J Sci Comput (2018) 75:1757–1775
https://doi.org/10.1007/s10915-017-0608-3

The Linear Barycentric Rational Method for a Class of Delay Volterra Integro-Differential Equations

Ali Abdi[1] · **Jean–Paul Berrut**[2] ·
Seyyed Ahmad Hosseini[3]

Received: 21 February 2017 / Revised: 7 November 2017 / Accepted: 10 November 2017 /
Published online: 22 November 2017

Abstract A method for solving delay Volterra integro-differential equations is introduced. It is based on two applications of linear barycentric rational interpolation, barycentric rational quadrature and barycentric rational finite differences. Its zero–stability and convergence are studied. Numerical tests demonstrate the excellent agreement of our implementation with the predicted convergence orders.

Keywords Delay Volterra integro-differential equations · Linear barycentric rational method · Barycentric rational quadrature · Barycentric rational finite differences

Mathematics Subject Classification 65R20 · 45L05 · 65L20

1 Introduction

Natural phenomena in which a quantity varies in time and simultaneously depends on its past values are often modeled with Volterra integral equations. When these past values arise at a shifted abscissa as well as at present time, one speaks of an equation with delay or a delay

The results reported in this paper were obtained during the visit of the first and third authors to the University of Fribourg in 2016, which was supported by the program for research stays of that University. These authors wish to express their gratitude to J.-P. Berrut for making this visit possible.

✉ Jean–Paul Berrut
jean-paul.berrut@unifr.ch

Ali Abdi
a_abdi@tabrizu.ac.ir

Seyyed Ahmad Hosseini
a.hosseini@gu.ac.ir

[1] Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

[2] Department of Mathematics, University of Fribourg, Pérolles, 1700 Fribourg, Switzerland

[3] Department of Mathematics, Faculty of Sciences, Golestan University, Gorgan, Iran

equation. A classical system of delay Volterra integral equations (DVIEs) of the second kind is therefore often assumed to be

$$\begin{cases} y(t) = f\Big(t, y(t), y(t-\tau), \displaystyle\int_{t-\tau}^{t} K(t,s,y(s))\,\mathrm{d}s\Big), & t \in [t_0, T], \\ y(t) = \varphi(t), & t \in [t_0-\tau, t_0], \end{cases}$$

in which $y : \mathbb{R} \to \mathbb{R}^{D_1}$ is the unknown function and $K : \mathbb{R}^2 \times \mathbb{R}^{D_1} \to \mathbb{R}^{D_2}$, $D_1, D_2 \in \mathbb{N}$, the given so-called *kernel* of the integral operator characterizing the equation together with the given functions $f : \mathbb{R} \times \mathbb{R}^{D_1} \times \mathbb{R}^{D_1} \times \mathbb{R}^{D_2} \to \mathbb{R}^{D_1}$ and $\varphi : \mathbb{R} \to \mathbb{R}^{D_1}$; φ is the initial function and $\tau > 0$ is the (constant) delay. (An equation of the first kind is one in which the lone y on the left-hand side is absent, i.e., zero.)

In some models, f rather describes the behavior of the derivative of the unknown: one then obtains a system of so-called delay Volterra integro-differential equations (DVIDEs)

$$\begin{cases} y'(t) = f\Big(t, y(t), y(t-\tau), \displaystyle\int_{t-\tau}^{t} K(t,s,y(s))\,\mathrm{d}s\Big), & t \in [t_0, T], \\ y(t) = \varphi(t), & t \in [t_0-\tau, t_0] \end{cases} \tag{1}$$

for the same classes of functions as above. Practical examples of (1) are the modelling of the cohabitation of different biological species [19] or of the human immune response [25]. In the present paper, we shall introduce a numerical procedure for the solution of systems of equations such as (1). We thereby assume that the given functions f, K and φ are smooth enough for the system to have a unique solution y (for conditions guaranteeing this, see [34]). We further assume that the initial function φ is compatible with the solution y in the sense that $\varphi \in C^{d+2}[t_0 - \tau, t_0]$ and for the derivatives at t_0,

$$\varphi^{(j)}(t_0-) = y^{(j)}(t_0+), \qquad j = 0, \ldots, d+2, \tag{2}$$

where d is the parameter in the barycentric approximation which lies at the basis of our method, see the convergence theorem (Theorem 1). This eliminates the risk of primary singularities in the solution at multiples of τ [11,24,26], singularities which do not occur in most applications of DVIDEs in the natural sciences. (In fact, it would perhaps be preferable to call this property "consistency", but this term is already taken in the context of functional equations.)

In view of the advance in time, it is very natural, at least if nothing particular is known about the behavior of the solution, to approximate it at equispaced values of the t-variable, i.e., to calculate values $y_m \approx y(t_m)$, $t_m = t_0 + mh$. Several methods are already available [12,17,27,33–36]. Most of them, in particular the very popular Runge–Kutta and collocation ones, consider and/or compute values of y and/or f at intermediate points as well, e.g., Legendre points stretched to fit into the interval $[t_m, t_{m+1}]$; the resulting piecewise Legendre point set obviously breaks the regularity of the time variable. (We leave off here global methods which treat Volterra equations as the Fredholm ones: although their convergence may be spectral [30], they require the determination of T in advance and a restart from scratch for any increase of T; moreover, they require solving a dense system of equations, which becomes very costly for large D_1 and/or D_2.)

In contrast, quadrature methods merely consider values at the equispaced t_m. The method we shall present is an extension to DVIDEs of the quadrature method introduced in [6]. The treatment of the right-hand side will thereby be an application of the composite barycentric rational quadrature (CBRQ) rule presented in that same article. Thus the approximation at t_m of y', the left-hand side of (1), remains to be chosen. To stay with the mere values at equispaced t_m, divided differences are the natural way to go. Since only values at t_k, $k \leq m$, are known at time t_m, left one-sided differences should be used. For relatively large time steps h, customary polynomial differences work well only for small numbers of points, for as this number increases, the approximation becomes ill-conditioned in view of the Runge phenomenon affecting the underlying polynomial interpolant [22]; this limits the order of convergence of the polynomial method. We therefore use instead the barycentric rational finite differences (RFD) formula introduced in [22]; they have the advantage of leading to a fully (linear) rational method. We recall that both the quadrature and the differentiation methods introduced in [21], resp. [22], are based on the exact integration, resp. differentiation, of the linear barycentric rational interpolant presented in [14] and further discussed in [7] and [16].

After the present introduction, Sect. 2 briefly reviews the two applications of linear barycentric rational interpolation which lie at the basis of this work, namely the composite barycentric rational quadrature and the barycentric rational finite differences. Section 3 introduces the method for solving DVIDEs based on barycentric rational interpolation and discuss its zero–stability and convergence. Section 4 illustrates the results with numerical experiments. General remarks about the method conclude the paper.

2 A Short Review of the CBRQ Rule and the RFD Formulas

In this section, we recall the construction of the CBRQ rule for the approximation of definite integrals and of the RFD formulas for the approximation of the first derivative of a function at a grid point. Both are based on an exact application of the corresponding operator to a linear barycentric rational interpolant instead of the interpolating polynomial. Linear barycentric rational interpolants have been introduced by the second author in [4]. They consist in replacing the weights λ_j in the barycentric formula

$$p_n(t) = \sum_{j=0}^{n} \frac{\lambda_j}{t-t_j} f(t_j) \Bigg/ \sum_{j=0}^{n} \frac{\lambda_j}{t-t_j}, \qquad \lambda_i = 1 \Bigg/ \prod_{j=0, j\neq i}^{n} (t_i - t_j)$$

for the interpolating polynomial p_n with more appropriate weights β_j, which are still independent of the interpolated function, but are chosen in such a way that bad properties of the polynomial such as ill–conditioning and Runge's phenomenon are avoided, and convergence and well–conditioning are guaranteed.

