

# Approximate solution of Volterra-Fredholm integral equations using generalized barycentric rational interpolant

Hadis Azin      Fakhrodin Mohammadi\*

**Abstract.** It is well-known that interpolation by rational functions results in a more accurate approximation than the polynomials interpolation. However, classical rational interpolation has some deficiencies such as uncontrollable poles and low convergence order. In contrast with the classical rational interpolants, the generalized barycentric rational interpolants which depend linearly on the interpolated values, yield infinite smooth approximation with no poles in real numbers. In this paper, a numerical collocation approach, based on the generalized barycentric rational interpolation and Gaussian quadrature formula, was introduced to approximate the solution of Volterra-Fredholm integral equations. Three types of points in the solution domain are used as interpolation nodes. The obtained numerical results confirm that the barycentric rational interpolants are efficient tools for solving Volterra-Fredholm integral equations. Moreover, integral equations with Runge's function as an exact solution, no oscillation occurs in the obtained approximate solutions so that the Runge's phenomenon is avoided.

## §1 Introduction

A simple method for approximating a given function  $f$  on the interval  $[a, b]$  is to use interpolation functions. Let  $a \leq x_0 < x_1 < \dots < x_n \leq b$  be a set of distinct points in the interval  $[a, b]$ . An interpolant is a function  $p$  which has the same value as  $f$  at these node points, i.e.,

$$p(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

Depending on the distribution of the node points and the requested accuracy, several types of interpolation functions have been used to approximate the function  $f$ . The polynomial interpolant is the most simple and the easiest form of interpolant functions frequently applied to approximate the solution of differential and integral equations [1–4]. However, the application

---

Received: 2019-01-01.      Revised: 2021-02-01.

MR Subject Classification: 35R11, 33C47, 65M70.

Keywords: Barycentric rational interpolation, Volterra-Fredholm integral equations, Gaussian quadrature, Runge's phenomenon.

Digital Object Identifier(DOI): <https://doi.org/10.1007/s11766-024-3711-x>.

\*Corresponding author.

of the polynomial interpolant has some significant shortcomings such as unstable numerical computations and extremely high computational cost [5, 6]. Moreover, for a large number of equidistant node points, polynomial interpolants exhibit large oscillations around the interval boundaries which is known as the Runge's phenomenon [6, 7]. There are some approaches that improve the performance of polynomial interpolant. If the distribution of node points can be chosen, the best option is to use the shifted Chebyshev points in the interval  $[a, b]$ . On the contrary, if the position of the interpolating points is fixed, the other kind of interpolant functions such as piecewise polynomials and rational functions must be used [6–8].

The classical rational interpolants deal with interpolation problems by using rational polynomial functions. Suppose that  $x_0 < x_1 < \dots < x_n$  are  $n + 1$  distinct real numbers and  $f$  is a given function. A rational interpolation function for  $f$  at these points is an irreducible rational function  $\Phi^{\mu, \nu}(x) = \frac{P^\mu(x)}{Q^\nu(x)}$  in which  $P^\mu(x)$  and  $Q^\nu(x)$  are two polynomials of degree at most  $\mu$  and  $\nu$  respectively, with  $n + 1 = \mu + \nu + 1$  and such that [9]

$$\Phi^{\mu, \nu}(x_i) = \frac{P^\mu(x_i)}{Q^\nu(x_i)} = f(x_i), \quad i = 0, 1, \dots, n. \quad (1)$$

It is well-known that the use of rational functions can also lead to much better approximation in comparison with the polynomial interpolants. Specifically, for the functions with a pole, rational interpolants are more effective than the polynomial interpolations and yield more accurate approximation. Nevertheless, there are some disadvantages to the practical implementation of the classical rational interpolation which can be described as:

- The interpolation problem (1) results in a complicated and high-cost procedure for computing rational function  $\Phi^{\mu, \nu}$ .
- Contrary to the polynomial interpolation, the rational interpolation problem may be unsolvable, and result in unattainable points.
- The rational interpolant  $\Phi^{\mu, \nu}$  can have some poles in the interval of interpolation.

To improve the application of the classical rational interpolants, a variant representation of rational interpolant, called the barycentric form, was suggested by Berrut and Mittelmann [10] and later extended by Berrut and Trefethen [6]. The barycentric formulation of the rational interpolation avoids poles by using the numerator and denominator of higher degrees. Moreover, the barycentric rational interpolants depend linearly on the data  $f(x_i)$  which leads to low-cost and efficient algorithms for computing them. More recently, Floater and Hormann [7] have introduced a family of generalized barycentric rational interpolants that provide an arbitrary high approximation order and have no real poles in the interval of interpolation. It also has been proved that this generalized formulation avoids oscillations such as Runge's phenomenon [7].

The modeling of such phenomena in many areas of science and engineering such as fluid mechanics, quantum mechanics, water waves, biological systems, kinetics in chemistry and spatial-temporal development lead to Volterra-Fredholm integral equations. Such important applications have motivated researchers to investigate analytical and numerical solutions of this kind of integral equations. Recently, wavelet method [11–13], polynomials spectral methods

[1, 14–18], Least squares approximation method [19], operational methods [20], hybrid functions [21], Sinc Nyström method [22] and modified block-pulse functions [23],  $\varepsilon$ -SVR method [24], have been used to approximate the solution of Volterra-Fredholm integral equations.

In this paper, the generalized barycentric rational interpolant introduced by Floater and Hormann [7] was taken into account and their features were reviewed thoroughly. Then, a numerical collocation approach based on these generalized barycentric rational interpolant was introduced to approximate the solution of the following Volterra-Fredholm integral equations

$$\alpha(x)y(x) + \beta(x)y(h(x)) = f(x) + \mu_1 \int_a^{h(x)} k_1(x, t)y(t)dt + \mu_2 \int_a^b k_2(x, t)y(h(t)) dt, \quad (2)$$

where  $k_i(x, t)$ ,  $i = 1, 2$  are known kernel functions on the interval  $[a, b] \times [a, b]$ . The known functions  $\alpha(x)$ ,  $\beta(x)$ ,  $f(x)$  and  $0 \leq h(x) < \infty$  are defined for  $a \leq x \leq b$  and  $y(x)$  is the unknown function. Moreover, and  $\mu_1$ ,  $\mu_2$  are real constants and  $\mu_1^2 + \mu_2^2 \neq 0$ . Three kind of points in the interval  $[a, b]$  are used as interpolation nodes in the proposed method. The convergence analysis of the proposed collocation method is also given. The performance of the proposed scheme is validated through some illustrative examples.

## §2 Barycentric rational interpolant

For  $(n + 1)$  distinct points  $a \leq x_0 < x_1 < \dots < x_n \leq b$  and continuous function  $f$  on the interval  $[a, b]$ , the barycentric formulation of the rational interpolant is defined as [10]:

$$\phi(x) = \frac{\sum_{i=0}^n \frac{w_i f(x_i)}{(x-x_i)}}{\sum_{i=0}^n \frac{w_i}{(x-x_i)}} \quad (3)$$

where  $w_0, \dots, w_n$  are real valued weights and must be chosen such that  $\phi(x)$  is pole free interpolant for the function  $f$  in the interval  $[a, b]$ . Recently, different kinds of weights have been chosen in the barycentric rational formulation (3). For example, the barycentric formulation of the Lagrange interpolation are investigated in Ref. [6]. Indeed, the classical Lagrange polynomial interpolant for the function  $f$  can be presented in barycentric form (3) by defining the weights  $w_i$  as:

$$w_i = \frac{1}{\prod_{j=0, j \neq i}^n (x_j - x_i)}, \quad i = 0, 1, \dots, n.$$

More appropriate weights for the barycentric rational formulation (3) were considered by Berrut as follows [5]:

$$w_i = (-1)^i, \quad i = 0, 1, \dots, n,$$

which yield the following rational interpolation function

$$\phi(x) = \frac{\sum_{i=0}^n \frac{(-1)^i f(x_i)}{(x-x_i)}}{\sum_{i=0}^n \frac{(-1)^i}{(x-x_i)}}. \quad (4)$$

It has been shown that the rational interpolant (4) has no poles in real number  $\mathbb{R}$ . Moreover, it has been applied to approximate the Runge’s function  $f(x) = \frac{1}{1+25x^2}$  for several distributions of node points  $x_i$  and the obtained numerical results confirm converges to  $f$  at the rate of  $O(\frac{1}{n})$  [5, 7].

