



An improved radial basis functions method for the high-order Volterra–Fredholm integro-differential equations

Farnaz Farshadmoghadam¹ · Haman Deilami Azodi¹ · Mohammad Reza Yaghouti¹

Received: 16 December 2020 / Accepted: 15 August 2021 / Published online: 24 August 2021
© Islamic Azad University 2021

Abstract

This paper is aimed at rectifying the numerical solution of linear Volterra–Fredholm integro-differential equations with the method of radial basis functions (RBFs). In this method, the spectral convergence rate can be acquired by infinitely smooth radial kernels such as Gaussian RBF (GA-RBF). These kernels are made by a free shape parameter, and the highest accuracy can often be achieved when this parameter is small, but herein the coefficient matrix of interpolation is ill-conditioned. Alternative bases can be used to improve the stability of method. One of them is based on the eigenfunction expansion for GA-RBFs which is utilized in this study. The Legendre–Gauss–Lobatto integration rule is applied to estimate the integral parts. Moreover, the error analysis is discussed. The results of numerical experiments are presented to demonstrate stable solutions with high accuracy compared to the standard GA-RBFs, the analytical solutions, and the other methods.

Keywords Volterra–Fredholm integro-differential equation · Gaussian radial basis function · Legendre–Gauss–Lobatto quadrature · Eigenfunction expansion · Collocation method

Mathematics Subject Classification 45J05 · 65D15 · 65R20

Introduction

The study of integro-differential equations (IDEs) arises from its abundant applications in the real world, such as finance [46], ecology and evolutionary biology [37], medicine [20], mechanics and plasma physics [2, 23, 27] and other cases. Therefore, computational methods to solve these types of equations are noteworthy by many scientists. Some methods for solving these equations include: Bessel functions [50], Wavelets [11, 42], Adomian decomposition method [25, 29], compact finite difference method [51, 52], Galerkin method [7, 35], homotopy analysis method [24, 39] and radial basis functions method [12, 21, 28, 30, 45].

Recently, the radial basis functions (RBFs) meshless method has been used as an effective technique to interpolate and approximate the solution of differential equations (DEs), and integral equations (IEs). In the RBF method, we do not mesh the geometric space of the problem, but instead use some scattered points in the desired domain. Since the RBF method uses pairwise distances between points, so it is efficient for problems with high dimensions and irregular domains. First time, Hardy introduced MQ-RBF which was motivated by the cartographic problem [26], and later Kansa utilized it to solve parabolic, hyperbolic and elliptic partial differential equations [32]. Frank [18] examined about 30 two-dimensional approaches to interpolation and demonstrated that the best ones are MQ and TPS.

Radial basis functions are classified into two main classes [8, 33]:

Class 1. Infinitely smooth RBFs.

These basis functions are infinitely differentiable and depend on a parameter, called shape parameter ϵ (analogous Multiquadric(MQ), inverse multiquadric (IMQ) and Gaussian (GA)).

Class 2. Infinitely smooth (except at centers) RBFs.

✉ Mohammad Reza Yaghouti
yaghouti@guilan.ac.ir

Farnaz Farshadmoghadam
farshadmoghadam@phd.guilan.ac.ir

Haman Deilami Azodi
haman.d.azodi@gmail.com

¹ Department of Applied Mathematics and Computer Sciences, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran



These basis functions are not infinitely differentiable and are lacking the shape parameter. Also, these functions have less accurate than ones in Class 1. (analogous Thin Plate Splines (TPS)).

Selecting the optimal shape parameter is still an open problem. Some attempts have been made to overcome this problem. For MQ-RBF, Hardy [26] expressed $\epsilon = 0.815d$ in which $d = \frac{1}{N} \sum_{i=1}^N d_i$, N is the number of data and d_i is the distance between the i^{th} data point and its closest neighborhood. Franke [19] defined $\epsilon = \frac{1.25D}{\sqrt{N}}$ where D is the diameter of the smallest circle containing all data points. Carlson and Foley [5] declared that a good value of ϵ is independent of the number and position of the grid points. Furthermore, they employed the same value of ϵ for MQ and IMQ RBF interpolation. Rippa [43] showed that the optimal value depends on the number and position of grid points, the approximated function, the accuracy of calculation and the kind of RBF interpolation. Moreover, Rippa introduced an algorithm based on the Leave-One-Out Cross Validation (LOOCV) method for any RBF and any dimension by minimizing a cost function. In [3], Azarboni et al. presented the Leave-Two-Out Cross Validation (LTOCV) method which is more accurate than the LOOCV one. In RBFs, when the shape parameter is small and the basis function is near to be flat, the accuracy of method increases, but in this case, the system of equations is ill-conditioned. To overcome this problem, researchers have made many efforts that have led to the production of stable methods for small values of the shape parameter, like the Contour-Pade method [17], the RBF-QR method [15, 16, 34] and the Hilbert-Schmidt SVD method [6].

Fasshauer et al. established a new procedure to calculate and evaluate Gaussian radial basis function interpolants in a stable way with a concentration on small values of the shape parameter; see [14] and references therein. Using this, Rashidinia et al. introduced an effective stable method to estimate Gaussian radial basis function interpolants which has been performed by the eigenfunction expansion [41]. This approach is based on the properties of the orthogonal eigenfunctions and their zeros. They tested various interpolations and solved the boundary value problems in one and two dimensions by applying their method.

In this paper, we focus on high-order Volterra–Fredholm integro-differential equations in the following general form

$$\sum_{k=0}^m p_k(x)y^{(k)}(x) + \theta_1 \int_a^b k_1(x,t)y^{(l_1)}(t)dt + \theta_2 \int_a^x k_2(x,t)y^{(l_2)}(t)dt = f(x), \quad a \leq x \leq b, \tag{1}$$

with initial-boundary conditions

$$\sum_{k=0}^{m-1} (r_{jk}y^{(k)}(a) + s_{jk}y^{(k)}(b)) = q_j, \quad j = 0, 1, 2, \dots, m - 1. \tag{2}$$

, where $r_{jk}, s_{jk}, q_j, \theta_1, \theta_2, a$ and b are real constants, $m \in \mathbb{N}$, l_1, l_2 are two non-negative integers, $y^{(0)}(x) = y(x)$ is an unknown function, p_k and f are known and continuous functions on $[a, b]$ and the kernels k_1 and k_2 are known and analytical functions that have suitable derivatives on $[a, b] \times [a, b]$. With the help of Legendre–Gauss–Lobatto approximation for integrals and collocation points, we intend to develop the eigenfunction expansion method to solve (1) under the conditions (2).

The method proposed in this work transforms the main problem to a system of linear algebraic equations. By solving the resulting system, the coefficients of expansion for the solution are found. In illustrative examples, very small values for the shape parameter are employed. Numerical results show stable solutions with high accuracy and appropriate condition number compared to the standard GA-RBF method.

This paper is arranged as follows: In Sect. 2, a review of previous studies about standard RBFs interpolation and eigenfunction expansions are mentioned. The Legendre–Gauss–Lobatto nodes and weights are described in Sect. 3. In Sect. 4, the present method is explained. Section 5 is assigned to a discussion on the residual error. Numerical examples are included in Sect. 6.

The preliminaries and basic definitions

In this section, we briefly describe some basic definitions and items required in this work.

Standard RBFs interpolation method

Definition 1 [48] A function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is called radial basis whenever there exists a univariate function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ such that $\Phi(x) = \varphi(r)$, where $r = \|x\|$ and $\|\cdot\|$ is usual Euclidean norm.

Definition 2 A real-valued continuous function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is called positive definite whenever for any N distinct points $\{x_i\}_{i=1}^N \in \mathbb{R}^d$ and $\mathbf{G} = (g_1, g_2, \dots, g_N)^T \in \mathbb{R}^N$, we have

$$\sum_{i=1}^N \sum_{j=1}^N g_i g_j \Phi(x_i - x_j) \geq 0. \tag{3}$$

The function Φ is called strictly positive definite if unequal in (3) becomes equal only for $\mathbf{G} = 0$.

Definition 3 (Scattered data interpolation problem [13, 38, 47]). For given data $f_i \in \mathbb{R}$ at scattered data $\{x_i\}_{i=1}^N \in \mathbb{R}^d$, we find a continuous function W such that

$$W(x_i) = f_i, \quad i = 1, 2, \dots, N. \tag{4}$$

One of the efficient methods for interpolating scattered points is the RBF method. An RBF approximation is presented as follows:

$$W(x) = \sum_{i=1}^N c_i \varphi(\|x - x_i\|), \quad x, x_i \in \mathbb{R}^d. \tag{5}$$

To interpolate the given values $\{f_i\}_{i=1}^N$ at scattered data set $\mathbf{x} = \{x_i\}_{i=1}^N$, namely center nodes, where $\varphi(r)$ is a radial kernel.

