Computer Physics Communications 184 (2013) 2145–2158

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

# The solitary wave solution of coupled Klein–Gordon–Zakharov equations via two different numerical methods

Mehdi Dehghan *, Ahmad Nikpour

*Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran, Iran*

ARTICLE INFO

*Article history:*
Received 18 July 2012
Received in revised form
15 March 2013
Accepted 19 April 2013
Available online 29 April 2013

*Keywords:*
Differential quadrature method
Klein–Gordon–Zakharov (KGZ) equations
Radial basis functions (RBFs)
Inverse Multiquadric (IMQ)
Thin Plate Spline (TPS)

ABSTRACT

In this research, we propose two different methods to solve the coupled Klein–Gordon–Zakharov (KGZ) equations: the Differential Quadrature (DQ) and Globally Radial Basis Functions (GRBFs) methods. In the DQ method, the derivative value of a function with respect to a point is directly approximated by a linear combination of all functional values in the global domain. The principal work in this method is the determination of weight coefficients. We use two ways for obtaining these coefficients: cosine expansion (CDQ) and radial basis functions (RBFs-DQ), the former is a mesh-based method and the latter categorizes in the set of meshless methods. Unlike the DQ method, the GRBF method directly substitutes the expression of the function approximation by RBFs into the partial differential equation. The main problem in the GRBFs method is ill-conditioning of the interpolation matrix. Avoiding this problem, we study the bases introduced in Pazouki and Schaback (2011) [44]. Some examples are presented to compare the accuracy and easy implementation of the proposed methods. In numerical examples, we concentrate on Inverse Multiquadric (IMQ) and second-order Thin Plate Spline (TPS) radial basis functions. The variable shape parameter (exponentially and random) strategies are applied in the IMQ function and the results are compared with the constant shape parameter.



## 1. Introduction

The plasma is modeled in terms of two intermingled fluids, the electron fluid and the ion fluid. Two timescales, fast and slow, arise from the wide difference between the mass of an electron and that of an ion. In response to a given force, the acceleration of an electron will be much greater than that of an ion, because its mass is roughly two thousand times less [1,2]. Choosing $u(x, t)$ as a complex function of the fast timescale component of the electric field raised by electrons, and $v(x, t)$ to be the real function of the deviation of ion density from its equilibrium, we can describe the interaction between the Langmuir waves and ion acoustic waves in a plasma by coupled Klein–Gordon–Zakharov equations [1]

$$\begin{aligned} &u_{tt} - u_{xx} + u + vu + |u|^2 u = 0, \\ &v_{tt} - v_{xx} = (|u|^2)_{xx}, \end{aligned} \tag{1}$$

with the initial conditions

$$\begin{cases} u(x, 0) = u_0(x); & u_t(x, 0) = u_1(x), \\ v(x, 0) = v_0(x); & v_t(x, 0) = v_1(x), \end{cases} \tag{2}$$

and the boundary conditions

$$\begin{cases} u(a, t) = u_a(t); & u(b, t) = u_b(t), \\ v(a, t) = v_a(t); & v(b, t) = v_b(t), \end{cases} \tag{3}$$

where $x \in \Omega = [a, b] \subset R$ and $0 \le t \le T$.

Time invariance, that is, invariance under the transformation $t \longrightarrow t + \delta t$, leads to the conservation of the quantity [3]

$$\int_a^b \left[ |u_t|^2 + |u_x|^2 + |u|^2 + v|u|^2 + \frac{1}{2}|m|^2 + \frac{1}{2}|v|^2 + \frac{1}{2}|u|^4 \right] dx = \text{const}, \tag{4}$$

for system (1)–(3) where $m$ is given by

$$m = -w_x, \qquad w_{xx} = v_t.$$

KGZ equations have a similar shape to Zakharov equations and the Klein–Gordon–Schrödinger equations. The existence of solutions and stability behavior for KGZ equations are discussed in [4–8]. A few numerical studies of Eqs. (1)–(3) are available. Wang et al. [3] applied two finite difference schemes [9] to find the numerical solution of problem (1)–(3). A finite difference scheme

* Corresponding author. Tel.: +98 21 64542503; fax: +98 21 66497930.
*E-mail addresses:* mdehghan@aut.ac.ir, mdehghan.aut@gmail.com (M. Dehghan), nikpour@aut.ac.ir (A. Nikpour).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.04.010

with a parameter $\theta$ was proposed by Chen and Zhang [10]. Also, Ghoreishi et al. [11] used Chebyshev cardinal functions and they employed the operational matrices of derivatives to reduce the PDE to nonlinear algebraic equations.