### §3 Floater-Hormann rational interpolant

In 2007, a generalized version of the barycentric rational interpolant was introduced by Floater and Hormann [7] which had some advantages over the Berrut’s barycentric form (3). This family of barycentric rational interpolant provides infinitely smooth functions that have arbitrary high approximation order for any distribution of node points [25]. For a given function  $f : [a, b] \rightarrow \mathbb{R}$ , arbitrary nodes  $a \leq x_0 < x_1 < \dots < x_{n-1} < x_n \leq b$  and  $0 \leq d \leq n$ , the Floater-Hormann barycentric rational interpolant is defined as

$$r(x) = \frac{\sum_{i=0}^{n-d} \lambda_i(x)p_i(x)}{\sum_{i=0}^{n-d} \lambda_i(x)} \tag{5}$$

in which  $p_i(x)$  is the polynomial interpolation of degree at most  $d$  for the points  $x_i, x_{i+1}, \dots, x_{i+d}$ , and

$$\lambda_i(x) = \frac{(-1)^i}{(x - x_i)(x - x_{i+1}) \dots (x - x_{i+d})}.$$

By definition of the Floater-Hormann rational interpolant (5) we can realize that the  $\lambda_i$  only depends on the node points  $x_i$ . Therefore,  $r$  is the linear function of the values  $f(x_i)$  and by using the well-known Lagrange polynomials it can be written in the following form:

$$r(x) = \frac{1}{\sum_{i=0}^{n-d} \lambda_i(x)} \left( \sum_{i=0}^{n-d} \lambda_i(x) \sum_{j=i}^{i+d} f(x_j) L_{ij}(x) \right) = \sum_{i=0}^n f(x_i) \psi_i(x), \tag{6}$$

where  $L_{ij}$  is the Lagrange basis polynomial

$$L_{ij}(x) = \prod_{l=0, l \neq j}^{n+d} \frac{x - x_l}{x_j - x_l}, \quad i = 0, 1, \dots, n - d, \quad j = i, i + 1, \dots, i + d,$$

and

$$\psi_i(x) = \begin{cases} \frac{\sum_{j=0}^i \lambda_j(x)L_{ij}(x)}{\sum_{i=0}^{n-d} \lambda_i(x)}, & i = 0, 1, \dots, d, \\ \frac{\sum_{j=i-d}^i \lambda_j(x)L_{ij}(x)}{\sum_{i=0}^{n-d} \lambda_i(x)} & i = d + 1, \dots, n - d, \\ \frac{\sum_{j=i-d}^{n-d} \lambda_j(x)L_{ij}(x)}{\sum_{i=0}^{n-d} \lambda_i(x)} & i = n - d + 1, \dots, n. \end{cases} \tag{7}$$

**Remark 1.** It is easy to see that for  $d = 0$ , the Floater-Hormann rational interpolant  $r$  is exactly the Berrut's interpolant as defined in (4).

Now, let's review some fundamental properties of the generalized Floater-Hormann rational interpolant. For more details on the Floater-Hormann rational interpolant and their special features can refer to Refs. [7, 25, 26].

### 3.1 The barycentric formulation

Consider the Lagrange form of the polynomial interpolant  $p_i(x)$  as follows:

$$p_i(x) = \sum_{i=0}^{n-d} \left( \prod_{j=0, j \neq i}^{n-d} \frac{x - x_j}{x_i - x_j} \right) f(x_i), \quad i = 0, 1, \dots, n. \tag{8}$$

The Floater-Hormann rational interpolant  $r(x)$  can be rewritten as:

$$r(x) = \frac{\sum_{i=0}^n \frac{w_i f(x_i)}{(x-x_i)}}{\sum_{i=0}^n \frac{w_i}{(x-x_i)}}$$

where

$$w_k = \sum_{i \in J_k} (-1)^i \prod_{j=i, j \neq k}^{i+d} \frac{1}{x_k - x_j}, \quad J_k = \{i : 0 \leq i \leq n - d \text{ and } k - d \leq i \leq k\}.$$

### 3.2 Rate of approximation

Let  $\| \cdot \|_\infty$  be the infinity norm as  $\|u\|_\infty = \max_{a \leq x \leq b} |u(x)|$ . The following Theorems from Ref. [7] provide the approximation order of the Floater-Hormann rational interpolants  $r$ .

**Theorem 1.** [7] Let  $1 \leq d$  and  $f \in C^{d+2}[a, b]$ , then

$$\|r - f\|_\infty \leq \begin{cases} h^{d+1} (b - a) \frac{\|f^{(d+2)}\|_\infty}{d+2}, & \text{for odd } n - d, \\ h^{d+1} \left( (b - a) \frac{\|f^{(d+2)}\|_\infty}{d+2} + \frac{\|f^{(d+1)}\|_\infty}{d+1} \right), & \text{for even } n - d, \end{cases}$$

where  $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|$ .

**Theorem 2.** [7] Let  $f \in C^2[a, b]$ . Then for  $d = 0$  the approximation error for the rational interpolant (5) can be derived as

$$\|r - f\|_\infty \leq \begin{cases} h(1 + \rho) (b - a) \frac{\|f''\|_\infty}{2}, & \text{for odd } n, \\ h(1 + \rho) \left( \frac{(b-a)\|f''\|_\infty}{2} + \|f'\|_\infty \right), & \text{for even } n, \end{cases}$$

where  $h = \max_{0 \leq i \leq n-1} |x_{i+1} - x_i|$  and

$$\rho = \max_{1 \leq i \leq n-2} \min \left\{ \frac{x_{i+1} - x_i}{x_i - x_{i-1}}, \frac{x_{i+1} - x_i}{x_{i+2} - x_{i+1}} \right\}.$$

The above Theorems reveal that for a enough smooth function  $f$  and any distribution of node points  $x_i$ , the Floater-Hormann rational interpolant  $r$  convergences to  $f$  with the rate of  $O(h^{d+1})$ .

### 3.3 Absence of poles

An interesting advantage of the Floater-Hormann rational interpolant  $r$  is that it is free of real poles.