In order to acquire unknown coefficients $\mathbf{C} = (c_1, c_2, \dots, c_N)^T$, we apply the interpolation conditions (4) which is equivalent to solving the following system

$$\mathbf{A}\mathbf{C} = \mathbf{F}, \tag{6}$$

so that $\mathbf{A}_{ij} = \varphi(\|x_i - x_j\|)$ and $\mathbf{F} = (f_1, f_2, \dots, f_N)^T$.

If the function φ is strictly positive definite, then the matrix \mathbf{A} is positive definite and so non-singular. Therefore, the system of equations corresponding to the interpolation problem has a unique solution. Some popular choices of radial kernels are summarized in Table 1.

The following theorem is related to the convergence of RBFs interpolation:

Theorem 1 If $\{x_i\}_{i=1}^N$ are N nodes in the convex space $\Omega \subset \mathbb{R}^d$ and

$$h = \max_{x \in \Omega} \min_{1 \leq i \leq N} \|x - x_i\|_2,$$

when $\hat{\phi}(\eta) < c(1 + |\eta|)^{-2l+d}$ for any $y(x)$ satisfies in $\int (\hat{y}(\eta))^2 / \hat{\phi}(\eta) d\eta < \infty$, then

$$\|y_N^{(\alpha)} - y^{(\alpha)}\|_\infty < ch^{l-\alpha},$$

in which ϕ refers to RBFs and the constant c depends on RBFs, $\hat{\phi}$ and \hat{y} are supposed to be the Fourier transforms of

Table 1 Some radial functions that are strictly positive definite, infinitely smooth and depend on a free shape parameter ε which controls their flatness

Name of Function	RBF Form
Gaussian (GA)	$\varphi(r) = e^{-\varepsilon^2 r^2}$
Multiquadric (MQ)	$\varphi(r) = \sqrt{1 + \varepsilon^2 r^2}$
Inverse multiquadric (IMQ)	$\varphi(r) = \frac{1}{\sqrt{1 + \varepsilon^2 r^2}}$
Inverse quadric (IQ)	$\varphi(r) = \frac{1}{1 + \varepsilon^2 r^2}$

ϕ and y respectively, $y^{(\alpha)}$ denotes the α^{th} derivative of y , y_N is the RBFs approximation of y , d is space dimension, l and α are non-negative integers.

Proof A complete proof is presented in [49, 53]. □

It can be seen that not only RBFs itself but also its any order derivative has a good convergence.

An eigenfunction expansion for Gaussian RBFs

Let the Gaussian kernel be a positive definite function. So, the coefficient matrix in the system (6) is non-singular but often ill-condition. Using Mercer’s theorem, a decomposition of the matrix \mathbf{A} is found which overcomes the ill-conditioning of system.

Theorem 2 (Mercer). If $X \subseteq \mathbb{R}^d$ be closed, then a continuous positive definite kernel $K : X \times X \rightarrow \mathbb{R}$ has an eigenfunction expansion

$$K(x, y) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(y), \quad x, y \in X,$$

so that λ_n are positive eigenvalues, φ_n are orthogonal eigenfunctions in $L_2(X)$ and the series converges absolutely and uniformly. Moreover,

$$\lambda_n \varphi_n(x) = \int_X K(x, y) \varphi_n(y) dy, \quad x \in X, n \geq 0.$$

Proof A complete explanation is given in [36, 40, 41]. □

So, GA-RBFs can be written by

$$e^{-\varepsilon^2 \|x-y\|^2} = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(y). \tag{7}$$

The eigenfunctions φ_n are orthogonal with respect to the weight function $\rho(x) = \frac{\alpha}{\sqrt{\pi}} e^{-\alpha^2 x^2}$, and

$$\varphi_n(x) = \sqrt{\gamma_n} e^{-\delta^2 x^2} H_{n-1}(\alpha \beta x), \quad n = 1, 2, \dots, \tag{8}$$

where H_{n-1} are Hermite polynomials of degree $n - 1$ defined by the following recursion relation

$$\begin{cases} H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), & n \geq 0, \\ H_{-1}(x) = 0, \quad H_0(x) = 1. \end{cases} \tag{9}$$

In addition,

$$\int_{\mathbb{R}} H_{n-1}^2(x)e^{-x^2} dx = \sqrt{\pi}2^{n-1}\Gamma(n), \quad n = 1, 2, \dots,$$

and

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2 + \delta^2 + \epsilon^2}} \left(\frac{\epsilon^2}{\alpha^2 + \delta^2 + \epsilon^2} \right)^{n-1}, \quad n = 1, 2, \dots,$$

$$\beta = \left(1 + \left(\frac{2\epsilon}{\alpha} \right)^2 \right)^{\frac{1}{4}}, \quad \gamma_n = \frac{\beta}{2^{n-1}\Gamma(n)},$$

$$\delta^2 = \frac{\alpha^2}{2}(\beta^2 - 1).$$

Also, by applying (8) and (9) we get

$$\begin{cases} \varphi_{n+1}(x) = \alpha\beta x \sqrt{\frac{2}{n}}\varphi_n(x) - \sqrt{\frac{n-1}{n}}\varphi_{n-1}(x), & n \geq 1, \\ \varphi_0(x) = 0, \quad \varphi_1(x) = \sqrt{\beta}e^{-\delta^2 x^2}. \end{cases} \quad (10)$$

The parameters ϵ and α are called the shape and the global scale parameters, respectively. Although they are freely chosen, but they have an important effect on the accuracy of the method. For GA-RBFs, if α is fixed and $\epsilon \rightarrow 0$, then $\beta \rightarrow 1$, $\delta \rightarrow 0$, the eigenfunctions φ_n converge to the normalized Hermite polynomials $\tilde{H}_n(x) = \frac{1}{\sqrt{2^{n-1}\Gamma(n)}}H_{n-1}(\alpha x)$ and the eigenvalues behave similarly $\left(\frac{\epsilon^2}{\alpha^2}\right)^{n-1}$. This content indicates that one of the sources of ill-conditioning is the eigenvalues λ_n [6, 14].

The stable method for Gaussian RBF interpolation in one dimension space

By applying the assumptions mentioned before, we now interpret the interpolation by the Gaussian RBF using Mercer’s series. According to Eq. (5), the Gaussian RBF approximation is as the following form

$$W(x) = \sum_{i=1}^N c_i e^{-\epsilon^2(x-x_i)^2}. \quad (11)$$

To use Mercer’s series, we have to truncate the expansion at the finite length L . If L is selected large enough, then interpolation by the truncated expansion converges to the one based on the full series [22, 41]. So, the Gaussian kernel can be approximated as follows

$$W_L(x) = \sum_{i=1}^N c_i \sum_{n=1}^L \lambda_n \varphi_n(x) \varphi_n(x_i) = \Phi_1^T(x) \Lambda \Phi_1 \mathbf{C}, \quad (12)$$

, where $\Phi_1^T(x) = (\varphi_1(x), \dots, \varphi_N(x))$, $\mathbf{C} = (c_1, \dots, c_N)^T$,

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix}, \quad \Phi_1 = \begin{pmatrix} \varphi_1(x_1) & \dots & \varphi_1(x_N) \\ \vdots & & \vdots \\ \varphi_N(x_1) & \dots & \varphi_N(x_N) \end{pmatrix}.$$

To obtain unknown coefficients c_i , the linear system of Eqs. (6) is written as follows

$$\mathbf{F} = \Phi_1^T \Lambda \Phi_1 \mathbf{C},$$

then

$$\mathbf{C} = \Phi_1^{-1} \Lambda^{-1} \Phi_1^{-T} \mathbf{F}. \quad (13)$$

By substituting (13) into (12) we achieve

$$W_L(x) = \Phi_1^T(x) \Phi_1^{-T} \mathbf{F}. \quad (14)$$

This stable interpolation function is called “interpolant with the cardinal basis” which lacks the matrix Λ as one of the sources of ill-conditioning.

