Linear barycentric rational Hermite quadrature and its application to Volterra integral equations

Ali Abdi · Kai Hormann · Seyyed Ahmad Hosseini

Abstract

Quadrature rules are a common numerical tool for approximating definite integrals. While most classical rules are based on polynomial interpolation, recent results reveal the efficiency and effectiveness of quadrature rules based on linear barycentric rational interpolants. In this paper, we derive new quadrature rules from barycentric rational Hermite interpolants and prove their convergence orders. We then use the proposed quadrature rules to construct direct quadrature methods for solving Volterra integral equations. We provide several numerical experiments that validate our theoretical results and illustrate the efficiency of our new quadrature rules and methods.

Citation Info

Journal
Journal of Computational and Applied Mathematics

Volume
474, March 2026

Article
117009, 14 pages

DOI
10.1016/j.cam.2025.117009

1 Introduction

As shown by Berrut and Mittelmann [11], any rational interpolant to data $f_0,\dots,f_n$ given at a discrete set of interpolation nodes $t_0,\dots,t_n$ can be written in its *barycentric form*

$$r_n(t) = \sum_{k=0}^{n} \frac{\beta_k}{t-t_k} f_k \Bigg/ \sum_{k=0}^{n} \frac{\beta_k}{t-t_k} \tag{1}$$

for certain *barycentric weights* $\beta_0,\dots,\beta_n$. Vice versa, fixing *any* set of non-zero barycentric weights, the barycentric form gives a rational interpolant [54] that is linear in the data and hence often called a *linear barycentric rational interpolant* (LBRI).

Barycentric rational interpolation offers an elegant approach to avoid a common problem of rational interpolation, namely the occurrence of poles in the interpolation interval, which is undesirable in many situations. Schneider and Werner [48] derive necessary conditions on the barycentric weights to prevent poles, and Berrut [8] later found a very simple choice that guarantees the absence of poles in the interpolation interval. He also observes that the approximation order of this interpolation method is linear and even quadratic if modified slightly. Floater and Hormann [31] prove this behaviour and introduce a whole family of LBRIs, which can be expressed as a rational blend of local polynomial interpolants of degree $d \leqslant n$.

The interpolants from this family are guaranteed to have no real poles, reproduce polynomials up to degree d, and include Berrut's interpolants as well as polynomial interpolants as special cases (for $d = 0$ and $d = n$, respectively). Moreover, they can be evaluated efficiently and numerically stable in $O(n)$ runtime [33], and they have some remarkable properties, both with respect to the approximation order and to the Lebesgue constant. On the one hand, their approximation order is essentially $O(h^{d+1})$ [31], where h is the maximal distance between neighbouring interpolation nodes, and this rate carries over to the derivatives, in the sense that the k-th derivative has an approximation order $O(h^{d+1-k})$ for $k \leqslant d$ [9, 24, 40]. On the other hand, the Lebesgue constant of these barycentric rational interpolants grows logarithmically with n for equidistant nodes [13, 14], a stark contrast to the exponential growth experienced by polynomial interpolation.

Due to these attractive features, LBRIs have recently gained popularity and are used in the construction of various numerical methods for solving PDEs [41, 46] and different classes of time-dependent problems [1, 3, 5, 6, 10, 42, 43, 45]. In particular, some of these methods rely on efficient and effective quadrature rules that can be derived from LBRIs [39].

The first goal of this paper is to construct similar quadrature rules based on barycentric rational Hermite interpolation. We depart from the observation by Schneider and Werner [49] that a barycentric form, similar to the one in (1), also exists for rational Hermite interpolants, and we recall the iterative approach by Cirillo and Hormann [22] for constructing linear barycentric rational Hermite interpolants (LBRHIs). They show

how to successively add m correction terms to the initial Lagrange interpolant, so as to interpolate the data up to the m-th derivative. As for LBRIs, the barycentric form can be used to efficiently evaluate these LBRHIs in a numerically stable way, and they turn out to be particularly suited for equidistant nodes. In this setting, their approximation order is $O^{(m+1)(d+1)}$ [25], and in the special case of prescribed functions values and first derivatives ($m = 1$), the Lebesgue constant associated with these interpolants is bounded from above by a constant [23].

After briefly reviewing LBRHIs in Section 2, we introduce a *barycentric rational Hermite quadrature* (BRHQ) as well as a *composite barycentric rational Hermite quadrature* (CBRHQ) rule in Section 3. We prove the order of convergence for both rules and validate the theoretical results with a series of numerical experiments in Section 4.

To further demonstrate the practicality of these new quadrature rules, we use them to derive direct methods for solving Volterra integral equations (VIEs) of the second kind,

$$y(t) = g(t) + \int_{t_0}^{t} k(t, s, y(s))\,\mathrm{d}s, \qquad t \in I = [t_0, T], \tag{2}$$

where $g\colon I \to \mathbb{R}^D$ and $k\colon S \times \mathbb{R}^D \to \mathbb{R}^D$ are given functions, D stands for the dimension of the system, and $S = \{(t, s) : t_0 \leqslant s \leqslant t \leqslant T\}$. In the subsequent discussion, we assume that g and k are sufficiently smooth to guarantee the existence of a unique smooth solution y [16, 44].

Volterra equations are a class of mathematical equations that extend the concept of ordinary differential equations by incorporating an integral term. These equations are instrumental in mathematical modelling of complex dynamic systems and natural phenomena with memory where the current state depends not only on its current conditions, but also on the accumulated effects of past interactions. In essence, they are more than just a small portion of functional equations and hold a significant position in time-dependent problems as well as other areas of analysis and practical applications.

In general, equations of Volterra type (2) cannot be solved analytically, necessitating numerical methods that yield approximations of the exact solutions. An effective numerical method should simultaneously have acceptable accuracy, suitable stability properties, and a modest computational cost. Over the recent decades, a multitude of comprehensive numerical investigations has been developed to address the numerical solutions for various kinds of Volterra equations. Broadly, the methods can be categorized into three main classes: global methods, local methods, and boundary value methods. A representative global method is the spectral method. They approximate the solution over the entire domain using global basis functions and often exhibit exponential convergence in smooth problems [29]. Local methods include multistep and/or multistage methods [20, 34, 36], direct quadrature methods [2, 3, 10, 18, 44], collocation and piecewise collocation [19, 26, 27], general linear methods [4, 37], as well as combinations of certain classes of ordinary differential equation (ODE) methods paired with appropriate quadrature rules [7, 53]. These methods typically employ discretization strategies that facilitate adaptive or stepwise progression. Boundary value methods, which fall between global and local approaches, treat the problem on a fixed interval with boundary conditions and often exhibit favorable stability properties [28]. Advantages and drawbacks of all classes have been discussed extensively in the literature; see [16, 17] and the references therein.

Despite the exponential convergence of global methods, they are not among the most effective general-purpose methods for solving VIEs due to some inherent drawbacks. One of them stems from the fact that these methods treat Volterra equations as Fredholm equations. As a result, the functions involved must be sampled at points clustered near the boundary, which necessitates knowing the value of the endpoint of the integration interval in advance. This requirement may not seem very natural, particularly in time-dependent problems where the value of T might not be known beforehand or could dynamically change over the course of the problem. It hinders the efficient treatment of steep gradients occurring away from the boundary and impedes the use of step control procedure, and that a restart from scratch for any increase of T is required; thereby rendering real-time or dynamic applications impractical. Moreover, a large dense system of (nonlinear) equations must be solved to obtain the approximate solution, which becomes a significant challenge and increases the computational cost, especially for large D.

Local methods are free of these drawbacks and commonly used for solving Volterra type equations, because they have desirable stability properties and good accuracy. In particular, direct quadrature methods are among the simplest and most straightforward approaches for numerically solving VIEs, encompassing reducible quadrature methods [47], exponential fitting direct quadrature methods [21], direct spectral integration [50], and various composite schemes. Among these, composite Newton–Cotes formulas, Gregory's rules [12], and hybrid Newton–Cotes–Gregory approaches [38] are commonly used. However, it is important

to note that these methods may encounter challenges when the required order of accuracy is increased, as the appearance of Runge's phenomenon under these circumstances might cause problems and render them impractical [32]. It should be noted that, composite Newton–Cotes formulas mitigate some issues related to Runge's phenomenon, making them suitable for achieving higher accuracy; nevertheless, they have the drawback that these formulas require alternating between different quadrature rules to obtain a method with a prescribed order of accuracy, a strategy that is typically effective only for low-order approximations.

In Section 5 we discuss how to solve the VIE in (2) with a direct quadrature method based on our global and composite Hermite quadrature rules. We prove the convergence orders of both methods and verify the theoretical results in Section 6. Our numerical experiments confirm the robustness of these methods for solving VIEs.

2 Linear barycentric rational Hermite interpolation

Before we can introduce our novel barycentric Hermite quadrature rules, let us recall the ideas that led to the iterative approach for solving the Hermite interpolation problem in [22]. Let f be a real-valued and continuously differentiable function over the interval $[a,b]$ and consider the $n+1$ distinct interpolation nodes

$$a = t_0 < t_1 < \cdots < t_n = b.$$

The first order Hermite interpolation problem then consists in finding a function $r_n^{(1)}[f] \in C^1[a,b]$, such that

$$r_n^{(1)}[f](t_k) = f_k, \qquad r_n^{(1)}[f]'(t_k) = f_k', \qquad k = 0,1,\ldots,n,$$

where $f_k = f(t_k)$ and $f_k' = f'(t_k)$.

