Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 10, No. 1, 2022, pp. 144-157
DOI:10.22034/cmde.2021.44736.1890

Collocation method based on radial basis functions via symmetric variable shape parameter for solving a particular class of delay differential equations

Asadollah Torabi Giklou[1,2], **Mojtaba Ranjbar**[3,4,*], **Mahmoud Shafiee**[2], **and Vahid Roomi**[3]

[1]Department of Mathematics, Guilan Science and Research Branch, Islamic Azad University, Rasht, Iran.

[2]Department of Mathematics, Rasht Branch, Islamic Azad University, Rasht, Iran.

[3]Department of Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran.

[4]Faculty of Financial Sciences, Kharazmi University, Tehran, Iran.

Abstract

In this article, we use the collocation method based on the radial basis functions with symmetric variable shape parameter (SVSP) to obtain numerical solutions of neutral-type functional-differential equations with proportional delays. In this method, we control the absolute errors and the condition number of the system matrix through the program prepared with Maple 18.0 by increasing the number of collocation points that have a direct effect on the defined shape parameter. Also, we present the tables of the rate of the convergence (ROC) to investigate and show the convergence rate of this method compared to the RBF method with constant shape parameter. Several examples are given to illustrate the efficiency and accuracy of the introduced method in comparison with the same method with the constant shape parameter (CSP) as well as other analytical and numerical methods. Comparison of the obtained numerical results shows the considerable superiority of the collocation method based on RBFs with SVSP in accuracy and convergence over the collocation method based on the RBFs with CSP and other analytical and numerical methods for delay differential equations (DDEs).

Keywords. Functional-differential equation, Proportional delay, Radial basis function, Variable shape parameter.

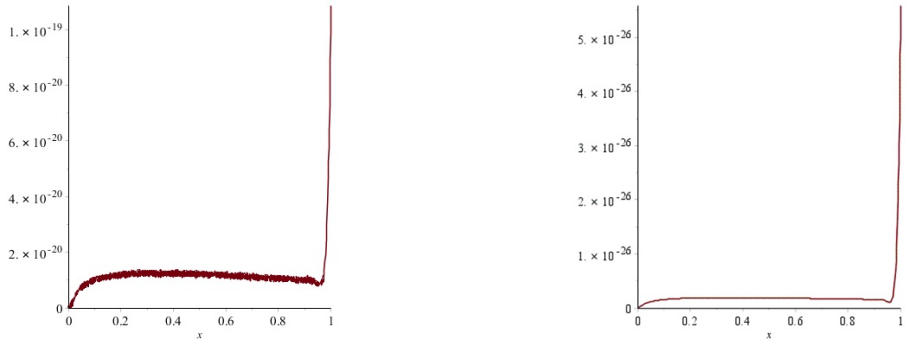
2010 Mathematics Subject Classification. 65L05, 34K06, 34K28.

1. Introduction

In modeling many natural phenomena, paying attention to hereditary effects and not ignoring them leads to the emergence of a differential equation in which the state variable or its derivatives appear with the delay arguments. These equations appear in the modeling of many science and engineering problems, especially in physics, chemistry, economics, population dynamics, species interaction, physiology, and many other fields. We focus on a particular class of DDEs, namely neutral functional-differential equations with proportional delays, which have the constraint that the derivatives of delay terms can appear in the equations. Such functional-differential equations play a major role in describing many different real life phenomena ([16]). In addition, different analytical and numerical methods for obtaining the solutions of this type of delay differential equations have been presented by different authors.

Wang et al. applied the continuous Runge-Kutta methods ([35]) and one-legθ-methods ([34]) to compute the approximate solutions of neutral-type delay differential equations. Ishiwata et al. in [16] used the rational approximation method to obtain the approximate solutions of delay differential equations with proportional delays. Also, applying the collocation method, these types of equations are solved approximately in [17]. Ghomanjani et al. in [10] applied the Bezier control points method to get very accurate approximate analytical solutions. In [37], the reproducing kernel Hilbert space method (RKHSM) was applied to neutral functional-differential equations with proportional delays. Chen and his collaborator in [4] applied the variational iteration method for solving a neutral functional-differential

Received: 26 February 2021 ; Accepted: 14 June 2021.
∗ Corresponding author. Email: ranjbar633@gmail.com.



(A) CSP (B) SVSP

FIGURE 1. Graphs of absolute error versus shape parameter with GA-RBF, ($N = 24, c^* = 0.5$), for Example 1.

equation with proportional delays. The authors in [31] derived an efficient numerical scheme for DDE and stochastic delay differential equation (SDDE) based on Legendre spectral-collocation method, which proved to be numerical methods that can significantly speed up the computation. Davaeifar and Rashidinia in [5] considered a system of multi pantograph type delay differential equations with variable coefficients subject to the initial conditions and proposed a collocation method to obtain an approximate solution. Based on one of the Appell polynomials, namely Genocchi polynomials, a new simple and effective algorithm is presented by Isah and Phang to solve delay differential equations with neutral terms, generalized Pantograph equations and delay differential system with constant and variable coefficients ([15]). Recently, the author in [8] examined asymptotic mean-square stability analysis of stochastic linear theta scheme for n-dimensional stochastic delay differential equations. Also, a direct method is proposed in [26] for solving Volterra-Fredholm integral equations with time delay by using orthogonal functions and their stochastic operational matrix of integration. The readers are also encouraged to see theoretical and numerical studies in [13, 24, 38–42].

We have allocated this work to introduce the collocation method based on radial basis functions with constant and symmetric variable shape parameter strategies for the neutral functional-differential equations with proportional delays used in [4],

$$(u(t) + a(t)u(p_m t))^{(m)} = \beta u(t) + \sum_{k=0}^{m-1} b_k(t)u^{(k)}(p_k t) + f(t), \tag{1.1}$$

with the initial conditions

$$u^{(l)}(0) = \gamma_l, \quad l = 0, 1, ..., m-1, \tag{1.2}$$

where $a(t)$ and $b_k(t)$ are known analytical functions and β, p_k and γ_k are given constants with $0 < p_k < 1$ for $k = 0, 1, ..., m$.

This paper is organized as follows: after introducing the global radial basis functions in the second section, functional-differential equations are implemented by RBFs in section 3. Next, in section 4, RBF method is applied to some initial value problems. Finally, the conclusion is given in section 5.

2. Collocation method based on the radial basis functions

In the last 25 years, the meshless method of radial basis functions has been considered by many researchers to approximate the solutions of various differential equations due to its significant advantages. This method was first proposed in 1971 by Hardy ([14]) for the interpolation of multivariate functions with scattered data. The properties and advantages of radial basis functions have motivated many researchers to apply this method to approximate the solutions of a variety of differential equations. Edvard Kanasa in [18, 19] introduced a method for the estimation of partial derivatives. Afterward, this method was applied as a meshless method for numerical solutions of many partial differential equations (PDEs) and ordinary differential equations (ODEs) based on the collocation scheme. Because

of the collocation technique, RBF method does not need to evaluate any integral. One of the most important features of the radial basis functions method is that they are not sensitive to the problem space dimension. Therefore, this method can be easily used for higher dimensional problems.

Khodayari and Ranjbar in [23] used the global radial basis functions-based differential quadrature (RBFs-DQ) method for solving multi-dimensional Black-Scholes equation. Rashidinia and Khasi applied a stable Gaussian radial basis function method for solving Helmholtz equations ([30]). Recently, the authors in [33] employed a combination of variational iteration method and Padé approximation method, called the VIM-Padé technique, to solve some nonlinear initial value problems and a delay differential equation.

In recent decades, many researchers have used the RBFs method to solve a variety of differential equations (see [1, 6, 7, 11, 22, 25, 27, 28] and the references cited therein).

