BIT manuscript No.
(will be inserted by the editor)

Superconvergent interpolants for collocation methods applied to Volterra integro-differential equations with delay

Mohammad Shakourifar · **Wayne Enright**

Received: date / Accepted: date

Abstract Standard software based on the collocation method for differential equations, delivers a continuous approximation (called the collocation solution) which augments the high order discrete approximate solution that is provided at mesh points. This continuous approximation is less accurate than the discrete approximation. For 'non-standard' Volterra integro-differential equations with constant delay, that often arise in modeling predator-prey systems in Ecology, the collocation solution is C^0 continuous. The accuracy is $O(h^{s+1})$ at off-mesh points and $O(h^{2s})$ at mesh points where s is the number of Gauss points used per subinterval and h refers to the step-size. We will show how to construct C^1 interpolants with an accuracy at off-mesh points and mesh points of the same order ($2s$). This implies that even for coarse mesh selections we achieve an accurate and smooth approximate solution. Specific schemes are presented for $s = 2, 3$, and numerical results demonstrate the effectiveness of the new interpolants.

Keywords Delay Volterra integro-differential equations · Piecewise polynomial collocation · Bootstrapping · Order conditions

Mathematics Subject Classification (2000) 65R20 · 65L60 · 65L06

1 Introduction and Motivation

Progress in the qualitative and numerical understanding of Delay Volterra Integro-Differential Equations (DVIDEs) is enhancing the use of this type of equations in

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

M. Shakourifar
Department of Computer Science, University of Toronto
E-mail: shakouri@cs.toronto.edu

W.H. Enright
Department of Computer Science, University of Toronto
E-mail: enright@cs.toronto.edu

various fields of Biology [14], Medicine [6] and Engineering. Many DVIDEs arising as mathematical models in population dynamics, rheology, etc. have the general 'non-standard' form,

$$y'(t) = f(t,y(t)) + \int_{t-\tau}^{t} K(t,s,y(t),y(s))ds, \tag{1.1}$$

for $t \in [t_0,T]$, $f : R \times R^m \to R^m$ and $K : R \times R \times R^m \times R^m \to R^m$. To make the problem well-defined, a unique solution $y(t)$ is usually identified by specifying an initial function $\phi(t)$ for $t \in [t_0-\tau,t_0]$, with $\phi(t) : R \to R^m$. As a particular case in traditional population biology, the following system models the dynamics of two interacting species often called a 'predator-prey' system,

$$\begin{cases} N_1'(t) = N_1(t)\left(\varepsilon_1 - \gamma_1 N_2(t) - \int_{t-\tau}^{t} F_1(t-s)N_2(s)ds\right), \\ N_2'(t) = N_2(t)\left(-\varepsilon_2 + \gamma_2 N_1(t) + \int_{t-\tau}^{t} F_2(t-s)N_1(s)ds\right), \end{cases} \tag{1.2}$$

for $t \in [0,T]$, where $\varepsilon_i > 0$, $\gamma_i \geq 0$, $F_i(t) \geq 0$ is continuous, and $N_1(t) = \phi_1(t)$, $N_2(t) = \phi_2(t)$ for $t \in [t_0-\tau,t_0]$. $N_1(t)$ and $N_2(t)$ represent two populations (prey and predator) at time t (see for example [5,18] for more details). Multi-species ecological systems can be derived as extensions of this system [19,1]. For sufficient conditions regarding the existence of solutions and also the long term behavior of special types of such systems see [10,11].

Numerical investigations concerning the approximate solution of 'standard' Volterra integro-differential equations have been considered over the last few decades [4]. In particular, after the systematic introduction of Runge-Kutta (RK) methods for ODEs by J.C. Butcher, attempts were made to formally generalize these methods to Volterra integro-differential equations (VIDEs). As a result, two particular subclasses of Volterra RK (VRK) methods, namely, the VRK methods of *Pouzet type*, and VRK methods of *Bel'tyukov type* have received attention in the literature. The advantage of these methods is that they may have explicit forms (no implicit algebraic system to be solved) despite the presence of the Volterra integral term in the equation. *Pouzet* and *Bel'tyukov* type methods were extended to 'standard' VIDEs with constant delay for fixed stepsize implementation (see for example [21,22,20]). One can straightforwardly extend these methods to the 'non-standard' DVIDEs with constant delay. A major deficiency of these methods is that the discretization is a part of the method and no *continuous interpolant* is produced. There have been some attempts to make such methods more effective and convenient for users. In particular, some investigations [7] have focused on the development of smooth interpolants augmenting the discrete solution generated by the discrete numerical method. Although the computation of these interpolants imposes an inevitable cost, they allow one to consider the resulting method as providing a continuous approximate solution to the original equation throughout the domain of interest that might be used for visualization, root-finding and other purposes.

One approach for developing numerical methods that determine such smooth interpolants of the discrete solution is based on the use of continuous Runge-Kutta (CRK) methods. An $s-$stage, p^{th} order, explicit discrete Runge-Kutta formula when applied to the standard initial value problem (IVP),

$$y' = f(t,y), \quad y(a) = y_0, \quad \text{for } t \in [a,b],$$

determines,

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} w_i k_i,$$

where,

$$k_i = f(t_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{i,j} k_j), \quad i = 1, \ldots, s.$$

An additional $(\bar{s} - s)$ stages are introduced to obtain an optimal-order continuous (rather than 'discrete') approximation for any $t \in (t_n, t_{n+1})$ as,

$$u_n(t) = y_n + h_n \sum_{j=1}^{\bar{s}} b_j(\tau) k_j,$$

where $b_j(\tau)$ is a polynomial of degree at most $p+1$ and $\tau = \frac{t-t_n}{h_n}$. The parameters defining such a CRK method are usually chosen to ensure that the piecewise polynomial $u(t)$, defined by the mesh $t_0 < t_1 < \ldots < t_N$ and the associated polynomials $u_n(t)$ $n = 0, 1, \ldots, (N-1)$, is in $C^1[a,b]$ (to accomplish this the first additional stage k_{s+1} is defined by $k_{s+1} = f(t_{n+1}, y_{n+1})$). Different criteria, such as the quality of the defect and the corresponding defect estimate associated with the continuous interpolant, are used to determine the number of additional stages as well as the specific coefficients used to define them.