**Theorem 3.** [7] For  $0 \leq d \leq n$ , the Floater-Hormann rational interpolant (5) has no real poles.

## §4 Application in solving VFIEs

The main aim of this section is to present a numerical approach based on the Floater-Hormann rational interpolant for solving VFIEs (2). It is worth mentioning that the known functions in Eq. (2) satisfy the conditions that this equation has a unique solution [27, 28]. First, we consider  $(n + 1)$  distinct node points in the interval  $[a, b]$ . In our approach, three types of node points are considered:

**Type 1:** A uniform partition (UP) on the interval  $[a, b]$ :

$$x_0 = a, \quad x_i = x_{i-1} + h, \quad , i = 1, 2, \dots, n, \quad h = \frac{b - a}{n}. \tag{9}$$

**Type 2:** The shifted Chebyshev Gauss (CG) points in the interval  $[a, b]$ :

$$x_i = \frac{b + a}{2} - \frac{b - a}{2} \cos\left(\frac{(2i + 1)\pi}{2n + 2}\right), \quad i = 0, 1, \dots, n. \tag{10}$$

**Type 3:** The shifted Chebyshev Gauss-Lobatto (CGL) points in the interval  $[a, b]$ :

$$x_i = \frac{b + a}{2} - \frac{b - a}{2} \cos\left(\frac{i\pi}{n}\right), \quad i = 0, 1, \dots, n. \tag{11}$$

Now, let  $y_i = y(x_i)$  be the value of the unknown function  $y(x)$  at the prespecified node points  $x_i$ . The function  $y(x)$  and  $y(h(x))$  can be approximated by means of the Floater-Hormann rational interpolant (6) as

$$y(x) \simeq y_n(x) = \sum_{i=0}^n y_i \psi_i(x), \quad x \in [a, b]. \tag{12}$$

$$y(h(x)) \simeq y_n(h(x)) = \sum_{i=0}^n y_i \psi_i(h(x)), \quad x \in [a, b],$$

Substitute these approximate functions into relation (2), we get:

$$\begin{aligned} \alpha(x) \sum_{i=0}^n y_i \psi_i(x) + \beta(x) \sum_{i=0}^n y_i \psi_i(h(x)) &= f(x) + \mu_1 \sum_{i=0}^n y_i \int_a^{h(x)} k_1(x, t) \psi_i(t) dt \\ &+ \mu_2 \sum_{i=0}^n y_i \int_a^b k_2(x, t) \psi_i(h(t)) dt. \end{aligned}$$

Collocating this equation at  $(n + 1)$  roots of the shifted Chebyshev polynomials  $T_{n+1}(x)$  in  $[a, b]$ , namely:

$$\tau_j = \frac{b + a}{2} - \frac{b - a}{2} \cos \left( \frac{(2j + 1)\pi}{2n + 2} \right), \quad j = 0, 1, \dots, n. \tag{13}$$

we get:

$$\begin{aligned} \alpha(\tau_j) \sum_{i=0}^n y_i \psi_i(\tau_j) + \beta(\tau_j) \sum_{i=0}^n y_i \psi_i(h(\tau_j)) &= f(\tau_j) + \mu_1 \sum_{i=0}^n y_i \int_a^{h(\tau_j)} k_1(\tau_j, t) \psi_i(t) dt \\ &+ \mu_2 \sum_{i=0}^n y_i \int_a^b k_2(\tau_j, t) \psi_i(h(t)) dt, \quad j = 0, 1, \dots, n. \end{aligned}$$

The above relation can be rewritten in the following form:

$$\begin{aligned} \sum_{i=0}^n \left[ \alpha(\tau_j) \psi_i(\tau_j) + \beta(\tau_j) \psi_i(h(\tau_j)) - \mu_1 \int_a^{h(\tau_j)} k_1(\tau_j, t) \psi_i(t) dt - \mu_2 \int_a^b k_2(\tau_j, t) \psi_i(h(t)) dt \right] y_i \\ = f(\tau_j) \quad j = 0, 1, \dots, n. \end{aligned} \tag{14}$$

By using the  $M$ -point Gauss-Legendre quadrature formula in the intervals  $[a, h(\tau_j)]$  and  $[a, b]$ , the integral terms in (14) can be approximated as:

$$\int_a^{h(\tau_j)} k_1(\tau_j, t) \psi_i(t) dt \simeq \Delta_{ij} = \frac{h(\tau_j) - a}{2} \sum_{k=1}^M \omega_k k_1(\tau_j, \mu_{jk}) \psi_i(\mu_{jk}), \tag{15}$$

$$\int_a^b k_2(\tau_j, t) \psi_i(h(t)) dt \simeq \Theta_{ij} = \frac{b - a}{2} \sum_{k=1}^M \omega_k k_2(\tau_j, \eta_{jk}) \psi_i(h(\eta_{jk})), \tag{16}$$

in which

$$\mu_{jk} = \frac{h(\tau_j) - a}{2} t_k + \frac{h(\tau_j) + a}{2}, \quad \eta_{jk} = \frac{b - a}{2} t_k + \frac{b + a}{2}, \tag{17}$$

$t_k$  is the  $k$ -th root of the Legendre polynomial  $P_M(t)$  and  $\omega_k$  is the  $k$ -th Gauss-Legendre quadrature weight and

$$\omega_k = \frac{2}{(1 - x_k^2) [P'_M(x_k)]^2}, \quad k = 1, 2, \dots, M. \tag{18}$$

By introducing

$$\Lambda_{ij} = \alpha(\tau_j) \psi_i(\tau_j) + \beta(\tau_j) \psi_i(h(\tau_j)), \tag{19}$$

and applying the quadrature approximations in (15) and (16), the relation (14) can be expressed in a matrix form as follows:

$$AY = F, \tag{20}$$

in which

$$A = [\Lambda_{ij} - \mu_1 \Delta_{ij} - \mu_2 \Theta_{ij}]_{(n+1) \times (n+1)}, \quad Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad F = \begin{bmatrix} f(\tau_0) \\ f(\tau_1) \\ \vdots \\ f(\tau_n) \end{bmatrix}. \tag{21}$$

The linear system (20) can be solved for the unknown vector  $Y$ . Consequently, the approximate solution of VFIE (3) can be derived by inserting the obtained vector  $Y$  in (12). The algorithm

of the proposed numerical method in pseudocode format is designed in Algorithm 1.

---

**Algorithm 1** Algorithm in pseudocode format
 

---

**Inputs:**  $n, d, M, a, b, \mu_1, \mu_2$ , the functions  $f(x), \alpha(x), \beta(x), h(x), k_1(x, t)$  and  $k_2(x, t)$ .

**Step 1:** Define the unknown vector  $Y = [y_0, y_2, \dots, y_n]^T$ .

**Step 2:** Construct the rational interpolation functions  $\psi_i(x)$  from Eq. (7).

**Step 3:** Compute the interpolation nodes  $x_i$  from Eqs. (9)-(11).

**Step 3:** Compute the collocation points  $\tau_j$  from Eq. (13).

**Step 5:** Compute the Gaussian nodes  $\mu_{jk}$  and  $\eta_{jk}$  from Eq. (17).

**Step 5:** Compute the weights  $\omega_k$  from Eq. (18).

**Step 6:** Compute  $\Delta_{ij}, \Theta_{ij}$  and  $\Lambda_{ij}$  from Eqs. (15)-(16) and (19).

**Step 7:** Compute the vector  $F$  and coefficient matrix  $A$  from Eq. (21).

**Step 8:** Solve the obtained system algebraic equations in (20) to get the unknown vector  $Y$ .

**Step 9:** Compute the approximate function  $y(x)$  by inserting the obtained vector  $Y$  in Eq. (12).

**Outputs:** The approximate solution  $y_n(x) \simeq \sum_{i=0}^n y_i \psi_i(x)$ .

---

**The special features of the method:**

- A new kind of generalized barycentric rational interpolants with the arbitrary rate of convergence is used as basis.
- The implementation of the proposed numerical approach is very simple and low-cost.
- The presented method yields an infinitely smooth solution for VFIEs.
- By using the generalized barycentric rational functions, oscillation in the obtained solution (Runge's phenomenon) is avoided.