Theorem 3 [41] Let

$$W(x) = \sum_{i=1}^N c_i \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(x_i),$$

and W_L be in the form of Eq. (12). Then,

$$|W(x) - W_L(x)| \leq \sqrt{\frac{2}{\pi L}} \frac{NC\alpha\beta}{\sqrt{\alpha^2 + \delta^2 + \epsilon^2}} e^{\alpha^2 b^2} \frac{E^L}{1 - E},$$

when $L \rightarrow \infty$ then $|W(x) - W_L(x)| \rightarrow 0$, where $C = \max\{|c_j|, j = 1, \dots, N\}$, and

$$E = \frac{\epsilon^2}{\alpha^2 + \epsilon^2 + \delta^2}.$$

Legendre–Gauss–Lobatto nodes and weights

Let $\{P_i\}_{i \geq 1}$ be Legendre polynomial functions of order i which are orthogonal with respect to the weight function $\omega(x) = 1$, on the interval $[-1, 1]$. These polynomials satisfy in the following formulas

$$P_1(x) = 1, \quad P_2(x) = x,$$

$$P_{i+1}(x) = \left(\frac{2i+1}{i+1} \right) x P_i(x) - \left(\frac{i}{i+1} \right) P_{i-1}(x), \quad i = 2, 3, \dots$$

The Legendre–Gauss–Lobatto nodes can be found as follows

$$(1 - x^2) P'_N(x_j) = 0, \quad (15)$$

$$-1 = x_1 < x_2 < \dots < x_N = 1,$$

, where $P'_i(x)$ is the derivative of $P_i(x)$. No explicit formula for the nodes $\{x_j\}_{j=2}^{N-1}$ is known, and so they are computed numerically using subroutines [9, 10]. The integral approximation of function f on the interval $[-1, 1]$ is

$$\int_{-1}^1 f(x)dx = \sum_{j=1}^N w_j f(x_j) + E_f, \tag{16}$$

, where w_j are Legendre–Gauss–Lobatto weights presented in [44]

$$w_j = \frac{2}{N(N-1)} \frac{1}{(P_{N-1}(x_j))^2}, \quad j = 1, 2, \dots, N,$$

and E_f is the error term specified by

$$E_f = -\frac{N(N-1)^3 2^{2N-1} [(N-2)!]^4}{(2N-1)[(2N-2)!]^3} f^{(2N-2)}(\zeta),$$

$$\zeta \in (-1, 1).$$

Also, if $f(x)$ is a polynomial of degree $\leq 2N - 3$ then the integration in Eq. (16) is exact.

Method of solution

The main aim of this section is to solve high-order linear Volterra–Fredholm integro-differential equations in the form (1) with the conditions (2). First of all, we consider the grid points $\{x_i\}_{i=1}^N$ into integral domain $[a, b]$. Then, the unknown solution of equation can be approximated by GA-RBFs as follows

$$y(x) = \Phi_1^T(x) \Phi_1^{-T} \mathbf{S}, \tag{17}$$

, where $\mathbf{S} = (s(x_1), \dots, s(x_N))^T$ is an unknown vector, $\Phi_1^T(x)$ and Φ_1 are defined in the Sect. 2.3. By substituting Eq. (17) into Eqs. (1) and (2), we get

$$\sum_{k=0}^m p_k(x) (\Phi_1^{(k)}(x))^T \Phi_1^{-T} \mathbf{S}$$

$$+ \theta_1 \int_a^b k_1(x, t) (\Phi_1^{(l_1)}(t))^T \Phi_1^{-T} \mathbf{S} dt$$

$$+ \theta_2 \int_a^x k_2(x, t) (\Phi_1^{(l_2)}(t))^T \Phi_1^{-T} \mathbf{S} dt = f(x), \tag{18}$$

and

$$\left[\sum_{k=0}^m p_k(x_i) \Phi_1^{(k)}(x_i) + \theta_1 \frac{b-a}{2} \sum_{j=1}^M w_{1j} k_1 \left(x_i, \frac{b-a}{2} \tau_{1j} + \frac{b+a}{2} \right) \Phi_1^{(l_1)} \left(\frac{b-a}{2} \tau_{1j} + \frac{b+a}{2} \right) \right. \\ \left. + \theta_2 \frac{x_i-a}{2} \sum_{j=1}^M w_{2j} k_2 \left(x_i, \frac{x_i-a}{2} \tau_{2j} + \frac{x_i+a}{2} \right) \Phi_1^{(l_2)} \left(\frac{x_i-a}{2} \tau_{2j} + \frac{x_i+a}{2} \right) \right]^T \Phi_1^{-T} \mathbf{S} \\ \simeq f(x_i), \quad i = 1, 2, \dots, N - m, \tag{21}$$

$$\sum_{k=0}^{m-1} \left(r_{jk} \Phi_1^{(k)}(a) + s_{jk} \Phi_1^{(k)}(b) \right)^T \Phi_1^{-T} \mathbf{S} = q_j, \tag{19}$$

$$j = 0, 1, 2, \dots, m - 1.$$

Now, by collocating Eq. (18) at interior points $\{x_i\}_{i=1}^{N-m}$ we have

$$\sum_{k=0}^m p_k(x_i) (\Phi_1^{(k)}(x_i))^T \Phi_1^{-T} \mathbf{S}$$

$$+ \theta_1 \int_a^b k_1(x_i, t) (\Phi_1^{(l_1)}(t))^T \Phi_1^{-T} \mathbf{S} dt$$

$$+ \theta_2 \int_a^{x_i} k_2(x_i, t) (\Phi_1^{(l_2)}(t))^T \Phi_1^{-T} \mathbf{S} dt = f(x_i),$$

$$i = 1, 2, \dots, N - m.$$

To use the Legendre–Gauss–Lobatto rule for Eq. (20), we apply the following transformations

$$t = \frac{b-a}{2} \tau_1 + \frac{b+a}{2}, \quad \tau_1 \in [-1, 1], \quad \text{for Fredholm,}$$

$$t = \frac{x_i-a}{2} \tau_2 + \frac{x_i+a}{2}, \quad \tau_2 \in [-1, 1], \quad \text{for Volterra.}$$

So,

$$\sum_{k=0}^m p_k(x_i) (\Phi_1^{(k)}(x_i))^T \Phi_1^{-T} \mathbf{S}$$

$$+ \theta_1 \frac{b-a}{2} \int_{-1}^1 k_1 \left(x_i, \frac{b-a}{2} \tau_1 + \frac{b+a}{2} \right) \\ \left(\Phi_1^{(l_1)} \left(\frac{b-a}{2} \tau_1 + \frac{b+a}{2} \right) \right)^T \Phi_1^{-T} \mathbf{S} d\tau_1$$

$$+ \theta_2 \frac{x_i-a}{2} \int_{-1}^1 k_2 \left(x_i, \frac{x_i-a}{2} \tau_2 + \frac{x_i+a}{2} \right) \\ \left(\Phi_1^{(l_2)} \left(\frac{x_i-a}{2} \tau_2 + \frac{x_i+a}{2} \right) \right)^T \Phi_1^{-T} \mathbf{S} d\tau_2 \\ = f(x_i), \quad i = 1, 2, \dots, N - m,$$

and by using (16) we write

, where $\tau_{\beta j}$ and $w_{\beta j}$, ($\beta = 1, 2$) are the Legendre–Gauss–Lobatto nodes and weights into $[-1, 1]$, respectively. Finally, an $N \times N$ system of linear algebraic equations is produced such that the $N - m$ first row is related to Eq. (21) and m remained row is related to Eq. (19). By use of the `linsolve` command in MATLAB, the value of unknown vector \mathbf{S} and consequently the solution $y(x)$ is obtained.

Error analysis

In this section, we try to find an upper bound for the residual error when the presented method is derived for (1). Suppose that y and \tilde{y} are the exact solution and the approximate one, respectively. Define

$$RE(x) = \sum_{k=0}^m p_k(x)(y^{(k)}(x) - \tilde{y}^{(k)}(x)) + \theta_1 \int_a^b k_1(x, t)(y^{(l_1)}(t) - \tilde{y}^{(l_1)}(t))dt + \theta_2 \int_a^x k_2(x, t)(y^{(l_2)}(t) - \tilde{y}^{(l_2)}(t))dt.$$

The main result of this section is formulated through the following theorem.