CRK methods have been applied to a more general class of DVIDEs in [17]. When applied to (1.1), the k_i's (required for the definition of the local interpolant $u_n(t)$) are defined by the following system of equations (see [17]),

$$k_i = f(t_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{i,j} k_j) + \int_{t_n + c_i h_n - \tau}^{t_n + c_i h_n} K(t_n + c_i h_n, s, y_n + h_n \sum_{j=1}^{i-1} a_{i,j} k_j, u(s)) ds,$$

for $i = 1, 2, \ldots, \bar{s}$, where $u(s)$ is the piecewise polynomial numerical solution defined by $[u_i(s)]_{i=0}^{N-1}$ for $s \in [t_0, t_N]$. For this CRK, we have a system of $\bar{s} \times m$ *coupled* equations (defining the k_i's), since $u_n(t)$ is defined in terms of all the k_i's introduced on this step. This CRK method allows for variable stepsize implementation as well as error control, and is able to cope with general time-dependent delays. However, for VIDEs with constant delays and fixed stepsize, we would still need to solve this implicit system of equations on each step and the order of accuracy would not be optimal. In this investigation we will show that for problem (1.1) and methods which determine high order approximations at mesh points, CRKs can be implemented with a significant improvement in efficiency to produce high order approximations for all $t \in [t_0, t_N]$.

Another approach which determines a C^0 interpolant to approximate the solution of (1.1), and which is the approach we will focus on in this paper, is based on collocation formulas. A detailed analysis and justification of this approach for standard DVIDEs can be found in [2,3]. The approach and analysis was extended to apply to problems of the form (1.1) in [16]. Despite the high order approximations at the mesh points, the C^0 interpolant is not as accurate at off-mesh points for these methods. Collocation methods applied to Volterra equations with more general non-vanishing

delays, $\theta(t) := t - \tau(t)$, often impose limitations on the stepsize selection strategy. The points characterized by $\theta(\xi_{\mu+1}) = \xi_\mu$ for $\xi_\mu = t_0, \mu = 0, 1, \ldots$ are called primary discontinuity points. The solutions of DVIDEs can suffer from a loss of regularity at these points [2]; therefore, the detection and inclusion of them as mesh points (constrained mesh) is necessary for a reliable numerical method. Aside from including the discontinuity points in the mesh which is a requirement for any DVIDEs solver, the mesh has to be $\theta-$invariant, i.e., the mesh points within the interval $[\xi_\mu, \xi_{\mu+1}]$ must all be mapped (under $\theta(t) = t - \tau(t)$) onto mesh points within the interval $[\xi_{\mu-1}, \xi_\mu]$ where ξ's are the propagated discontinuity points. In addition, the delay argument is assumed to be increasing. These are strong conditions and are essential for both the convergence analysis and for ease of implementation. Collocation method has been shown to be very effective if a constant stepsize strategy is adopted. For more general delays, the $\theta-$invariant property is unlikely to be satisfied and this makes the method impractical. In this investigation, in contrast to the approach in [17], we assume the collocation method produces an approximate solution on a uniform mesh with superconvergent behavior at mesh points. Then, we introduce a post-processing procedure to construct improved interpolants, using additional *explicit* stages, in order to improve the off-mesh accuracy to be consistent with the mesh-point accuracy. It is worth mentioning that the proposed bootstrapping procedure cab be applied to more general delays based on a variable stepsize implementation as long as the collocation solution with high order accuracy at mesh points is available.

Discontinuous Galerkin method is a different numerical approach to non-standard (non-delay) VIDEs where f (see (1.1)) is linear. It was investigated in [13] and the associated error bounds at the mesh points were established.

Numerical methods generating continuous polynomial approximations to DVIDEs will generally involve solving implicit systems of equations over each subinterval. The size of the implicit system grows as the desired off-mesh accuracy becomes more stringent. In this paper, we show how to improve the accuracy and smoothness of the collocation interpolants for DVIDEs using a few additional **explicit** computations (on each step).

2 Overview of the Collocation Methods

In the application of the collocation approach to (1.1) we will assume that the following standard properties are satisfied:

- f and K are sufficiently smooth functions over their respective domains,
- a *constrained* uniform mesh is employed (i.e. $h := \frac{\tau}{r}$ for some $r \in N$) ,
- the collocation abscissae are chosen to be the Gauss points ($0 < c_1 < c_2 < \ldots < c_s < 1$),
- the collocation solution associated with step n is a vector of piecewise polynomials (each of degree $\leq s$), $u_h(t)$, which is the unique solution of

$$u_h'(t_n + c_i h) = f(t_n + c_i h, u_h(t_n + c_i h)) + \int_{t_n + c_i h - \tau}^{t_n + c_i h} K(t_n + c_i h, s, u_h(t_n + c_i h), u_h(s))ds, \tag{2.1}$$

  subject to the initial condition $u_h(t) := \phi(t)$ when $t \in [-\tau, 0]$.

Note that $u_h(t)$ can be expressed in the form $u_h(t_n+vh) = y_n + h\sum_{j=1}^{s}\beta_j(v)k_j$ where $k_j = u_h'(t_n+c_jh)$, $y_n := u_h(t_n)$, $\beta_j(v) := \int_0^v L_j(s)ds$ and the L_j's are the Lagrangian basis functions associated with the Gauss points. Using this notation, (2.1) can be rewritten as,

$$k_i = f(t_n+c_ih, y_n+h\sum_{j=1}^{s}a_{ij}k_j) + F_n(t_n+c_ih) + hc_i\sum_{l=1}^{s}b_lK(t_n+c_ih, t_n+c_ic_lh, y_n+h\sum_{j=1}^{s}a_{ij}k_j, y_n+h\sum_{j=1}^{s}\beta_j(c_ic_l)k_j), \quad (2.2)$$

where we have set $a_{ij} := \beta_j(c_i)$, $b_l := \beta_l(1)$ for $1 \le i \le s$, $1 \le n \le N-1$. Note that the *lag term* F_n refers to the integrals computed over past intervals. Also, in the implementations we are considering we will use the interpolatory quadrature formulas using the s abscissae based on the Gauss points $\{c_i\}$ to discretize the integral terms. Once this implicit nonlinear system of dimension $m \times s$ is solved, the collocation solution is completely determined. The theory of collocation at Gauss points for the standard and non-standard DVIDEs (see for example [2,3,16]) reveals that for sufficiently small h we achieve local superconvergence at the mesh points, i.e.,

$$\|y(t_i) - u_h(t_i)\|_\infty = O(h^{2s}), \quad i = 0,1,\ldots,N$$

while the global order of convergence achieved by the collocation interpolant is,

$$\|y(t) - u_h(t)\|_\infty = O(h^{s+1}), \quad \text{for} \quad t \in [t_0, T].$$

In the next section, we will demonstrate how to extend the high accuracy that is available at the mesh points to the off-mesh points by constructing new improved continuous approximations.