The classical polynomial Hermite interpolant to these data can be expressed as

$$r_n^{(1)}[f](t) = \sum_{k=0}^{n} \ell_k^{(0)}(t) f_k + \sum_{k=0}^{n} \ell_k^{(1)}(t) f_k', \tag{3}$$

where

$$\ell_k^{(0)}(t) = \big(1 - 2(t-t_k)\ell_k'(t_k)\big)\ell_k(t)^2, \qquad \ell_k^{(1)}(t) = (t-t_k)\ell_k(t)^2,$$

with ℓ_k denoting the k-th Lagrange basis polynomial

$$\ell_k(t) = \prod_{j=0,\, j\neq k}^{n} \frac{t-t_j}{t_k-t_j}.$$

Cirillo and Hormann [22] propose to write the interpolant in (3) instead in terms of the Lagrange interpolant

$$r_n^{(0)}[f](t) = \sum_{k=0}^{n} \ell_k(t) f_k$$

and a correction term, namely as

$$r_n^{(1)}[f](t) = r_n^{(0)}[f](t) + \sum_{k=0}^{n} (t-t_k)\ell_k(t)^2 \big(f_k' - r_n^{(0)}[f]'(t_k)\big),$$

and they observe that this is a Hermite interpolant for *any* choice of basis functions $\ell_0, \ell_1, \ldots, \ell_n$ that satisfies the Lagrange property $\ell_k(t_j) = \delta_{j,k}$ for $j,k = 0,1,\ldots,n$.

In an attempt to carry over the favourable properties of LBRIs to the Hermite setting, they suggest to replace the Lagrange basis polynomials ℓ_k with the basis functions

$$b_k(t) = \frac{\beta_k}{t-t_k} \Bigg/ \sum_{j=0}^{n} \frac{\beta_j}{t-t_j} \tag{4}$$

of the LBRI in (1) and show that the resulting LBRHI can be expressed in barycentric form as

$$r_n^{(1)}[f](t) = \sum_{k=0}^{n} b_k^{(0)}(t) f_k + \sum_{k=0}^{n} b_k^{(1)}(t) f_k', \tag{5}$$

where

$$b_k^{(0)}(t) = \left(\frac{\beta_k^{(0)}}{t-t_k} + \frac{\beta_k^{(1)}}{(t-t_k)^2}\right) \Bigg/ \sum_{j=0}^{n}\left(\frac{\beta_j^{(0)}}{t-t_j} + \frac{\beta_j^{(1)}}{(t-t_j)^2}\right),$$

$$b_k^{(1)}(t) = \frac{\beta_k^{(1)}}{t-t_k} \Bigg/ \sum_{j=0}^{n}\left(\frac{\beta_j^{(0)}}{t-t_j} + \frac{\beta_j^{(1)}}{(t-t_j)^2}\right).$$

The weights $\beta_k^{(0)}$ and $\beta_k^{(1)}$ are defined in terms of the barycentric weights β_k as

$$\beta_k^{(0)} = 2\beta_k \nu_k, \qquad \beta_k^{(1)} = \beta_k^2, \qquad k = 0, 1, \ldots, n, \tag{6}$$

where

$$\nu_k = \sum_{j=0,\, j\neq k}^{n} \frac{\beta_j}{t_k - t_j},$$

and they depend only on the interpolation nodes t_k, but not on the data f_k and f_k'.

For the barycentric weights proposed by Floater and Hormann [31],

$$\beta_k = \sum_{i=\max(0,k-d)}^{\min(k,n-d)} (-1)^i \prod_{j=i,\, j\neq k}^{i+d} \frac{1}{t_k - t_j}, \tag{7}$$

where $0 \leqslant d \leqslant n$, the following theorem from [22] gives the rate of convergence of the rational Hermite interpolant (5) via a bound on the interpolation error in the maximum norm.

Theorem 1. *For any $f \in C^{2(d+2)}[a,b]$, we have*

$$\|f - r_n^{(1)}[f]\| \leqslant C h^{2(d+1)},$$

where $h = \max_{0\leqslant k\leqslant n-1}(t_{k+1} - t_k)$ is the global mesh size and the constant C depends only on d, the derivatives of f, the interval length $b-a$, and, only in the case $d=0$, on the maximal local mesh ratio

$$\rho = \max_{1\leqslant k\leqslant n-2} \min\left\{\frac{t_{k+1}-t_k}{t_k - t_{k-1}}, \frac{t_{k+1}-t_k}{t_{k+2}-t_{k+1}}\right\}.$$

In the following, we are mainly interested in the case of equispaced nodes, when the weights in (7) can be replaced by

$$\bar{\beta}_k = (-1)^d d!\, h^d \beta_k = (-1)^k \sum_{i=d}^{n} \binom{d}{i-k}, \qquad k = 0, 1, \ldots, n \tag{8}$$

and the Lebesgue constant associated with the LBRHI in (5) is bounded from above by a constant [23], in contrast to the exponential growth experienced by the classical polynomial Hermite interpolants. Moreover, they do not suffer from the Runge phenomenon in this setting. Consequently, LBRHIs can be considered a viable and practical alternative to polynomial Hermite interpolants.

3 Linear barycentric rational Hermite quadrature

Numerical integration is essential for simulations, data analysis, and numerical modelling, and it plays a crucial role in solving practical problems in numerous areas of computational science and engineering. The linearity of barycentric rational Hermite interpolation in the data makes it well-suited for applications. Our main concern here is using this property for introducing a quadrature formula based on the LBRHI in (5) to approximate the definite integral

$$I(f) = \int_a^b f(t)\,\mathrm{d}t, \tag{9}$$

for a real integrable function f over the integration interval $[a,b]$.

To this end, we consider the uniform partition $a = t_0 < t_1 < \cdots < t_n = b$ of $[a,b]$ with stepsize $h = (b-a)/n$, that is, $t_k = a + kh$ for $k = 0,1,\ldots,n$. Integrating the LBRHI in (5) with respect to this partition leads to the linear barycentric rational Hermite quadrature (BRHQ) formula

$$\int_a^b f(t)\,\mathrm{d}t \approx \int_a^b r_n^{(1)}[f](t)\,\mathrm{d}t = h\sum_{k=0}^{n} w_{n,k}^{(0)} f_k + h^2 \sum_{k=0}^{n} w_{n,k}^{(1)} f_k' = Q_n^G, \tag{10}$$

with quadrature weights

$$\begin{aligned} w_{n,k}^{(0)} &= h^{-1}\int_a^b b_k^{(0)}(t)\,\mathrm{d}t = \int_0^n \phi_{n,k}^{(0)}(x)\,\mathrm{d}x, \\ w_{n,k}^{(1)} &= h^{-2}\int_a^b b_k^{(1)}(t)\,\mathrm{d}t = \int_0^n \phi_{n,k}^{(1)}(x)\,\mathrm{d}x, \end{aligned} \tag{11}$$

where

$$\begin{aligned} \phi_{n,k}^{(0)}(x) &= \left(\frac{2\bar\beta_k\bar\nu_k}{x-k} + \frac{\bar\beta_k^2}{(x-k)^2}\right) \Bigg/ \sum_{j=0}^{n}\left(\frac{2\bar\beta_j\bar\nu_j}{x-j} + \frac{\bar\beta_j^2}{(x-j)^2}\right), \\ \phi_{n,k}^{(1)}(x) &= \frac{\bar\beta_k^2}{x-k} \Bigg/ \sum_{j=0}^{n}\left(\frac{2\bar\beta_j\bar\nu_j}{x-j} + \frac{\bar\beta_j^2}{(x-j)^2}\right), \end{aligned}$$

with $\bar\beta_k$ as in (8) and $\bar\nu_k = \sum_{j=0,\,j\neq k}^{n} \bar\beta_j/(k-j)$.

Since the integrands in (11) are rational functions that cannot be integrated analytically without further information about their properties, they must be computed numerically up to machine precision, using, for example, the routines implemented in the Chebfun system [51] or, alternatively, with Gauss–Legendre or Clenshaw–Curtis rules [35, 52]. It is worth noting that barycentric form offers greater flexibility than Gauss-Legendre quadrature, as it allows for arbitrary node distributions instead of being limited to the roots of Legendre polynomials. Moreover, it provides excellent stability properties and simpler weight computation through the barycentric formula. Barycentric rational quadrature is also better suited for handling endpoint singularities and functions with sharp gradients, due to its use of a rational approximation basis.

As LBRHIs reproduce polynomials of degree $2d+1$ if $n-d$ is even and of degree $2d+3$ if $n-d$ is odd [22, Corollary 1], it is clear that the BRHQ rule (10) is precise up to these degrees. More precisely, as shown in the proof of [22, Theorem 3], for $f \in C^{2d+4}[a,b]$, we have

$$\|f - r_n^{(1)}[f]\| \leqslant C h^{2(d+1)} (b-a)^{2\delta} \|f^{(2d+2+2\delta)}\|, \tag{12}$$

where the constant C depends only on d and $\delta = 0$ if $n-d$ is even and $\delta = 1$ if $n-d$ is odd; this means that the *degree of precision* of the BRHQ rule (10) is at least $2(d+\delta)+1$. Analogous to the proof of Theorem 6.2 in [39], it is easy to show that the BRHQ rule (10) with $\delta = 0$ and $\delta = 1$ is not exact for the functions $x^{2(d+1)}$ and $x^{2(d+2)}$, respectively, hence the exact degree of precision of the BRHQ rule (10) is $2(d+\delta)+1$. In what follows, we prove that the convergence rate is $O(h^{2(d+1)})$.