## §5 Convergence analysis

This section deals with the convergence analysis of the proposed barycentric rational interpolation method.

**Theorem 4.** Suppose that  $\bar{Y} = [\bar{y}_0, \bar{y}_1, \dots, \bar{y}_n]^T$  and  $\bar{y}_n(x) = \sum_{i=0}^n \bar{y}_i \psi_i(x)$  are the solutions of (20) and their corresponding solutions for the VFIE (2), respectively. Moreover, let  $y_n(x) = \sum_{i=0}^n y_i \psi_i(x)$  be the Floater and Hormann approximation of the exact solution  $y(x)$ , then

$$\|y(x) - \bar{y}_n(x)\|_{\infty} \leq Ch^{d+1} + \Upsilon \delta_n,$$

where  $\delta_n = \sum_{i=0}^n |y_i - \bar{y}_i|$  and  $\Upsilon = \max_{0 \leq i \leq n} \|\psi_i\|_{\infty}$ .

*Proof.* It is easy to verify that

$$\|y(x) - \bar{y}_n(x)\|_{\infty} \leq \|y(x) - y_n(x)\|_{\infty} + \|y_n(x) - \bar{y}_n(x)\|_{\infty}.$$

From Theorems 1 and 2, it can be concluded that for a constant  $C$  we have:

$$\|y(x) - y_n(x)\|_{\infty} \leq Ch^{d+1}. \quad (22)$$

Moreover, it is easy to see that:

$$\|y_n(x) - \bar{y}_n(x)\|_\infty \leq \left| \sum_{i=0}^n (y_i - \bar{y}_i)\phi_i(x) \right| \leq \sum_{i=0}^n |y_i - \bar{y}_i| |\phi_i(x)| \leq \Upsilon \sum_{i=0}^n |y_i - \bar{y}_i| = \Upsilon \delta_n. \quad (23)$$

Now, Eqs. (22) and (23) yield

$$\|y_n(x) - \bar{y}_n(x)\|_\infty \leq Ch^{d+1} + \Upsilon \delta_n,$$

that proves the result. □

**Theorem 5.** Suppose that  $k_i(x, t), i = 1, 2, \alpha(x), \beta(x)$  and  $h(x)$  are continuous functions. Let  $\bar{y}_n(x)$  and  $y(x)$  be the approximate and exact solutions of the VFIE (2), respectively. Then, if  $M_\alpha \geq \mathcal{K}_1(M_h + a) |\mu_1|$  we have

$$\frac{\|y(x) - \bar{y}_n(x)\|_\infty}{\|y(h(x)) - \bar{y}_n(h(x))\|_\infty} \leq \frac{M_\beta + (b - a)\mathcal{K}_2 |\mu_2|}{M_\alpha - \mathcal{K}_1(M_h + a) |\mu_1|}.$$

where  $\mathcal{K}_i = \max_{0 \leq x, t \leq 1} |k_i(x, t)|, i = 1, 2, M_\alpha = \max_{a \leq x \leq b} |\alpha(x)|, M_\beta = \max_{a \leq x \leq b} |\beta(x)|$  and  $M_h = \max_{a \leq x \leq b} |h(x)|$ .

*Proof.* By substituting the approximate solution  $\bar{y}_n(x)$  in (2), we get:

$$\alpha(x)\bar{y}_n(x) + \beta(x)\bar{y}_n(h(x)) = f(x) + \mu_1 \int_a^{h(x)} k_1(x, t)\bar{y}_n(t)dt + \mu_2 \int_a^b k_2(x, t)\bar{y}_n(h(t))dt. \quad (24)$$

Subtracting Eq. (24) from (2) we obtain:

$$\begin{aligned} & \alpha(x)(y(x) - \bar{y}_n(x)) + \beta(x)(y(h(x)) - \bar{y}_n(h(x))) \\ &= \mu_1 \int_a^{h(x)} k_1(x, t)(y(t) - \bar{y}_n(t))dt + \mu_2 \int_a^b k_2(x, t)(y(h(t)) - \bar{y}_n(h(t)))dt. \end{aligned}$$

From this relation it can be concluded:

$$\begin{aligned} & M_\alpha \|y(x) - \bar{y}_n(x)\|_\infty - M_\beta \|y(h(x)) - \bar{y}_n(h(x))\|_\infty \\ & \leq |\mu_1| \int_a^{h(x)} |k_1(x, t)| |y(t) - \bar{y}_n(t)| dt + |\mu_2| \int_a^b |k_2(x, t)| |y(h(t)) - \bar{y}_n(h(t))| dt. \\ & \leq |\mu_1| \|y(x) - \bar{y}_n(x)\|_\infty \int_a^{h(x)} |k_1(x, t)| dt + |\mu_2| \|y(h(x)) - \bar{y}_n(h(x))\|_\infty \int_a^b |k_2(x, t)| dt. \\ & \leq \mathcal{K}_1(M_h + a) |\mu_1| \|y(x) - \bar{y}_n(x)\|_\infty + (b - a)\mathcal{K}_2 |\mu_2| \|y(h(x)) - \bar{y}_n(h(x))\|_\infty. \end{aligned}$$

Consequently, we have

$$\frac{\|y(x) - \bar{y}_n(x)\|_\infty}{\|y(h(x)) - \bar{y}_n(h(x))\|_\infty} \leq \frac{M_\beta + (b - a)\mathcal{K}_2 |\mu_2|}{M_\alpha - \mathcal{K}_1(M_h + a) |\mu_1|}.$$

which proves the result. □

### §6 Numerical computations

In order to demonstrate the accuracy and efficiency of the presented scheme, it has been applied to approximate solutions of different VFIEs. All numerical computations have been carried out by using Maple 17 software in a PC with Core i7 CPU and 8 GB RAM. The

root-mean-square error (RMSE) of the obtained numerical solution is estimated as

$$\|e_n\|_2 \simeq \sqrt{\frac{\sum_{i=1}^m |e_n(x_i)|^2}{m}},$$

where  $e(x) = \bar{y}_n(x) - y(x)$  is the error function of the derived approximate solution  $\bar{y}_n(x)$  by the generalized barycentric rational method.

**Example 1.** As first example, we consider the following VFIE

$$xy(x) + \sin(x)y(h(x)) = f(x) + \frac{1}{10} \int_{-1}^{h(x)} (x-t)y(t)dt + \int_{-1}^1 x^2t^2y(h(t))dt$$

where  $h(x) = \frac{x}{5}$ , the exact solution is the Runge's function  $y(x) = \frac{1}{1+25x^2}$  and  $f(x)$  is compatible with  $y(x)$ . The presented method in Section 4 for different types of node points  $x_i$  is used to solve this VFIE. Table 1 provides the RMSE of the obtained numerical solution and the elapsed CPU time (in seconds) for different values of  $n$  and  $d$ . Fig. 1 shows the graph of the approximate solutions for various values of  $n$ ,  $d$  and three types of node points in the interval  $[-1, 1]$ . Moreover, the absolute error functions for different values of  $n$  and  $d$  are plotted in Fig. 2. The presented results confirm the efficiency and limited computational cost of the presented algorithm. Moreover, it can be concluded that, for all three types of node points, the obtained approximate solutions converge well as the number of basis functions increases and there is no oscillation in the obtained results. In other word, by using the generalized barycentric rational functions, Runge's phenomenon is avoided.