3 Derivation of Improved Interpolants

Deriving these improved or *superconvergent* interpolants has been investigated for methods for boundary value ordinary differential equations (BVODEs) where the collocation solution is also available throughout the integration interval. The additional computation concerned with the superconvergent interpolant occurs after the collocation software delivers the solution over the integration interval. Some authors have used information available from adjacent subintervals, both backward and forward, to construct an improved interpolant of order $O(h^{2s})$ over each subinterval [15]. Others, restrict the superconvergent interpolant to use information only from within the current subinterval. In [8], the authors use the theoretical framework of continuous Runge-Kutta formulas for IVPs to construct superconvergent interpolants of order $O(h^{2s})$ for BVODEs. They use a continuous version of the well-known (Butcher) *order conditions* as a basis to augment the collocation solution with inexpensive mono implicit Runge-Kutta (MIRK) stages. This scheme uses the minimum number of additional stages required to achieve the desired accuracy. Runge-Kutta theory for Volterra integro-differential equations was developed by Lubich in [12]. Later,

Brunner showed that the collocation method applied to Volterra integro-differential equations can be interpreted as an *extended* $s-$stage implicit *Pouzet* Runge-Kutta method [2] with optimal order $2s$ at mesh points, where s is the number of Gauss points. Therefore, it might be possible to develop numerical schemes, based on direct use of the continuous version of the collocation-based RK order conditions for VIDEs, to obtain high accuracy at off-mesh points by re-using the collocation stages already computed. In this case, the additional stages are not computed as straightforward as in BVODEs due to the presence of lag terms, and we might even need to use implicit stage derivatives which is undesirable. In addition to complicated algebraic (order condition) equations that arise, this approach is restricted to first order DVIDEs.

An alternative technique, uses the *bootstrapping* approach, first introduced in [7] for constructing a sequence of continuous interpolants for IVPs using the discrete information produced by a RK method. It was subsequently used for constructing high order interpolants for BVODEs in [9]. This scheme does not necessarily use the fewest number of extra stages, but is straightforward to apply to higher-order differential equations. Both these approaches have been developed for ODEs where the underlying equation possesses sufficient smoothness. We introduce an extension of the bootstrapping approach which can be used for the DVIDE (1.1).

Given the collocation solution $u_h(t)$ of (1.1) we define, on step n,

$$\begin{aligned} \tilde{k}_{0,1} &:= f(t_n, y_n) + \int_{t_n-\tau}^{t_n} K(t_n, s, y_n, u_h(s))ds \\ \tilde{k}_{0,2} &:= f(t_{n+1}, y_{n+1}) + \int_{t_{n+1}-\tau}^{t_{n+1}} K(t_{n+1}, s, y_{n+1}, u_h(s))ds \end{aligned} \tag{3.1}$$

and we construct a Hermite-Birkhoff polynomial of degree at most $s+1$, $u^{(1)}(t)$, interpolating mesh data y_n, y_{n+1}, derivative data $\tilde{k}_{0,1}$ and $\tilde{k}_{0,2}$ and $s-2$ additional stages,

$$\begin{aligned} \tilde{k}_{0,2+i} &:= f(t_n + \rho_i^{(0)} h, u_h(t_n + \rho_i^{(0)} h)) \\ &+ \int_{t_n+\rho_i^{(0)}h-\tau}^{t_n+\rho_i^{(0)}h} K(t_n + \rho_i^{(0)} h, s, u_h(t_n + \rho_i^{(0)} h), u_h(s))ds, \\ i &= 1, \ldots, s-2, \end{aligned}$$

where $\rho_i^{(0)}$'s are arbitrary abscissae in $(0,1)$. The interpolant $u^{(1)}(t)$ can be written as,

$$\begin{aligned} u^{(1)}(t_n + \theta h) &= b_1(\theta) y_n + b_2(\theta) y_{n+1} \\ &+ h b_3(\theta) \tilde{k}_{0,1} + h b_4(\theta) \tilde{k}_{0,2} + h \sum_{i=1}^{s-2} b_{i+4}(\theta) \tilde{k}_{0,2+i}, \end{aligned} \tag{3.2}$$

where $\theta = \frac{(t-t_n)}{h}$ and the $b_i(\theta)$'s are known polynomials of degree $\leq s+1$. This selection of $\tilde{k}_{0,1}$ and $\tilde{k}_{0,2}$ reduces the cost of determining the stages needed on each step since $\tilde{k}_{0,2}$ for step n equals $\tilde{k}_{0,1}$ for step $n+1$. We then introduce, for the purpose

of analysis, the Hermite-Birkhoff polynomial $y^{(1)}(t)$ interpolating the exact solutions $y(t_n)$, $y(t_{n+1})$, the exact derivative data $\bar{k}_{0,1}$ and $\bar{k}_{0,2}$,

$$\bar{k}_{0,1} := f(t_n, y(t_n)) + \int_{t_n-\tau}^{t_n} K(t_n, s, y(t_n), y(s))ds,$$
$$\bar{k}_{0,2} := f(t_{n+1}, y(t_{n+1})) + \int_{t_{n+1}-\tau}^{t_{n+1}} K(t_{n+1}, s, y(t_{n+1}), y(s))ds.$$

and exact additional $s-2$ derivative data,

$$\begin{aligned}\bar{k}_{0,2+i} :=\ & f(t_n + \rho_i^{(0)}h, y(t_n + \rho_i^{(0)}h)) \\ & + \int_{t_n+\rho_i^{(0)}h-\tau}^{t_n+\rho_i^{(0)}h} K(t_n + \rho_i^{(0)}h, s, y(t_n + \rho_i^{(0)}h), y(s))ds, \\ i \ =\ & 1, \ldots, s-2\end{aligned}$$