Theorem 2. *For any $f \in C^{2(d+2)}[a,b]$, the approximation error of the BRHQ rule (10) satisfies*

$$\left|I(f) - Q_n^G\right| \leqslant C h^{2(d+1)},$$

where C depends only on d, the derivatives of f, and the interval length $b-a$.

Proof. Assuming that the quadrature weights in (11) are approximated by a quadrature rule that converges at least at the rate $O(h^{2(d+1)})$, it suffices to show that

$$\int_a^b \left(f(t) - r_n^{(1)}[f](t)\right)\mathrm{d}t \tag{13}$$

converges to zero as $O(h^{2(d+1)})$. To this end, we recall from [22] that the interpolation error of the LBRHI in (5) can be written as

$$f(t) - r_n^{(1)}[f](t) = \frac{A(t)}{W(t)^2},$$

where $A(t)=\sum_{i=0}^{n}\beta_i\sum_{j=0}^{n}\beta_j f[t,t_i,t_j]$, with $f[t,t_i,t_j]$ denoting the second divided difference of f at the points t,t_i,t_j, and

$$W(t)=\sum_{k=0}^{n}\frac{\beta_k}{t-t_k}$$

is the denominator of the barycentric basis functions in (4). Since the function $\frac{1}{W(t)^2}$ is a non-negative function, the mean value theorem for integrals asserts that

$$\int_a^b\big(f(t)-r_n^{(1)}[f](t)\big)\,\mathrm{d}t=A(\tau)\int_a^b\frac{1}{W(t)^2}\,\mathrm{d}t,$$

for some $\tau\in[a,b]$. We know from the proof of Theorem 3 in [22] that $A(t)$ is bounded by a constant; moreover, for remember from [31, Section 4] that

$$W(t)=\sum_{k=0}^{n-d}\lambda_k(t),$$

in which $\lambda_k^{-1}(t)=(-1)^k\prod_{j=0}^{d}(t-t_{k+j})$ are the blending functions in the Floater-Hormann interpolants. It was proved in Theorems 2 and 3 in [31] that

$$|W(t)|\geqslant\begin{cases}\dfrac{1}{d!h^{d+1}}, & d\geqslant 1,\\[2mm] \dfrac{1}{(1+\beta)h}, & d=0,\end{cases}$$

with β as the local mesh ratio defined by

$$\beta=\max_{1\leqslant i\leqslant n-2}\min\left\{\frac{t_{i+1}-t_i}{t_i-t_{i-1}},\frac{t_{i+1}-t_i}{t_{i+2}-t_{i+1}}\right\}.$$

Therefore,

$$\left|\int_a^b\big(f(t)-r_n^{(1)}[f](t)\big)\,\mathrm{d}t\right|\leqslant Ch^{2(d+1)},$$

where the constant C depends only on the integrand function f, d, and the interval length $b-a$. □

The basic idea behind the BRHQ rule (10) is to approximate the integrand using the linear barycentric rational Hermite interpolant (5), and to then apply the operator to the latter. If the resulting quadrature formula is directly used in discretization methods for the numerical solution of time-dependent problems, specifically Volterra type equations, the computational cost may grow significantly as the number of partition nodes increases, since new quadrature weights must be computed at each step. By considering the fact that the barycentric weights (11) are independent of the nodes, hence translation invariant, a composite version of the BRHQ rule can be constructed to remedy this drawback.

To this end, we consider the uniform partition $a=t_0<t_1<\cdots<t_N=b$ of $[a,b]$ with stepsize $h=(b-a)/N$, that is, $t_k=a+kh$ for $k=0,1,\ldots,N$. To define the composite BRHQ formula for approximating the integral of a given function f over $[a,b]$, let d and n with $0\leqslant d\leqslant n\leqslant N/2$ and set $p=\lfloor N/n\rfloor-1$. The CBRHQ rule can then be written as

$$\begin{aligned}I(f)&=\sum_{j=0}^{p-1}\int_{t_{jn}}^{t_{(j+1)n}}f(t)\,\mathrm{d}t+\int_{t_{pn}}^{t_N}f(t)\,\mathrm{d}t\\&\approx\sum_{j=0}^{p-1}\sum_{k=0}^{n}\big(hw_{n,k}^{(0)}f_{jn+k}+h^2w_{n,k}^{(1)}f'_{jn+k}\big)+\sum_{k=0}^{N-pn}\big(hw_{N-pn,k}^{(0)}f_{pn+k}+h^2w_{N-pn,k}^{(1)}f'_{pn+k}\big)=Q_m^C,\end{aligned}\tag{14}$$

where

$$\begin{aligned}w_{m,k}^{(0)}&=h^{-1}\int_{t_0}^{t_m}b_{m,k}^{(0)}(t)\,\mathrm{d}t=\int_0^m\phi_{m,k}^{(0)}(x)\,\mathrm{d}x,\\w_{m,k}^{(1)}&=h^{-2}\int_{t_0}^{t_m}b_{m,k}^{(1)}(t)\,\mathrm{d}t=\int_0^m\phi_{m,k}^{(1)}(x)\,\mathrm{d}x\end{aligned}$$

Table 1: Test problems.

Experiment	Integrand	Integration interval	Figure	Table
1	$1/(1+t^2)$	$[-1,1]$	—	2
2	$\sin(t)$	$[-4,5]$	—	3
3	$(1+\tanh(-9t+1))/2$	$[0,1]$	—	4
4	$e^{-(t-1/2)^2/2}$	$[0,3]$	—	5
5	$\sin^2 t\cos(10\tanh t)$	$[0,1]$	1	—
6	$(1+\ln t)\cos(t\ln t)$	$[100,200]$	2	—

for $m = n, n+1, \ldots, 2n-1$ and $k = 0, \ldots, m$ and $f_{\ell n+k}$. Notice that in the case of $n \leqslant N \leqslant 2n$, we just have the last term of (14) which is the BRHQ rule. The following theorem establishes the order of convergence of the CBRHQ formula (14).

Theorem 3. *Suppose N, n are positive integers with $n \leqslant N$, d is non-negative integer with $d \leqslant n \leqslant N/2$, and $f \in C^{2(d+2)}[a,b]$. Then the CBRHQ formula (14) converges at the rate $O(h^{2(d+\delta)+2})$, where $\delta = 0$ if $n-d$ is even and $\delta = 1$ if $n-d$ is odd.*

Proof. Using the change of variable $t = a + xh$, $x \in [0,m]$ and noting that the LBRHIs (5) error is bounded by $Ch^{2(d+1)}$, it is easy to see, by taking maximum norm, that each local quadrature formula tends to zero as $O(h^{2(d+1)+1})$. Since n is fixed and there are $p+1 = O(n) = O(1/h)$ integrals to be computed, the order of the CBRHQ formula (14) is $2d+2$. Moreover, according to Theorem 3 in [22] and relation (12), in the case of $\delta = 1$, the bound on the interpolation error involves an additional factor $(nh)^2$, so the order of convergence is two units larger than the stated above, that is, $2(d+\delta)+2$, where $\delta = 0$ for even values of $n-d$ and $\delta = 1$ for odd values of $n-d$. □

4 Numerical experiments with the quadrature rules

In this section, some numerical experiments are provided to demonstrate the efficiency, robustness, and accuracy of the introduced quadrature formulas as well as verifying the theoretical results. To this end, a set of test problems are listed in Table 1. For each experiment, we report the approximation error $E_N = |I(f) - Q_N|$ between the exact value of the definite integral and values of the BRHQ rule (10) and the CBRHQ rule (14), as well as its corresponding experimental convergence orders $O_G^Q = \log_2(E_N^G/E_{N/2}^G)$ and $O_C^Q = \log_2(E_N^C/E_{N/2}^C)$. The superscripts G and C in E_N are used to refer to the corresponding errors of the BRHQ rule (10) and the CBRHQ rule (14), respectively.

Tables 2–5 present the numerical results for various choices of n and d for the first four experiments. They demonstrate the efficiency and power of the proposed quadrature formulas and confirm the theoretical convergence orders stated in Theorems 2 and 3. It should be noted that for some values of N, the approximations reaches machine precision, which explains some smaller experimental orders in the tables. Moreover, according to Theorem 3, the CBRHQ errors are expected to decrease with order $2(d+\delta)+2$, where $\delta = 1$ for odd values of $n-d$ and $\delta = 0$ for even values of $n-d$, which is numerically verified by the experimental approximation orders reported in these tables.

In Figure 1, the logarithmic error $\log_{10}(E_N)$ is plotted versus $\log_{10}(N)$, together with the slope lines corresponding to the expected convergence rates, for Experiment 5, using the BRHQ rule (10) and the CBRHQ rule (14).