The two sets of weights presented in [4] are $\beta_j = (-1)^j$ and the same with the first and last ones halved; this requires that the nodes are ordered according to $t_0 < t_1 < \ldots < t_n$. The corresponding two interpolants are excellently conditioned, but the first converges as $O(h)$ and the second as $O(h^2)$, where h is the usual steplength $h := \max\{t_{i+1} - t_i\}$. Another linear rational interpolant, which depends on a parameter d and converges at the rate $O(h^{d+1})$, has been introduced in [14]. Its barycentric weights for equispaced nodes—only the latter are used in the present work—are

$$\beta_j = \frac{(-1)^{j-d}}{2^d} \sum_{i \in J_j} \binom{d}{j-i}, \qquad J_j := \{i \in \{0, \ldots, n-d\} : j-d \leq i \leq j\}. \qquad (3)$$

As it is based on a blend of all interpolating polynomials of $d+1$ consecutive nodes, it may be ill-conditioned and/or unstable for large d and/or arbitrary sets of points [5], but is excellent for equispaced nodes and not too large d. This makes it an interesting choice for solving smooth problems on the basis of equispaced samples [7]. This is the case of the quadrature method for Volterra integral equations introduced in [6].

2.1 The CBRQ Rule

The basic idea here is to replace the function to be approximated, say g, by the linear barycentric rational interpolant and apply the operator to the latter. In the global barycentric rational quadratic rule [21], the integrand is approximated by a single interpolant on the whole interval of integration,

$$r_n(t) = \sum_{j=0}^{n} g(t_j)\ell_j(t), \quad \ell_j(t) := \frac{\beta_j}{t-t_j} \Big/ \sum_{i=0}^{n} \frac{\beta_i}{t-t_i},$$

which for equispaced nodes leads to the rule $\int_{t_0}^{t_n} g(s)\mathrm{d}s \approx h \sum_{k=0}^{n} w_k g(t_k)$, with the quadrature weights $w_k = h^{-1} \cdot \int_{t_0}^{t_n} \ell_k(t)\mathrm{d}t$; the latter are evaluated in machine precision by means of an accurate quadrature rule such as Gauss or Clenshaw–Curtis, both provided in the Chebfun system [31]. In the corresponding quadrature method for the solution of classical VIEs, the number of terms in the rule becomes larger and larger with increasing n. The last two authors have therefore suggested with Klein in [6] to construct instead a *composite* barycentric rational quadrature rule, in which the interval of interpolation is divided into subintervals of length nh and a last interval of smaller length, and to use only the weights corresponding to the said subintervals; we shall use the same rules here.

For integrals such as that in (1.1) over intervals of length $\tau > 0$, we assume that there exists a positive integer q, such that

$$h = \frac{\tau}{q},$$

and consider a uniform grid

$$0 = x_0 < x_1 < \cdots < x_q = \tau,$$

with $x_{j+1} - x_j = h$, $j = 0, 1, \ldots, q-1$. We choose a value of the parameter d, a fixed number n of nodes with $d \leq n \leq q/2$ and we set $p := \lfloor q/n \rfloor - 1$. After an obvious change of variable, we approximate the integral with the composite rule

$$\int_0^{\tau} g(s)\mathrm{d}s \approx h \sum_{j=0}^{p-1} \sum_{k=0}^{n} w_k^{(n)} g(x_{jn+k}) + h \sum_{k=0}^{q-pn} w_k^{(q-pn)} g(x_{pn+k}), \tag{4}$$

where

$$w_k^{(r)} = r \int_0^1 \frac{\frac{\beta_k}{rs-k}}{\sum_{j=0}^{r} \frac{\beta_j}{rs-j}} \mathrm{d}s, \quad r = n, n+1, \ldots, 2n-1.$$

The convergence order of the rule (4) is given in the following proposition.

Proposition 1 [6] *Suppose n, d and q, $d \leq n \leq q/2$, are positive integers, $g \in C^{d+2}[0, \tau]$. Then the absolute error in the approximation of the integral of g with (4) is bounded by Ch^{d+1}, where the constant C depends only on d, on derivatives of g and on the interval length τ.*

When $n - d$ is odd, the bound on the interpolation error given in [14], Thm. 2, involves an additional factor nh, so that the order of the related quadrature rule increases to $d + 2$. Notice that in some of our computations we used n close to q, which implies $p = 0$; then the method is global, not composite [6].

2.2 The RFD Formulas

In the same fashion, linear rational finite differences are obtained by exactly differencing the linear barycentric rational interpolant [22]. In practice, central differences do not perform better than polynomial differences, when the nodes are equispaced. In the case we are concerned with here, however, the derivatives are needed at t_m, the right extremity of the interval of interpolation, so that left–sided differences are necessary. In that case and with a large number of nodes, rational differences are much more accurate than polynomial ones [22].

Let m be a positive integer. For the approximation of the first derivative of a sufficiently smooth function g at the node t_m via $t_{m-n}, \ldots, t_m$, we compute [2,20,22]

$$g'(t_m) \approx \frac{1}{h}\sum_{j=0}^{n} v_j g(t_{m-j}), \tag{5}$$

where

$$v_j = \begin{cases} \dfrac{\beta_{n-j}}{j\beta_n}, & j \neq 0, \\ -\displaystyle\sum_{l=1}^{n} v_l\,, & j = 0. \end{cases}$$

Proposition 2 [20,22] *Suppose m, n and d, $d \le n$, are positive integers, $t_{m-n}, \ldots, t_m$ are equispaced nodes in the interval $[a, b]$ and $g \in C^{d+2}[a, b]$. Then the absolute error of the approximation formula* (5) *is bounded by Ch^d.*

One can prove that

$$\sum_{j=0}^{n} v_j e^{-jz} = z + O(z^{d+1}).$$

Some values of v_j are given in Sect. 3.1 below.

In fact, when $n - d$ is odd, and in view of the increase of the guaranteed order of the interpolation error by one unit mentioned after Proposition 1, the order of the related divided difference increases to $d + 1$.

Further results on the convergence of the derivatives of the Floater–Hormann family of interpolants have just been published [13].

3 Description of the Method for DVIDEs

Consider a uniform grid

$$t_0 - \tau = t_{-q} < t_{-q+1} < \cdots < t_0 < t_1 < \cdots < t_N, \quad t_N \ge T,$$

and let $t_{i+1} - t_i = h$, $i = -q, -q + 1, \ldots, N - 1$, with $h = \tau/q$ as in Sect. 2. Let d, n, and p be as in Sect. 2.1. Applying the RFD formula (5) and the CBRQ rule (4) to the derivative and integral parts of (1) at the point t_m, respectively, yields

$$\sum_{j=0}^{n} v_j y_{m-j} = hf\big(t_m, y_m, y_{m-q}, \widetilde{Q}_m(h)\big), \quad m = 1, 2, \ldots, N, \tag{6}$$

for the unknown y_m, where

$$\widetilde{Q}_m(h) = h \sum_{j=0}^{p-1} \sum_{k=0}^{n} w_k^{(n)} K(t_m, t_{jn+k+m-q}, y_{jn+k+m-q}) + h \sum_{k=0}^{q-pn} w_k^{(q-pn)} K(t_m, t_{pn+k+m-q}, y_{pn+k+m-q}).$$

Here y_m, $m = 1, 2, \ldots, N$, is an approximation to the exact solution y of (1) at the mesh point t_m. According to the initial condition in (1), we may take

$$y_m = \varphi(t_m), \quad m = -q, -q+1, \ldots, 0.$$

If the functions f and/or K are nonlinear in y, the fact that y_m appears on both sides of (6) implies that nonlinear algebraic equations must be solved at each step; our implementation uses Newton's method for this purpose.

In the next two subsections, we shall discuss the zero–stability and the convergence of the method (6).

3.1 Zero–Stability

It is well-known that the zero–stability of a method is a necessary condition for the convergence of that method [23]. The zero–stability of a method is warranted, if the numerical solution y_m of problem (1) with $f(\cdot, \cdot, \cdot, \cdot) \equiv 0$ is bounded. The method (6) applied to such problems gives the difference equation $\sum_{j=0}^{n} v_j y_{m-j} = 0$. We may then define its zero–stability as follows.