The interpolant $y^{(1)}(t)$ can then be written as,

$$\begin{aligned}y^{(1)}(t_n + \theta h) =\ & b_1(\theta)y(t_n) + b_2(\theta)y(t_{n+1}) \\ & + hb_3(\theta)\bar{k}_{0,1} + hb_4(\theta)\bar{k}_{0,2} + h\sum_{i=1}^{s-2} b_{i+4}(\theta)\bar{k}_{0,2+i}\end{aligned} \tag{3.3}$$

Since constructed mesh ensures that the primary discontinuity points are included in the set of mesh points, the exact solution $y(t)$ is sufficiently smooth over (t_n, t_{n+1}). Therefore, according to the theory of Hermite-Birkhoff interpolation we have,

$$\|y(t) - y^{(1)}(t)\| = O(h^{s+2})$$

In addition, since the available approximate solutions y_n and y_{n+1} are $O(h^{2s})$, and $u_h(t)$ approximates the exact solution $y(t)$ to $O(h^{s+1})$, from (3.2) and (3.3) we have (assuming the boundedness of partial derivatives or suitable Lipschitz conditions for f and K),

$$\|y^{(1)}(t) - u^{(1)}(t)\| = O(h^{s+2})$$

and, using the triangle inequality, we obtain,

$$\|y(t) - u^{(1)}(t)\| = O(h^{s+2}).$$

As seen above, for collocation based on $s = 2$ Gauss points we get the optimal order by computing only two extra explicit derivative values $\tilde{k}_{0,1}$ and $\tilde{k}_{0,2}$. For $s > 2$ it is natural to choose the ρ_i's to be a subset of the Gauss points because these values are already available from the original collocation computations. We now consider constructing an interpolant $u^{(2)}(t)$ of degree $s+2$ by interpolating the mesh data y_n, y_{n+1} and additional $s+1$ stages,

$$\begin{aligned}\tilde{k}_{1,s+i} :=\ & f(t_n + \rho_i^{(1)}h, u^{(1)}(t_n + \rho_i^{(1)}h)) \\ & + \int_{t_n+\rho_i^{(1)}h-\tau}^{t_n+\rho_i^{(1)}h} K(t_n + \rho_i^{(1)}h, s, u^{(1)}(t_n + \rho_i^{(1)}h), u^{(1)}(s))ds, \\ i \ =\ & 1, \ldots, s+1\end{aligned} \tag{3.4}$$

Notice that the previously defined end point derivative data $\tilde{k}_{0,1}$ and $\tilde{k}_{0,2}$ are not accurate enough to be used in this case. Also, $u^{(2)}(t)$ is only in $C^0[t_0,T]$ unless the set of abscissae $\{\rho_i^{(1)};\quad i=1,\ldots,s+1\}$ contains the subset $\{0,1\}$. Applying a similar analysis as above, we conclude that the new constructed interpolant satisfies the following error bound,

$$\|y(t)-u^{(2)}(t)\| = O(h^{s+3})$$

The bootstrapping algorithm proceeds in a similar manner and produces a sequence of interpolants until an interpolant of the desired order is obtained. In general, having computed the interpolant $u^{(q)}(t), q \geq 1$ of order $O(h^{s+q+1})$, the new interpolant $u^{(q+1)}(t)$ of order $O(h^{\min\{s+q+2,2s\}})$ is constructed as,

$$u^{(q+1)}(t_n+\theta h) = b_1(\theta)y_n + b_2(\theta)y_{n+1} + h\sum_{i=1}^{s+q} b_{i+2}(\theta)\tilde{k}_{q,i} \quad ,$$

where $b_i(\theta)$'s are Hermite-Birkhoff polynomials of degree $s+q+1$ (for ease of notation, we have used the notation b_1, b_2 and b_{i+2} for all intermediate and optimal interpolants while they have different meanings depending on the selection of abscissae) and,

$$\begin{aligned}\tilde{k}_{q,i} :=\ & f(t_n+\rho_i^{(q)}h, u^{(q)}(t_n+\rho_i^{(q)}h)) \\ & + \int_{t_n+\rho_i^{(q)}h-\tau}^{t_n+\rho_i^{(q)}h} K(t_n+\rho_i^{(q)}h, s, u^{(q)}(t_n+\rho_i^{(q)}h), u^{(q)}(s))ds, \\ i\ =\ & 1,\ldots,s+q \end{aligned} \tag{3.5}$$

In order for $u^{(q+1)}(t)$ to be C^1 continuous, the set of abscissae corresponding to $\{\tilde{k}_{q,i}; i=1,\ldots,s+q\}$ has to contain $\{0,1\}$. The error bound

$$\|y(t)-u^{(q+1)}(t)\| \sim O(h^{\min\{s+q+2,2s\}})$$

is valid for any selection of distinct abscissae in $[0,1]$ used on each step of the bootstrapping process.

The algorithm requires storing not only the optimal-order interpolants for each of the past subintervals, but also all intermediate polynomials in the past r steps. This requirement could be dropped by using the optimal-order interpolants in computing the lag terms appearing in stage calculations. This leads to a variation of the outlined

Table 3.1 Number of additional explicit stages required to construct a superconvergent interpolant of order $2s$

s	1	2	3	4
Number of additional stages	-	2	5	11

algorithm for which we have,

$$\begin{aligned}\tilde{k}_{q,i} := & f(t_n+\rho_i^{(q)}h, u^{(q)}(t_n+\rho_i^{(q)}h)) \\ & + \int_{t_n+\rho_i^{(q)}h-\tau}^{t_n} K(t_n+\rho_i^{(q)}h, s, u^{(q)}(t_n+\rho_i^{(q)}h), u^{(s-1)}(s))ds \\ & + \int_{t_n}^{t_n+\rho_i^{(q)}h} K(t_n+\rho_i^{(q)}h, s, u^{(q)}(t_n+\rho_i^{(q)}h), u^{(q)}(s))ds, \\ i = & \ 1,\ldots,s+q\end{aligned}$$

where $u^{(s-1)}(s)$ is the optimal-order interpolant of order $O(h^{2s})$, already computed in the past steps, and $u^{(q)}(s)$ is the most updated interpolant in the current step. One obvious advantage of this variation is that $\tilde{k}_{0,1}$ can be re-used in all bootstrapping steps and this reduces the number of required additional stages. Table 3.1 reports, for this variation, the total number of additional stages required for constructing superconvergent interpolants of order $O(h^{2s})$ for various values of s.