The BRHQ rule is very accurate but slow, particularly when dealing with the approximation of integrals over large integration intervals. In such instances, switching to the CBRHQ rule, having an order of at least $2d+2$, turns out to be more efficient and cost-effective. Several practical and physical problems, such as analysing water waves on sloping beaches, have motivated the study of irregular oscillatory integrands [30]. In the final experiment, a definite integral with such irregular oscillatory integrand is approximated with the CBRHQ rule (14). The numerical results in Figure 2 confirm the theoretical results given in Theorem 3 and show the high efficiency and capability of the CBRHQ rule for approximating such integrals. Moreover, the numerical results of the LBRI-based composite quadrature rule introduced in [10, 39], which is referred to as the CBRQ rule, have been included in this figure. These results illustrate that the CBRHQ rule is more accurate than the CBRQ rule of the same order.

Table 2: Error and approximation order for Experiment 1.

N		10	20	40	80	160	320
$d=0$	E_N^G	6.37×10^{-4}	1.79×10^{-4}	4.71×10^{-5}	1.21×10^{-5}	3.07×10^{-6}	7.72×10^{-7}
	O_G^Q		1.83	1.93	1.96	1.98	1.99
$d=2$	E_N^G	1.15×10^{-7}	1.53×10^{-9}	2.17×10^{-11}	3.22×10^{-13}	5.55×10^{-15}	6.66×10^{-16}
	O_G^Q		6.23	6.14	6.07	5.86	3.06
$(n,d)=(2,0)$	E_N^C	9.15×10^{-4}	2.29×10^{-4}	5.72×10^{-5}	1.43×10^{-5}	3.57×10^{-6}	8.93×10^{-7}
	O_C^Q		2.00	2.00	2.00	2.00	2.00
$(n,d)=(8,1)$	E_N^C	8.56×10^{-6}	3.74×10^{-7}	4.73×10^{-10}	9.39×10^{-12}	1.47×10^{-13}	2.22×10^{-15}
	O_C^Q		4.52	9.63	5.66	6.00	6.05

Table 3: Error and approximation order for Experiment 2.

N		10	20	40	80	160	320
$d=1$	E_N^G	5.37×10^{-4}	3.19×10^{-5}	1.96×10^{-6}	1.21×10^{-7}	7.55×10^{-9}	4.71×10^{-10}
	O_G^Q		4.07	4.02	4.02	4.00	4.00
$d=3$	E_N^G	3.25×10^{-6}	6.79×10^{-9}	1.66×10^{-11}	4.86×10^{-14}	3.33×10^{-16}	1.11×10^{-15}
	O_G^Q		8.90	8.68	8.42	7.19	-1.75
$(n,d)=(5,1)$	E_N^C	2.43×10^{-4}	2.35×10^{-5}	1.54×10^{-6}	9.71×10^{-8}	6.08×10^{-9}	3.80×10^{-10}
	O_C^Q		3.37	3.93	3.99	4.00	4.00
$(n,d)=(5,2)$	E_N^C	2.50×10^{-5}	5.14×10^{-8}	1.78×10^{-10}	6.76×10^{-13}	2.66×10^{-15}	8.88×10^{-16}
	O_C^Q		8.93	8.17	8.04	7.99	1.58

Table 4: Error and approximation order for Experiment 3.

N		10	20	40	80	160	320
$(n,d)=(5,1)$	E_N^C	2.18×10^{-5}	1.07×10^{-6}	3.37×10^{-8}	3.48×10^{-9}	2.23×10^{-10}	1.40×10^{-11}
	O_C^Q		4.35	4.99	3.28	3.96	3.99
$(n,d)=(6,2)$	E_N^C	8.32×10^{-5}	6.73×10^{-8}	1.95×10^{-9}	2.45×10^{-11}	3.84×10^{-13}	6.11×10^{-15}
	O_C^Q		10.27	5.11	6.31	6.00	5.97

Table 5: Error and approximation order for Experiment 4.

N		10	20	40	80	160	320
$(n,d)=(3,1)$	E_N^C	7.19×10^{-6}	6.57×10^{-7}	3.97×10^{-8}	2.68×10^{-9}	1.66×10^{-10}	1.05×10^{-11}
	O_C^Q		3.45	4.05	3.89	4.01	3.98
$(n,d)=(6,1)$	E_N^C	1.39×10^{-5}	2.57×10^{-7}	3.08×10^{-9}	3.57×10^{-11}	5.46×10^{-13}	8.66×10^{-15}
	O_C^Q		5.76	6.38	6.43	6.03	5.98

As a comparison with the Filon and Levin methods [15], we should say that the Filon and Levin methods are specialized techniques designed primarily for efficiently evaluating highly oscillatory integrals. In contrast, barycentric rational quadrature offers greater flexibility in node selection and excels in handling singularities due to its rational basis but is less specialized for very high-frequency oscillations. Thus, while Filon and Levin methods are typically preferred for integrals dominated by oscillatory behavior, barycentric rational quadrature is advantageous when dealing with functions featuring singularities or rapidly varying features not necessarily tied to oscillations.

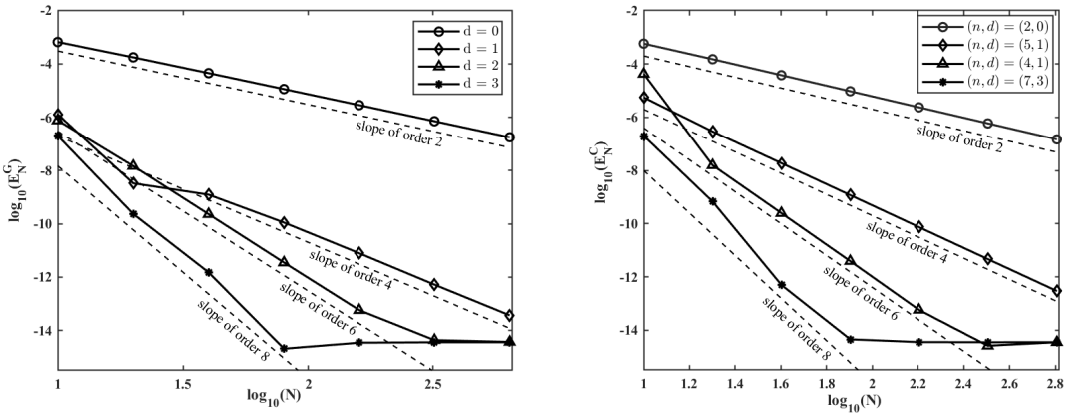

Figure 1: Log-log-plots of the approximation error for Experiment 5 with the BRHQ rule (10) (left) and the CBRHQ rule (14) (right).

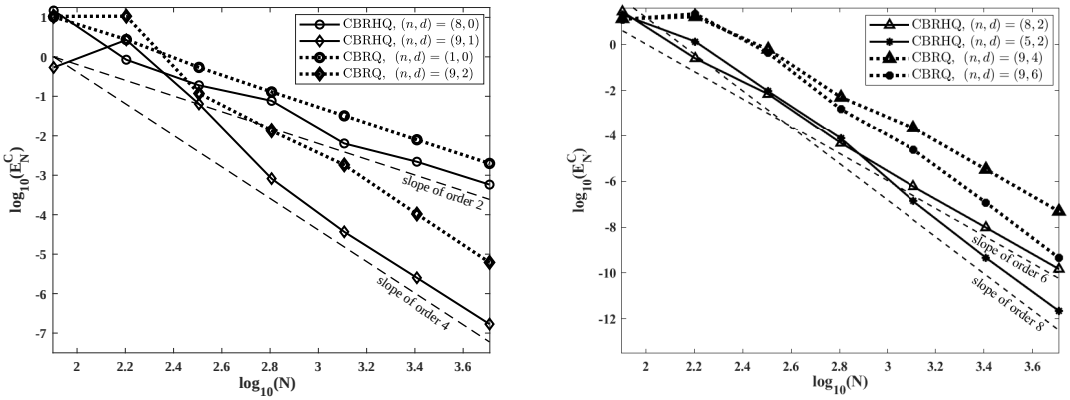

Figure 2: Log-log-plots of the approximation error for Experiment 6 with the CBRHQ rule (14) and the CBRQ rule [10, 39] of orders 2 and 4 (left), and of orders 6 and 8 (right).

5 Linear barycentric rational Hermite quadrature method (LBRHQM) for VIEs

In this section, we are interested in using the advantages of the quadrature rules introduced above, such as smoothness, high accuracy, and arbitrarily high convergence order to construct direct quadrature methods for the numerical solution of VIEs (2). To this end, we introduce two methods based on the BRHQ rule (10) and the CBRHQ rule (14), called the global barycentric rational Hermite quadrature method (GBRHQM) and the composite barycentric rational Hermite quadrature method (CBRHQM), and we discuss their convergence in detail.

5.1 The global version

Let $T_N = \{t_0, t_1, \ldots, t_N = T\}$ be a uniform partition of the given interval I with the fixed stepsize $h = t_{i+1} - t_i = (T - t_0)/N$, $i = 0, 1, \ldots, N-1$ and set $F_1(s, y(s)) = k(t_m, s, y(s))$. The GBRHQM for (2) can be obtained by applying the BRHQ rule (10) to the integral part of (2) at the mesh point t_m as

$$y_m = g(t_m) + h \sum_{k=0}^{m} w_{m,k}^{(0)} \widetilde{F}_{1,k} + h^2 \sum_{k=0}^{m} w_{m,k}^{(1)} \widetilde{F}'_{1,k}, \qquad m = n+1, \ldots, N, \tag{15}$$

with $w^{(0)}_{m,k}$ and $w^{(1)}_{m,k}$ given by (11). Here, y_m represents an approximation to the exact solution y of (2) at the mesh point t_m and

$$\widetilde{F}_{1,k} = F_1(t_k, y_k),$$
$$\widetilde{F}'_{1,k} = F'_1(t_k, y_k) = k_s(t_m, t_k, y_k) + k_y(t_m, t_k, y_k)y'_k,$$

where k_s and k_y denote the partial derivatives of the kernel k with respect to s and y, respectively.