Table 1. The RMSE and elapsed CPU time (in seconds) for different values of  $n$  and  $d$  (Example 1).

| $(n, d)$ | $\ e_n\ _2$ |          |          | CPU time |         |         |
|----------|-------------|----------|----------|----------|---------|---------|
|          | UP          | CG       | CGL      | UP       | CG      | CGL     |
| (24, 10) | 1.77E-03    | 2.31E-03 | 2.36E-03 | 4.765    | 8.750   | 9.313   |
| (30, 14) | 4.17E-04    | 5.48E-04 | 5.59E-04 | 12.453   | 21.563  | 23.938  |
| (36, 16) | 9.77E-05    | 1.47E-04 | 1.50E-04 | 25.407   | 38.516  | 37.907  |
| (48, 20) | 6.50E-06    | 1.28E-05 | 1.31E-05 | 91.531   | 139.891 | 124.016 |

**Example 2.** Now, we consider the following VFIE

$$e^{-x}y(x) + \cos(2x)y(h(x)) = f(x) + \int_0^{h(x)} e^{x-t}y(t)dt + \int_0^1 \frac{y(h(t))}{x^2+t^2}dt$$

where  $h(x) = \frac{x}{3}$ ,  $x^2$  and  $f(x)$  is chosen such that the exact solution is the rational function  $y(x) = \frac{x-x^2}{x^4+x^2+2}$ . The proposed barycentric rational interpolant scheme has been utilized to solve this VFIE for various choices of  $n$  and  $d$ . The RMSE of the approximate solutions and the elapsed CPU time (in seconds) for three types of node points and various  $n$  and  $d$  are provided in Tables 2 and 3, respectively. For  $h(x) = \frac{x}{3}$  and  $h(x) = x^2$ , Figs. 3 and 4 show the absolute error of approximate solutions for three types of node points and various  $n$  and  $d$ . From these results, we can conclude that for all types of node points the proposed barycentric rational

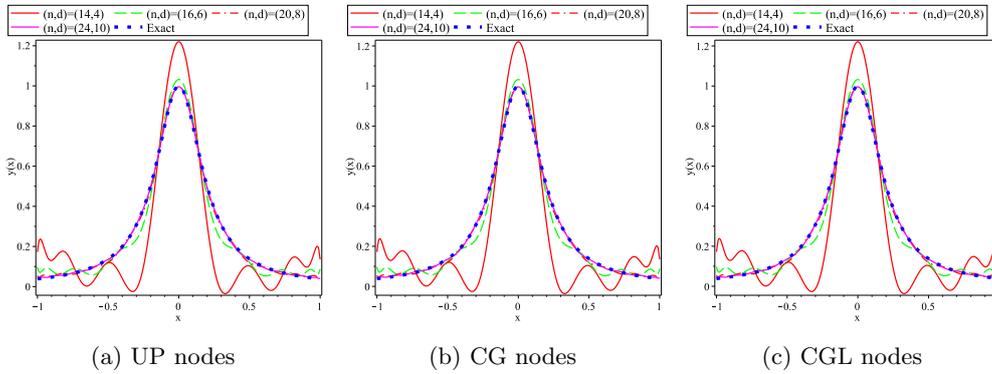


Figure 1. The approximate solutions for various values of  $n, d$  and three types of node points in the interval  $[-1, 1]$ . (Example 1).

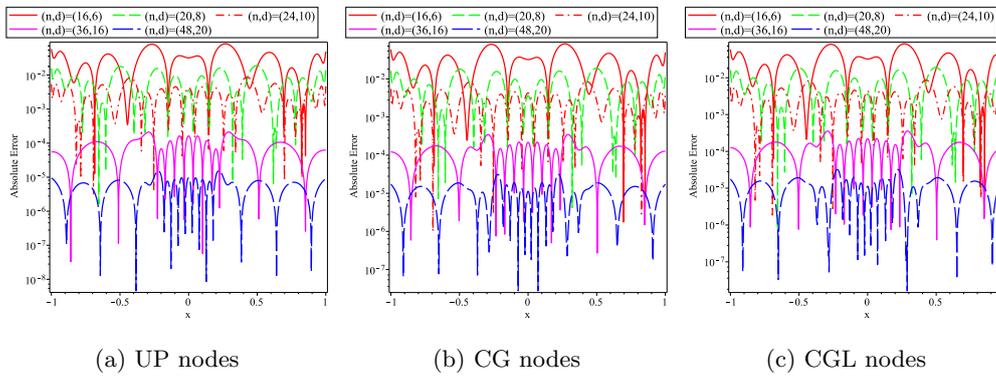


Figure 2. The absolute error for various values of  $n, d$  and three types of node points (Example 1).

*interpolation method is efficient and low-cost for solving this VFIE. Moreover, as the number of basis functions i.e.  $n$  increases, the accuracy of the numerical results increases significantly without any oscillation.*

Table 2. The RMSE of the obtained numerical solutions for different values of  $n$  and  $d$  (Example 2).

| $(n, d)$ | $h(x) = \frac{x}{3}$ |          |          | $h(x) = x^2$ |          |          |
|----------|----------------------|----------|----------|--------------|----------|----------|
|          | UP                   | CG       | CGL      | UP           | CG       | CGL      |
| (6, 4)   | 3.63E-04             | 3.26E-04 | 2.77E-04 | 4.27E-04     | 3.91E-04 | 3.32E-04 |
| (8, 6)   | 3.29E-06             | 1.67E-06 | 2.66E-07 | 7.67E-06     | 3.81E-06 | 3.32E-07 |
| (10, 8)  | 8.90E-08             | 1.62E-07 | 2.71E-07 | 1.67E-07     | 4.04E-07 | 5.02E-07 |
| (12, 10) | 5.50E-08             | 6.71E-08 | 7.30E-08 | 5.99E-09     | 3.47E-08 | 2.26E-08 |

Table 3. The elapsed CPU time (in second) for different values of  $n$  and  $d$  (Example 2).

| $(n, d)$ | $h(x) = \frac{x}{3}$ |       |       | $h(x) = x^2$ |        |        |
|----------|----------------------|-------|-------|--------------|--------|--------|
|          | UP                   | CG    | CGL   | UP           | CG     | CGL    |
| (6, 2)   | 3.531                | 4.157 | 3.609 | 3.094        | 3.390  | 3.750  |
| (8, 4)   | 4.641                | 5.062 | 4.703 | 5.110        | 5.125  | 5.575  |
| (10, 6)  | 5.438                | 6.719 | 6.766 | 6.247        | 7.181  | 6.766  |
| (12, 8)  | 7.328                | 9.328 | 9.562 | 9.873        | 10.766 | 10.188 |

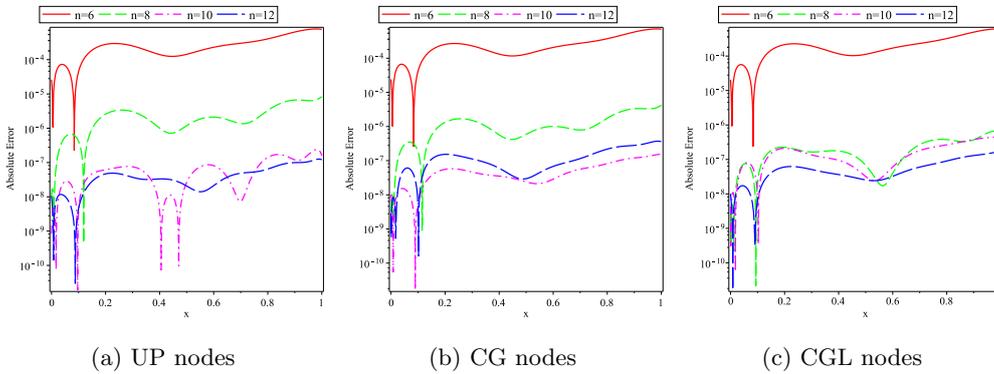


Figure 3. The absolute error for  $h(x) = \frac{x}{3}$  and various values of  $n$  and  $d = n - 2$  (Example 2).