Theorem 4 Assume $M_p, M, R_1, R_2, R'_1, R'_2$ are real positive numbers and

$$E = \frac{\epsilon^2}{\alpha^2 + \epsilon^2 + \delta^2},$$

$$R(N) = \frac{N(N-1)^3 2^{2N-1} [(N-2)!]^4}{(2N-1)[(2N-2)!]^3}.$$

For the residual error function, that is $RE(x)$, we have the following bound

$$|RE(x)| \leq M_p \sum_{k=0}^m T_k + |\theta_1| \left(\frac{b-a}{2} NR_1 T_{l_1} + R(N) R'_1 \right) + |\theta_2| \left(\frac{b-a}{2} NR_2 T_{l_2} + R(N) R'_2 \right),$$

in which

$$T_j = \frac{NcM\beta\alpha\epsilon^{\alpha^2 b^2} E^L}{\sqrt{\alpha^2 + \epsilon^2 + \delta^2} 2^L \left(\frac{L}{2}\right)!} \sum_{n=L+1}^{\infty} \sum_{i=0}^j \binom{j}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!}, \quad j = k, l_1, l_2.$$

Proof Using (1) one can write

$$RE(x) = \sum_{k=0}^m p_k(x)(y^{(k)}(x) - \tilde{y}^{(k)}(x)) + \theta_1 \int_a^b k_1(x, t)(y^{(l_1)}(t) - \tilde{y}^{(l_1)}(t))dt + \theta_2 \int_a^x k_2(x, t)(y^{(l_2)}(t) - \tilde{y}^{(l_2)}(t))dt. \tag{22}$$

We define $e(x) = y(x) - \tilde{y}(x)$. By utilizing (16), it is obvious

$$\int_a^b k_1(x, t)e^{(l_1)}(t)dt = \frac{b-a}{2} \int_{-1}^1 k_1\left(x, \frac{b-a}{2}z + \frac{b+a}{2}\right)e^{(l_1)}\left(\frac{b-a}{2}z + \frac{b+a}{2}\right)dz = \frac{b-a}{2} \sum_{j=1}^N \omega_j k_1\left(x, \frac{b-a}{2}z_j + \frac{b+a}{2}\right)e^{(l_1)}\left(\frac{b-a}{2}z_j + \frac{b+a}{2}\right) + E_{k_1}, \tag{23}$$

, where there exists an $\zeta \in (-1, 1)$ so that

$$E_{k_1} = -\frac{N(N-1)^3 2^{2N-1} [(N-2)!]^4}{(2N-1)[(2N-2)!]^3} \times \frac{\partial^{2N-2}}{\partial \zeta^{2N-2}} \left(k_1\left(x, \frac{b-a}{2}\zeta + \frac{b+a}{2}\right)e^{(l_1)}\left(\frac{b-a}{2}\zeta + \frac{b+a}{2}\right) \right).$$

Moreover, for each $a \leq x \leq b$

$$\left| \int_a^x k_2(x, t)e^{(l_2)}(t)dt \right| \leq \int_a^b \left| k_2(x, t)e^{(l_2)}(t) \right| dt = \frac{b-a}{2} \int_{-1}^1 \left| k_2\left(x, \frac{b-a}{2}z + \frac{b+a}{2}\right)e^{(l_2)}\left(\frac{b-a}{2}z + \frac{b+a}{2}\right) \right| dz = \frac{b-a}{2} \sum_{j=1}^N \omega'_j \left| k_2\left(x, \frac{b-a}{2}z_j + \frac{b+a}{2}\right)e^{(l_2)}\left(\frac{b-a}{2}z_j + \frac{b+a}{2}\right) \right| + E_{k_2}, \tag{24}$$

, where there exists an $\eta \in (-1, 1)$ so that

$$E_{k_2} = \frac{N(N-1)^3 2^{2N-1} [(N-2)!]^4}{(2N-1)[(2N-2)!]^3} \times \frac{\partial^{2N-2}}{\partial \eta^{2N-2}} \left| k_2\left(x, \frac{b-a}{2}\eta + \frac{b+a}{2}\right)e^{(l_2)}\left(\frac{b-a}{2}\eta + \frac{b+a}{2}\right) \right|.$$

On the other hand, for every $0 \leq k \leq m$ one obtains

$$|y^{(k)}(x) - \bar{y}^{(k)}(x)| = \left| \sum_{j=1}^N c_j \sum_{n=L+1}^{\infty} \lambda_n \varphi_n^{(k)}(x) \varphi_n(x_j) \right| \leq \sum_{j=1}^N |c_j| \sum_{n=L+1}^{\infty} |\lambda_n| |\varphi_n^{(k)}(x) \varphi_n(x_j)|.$$

If $c = \max \{ |c_j| : j = 1, \dots, N \}$ then we have

$$|e^{(k)}(x)| \leq c \sum_{j=1}^N \sum_{n=L+1}^{\infty} |\lambda_n| |\varphi_n^{(k)}(x) \varphi_n(x_j)|. \tag{25}$$

Since

$$\varphi_n(x) = \sqrt{\frac{\beta}{2^{n-1}(n-1)!}} e^{-\delta^2 x^2} H_{n-1}(\alpha\beta x),$$

we can write

$$\varphi_n^{(k)}(x) = \sqrt{\frac{\beta}{2^{n-1}(n-1)!}} \sum_{i=0}^k \binom{k}{i} \frac{d^{k-i}}{dx^{k-i}} e^{-\delta^2 x^2} \frac{d^i}{dx^i} H_{n-1}(\alpha\beta x).$$

Because of $\frac{d}{dx} e^{-\delta^2 x^2} = -2\delta^2 x e^{-\delta^2 x^2}$, for every $0 \leq i \leq k$ there exists an $M_i > 0$ on the interval $[a, b]$ such that

$$\left| \frac{d^{k-i}}{dx^{k-i}} e^{-\delta^2 x^2} \right| \leq M_i e^{-\delta^2 x^2}.$$

Putting $M = \max \{ M_i : i = 0, 1, \dots, k \}$ enables one to conclude

$$\begin{aligned} |\varphi_n^{(k)}(x)| &\leq M \sqrt{\frac{\beta}{2^{n-1}(n-1)!}} e^{-\delta^2 x^2} \sum_{i=0}^k \binom{k}{i} \left| \frac{d^i}{dx^i} H_{n-1}(\alpha\beta x) \right| \\ &= M \sqrt{\frac{\beta}{2^{n-1}(n-1)!}} e^{-\delta^2 x^2} \sum_{i=0}^k \binom{k}{i} \frac{(2\alpha\beta)^i n!}{(n-1-i)!} |H_{n-1-i}(\alpha\beta x)| \end{aligned}$$

According to the asymptotic expansion of Hermite polynomials [41],

$$H_n(x) \sim e^{\frac{x^2}{2}} \frac{n!}{\left(\frac{n}{2}\right)!} \cos\left(x\sqrt{2n+1} - \frac{n\pi}{2}\right), \quad \text{for large } n,$$

, it is easy to see that

$$\begin{aligned} |\varphi_n^{(k)}(x)| &\leq M \sqrt{\frac{\beta}{2^{n-1}(n-1)!}} e^{\left(\frac{\alpha^2 \beta^2}{2} - \delta^2\right)x^2} \\ &\quad \times \sum_{i=0}^k \binom{k}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!} \\ &\quad \left| \cos\left(x\sqrt{2n-2i-1} - \frac{\pi}{2}(n-1-i)\right) \right|. \end{aligned}$$

Let $E = \frac{\epsilon^2}{\alpha^2 + \epsilon^2 + \delta^2} < 1$. From (25), we determine

$$\begin{aligned} |e^{(k)}(x)| &\leq \frac{NcM\beta\alpha e^{\alpha^2 b^2}}{\sqrt{\alpha^2 + \epsilon^2 + \delta^2}} \sum_{n=L+1}^{\infty} \frac{E^{n-1}}{2^{n-1} \left(\frac{n-1}{2}\right)!} \\ &\quad \sum_{i=0}^k \binom{k}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!} \\ &\leq \frac{NcM\beta\alpha e^{\alpha^2 b^2} E^L}{\sqrt{\alpha^2 + \epsilon^2 + \delta^2} 2^L \left(\frac{L}{2}\right)!} \\ &\quad \sum_{n=L+1}^{\infty} \sum_{i=0}^k \binom{k}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!}. \end{aligned}$$

Subsequently, if we assume $M' = \frac{cM\beta\alpha e^{\alpha^2 b^2}}{\sqrt{\alpha^2 + \epsilon^2 + \delta^2}}$ it is acquired

$$|e^{(k)}(x)| \leq \frac{M'NE^L}{2^L \left(\frac{L}{2}\right)!} \sum_{n=L+1}^{\infty} \sum_{i=0}^k \binom{k}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!} = T_k. \tag{26}$$

The inequality (26) shows that when $L \rightarrow \infty$ then $e^{(k)}(x)$ converges to zero. Now, suppose that

$$\begin{aligned} R_1 &= \max_{1 \leq j \leq N} \left| \omega_j k_1 \left(x, \frac{b-a}{2} z_j + \frac{b+a}{2} \right) \right|, \\ R_2 &= \max_{1 \leq j \leq N} \left| \omega'_j k_2 \left(x, \frac{b-a}{2} z_j + \frac{b+a}{2} \right) \right|, \\ R'_1 &= \max_{x, \zeta} \frac{\partial^{2N-2}}{\partial \zeta^{2N-2}} \left| k_1 \left(x, \frac{b-a}{2} \zeta + \frac{b+a}{2} \right) \right| e^{(l_1)} \\ &\quad \left(\frac{b-a}{2} \zeta + \frac{b+a}{2} \right), \\ R'_2 &= \max_{x, \eta} \frac{\partial^{2N-2}}{\partial \eta^{2N-2}} \left| k_2 \left(x, \frac{b-a}{2} \eta + \frac{b+a}{2} \right) \right| e^{(l_2)} \\ &\quad \left(\frac{b-a}{2} \eta + \frac{b+a}{2} \right), \\ R(N) &= \frac{N(N-1)^3 2^{2N-1} [(N-2)!]^4}{(2N-1)[(2N-2)!]^3}. \end{aligned}$$