A radial basis function on R^d is a function of the form

$$\phi_*(\mathbf{t}) = \phi(\|\mathbf{t} - \mathbf{t}^*\|_2),$$

where $\mathbf{t}, \mathbf{t}^* \in R^d$ and $\|.\|_2$ denotes the Euclidean distance between $\mathbf{t}$ and $\mathbf{t}^*$. The radial basis function ϕ is radially symmetric about the center $\mathbf{t}^*$. Many different radial basis functions have been used in the literature. Some of the well-known RBFs are listed in Table 1, where $r = \|\mathbf{t} - \mathbf{t}^*\|_2$ and c is a free positive parameter, often referred to as the shape parameter, to be specified by the user.

TABLE 1. Some well-known functions that generate RBFs

Name of radial basis function	Definition
Multiquadric (MQ)	$\phi(r) = \sqrt{c^2 + r^2}$
Inverse Multiquadric (IMQ)	$\phi(r) = \frac{1}{\sqrt{c^2+r^2}}$
Inverse Quadric (IQ)	$\phi(r) = \frac{1}{c^2+r^2}$
Gaussian (GA)	$\phi(r) = e^{-c^2 r^2}$
Thin Plate Spline (TPS)	$\phi(r) = r^2 \log(r)$

2.1. **RBFs interpolation.** RBFs interpolation method was first studied by Roland Hardy for the interpolation of scattered data ([14]). Polynomial methods had previously been used, but they do not have an insolvency property for two-dimensional and higher dimensional scattered data. Researchers' interest in using this interpolation method developed after the review of Franke ([9]). The convergence property of the RBFs method has been shown by Buhman in [2].

Now, we employ RBF interpolation method at N separately nodes $(\mathbf{t}_i, \mathbf{f}_i)$ with $i = 1, 2, ..., N, \mathbf{t}_i \in R^d$ and $\mathbf{f}_i \in R$. For this purpose, if one chooses N center points $\mathbf{t}_1^*, ..., \mathbf{t}_N^*$ in R^d, then the basic RBF interpolant for data values $f(\mathbf{t}_i)$ at scattered data $\mathbf{t}_i, i = 1, 2, ..., N$ in d dimensions takes the form

$$P(\mathbf{t}) = \sum_{i=1}^{N} \lambda_i \phi(\|\mathbf{t} - \mathbf{t}_i^*\|_2),$$

where $\phi(r)$ with $r = \|\mathbf{t} - \mathbf{t_i}^*\|_2$ can be chosen from Table 1. The coefficients λ_j are chosen by enforcing the interpolation conditions

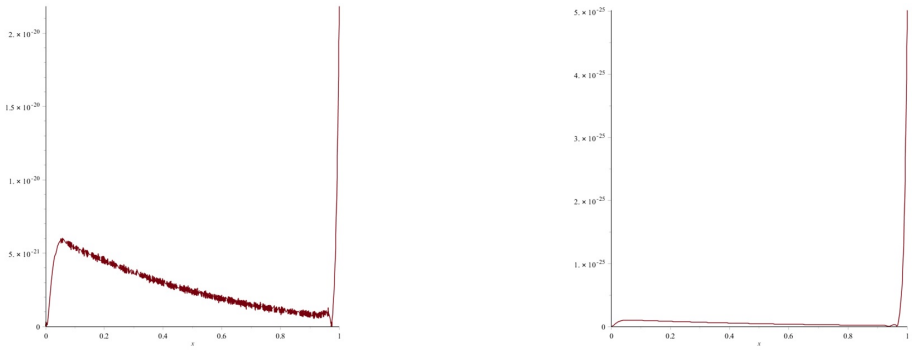
$$P(\mathbf{t}_j) = f(\mathbf{t}_j), \quad j = 1, 2, ...N,$$

at a set of points that usually coincides with the N centers. Enforcing the interpolation condition at the N centers results in the $N \times N$ linear system of equations which can be represented as the matrix equation

$$\mathbf{A}\lambda = \mathbf{f}. \tag{2.1}$$

Solving this system, the solution of the interpolation problem is obtained as

$$\lambda = \mathbf{A}^{-1}\mathbf{f}. \tag{2.2}$$



(A) CSP (B) SVSP

FIGURE 2. Graphs of absolute error versus shape parameter with GA-RBF, ($N = 24, c^* = 0.5$), for Example 2.

If the radial basis functions are of the Gaussian type (GA-RBF), it can be shown that the matrix $\mathbf{A}$ with entries $a_{ji} = \phi(\|\mathbf{t}_j^* - \mathbf{t}_i^*\|_2), \quad i, j = 1, ..., N$ is positive definite for distinct interpolation nodes ([32]). The condition number $\kappa_s(\mathbf{A})$ for a given square matrix $\mathbf{A}$ is defined as

$$\kappa_s(\mathbf{A}) = \|\mathbf{A}\|_s \|\mathbf{A}^{-1}\|_s, \qquad s = 1, 2, \infty. \tag{2.3}$$

The condition number depends on various factors. One of the most important factors is the suitable selection of the shape parameter value. For a fixed number of interpolation nodes, if the shape parameter is large, then RBF methods are very accurate. However, the system matrix will be very ill-conditioned in this case.

2.2. **Variable shape parameter strategy.** The choice of a suitable value for shape parameters of RBF methods is an important topic for applying RBF method for interpolation or approximation. A large value shape parameter c produces poor approximation and a well-conditioned system matrix $\mathbf{A}$. The system matrix will be ill-conditioned when we choose a small shape parameter and will result a very accurate RBF approximation. There are many mathematical techniques for dealing with this problem in the use of RBFs method. Many algorithms and techniques have been presented to improve bad condition numbers in older works such as the multilevel method, domain decomposition method, truncated RBF method, RBF with variable shape parameter and knot optimization method ([21]).

The concept of variable shape parameters was proposed by Kansa and Carlson in [20]. They considered a set of shape parameters which minimizes an error function over some evaluation nodes. In fact, an important advantage of using the variable shape parameter in radial basis functions is that it usually improves the interpolation matrix condition number.

Although working with constant shape parameter is simple, some works show that the use of the variable shape parameter in radial basis functions has a considerable advantage over the constant shape parameter. So far, mathematicians have proposed several strategies for determining the variable shape parameter. First, three formulas are suggested by Kansa in [18] to determine different shape parameter corresponding to each center points. Then, a random variable shape parameter strategy with

$$c_j = c_{\min} + (c_{\max} - c_{\min}) \times rand(1, N) \ , \ j = 1, 2, ..., N, \tag{2.4}$$

is introduced by Sarra and Sturgill, in which c_j is the jth shape parameter corresponding to the jth center point and the minimum and maximum of c_j are shown by $c_{\min}$ and $c_{\max}$, respectively. Also, the function that produces N uniformly distributed pseudo-random numbers on $[0, 1]$ is shown by $rand(1, N)$ ([32]). A trigonometric VSP for generalized multiquadric radial basis functions (MQ-RBF) with the formula

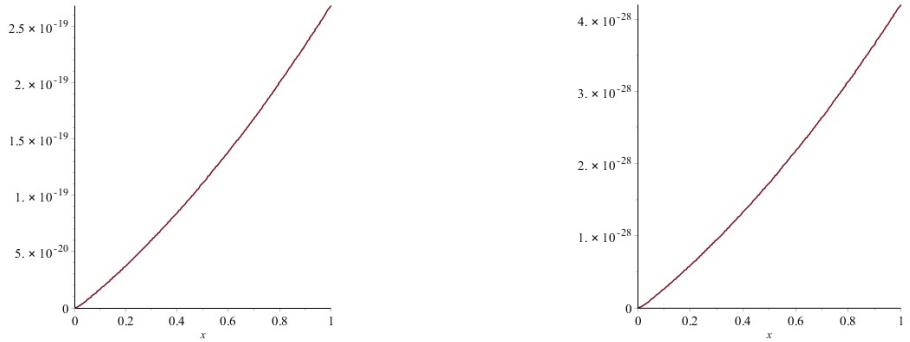
$$c_j = c_{\min} + (c_{\max} - c_{\min}) \sin(j), \qquad j = 1, 2, \cdots, N. \tag{2.5}$$

is used by Xiang et al. in [36]. Since sin function produces non-positive shape parameters, Golbabaie and Rabiei changed the argument of the function and, with combination of Kansa's formula, suggested a hybrid shape parameter

strategy ([12]). A new formula for computing variable shape parameter, called symmetric variable shape parameter (SVSP), is introduced by Ranjbar in [29] as