Definition 1 The method (6) is said to be zero–stable, if the roots of the polynomial

$$\rho(x) := v_0 x^n + v_1 x^{n-1} + \cdots + v_n$$

lie inside or on the unit circle, those on the circle being simple.

To assist the reader in the use of the method (6) and the study of its zero–stability, the vector $v = [v_0 \quad v_1 \quad \cdots \quad v_n]^T$ is given for some special cases in Table 1. Note that for $d = n - 1$ the Floater–Hormann interpolant coincides with the interpolating polynomial. As a consequence, the coefficients v_i's, $i = 0, 1, \ldots, n$, are those of the cooresponding backward differentiation formula (BDF) [10,23]. For example, in the case of $(n, d) = (4, 3)$ and $(n, d) = (6, 5)$, the v_i's are the coefficients of the BDFs of orders 4 and 6, respectively.

To investigate the zero–stability of the method (6), we use *Bistritz' stability criterion* [8], which determines how many zeros of $\rho(x)/(1-x)$ lie strictly inside the unit disk. For example, the Bistritz stability sequences of the polynomial $\rho(x)/(1-x)$ corresponding to the pairs (n, d) of Table 1 are given in Table 2. The number of sign changes, nsc, in these sequences gives the number of zeros of $\rho(x)/(1-x)$ that lie outside of the closed unit disk. From this table, we find that the method with the values of (n, d) given in Table 1 is zero–stable except for $(n, d) = (7, 6)$ and $(11, 6)$. Continuing in this way, one can show that the method (6) is zero–stable for every choice of $n \geq d$ with $n \leq 20$ and $d \leq 6$, except for $(n, d) = (7, 6)$, $(8, 6)$, $(11, 6)$, and $(12, 6)$.

Table 1 Weights for left-sided RFD formula (5) for various pairs (n, d)

(n,d)	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
$(4,3)$	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$							
$(6,4)$	$\frac{9}{4}$	-5	$\frac{11}{2}$	$-\frac{14}{3}$	$\frac{11}{4}$	-1	$\frac{1}{6}$					
$(6,5)$	$\frac{49}{20}$	-6	$\frac{15}{2}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$					
$(7,5)$	$\frac{1019}{420}$	-6	8	$-\frac{25}{3}$	$\frac{25}{4}$	$-\frac{16}{5}$	1	$-\frac{1}{7}$				
$(7,6)$	$\frac{363}{140}$	-7	$\frac{21}{2}$	$-\frac{35}{3}$	$\frac{35}{4}$	$-\frac{21}{5}$	$\frac{7}{6}$	$-\frac{1}{7}$				
$(8,5)$	$\frac{681}{280}$	-6	8	$-\frac{26}{3}$	$\frac{15}{2}$	$-\frac{26}{5}$	$\frac{8}{3}$	$-\frac{6}{7}$	$\frac{1}{8}$			
$(9,6)$	$\frac{49}{19}$	-7	11	-14	14	$-\frac{56}{5}$	7	$-\frac{22}{7}$	$\frac{7}{8}$	$-\frac{1}{9}$		
$(11,6)$	$\frac{165}{64}$	-7	11	-14	$\frac{57}{4}$	$-\frac{63}{5}$	$\frac{21}{2}$	$-\frac{57}{7}$	$\frac{21}{4}$	$-\frac{22}{9}$	$\frac{7}{10}$	$-\frac{1}{11}$

Table 2 Bistritz stability sequences of the polynomial $\rho(x)/(1-x)$ for various pairs (n, d)

(n,d)	Bistritz stability sequences											nsc
$(4,3)$	-2	-4	$-\frac{30}{7}$	$-\frac{16}{3}$								0
$(6,4)$	-2	-6	$-\frac{242}{29}$	$-\frac{1255}{168}$	$-\frac{178}{117}$	$-\frac{32}{3}$						0
$(6,5)$	-2	-6	$-\frac{1330}{157}$	$-\frac{1181}{141}$	$-\frac{86}{105}$	$-\frac{208}{15}$						0
$(7,5)$	-2	-7	$-\frac{1884}{137}$	$-\frac{859}{325}$	$-\frac{1824}{65}$	$-\frac{22}{81}$	$-\frac{3712}{105}$					0
$(7,6)$	-2	-7	$-\frac{668}{49}$	$-\frac{259}{109}$	$-\frac{1887}{44}$	$\frac{15}{233}$	$-\frac{4832}{105}$					2
$(8,5)$	-2	-8	$-\frac{2226}{179}$	$-\frac{893}{91}$	$-\frac{262}{73}$	$-\frac{2413}{156}$	$-\frac{103}{157}$	$-\frac{2176}{105}$				0
$(9,6)$	-2	-9	$-\frac{511}{29}$	$-\frac{249}{59}$	$-\frac{9173}{257}$	$-\frac{89}{114}$	$-\frac{4723}{66}$	$-\frac{4}{45}$	$-\frac{5389}{76}$			0
$(11,6)$	-2	-11	$-\frac{1599}{74}$	$-\frac{349}{57}$	$-\frac{864}{25}$	$-\frac{189}{71}$	$-\frac{2507}{79}$	$-\frac{97}{93}$	$-\frac{11408}{131}$	$\frac{1}{88}$	$-\frac{10981}{124}$	2

3.2 Convergence Analysis

Let $y \in C^{d+2}[t_0 - \tau, T]$, $K \in C^{d+2}(\Omega)$ where $\Omega := S \times \mathbb{R}^{D_1}$ with $S = \{(t, s) : t_0 \leq t \leq T,\ t - \tau \leq s \leq t\}$, f be continuous on $\mathscr{R} := [t_0, T] \times \mathbb{R}^{D_1} \times \mathbb{R}^{D_1} \times \mathbb{R}^{D_2}$, and

$$\forall\ (t, y_1, u, z), (t, y_2, u, z) \in \mathscr{R} : \quad \left\| f(t, y_1, u, z) - f(t, y_2, u, z) \right\| \leq L_1 \left\| y_1 - y_2 \right\|, \tag{7a}$$

$$\forall\ (t, y, u_1, z), (t, y, u_2, z) \in \mathscr{R} : \quad \left\| f(t, y, u_1, z) - f(t, y, u_2, z) \right\| \leq L_2 \left\| u_1 - u_2 \right\|, \tag{7b}$$

$$\forall\ (t, y, u, z_1), (t, y, u, z_2) \in \mathscr{R} : \quad \left\| f(t, y, u, z_1) - f(t, y, u, z_2) \right\| \leq L_3 \left\| z_1 - z_2 \right\|, \tag{7c}$$

$$\forall\ (t, s, y_1), (t, s, y_2) \in \Omega : \quad \left\| K(t, s, y_1) - K(t, s, y_2) \right\| \leq L_4 \left\| y_1 - y_2 \right\|, \tag{7d}$$

where $\| \cdot \|$ is any norm in the corresponding spaces.

Definition 2 The local truncation error of the method (6) at the point t_m is defined by

$$\mathscr{L}[y(t_m); h] := \sum_{j=0}^{n} v_j y(t_{m-j}) - hf\big(t_m, y(t_m), y(t_{m-q}), Q_m(h)\big), \quad m = 1, 2, \ldots, N, \tag{8}$$

where

$$Q_m(h) = h\sum_{j=0}^{p-1}\sum_{k=0}^{n} w_k^{(n)} K(t_m, t_{jn+k+m-q}, y(t_{jn+k+m-q})) \\ + h\sum_{k=0}^{q-pn} w_k^{(q-pn)} K(t_m, t_{pn+k+m-q}, y(t_{pn+k+m-q})).$$

In view of proving the convergence of the proposed method, we state some lemmas.