By these notations and deriving (23), (24), we realize

$$\begin{aligned}
 & \left| \int_a^b k_1(x, t)e^{(l_1)}(t)dt \right| \\
 & \leq \frac{b-a}{2} R_1 \frac{M'N^2E^L}{2^L \left(\frac{L}{2}\right)!} \\
 & \quad \sum_{n=L+1}^{\infty} \sum_{i=0}^{l_1} \binom{l_1}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!} + R'_1 R(N), \\
 & \left| \int_a^b k_2(x, t)e^{(l_2)}(t)dt \right| \\
 & \leq \frac{b-a}{2} R_2 \frac{M'N^2E^L}{2^L \left(\frac{L}{2}\right)!} \\
 & \quad \sum_{n=L+1}^{\infty} \sum_{i=0}^{l_2} \binom{l_2}{i} \frac{(2\alpha\beta)^i n!}{\left(\frac{n-i-1}{2}\right)!} + R'_2 R(N).
 \end{aligned} \tag{27}$$

If we set $M_p = \max \{ |p_k(x)| : k = 0, 1, \dots, m \}$, by applying (22), (23) and (27) the expected result is achieved. \square

Numerical examples

In this section, several examples are given to assess the performance of present method. A uniform data set from size N in one dimension is considered. The number of Legendre–Gauss–Lobatto nodes is denoted by M . Assume that y and \tilde{y} refer to the exact solution and the approximate one, respectively. The accuracy of method is measured by applying the absolute error, namely $|e(x)|$, the maximum absolute error (L_∞ error) and the Root-Mean-Square error (RMS error) overall N data points as follows:

$$L_\infty = \max_{a \leq x_i \leq b} |e(x_i)|, \quad i = 1, \dots, N, \tag{28}$$

$$RMS = \sqrt{\frac{\sum_{i=1}^N e^2(x_i)}{N}}. \tag{29}$$

All of the numerical computations are performed in MATLAB R2015a, on a laptop with an Intel(R) Core(TM) i5-6200U, CPU 2.40GHz, 8GB(RAM).

Example 1 Let us consider linear Volterra IDE of the first kind

$$\int_0^x \cos(x-t)y''(t)dt = 2 \sin(x), \quad 0 \leq x \leq 1,$$

with the initial conditions $y(0) = y'(0) = 0$. The exact solution for this equation is $y(x) = x^2$.

Put $M = 4, \epsilon = 0.4$ and $\alpha = 0.1$. The comparison of RMS error, L_∞ error and condition number for different values of N between the present method and the standard GA-RBF one are given in Table 2. It indicates that the error values and condition numbers of present method are better than those of the standard one. Because the eigenfunctions φ_n are orthogonal, so the matrix Φ_1 in Eq. (14) is relatively well behaved. Also, by removing the matrix Λ into (14), as one of the sources of ill-conditioning for the small values of ϵ , the present method is more stable, and the condition number is significantly less than that of classical GA-RBF. Figure 1 displays the absolute error as a function of $x \in [0, 1]$ for $N = \{4, 6, 11\}$ which decreases when the number of data points increases. The RMS error as a function of $\epsilon \in [0.01, 1]$ for $N = \{4, 6, 11\}$ is shown in Figure 2. It can be seen that for small ϵ the RMS error value is less for $N = \{4, 6\}$.

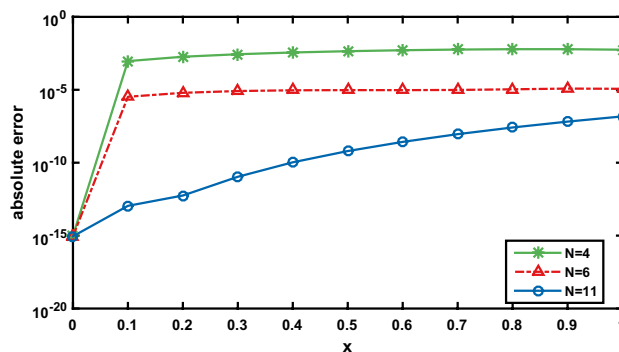


Fig. 1 Absolute error of present method as a function of x for different values of N for Example 1

Table 2 Comparison between the present method and the standard GA-RBF one with $M = 4, \epsilon = 0.4$ and $\alpha = 0.1$ for Example 1

Method		$N = 4$	$N = 6$	$N = 11$
Present method	RMS error	4.2206e-03	8.8118e-06	5.0092e-08
	L_∞ error	5.6047e-03	1.1656e-05	1.5002e-07
	Condition number	42.3291	251.1209	5.1999e+05
Standard GA-RBF	RMS error	1.2931e-02	1.2942e-04	7.8357e-07
	L_∞ error	1.8719e-02	1.5548e-04	1.0705e-06
	Condition number	1.8794e+04	1.3602e+09	7.6208e+17

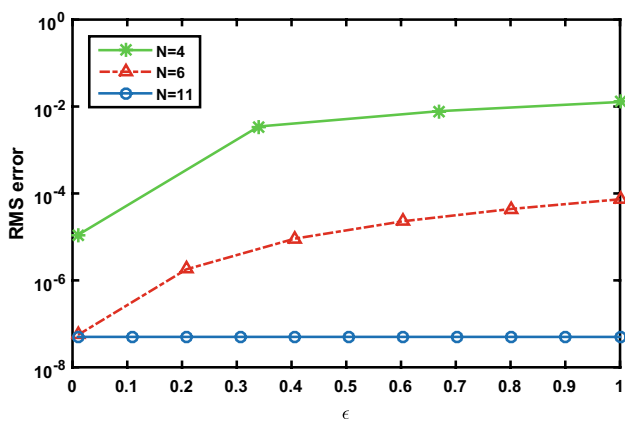


Fig. 2 RMS error of present method as a function of ϵ for different values of N for Example 1

Example 2 Consider first-order Volterra IDE

$$y' = 1 - 2x \sin(x) + \int_0^x y(t)dt, \quad y(0) = 0, \quad 0 \leq x \leq 1.$$

The exact solution is $y(x) = x \cos(x)$.

We use $M = 5$, $\epsilon = 0.09$ and $\alpha = 1.9$. The comparison of RMS error for different values of N between the present method and the methods introduced by [4] are expressed in Table 3. It shows that the error values in the present method are better than those of [4]. Table 4 declares the RMS error results of present method, L_∞ error and condition number for different values of N . Figure 3 displays the absolute error as a function of $x \in [0, 1]$ for $N = \{5, 8, 12\}$, which decreases by increasing the number of data points. The RMS error as a function of $\epsilon \in [0.01, 2]$ for $N = \{5, 8, 12\}$ is shown in Fig. 4.

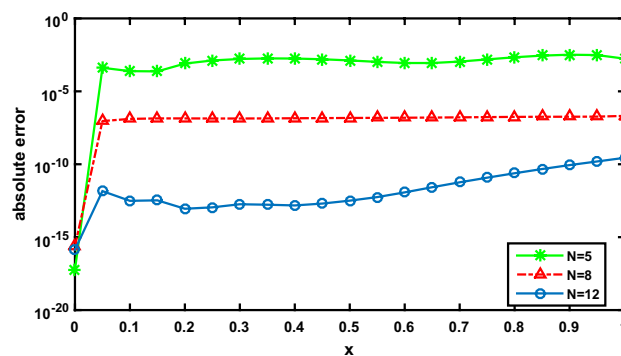


Fig. 3 Absolute error of present method as a function of x for different values of N for Example 2

Example 3 Let us consider linear Fredholm IDE

$$\begin{aligned} y''(x) + xy'(x) - xy(x) &= e^x - 2 \sin(x) \\ &+ \int_{-1}^1 \sin(x)e^{-t}y(t)dt, \quad -1 \leq x \leq 1, \end{aligned}$$

under the initial conditions $y(0) = 1$ and $y'(0) = 1$. Here, the exact solution is $y(x) = e^x$.