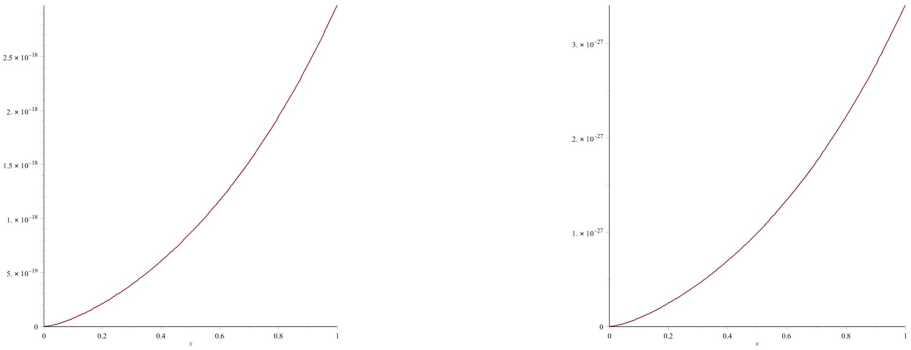
$$c_j = c^* \exp(-\frac{1}{2}(\frac{j-\mu}{\sigma})^2), \tag{2.6}$$

where $c^* \in [c_{\min}, c_{\max}]$ is an arbitrary shape parameter, $\mu = 0.5N$, $\sigma = 0.25N$ and N is the total number of centers. Note that μ and σ control the shape parameter values c_j around c^*. The main purpose of this paper is solving a particular class of delay differential equations using GA-RBFs with SVSP.



(A) CSP (B) SVSP

FIGURE 3. Graphs of absolute error versus shape parameter with GA-RBF, $(N = 24, c^* = 0.5)$, for Example 3.



(A) CSP (B) SVSP

FIGURE 4. Graphs of absolute error versus shape parameter with GA-RBF, $(N = 24, c^* = 0.5)$, for Example 4.

3. Solving neutral functional-differential equations with RBFs

In this section, we intend to solve neutral functional-differential equations with proportional delays using the collocation method based on the radial basis functions with constant and variable shape parameters. In the RBF approximation method, the unknown $u(t)$ is approximated by

$$u(t) \simeq u_N(t) = \sum_{j=1}^{N} \lambda_j \phi_j(t), \tag{3.1}$$

where RBF used in this paper is Gaussian (GA) as

$$\phi_j(t) = \phi(\|t - t_j\|, c_j) = e^{-c_j^2 \|t-t_j\|^2}, \tag{3.2}$$

where t_j is coordinate of node j and shape parameter c_j is SVSP. Using the linear combination Eq(3.1), the sth derivatives in Eq(1.1) are approximated as

$$u^{(s)}(t) \simeq u_N^{(s)}(t) = \sum_{j=1}^{N} \lambda_j \phi_j^{(s)}(t), \tag{3.3}$$

where $s = 1, 2, ...m$. Equations Eq(3.1) and Eq(3.3) can be expressed in the matrix form by

$$u_N(t) = \Phi(t)\Lambda,$$

and

$$u_N^{(s)}(t) = \Phi^{(s)}(t)\Lambda, \tag{3.4}$$

where the RBFs vector $\Phi(t)$ and the coefcients vector Λ are given respectively by

$$\Phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_N(t)],$$

and

$$\Lambda = [\lambda_1, \lambda_2, ..., \lambda_N]^T.$$

By substituting Eq(3.1) and Eq(3.3) into Eq(1.1), we get

$$\sum_{j=1}^{N} \lambda_j \phi_j^{(m)}(t) + \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} a^{(m-k)}(t)(\sum_{j=1}^{N} \lambda_j \phi_j^{(k)}(p_m t)) = \beta \sum_{j=1}^{N} \lambda_j \phi_j(t) + \sum_{k=0}^{m-1} b_k(t)(\sum_{j=1}^{N} \lambda_j \phi_j^{(k)}(p_k t)) + f(t). \tag{3.5}$$

Its the matrix form is

$$\Phi^{(m)}(t)\Lambda + \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} a^{(m-k)}(t)\Phi^{(k)}(p_m t)\Lambda = \beta\Phi(t)\Lambda + \sum_{k=0}^{m-1} b_k(t)\Phi^{(k)}(p_k t)\Lambda + f(t). \tag{3.6}$$

To find the unknown coefficients, the collocation points $t_i = i\frac{T}{N}$, $i = 1, 2, ..., N - m$ are put into Eq(3.6) and the equations are obtained as

$$\Phi^{(m)}(t_i)\Lambda + \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} a^{(m-k)}(t_i)\Phi^{(k)}(p_m t_i)\Lambda = \beta\Phi(t_i)\Lambda + \sum_{k=0}^{m-1} b_k(t_i)\Phi^{(k)}(p_k t_i)\Lambda + f(t_i),$$

where $i = 1, 2, ..., N - m$. This system can be rewritten as

$$[\Phi^{(m)}(t_i) + \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} a^{(m-k)}(t_i)\Phi^{(k)}(p_m t_i) - \beta\Phi(t_i) - \sum_{k=0}^{m-1} b_k(t_i)\Phi^{(k)}(p_k t_i)]\Lambda = f(t_i).$$

Now, the fundamental matrix equation corresponding to Eq(1.1) can be written as

$$A\Lambda = F. \tag{3.7}$$

Here, $F = [f(t_1), f(t_2), ..., f(t_{N-m})]^T$ and $A = [A_1, A_2, ..., A_{N-m}]^T$ where

$$A_i = [\Phi^{(m)}(t_i) + \sum_{k=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} a^{(m-k)}(t_i)\Phi^{(k)}(p_m t_i) - \beta\Phi(t_i) - \sum_{k=0}^{m-1} b_k(t_i)\Phi^{(k)}(p_k t_i)], \ i = 1, 2, ..., N - m.$$

Using Eq(3.1) and Eq(3.3) at $t = 0$, the initial conditions given in Eq(1.2) can be rewritten in the matrix form

$$\sum_{k=1}^{N} \lambda_k \phi_k^{(l)}(0) = \Phi^{(l)}(0)\Lambda = \gamma_l, \quad l = 0, 1, ..., m - 1. \tag{3.8}$$

Thus, the matrix form of Eq(1.2) is

$$B\Lambda = \Gamma, \tag{3.9}$$

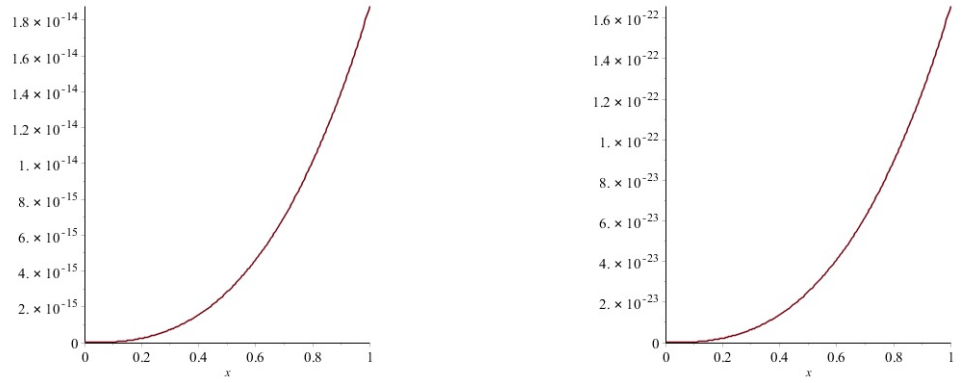
Here, $\Gamma = [\gamma_1, \gamma_2, ..., \gamma_{m-1}]^T$ and $B = [B_0, B_1, ..., B_{m-1}]^T$ where

$$B_l = [\phi_1^{(l)}(0), \phi_2^{(l)}(0), ..., \phi_N^{(l)}(0)],\ l = 0, 1, ..., m-1.$$

Finally, by adding m algebraic equations Eq(3.9) to equations Eq(3.7), we reduce Eq(1.1) under conditions Eq(1.2) to the following linear system of algebraic equations.

$$\begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} \Lambda \end{bmatrix} = \begin{bmatrix} F \\ \Gamma \end{bmatrix}. \tag{3.10}$$

Matrix equation Eq(3.10) is a system of $(N \times N)$ linear algebraic equations. The unknown coefficients vector Λ can be uniquely determined after solving this system. Then, by putting it into Eq(3.1), approximate solution $u_N(t)$ at any point t in terms of RBFs can be obtained.