Lemma 1 *Let $y \in C^{d+2}[t_0 - \tau, T]$, $K \in C^{d+2}(\Omega)$, f be continuous on $\mathscr{R}$ and let the condition (7c) be satisfied. Then*

$$\mathscr{L}[y(t_m); h] = O(h^{d+1}), \quad m = 1, 2, \ldots, N.$$

Proof Equation (1) at the mesh point t_m is

$$y'(t_m) - f\Big(t_m, y(t_m), y(t_{m-q}), \int_{t_m-\tau}^{t_m} K(t_m, s, y(s))\,\mathrm{d}s\Big) = 0, \quad m = 1, 2, \ldots, N. \tag{9}$$

Subtracting (9) from (8) yields

$$\frac{1}{h}\mathscr{L}[y(t_m); h] = \frac{1}{h}\sum_{j=0}^{n} v_j y(t_{m-j}) - y'(t_m) \\ + f\Big(t_m, y(t_m), y(t_{m-q}), \int_{t_m-\tau}^{t_m} K(t_m, s, y(s))\,\mathrm{d}s\Big) \\ - f\big(t_m, y(t_m), y(t_{m-q}), Q_m(h)\big).$$

Using Proposition 2 and (7c), we have

$$\frac{1}{h}\mathscr{L}[y(t_m); h] \leq C_1 h^d + L_3 \left\| \int_{t_m-\tau}^{t_m} K(t_m, s, y(s))\,\mathrm{d}s - Q_m(h)\right\|,$$

and with Proposition 1,

$$\frac{1}{h}\mathscr{L}[y(t_m); h] \leq C_1 h^d + L_3 C_2 h^{d+1},$$

where C_1 and C_2 are constants. So,

$$\mathscr{L}[y(t_m); h] = O(h^{d+1}). \qquad \square$$

Lemma 2 *Let n and d be such that (6) is zero–stable. If the sequences $\{e_k\}$, $\{\tilde{e}_k\}$ and $\{\ell_k\}$ satisfy the difference equation*

$$\sum_{j=0}^{n} v_j e_{m-j} = h\tilde{e}_m + \ell_m, \quad m = 0, 1, \ldots,$$

where $e_{-n} = \cdots = e_{-1} = 0$, then

$$e_m = h\sum_{j=0}^{m} a_{m-j}\tilde{e}_j + \sum_{j=0}^{m} a_{m-j}\ell_j,$$

where $\sum_{j=0}^{\infty} a_j x^j$ *is the power series for* $(v_n + v_{n-1}x + \ldots + v_0 x^n)^{-1}$ *and* $|a_j| \leq C < \infty$ *for all* j.

Proof The proof is similar to that of Lemma 2.4 in [1]. □

Lemma 3 (Discrete Gronwall inequality) *Suppose the nonnegative sequences* $\{r_m\}$ *and* $\{\delta_m\}$ *satisfy the difference inequality*

$$r_m \leq h\Lambda \sum_{j=0}^{m-1} r_j + \delta_m, \quad m = 1, 2, \ldots$$

where $\delta_m \leq \Delta$. *Then,*

$$r_m \leq \Delta e^{\Lambda N h}, \quad m = 1, 2, \ldots, N.$$

We are now in position to state our main theorem about the convergence of the method (6).

Theorem 1 *Let* $y \in C^{d+2}[t_0 - \tau, T]$, $K \in C^{d+2}(\Omega)$, f *be continuous on* $\mathscr{R}$ *and the conditions* (7) *hold. Also, let* n *and* d *be such that* (6) *is zero–stable. Then, the method* (6) *is convergent of order* d:

$$\|y_m - y(t_m)\| \leq Ch^d.$$

Proof Define $e_m = y(t_m) - y_m$. From (6) and (8), we have

$$\sum_{j=0}^{n} v_j e_{m-j} = h\tilde{e}_m + \mathscr{L}[y(t_m); h], \quad m = 1, 2, \ldots \tag{10}$$

where

$$\tilde{e}_m = f\big(t_m, y(t_m), y(t_{m-q}), Q_m(h)\big) - f\big(t_m, y_m, y_{m-q}, \widetilde{Q}_m(h)\big), \quad m = 1, 2, \ldots$$

and $e_{-q} = \cdots = e_0 = 0$. Applying Lemma 2 to (10) yields

$$e_m = h \sum_{j=0}^{m} a_{m-j} \tilde{e}_j + S_m, \tag{11}$$

where

$$S_m = \sum_{j=0}^{m} a_{m-j} \mathscr{L}[y(t_j); h] = \sum_{j=1}^{m} a_{m-j} \mathscr{L}[y(t_j); h].$$

By Lemma 1, we have $\|S_m\| = mC \cdot O(h^{d+1}) = mhC \cdot O(h^d)$, where C is the constant appearing in Lemma 2. So,

$$\sup_{1 \leq m \leq N} \|S_m\| = O(h^d), \quad \text{as} \quad h \to 0, \tag{12}$$

with $Nh = T - t_0$ remaining fixed.

Now, using (7), we have

$$\begin{aligned} \|\tilde{e}_j\| \leq & \left\| f\big(t_j, y(t_j), y(t_{j-q}), Q_j(h)\big) - f\big(t_j, y_j, y(t_{j-q}), Q_j(h)\big) \right\| \\ & + \left\| f\big(t_j, y_j, y(t_{j-q}), Q_j(h)\big) - f\big(t_j, y_j, y_{j-q}, Q_j(h)\big) \right\| \end{aligned}$$

$$
\begin{aligned}
&+ \left\| f\big(t_j, y_j, y_{j-q}, Q_j(h)\big) - f\big(t_j, y_j, y_{j-q}, \widetilde{Q}_j(h)\big)\right\| \\
&\leq L_1\|e_j\| + L_2\|e_{j-q}\| + hL_5 \sum_{i=0}^{q} \|e_{j-q+i}\|, \qquad (13)
\end{aligned}
$$

with $L_5 = 2L_3L_4 \max\left\{|w_k^{(n)}|, |w_k^{(q-pn)}|\right\}$. Substituting (13) in (11) yields

$$\|e_m\| \leq hC \sum_{j=0}^{m} \left\{L_1\|e_j\| + L_2\|e_{j-q}\| + hL_5 \sum_{i=0}^{q} \|e_{j-q+i}\|\right\} + \|S_m\|.$$

Define $E_j = \max\{\|e_l\|,\ l = j-q, \ldots, j\}$. So,

$$\|e_m\| \leq hC \sum_{j=0}^{m} (L_1 + L_2 + (q+1)hL_5)E_j + \|S_m\|.$$

This implies

$$E_m \leq hL \sum_{j=0}^{m} E_j + \|S_m\|,$$

where

$$L = C(L_1 + L_2 + 2\tau L_5) \geq C(L_1 + L_2 + (q+1)hL_5).$$

Thus, for $hL < 1$,

$$E_m \leq \frac{hL}{1-hL} \sum_{j=0}^{m-1} E_j + \frac{\|S_m\|}{1-hL}.$$

Using Lemma 3 yields

$$E_m \leq \left(\frac{1}{1-hL} \sup_m \|S_m\|\right) e^{\frac{L}{1-hL}Nh}.$$

From this and (12), we conclude that $E_m = O(h^d)$, and so $\|e_m\| = O(h^d)$. □

According to the remarks following Propositions 1 and 2, the error may be expected to be of order $d+1$, when $n-d$ is odd.