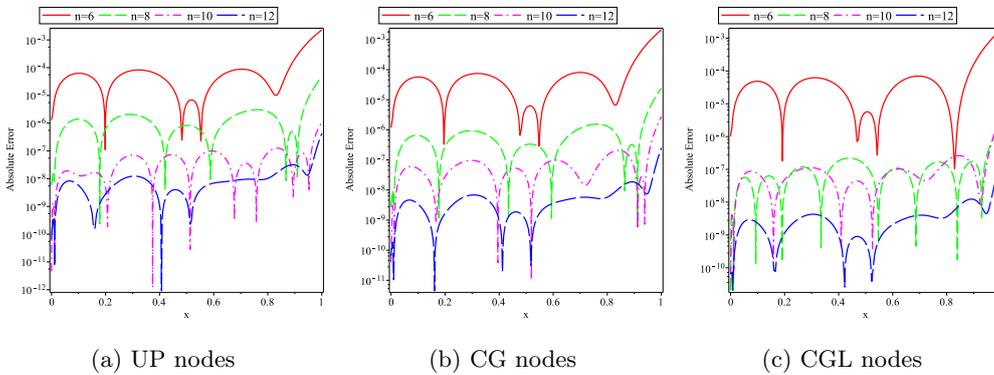


Figure 4. The absolute error for  $h(x) = x^2$  and various values of  $n$  and  $d = n - 2$  (Example 2).

**Example 3.** Let us consider the following VFIE [15, 17]

$$x^2 y(x) + e^x y(h(x)) = f(x) + \int_0^{h(x)} e^{x+t} y(t) dt - \int_0^1 e^{x-h(t)} y(h(t)) dt$$

where  $h(x) = 2x$  and  $f(x)$  is compatible with the exact solution  $y(x) = \sin(x)$ . The described scheme in Section 4 with different values of  $n$  and  $d$  has been applied to approximate solution of this problem. In Table 4, the RMSE of the obtained numerical solutions for different values of  $n$

and  $d$  were compared to those achieved by methods in Refs. [1, 15, 17, 29]. The elapsed CPU time (in second) for various values of  $n$  and  $d$  are tabulated in Table 5. These results confirm the efficiency and low computational cost of the proposed method in solving this VFIE. In addition, it can be concluded that for all types of node points, the barycentric rational interpolation method is more accurate than the previous methods. Moreover, it is clear that the obtained approximate solutions converge well as the number of basis function  $n$  increases.

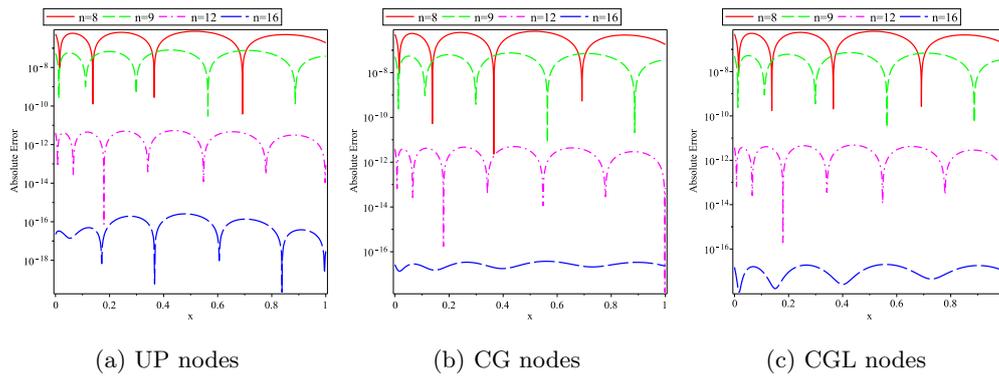


Figure 5. The absolute error for various values of  $n$  and  $d = n - 2$  (Example 3).

Table 4. The RMSE of the obtained numerical solutions for different values of  $n$  and  $d = n - 2$  (Example 3).

| $n$ | Present Method |          |          | Ref. [17] | Ref. [15] | Ref. [29] | Ref. [1] |
|-----|----------------|----------|----------|-----------|-----------|-----------|----------|
|     | UP             | CG       | CGL      |           |           |           |          |
| 5   | 4.51E-05       | 2.26E-05 | 3.67E-05 | 2.93E-05  | 6.23E-05  | 3.68E-04  | 6.23E-05 |
| 8   | 4.23E-08       | 6.31E-08 | 4.07E-08 | 3.94E-08  | 1.89E-08  | 1.24E-05  | 1.77E-07 |
| 9   | 4.97E-09       | 4.67E-09 | 1.23E-09 | 2.29E-09  | 2.35E-08  | 3.46E-07  | 7.21E-06 |
| 12  | 3.24E-12       | 3.07E-12 | 2.97E-12 | —         | —         | —         | —        |
| 16  | 2.9E-16        | 1.83E-17 | 1.21E-17 | —         | —         | —         | —        |

Table 5. The elapsed CPU time (in seconds) for different values of  $n$  and  $d = n - 2$  (Example 3).

| $n$ | UP    | CG    | CGL   |
|-----|-------|-------|-------|
| 5   | 0.859 | 0.860 | 0.797 |
| 8   | 1.328 | 1.641 | 1.578 |
| 9   | 1.750 | 2.016 | 1.890 |
| 12  | 3.625 | 4.015 | 4.016 |
| 16  | 8.297 | 9.531 | 9.000 |

**Example 4.** Consider the following VFIE [15, 17]

$$y(x) = f(x) + \int_0^{h(x)} e^{x+t}y(t)dt + \int_0^1 e^{x+h(t)}y(h(t))dt$$

where  $h(x) = \frac{x}{2}, \ln(x + 1)$  and  $f(x)$  is chosen such that the exact solution is  $y(x) = e^{-x}$ . The suggested approach in Section 4 with various values of  $n$  and  $d$  is used to solve this VFIE. Table 6 provides a comparison between the obtained results by our proposed algorithm and those derived in Refs. [15, 17]. The elapsed CPU time (in seconds) for different values of  $n$  and  $d$  are also presented in Table 7. Moreover, Figs. 6 and 7 show the absolute error functions with different values of  $n$  and  $d$ . These results reveal that the proposed method is accurate and low-cost for solving such problems and the approximate solutions converge to the exact solution as the number of basis function increases. From these results, we can also see that for all types of node points, the barycentric rational interpolation method is more accurate than the reported results in Refs. [15, 17].

