i-I}^{k} a_{k+n,m}^{Il_1}a_{k+n,\underline{m}}^{Jl_2}\gamma\tau^{IJ} \\
&\quad + \sum_{J=m}^{i}\sum_{I=k+1}^{k+n} a_{k+n,\underline{m}}^{Il_1}a_{k+n,m}^{Jl_2}\gamma\tau^{IJ} + \sum_{I=m}^{i}\sum_{J=k+1}^{k+n} a_{k+n,m}^{Il_1}a_{k+n,\underline{m}}^{Jl_2}\gamma\tau^{IJ}\Big)\Big]
\end{aligned} \tag{7.6}$$

For notational convenience, the double sum over l_1 and l_2 will be suppressed along with l_1 and l_2 dependence, along with denoting $a_{km}^I \equiv a_{km}^{Il_1}$ and $b_{km}^J \equiv a_{km}^{Jl_2}$. Now let $k=0$, which implies that m, i and $s=0$ in the theorem statement. Theorem (5.3) shows that the theorem statement holds for $k=0$.

If the theorem statement holds for arbitrary k, i.e. $i \le k$ and $s \le s^*$, then

$$a_{k+n,m}^I = a_{km}^I + \sum_{s=s^*+1}^{d} \underline{a}_{k+n,m}^{Is}\tau^s \equiv a_{km}^I + p_{s^*+1}^a \tag{7.7}$$

with a similar statement for $b_{k+n,m}^J$. Inserting (7.7) into (7.6) and expanding, we have

$$\begin{aligned}
&(i+1)^{-1}\sum_{I+J}\sum_{=i}\Big(a_{km}^I b_{km}^J + p_{s^*+1}^a b_{km}^J + p_{s^*+1}^b a_{km}^I + p_{s^*+1}^a p_{s^*+1}^b \\
&+ \sum_{\underline{m}=-1}^{m-1}\Big(\sum_{J=m}^{i}\sum_{I=i-J}^{k}\Big(a_{k,\underline{m}}^I b_{km}^J + p_{s^*+1}^a b_{km}^J + p_{s^*+1}^b a_{k,\underline{m}}^I + p_{s^*+1}^a p_{s^*+1}^b\Big)\gamma\tau^{IJ} \\
&+ \sum_{I=m}^{i}\sum_{J=i-I}^{k}\Big(a_{km}^I b_{k,\underline{m}}^J + p_{s^*+1}^a b_{k,\underline{m}}^J + p_{s^*+1}^b a_{km}^I + p_{s^*+1}^a p_{s^*+1}^b\Big)\gamma\tau^{IJ}
\end{aligned}$$

$$+\sum_{J=m}^{i}\sum_{I=k+1}^{k+n}\Big(a^I_{k,\underline{m}}b^J_{km}+p^a_{s^*+1}b^J_{km}+p^b_{s^*+1}a^I_{k,\underline{m}}+p^a_{s^*+1}p^b_{s^*+1}\Big)\gamma\tau^{IJ}$$
$$+\sum_{I=m}^{i}\sum_{J=k+1}^{k+n}\Big(a^I_{km}b^J_{k,\underline{m}}+p^a_{s^*+1}b^J_{k,\underline{m}}+p^b_{s^*+1}a^I_{km}+p^a_{s^*+1}p^b_{s^*+1}\Big)\gamma\tau^{IJ}\Big) \tag{7.8}$$

and note that the first multiplication in the first, second and third lines of (7.8) are the same terms from (7.5). In the second and third terms in the first line of (7.7), note that a and b may have constant terms. It follows that these terms will have a minimum power of τ given by the minimum power in p_{s^*+1}, which is s^*+1 and hence, they may be written in the form $P(\tau)\tau^{s^*+1}$ where P is a polynomial with a constant term. All the second and third terms in lines two through five have a larger power of τ than s^*+1 and are thus considered higher order. This is also true for all the terms of the form $p^a_{s^*+1}p^b_{s^*+1}$.