To construct an infinitely smooth approximate solution on the whole interval $[t_0, T]$, it is natural and elegant to make use of the same barycentric rational interpolant as the one which lies at the basis of our scheme for determining the discrete approximations $y_0, \ldots, y_N$, i.e.,

$$r_N(t) = \sum_{j=0}^{N} \frac{\beta_j}{t-t_j} y_j \Bigg/ \sum_{j=0}^{N} \frac{\beta_j}{t-t_j}, \qquad (14)$$

with the weights β_j from (3). Under the hypotheses of Theorem 1 and with the choice d of the parameter, by Theorem 2 in [14] and Theorem 1 the maximum absolute error in r_N may be bounded as

$$
\begin{aligned}
\max_{t_0\le t\le T}\|r_N-y\|_\infty &\le \max_{t_0\le t\le T}\left\|\frac{\sum_{j=0}^{N}\frac{\beta_j}{t-t_j}y(t_j)}{\sum_{j=0}^{N}\frac{\beta_j}{t-t_j}}-y(t)\right\|_\infty \\
&\quad+\max_{t_0\le t\le T}\left\|\frac{\sum_{j=0}^{N}\frac{\beta_j}{t-t_j}\left(y_j-y(t_j)\right)}{\sum_{j=0}^{N}\frac{\beta_j}{t-t_j}}\right\|_\infty \\
&\le Ch^{d+1}+\Lambda_N\max_{0\le j\le N}\left\|y(t_j)-y_j\right\|_\infty,
\end{aligned}
$$

where C is a generic constant that does not depend on N, and Λ_N is the Lebesgue constant associated with the linear barycentric rational interpolant r_N:

$$\Lambda_N=\max_{t_0\le t\le T}\sum_{j=0}^{N}\left|\frac{\beta_j}{t-t_j}\right|\Bigg/\left|\sum_{j=0}^{N}\frac{\beta_j}{t-t_j}\right|.$$

Since Λ_N is known to be bounded by $C\log(N)$ for equispaced nodes [9], one has

$$\max_{t_0\le t\le T}\|r_N-y\|_\infty\le C\log(N)h^d.$$

The derivatives of any order of y may be approximated with the corresponding derivatives of (14) by means of the simple formula by Schneider and Werner [28].

4 Numerical Verifications

In this section, we apply the method (6) with various choices of n, d, and q to a number of linear and nonlinear DVIDEs to demonstrate the efficiency and accuracy of the scheme. The numerical results below confirm the theoretical convergence estimates derived in Sect. 3.2.

Following a suggestion by a referee, we also present results of numerical experiments with the Runge–Kutta–Gauss method and Pouzet quadrature formula (RKGP) [35], in the form

$$
\begin{aligned}
Y_i^{[m]}&=y_m+h\sum_{j=1}^{s}a_{ij}f\left(t_j^{[m]},Y_j^{[m]},Y_j^{[m-q]},z_j^{[m]}\right),\qquad i=1,2,\ldots,s\\
y_{m+1}&=y_m+h\sum_{j=1}^{s}b_jf\left(t_j^{[m]},Y_j^{[m]},Y_j^{[m-q]},z_j^{[m]}\right),\qquad m\ge 0.
\end{aligned}
\tag{15}
$$

Here $h=\tau/q$ is the stepsize with q as a given positive integer, $t_j^{[m]}:=t_m+c_jh$ with $t_m=t_0+mh$ are the stage points, and $Y_j^{[m]}$,

$$
\begin{aligned}
z_j^{[m]}:=h\sum_{r=1}^{s}a_{jr}K\left(t_j^{[m]},t_r^{[m]},Y_r^{[m]}\right)+h\sum_{k=1}^{q}\sum_{r=1}^{s}b_rK\left(t_j^{[m]},t_r^{[m-k]},Y_r^{[m-k]}\right)\\
-h\sum_{r=1}^{s}a_{jr}K\left(t_j^{[m]},t_r^{[m-q]},Y_r^{[m-q]}\right)
\end{aligned}
$$

and y_m are approximations to

$$y\left(t_j^{[m]}\right),\quad z\left(t_j^{[m]}\right):=\int_{t_j^{[m-q]}}^{t_j^{[m]}}K\left(t_j^{[m]},s,y(s)\right)\mathrm{d}s,\quad\text{and}\quad y(t_m),$$

Table 3 Numerical results for example (16) with (n, d) equal to (4, 3) and (6, 4)

q	$(n,d)=(4,3)$					$(n,d)=(6,4)$				
	e_{max}	e_N	O_N	e_I	O_I	e_{max}	e_N	O_N	e_I	O_I
8	6.62e−05	1.49e−06		5.43e−04		6.15e−05	1.97e−06		7.81e−05	
16	4.25e−06	7.54e−08	4.30	3.64e−05	3.90	4.16e−06	8.44e−08	4.54	4.16e−06	4.23
32	2.68e−07	4.31e−09	4.13	2.31e−06	3.98	2.66e−07	4.51e−09	4.23	2.66e−07	3.97
64	1.68e−08	2.58e−10	4.06	1.45e−07	3.99	1.68e−08	2.64e−10	4.09	1.68e−08	3.98
128	1.05e−09	1.58e−11	4.03	9.09e−09	4.00	1.05e−09	1.60e−11	4.04	1.05e−09	4.00
256	6.59e−11	9.75e−13	4.02	5.68e−10	4.00	6.59e−11	9.79e−13	4.03	6.59e−11	3.99

respectively. The method (15) is characterized by the abscissae c_j, the coefficients a_{ij} and the weights b_j, which are given for Gauss methods of various orders in [10, p. 232]. The stability of such methods has been studied in [18].

Let us denote by e_m the absolute error of the approximation y_m, $m = 0, 1, \ldots, N$, and by $e_{\max}$ the maximum of these errors, i.e.,

$$e_m := \|y(t_m) - y_m\|_\infty, \quad m = 0, 1, \ldots, N,$$
$$e_{\max} := \max_{1 \le m \le N} e_m.$$

Also, we denote by e_I the maximal absolute error of the interpolation (14) of the approximations y_m, $m = 0, 1, \ldots, N$, i.e.,

$$e_I := \max_{t_0 \le t \le T} \|r_N(t) - y(t)\|_\infty.$$

To estimate e_I, the maximum was taken over the values $|r_N(t) - y(t)|$ at about 3000 equispaced t in the interval $[t_0, T]$, with $r_N(t)$ evaluated by formula (14). Finally, we denote the experimental orders in the approximation of $y(t_N)$ by O_N and in the interpolation by O_I.

As a first example, we considered the DVIDE with partially variable coefficients [34]

$$y'(t) = -(6 + \sin t)y(t) + y\left(t - \frac{\pi}{4}\right) - \int_{t-\frac{\pi}{4}}^{t} \sin(s)y(s)\mathrm{d}s + 5\exp(\cos t), \quad t \ge 0, \quad (16)$$

with an initial condition on the interval $\left[-\frac{\pi}{4}, 0\right]$ such that the exact solution is given by $y(t) = \exp(\cos t)$. Here we chose $T = 9\pi$.

Table 3 shows the numerical results for (n, d) equal to (4, 3) and (6, 4). In both cases, as to be expected from the comments following Propositions 1 and 2, the observed order was four. The interpolation errors decreased with order four as well. The accuracy of the obtained numerical solutions over the whole interval $[t_0, T]$ is displayed in Fig. 1. Also, the number of kernel evaluations versus the error with both the method (6) and RKGP are plotted in Fig. 2, which illustrates that the method (6) is competitive with RKGP of the same order. Let us recall that the barycentric rational method computes the solution only at equispaced abscissae and yields, through formula (14), a global infinitely smooth approximation to y (see the e_I–column of our tables), this in contrast to Runge–Kutta methods, which require quite some thoughts and computation for a dense output [32].