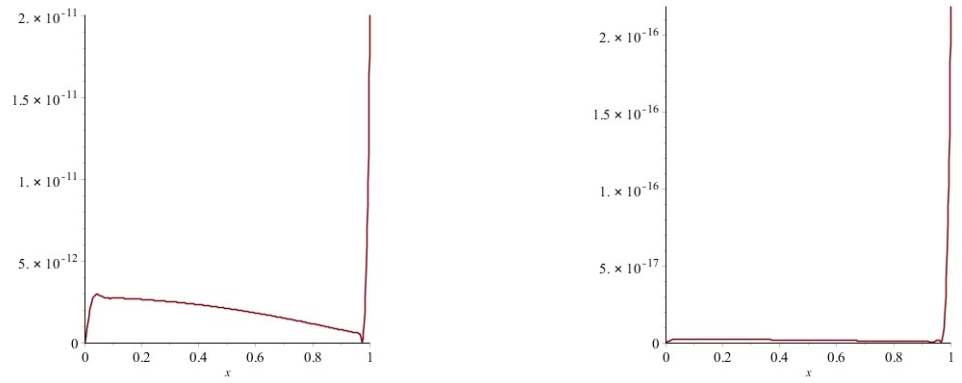


(A) CSP (B) SVSP

FIGURE 5. Graphs of absolute error versus shape parameter with GA-RBF, $(N = 24, c^* = 0.5)$, for Example 5.

TABLE 2. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.9, 1.0$ and comparing with other methods for Example 1.

t	*Method*[34]	*Method*[35]	*Method*[4]	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
			$n = 8$	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	2.57×10^{-3}	4.55×10^{-4}	3.72×10^{-4}	2.1649×10^{-11}	2.7644×10^{-11}	1.0010×10^{-20}	1.5429×10^{-27}
0.2	8.86×10^{-3}	4.24×10^{-4}	7.08×10^{-4}	2.7862×10^{-11}	3.5567×10^{-11}	1.1804×10^{-20}	1.8279×10^{-27}
0.3	1.72×10^{-2}	1.12×10^{-3}	1.01×10^{-3}	3.0163×10^{-11}	3.8505×10^{-11}	1.2545×10^{-20}	1.9050×10^{-27}
0.4	2.66×10^{-2}	1.35×10^{-3}	1.29×10^{-3}	3.0833×10^{-11}	3.9356×10^{-11}	1.2550×10^{-20}	1.9055×10^{-27}
0.5	3.63×10^{-2}	1.52×10^{-3}	1.54×10^{-3}	3.0817×10^{-11}	3.9345×10^{-11}	1.2224×10^{-20}	1.8629×10^{-27}
0.6	4.58×10^{-2}	1.66×10^{-3}	1.76×10^{-3}	2.9997×10^{-11}	3.8299×10^{-11}	1.3033×10^{-20}	1.7984×10^{-27}
0.7	5.47×10^{-2}	1.75×10^{-3}	1.97×10^{-3}	2.8678×10^{-11}	3.6565×10^{-11}	1.1488×10^{-20}	1.7245×10^{-27}
0.8	6.29×10^{-2}	1.81×10^{-3}	2.15×10^{-3}	2.8744×10^{-11}	3.6887×10^{-11}	1.0673×10^{-20}	1.6452×10^{-27}
0.9	7.02×10^{-2}	1.84×10^{-3}	2.32×10^{-3}	1.9666×10^{-11}	2.3121×10^{-11}	1.1245×10^{-20}	1.5788×10^{-27}
1.0	7.66×10^{-2}	1.85×10^{-3}	2.47×10^{-3}	4.0925×10^{-10}	6.4970×10^{-10}	1.0874×10^{-19}	5.5954×10^{-26}
$RMSE$	-	-	1.70×10^{-3}	1.2084×10^{-10}	1.9031×10^{-10}	2.4625×10^{-20}	1.1538×10^{-26}
E_∞	-	-	2.47×10^{-3}	4.0925×10^{-10}	6.4970×10^{-10}	1.0874×10^{-19}	5.5954×10^{-26}
E_2	-	-	1.60×10^{-3}	5.6736×10^{-11}	8.5906×10^{-11}	1.6517×10^{-20}	6.4819×10^{-27}
$\kappa_s(A)$	-	-	-	4.6553×10^{25}	8.2925×10^{17}	1.3096×10^{52}	2.9128×10^{40}



(A) CSP (B) SVSP

FIGURE 6. Graphs of absolute error versus shape parameter with GA-RBF, $(N = 24, c^* = 2.5)$, for Example 6.

TABLE 3. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.9, 1.0$ and comparing with other methods for Example 2.

t	$Method$[34]	$Method$[35]	$Method$[4]	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
			$n = 6$	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	4.65×10^{-3}	8.68×10^{-4}	2.15×10^{-4}	7.4275×10^{-11}	9.5638×10^{-11}	5.2869×10^{-21}	9.6698×10^{-27}
0.2	1.45×10^{-2}	1.49×10^{-3}	3.39×10^{-3}	6.1921×10^{-11}	7.9620×10^{-11}	4.6674×10^{-21}	8.1024×10^{-27}
0.3	2.57×10^{-2}	1.90×10^{-3}	8.75×10^{-3}	5.1750×10^{-11}	6.6562×10^{-11}	3.8398×10^{-21}	6.6896×10^{-27}
0.4	3.60×10^{-2}	2.16×10^{-3}	1.56×10^{-2}	4.2224×10^{-11}	5.4293×10^{-11}	2.7821×10^{-21}	5.4767×10^{-27}
0.5	4.43×10^{-2}	2.28×10^{-3}	2.35×10^{-2}	3.4279×10^{-11}	4.4072×10^{-11}	2.4884×10^{-21}	4.4448×10^{-27}
0.6	5.03×10^{-2}	2.31×10^{-3}	3.20×10^{-2}	2.7772×10^{-11}	3.5741×10^{-11}	1.9635×10^{-21}	3.5713×10^{-27}
0.7	5.37×10^{-2}	2.27×10^{-3}	4.06×10^{-2}	2.1247×10^{-11}	2.7187×10^{-11}	1.3086×10^{-21}	2.8354×10^{-27}
0.8	5.47×10^{-2}	2.17×10^{-3}	4.93×10^{-2}	2.0223×10^{-11}	2.6742×10^{-11}	1.0025×10^{-21}	2.2188×10^{-27}
0.9	5.35×10^{-2}	2.03×10^{-3}	5.76×10^{-2}	1.1880×10^{-11}	2.2585×10^{-11}	8.4996×10^{-22}	1.8549×10^{-27}
1.0	5.03×10^{-2}	1.86×10^{-3}	6.55×10^{-2}	1.3115×10^{-9}	2.1447×10^{-9}	2.1851×10^{-20}	5.0121×10^{-25}
$RMSE$	-	-	3.69×10^{-2}	3.8065×10^{-10}	6.2121×10^{-10}	5.3726×10^{-21}	1.0245×10^{-25}
E_∞	-	-	6.55×10^{-2}	1.3115×10^{-9}	2.1447×10^{-9}	2.1851×10^{-20}	5.0121×10^{-25}
E_2	-	-	3.37×10^{-2}	1.6131×10^{-10}	2.6014×10^{-10}	3.9466×10^{-21}	5.6317×10^{-26}
$\kappa_s(A)$	-	-	-	4.6553×10^{25}	8.2925×10^{17}	1.3096×10^{52}	2.9128×10^{40}