Assume $M = 4$, $\epsilon = 0.01$ and $\alpha = 1.9$. The comparison of absolute errors on the interval $[-1, 1]$ between the present method and the method described by [21] are expressed in Table 5. It indicates that the absolute error values of present method for $N = 13$ are better than those of described in [21] for $N = 1000$. The RMS error in [21] for $N = 1000$ is $2.87486e-08$, whereas the RMS error of our method is $4.1029e-11$ for $N = 13$. Table 6 presents the RMS error results, L_∞ error and condition number of present method for different values of N . Figure 5 displays the absolute error as a function of $x \in [-1, 1]$ for $N = \{6, 8, 11, 13\}$, which

Table 3 Comparison between the present method and the methods of [4] with $M = 5$, $\epsilon = 0.09$ and $\alpha = 1.9$ for Example 2

Method		$N = 10$	$N = 15$	$N = 20$
Present method	RMS error	1.7685e-10	7.9087e-11	7.3685e-11
GRBF of [3] for GA-RBF	RMS error	2.8852e-07	2.6959e-07	9.0134e-07
IGRBF of [3] for GA-RBF	RMS error	4.1743e-08	1.9768e-07	3.0391e-07
GRBF of [3] for MQ-RBF	RMS error	3.2830e-07	1.9996e-06	4.7677e-07
IGRBF of [3] for MQ-RBF	RMS error	3.4020e-08	8.0225e-07	1.0683e-07
GRBF of [3] for IMQ-RBF	RMS error	8.0619e-07	3.6005e-07	1.1740e-06
IGRBF of [3] for IMQ-RBF	RMS error	8.9479e-08	6.0615e-08	2.5984e-07

Table 4 Numerical results with $M = 5$, $\epsilon = 0.09$ and $\alpha = 1.9$ for Example 2

Method		$N = 5$	$N = 8$	$N = 12$
Present method	RMS error	1.3318e-03	1.5216e-07	8.4876e-11
	L_∞ error	1.7431e-03	2.0642e-07	2.7553e-10
	condition number	81.7126	1.3223e+03	1.5774e+05

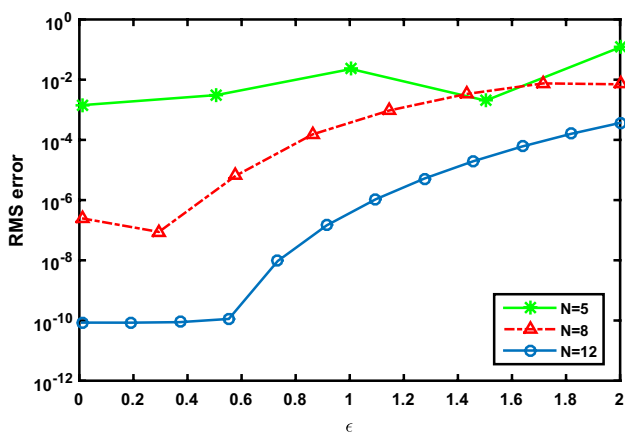


Fig. 4 RMS error of present method as a function of ϵ for different values of N for Example 2

Table 5 Comparison of the absolute errors between the present method and the method of [21] with $M = 4$, $\epsilon = 0.01$ and $\alpha = 1.9$ for Example 3

x	Present method $N = 13$	MQ-RBF of [20] $N = 1000$
-1	1.4744e-10	8.89585e-08
-0.8	6.8639e-10	4.64598e-08
-0.6	1.6234e-10	1.21271e-08
-0.4	4.4214e-11	7.59065e-09
-0.2	1.3468e-11	4.60614e-09
0	0	0
0.2	1.3466e-11	1.85311e-09
0.4	44190e-11	2.70376e-09
0.6	1.6226e-10	5.7854e-09
0.8	68395e-10	2.50214e-08
1	9.5750e-12	8.16448e-08

decreases by increasing the number of data points. The RMS error as a function of $\epsilon \in [0.01, 1.5]$ for $N = \{6, 8, 11, 13\}$ is shown in Fig. 6.

Example 4 We consider eight-order Fredholm IDE with initial conditions as follows

Table 6 Numerical results with $M = 4$, $\epsilon = 0.01$ and $\alpha = 1.9$ for Example 3

Method		$N = 6$	$N = 8$	$N = 11$	$N = 13$
Present method	RMS error	2.6653e-03	2.6575e-05	1.1307e-08	4.1029e-11
	L_∞ error	6.5134e-03	7.4971e-05	3.7386e-08	1.4744e-10
	condition number	484.6061	4.1017e+03	1.8281e+05	2.4998e+06

$$y^{(8)}(x) = -8e^x + x^2 + y(x) + \int_0^1 x^2 y'(t) dt, \quad 0 \leq x \leq 1,$$

$$y(0) = 1, y'(0) = 0, y''(0) = -1, y'''(0) = -2,$$

$$y^{(4)}(0) = -3, y^{(5)}(0) = -4, y^{(6)}(0) = -5, y^{(7)}(0) = -6.$$

The exact solution is $y(x) = (1 - x)e^x$.

Let $M = 4$, $\epsilon = 0.01$ and $\alpha = 0.9$. Table 7 announces the RMS error results, L_∞ error and condition number of present method for different values of N . Figure 7 portrays the absolute error as a function of $x \in [0, 1]$ for $N = \{6, 10, 12, 13\}$, which decreases by increasing the number of data points. The RMS error as a function of $\epsilon \in [0.001, 1.5]$ for $N = \{6, 10, 12, 13\}$ is shown in Fig. 8.

Example 5 [31] Consider linear Fredholm–Volterra IDE

$$y'(x) = \frac{27}{5} - \frac{1}{5}x^6 - \frac{3}{4}x^5 - x^4 - \frac{1}{2}x^3 + 3x^2 + \frac{41}{20}x$$

$$+ \int_{-1}^1 (x - t)y(t)dt + \int_{-1}^x xy(t)dt, \quad -1 \leq x \leq 1,$$

with the initial condition $y(0) = 1$. The exact solution of this equation is $y(x) = (x + 1)^3$.

Put $M = 5$, $\epsilon = 0.01$ and $\alpha = 1.9$. Table 8 gives the RMS error results, L_∞ error and condition number of present method for different values of N . Figure 9 displays the absolute error as a function of $x \in [-1, 1]$ for $N = \{5, 7, 9, 10\}$, which decreases by increasing the number of data points. The RMS error as a function of $\epsilon \in [0.01, 1.5]$ for $N = \{5, 7, 9, 10\}$ is shown in Fig. 10.

Example 6 Consider fifth-order linear Fredholm–Volterra IDE with initial conditions given by

$$y^{(5)}(x) - xy^{(2)}(x) + xy(x) = -e^{-x} - \frac{1}{2}e^{2x} - x^2$$

$$+ \frac{1}{2} \int_0^1 e^{2x+t} y(t) dt + \int_0^x x e^t y(t) dt, \quad 0 \leq x \leq 1,$$

$$y(0) = y^{(2)}(0) = y^{(4)}(0) = 1, \quad y^{(1)}(0) = y^{(3)}(0) = -1.$$

The exact solution of the above equation is $y(x) = e^{-x}$.

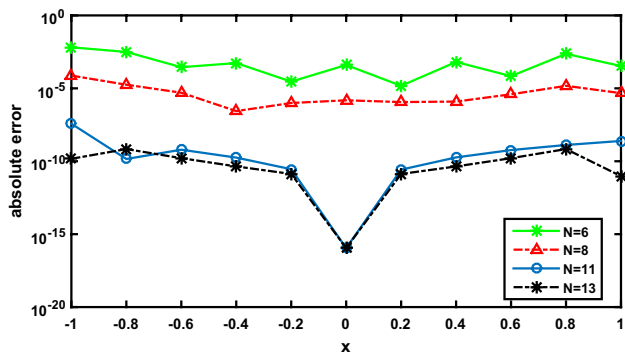


Fig. 5 Absolute error of present method as a function of x for different values of N for Example 3

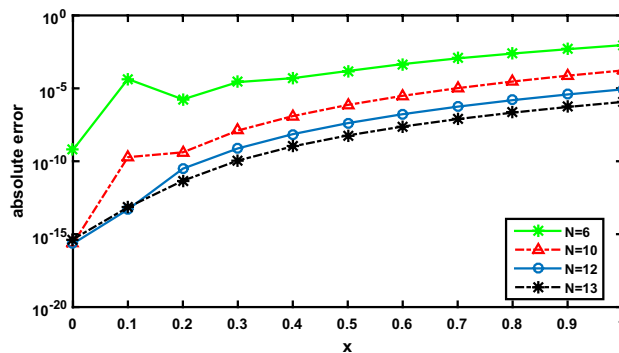


Fig. 7 Absolute error of present method as a function of x for different values of N for Example 4

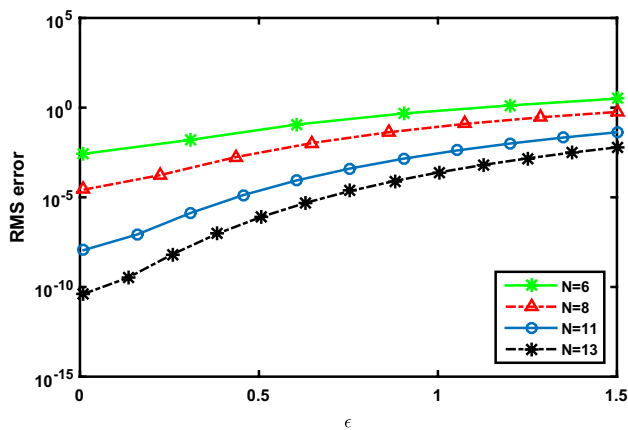


Fig. 6 RMS error of present method as a function of ϵ for different values of N for Example 3