To compute the values of the unknowns y_m, an approximation of the exact value y' at t_m must be provided. To supply it, first differentiate (2) to obtain

$$y'(t) = g'(t) + k(t, t, y(t)) + \int_{t_0}^{t} k_t(t, s, y(s))\,\mathrm{d}s, \qquad t \in I. \tag{16}$$

Then applying the BRHQ rule (10) to the integral part of the latter at the point t_m yields

$$y'_m = g'(t_m) + \widetilde{F}_{1,m} + h\sum_{k=0}^{m} w^{(0)}_{m,k}\widetilde{F}_{2,k} + h^2\sum_{k=0}^{m} w^{(1)}_{m,k}\widetilde{F}'_{2,k}, \qquad m = n+1, \ldots, N, \tag{17}$$

where y'_m denotes the approximation of the exact value y' of (2) at t_m, and the $\widetilde{F}_{2,k}$ and $\widetilde{F}'_{2,k}$ stem from the function F_2, defined by $F_2(s, y(s)) = k_t(t_m, s, y(s))$, with k_t denoting the partial derivative of the kernel k with respect to t. Given this function, we let

$$\widetilde{F}_{2,k} = F_2(t_k, y_k),$$
$$\widetilde{F}'_{2,k} = F'_2(t_k, y_k) = k_{ts}(t_m, t_k, y_k) + k_{ty}(t_m, t_k, y_k)y'_k,$$

with k_{ts} and k_{ty} denoting the second-order partial derivatives of k with respect to t, s and t, y, respectively. In order to obtain the approximations, the nonlinear algebraic equations (15) and (17) must be solved simultaneously at each step.

To implement the method, we further need a set of initial values that guarantee the adequate accuracy of the approximations. To obtain them, we use the BRHQ rule (10) to approximate the integral part of (2) and (16) from t_0 to t_m, $m = 1, 2, \ldots, n$, which gives

$$\begin{aligned} y_m &= g(t_m) + h\sum_{k=0}^{n} \bar{w}^{(0)}_{m,k}\widetilde{F}_{1,k} + h^2\sum_{k=0}^{n} \bar{w}^{(1)}_{m,k}\widetilde{F}'_{1,k}, \\ y'_m &= g'(t_m) + \widetilde{F}_{1,m} + h\sum_{k=0}^{n} \bar{w}^{(0)}_{m,k}\widetilde{F}_{2,k} + h^2\sum_{k=0}^{n} \bar{w}^{(1)}_{m,k}\widetilde{F}'_{2,k}, \end{aligned} \tag{18}$$

where

$$\bar{w}^{(0)}_{m,k} = \int_0^m \left(\frac{2\beta_k\,\bar{\nu}_{k,1}}{x-k} + \frac{\beta_k^2}{(x-k)^2}\right) \Bigg/ \sum_{j=0}^{n}\left(\frac{2\beta_j\,\bar{\nu}_{j,1}}{x-j} + \frac{\beta_j^2}{(x-j)^2}\right)\mathrm{d}x,$$
$$\bar{w}^{(1)}_{m,k} = \int_0^m \frac{\beta_k^2}{x-k} \Bigg/ \sum_{j=0}^{n}\left(\frac{2\beta_j\,\bar{\nu}_{j,1}}{x-j} + \frac{\beta_j^2}{(x-j)^2}\right)\mathrm{d}x,$$

with barycentric weights β_j that depend on n, but not on m. It should be noted that the system (18) includes $2n$ equations in the $2n$ unknowns y_m and y'_m, $m = 1, 2, \ldots, n$, which must be solved simultaneously. The next theorem establishes the convergence rate of the initial values obtained from (18) in terms of the parameters of the method.

Theorem 4. *Assume that $g \in C^{2(d+2)}(I)$ and $k \in C^{2(d+2)}(S \times \mathbb{R}^D)$, that n and d with $d \leqslant n$ are respectively positive and non-negative integers, and let $\boldsymbol{e}_n = \left(\boldsymbol{e}_n^{(0)}, \boldsymbol{e}_n^{(1)}\right)^T$, where $\boldsymbol{e}_n^{(0)} = \left(e_1^{(0)}, \ldots, e_n^{(0)}\right)^T$ and $\boldsymbol{e}_n^{(1)} = (e_1^{(1)}, \ldots, e_n^{(1)})^T$ are the initial errors with $e_i^{(0)} = y(t_i) - y_i$ and $e_i^{(1)} = y'(t_i) - y'_i$, $i = 1, 2, \ldots, n$. Then, $\|\boldsymbol{e}_n\|_\infty$ tends to zero at the rate $O(h^{2(d+\delta)+3})$, where $\delta = 0$ for even values of $n-d$ and $\delta = 1$ for odd values of $n-d$.*

Proof. Substituting the exact values for y and y' in the initial value procedure (18) and including the consistency errors $R_{0,m}(h)$ and $R_{1,m}(h)$ gives

$$\begin{aligned} y(t_m) &= g(t_m) + h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} F_1(t_k, y(t_k)) + h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} F_1'(t_k, y(t_k)) + R_{0,m}(h), \\ y'(t_m) &= g'(t_m) + F_1(t_m, y(t_m)) + h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} F_2(t_k, y(t_k)) + h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} F_2'(t_k, y(t_k)) + R_{1,m}(h). \end{aligned} \tag{19}$$

Further substituting $t = t_m$ in (2) and (16) and subtracting from (19) yields

$$\begin{aligned} R_{0,m}(h) &= \int_{t_0}^{t_m} k(t_m, s, y(s))\,\mathrm{d}s - h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} F_1(t_k, y(t_k)) - h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} F_1'(t_k, y(t_k)), \\ R_{1,m}(h) &= \int_{t_0}^{t_m} k_t(t_m, s, y(s))\,\mathrm{d}s - h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} F_2(t_k, y(t_k)) - h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} F_2'(t_k, y(t_k)). \end{aligned}$$

Again, using the change of variable $t = a + xh$, $x \in [0, m]$ and taking the maximum norm, we find that the consistency errors are bounded by Ch^{2d+3} where C is a constant that depends only on d, the derivatives of f, and the interval length. It must be noted that, in view of the reasons mentioned in the proof of Theorem 3, if $n - d$ is odd, then the order of convergence for the initial values is two units larger than stated before, that is, $2(d + \delta) + 3$, where δ is the same quantity as stated in Theorem 3.

Subtracting the initial values in (18) from (19) and using the mean value theorem gives

$$\begin{aligned} e_m &= h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} k_y(t_m, t_k, \xi_{0,k}) e_k \\ &\quad + h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} \big(k_{sy}(t_m, t_k, \eta_{0,k}) e_k + k_{yy}(t_m, t_k, \zeta_{0,k}) y'(t_k) e_k + k_y(t_m, t_k, y_k) e_k' \big) \\ &\quad + R_{0,m}(h), \qquad m = 1, 2, \ldots, n, \\ e_m' &= k_y(t_m, t_m, \xi_m) + h\sum_{k=0}^{n} \bar{w}_{m,k}^{(0)} k_{ty}(t_m, t_k, \xi_{1,k}) e_k \\ &\quad + h^2 \sum_{k=0}^{n} \bar{w}_{m,k}^{(1)} \big(k_{tsy}(t_m, t_k, \eta_{1,k}) e_k + k_{tyy}(t_m, t_k, \zeta_{1,k}) y'(t_k) e_k + k_{ty}(t_m, t_k, y_k) e_k' \big) \\ &\quad + R_{1,m}(h), \qquad m = 1, 2, \ldots, n, \end{aligned}$$

where $\xi_{i,k}$, $\eta_{i,k}$, $\zeta_{i,k}$, $i = 0, 1$, and ξ_m are the interior points of the line segments connecting the exact value of y and its approximation at the corresponding functions. These equations can be written in block matrix form as

$$(\mathcal{D}_n - h\mathcal{W}_n)\boldsymbol{e}_n = \boldsymbol{R}_n(h),$$

where

$$\mathcal{D}_n = \begin{pmatrix} \mathcal{I}_n & 0 \\ -\mathcal{W}_n^{(2)} & \mathcal{I}_n \end{pmatrix}, \qquad \mathcal{W}_n = \begin{pmatrix} \mathcal{W}_n^{(0)} + h\mathcal{W}_n^{(1)} & h\overline{\mathcal{W}}_n^{(0)} \\ \mathcal{W}_n^{(3)} + h\mathcal{W}_n^{(4)} & h\overline{\mathcal{W}}_n^{(1)} \end{pmatrix},$$

with $\mathcal{I}_n$ denoting the $n \times n$ identity matrix and the other matrices being defined as