Some authors applied analytical methods and obtained an exact solution for KGZ equations [12–14]. Biswas et al. [12,13] obtained solitons and cnoidal waves of the Klein–Gordon–Zakharov equations with power law nonlinearity by applying the solitary wave Ansatz method, the travelling wave hypothesis method, the $(G'/G)$ method and the mapping method. Solitons are solitary waves with an elastic scattering property. They appear as a result of a balance between weak nonlinearity and dispersion. Solitons retain their shapes and speed after colliding with each other. The KdV equation is the pioneering model that gives rise to solitons [15]. Also, the review paper [16] gives an extensive overview of the soliton solutions for some well-known partial differential equations such as KdV, mKdV, sine-Gordon, and nonlinear Schrödinger equations. Different analytical methods of treatment as well as those of numerical techniques are presented in [16]. As is said in [16] solitons are a special kind of localized wave, essentially of a nonlinear kind.

Cnoidal wave theory is based on equations developed by Korteweg and de Vries [17]. The resulting equations contain Jacobian elliptical functions, commonly designated cn, so the name cnoidal is used to designate this wave theory [18]. Also, on some occasions, the solitons are the limit case of cnoidal waves [19].

In this paper, we study the solitary wave (soliton) of problem (1)–(3) by two different methods: the differential quadrature method and the globally radial basis functions method.

The differential quadrature (DQ) method was introduced by Bellman et al. [20] for the numerical solution of PDEs in 1972. So far many PDEs have been examined by the DQ method [21–23]. This method originated from the numerical integration approach and is a global method. The DQ method is an approximation to derivatives of a function using the weighted sum of the functional values at certain discrete points. Hence, due to direct approximation of derivatives, its implementation for nonlinear PDEs is almost easy. However the key point of the DQ method is to determine the weight coefficients. So far several techniques have been used to determine these coefficients such as Lagrange interpolation polynomials [24], Fourier series expansion [24], the Moving Least Squares (MLS) [25–27], Spline [28] and radial basis functions [29].

In this work we use Cosine expansion-based differential quadrature (CDQ) and RBF-based differential quadrature (RBF-DQ) methods for the numerical study of the coupled KGZ equations. Note that CDQ is a mesh-based method and RBF-DQ categorizes in a set of meshless methods.

We also utilize the globally radial basis functions (GRBFs) collocation method. This method has been widely applied in obtaining the numerical solutions of time dependent partial differential equations [30–37]. The application of RBFs based on collocation in a set of scattered nodes for the solution of PDEs was first proposed by Kansa [38,39]. Unlike the DQ method, the GRBFs method directly substitutes the expression of function approximation by RBFs into the PDEs.

Although the GRBFs method has advantages such as ease of programming, potential spectral accuracy and not needing to generate a mesh on the domain, it suffers from ill-conditioning of the interpolation matrix [40]. One technique to eliminate this problem is based on the basis transformation from a very ill-conditioned one (made of the translations of the radial function) to a much better conditioned one, which spans the same space. The use of algorithms based on basis transformation was first proposed by authors of [41,42]. They combined series expansions of the kernel with a QR decomposition of the interpolation matrix and obtained the so-called RBF-QR algorithm. Piret and Hanert [43] developed this algorithm for the numerical study of fractional diffusion equations. Recently, Pazouki and Schaback [44,45] obtained bases for spanned space by positive definite and conditionally positive definite RBFs. We will discuss more on these bases in Section 4.

Inverse Multiquadric (positive definite) and second-order Thin Plate Spline (conditionally positive definite) are applied in the GRBFs method. Furthermore, we use the bases introduced in [44] for spanned space by the IMQ function and compare them with IMQ in two different properties: the condition number of the interpolation matrix and the optimal interval of shape parameter.

The organization of this paper is as follows. In Section 2 we describe the numerical methods CDQ, RBF-DQ and GRBFs. In Section 3 we apply these methods on the coupled Klein–Gordon–Zakharov equations. The process of obtaining the new bases for spanned space by positive definite RBFs is explained in Section 4. The numerical methods are tested on two examples and the numerical results are presented in Section 5. Finally Section 6 is dedicated to a brief conclusion. Note that we have computed the numerical results by Matlab programming.

## 2. Numerical methods

### 2.1. The differential quadrature (DQ) method

Based on the differential quadrature theory, the derivative value $u^{(l)}(x)$ with respect to $x$ at a point $x_i$ is approximated by a linear combination of all functional values in the global domain by

$$u^{(l)}(x_i) \simeq \sum_{j=1}^{N} w_{ij}^{(l)} u(x_j); \quad i = 1, \ldots, N, \tag{5}$$

where $w_{ij}^{(l)}$ represent the weight coefficients, and $N$ is the number of points in the global domain. So far authors used several sets of bases such as *Lagrange interpolation polynomials* [24], *Cosine expansion* [24], *MLS* [25,26] and *RBFs* [29] to obtain these coefficients. The weight coefficients related to *polynomials* and *Cosine* basis are explicitly specified by Shu [24]. In the next two subsections we describe the differential quadrature method based on *Cosine* basis and *RBFs* approaches.