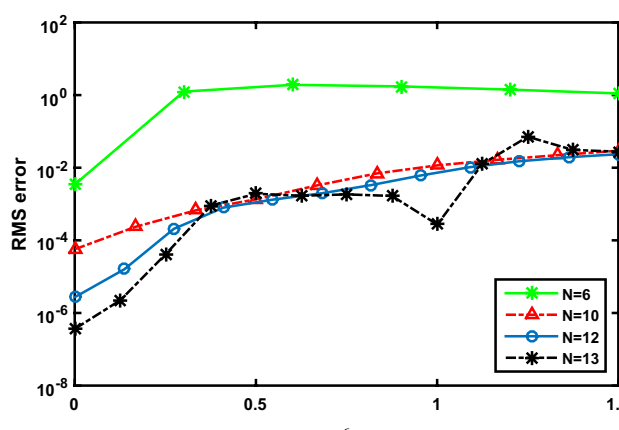


Fig. 8 RMS error of present method as a function of ϵ for different values of N for Example 4

Suppose $M = 4$, $\epsilon = 0.01$ and $\alpha = 1.9$. Table 9 declares the RMS error results, L_∞ error and condition number of present method for different values of N . Figure 11 displays the absolute error as a function of $x \in [0, 1]$ for $N = \{10, 11, 12, 13\}$, which decreases when the number

of data points increases. The RMS error as a function of $\epsilon \in [0.01, 2]$ for $N = \{10, 11, 12, 13\}$ is demonstrated in Fig. 12.

Table 7 Numerical results with $M = 4$, $\epsilon = 0.01$ and $\alpha = 0.9$ for Example 4

Method		$N = 6$	$N = 10$	$N = 12$	$N = 13$
Present method	RMS error	3.8878e-03	5.7652e-05	2.7817e-06	3.7550e-07
	L_∞ error	9.1944e-03	1.6777e-04	8.4842e-06	1.1661e-06
	condition number	1.5696e+05	1.9955e+11	1.9873e+13	1.3529e+14

Table 8 Numerical results with $M = 5$, $\epsilon = 0.01$ and $\alpha = 1.9$ for Example 5

Method		$N = 5$	$N = 7$	$N = 9$	$N = 10$
Present method	RMS error	5.1164e-05	1.2978e-09	2.0725e-14	3.8220e-15
	L_∞ error	1.1302e-04	3.3992e-09	6.1728e-14	1.1990e-14
	condition number	47.6453	190.9572	1.1873e+03	3.6906e+03

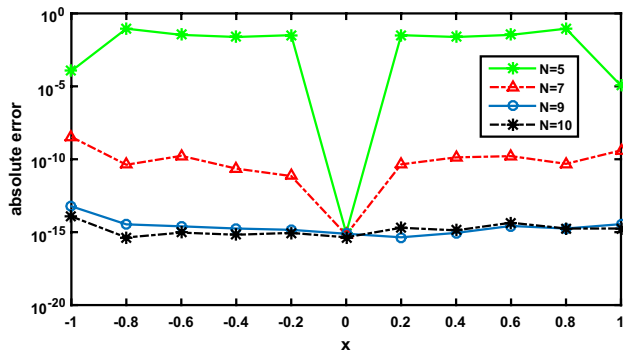


Fig. 9 Absolute error of the presented method as a function of x for different values of N for Example 5

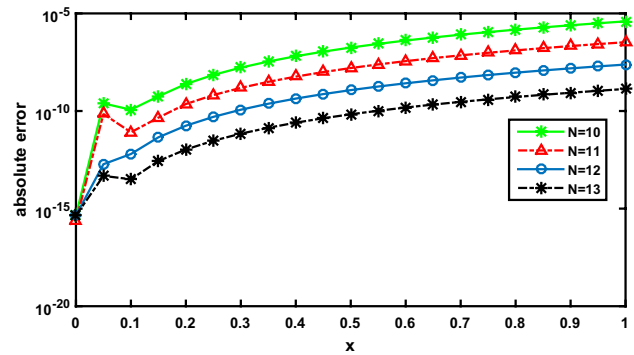


Fig. 11 Absolute error of present method as a function of x for different values of N for Example 6

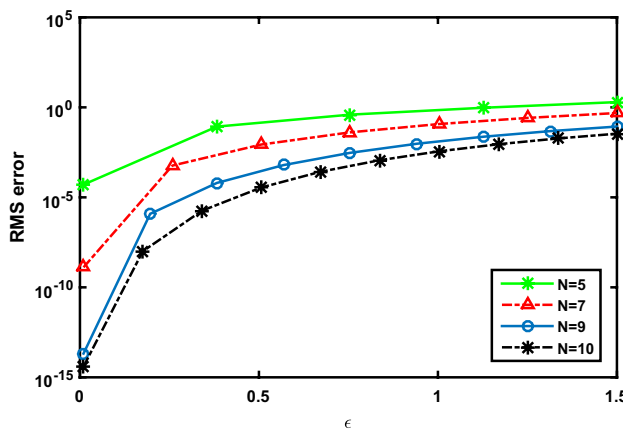


Fig. 10 RMS error of present method as a function of ϵ for different values of N for Example 5

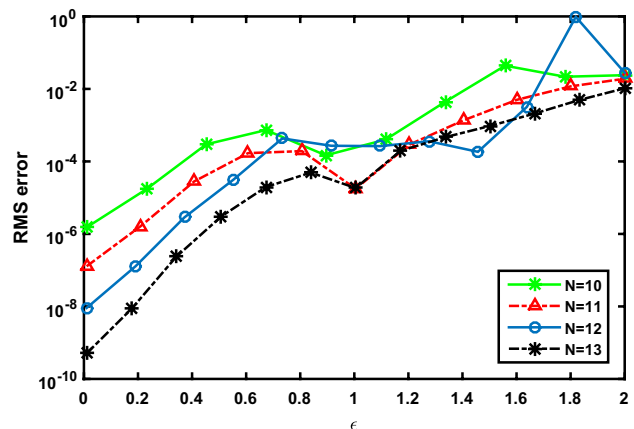


Fig. 12 RMS error of present method as a function of ϵ for different values of N for Example 6

Conclusion

We described an improved approach of Gaussian RBFs to solve high-order linear Volterra–Fredholm integro-differential equations. The small values of ϵ were well known to make highly accurate results, but the coefficient matrix of interpolation became ill-conditioned. We used very small values for ϵ and employed the eigenfunction expansion for GA-RBFs. The standard RBF did not work for illustrative

examples except for the first one. We did not mention the numerical results of standard RBF for those examples. In those cases, the impression of present version of RBF was seen. Generally, in Figures attributed to RMS error as a function of ϵ for fixed N , the RMS error increased by increasing ϵ . Also, according to Figures corresponding to the absolute error as a function of x , by increasing N the absolute error rate decreased. Besides, the condition number in the present method was small and acceptable compared to the standard GA-RBF one. These were reasons for the efficiency of present method.

Table 9 Numerical results with $M = 4$, $\epsilon = 0.01$ and $\alpha = 1.9$ for Example 6

Method		$N = 10$	$N = 11$	$N = 12$	$N = 13$
Present method	RMS error	1.5196e-06	1.2825e-07	9.0800e-09	5.0552e-10
	L_∞ error	3.9221e-06	3.3565e-07	2.4039e-08	1.3511e-09
	condition number	7.5229e+08	7.6671e+09	8.2520e+10	8.0290e+11