Next we studied the equation

$$y'(t) = y^3(t) + y(t-1) - \int_{t-1}^{t} \left(10 + 2(t-s)^2\right)y^2(s)\mathrm{d}s + f(t), \quad t \in [-1, 1], \quad (17)$$

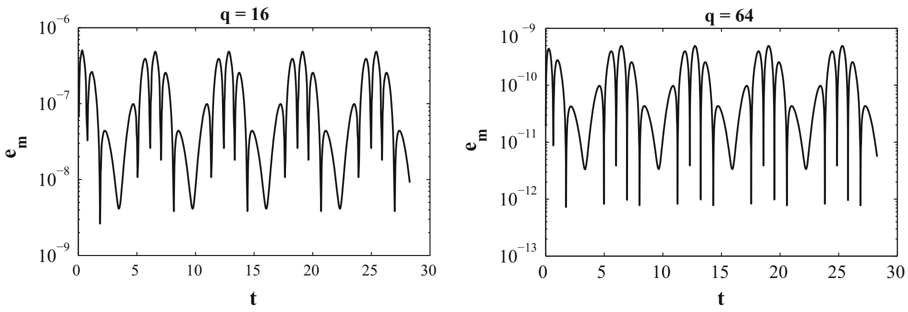

Fig. 1 Error for example (16) with $(n, d) = (7, 5)$ and $q = 16, 64$ over the interval $[0, 9\pi]$

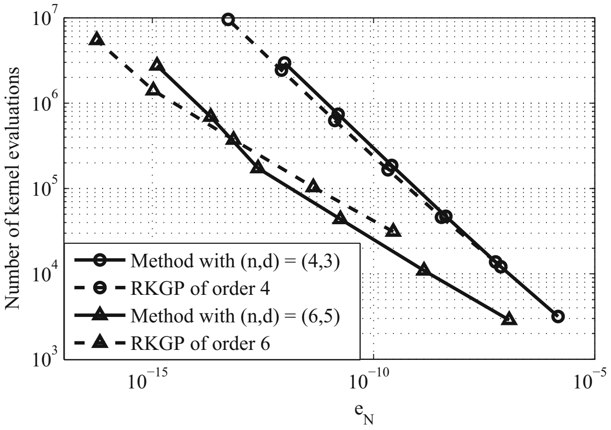

Fig. 2 Comparison of the number of kernel evaluations with the methods of order 4 and order 6 applied to example (16)

Table 4 Numerical results for example (17) with (n, d) equal to (8, 5) and (13, 6)

q	$(n, d) = (8, 5)$					$(n, d) = (13, 6)$				
	e_{max}	e_N	O_N	e_I	O_I	e_{max}	e_N	O_N	e_I	O_I
16	1.62e−02	7.67e−03		1.63e−02		1.84e−02	6.93e−03		2.05e−02	
32	1.26e−03	1.38e−04	5.80	1.36e−03	3.58	6.13e−04	3.28e−05	7.72	6.85e−04	4.90
64	3.77e−05	3.61e−06	5.26	3.85e−05	5.14	1.57e−05	2.75e−06	3.58	1.61e−05	5.41
128	7.12e−07	1.21e−07	4.90	7.14e−07	5.75	2.51e−07	3.16e−08	6.44	2.51e−07	6.00
256	1.17e−08	2.23e−09	5.76	1.17e−08	5.93	2.37e−09	1.66e−10	7.57	2.37e−09	6.73
512	1.85e−10	3.63e−11	5.94	1.85e−10	5.98	1.93e−11	3.98e−13	8.70	1.93e−11	6.94

whose exact solution is Runge's function $y(t) = 1/(1 + 25t^2)$. The initial condition on $[-2, -1]$ was the value of that exact solution.

Table 4 shows the numerical results for (n, d) equal to (8, 5) and (13, 6). There is no Runge phenomenon, as the latter does not occur with this kind of rational interpolants and fixed d [14]. As to be expected from the comment following Propositions 1 and 2, the observed order was six and seven, respectively. The corresponding interpolation errors decreased with

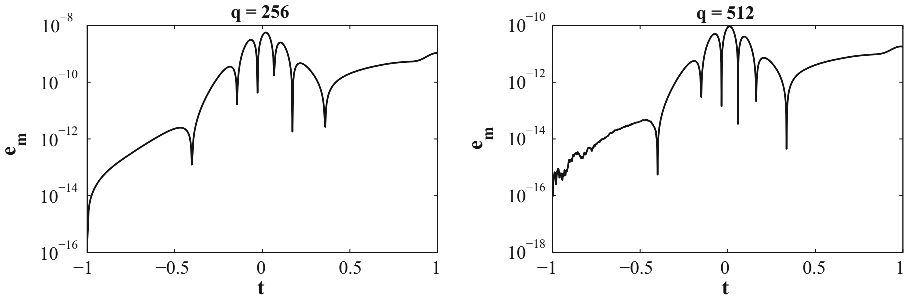

Fig. 3 Error for example (17) with $(n, d) = (10, 6)$ and $q = 256, 512$ over the interval $[-1, 1]$

Table 5 Numerical results for example (18) with (n, d) equal to (8, 5) and (10, 6)

q	$(n, d) = (8, 5)$					$(n, d) = (10, 6)$				
	e_{max}	e_N	O_N	e_I	O_I	e_{max}	e_N	O_N	e_I	O_I
10	4.55e−09	4.23e−09		4.55e−08		2.29e−09	2.04e−09		3.19e−09	
20	7.02e−11	6.98e−11	5.92	6.77e−10	6.07	3.47e−11	3.44e−11	5.89	5.08e−11	5.97
40	1.12e−12	1.12e−12	5.96	1.03e−11	6.04	5.54e−13	5.49e−13	5.97	8.12e−13	5.97
80	6.96e−14	5.87e−14	4.25	3.60e−13	4.84	3.53e−14	1.69e−14	5.02	8.28e−13	−0.03

the same orders. The accuracy of the obtained numerical solutions over the whole interval $[-1, 1]$ is displayed in Fig. 3.

We also studied the nonlinear two-dimensional system [34]

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = -4\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} 0 & \sin t \\ \cos t & 0 \end{pmatrix} \cdot \begin{pmatrix} y_1(t-\frac{\pi}{5}) \\ y_2(t-\frac{\pi}{5}) \end{pmatrix}$$
$$+ \frac{1}{\sqrt{2}} \int_{t-\frac{\pi}{5}}^{t} \begin{pmatrix} \frac{(1+\sin^2 s)y_1^2(s)}{1+y_1^2(s)} \\ \frac{(1+\cos^2 s)y_2^2(s)}{1+y_2^2(s)} \end{pmatrix} \mathrm{d}s + \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix}, \quad t \geq 0, \tag{18}$$

where $f(t) = (f_1(t), f_2(t))^T$ and where the initial condition on $[-\frac{\pi}{5}, 0]$ is chosen in such a way that the solution equals $y(t) = (\sin t, \cos t)^T$. We considered $T = 9\pi$ again.

Table 5 shows the numerical results for (n, d) equal to (8, 5) and (10, 6). The errors decreased with N until about machine precision. As to be expected from the comment following Propositions 1 and 2, the observed order is six in both cases. The interpolation errors also decreased with order six. They reached machine precision with $N = 80$, which explains some smaller experimental orders in the last row of the table. Moreover, in Fig. 4, we have plotted $\log(e_N)$ versus $\log(h)$, together with a line of slope d, if $n - d$ is even, and $d + 1$, if $n - d$ is odd, for two other choices of (n, d). The accuracy of the obtained numerical solutions over the whole interval $[t_0, T]$ is displayed in Fig. 5. Also, the number of kernel evaluations versus the error with the method (6) and RKGP are plotted in Fig. 6, which illustrates that the method (6) is competitive with RKGP of the same order.

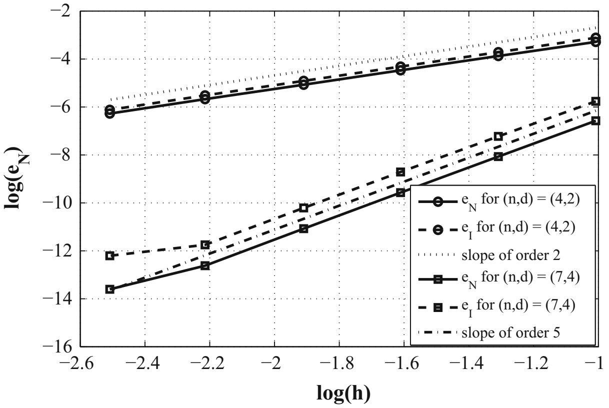

Fig. 4 Numerical results for example (18) with (n, d) equal to (4, 2) and (7, 4)