$$\begin{aligned} \mathcal{W}_n^{(0)} &= \big(\bar{w}_{i,j}^{(0)} k_y(t_i, t_j, \xi_{0,j})\big), & \mathcal{W}_n^{(1)} &= \big(\bar{w}_{i,j}^{(1)} \big(k_{sy}(t_i, t_j, \eta_{0,j}) + k_{yy}(t_i, t_j, \zeta_{0,j}) y'(t_j)\big)\big), \\ \mathcal{W}_n^{(2)} &= \operatorname{diag}\big(k_y(t_i, t_i, \xi_i)\big), & \mathcal{W}_n^{(3)} &= \big(\bar{w}_{i,j}^{(0)} k_{ty}(t_i, t_j, \xi_{1,j})\big), \\ \mathcal{W}_n^{(4)} &= \big(\bar{w}_{i,j}^{(1)} \big(k_{tsy}(t_i, t_j, \eta_{1,j}) + k_{tyy}(t_i, t_j, \zeta_{1,j}) y'(t_j)\big)\big), \\ \overline{\mathcal{W}}_n^{(0)} &= \big(\bar{w}_{i,j}^{(1)} k_y(t_i, t_j, y_j)\big), & \overline{\mathcal{W}}_n^{(1)} &= \big(\bar{w}_{i,j}^{(1)} k_{ty}(t_i, t_j, y_j)\big), \end{aligned}$$

and

$$\boldsymbol{R}_n(h) = \big(\boldsymbol{R}_n^{(0)}(h), \boldsymbol{R}_n^{(1)}(h)\big)^T,$$

where $\boldsymbol{R}_n^{(0)}(h) = \big(R_{0,1}(h), \ldots, R_{0,n}(h)\big)^T$ and $\boldsymbol{R}_n^{(1)}(h) = \big(R_{1,1}(h), \ldots, R_{1,n}(h)\big)^T$.

The differentiability hypothesis on the kernel k in (2) indicates that the corresponding functions and their partial derivatives in the aforementioned matrices have a maximum absolute value. Furthermore, since n is fixed in the initial value procedure (18) and only the stepsize h is variable, the initial quadrature weights have a maximum absolute value as well. This implies that the norm of the matrix $\mathcal{W}_n$ is bounded and can be made as small as necessary by reducing the stepsize h. Therefore, since the block matrix $\mathcal{D}_n$ is nonsingular, there exists, for sufficiently small h, a positive constant C such that

$$\|\boldsymbol{e}_n\|_\infty = \|(\mathcal{D}_n - h\mathcal{W}_n)^{-1}\|_\infty \|\boldsymbol{R}_n(h)\|_\infty \leqslant \frac{1}{\|\mathcal{D}_n\|_\infty - h\|\mathcal{W}_n\|_\infty} \|\boldsymbol{R}_n(h)\|_\infty \leqslant C\|\boldsymbol{R}_n(h)\|_\infty,$$

which implies $\|\boldsymbol{e}_n\|_\infty = O(h^{2(d+\delta)+3})$. □

The next theorem states the order of convergence of the GBRHQM, which can be obtained in a way that is similar to the one used in Theorem 7.2 in [44] by considering certain conditions on the functions g and k.

Theorem 5. *Assume that $g \in C^{2(d+2)}(I)$ and $k \in C^{2(d+2)}(S \times \mathbb{R}^D)$, that n and d with $d \leqslant n$ are respectively positive and non-negative integers, and let the nodes be equispaced. Then, the approximate solution of (2) obtained by the global method with the initial value procedure (18) converges at the rate $O(h^{2d+2})$, if the utilized initial errors tend to zero as h decreases.*

5.2 The composite version

As mentioned before, the implementation of the GBRHQM for solving time-dependent problems implies considerable computational cost; since the corresponding quadrature weights must be calculated at each step, and as the number of nodes increases, the number of terms in the BRHQ rule (10) becomes larger and larger. In order to avoid computing new quadrature weights for every step, the CBRHQ rule can be considered instead. Let T_N be again a uniform partition of the interval I and assume that d and n are as introduced above. Applying the CBRHQ rule (14) to the integrals in (2) and (16) at the mesh point t_m yields

$$\begin{aligned}
y_m &= g(t_m) + \sum_{j=0}^{p-1}\sum_{k=0}^{n}\left(h\,w_{n,k}^{(0)}\widetilde{F}_{1,jn+k} + h^2 w_{n,k}^{(1)}\widetilde{F}'_{1,jn+k}\right) \\
&\quad + \sum_{k=0}^{m-pn}\left(h\,w_{m-pn,k}^{(0)}\widetilde{F}_{1,pn+k} + h^2 w_{m-pn,k}^{(1)}\widetilde{F}'_{1,pn+k}\right), \quad m = n+1,\ldots,N, \\
y'_m &= g'(t_m) + \widetilde{F}_{1,m} + \sum_{j=0}^{p-1}\sum_{k=0}^{n}\left(h\,w_{n,k}^{(0)}\widetilde{F}_{2,jn+k} + h^2 w_{n,k}^{(1)}\widetilde{F}'_{2,jn+k}\right) \\
&\quad + \sum_{k=0}^{m-pn}\left(h\,w_{m-pn,k}^{(0)}\widetilde{F}_{2,pn+k} + h^2 w_{m-pn,k}^{(1)}\widetilde{F}'_{2,pn+k}\right), \quad m = n+1,\ldots,N,
\end{aligned} \tag{20}$$

where $p = \lfloor m/n \rfloor - 1$. As before, utilizing the composite method as a global one necessitates a procedure for obtaining initial values with adequate accuracy. So, the procedure (18) can be used again to provide suitable approximations of the initial values.

The convergence rate of the CBRHQM (20) can easily be deduced with the same ingredients as in Theorem 5 and the help of Theorems 3 and 4.

Theorem 6. *Assume that $g \in C^{2(d+2)}(I)$ and $k \in C^{2(d+2)}(S \times \mathbb{R}^D)$, that n and d with $d \leqslant n$ are respectively positive and non-negative integers, and let the nodes be equispaced. Then, the absolute errors of the CBRHQM (20) tend to zero at the rate $O(h^{2(d+\delta)+2})$, where $\delta = 0$ for even values of $n-d$, and $\delta = 1$ for odd values of $n-d$, if the errors of the initial values from procedure (18) vanish as the stepsize h is reduced.*

6 Numerical experiments with the LBRHQM

To confirm the theoretical orders of convergence derived in Section 5, we shall now examine the efficiency of the aforementioned methods and the initial value procedure, as well as the accuracy of the approximations by applying them with various choices of n and d to several VIEs. For every example, the approximation quality is measured by computing $e_h^S = \max_{1\leqslant m\leqslant n}\|y(t_m) - y_m\|_\infty$, the maximum norm of the errors at

Table 6: Numerical results of the CBRHQM applied to the nonlinear VIE in (21).

h		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
$(n,d,d_s)=(2,0,0)$	$e_h^C(T)$	7.64×10^{-4}	1.73×10^{-4}	2.72×10^{-5}	7.21×10^{-6}	1.94×10^{-6}	4.99×10^{-7}
	O_C		2.14	2.67	1.92	1.89	1.96
$(n,d,d_s)=(5,1,1)$	$e_h^C(T)$	8.13×10^{-3}	2.59×10^{-5}	3.67×10^{-7}	4.36×10^{-8}	2.88×10^{-9}	1.68×10^{-10}
	O_C		8.29	6.14	3.07	3.92	4.10
$(n,d,d_s)=(11,3,2)$	$e_h^C(T)$	1.97×10^{-2}	5.59×10^{-6}	2.15×10^{-9}	6.81×10^{-12}	3.31×10^{-14}	2.13×10^{-14}
	O_C		11.78	11.34	8.30	7.68	0.64

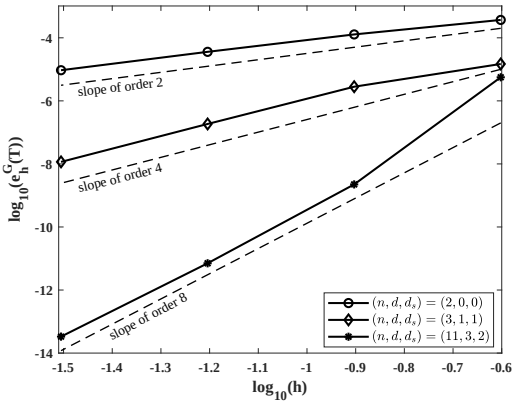

Figure 3: Log-log-plot of the approximation error of the GBRHQM applied to the nonlinear VIE in (21).

the initial values and $e_h(T)=\|y(t_N)-y_N\|_\infty$, the maximum norm of the error at $t_N=T$, the extremity of the integration interval. In the following numerical experiments, the superscripts G and C in $e_h(T)$ are used to refer to the errors of the GBRHQM and the CBRHQM, respectively. Moreover, the initial values are computed with the parameter d_s, and the experimental approximation orders for the initial value procedure, the GBRHQM, and the CBRHQM are respectively computed by

$$O_S=\log_2\big(e_h^S/e_{h/2}^S\big),\qquad O_G=\log_2\big(e_h^G(T)/e_{h/2}^G(T)\big),\qquad O_C=\log_2\big(e_h^C(T)/e_{h/2}^C(T)\big).$$

As a first example, consider the nonlinear VIE

$$y(t)=1+\int_0^t (t-s)^3(4-t+s)e^{-t+s}\frac{y^4(s)}{1+2y^2(s)+2y^4(s)}\,\mathrm{d}s,\qquad t\in[0,10], \tag{21}$$

arising in the analysis of neural networks with post-inhibitory rebound [20, 34]. A reference solution at the endpoint $T=10$ of the integration interval is given by $y(10)=1.25995582337233$. The numerical results of the CBRHQM for this equation are reported in Table 6 for $(n,d,d_s)=(2,0,0)$, $(5,1,1)$, and $(11,3,2)$. According to Theorem 6, the expected orders of convergence are 2, 4, and 8, respectively, which are confirmed by the numerical results. Moreover, Figure 3 shows the logarithmic error $\log_{10}(e_h^G(T))$ of the GBRHQM for this equation, plotted versus $\log_{10}(h)$, together with the slope lines corresponding to the expected convergence rates. These numerical results confirm the expected orders of convergence for various choices of (n,d) as predicted by Theorem 5. In the case of $(n,d,d_s)=(11,3,2)$, the approximation reaches machine precision for $h=2^{-5}$, which explains some smaller experimental orders in the last column of the table.