So, summing over l_1 and l_2 and explicitly denoting the lowest power of τ in the first three lines of (7.8) along with the lowest power of τ in the last two lines, we have

$$\begin{aligned}a^{i+1}_{k+n+1,m}&=a^{i+1}_{k+1,m}+P\tau^{k+m-i}+G(\tau)\\&\quad+\sum_{l_1=1}^{2L}\sum_{l_2=1}^{2L}c^l_{l_1l_2}\sum_{\underline{m}=m^*}^{m-1}\Big(\sum_{J=m}^{i}\sum_{I=k+1}^{k+n}a^{Il_1}_{k,\underline{m}}a^{Jl_2}_{km}\gamma\tau^{IJ}\\&\quad+\sum_{I=m}^{i}\sum_{J=k+1}^{k+n}a^{Il_1}_{km}a^{Jl_2}_{k,\underline{m}}\gamma\tau^{IJ}\Big)\end{aligned} \tag{7.9}$$

where P is a polynomial with respect to τ with a constant term and $G(\tau)$ contains the higher order terms. In addition, both $a^{il_1}_{km}$ and $a^{il_2}_{km}$ may have constant terms, so that the minimum power of τ in the second line of (7.9) is $\min I+J-i=k+1+m-i=s^*+1$ over both sums. Hence, $\underline{a}^{i+1,s}_{k+n+1,m}=\underline{a}^{i+1,s}_{k+1,m}$ if $s\le k+m-i-1$ and so (5.6) and (5.7) both hold for $k+1$, and thus all k. □

Note that both powers of $\Delta_m(t)$ along with powers of τ from Lemma 5.2 are constant along off diagonals in each block array. Hence the main diagonal blocks dictate keeping powers of $\Delta_m(t)$ up to $i\le k+1$, and then keeping powers of τ up to $k+m-i$.

PSM also provides an explicit a priori error bound which does not involve derivatives of the vector field, [13]. Note that over $[\tau_{m-1},\tau_m]$, the structure in Theorem 5.3 can be expanded in a series around the single center τ_{m-1}. Using this representation, the error estimate for PSM may be extended to the deviating argument case.

Some notation will now be introduced and to facilitate, there is a strong overlap with the notation in [13]. Denote by $\|\mathbf{v}\|_\infty=\max_l v^l$, the maximum over the components of $\mathbf{v}$. The vector field will be polynomial and written compactly as

$$\mathbf{f}(\mathbf{u},\mathbf{u}^*)=\sum_{\mathbf{i}=0}^{d_\mathbf{f}}\sum_{\mathbf{j}=0}^{d_\mathbf{f}}\mathbf{B_{ij}}\mathbf{u}^\mathbf{i}\mathbf{u}^\mathbf{j}_*$$

where the sum over the ordered list of indices implies a sum over each index in the list ranging from 0 to $d_\mathbf{f}$, the exponentiation between ordered lists is componentwise

and the asterisk denotes the delay terms. Denote by $\|\mathbf{B_{ij}}\|$ the maximum row sum of coefficient magnitudes, the subordinate matrix norm to the vector norm $\|\cdot\|_\infty$.

If one considers (3.2) globally, then $\|\mathbf{B_{ij}}\|$ would be the norm of the vector field. There is also occasion to consider the local vector field for the ODE in $[\tau_m, \tau_{m+1}]$ with the delay info considered 'known' and hence, the norm of the vector field would need to incorporate the delay information into the coefficient matrix. Dub the local ODE as $\mathbf{f}_*(\mathbf{u})$.

Theorem 7.2. *For finite $m \geq 0$, the expansion $\mathbf{u}_{km}(t) = \sum_{\underline{m}=-1}^{m} \mathbf{p}_{k,\underline{m}}(t)$ generated by (4.3) is such that*

$$\|\mathbf{u}(t) - \mathbf{u}_{km}(t)\|_\infty \leq C_{km}\Big(F(t-\tau_m; \|\mathbf{B_{ij}}\|_\infty, d_\mathbf{f}) - \sum_{q=0}^{k} z_q (t-\tau_m)^q\Big) \tag{7.10}$$

when $t \in [\tau_m, T]$ with $d_\mathbf{f}+1$ being the vector field's degree in (3.2) and constant C_{km} dependent on $\|\mathbf{\Psi}\|_\infty$, the norm of the initial data along with

$$F(t;a,b) = \exp(at), \quad b = 0 \quad \text{and} \quad F(t;a,b) = \frac{1}{(1-abt)^{1/b}}, \quad b \geq 1$$

The sequence elements z_q are the expansion coefficients of F which solves $z'(t) = az^b$.