TABLE 4. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.9, 1.0$ and comparing with other methods for Example 3.

t	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	1.0791×10^{-8}	1.6288×10^{-12}	1.6605×10^{-20}	2.6166×10^{-29}
0.2	3.3662×10^{-8}	4.1403×10^{-12}	3.7382×10^{-20}	5.8659×10^{-29}
0.3	6.4303×10^{-8}	6.8491×10^{-12}	5.9967×10^{-20}	9.3993×10^{-29}
0.4	1.0222×10^{-7}	9.7779×10^{-12}	8.4382×10^{-20}	1.3215×10^{-28}
0.5	1.4775×10^{-7}	1.2930×10^{-11}	1.1058×10^{-19}	1.7313×10^{-28}
0.6	2.0158×10^{-7}	1.6302×10^{-11}	1.3859×10^{-19}	2.1693×10^{-28}
0.7	2.6462×10^{-7}	1.9889×10^{-11}	1.6841×10^{-19}	2.6356×10^{-28}
0.8	3.3795×10^{-7}	2.3689×10^{-11}	2.0003×10^{-19}	3.1303×10^{-28}
0.9	4.2280×10^{-7}	2.7704×10^{-11}	2.3349×10^{-19}	3.6532×10^{-28}
1.0	5.2058×10^{-7}	3.1933×10^{-11}	2.6871×10^{-19}	4.2043×10^{-28}
$RMSE$	2.5302×10^{-7}	1.7363×10^{-11}	1.4427×10^{-19}	2.4429×10^{-28}
E_∞	5.2058×10^{-7}	3.1933×10^{-11}	2.6871×10^{-19}	4.2043×10^{-28}
E_2	2.3827×10^{-7}	1.6752×10^{-11}	1.4184×10^{-19}	2.2198×10^{-28}
$\kappa_s(A)$	4.6553×10^{25}	8.2925×10^{17}	1.3096×10^{52}	2.9128×10^{40}

4. ILLUSTRATIVE EXAMPLES

In this section, we present some examples to show the efficiency and accuracy of the proposed method. In order to compare the results, we provided examples 1-5 examined by Chen and Wang in [4] and example 6 of a nonlinear

TABLE 5. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.5$ and comparing with other methods for Example 4.

t	$Method$[34]	$Method$[35]	$Method$[4]	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
			$n = 6$	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	6.10×10^{-3}	1.00×10^{-3}	1.67×10^{-4}	1.0791×10^{-8}	4.2973×10^{-12}	7.5070×10^{-20}	8.7358×10^{-29}
0.2	2.58×10^{-2}	2.02×10^{-3}	7.15×10^{-4}	3.3662×10^{-8}	1.3384×10^{-11}	2.1190×10^{-19}	2.4442×10^{-28}
0.3	6.47×10^{-2}	3.07×10^{-3}	1.73×10^{-3}	6.4303×10^{-8}	2.5553×10^{-11}	3.8891×10^{-19}	4.4712×10^{-28}
0.4	1.37×10^{-1}	4.17×10^{-3}	3.30×10^{-3}	1.0222×10^{-11}	4.0609×10^{-11}	6.0540×10^{-19}	6.9479×10^{-28}
0.5	2.81×10^{-1}	5.34×10^{-3}	5.55×10^{-3}	1.4775×10^{-7}	5.8687×10^{-11}	8.6410×10^{-19}	9.9067×10^{-28}
$RMSE$	-	-	-	2.5302×10^{-7}	1.0047×10^{-10}	1.4112×10^{-18}	1.6156×10^{-27}
E_∞	-	-	-	5.2058×10^{-7}	2.0670×10^{-10}	2.9778×10^{-18}	3.4076×10^{-27}
E_2	-	-	-	2.3827×10^{-7}	9.4618×10^{-11}	1.3710×10^{-18}	1.5697×10^{-27}
$\kappa_s(A)$	-	-	-	4.6553×10^{25}	8.2925×10^{17}	1.3096×10^{52}	2.9128×10^{40}

TABLE 6. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.9, 1.0$ and comparing with other methods for Example 5.

t	$Method$[35]	$Method$[4]	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
		$n = 6$	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	4.97×10^{-5}	9.09×10^{-12}	5.4908×10^{-7}	3.4822×10^{-9}	3.8050×10^{-17}	3.3972×10^{-25}
0.2	4.43×10^{-4}	2.98×10^{-10}	3.8228×10^{-8}	2.4193×10^{-8}	2.4541×10^{-18}	2.1803×10^{-24}
0.3	1.57×10^{-3}	2.33×10^{-9}	1.1650×10^{-5}	7.3664×10^{-8}	7.2363×10^{-18}	6.4149×10^{-24}
0.4	3.85×10^{-3}	1.01×10^{-8}	2.5573×10^{-5}	1.6160×10^{-7}	1.5586×10^{-15}	1.3800×10^{-23}
0.5	7.78×10^{-3}	3.20×10^{-8}	4.7056×10^{-5}	2.9726×10^{-7}	2.8324×10^{-15}	2.5060×10^{-23}
0.6	1.39×10^{-2}	8.24×10^{-8}	7.7554×10^{-5}	4.8981×10^{-7}	4.6277×10^{-15}	4.0924×10^{-23}
0.7	2.28×10^{-2}	1.85×10^{-7}	1.1855×10^{-4}	7.4862×10^{-7}	7.0297×10^{-15}	6.2141×10^{-23}
0.8	3.53×10^{-2}	3.76×10^{-7}	1.7161×10^{-4}	1.0835×10^{-6}	1.0127×10^{-14}	8.9499×10^{-23}
0.9	5.19×10^{-2}	7.09×10^{-7}	2.3837×10^{-4}	1.5048×10^{-6}	1.4016×10^{-14}	1.2383×10^{-22}
1.0	7.34×10^{-2}	1.26×10^{-6}	3.2062×10^{-4}	2.0238×10^{-6}	1.8797×10^{-14}	1.6604×10^{-22}
$RMSE$	-	-	1.3763×10^{-4}	8.6888×10^{-7}	7.7148×10^{-15}	6.8166×10^{-23}
E_∞	-	-	3.2062×10^{-4}	2.0238×10^{-6}	1.8797×10^{-14}	1.6604×10^{-22}
E_2	-	-	1.2487×10^{-4}	7.8836×10^{-7}	7.3562×10^{-15}	6.5000×10^{-23}
$\kappa_s(A)$	-	-	4.6553×10^{25}	8.2925×10^{17}	1.3096×10^{52}	2.9128×10^{40}

TABLE 7. Absolute errors between the exact and approximate solution $u(t)$ at $t = 0.1, 0.2, ..., 0.9, 1.0$ and comparing with other methods for Example 6.