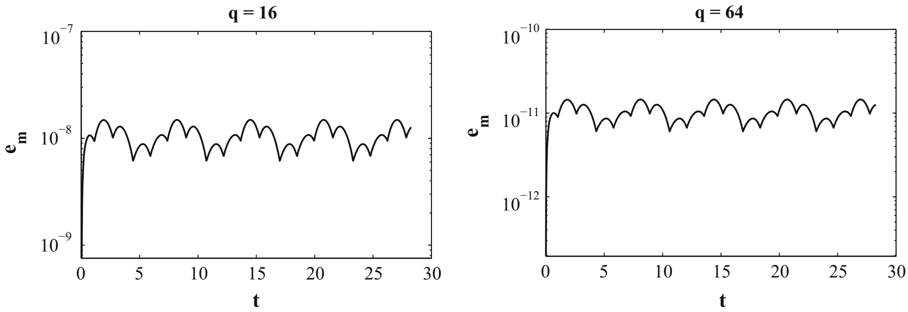

Fig. 5 Error for example (18) with $(n, d) = (9, 4)$ and $q = 16, 64$ over the interval $[0, 9\pi]$

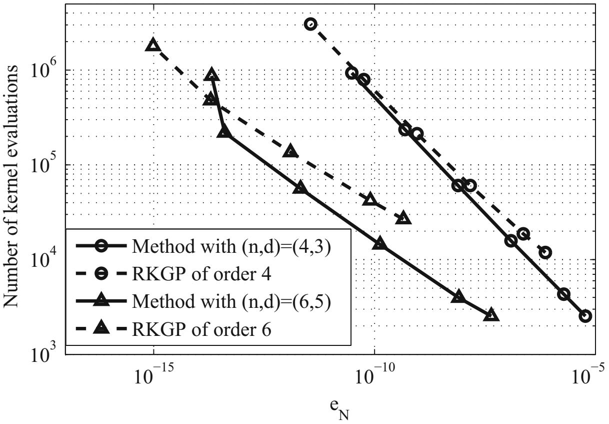

Fig. 6 Comparison of the number of kernel evaluations with the methods of order 4 and order 6 applied to example (18)

To demonstrate the capability and efficiency of the proposed method in solving stiff problems, we studied the singularly perturbed nonlinear system of DVIDEs [33]

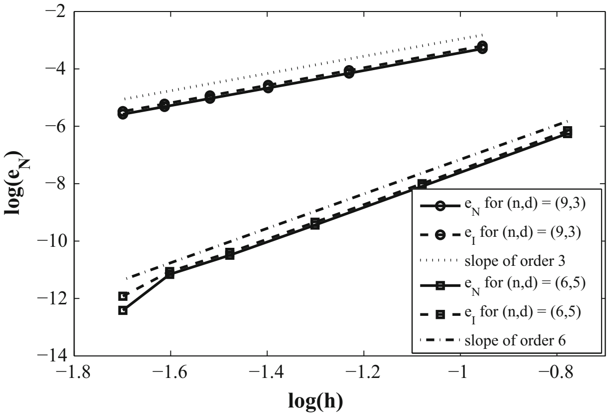

Fig. 7 Numerical results for example (19) with (n, d) equal to (9, 3) and (6, 5)

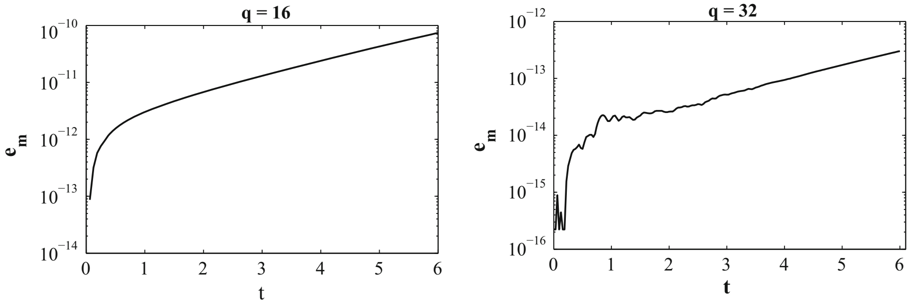

Fig. 8 Error for example (19) with $(n, d) = (9, 6)$ and $q = 16, 32$ over the interval [0, 6]

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1(t) \\ \varepsilon y_2(t) \end{pmatrix} = \begin{pmatrix} \alpha y_1(t) + 2y_2^2(t) \\ -(1 + y_1(t))y_2(t) \end{pmatrix} + \int_{t-1}^{t} \begin{pmatrix} y_1(s)y_2(s) \\ y_1(s) - y_2(s) \end{pmatrix} \mathrm{d}s + \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix}, \quad t > 0, \tag{19}$$

where α and ε are fixed parameters. For this problem, $f_1(t)$ and $f_2(t)$ and the initial condition on $[-1, 0]$ are chosen in such a way that the exact solution equals $y(t) = \big(\exp(-0.5t) + \exp(-0.2t), -\exp(-0.5t) + \exp(-0.2t)\big)^T$.

The authors of [33] have chosen $\alpha = -1000$ and $\varepsilon = 10^{-6}$. However, with these values of α and ε, the problem does not seem very stiff: we achieved machine precision already with small values of q and d. For that reason we have taken $\alpha = -10^{-5}$ and $\varepsilon = 10^{-10}$. Here we considered $T = 6$.

In Fig. 7, we have again plotted $\log(e_N)$ versus $\log(h)$ together with a line of slope d, if $n - d$ is even, or $d + 1$, if $n - d$ is odd, for (n, d) equal to (9, 3) and (6, 5). The accuracy of the solutions over the whole interval $[t_0, T]$ is displayed in Fig. 8. Moreover, in Table 6, we present the numerical results of our method with $(n, d) = (4, 3)$ and RKGP of order four and stage order two. They show that the accuracy of our method with the *stiff* problem (19) is higher than that of RKGP and the same order. Also, this table indicates that RKGP suffers from *order reduction* [15, p. 225-227], which is not the case of method (6).

Table 6 Numerical results for example (19) with $(n, d) = (4, 3)$ and RKGP of order four and stage order two

q		8	16	32	64	128	256
Method (6) with	e_N	3.23e−05	1.97e−06	1.22e−07	7.56e−09	4.70e−10	2.68e−11
$(n, d) = (4, 3)$	O_N		4.04	4.01	4.01	4.01	4.13
RKGP of	e_N	2.04e−04	5.09e−05	1.32e−05	3.17e−06	7.65e−07	1.58e−07
order 4	O_N		2.00	1.95	2.06	2.05	2.28

Table 7 Numerical results for example (20) with (n, d) equal to $(3, 2)$ and $(6, 4)$

q		30	60	120	240	480	960
Method (6) with	e_N	1.24e−01	1.67e−02	2.17e−03	2.72e−04	3.09e−05	1.10e−06
$(n, d) = (3, 2)$	O_N		2.89	2.94	3.00	3.14	4.81
Method (6) with	e_N	1.38e−02	9.19e−04	5.88e−05	3.70e−06	2.30e−07	1.38e−08
$(n, d) = (6, 4)$	O_N		3.91	3.97	3.99	4.01	4.06

Finally, following a suggestion by the other referee, we considered a realistic example in the form of a mathematical model of the human immune response with delay [3,25]

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t} X_U(t) = s_U - \alpha_1 X_U(t) B(t) - \mu_{X_U} X_U(t), \\ \frac{\mathrm{d}}{\mathrm{d}t} X_I(t) = \alpha_1 X_U(t) B(t) - \alpha_2 X_I(t) A_R(t) - \mu_{X_I} X_I(t), \\ \frac{\mathrm{d}}{\mathrm{d}t} B(t) = \alpha_{20} B(t) \Big(1 - \frac{B(t)}{\sigma}\Big) - \alpha_3 B(t) I_R(t) - \alpha_4 B(t) A_R(t), \\ \frac{\mathrm{d}}{\mathrm{d}t} I_R(t) = s_{I_R} + \int_{t-\tau}^{t} w(s-t) B(s) \mathrm{d}s - \mu_{I_R} I_R(t), \\ \frac{\mathrm{d}}{\mathrm{d}t} A_R(t) = s_{A_R} + \int_{t-\tau}^{t} w(s-t) B(s) \mathrm{d}s - \mu_{A_R} A_R(t), \end{cases} \tag{20}$$

in which the components X_U, X_I, and B stand for uninfected target cells, infected cells, and bacteria, and the variables I_R and A_R denote innate and adaptive responses, respectively. Here we considered $t \in [0, 50]$ with $\tau = 6$ and $w(t) = \frac{\ln 2}{6} \exp(\frac{\ln 2}{6} t)$. For a detailed description of the model and the selection of parameter values, as well as initial conditions, see [3,25].