This result holds for non-constant initial data.

Proof. Approximating (3.2) via (4.3), note that in each τ-step, the DDE reduces to an ODE with non-constant coefficients and/or non-homogeneous terms involving the solution from the previous τ-step.

To recover an ODE, the vector field must be modified to include the delay information in its coefficients. Summing $\mathbf{B}_{ij}\mathbf{u}_*^j$ over j would yield coefficients for the ODE $\mathbf{f}_*(\mathbf{u})$ over the particular τ-step. Noting that (3.3) holds in the limit, we have

$$\mathbf{u}_*(t) \leq \max_{[0,t]} \mathbf{u}_* \leq \max_{[0,t]} \mathbf{u} \leq M_\Psi + tM_f \tag{7.11}$$

using the notation of Proposition 3.1, and the coefficient matrix satisfies the bound

$$\|\mathbf{f}_*\| \leq \|\mathbf{B}_{ij}\|\big((\|M_\Psi + \tau_m M_f)^*\big)^{d_\mathbf{f}+1} \equiv \|\mathbf{f}_m\|$$

where the asterisk indicates $\max\{\cdot, 1\}$. Hence, (3.2) becomes the ODE initial value problem $\mathbf{u}' = \mathbf{f}_*(\mathbf{u})$ with initial data $\mathbf{u}(\tau_m) = \mathbf{u}_{km}(0)$, to which the result from [13] can be applied. For convenience, $\|\cdot\|_\infty$ is shortened to $\|\cdot\|$, $F(\cdot; \|\mathbf{f}_m\|, d_\mathbf{f}+1)$ is shortened to $F_m(\cdot)$ and $\|\mathbf{e}_{km}(t)\| \equiv \|\mathbf{u}(t) - \mathbf{u}_{km}(t)\|$. This yields

$$\|\mathbf{e}_{km}(t)\| \leq \|\mathbf{u}^*_{k,m-1}(\tau_m)\|\Big(F_m(t-\tau_m) - \sum_{q=0}^{k} z_{qm}(t-\tau_m)^q\Big) \tag{7.12}$$

where the asterisk again indicates $\max\{\cdot, 1\}$ and z_{qm} are the expansion coefficients of F_m which solves $z'(t) = \|\mathbf{f}_m\| z^{d_\mathbf{f}+1}$. Using the fact that $\|\mathbf{e}_{km}(t)\| \geq 0$ along with the substitution $\|\mathbf{u}^*_{k,m-1}(\tau_m)\| = \|\mathbf{u}(\tau_m) - \mathbf{e}_{k,m-1}(\tau_m)\|$, we have

$$\begin{aligned}\|\mathbf{u}^*_{k,m-1}(\tau_m)\| &\leq \max\{\|\mathbf{u}(\tau_m)\| + \|\mathbf{e}_{k,m-1}(\tau_m)\|, 1 + \|\mathbf{e}_{k,m-1}(\tau_m)\|\} \\ &\leq \max\{\|\mathbf{u}(\tau_m)\|, 1\} + \|\mathbf{e}_{k,-1}(\tau_m)\|\end{aligned}$$

Using (3.3) in the limit as before to bound the solution, then (7.12) becomes

$$\begin{aligned}&\|\mathbf{e}_{km}(t)\|\\ &\le \left((M_\Psi+\tau_m M_f)^*+\|\mathbf{e}_{k,m-1}(\tau_m)\|\right)\Big(F_m(t-\tau_m)-\sum_{q=0}^{k} z_{qm}(t-\tau_m)^q\Big)\end{aligned} \tag{7.13}$$

and (7.13) may then be continued to

$$\|\mathbf{e}_{km}(t)\| \le C_{km}\Big(F_m(t-\tau_m)-\sum_{q=0}^{k} z_{qm}(t-\tau_n)^q\Big)$$

which is (7.10), where

$$C_{km} \equiv (M_\Psi+\tau_m M_f)^* \sum_{\underline{m}=0}^{m}\prod_{n=\underline{m}}^{m}\Big(F_n(\tau_{n+1}-\tau_n)-\sum_{q=0}^{k} z_{qn}(\tau_{n+1}-\tau_n)^q\Big)$$