An important point regarding this algorithm is that, although the lower order stages belonging to previous steps are not re-used, all additional stages are computed explicitly despite the presence of an integral term evaluated over the current subinterval of integration. This will only be true if the improved interpolants are each computed sequentially in a step by step manner. It should be noted that in addition to the roundoff and truncation errors that already affect the results, we have additional sources of error when treating integral equations, i.e., quadrature discretization error and iteration error arising in the solution of the associated implicit equations. These additional sources of error contribute directly to the accuracy of the approximate solution.

As mentioned earlier, augmenting the high order discrete collocation solution has also been investigated for BVODEs [8,9]. A question arises whether the most effective superconvergent interpolants proposed for BVODEs can be used for extending the discrete collocation solution of DVIDEs. If so, the resulting method could be more efficient because the interpolants derived for BVODEs use a smaller number of extra explicit stages. Table 3.2 reports the number of additional stages required to construct superconvergent interpolants based on these BVODE formulas. It should be remarked that $\tilde{k}_{0,1}$ and $\tilde{k}_{0,2}$ (derivative data at end points) are calculated using only two function evaluations in the case of collocation applied to BVODEs while for DVIDEs they would be more expensive to compute due to the quadrature discretization and lag term calculations. Also, these values are re-used in all subsequent steps of the bootstrapping process for BVODEs while new derivative values (at least the

Table 3.2 Number of additional explicit stages based on BVODE interpolants of order $2s$

s	1	2	3	4
Number of additional stages	-	2	4	9

right end point derivative) have to be computed for DVIDEs on each bootstrapping step. This reduces the number of additional stages required for BVODE formulas compared to those in table 3.1.

Suppose we are stepping from t_n to t_{n+1}. We can define a local 'ODE' problem associated with the DVIDE (1.1) as:

$$\begin{aligned} z_n'(t) &= f(t, z_n(t)) + \int_{t-\tau}^{t} K(t, s, z_n(t), u_h(s))ds, \\ z_n(t_n) &= y_n \end{aligned} \tag{3.6}$$

where $u_h(s)$ in the integrand is the available low order collocation polynomial computed up to t_{n+1}. Repeated differentiation of the right hand side of (3.6) using the Leibnitz rule reveals that we have the same order of smoothness for this local problem as we have for f and K. This is because when $t \in [t_{n-r}, t_{n-r+1}]$, the collocation solution u_h is either the previously computed collocation polynomial of degree s or the predefined smooth function ϕ in case $t_{n-r+1} < t_0$. The local problem (3.6) could then be written in the following *smooth* ODE-equivalent form:

$$\begin{aligned} z_n'(t) &= \tilde{f}(t, z_n(t)), \\ z_n(t_n) &= y_n \end{aligned}$$

where $\tilde{f}$ has an obvious meaning. If we now apply the collocation method based on the Gauss points to this smooth ODE on the single interval $[t_n, t_{n+1}]$, we will get the collocation solution $\tilde{u}_h$ for which we have the following error bound,

$$\|z_n(t) - \tilde{u}_h(t)\|_\infty \sim O(h^{s+1}), \quad \text{for} \quad t \in [t_n, t_{n+1}]$$

It is easy to see that $\tilde{u}_h(t) \equiv u_h(t)$ where $u_h(t)$ is the collocation solution satisfying (2.1). Therefore, available superconvergent interpolants for BVODEs can now be employed for this local ODE to obtain an accurate **local** interpolant $\bar{u}_h(t)$ with the following error bound,

$$\|z_n(t) - \bar{u}_h(t)\|_\infty \sim O(h^{2s}), \quad \text{for} \quad t \in [t_n, t_{n+1}].$$

Notice that this error bound could be improved to $O(h^{2s+1})$, the optimal local accuracy of the superconvergent interpolant. Although the local order of accuracy is significantly improved by employing the interpolants constructed for ODEs, the error bound for the global accuracy remains $O(h^{s+1})$ since,

$$\|y(t) - z_n(t)\|_\infty \sim O(h^{s+1}), \quad \text{for} \quad t \in [t_n, t_{n+1}].$$

due to the implicit presence of low order collocation polynomial $u_h(t)$ in the definition of $\tilde{f}$. Any attempt to increase the accuracy of the interpolant appearing in the

integrand will result in emergence of implicit systems of equations similar to the situation for CRK methods. At the end, it has to be remarked that to achieve this local accuracy, $\tilde{f}$ has to be a smooth function. This would be the case when the mesh is necessarily $\theta-$invariant. We note that the number of Gauss points does not usually exceed 4 in practice. Now, as examples of our general approach, we derive specific interpolation schemes for $s = 2$ and $s = 3$.

3.1 Bootstrapping Interpolant for s = 2

For $s = 2$ it is straightforward to derive the interpolant $u^{(1)}(t) \in C^1[t_0, T]$ as we need no bootstrapping step. The interpolant is represented as,

$$u^{(1)}(t_n + \theta h) = b_1(\theta)y_n + b_2(\theta)y_{n+1} + hb_3(\theta)\tilde{k}_{0,1} + hb_4(\theta)\tilde{k}_{0,2}$$

where,

$$\begin{aligned} b_1(\theta) &= 2\theta^3 - 3\theta^2 + 1, \quad b_2(\theta) = -2\theta^3 + 3\theta^2, \\ b_3(\theta) &= \theta^3 - 2\theta^2 + \theta, \quad b_4(\theta) = \theta^3 - \theta^2, \end{aligned}$$

and $\tilde{k}_{0,1}, \tilde{k}_{0,2}$ are computed as in (3.1).