Table 6. The RMSE of the obtained numerical solutions for different values of  $n$  and  $d = n - 2$  (Example 4).

| $h(x) = \frac{x}{2}$ |                |            |            |            |            |
|----------------------|----------------|------------|------------|------------|------------|
| $n$                  | Present Method |            |            | Ref. [17]  | Ref. [15]  |
|                      | UP             | CG         | CGL        |            |            |
| 5                    | 1.5560E-05     | 1.4966E-05 | 1.3333E-05 | 3.7054E-07 | 3.3616E-04 |
| 8                    | 1.0224E-09     | 9.6185E-10 | 9.0913E-10 | 6.7434E-07 | 5.7654E-07 |
| 10                   | 8.4137E-13     | 7.9198E-13 | 7.6191E-13 | –          | –          |
| 16                   | 1.6181E-16     | 1.6202E-16 | 1.6205E-16 | –          | –          |
| $h(x) = \ln(x + 1)$  |                |            |            |            |            |
| $n$                  | Present Method |            |            | Ref. [17]  | Ref. [15]  |
|                      | UP             | CG         | CGL        |            |            |
| 5                    | 1.4311E-05     | 1.3783E-05 | 1.2325E-05 | 4.3023E-07 | 3.0506E-04 |
| 8                    | 9.7696E-10     | 9.1996E-10 | 8.7031E-10 | 5.7851E-07 | 5.6092E-07 |
| 10                   | 8.1359E-13     | 7.6620E-13 | 7.3732E-13 | –          | –          |
| 16                   | 1.6268E-16     | 1.6203E-16 | 1.6202E-16 | –          | –          |

**Example 5.** Finally, we consider the following FVIE [19]

$$y(x) = f(x) + \int_0^{h(x)} xty(t)dt + \int_0^1 (x - t)y(h(t))dt$$

where  $h(x) = \frac{x}{3}$  and  $f(x)$  is compatible with the exact solution  $y(x) = \log(x + 1)$ . The solution of this FVIE has been approximated by using the proposed method for various values of  $n$  and  $d$ . The absolute error functions for various choices of  $n$  and  $d$  are plotted in Fig. 8. For different choices of  $n$  and  $d$ , Table 8 provides a comparison between the RMSE of the obtained approximate solutions and those derived in Refs. [15, 19]. The elapsed CPU time (in seconds) for

Table 7. The elapsed CPU time (in seconds) for different values of  $n$  and  $d = n - 2$  (Example 4).

| $n$ | $h(x) = \frac{x}{2}$ |       |       | $h(x) = \ln(x + 1)$ |       |       |
|-----|----------------------|-------|-------|---------------------|-------|-------|
|     | UP                   | CG    | CGL   | UP                  | CG    | CGL   |
| 5   | 0.635                | 0.537 | 0.563 | 0.585               | 0.569 | 0.584 |
| 8   | 1.138                | 1.156 | 1.140 | 1.262               | 1.147 | 1.235 |
| 9   | 1.956                | 1.515 | 1.547 | 1.484               | 1.531 | 1.593 |
| 12  | 3.406                | 2.859 | 3.375 | 3.235               | 3.062 | 3.109 |
| 16  | 6.652                | 6.953 | 8.328 | 7.371               | 8.172 | 8.250 |

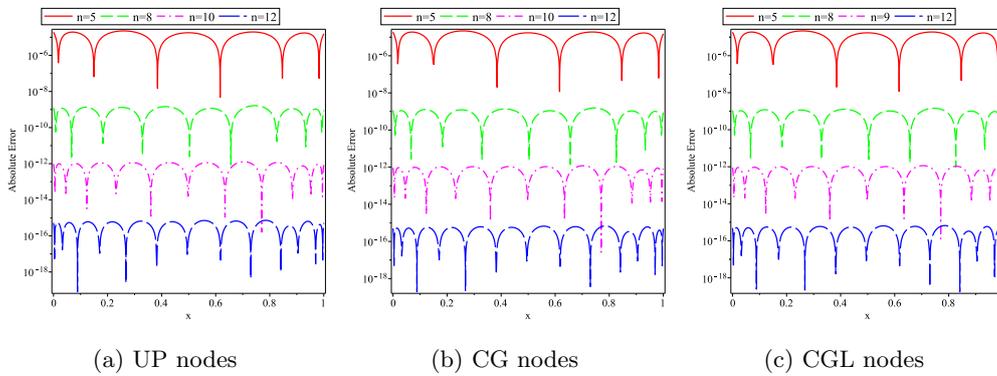


Figure 6. The absolute error for various values of  $n$ ,  $h(x) = \ln(1 + x)$  and  $d = n - 2$  (Example 4).

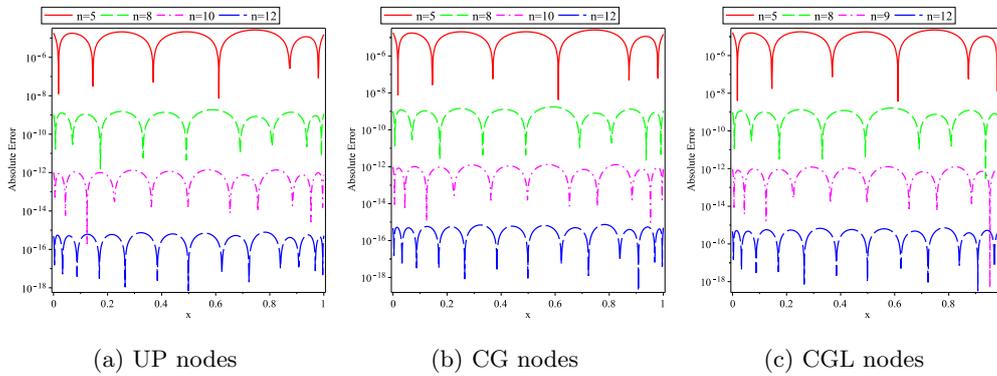


Figure 7. The absolute error for various values of  $n$ ,  $h(x) = \frac{x}{2}$  and  $d = n - 2$  (Example 4).

different values of  $n$  and  $d$  are provided in Table 9. According to these results, we can conclude that the proposed method is accurate and low-cost for such FVIE and the approximate solutions converge to the exact solution as the number of basis functions increases. Furthermore, we can see that for all types of node points, the barycentric rational interpolation method is more accurate than the reported results in Refs. [15, 19].

Table 8. The RMSE of the obtained numerical solutions for different values of  $n$  and  $d = n - 2$  (Example 5).

| $n$ | Present Method |          |          | Ref. [19] | Ref. [15] |
|-----|----------------|----------|----------|-----------|-----------|
|     | UP             | CG       | CGL      |           |           |
| 4   | 3.14E-05       | 3.12E-05 | 2.72E-05 | 3.64E-05  | 5.29E-05  |
| 5   | 3.87E-06       | 3.75E-06 | 3.40E-06 | 5.27E-06  | 1.02E-05  |
| 6   | 7.01E-07       | 4.80E-06 | 4.47E-06 | 7.72E-07  | 1.42E-06  |
| 9   | 1.52E-09       | 1.45E-09 | 1.39E-09 | –         | –         |
| 12  | 5.51E-12       | 5.27E-12 | 5.21E-12 | –         | –         |
| 16  | 5.35E-15       | 3.32E-15 | 3.98E-15 | –         | –         |

Table 9. The elapsed CPU time (in seconds) for different values of  $n$  and  $d = n - 2$  (Example 5).

| $n$ | UP    | CG    | CGL   |
|-----|-------|-------|-------|
| 4   | 0.765 | 0.719 | 0.781 |
| 5   | 0.843 | 0.875 | 0.906 |
| 6   | 1.093 | 1.032 | 1.109 |
| 9   | 1.937 | 2.015 | 2.250 |
| 12  | 3.034 | 3.609 | 3.797 |
| 16  | 7.641 | 8.844 | 8.859 |

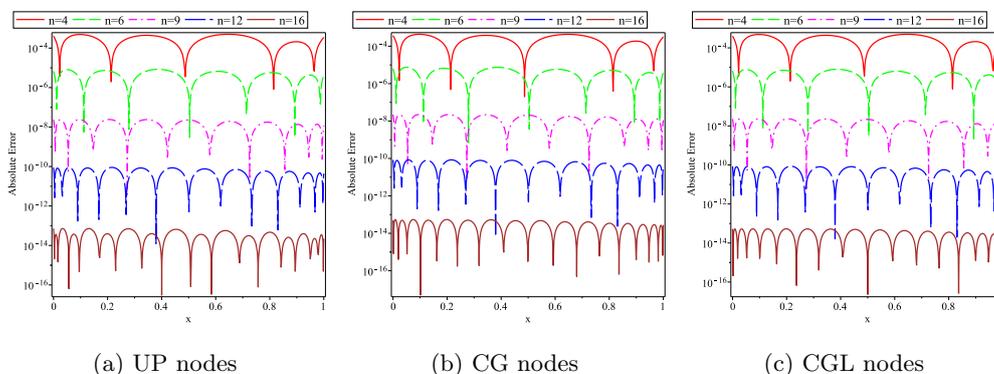


Figure 8. The absolute error for various values of  $n$ ,  $d = n - 2$  (Example 5).