Since the order of convergence of the CBRHQM is at least $2d+2$ and the cost of its computation is significantly less than for the global method, we investigate just the composite version for the following examples.

To demonstrate the efficiency and accuracy of the CBRHQM, consider the highly oscillatory VIE

$$y(t)=g(t)-\int_{-1}^t \cos(\omega(s-t))y(s)\,\mathrm{d}s,\qquad t\in[-1,1], \tag{22}$$

Table 7: Numerical results of the CBRHQM applied to the oscillatory VIE in (22) with $\omega = 100$.

h		2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}
$(n,d,d_s) = (1,0,0)$							
Initial values	e_h^S	3.10×10^{-5}	1.34×10^{-6}	4.51×10^{-8}	1.44×10^{-9}	4.52×10^{-11}	1.42×10^{-12}
	O_S		4.53	4.89	4.97	5.00	5.00
CBRHQM	$e_h^C(T)$	1.82×10^{-5}	1.05×10^{-6}	6.44×10^{-8}	4.01×10^{-9}	2.50×10^{-10}	1.56×10^{-11}
	O_C		4.12	4.03	4.01	4.00	4.00
$(n,d,d_s) = (4,2,2)$							
Initial values	e_h^S	2.29×10^{-6}	4.70×10^{-8}	6.34×10^{-10}	7.08×10^{-12}	6.03×10^{-14}	5.00×10^{-16}
	O_S		5.61	6.21	6.48	6.88	6.91
CBRHQM	$e_h^C(T)$	1.56×10^{-4}	3.49×10^{-8}	4.67×10^{-10}	7.06×10^{-12}	1.10×10^{-13}	1.78×10^{-15}
	O_C		12.13	6.22	6.05	6.00	5.95

Table 8: Numerical results of the CBRHQM applied to the oscillatory VIE in (22) with $\omega = 1000$.

h		2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}
$(n,d,d_s) = (1,0,0)$							
Initial values	e_h^S	2.98×10^{-3}	1.32×10^{-4}	7.54×10^{-6}	3.93×10^{-7}	1.37×10^{-8}	4.39×10^{-10}
	O_S		4.50	4.13	4.26	4.84	4.96
CBRHQM	$e_h^C(T)$	9.85×10^{-4}	5.69×10^{-5}	2.58×10^{-6}	1.52×10^{-7}	9.33×10^{-9}	5.81×10^{-10}
	O_C		4.11	4.46	4.09	4.03	4.01
$(n,d,d_s) = (4,2,2)$							
Initial values	e_h^S	5.08×10^{-3}	1.41×10^{-4}	6.21×10^{-7}	1.82×10^{-8}	2.40×10^{-10}	3.27×10^{-12}
	O_S		5.17	7.83	5.09	6.24	6.20
CBRHQM	$e_h^C(T)$	4.21×10^{-2}	3.93×10^{-5}	1.69×10^{-7}	5.92×10^{-9}	7.90×10^{-11}	1.20×10^{-12}
	O_C		10.07	7.86	4.84	6.23	6.04

where

$$g(t) = e^t + \frac{1}{e(1+\omega^2)}\left(e^{t+1} - \cos(\omega(t+1)) + \omega\sin(\omega(t+1))\right),$$

with the exact solution $y(t) = e^t$. The numerical results for the initial value procedure and the CBRHQM are reported in Tables 7 and 8 for $(n,d,d_s) = (1,0,0)$ and $(n,d,d_s) = (4,2,2)$, different values of ω, and various choices of the stepsize h. As to be expected from Theorem 4, the error of the initial values decreases with order $2(d_s+\delta)+3$, with $\delta = 1$ for odd $n-d_s$ and $\delta = 0$ for even $n-d_s$, and the errors of the CBRHQM decrease with the order $2(d+\delta)+2$, with $\delta = 1$ for odd $n-d$ and $\delta = 0$ for even $n-d$, as predicted by Theorem 6.

Next, consider the VIE

$$y(t) = g(t) + \int_0^t e^s \cos(\omega s)\, y(s)\, \mathrm{d}s, \qquad t \in [0,1], \tag{23}$$

where the function g is chosen such that the exact solution is $y(t) = \cos(t)$. Figure 4 shows the logarithmic error $\log_{10}(e_h^C(T))$ of the CBRHQM for this equation, plotted versus $\log_{10}(h)$, together with the expected convergence rates. Again, the numerical results for various choices of (n,d,d_s) confirm the orders expected from the theory.

Finally, consider the nonlinear VIE

$$y(t) = e^{-t} + \int_0^t e^{s-t}\left(y(s) + e^{-y(s)}\right)\mathrm{d}s, \qquad t \in [0,40], \tag{24}$$

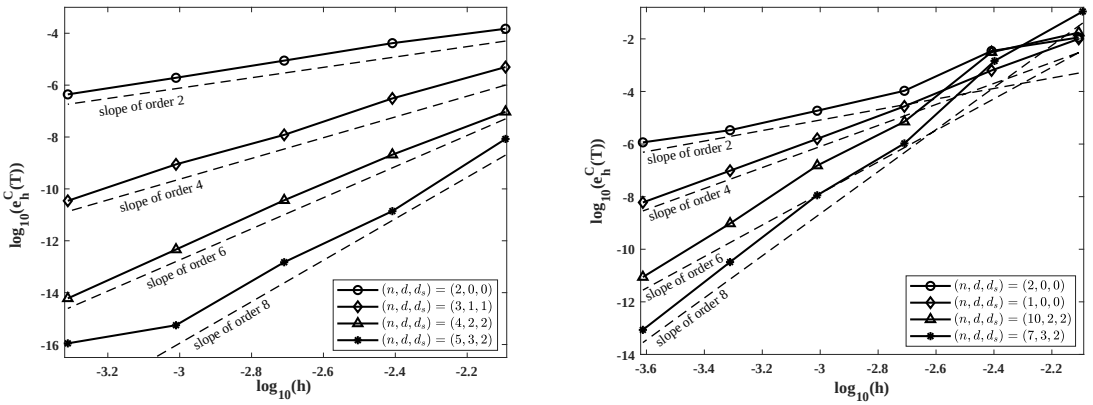

Figure 4: Log-log-plots of the approximation error of the CBRHQM applied to the VIE in (23) with $\omega = 100$ (left) and $\omega = 1000$ (right).

Table 9: Numerical results of the CBRHQM applied to the nonlinear VIE in (24).

h		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
$(n,d,d_s)=(5,1,1)$							
Initial values	e_h^S	2.60×10^{-4}	6.39×10^{-6}	1.96×10^{-7}	6.23×10^{-9}	1.98×10^{-10}	6.25×10^{-12}
	O_S		5.35	5.03	4.98	4.98	4.99
CBRHQM	$e_h^C(T)$	2.43×10^{-3}	1.34×10^{-4}	9.01×10^{-6}	6.24×10^{-7}	4.19×10^{-8}	2.73×10^{-9}
	O_C		4.18	3.89	3.85	3.90	3.94
$(n,d,d_s)=(8,2,2)$							
Initial values	e_h^S	1.38×10^{-5}	4.04×10^{-8}	2.80×10^{-10}	2.21×10^{-12}	1.80×10^{-14}	2.22×10^{-16}
	O_S		8.42	7.17	6.99	6.94	6.34
CBRHQM	$e_h^C(T)$	3.15×10^{-5}	3.78×10^{-7}	5.96×10^{-9}	1.09×10^{-10}	1.96×10^{-12}	3.02×10^{-14}
	O_C		6.38	5.99	5.77	5.80	6.02

where the exact solution is $y(t) = \log(t+e)$. The numerical results for this equation with parameters $(n,d,d_s) = (5,1,1)$ and $(n,d,d_s) = (8,2,2)$ are given in Table 9 and confirm once more our theoretical results.

Implementation of the proposed methods becomes somewhat more challenging when dealing with stiff VIEs, which require the use of sufficiently small step sizes h. Further investigation into the stability behavior of the proposed methods in the present manuscript will be addressed in future work.

Acknowledgments

The results reported in this paper were obtained during the visit of the third author to Università della Svizzera italiana (USI), which was supported by the Swiss National Science Foundation (SNSF) under grant No. IZSEZ0_216752. The third author wishes to express his gratitude to K. Hormann for making this visit possible. The work of the first author was supported by the research grant of the University of Tabriz under research grant No. 771.