References

1. Abramowitz, M., Stegun, I.A.: Handbook of mathematical functions: with formulas, graphs, and mathematical tables. Courier Corporation, vol. 55. SIAM (1964)
2. Avazzadeh, Z., Heydari, M.H., Cattani, C.: Legendre wavelets for fractional partial integro-differential viscoelastic equations with weakly singular kernels. *Eur. Phys. J. Plus* **134**(7), 368 (2019)
3. Azarboni, H.R., Keyanpour, M., Yaghouti, M.: Leave-Two-Out Cross Validation to optimal shape parameter in radial basis functions. *Eng. Anal. Bound. Elements* **100**, 204–210 (2019)
4. Biazar, J., Asadi, M.A.: Galerkin RBF for integro-differential equations. *British J. Math. Comput. Sci.* **11**(2), 1–9 (2015)
5. Carlson, R.E., Foley, T.A.: The parameter R^2 in multiquadric interpolation. *Comput. Math. Appl.* **21**(9), 29–42 (1991)
6. Cavoretto, R., Fasshauer, G.E., McCourt, M.: An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels. *Numer. Algorithms* **68**(2), 393–422 (2015)
7. Chen, J., He, M., Zeng, T.: A multiscale Galerkin method for second-order boundary value problems of Fredholm integro-differential equation II: Efficient algorithm for the discrete linear system. *J. Visual Commun. Image Represent.* **58**, 112–118 (2019)
8. Dehghan, M., Shokri, A.: A meshless method for numerical solution of the one-dimensional wave equation with an integral condition using radial basis functions. *Numer. Algorithms* **52**(3), 461 (2009)
9. Elnagar, G.N., Kazemi, M.A.: Pseudospectral Legendre-based optimal computation of nonlinear constrained variational problems. *J. Comput. Appl. Math.* **88**(2), 363–375 (1998)
10. Elnagar, G.N., Razzaghi, M.: A collocation-type method for linear quadratic optimal control problems. *Opt. Control Appl. Methods* **18**(3), 227–235 (1997)
11. Erfanian, M., Mansoori, A.: Solving the nonlinear integro-differential equation in complex plane with rationalized Haar wavelet. *Math. Comput. Simul.* **165**, 223–237 (2019)
12. Fakhr Kazemi, B., Jafari, H.: Error estimate of the MQ-RBF collocation method for fractional differential equations with Caputo-Fabrizio derivative. *Math. Sci.* **11**(4), 297–305 (2017)
13. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB, vol. 6. World Scientific, Singapore (2007)
14. Fasshauer, G.E., McCourt, M.J.: Stable evaluation of Gaussian radial basis function interpolants. *SIAM J. Sci. Comput.* **34**(2), A737–A762 (2012)
15. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.* **33**(2), 869–892 (2011)
16. Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.* **30**(1), 60–80 (2008)
17. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.* **48**(5–6), 853–867 (2004)
18. Franke, R.: A critical comparison of some methods for interpolation of scattered data. Tech. rep, Naval Postgraduate School Monterey (1979)
19. Franke, R.: Scattered data interpolation: tests of some methods. *Math. Comput.* **38**(157), 181–200 (1982)
20. Gallas, B., Barrett, H.H.: Modeling all orders of scatter in nuclear medicine. In: 1998 IEEE Nuclear Science Symposium Conference Record. 1998 IEEE Nuclear Science Symposium and Medical Imaging Conference (Cat. No. 98CH36255), vol. 3, pp. 1964–1968. IEEE (1998)
21. Golbabai, A., Seifollahi, S.: Radial basis function networks in the numerical solution of linear integro-differential equations. *Appl. Math. Comput.* **188**(1), 427–432 (2007)
22. Griebel, M., Rieger, C., Zwicknagl, B.: Multiscale approximation and reproducing kernel Hilbert space methods. *SIAM J. Numer. Anal.* **53**(2), 852–873 (2015)
23. Grigoriev, Y.N., Kovalev, V.F., Meleshko, S.V., Ibragimov, N.H.: Symmetries of Integro-Differential Equations: With Applications in Mechanics and Plasma Physics. Springer, Berlin (2010)
24. Hamoud, A., Ghadle, K.: Homotopy analysis method for the first order fuzzy Volterra-Fredholm integro-differential equations. *Ind. J. Electr. Eng. Comput. Sci.* **11**(3), 857–867 (2018)
25. Hamoud, A.A., Ghadle, K.P.: The combined Modified Laplace with Adomian decomposition method for Solving the nonlinear Volterra-Fredholm Integro Differential Equations. *J. Korean Soc. Ind. Appl. Math.* **21**(1), 17–28 (2017)
26. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **76**(8), 1905–1915 (1971)
27. Heydari, M.H., Laeli Dastjerdi, H., Nili Ahmadabadi, M.: An efficient method for the numerical solution of a class of nonlinear fractional fredholm integro-differential equations. *Int. J. Nonlinear Sci. Numer. Simul.* **19**(2), 165–173 (2018)
28. Heydari, M.H., Hosseininia, M.: A new variable-order fractional derivative with non-singular Mittag-Leffler kernel: application to variable-order fractional version of the 2D Richard equation. *Eng. Comput.* 1–12 (2020)
29. Hendi, F., Al-Qarni, M.: The variational Adomian decomposition method for solving nonlinear two-dimensional Volterra-Fredholm integro-differential equation. *J. King Saud Univ. Sci.* **31**(1), 110–113 (2019)
30. Hosseininia, M., Heydari, M.H., Avazzadeh, Z., Maalek Ghaini, F.M.: A hybrid method based on the orthogonal Bernoulli polynomials and radial basis functions for variable order fractional reaction-advection-diffusion equation. *Eng. Anal. Bound. Elements* **127**, 18–28 (2021)
31. İşler Acar, N., Daşcıoğlu, A.: A projection method for linear Fredholm-Volterra integro-differential equations. *J. Taibah Univ. Sci.* **13**(1), 644–650 (2019)
32. Kansa, E.J.: Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-i surface approximations and partial derivative estimates. *Comput. Math. Appl.* **19**(8–9), 127–145 (1990)
33. Khattak, A.J., Tirmizi, S., et al.: Application of meshfree collocation method to a class of nonlinear partial differential equations. *Eng. Anal. Bound. Elements* **33**(5), 661–667 (2009)
34. Larsson, E., Lehto, E., Heryudono, A., Fornberg, B.: Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. *SIAM J. Sci. Comput.* **35**(4), A2096–A2119 (2013)
35. Merad, A., Martín-Vaquero, J.: A Galerkin method for two-dimensional hyperbolic integro-differential equation with purely integral conditions. *Appl. Math. Comput.* **291**, 386–394 (2016)
36. Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Ser. A, Containing Pap. Math. Phys. Charact.* **209**(441–458), 415–446 (1909)
37. Mirrahimi, S.: Integro-differential models from ecology and evolutionary biology. Ph.D. thesis, Université Paul Sabatier (Toulouse 3) (2019)
38. Mirzaee, F., Samadyar, N.: Using radial basis functions to solve two dimensional linear stochastic integral equations on non-rectangular domains. *Eng. Anal. Bound. Elements* **92**, 180–195 (2018)
39. Mohamed, M.S., Gepreel, K.A., Alharthi, M.R., Alotabi, R.A.: Homotopy analysis transform method for integro-differential equations. *General Math. Notes* **32**(1), 32 (2016)
40. Pazouki, M., Schaback, R.: Bases for kernel-based spaces. *J. Comput. Appl. Math.* **236**(4), 575–588 (2011)



41. Rashidinia, J., Fasshauer, G.E., Khasi, M.: A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems. *Comput. Math. Appl.* **72**(1), 178–193 (2016)
42. Ray, S.S., Behera, S.: Two-dimensional wavelets operational method for solving Volterra weakly singular partial integro-differential equations. *J. Comput. Appl. Math.* **366**, 112411 (2020)
43. Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.* **11**(2–3), 193–210 (1999)
44. Shen, J., Tang, T.: *High order numerical methods and algorithms*. Chinese Science Press, Abstract and Applied Analysis (2005)
45. Uddin, M., Ullah, N., Shah, S.I.A.: Rbf Based Localized Method for Solving Nonlinear Partial Integro-Differential Equations. *Comput. Model. Eng. Sci.* **123**(3), 955–970 (2020)
46. Wang, W., Chen, Y., Fang, H.: On the variable two-step IMEX BDF method for parabolic integro-differential equations with nonsmooth initial data arising in finance. *SIAM J. Numer. Anal.* **57**(3), 1289–1317 (2019)
47. Wendland, H.: *Scattered Data Approximation*, vol. 17. Cambridge University Press, Cambridge (2004)
48. Wendland, H.: *Scattered data approximation* (2005)
49. Wu, Z.M., Schaback, R.: Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal.* **13**(1), 13–27 (1993)
50. Yüzbaşı, Ş., Şahin, N., Sezer, M.: Bessel polynomial solutions of high-order linear Volterra integro-differential equations. *Comput. Math. Appl.* **62**(4), 1940–1956 (2011)
51. Zhao, J., Corless, R.M.: Compact finite difference method for integro-differential equations. *Appl. Math. Comput.* **177**(1), 271–288 (2006)
52. Zheng, X., Qiu, W., Chen, H.: Three semi-implicit compact finite difference schemes for the nonlinear partial integro-differential equation arising from viscoelasticity. *Int. J. Model. Simul.* **41**(3), 234–242 (2020)
53. Zong-Min, W.: Radial basis function scattered data interpolation and the meshless method of numerical solution of PDEs. *Chin. J. Eng. Math.* **2** (2002)