t	$Method$[4]	GARBF-CSP	GARBF-SVSP	GARBF-CSP	GARBF-SVSP
	$n = 6$	$N = 12$	$N = 12$	$N = 24$	$N = 24$
0.1	3.4443×10^{-28}	1.2156×10^{-5}	1.0923×10^{-7}	2.7099×10^{-12}	2.1371×10^{-18}
0.2	2.8177×10^{-24}	9.6000×10^{-6}	8.8217×10^{-8}	2.6431×10^{-12}	2.0827×10^{-18}
0.3	5.4713×10^{-22}	9.6622×10^{-6}	8.8017×10^{-8}	2.5082×10^{-12}	1.9763×10^{-18}
0.4	2.2956×10^{-20}	8.7807×10^{-6}	8.0177×10^{-8}	2.3224×10^{-12}	1.8300×10^{-18}
0.5	4.1586×10^{-19}	7.9617×10^{-6}	7.2539×10^{-8}	2.0893×10^{-12}	1.6463×10^{-18}
0.6	4.4270×10^{-18}	6.9128×10^{-6}	6.3568×10^{-8}	1.8128×10^{-12}	1.4284×10^{-18}
0.7	3.2646×10^{-17}	5.6458×10^{-6}	4.9207×10^{-8}	1.4980×10^{-12}	1.1803×10^{-18}
0.8	1.8396×10^{-16}	4.3203×10^{-6}	5.2234×10^{-8}	1.1506×10^{-12}	9.0681×10^{-19}
0.9	8.4395×10^{-16}	5.1154×10^{-6}	6.2543×10^{-8}	7.8211×10^{-13}	6.9489×10^{-19}
1.0	3.2912×10^{-15}	1.3753×10^{-4}	3.8390×10^{-6}	2.0053×10^{-11}	2.1925×10^{-16}
$RMSE$	1.0760×10^{-15}	4.0390×10^{-5}	1.1104×10^{-6}	4.5438×10^{-12}	4.4783×10^{-17}
E_∞	3.2912×10^{-15}	1.3753×10^{-4}	3.8390×10^{-6}	2.0053×10^{-11}	2.1925×10^{-16}
E_2	6.3542×10^{-16}	1.7899×10^{-5}	4.6371×10^{-7}	3.0218×10^{-12}	2.4565×10^{-17}
$\kappa_s(A)$	-	1.6382×10^{10}	7.0578×10^{9}	1.9560×10^{29}	1.4888×10^{24}

TABLE 8. The error norms and the rate of convergence for various numbers of collocation points for Example 1

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.4567e-2	–	0.1506e-1	–	0.5662e-1	–	0.2078e0	–
8	0.7622e-6	12.5487	0.4318e-5	11.7680	0.5154e-5	13.4233	0.3147e-4	12.6889
12	0.5673e-10	23.4438	0.4092e-9	22.8480	0.8590e-10	27.1345	0.6497e-9	26.6065
16	0.2288e-14	35.1720	0.1830e-13	34.8129	0.6282e-15	41.1073	0.5173e-14	40.8117
20	0.5602e-19	47.5813	0.4679e-18	47.3872	0.2569e-20	55.6014	0.2186e-19	55.4544
24	0.1651e-19	6.7010	0.1087e-18	8.0059	0.6481e-26	70.7001	0.5595e-25	70.6208
28	0.1359e-20	16.1998	0.2535e-20	24.3813	0.1085e-31	86.2806	0.9449e-31	86.2239

neutral functional-differential equation examined in [3]. We solve these examples with the introduced method and

TABLE 9. The error norms and the rate of convergence for various numbers of collocation points for Example 2

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.4483e-2	–	0.1678e-1	–	0.6454e-1	–	0.2714e0	–
8	0.1432e-5	11.6122	0.9411e-5	10.8001	0.7380e-5	13.0942	0.5019e-4	12.4007
12	0.1613e-9	22.4219	0.1311e-8	21.8979	0.2601e-9	25.2875	0.2144e-8	24.8132
16	0.8527e-14	34.2314	0.7406e-13	34.0008	0.3206e-14	39.2926	0.2802e-13	39.0892
20	0.2522e-18	46.7346	0.2229e-17	46.6564	0.1810e-19	54.1562	0.1605e-18	54.0913
24	0.3942e-20	22.8087	0.2185e-19	25.3678	0.5631e-25	69.5504	0.5012e-24	69.5299
28	0.2634e-20	2.6155	0.1633e-19	1.8890	0.1094e-30	85.3150	0.9759e-30	85.3006

TABLE 10. The error norms and the rate of convergence for various numbers of collocation points for Example 3

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.2183	–	0.4309	–	0.2661e-1	–	0.5245e-1	–
8	0.2943e-3	9.5348	0.5665E-3	9.5710	0.2364e-5	13.4584	0.4549e-5	13.4931
12	0.4478e-7	21.6802	0.8538e-7	21.7037	0.1675e-10	29.2441	0.3193e-10	29.2673
16	0.2383e-11	34.2084	0.4528e-11	34.2203	0.1030e-16	49.7137	0.1957e-16	49.7252
20	0.6325e-16	47.2197	0.1199e-15	47.2302	0.8977e-22	52.2103	0.1702e-21	52.2199
24	0.1418e-18	33.4597	0.2687e-18	33.4618	0.2219e-27	70.8118	0.4204e-27	70.8159
28	0.5735e-19	5.8724	0.1085e-18	5.8828	0.2500e-33	88.8499	0.4730e-33	88.8586

TABLE 11. The error norms and the rate of convergence for various numbers of collocation points for Example 4

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.2237	–	0.5028	–	0.4612	–	1.0347	–
8	0.1287e-2	7.4414	0.2830e-2	7.4730	0.1124e-4	15.3244	0.2471e-4	12.0308
12	0.2382e-6	21.1971	0.5205e-6	21.2126	0.9461e-10	28.8193	0.2067e-9	28.8346
16	0.1429e-10	33.7918	0.3112e-10	33.8036	0.7018e-16	49.0618	0.1528e-15	49.0737
20	0.4135e-15	46.8326	0.8989e-15	46.8406	0.5890e-21	52.3795	0.1280e-20	52.3879
24	0.1371e-17	31.3134	0.2977e-17	31.3197	0.1569e-26	70.4016	0.3407e-26	70.4060
28	0.2061e-18	12.2927	0.4476e-18	12.2916	0.1866e-32	88.4988	0.4048e-32	88.5050

TABLE 12. The error norms and the rate of convergence for various numbers of collocation points for Example 5

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.1097e+1	–	0.2980e+1	–	0.4991	–	0.1305e+1	–
8	0.1686	2.7018	0.4357	2.7739	0.9712e-2	5.6834	0.2507e-1	5.7019
12	0.1248e-3	17.7785	0.3206e-3	17.7931	0.7883e-6	23.2301	0.2023e-5	23.2445
16	0.2100e-7	30.2067	0.5378e-7	30.2175	0.9538e-11	39.3571	0.2442e-10	39.3652
20	0.1335e-11	43.3055	0.3415e-11	43.3105	0.3754e-16	55.7730	0.9600e-16	55.7782
24	0.7356e-14	28.5274	0.1879e-13	28.5353	0.6500e-22	72.7644	0.1660e-21	72.7718
28	0.7479e-14	-0.1075	0.1910e-13	-0.1061	0.5477e-28	90.7343	0.1398e-27	90.7377

compare the obtained numerical results with the results of each of the variational iteration methods ([4]), the two-stage one-order Runge-Kutta method of Wang et al. ([35]), the one-leg θ-method of Wang and Li ([34]) with $\theta = 0.8$ and $h = 0.01$ and RBF method with constant shape parameter in Tables 2-7. Also, the E_∞, E_2 and RMS error norms of $u(t)$ and condition number of system matrix for these examples are listed in these tables to compare CSP and SVSP strategies. Throughout this section, we will study the performance of the collocation method based on the radial basis functions with symmetric variable shape parameter and constant shape parameter. All calculations and numerical and symbolic results provided for the examples, performed by using Maple 18.0 software. The accuracy of the method is measured through the following error norms. The formula for the root-mean-square error (RMSE) is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=0}^{N}(u_e(t_j) - u_a(t_j))^2}. \tag{4.1}$$

The error norms of the solution, E_∞ and E_2, are defined by

$$E_\infty = \|u_e - u_a\|_\infty = \max_{j=0,\dots N} |u_e(t_j) - u_a(t_j)|, \tag{4.2}$$