### §7 Concluding Remarks

A family of generalized barycentric rational interpolation functions has been reviewed and their properties have been investigated. Then, a numerical collocation method based on these generalized barycentric rational interpolant and Gaussian quadrature formula were introduced to approximate solution of Volterra-Fredholm integral equations. Three types of distinct points are used as interpolation nodes and the proposed algorithm is implemented in some illustrative

examples. The obtained numerical results were compared to those reported in the previous paper and confirmed the superiority and limited computational cost of the presented algorithm. Moreover, the presented method yields an infinitely smooth solution for VFIEs that avoid oscillations such as Runge's phenomenon.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

- [1] K Wang, Q Wang. *Lagrange collocation method for solving Volterra-Fredholm integral equations*, Applied Mathematics and Computation, 2013, 219(21): 10434-10440.
- [2] J P Berrut, S A Hosseini, G Klein. *The linear barycentric rational quadrature method for volterra integral equations*, SIAM Journal on Scientific Computing, 2014, 36(1): 105-123.
- [3] Y Jafarzadeh, B Keramati. *Numerical method for a system of integro-differential equations by Lagrange interpolation*, Asian-European Journal of Mathematics, 2016, 9(4): 1650077.
- [4] X J Yang. *New general calculi with respect to another functions applied to describe the newton-like dashpot models in anomalous viscoelasticity*, Thermal Science, 2019, 23(6B): 3751-3757.
- [5] J P Berrut. *Rational functions for guaranteed and experimentally well-conditioned global interpolation*, Computers and Mathematics with Applications, 1988, 15(1): 1-16.
- [6] J P Berrut, L N Trefethen. *Barycentric lagrange interpolation*, SIAM review, 2004, 46(3): 501-517.
- [7] M S Floater, K Hormann. *Barycentric rational interpolation with no poles and high rates of approximation*, Numer Math, 2007, 107: 315-331.
- [8] R J Rivlin. *An Introduction to the Approximation of Functions*, Blaisdell, Waltham, 1979.
- [9] J Stoer, R Bulirsch. *Introduction to Numerical Analysis*, 3rd edn, Springer, New York, 2002.
- [10] J P Berrut, H D Mittelmann. *Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval*, Comput Math Appl, 1997, 33(6): 77-86.
- [11] M Erfanian. *The approximate solution of nonlinear mixed Volterra-Fredholm-Hammerstein integral equations with RH wavelet bases in a complex plane*, Mathematical Methods in the Applied Sciences, 2018, <https://doi.org/10.1002/mma.4714>.
- [12] F Mohammadi. *A computational wavelet method for numerical solution of stochastic Volterra-Fredholm integral equations*, Wavelet and Linear Algebra, 2016, 3(1): 13-25.

- [13] F Mohammadi. *A Chebyshev wavelet operational method for solving stochastic Volterra-Fredholm integral equations*, International Journal of Applied Mathematical Research, 2015, 4(2): 217-227.
- [14] K Sadri, A Amini, C Cheng. *Low cost numerical solution for three-dimensional linear and nonlinear integral equations via three-dimensional Jacobi polynomials*, Journal of Computational and Applied Mathematics, 2017, 319: 493-513.
- [15] K Wang, Q Wang. *Taylor collocation method and convergence analysis for the Volterra-Fredholm integral equations*, Journal of Computational and Applied Mathematics, 2014, 260: 294-300.
- [16] S Yüzbaşı. *A collocation method based on Bernstein polynomials to solve nonlinear Fredholm-Volterra integro-differential equations*, Applied Mathematics and Computation, 2016, 273: 142-154.
- [17] S Nemati. *Numerical solution of Volterra-Fredholm integral equations using Legendre collocation method*, Journal of Computational and Applied Mathematics, 2015, 278: 29-36.
- [18] S A Hosseini, S Shahmorad, F Talati. *A matrix based method for two dimensional nonlinear Volterra-Fredholm integral equations*, Numerical Algorithms, 2015, 68(3): 511-529.
- [19] Q Wang, K Wang, S Chen. *Least squares approximation method for the solution of Volterra-Fredholm integral equations*, Journal of Computational and Applied Mathematics, 2014, 272: 141-147.
- [20] F Mirzaee, E Hadadiyan. *Using operational matrix for solving nonlinear class of mixed Volterra-Fredholm integral equations*, Mathematical Methods in the Applied Sciences, 2017, 40(10): 3433-3444.
- [21] S Mashayekhi, M Razzaghi, O Tripak. *Solution of the nonlinear mixed Volterra-Fredholm integral equations by hybrid of block-pulse functions and Bernoulli polynomials*, The Scientific World Journal, 2014, <https://doi.org/10.1155/2014/413623>.
- [22] Y Ma, J Huang, C Wang. *Numerical Solutions of Nonlinear Volterra-Fredholm-Hammerstein Integral Equations Using Sinc Nyström Method*, In Information Technology and Intelligent Transportation Systems, 2017, 455: 187-194.
- [23] F Mirzaee, E Hadadiyan. *Applying the modified block-pulse functions to solve the three-dimensional Volterra-Fredholm integral equations*, Applied Mathematics and Computation, 2015, 265: 759-767.
- [24] H Xu, L Fan. *Numerical solution of Volterra-Fredholm integral equations based on  $\epsilon$ -SVR method*, Journal of Computational and Applied Mathematics, 2016, 298: 201-210.
- [25] E Cirillo, K Hormann, J Sidon. *Convergence rates of derivatives of Floater-Hormann interpolants for well-spaced nodes*, Applied Numerical Mathematics, 2017, 116: 108-118.

- [26] J P Berrut, M S Floater, G Klein. *Convergence rates of derivatives of a family of barycentric rational interpolants*, Applied numerical mathematics, 2011, 61(9): 989-1000.
- [27] H Brunner. *On the numerical solution of Volterra-Fredholm integral equation by collocation methods*, SIAMJ Numer Anal, 1990, 27(4): 87-96.
- [28] L M Delves, J L Mohamed. *Computational Methods for Integral Equations*, Cambridge University Press, Cambridge, 1985.
- [29] K Maleknejad, Y Mahmoudi. *Taylor polynomial solution of high-order nonlinear Volterra-Fredholm integro-differential equations*, Applied Mathematics and Computation, 2003, 145(2-3): 641-653.

Department of Mathematics, University of Hormozgan , Bandar Abbas, P. O. Box 3995, Iran.  
Emails: h.azin1370@gmail.com, f.mohammadi62@hotmail.com