Note that a slightly less tight, but computationally nicer bound exists for the basic PSM result which could be adapted here, see [13]. □

8. General delay structure

There is a logical difficulty extending Proposition 3.1 to the state dependent delay case due to the changing nature of Δ with respect to iterations. Hence, the discussion of the approximation of this case via dPSM is heuristic. With regards to (5.6), note that Δ can only known approximately since $\mathbf{u}$ would only be known approximately. If one imagines a sequence of approximations Δ_k to Δ, these could differ in functional form each iteration, or possibly only in certain parameters, for example σ and/or $\tau=\tau_{-1}$ in the case of (3.8).

If a problem starts in the solution's basin of attraction so that the basic iteration would evolve to the correct fixed point, then $\Delta_{km}(t)=\Delta(t,\mathbf{u}_{km}(t))$ may be computed based on the computation of $\mathbf{u}_{km}$. At the very least, one needs to store the polynomial for the τ coefficients so that upon updating the set of τ_m at each iteration based on the currently computed $\Delta_k(t)$ would allow updating the previously computed coefficients so that they do not have to be recomputed each iteration. In other words, there would be invariance in previously computed coefficients once $\tau_{k-1,m}$ is updated to τ_{km} based on $\Delta_{km}(t)$.

Acknowledgments. The author would like to thank Drs. James Sochacki and Allen Holder for useful discussions, constructive criticisms and their donations of time and expertise during the writing of this article.

References

[1] C. T. H. Baker, C. A. H. Paul, D. R. Willé; *Issues in the Numerical Solution of Evolutionary Delay Differential Equations*, Adv. Comp. Math, 3 (1995), pp. 171-196.

[2] A. Bellen, N. Guglielmi, A. E. Ruehli; *Methods for Linear Systems of Circuit Delay Differential Equations of Neutral Type*, IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, 46 (1999) 1, pp. 212-215.

[3] D. C. Carothers, G. E. Parker, J. S. Sochacki, P. G. Warne; *Some Properties of Solutions to Polynomial Systems of Equations*, Elec. Jour. of Diff. Eqn., 2005 (2005) 40, pp. 1-17.

[4] E. A. Coddington, N. Levinson; *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955

[5] L. E. El'sgol'ts; *Introduction to the Theory of Differential Equations with Deviating Arguments*, (translated by R. J. McLaughlin), Holden-Day, San Francisco, 1966

[6] E. Fehlberg; *Numerical Integration of Differential Equations by Power Series Expansions, Illustrated by Physical Examples*, Technical Report NASA-TN-D-2356, NASA, (1964).
[7] V. M. Isaia; *dPSM and Polynomial Initial Data*, in preparation.
[8] S. B. Norkin; *Differential Equations of the Second Order with Retarded Argument*, Translations of Mathematical Monographs (L. J. Grimm) vol. 31, AMS, Providence, 1972
[9] G. E. Parker, J. S. Sochacki; *Implementing the Picard Iteration*, Neural, Parallel and Scientific Computations, 4 (1996) 1, pp. 97-112.
[10] G. E. Parker, J. S. Sochacki; *A Picard-Maclaurin Theorem for Initial Value PDEs*, Abstr. Appl. Anal., 5 (2000) 1, pp. 47-63.
[11] J. S. Sochacki, A. Tongen; *Exploring Polynomial Dynamical Systems: An Interesting Application of Power Series to Differential Equations*, Springer-Verlag, 2016.
[12] R. D. Stewart, W. Bair; *Spiking Neural Network Simulation: Numerical Integration with the Parker-Sochacki Method*, Journal Computational Neuroscience, 27 (2009), pp. 115-133.
[13] P. G. Warne, D. A. P. Warne, J. S. Sochacki, G. E. Parker, D. C. Carothers; *Explicit A-Priori Error Bounds and Adaptive Error Control for Approximation of Nonlinear Initial Value Differential Systems*, Comput. Math. Appl., 52 (2006), pp. 1695-1710.

Vincenzo Michael Isaia
Department of Mathematics, Rose-Hulman Institute of Technology, Terre Haute, IN 47803, USA
E-mail address: isaia@rose-hulman.edu