We compared our results with the reference solution obtained by solving the equivalent system of DDEs using the `dde23` command from the MATLAB DDE suite [29] with the tolerance 10^{-12}. Table 7 shows the numerical results for (n, d) equal to $(3, 2)$ and $(6, 4)$. We see from this table that the order of convergence does appear to be three and four, in line with the orders expected from the theory.

5 Conclusion

Linear barycentric rational interpolation provides a simple yet very efficient way of approximating a function by an infinitely smooth one from a sample at equispaced nodes. In this paper, we have presented an exemplary application, namely the numerical solution of a class of DVIDEs with constant delay. We have analyzed the convergence behavior of the corresponding method and discussed some numerical experiments, which demonstrate its efficiency and beautifully agree with the theoretical results. In fact, the accuracy was even better than that attained in [6] with the original method for classical Volterra integral equations. This likely is a consequence of the fact that the right–hand side approximates y', the derivative of the solution, so that the mathematical problem involves yet another integration.

The method is very elegant in the sense that it is fully based on linear barycentric rational interpolation with Floater–Hormann weights. The later are the best known today for equispaced nodes [5]. One may program the method with arbitrary weights as variable input: should in the future weights for equispaced nodes be discovered which are even better than (3), the user would then just have to introduce them into the existing program.

Acknowledgements The authors would like to thank Georges Klein for his careful reading of a draft of the manuscript, as well as the unkown referees for comments that have improved this work.

References

1. Baker, C.T.H., Ford, N.J.: Convergence of linear multistep methods for a class of delay-integro-differential equations. Int. Ser. Numer. Math. **86**, 47–59 (1988)
2. Baltensperger, R., Berrut, J.-P., Noël, B.: Exponential convergence of a linear rational interpolant between transformed Chebyshev points. Math. Comput. **68**, 1109–1120 (1999)
3. Beretta, E., Carletti, C., Kirschner, D.E., Marino, S.: Stability analysis of a mathematical model of the immune response with delays. In: Takeuchi, Y., Iwasa, Y., Sato, K. (eds.) Mathematics for Life Science and Medicine, pp. 177–206. Springer, Berlin (2007)
4. Berrut, J.-P.: Rational functions for guaranteed and experimentally well-conditioned global interpolation. Comput. Math. Appl. **15**, 1–16 (1988)
5. Berrut, J.-P.: Linear barycentric rational interpolation with guaranteed degree of exactness. In: Fasshauer G.E., Schumaker L.L. (eds.) Approximation Theory XV: San Antonio 2016, Springer Proceedings in Mathematics & Statistics, pp. 1–20 (2017)
6. Berrut, J.-P., Hosseini, S.A., Klein, G.: The linear barycentric rational quadrature method for Volterra integral equations. SIAM J. Sci. Comput. **36**, A105–A123 (2014)
7. Berrut, J.-P., Klein, G.: Recent advances in linear barycentric rational interpolation. J. Comput. Appl. Math. **259**, 95–107 (2014)
8. Bistritz, Y.: A circular stability test for general polynomials. Syst. Control Lett. **7**, 89–97 (1986)
9. Bos, L., De Marchi, S., Hormann, K., Klein, G.: On the Lebesgue constant of barycentric rational interpolation at equidistant nodes. Numer. Math. **121**, 461–471 (2012)
10. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 3rd edn. Wiley, New York (2016)
11. Brunner, H., Zhang, W.: Primary discontinuities in solutions for delay integro-differential equations. Methods Appl. Anal. **6**, 525–534 (1999)
12. Chen, H., Zhang, C.: Convergence and stability of extended block boundary value methods for Volterra delay integro-differential equations. Appl. Numer. Math. **62**, 141–154 (2012)
13. Cirillo, E., Hormann, K., Sidon, J.: Convergence rates of derivatives of Floater-Hormann interpolants for well-spaced nodes. Appl. Numer. Math. **116**, 108–118 (2017)
14. Floater, M.S., Hormann, K.: Barycentric rational interpolation with no poles and high rates of approximation. Numer. Math. **107**, 315–331 (2007)
15. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Problems, 2nd edn. Springer, Berlin (1996)
16. Hormann, K.: Barycentric interpolation. In Fasshauer G. E., Schumaker L.L., Approximation Theory XIV: San Antonio 2013, Springer Proceedings in Mathematics & Statistics, vol. 83, Springer, New York, pp. 197–218 (2014)
17. Huang, C.: Stability of linear multistep methods for delay integro-differential equations. Comput. Math. Appl. **55**, 2830–2838 (2008)
18. Huang, C., Vandewalle, S.: Stability of Runge-Kutta-Pouzet methods for Volterra integro-differential equations with delays. Front. Math. China **4**, 63–87 (2009)

19. Jerri, A.J.: Introduction to Integral Equations with Applications, 2nd edn. Sampling Publishing, Potsdam (2007)
20. Klein, G.: Applications of Linear Barycentric Rational Interpolation. Ph.D. thesis, University of Fribourg (2012)
21. Klein, G., Berrut, J.-P.: Linear barycentric rational quadrature. BIT Numer. Math. **52**, 407–424 (2012)
22. Klein, G., Berrut, J.-P.: Linear rational finite differences from derivatives of barycentric rational interpolants. SIAM J. Numer. Anal. **52**, 643–656 (2012)
23. Lambert, J.D.: Computational Methods in Ordinary Differential Equations. Wiley, New York (1973)
24. Li, D., Zhang, C.: L^{∞} error estimates of discontinuous Galerkin methods for delay differential equations. Appl. Numer. Math. **82**, 1–10 (2014)
25. Marino, S., Beretta, E., Kirschner, D.E.: The role of delays in innate and adaptive immunity to intracellular bacterial infection. Math. Biosci. Eng. **4**, 261–286 (2007)
26. Shakourifar, M., Dehghan, M.: On the numerical solution of nonlinear systems of Volterra integro-differential equations with delay arguments. Computing **82**, 241–260 (2008)
27. Shakourifar, M., Enright, W.: Superconvergent interpolants for collocation methods applied to Volterra integro-differential equations with delay. BIT Numer. Math. **52**, 725–740 (2012)
28. Schneider, C., Werner, W.: Some new aspects of rational interpolation. Math. Comput. **47**, 285–299 (1986)
29. Shampine, L.F., Thompson, S.: Solving DDEs in MATLAB. Appl. Numer. Math. **37**, 441–458 (2001)
30. Tang, T., Xu, X., Cheng, J.: On spectral methods for Volterra integral equations and the convergence analysis. J. Comput. Math. **26**, 825–837 (2008)
31. Trefethen, L.N. et al.: Chebfun Version 4.2, The Chebfun Development Team. http://www.maths.ox.ac.uk/chebfun/ (2011)
32. Tsitouras, C.: Runge-Kutta interpolants for high precision computations. Numer. Algorithms **44**, 291–307 (2007)
33. Wu, S., Gan, S.: Errors of linear multistep methods for singularly perturbed Volterra delay-integro-differential equations. Math. Comput. Simul. **79**, 3148–3159 (2009)
34. Zhang, C., Vandewalle, S.: General linear methods for Volterra integro-differential equations with memory. SIAM J. Sci. Comput. **27**, 2010–2031 (2006)
35. Zhang, C., Vandewalle, S.: Stability analysis of Runge-Kutta methods for nonlinear Volterra delay-integro-differential equations. IMA. J. Numer. Anal. **24**, 193–214 (2004)
36. Zhang, C., Vandewalle, S.: Stability analysis of Volterra delay-integro-differential equations and their backward differentiation time discretization. J. Comput. Appl. Math. **164–165**, 797–814 (2004)