3.2 Bootstrapping Interpolant for s = 3

In this case we need two bootstrapping steps. In the first step, (i.e., deriving a 5^{th} order piecewise polynomial interpolant) we use the first Gauss point $\rho_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}$, whose associated stage k_1 (see (2.2)) is already available from the collocation solution, as an additional abscissa which yields,

$$k_1 = f(t_n + \rho_1 h, u_h(t_n + \rho_1 h)) + \int_{t_n+\rho_1 h-\tau}^{t_n+\rho_1 h} K(t_n + \rho_1 h, s, u_h(t_n + \rho_1 h), u_h(s))ds,$$

$$u^{(1)}(t_n + \theta h) = b_1(\theta)y_n + b_2(\theta)y_{n+1} + h(b_3(\theta)\tilde{k}_{0,1} + b_4(\theta)\tilde{k}_{0,2} + b_5(\theta)k_1),$$

$$\begin{aligned} b_1(\theta) &= \sqrt{15}\theta^4 + (2 - 2\sqrt{15})\theta^3 + (\sqrt{15} - 3)\theta^2 + 1, \\ b_2(\theta) &= -\sqrt{15}\theta^4 + (-2 + 2\sqrt{15})\theta^3 + (3 - \sqrt{15})\theta^2, \\ b_3(\theta) &= (-\frac{5}{2} - \frac{\sqrt{15}}{3})\theta^4 + (6 + \frac{2\sqrt{15}}{3})\theta^3 + (\frac{9}{2} - \frac{\sqrt{15}}{3})\theta^2 + \theta, \\ b_4(\theta) &= (\frac{5}{2} - \frac{\sqrt{15}}{3})\theta^4 + (-4 + \frac{2\sqrt{15}}{3})\theta^3 + (\frac{3}{2} - \frac{\sqrt{15}}{3})\theta^2, \\ b_5(\theta) &= \frac{5\sqrt{15}}{3}\theta^4 - \frac{10\sqrt{15}}{3}\theta^3 + \frac{5\sqrt{15}}{3}\theta^2. \end{aligned}$$

In order to construct an optimal 6^{th} order (minimum number of stages) C^1 interpolant, we need four additional abscissae. We performed a search over 2 free parameters (abscissae) through various numerical experiments and chose $\rho_1^{(1)}=0, \rho_2^{(1)}=\frac{1}{2}, \rho_3^{(1)}=\frac{2}{3}, \rho_4^{(1)}=1$ as a result of an attempt to minimize the amplitudes of the Hermite-Birkhoff basis polynomials (note that $\rho_1^{(1)}$ and $\rho_4^{(1)}$ are correspondingly assumed 0 and 1 to ensure C^1 continuity). Having computed $\tilde{k}_{1,4}$, $\tilde{k}_{1,5}$, $\tilde{k}_{1,6}$ and $\tilde{k}_{1,7}$ according to (3.4) we obtain the 6^{th} order interpolant $u^{(2)}(t)$ as,

$$u^{(2)}(t_n+\theta h) = b_1(\theta)y_n + b_2(\theta)y_{n+1} + h\big(b_3(\theta)\tilde{k}_{1,4} + b_4(\theta)\tilde{k}_{1,5} + b_5(\theta)\tilde{k}_{1,6} + b_6(\theta)\tilde{k}_{1,7}\big),$$

where the Hermite-Birkhoff basis polynomials, for this choice of $\rho_1^{(1)}, \rho_2^{(1)}, \rho_3^{(1)}$ and $\rho_4^{(1)}$ are,

$$\begin{aligned}
b_1(\theta) &= 24\theta^5 - 65\theta^4 + 60\theta^3 - 20\theta^2 + 1,\\
b_2(\theta) &= -24\theta^5 + 65\theta^4 - 60\theta^3 + 20\theta^2,\\
b_3(\theta) &= 4\theta^5 - \frac{139}{12}\theta^4 + \frac{73}{6}\theta^3 - \frac{67}{12}\theta^2 + \theta,\\
b_4(\theta) &= 16\theta^5 - \frac{112}{3}\theta^4 + \frac{80}{3}\theta^3 - \frac{16}{3}\theta^2,\\
b_5(\theta) &= -\frac{27}{4}\theta^4 + \frac{27}{2}\theta^3 - \frac{27}{4}\theta^2,\\
b_6(\theta) &= 4\theta^5 - \frac{28}{3}\theta^4 + \frac{23}{3}\theta^3 - \frac{7}{3}\theta^2.
\end{aligned}$$

We have performed several numerical tests using various sets of abscissae, resulting in either C^0 (for example with $\rho_1^{(1)}=\frac{1}{3}, \rho_2^{(1)}=\frac{4}{5}, \rho_3^{(1)}=\frac{1}{5}, \rho_4^{(1)}=\frac{2}{3}$) or C^1 interpolants and, in each case observed the predicted improvement for order of convergence.

4 Numerical Results

In this section we present our numerical investigations of the improved interpolants derived in Sect. 3 (based on the procedure outlined in 3.1-3.5). The observed orders of convergence for the discrete collocation solution, the collocation interpolant and the improved interpolant are reported for $s=2,3$ for one typical problem. The problem is the well-known Volterra population system of 'predator-prey' dynamics (1.2) with the following set of parameters [16],

$$\varepsilon_1 = 0.02\,,\ \varepsilon_2 = 1,\ \gamma_1 = 1\,,\ \gamma_2 = 1,\ \tau = 0.2\,,\ T = 2.$$

The history functions are chosen to be $\phi_1(t)=\phi_2(t)=3$, $t\in[-\tau,0]$. We also set $F_1(t)=F_2(t)=\frac{1}{2!}t^3e^{-3t}$. This problem doesn't have a known closed form solution. A reference solution is determined using collocation at 5 Gauss points over a fine mesh of 320 subintervals. The observed rate of convergence is estimated in two ways. First,

the approximate solution is evaluated at a large number of non-mesh sample points on each subinterval. Then, **RI** is computed as the base 2 logarithm of the ratio of the maximum observed value of the global error associated with v_i to that of v_{i+1}, where v_i and v_{i+1} are correspondingly vector approximate solutions associated with two consecutive sample mesh (the number of mesh points is doubled for each increase in i.)