References

[1] A. Abdi, M. Arnold, H. Podhaisky, The barycentric rational numerical differentiation formulas for stiff ODEs and DAEs, Numer. Algorithms 97(1) (2024) 431–451.

[2] A. Abdi, J.-P. Berrut, S.A. Hosseini, Explicit methods based on barycentric rational interpolants for solving non-stiff Volterra integral equations, Appl. Numer. Math. 174 (2022) 127–141.

[3] A. Abdi, J.-P. Berrut, H. Podhaisky, The barycentric rational predictor–corrector schemes for Volterra integral equations, J. Comput. Appl. Math. 440 (2024) 115611.

[4] A. Abdi, D. Conte, Implementation of general linear methods for Volterra integral equations, J. Comput. Appl. Math. 386 (2021) 113261:1–12.

[5] A. Abdi, S.A. Hosseini, The barycentric rational difference-quadrature scheme for systems of Volterra integro-differential equations, SIAM J. Sci. Comput. 40 (2018) A1936–A1960.

[6] A. Abdi, S.A. Hosseini, H. Podhaisky, The linear barycentric rational backward differentiation formulae for stiff ODEs on nonuniform grids, Numer. Algorithms 98 (2025) 877–902.

[7] A. Abdi, S.A. Hosseini, H. Podhaisky, Numerical methods based on the Floater–Hormann interpolants for stiff VIEs, Numer. Algorithms 85 (2020) 867–886.

[8] J.-P. Berrut, Rational functions for guaranteed and experimentally well-conditioned global interpolation, Comput. Math. Appl. 15 (1988) 1–16.

[9] J.-P. Berrut, M.S. Floater, G. Klein, Convergence rates of derivatives of a family of barycentric rational interpolants, Appl. Numer. Math. 61 (2011) 989–1000.

[10] J.-P. Berrut, S.A. Hosseini, G. Klein, The linear barycentric rational quadrature method for Volterra integral equations, SIAM J. Sci. Comput. 36 (2014) A105–A123.

[11] J.-P. Berrut, H.D. Mittelmann, Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval, Comput. Math. Appl. 33 (1997) 77–86.

[12] P. Bocher, H. De Meyer, G. Berghe, On Gregory- and modified Gregory-type corrections to Newton–Cotes quadrature, J. Comput. Appl. Math. 50 (1994) 145–158.

[13] L. Bos, S. De Marchi, K. Hormann, On the Lebesgue constant of Berrut's rational interpolant at equidistant nodes, J. Comput. Appl. Math. 236 (2011) 504–510.

[14] L. Bos, S. De Marchi, K. Hormann, G. Klein, On the Lebesgue constant of barycentric rational interpolation at equidistant nodes, Numer. Math. 121 (2012) 461–471.

[15] H. Brass, K. Petras, Quadrature Theory: The Theory of Numerical Integration on a Compact Interval, American Mathematical Society, Rhode Island, 2011.

[16] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Equations, Cambridge University Press, Cambridge, 2004.

[17] H. Brunner, Volterra Integral Equations: An Introduction to Theory and Applications, Cambridge University Press, Cambridge, 2017.

[18] H. Brunner, P.J. van der Houwen, The Numerical Solution of Volterra Equations, in: CWI Monogr., North-Holland, Amsterdam, 1986.

[19] J.G. Blom, H. Brunner, H., The numerical solution of nonlinear Volterra integral equations of the second kind by collocation and iterated collocation methods, SIAM J. Sci. Stat. Comput. 8 (1987) 806–830.

[20] G. Capobianco, D. Conte, I. Del Prete, E. Russo, Fast Runge–Kutta methods for nonlinear convolution systems of Volterra integral equations, BIT 47 (2007) 259–275.

[21] A. Cardone, L.G. Ixaru, B. Paternoster, Exponential fitting Direct Quadrature methods for Volterra integral equations, Numer. Algorithms 55 (2010) 467–480.

[22] E. Cirillo, K. Hormann, An iterative approach to barycentric rational Hermite interpolation, Numer. Math. 140 (2018) 939–962.

[23] E. Cirillo, K. Hormann, On the Lebesgue constant of barycentric rational Hermite interpolants at equidistant nodes, J. Comput. Appl. Math. 349 (2019) 292–301.

[24] E. Cirillo, K. Hormann, J. Sidon, Convergence rates of derivatives of Floater–Hormann interpolants for well-spaced nodes, Appl. Numer. Math. 116 (2017) 108–118.

[25] E. Cirillo, K. Hormann, J. Sidon, Convergence rates of a Hermite generalization of Floater–Hormann interpolants, J. Comput. Appl. Math. 371 (2020) 112624:1–9.

[26] D. Conte, Z. Jackiewicz, B. Paternoster, Two-step almost collocation methods for Volterra integral equations, Appl. Math. Comput. 204 (2008) 839–853.

[27] D. Conte, and B. Paternoster, Multistep collocation methods for Volterra integral equations, Appl. Numer. Math. 59 (2009) 1721–1736.

[28] H. Chen and C. Zhang, Block boundary value methods for solving Volterra integral and integro-differential equations, J. Comput. Appl. Math. 236 (2012) 2822–2837.

[29] T.A. Driscoll, Automatic spectral collocation for integral, integro–differential, and integrally reformulated differential equations, J. Comput. Phys. 229 (2010) 5980–5998.

[30] G.A. Evans, Two robust methods for irregular oscillatory integrals over a finite range, Appl. Numer. Math. 14 (1994) 383–395.

[31] M.S. Floater, K. Hormann, Barycentric rational interpolation with no poles and high rates of approximation, Numer. Math. 107 (2007) 315–331.

[32] B. Fornberg, J.A. Reeger, An improved Gregory-like method for 1-D quadrature, Numer. Math. 141 (2019) 1–19.

[33] C. Fuda, R. Campagna, and K. Hormann, On the numerical stability of linear barycentric rational interpolation, Numer. Math. 152 (2022) 761–786.

[34] E. Hairer, C. Lubich, M. Schlichte, Fast numerical solution of nonlinear Volterra convolution equations, SIAM J. Sci. Stat. Comput. 6 (1985) 532–541.

[35] N. Hale, A. Townsend, Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights, SIAM J. Sci. Comput. 35 (2013) A652–A674.

[36] F.C. Hoppensteadt, Z. Jackiewicz, B. Zubik-Kowal, Numerical solution of Volterra integral and integro-differential equations with rapidly vanishing convolution kernels, BIT 47 (2007) 325–350.

[37] G. Izzo, Z. Jackiewicz, E. Messina, A. Vecchio, General linear methods for Volterra integral equations, J. Comput. Appl. Math. 234 (2010) 2768–2782.

[38] M. Javed, L.N. Trefethen, Euler–Maclaurin and Gregory interpolants, Numer. Math. 132 (2016) 201–216.

[39] G. Klein, J.-P. Berrut, Linear barycentric rational quadrature, BIT 52 (2012) 407–424.

[40] G. Klein, J.-P. Berrut, Linear rational finite differences from derivatives of barycentric rational interpolants, SIAM J. Numer. Anal. 50 (2012) 643–656.

[41] J. Li, Y. Cheng, Linear barycentric rational collocation method for solving heat conduction equation, Numer. Methods Partial Differ. Equ. 37 (2021) 533–545.

[42] J. Li, Y. Cheng, Barycentric rational method for solving biharmonic equation by depression of order, Numer. Methods Partial Differ. Equ. 37 (2021) 1993–2007.

[43] M. Li, C. Huang, The linear barycentric rational quadrature method for auto-convolution Volterra integral equations, J. Sci. Comput. 78 (2019) 549–564.

[44] P. Linz, Analytical and Numerical Methods for Volterra Equations, SIAM, Philadelphia, 1985.

[45] H. Liu, J. Huang, X. He, Bivariate barycentric rational interpolation method for two dimensional fractional Volterra integral equations, J. Comput. Appl. Math. 389 (2021) 113339.

[46] W.-H. Luo, T.-Z. Huang, X.-M. Gu, Y. Liu, Barycentric rational collocation methods for a class of nonlinear parabolic partial differential equations, Appl. Math. Lett. 68 (2017) 13–19.

[47] J. Ma, A class of reducible quadrature rules for the second-kind Volterra integral equations using barycentric rational interpolation, J. Comput. Appl. Math. 445 (2024) 115803.

[48] C. Schneider, W. Werner, Some new aspects of rational interpolation, Math. Comput. 47 (1986) 285–299.

[49] C. Schneider, W. Werner, Hermite interpolation: The barycentric approach, Computing 46 (1991) 35–51.

[50] T. Tang, X. Xu, J. Cheng, On spectral methods for Volterra integral equations and the convergence analysis, J. Comput. Math. 26 (2008) 825–837.

[51] L.N. Trefethen, et al., Chebfun Version 5.6.0, The Chebfun Development Team, http://www.chebfun.org, 2016.

[52] L.N. Trefethen, Is Gauss quadrature better than Clenshaw–Curtis? SIAM Rev. 50 (2008) 67–87.

[53] P.J. van der Houwen, H.J.J. te Riele, Backward differentiation type formulas for Volterra integral equations of the second kind, Numer. Math. 37 (1981) 205–217.

[54] W. Werner, Polynomial interpolation: Lagrange versus Newton, Math. Comput. 43 (1984) 205–217.