TABLE 13. The error norms and the rate of convergence for various numbers of collocation points for Example 6

	Rate of convergence with CSP				Rate of convergence with SVSP			
N	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$	E_2	$ROC(E_2)$	E_∞	$ROC(E_\infty)$
4	0.4567e-2	–	0.1506e-1	–	0.5662e-1	–	0.2078	–
8	0.7622e-6	12.5487	0.4318e-5	11.7680	0.5154e-5	13.4233	0.3147e-4	12.6889
12	0.5673e-10	23.4438	0.4092e-9	22.8480	0.8590e-10	27.1345	0.6497e-9	26.6065
16	0.2288e-14	35.1720	0.1830e-13	34.8129	0.6282e-15	41.1073	0.5173e-14	40.8117
20	0.5602e-19	47.5813	0.4679e-18	47.3872	0.2569e-20	55.6014	0.2186e-19	55.4544
24	0.1651e-19	6.7010	0.1087e-18	8.0059	0.6481e-26	70.7001	0.5595e-25	70.6208
28	0.1359e-20	16.1998	0.2535e-20	24.3813	0.1085e-31	86.2806	0.9449e-31	86.2239

and

$$E_2 = \|u_e - u_a\|_2 = \sqrt{\int_0^T (u_e(t) - u_a(t))^2 dt}, \tag{4.3}$$

where u_a and u_e are the approximate and the exact solutions of the problems, respectively. Using the following formula, we calculate the numerical rate of convergence (ROC) of the method.

$$ROC = \log(\frac{E_{s+1}}{E_s}) / \log(\frac{N_s}{N_{s+1}}), \tag{4.4}$$

where E_s is E_∞, E_2 or RMS error norms corresponding to the number of collocation points N_s. Therefore, some further numerical runs for different number of time steps have been performed. All graphs show the absolute error versus shape parameter with GA-RBF and $N = 24$.

Remark 4.1. In the tables of the following examples, n shows the repetition step of the variational iteration method (Method [4]) and N shows the number of central points in this manuscript.

Example 1. ([4]) Consider the first-order neutral functional-differential equation with proportional delay

$$u'(t) = -u(t) + \frac{1}{2}u(\frac{1}{2}t) + \frac{1}{2}u'(\frac{1}{2}t), \quad t \in [0, 1],$$

subject to initial condition $u(0) = 1$. The exact solution is $u(t) = e^{-t}$. We solve this example with the introduced method and compare the obtained numerical results with the results of each of the variational iteration method (for n=8) ([4]), the two-stage one-order Runge-Kutta method of Wang et al. ([35]), the one-leg θ-method of Wang and Li ([34]) with $\theta = 0.8$ and $h = 0.01$ in Table 2. Also, the E_∞, E_2 and RMS error norms of $u(t)$ and condition number of system matrix for this example are listed in this table to compare CSP and SVSP strategies. In Figure. 1 Absolute error graph of the solution of the RBF method with CSP and SVSP ($N = 24$) are shown. The reported numerical results in Tables 2, 8 show that, firstly, the RBF method with SVSP has a significant advantage over other comparison methods in terms of reducing solution error, especially with increasing N. Secondly, the numerical convergence rate analysis of the approximations obtained according to the Table 8 shows that the RBF method with SVSP has a regular and upward convergence rate with increasing N, while the RBF method with CSP has an irregular and oscillating convergence rate. Also, for each fixed N, the RBF-SVSP method has more accuracy and lower system matrix condition number than the RBF-CSP method, which reduces the computational error and the accuracy of the solution in the proposed method.

Example 2.([4]) Consider the following first-order neutral functional-differential equation with proportional delay

$$u'(t) = -u(t) + 0.1u(0.8t) + 0.5u'(0.8t) + (0.32t - 0.5)\exp(-0.8t) + \exp(-t),$$
$$t \geq 0,$$

subject to initial condition $u(0) = 0$ and with the exact solution $u(t) = te^{-t}$. Comparison of numerical results according to Tables 3 and 9 show that GA-RBF method with SVSP is more accurate than the GA-RBF method with CSP.

Example 3. ([4]) As another example, consider the following second-order neutral functional-differential equation with proportional delay

$$u''(t) = u'(\frac{1}{2}t) - \frac{1}{2}tu''(\frac{1}{2}t) + 2, \quad t \geq 0,$$
$$u(0) = 1, \quad u'(0) = 0.$$

The exact solution is $u(t) = 1 + t^2$. According to Tables 3 and 10, it is clear that for different values of N, the RBF-SVSP method is in a better position than the RBF-CSP method in terms of accuracy, condition number of system matrix and convergence rate growth.

Example 4. ([4]) Consider the second-order neutral functional-differential equation with proportional delay

$$u''(t) = \frac{3}{4}u(t) + u(\frac{1}{2}t) + u'(\frac{1}{2}t) + \frac{1}{2}u''(\frac{1}{2}t) - t^2 - t + 1, \quad t \geq 0,$$
$$u(0) = u'(0) = 0,$$

which has the exact solution $u(t) = t^2$. For this example, comparing the graph of the absolute errors of the proposed method with other methods in Fig. 4 and the values of the error norms and the system matrix condition number in Table. 5 and also, the numerical convergence rates analysis of the RBF-SVSP and RBF-CSP methods in Table. 11 all show the superiority of the proposed method over the compared methods and especially, over the RBF-CSP method in terms of the solution accuracy and reduction of the system ill-condition for the same N.

Example 5. ([4]) Consider the third-order neutral functional-differential equation with proportional delay

$$u'''(t) = u(t) + u'(\frac{1}{2}t) + u''(\frac{1}{3}t) + \frac{1}{2}u'''(\frac{1}{4}t) - t^4 - \frac{t}{2} - \frac{4}{3} + 21t, \quad t \in [0, 1],$$
$$u(0) = u'(0) = u''(0) = 0,$$

which has the exact solution $u(t) = t^4$. Tables 6, 12 shows that obtained solutions with RBF-SVSP method by increasing N are very accuracy in compared to the two-stage one-order Runge-Kutta method and variational iteration method (for n=6) and specially RBF-CSP method.

Example 6. Finally, consider the nonlinear first-order neutral functional-differential equation with proportional delay used in [3]

$$u'(t) = 1 - 2u^2(\frac{t}{2}), \quad t \in [0, 1],$$

subject to initial condition $u(0) = 0$. The exact solutions of this problem is $u(t) = \sin(t)$. Comparison of numerical results in Tables 7, 13 shows that GA-RBF method with SVSP is more accurate than the GA-RBF method with CSP.

5. Conclusion

In this work, an easy-to-code numerical method based on collocation method based on the radial basis functions with constant and variable shape parameter has been successfully applied to find the solutions of neutral delay differential equations. In the last section, by several examples, we showed that the collocation method based on the radial basis functions with SVSP can accurately solve a large variety of such problems including linear, nonlinear and high-order DDEs. It is concluded from figures (Fig1.-Fig 6.) that GA-RBF method with SVSP is an accurate and efficient method to solve neutral functional-differential equations. Finally, the numerical rate of convergence of the numerical approximates was also obtained. The advantage of SVSP strategy over CSP strategy can also be seen in the analysis of tables (Table 8 -Table 13), especially when the number of nodes tends to increase.

Acknowledgment

The authors would like to thank anonymous referees for their carefully reading the manuscript and such valuable comments, which has improved the manuscript significantly.

References

[1] H. Adibi and J. Es'haghi, *Numerical solution for biharmonic equation using multilevel radial basis functions and domain decomposition methods*. Appl. Math. Comput., *186* (2007), 246–255.

[2] M. D. Buhmann, *Radial basis functions*, Acta Numerica, *9* (2000), 1-38.

[3] M. Chamek, T. M. Elzaki and N. Brik, *Semi-analytical solution for some proportional delay differential equations*, SN Applied Sciences, *148*(1) (2019), DOI: 10.1007/s42452-018-0130-8.