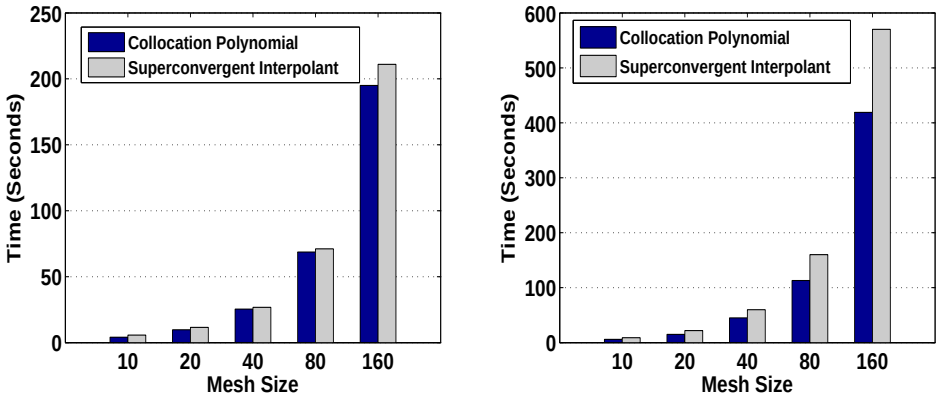
To calculate a second estimate, denoted **RII** (that can be computed without any knowledge of the reference solution), a total of 320 equally-spaced sample points are identified. These sample points are used to estimate the order of convergence by observing how well the coarse mesh solutions agree as i increases. Suppose that v_i, v_{i+1} and v_{i+2} are correspondingly vector approximate solutions associated with three consecutive sample meshe evaluated at the same common sample points. Then, the rate of convergence can be estimated by **RII** defined as $\log_2(\frac{\max|v_{i+1}-v_i|}{\max|v_{i+2}-v_{i+1}|})$.

For $s = 2,3$, tables 4.1 and 4.2 report **RI** and **RII** for the collocation polynomial and the superconvergent interpolant. In addition, these estimates of the observed order are reported for the collocation solution at the mesh points which represent the order of convergence for the discrete collocation solution. **GE** stands for the maximum global error over the sample mesh, and **Thr. O.** indicates the theoretical expected order of convergence. Tables 4.1 and 4.2 show that the expected theoretical order of convergence is achieved for the superconvergent interpolants on this problem. Our numerical experiments demonstrate that the reported rates of convergence become less consistent with that predicted by our analysis as the sample mesh becomes finer and the accuracy of the approximate solution starts to approach the accuracy of the generated reference solution. This could be resolved by carrying out the computations for the reference solution in higher precision. While the superconvergent interpolants do exhibit the same order, they seem to be less accurate than the discrete collocation solution. However, we should note that both RI and RII are calculated by sampling at a large number of non-mesh points relative to the number of mesh points. Also, the additional explicit calculations for the superconvergent interpolants will result in larger roundoff and discretization errors.

We ran several experiments by increasing the accuracy of the derivative data (by increasing the optimal number of stages by one and generating new polynomials, or alternatively by recalculating the derivative data in the last stage based on the computed high-order optimal interpolant) and monitored the global error associated with the superconvergent interpolant. The global errors associated with the improved interpolants become closer to the discrete global error for $s = 2$, however, the orders of convergence start to become less consistent due to the reason mentioned above. For $s = 3$, the roundoff error resulting from the stage calculations, polynomial evaluations, etc. can not be ignored and indicates the need for higher precisions if improved optimal-order interpolants are to be realized. As seen in table 4.3, for $s = 2$, the global errors become closer to the discrete global errors (table 4.1); however, the orders become less consistent with the theory. The new superconvergent interpolant for this test was generated by recalculating stage derivatives using the most recent optimal interpolant and also using one additional stage derivative evaluated at abscissa $c = \frac{2}{3}$. For more examples of such phenomena in case of boundary value ODEs see [9].

We also ran the same numerical experiments using the improved variation of the bootstrapping process. The global errors were almost identical with those reported in the tables 4.1 and 4.2, however this variation required less storage space.

Figure 4.1 displays the execution time required for the computation of the collocation polynomial and also the total time for computing both the underlying collocation polynomial and the associated superconvergent interpolant for various mesh sizes with $s = 2, 3$. As can be seen from the figure, the major execution cost is for the original collocation calculations which involve solving a nonlinear system of equations on each subinterval. Another major cost is the construction of the superconvergent interpolant after the the collocation solution has been computed. However, as these figures show, this is only a fraction of the original collocation cost. The code has been implemented in a numeric-symbolic fashion and therefore the execution times are not optimal. However, it allows a comparison between the setup of the collocation and superconvergent interpolants. All computations are carried out in Matlab in double precision.



(a) s = 2 (b) s = 3

Fig. 4.1 Execution Times for s = 2 and s = 3 Gauss Points

Acknowledgements This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

1. Bocharov, G.A., Rihan, F.A.: Numerical modelling in biosciences using delay differential equations. J Comput. Appl. Math. 125 (2000), pp. 183-199.
2. Brunner, H.: Collocation Methods for Volterra Integral and Related Functional Equations. Cambridge University Press, Cambridge, UK, 2004.

Table 4.1 Accuracy and observed order on predator-prey problem with different mesh size for $s = 3$

y(t)	N	Discrete Solution		Collocation Polynomial			Superconvergent Interpolant		
		GE	RI	GE	RI	RI	GE	RI	RI
$N_1(t)$	10	1.89×10^{-4}	-	4.24×10^{-3}	-	-	3.14×10^{-3}	-	-
	20	1.22×10^{-5}	3.9	7.85×10^{-4}	-	2.4	1.98×10^{-4}	-	3.9
	40	7.54×10^{-7}	4.0	1.13×10^{-4}	2.4	2.7	1.27×10^{-5}	3.9	3.9
	80	4.69×10^{-8}	4.0	1.50×10^{-5}	2.8	2.9	7.92×10^{-7}	4.0	4.0
	160	2.93×10^{-9}	4.0	1.92×10^{-6}	3.2	2.9	4.97×10^{-8}	4.1	3.9
$N_2(t)$	10	4.11×10^{-4}	-	3.98×10^{-3}	-	-	3.32×10^{-3}	-	-
	20	2.64×10^{-5}	3.9	6.19×10^{-4}	-	2.6	2.18×10^{-4}	-	3.9
	40	1.64×10^{-6}	4.0	9.49×10^{-5}	2.7	2.7	1.37×10^{-5}	3.9	3.9
	80	1.03×10^{-7}	3.9	1.29×10^{-5}	2.7	2.8	8.62×10^{-7}	4.0	3.9
	160	6.46×10^{-9}	3.9	1.67×10^{-6}	3.1	2.9	5.38×10^{-8}	4.1	4.0
Thr. O.		4		3			4		

Table 4.2 Accuracy and observed order on predator-prey problem with different sample mesh for $s = 3$

y(t)	N	Discrete Solution		Collocation Polynomial			Superconvergent Interpolant		
		GE	RI	GE	RI	RI	GE	RI	RI
$N_1(t)$	10	5.07×10^{-7}	-	6.06×10^{-4}	-	-	3.32×10^{-5}	-	-
	20	7.49×10^{-9}	6.0	3.87×10^{-5}	-	3.9	5.81×10^{-7}	-	5.8
	40	1.26×10^{-10}	5.8	2.54×10^{-6}	3.9	3.9	8.82×10^{-9}	5.8	6.0
	80	2.14×10^{-12}	5.8	1.58×10^{-7}	4.0	4.0	1.41×10^{-10}	6.0	5.9
	160	3.37×10^{-14}	5.9	9.93×10^{-9}	4.2	3.9	2.23×10^{-12}	5.9	5.9
$N_2(t)$	10	1.56×10^{-6}	-	6.35×10^{-4}	-	-	3.60×10^{-5}	-	-
	20	2.40×10^{-8}	6.0	4.24×10^{-5}	-	3.9	7.52×10^{-7}	-	5.5
	40	3.99×10^{-10}	5.9	2.69×10^{-6}	3.8	3.9	1.19×10^{-8}	5.5	5.9
	80	6.23×10^{-12}	6.0	1.69×10^{-7}	4.0	3.9	1.83×10^{-10}	5.9	6.0
	160	9.76×10^{-14}	5.9	1.05×10^{-8}	4.2	4.0	2.85×10^{-12}	5.9	6.0
Thr. O.		6		4			6		

Table 4.3 Error Rates for s = 2 with additional stage evaluations

y(t)	N	Superconvergent Interpolant		
		GE	RI	RI
$N_1(t)$	10	2.47×10^{-4}	-	-
	20	1.24×10^{-5}	-	4.3
	40	8.37×10^{-7}	4.3	3.8
	80	1.39×10^{-7}	3.8	2.5
	160	3.64×10^{-8}	2.8	1.9
$N_2(t)$	10	5.34×10^{-4}	-	-
	20	2.70×10^{-5}	-	4.3
	40	1.90×10^{-6}	4.3	3.8
	80	2.86×10^{-7}	3.7	2.7
	160	8.83×10^{-8}	2.7	1.7
Thr. O.		4		

3. Brunner, H.: The numerical solution of neutral Volterra integro-differential equations with delay arguments. Ann. Numer. Math. 1 (1994), pp. 309-322.
4. Brunner, H., Van der Houwen, P.J.: The Numerical Solution of Volterra Equations. CWI Monographs, vol. 3, North-Holland, Amsterdam, 1986.
5. Cushing, J. M.: Integrodifferential Equations and Delay Models in Population Dynamics. Lecture Notes in Biomathematics 20, Springer, Berlin , 1977.
6. De Gaetano, A. , Arino, O.: Mathematical modelling of the intravenous glucose tolerance test. J. Math. Biol., 40 (2000), pp. 136-168.
7. Enright, W. H., Jackson, K. R., Nørsett, S. P., Thomsen, P. G.: Interpolants for Runge-Kutta formulas. ACM Transactions on Mathematical Software (TOMS) 12 (1986), pp. 193-218.
8. Enright, W. H. , Muir, P. H.: Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations. SIAM J Sci. Comput. 21 (1999), pp. 227-254.
9. Enright, W. H., Sivasothinathan, R.: Superconvergent interpolants for collocation methods applied to mixed-order BVODEs. ACM Transactions on Mathematical Software (TOMS) 26 (2000), pp. 323-351.
10. Gopalsamy, K.: Stability and Oscillation in Delay Differential Equations of Population Dynamics. Kluwer Academic Publishers, Boston, 1992.
11. Kuang, Y.: Delay Differential Equations with Applications in Population Dynamic. Academic Press, San Diego, CA, 1993.
12. Lubich, C.: Runge-Kutta theory for Volterra integro-differential equations. Numer. Math. 40 (1982), pp. 119-135.
13. Ma, J., Brunner, H.: A posteriori error estimates of discontinuous Galerkin methods for non-standard Volterra integro-differential equations. IMA J. Numer. Anal. 26 (2006), pp. 78-95.
14. Marino, S., Beretta, E., Kirschner, D. E.: The role of delays in innate and adaptive immunity to intracellular bacterial infection. Math. Biosci. Eng. 4 (2007), pp. 261-88.
15. Pruess S.: Interpolation schemes for collocation solutions of two point boundary value problems. SIAM J. Sci. Comput. 7 (1986), pp. 322-333.
16. Shakourifar, M., Dehghan, M.: On the numerical solution of nonlinear systems of Volterra integro-differential equations with delay arguments. Computing 82 (2008), pp. 241-260.
17. Shakourifar, M., Enright, W.H.: Reliable approximate solution of systems of Volterra integro-differential equations with time-dependent delays. SIAM J Sci. Comput. to appear.
18. Volterra, V.: Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. Memorie del R. Comitato talassografico italiano 43 (1927), pp. 1-142.
19. Volterra, V.: The general equations of biological strife in the case of historical actions. Proc. Edinburgh Math. Soc., 2 (1939), pp. 4-10.
20. Wang, W. S., Li, S. F.: Convergence of Runge-Kutta methods for neutral Volterra delay-integro-differential equations. Front. Math. China., 4 (2009), pp. 195-216.
21. Yu, Y., Wen, L., Li, S.: Nonlinear stability of RungeKutta methods for neutral delay integro-differential equations. Appl. Math. Comput. 191 (2007), pp. 543-549.
22. Zhang, C., Vandewalle, S.: Stability analysis of RungeKutta methods for nonlinear Volterra delay-integro-differential equations. IMA J. Numer. Anal., 24 (2004), pp. 193-214.