[4] X. Chen and L. Wang, *The variational iteration method for solving a neutral functionaldifferential equation with proportional delays*, Computers and Mathematics with Applications, *59* (2010), 26962702.

[5] S. Davaeifar and J. Rashidinia, *Solution of a system of delay differential equations of multi pantograph type*, Journal of Taibah University for Science, *11* (2017), 11411157.

[6] M. Dehghan and A. Shokri, *Numerical solution of the nonlinear Klein–Gordon equation using radial basis functions*, J. Comput. Appl. Math., *230*(2) (2009), 400–410.

[7] M. Dehghan and M. Tatari, *Determination of a control parameter in a one-dimentional parabolic equation using the method of radial basis functions*, Math. Comp. Modeling, *44* (2006), 1160-1168.

[8] O. Farkhondeh Rouz, *Preserving asymptotic mean-square stability of stochastic theta scheme for systems of stochastic delay differential equations*, Computational Methods for Differential Equations, *8*(3) (2020), 468-479.

[9] R. Franke, *Scattered data interpolation: tests of some methods*, Math. Comput., *38* (1982), 181-200.

[10] F. Ghomanjani and M. H. Farahi, *The Bezier Control Points Method for Solving Delay Differential Equation*, Intelligent Control and Automation, *3* (2012), 188-196.

[11] A. Golbabai, M. Mammadova, and S. Seifollahi, *Solving a system of nonlinear integral equations by an RBF network*, Comput. Math. Appl., *57* (2009), 1651–1658.

[12] A. Golbabai and H. Rabiei, *Hybrid shape parameter strategy for the RBF approximation of vibrating systems*, International Journal of Computer Mathematics, *89*(17) (2012), 2410-2427.

[13] S. Gumgum, N. B. Savasanerial, O. K. Kurkcu, and M. Sezer, *Lucas polynomial solution for neutral differential equations with proportional delays*, TWMS Journal of Applied and Engineering Mathematics, *10*(1) (2020), 259-269.

[14] R. L. Hardy, *Multiquadric equations of topography and other irregular surfaces*, J. Geophy. Res., *76* (1971), 1905-1915.

[15] A. Isah and C. Phang, *Operational matrix based on Genocchi polynomials for solution of delay differential equations*, Ain Shams Engineering Journal, *9*(4) (2018), 2123-2128.

[16] E. Ishiwata and Y. Muroya, *Rational approximation method for delay differential equations with proportional delay*, Appl. Math. Comput., *187*(2) (2007), 741-747.

[17] E. Ishiwata, Y. Muroya, and H. Brunner, *A super-attainable order in collocation methods for differential equations with proportional delay*, Appl. Math. Comput., *198*(1) (2008), 227-236.

[18] E. J. Kansa, *Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics I: surface approximations and partial derivative estimates*, Computers and Mathematics with Applications, *19* (1990), 127145.

[19] E. J. Kansa, *Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics II: solutions to parabolic, hyperbolic, and elliptic partial differential equations*, Computers and Mathematics with Applications, *19* (1990), 147161.

[20] E. J. Kansa and R. E. Carlson, *Improved accuracy of multiquadric interpolation using variable shape parameters*, Comput. Math. Appl., *24* (1992), 99-120.

[21] E. J. Kansa and Y. C. Hon, *Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations*, Comput. Math. Appl., *39* (2000), 123-137.

[22] A. J. Khattak, S. I. A. Tirmizi, and S. U. Islam, *Application of meshfree collocation method to a class of nonlinear partial differential equations*, Eng. Anal. Bound. Elem., *33* (2009), 661–667.

[23] L. Khodayari and M. Ranjbar, *A Numerical Study of RBFs-DQ Method for Multi-Asset Option Pricing Problems*, Bol. Soc. Paran. Mat., *36*(1) (2018), 9–23.

[24] O. K. Kurkcu, E. Aslan, and M. Sezer, *A novel hybrid method for solving combined functional neutral differential equations with several delays and investigation of convergence rate via residual function*. Computational Methods for Differential Equations, *7*(3) (2019), 396-417.
[25] N. Mai-Duy, *Solving high order ordinary differential equations with radial basis function networks*, Int. J. Numer. Meth. Eng., *62* (2005), 824–852.
[26] M. Nouri M, *Solving Ito integral equations with time delay via basis functions*, Computational Methods for Differential Equations, *8*(2) (2020), 268-281.
[27] K. Parand, S. Abbasbandy, S. Kazem, and A. Rezaei, *Comparison between two common collocation approaches based on radial basis functions for the case of heat transfer equations arising in porous medium*. Commun. Nonlinear Sci. Numer. Simul., *16* (2011), 1396–1407.
[28] M. Ranjbar, H. Adibi and M. Lakestani, *Numerical solution of homogeneous Smoluchowski's coagulation equation*, Int. J. Comput. Math., *87*(9) (2010) 2113–2122.
[29] M. Ranjbar, *A new variable shape parameter strategy for Gaussian radial basis function approximation methods, Annals of the University of Craiova*, Mathematics and Computer Science Series, *42*(2) (2015), 260-272.
[30] J. Rashidinia and M. Khasi, *Stable Gaussian radial basis function method for solving Helmholtz equations*, Computational Methods for Differential Equations, *7*(1) (2019), 138-151.
[31] U. K. Sami and A. Ishtiaq, *Application of Legendre spectral-collocation method to delay differential and stochastic delay differential equation*, AIP ADVANCES **8**, 035301 (2018), Doi: 10.1063/1.5016680.
[32] S. A. Sarra and D. Sturgill, *A random variable shape parameter strategy for radial basis function approximation methods*, Engineering Analysis with Boundary Elements, *33* (2009), 1239-1245.
[33] A. Torabi Giklou, M. Ranjbar, M. Shaffee, and V. Roomi, *VIM-Padé technique for solving nonlinear and delay initial value problems*, Computational Methods for Differential Equations, (2020), Doi: 10.22034/cmde.2020.35417.1606.
[34] W. Wang and S. Li, *On the one-leg θ-methods for solving nonlinear neutral functional differential equations*, Appl. Math. Comput., *193*(1) (2007), 285-301.
[35] W. Wang, Y. Zhang, and S. Li, *Stability of continuous Runge-Kutta-type methods for nonlinear neutral delay-differential equations*, Appl. Math. Modell., *33*(8) (2009), 3319-3329.
[36] S. Xiang, K. M. Wang, Y. T. Ai, Yun-dong Sha, and H. Shi, *Trigonometric variable shape parameter and exponent strategy for generalized multiquadric radial basis function approximation*, Applied Mathematical Modelling, *36* (2012), 1931-1938.
[37] L. Xueqin and G. Yue, *The RKHSM for solving neutral functional-differential equations with proportional delays*, Math. Meth. Appl. Sci., *36* (2013), 642649.
[38] S. Yalcinbas, M. Aynigul, and M. Sezer, *A collocation method using Hermite polynomials for approximate solution of pantograph equations*, Journal of the Franklin Institute, *348*(6) (2011) 1128-1139.
[39] S. Yuzbasi and M. Karacayir, *A Galerkin-Like Approach to Solve Multi-Pantograph Type Delay Differential Equations*, FILOMAT, *32* (2018), 409-422.
[40] S. Yuzbasi, N. Sahin, and M.Sezer, *A Bessel Polynomial Approach For Solving Linear Neutral Delay Differential Equations With Variable Coefficients*, Journal Advanced Research in Differential Equations, *3* (2011), 81-101.
[41] S. Yuzbasi, N. Sahin, and M. Sezer, *Bessel Collocation Method For Numerical Solution Of Generalized Pantograph Equations*, Numerical Methods for Pratial Differential Equations, *28* (2012), 1105-1123.
[42] S. Yuzbasi and M. Sezer, *An exponential approximation for solutions of generalized pantograph-delay differential equations*, Applied Mathematical Modelling, *37*(22) (2013), 9160-9173.