#### 2.1.1. The Cosine expansion based differential quadrature (CDQ) method

Fourier series expansion of an even function $u(x)$ defined on the interval $[0, \pi]$ can be written as [24]

$$u(x) \simeq U_{N+1}(x) = a_0 + \sum_{j=1}^{N} a_j \cos(jx). \tag{6}$$

$U_{N+1}(x)$ given by Eq. (6) constitutes a linear vector space $V_{N+1}$. Two sets of base vectors in $V_{N+1}$ can be chosen as

$$c_k(x) = \cos(kx), \quad k = 0, 1, \ldots, N, \tag{7}$$

$$c_k(x) = \frac{C(x)}{p(x_k)(\cos x - \cos x_k)}, \quad k = 0, 1, \ldots, N, \tag{8}$$

where

$$C(x) = \prod_{k=0}^{N} (\cos x - \cos x_k),$$

$$p(x_i) = \prod_{k=0, k\neq i}^{N} (\cos x_i - \cos x_k).$$

If we use the bases (7) and (8) to determine the weight coefficients in (5), then the weighting coefficients of the first-order derivative are found as [24]

$$\begin{cases} w_{ij}^{(1)} = \dfrac{-p(x_i)\sin x_i}{p(x_j)(\cos x_i - \cos x_j)}, \\ \quad i \neq j \text{ and } i, j = 0, 1, \ldots, N, \\ w_{ii}^{(1)} = -\displaystyle\sum_{j=0, j\neq i}^{N} w_{ij}^{(1)}. \end{cases} \tag{9}$$

To determine weight coefficients of the second-order derivative, the formula

$$\begin{cases} w_{ij}^{(2)} = w_{ij}^{(1)}\left(2w_{ii}^{(1)} + \dfrac{2\sin x_i}{\cos x_i - \cos x_j} + \cot x_i\right), \\ \quad i \neq j \text{ and } i, j = 0, 1, \ldots, N, \\ w_{ii}^{(2)} = -\displaystyle\sum_{j=0, j\neq i}^{N} w_{ij}^{(2)} \end{cases} \tag{10}$$

is used.

For the case that the interval is not $[0, \pi]$ but $[a, b]$, a variable change can be used as

$$y = \frac{x-a}{b-a}\pi,$$

to map the interval $[a, b]$ in the $x$ domain to the interval $[0, \pi]$ in the $y$ domain. By noting this variable change, the weight coefficients (9)–(10) are updated as [24]

$$\begin{cases} w_{ij}^{(1)} = \dfrac{-\alpha p(y_i)\sin y_i}{p(y_j)(\cos y_i - \cos y_j)}, \\ \quad i \neq j \text{ and } i, j = 0, 1, \ldots, N, \\ w_{ii}^{(1)} = -\displaystyle\sum_{j=0, j\neq i}^{N} w_{ij}^{(1)}. \end{cases} \tag{11}$$

$$\begin{cases} w_{ij}^{(2)} = w_{ij}^{(1)}\left(2w_{ii}^{(1)} + \dfrac{2\alpha\sin y_i}{\cos y_i - \cos y_j} + \alpha\cot y_i\right), \\ \quad i \neq j \text{ and } i, j = 0, 1, \ldots, N, \\ w_{ii}^{(2)} = -\displaystyle\sum_{j=0, j\neq i}^{N} w_{ij}^{(2)} \end{cases} \tag{12}$$

where $\alpha = \pi/(b-a)$. For more details we refer the interested reader to [46,47].

### 2.1.2. RBF based differential quadrature (RBF-DQ) method

In the RBF-DQ method, for determining weight coefficients, the radial basis functions are used as test functions [29]. Because of using RBFs, this method is a meshless method, while the CDQ method that was reviewed in the previous subsection is a mesh-based method. Applying RBFs as a test function, the relation (5) changes to the following form:

$$\varphi^{(l)}(\|x_i - x_k\|) = \sum_{j=1}^{N} w_{ij}^{(l)}\varphi_k(x_j); \quad k = 1, \ldots, N, \tag{13}$$

where $\varphi_k(x_j) = \varphi(\|x_j - x_k\|)$ and $N$ is the number of nodes in the global domain. The function $\varphi$ in (13) is the radial basis function. Common choices for conditionally positive definite RBFs of order $m$ are listed in Table 1. We can write relation (13) in the matrix form as $\Psi_{x_i}^{(l)} = A_{n_i} w_{x_i}^{(l)}$, where

**Table 1**
The definitions of some radial basis functions ($r = \|x - x_j\|$, $c > 0$).

| Name of functions | Definition | Order $m$ |
|---|---|---|
| Gaussian (GA) | $\varphi(r) = e^{-cr^2}$ | 0 |
| Multiquadric (MQ) | $\varphi(r) = (r^2 + c^2)^{1/2}$ | 1 |
| Inverse quadratic (IQ) | $\varphi(r) = \frac{1}{r^2+c^2}$ | 0 |
| Inverse Multiquadric (IMQ) | $\varphi(r) = (r^2 + c^2)^{-1/2}$ | 0 |
| $k$-order Thin Plate Spline (TPS) | $\varphi(r) = r^{2k}\log r$ | $k+1$ |

$$\Psi_{x_i}^{(l)} = \left[\varphi_1^{(l)}(x_i)\, \varphi_2^{(l)}(x_i) \cdots \varphi_N^{(l)}(x_i)\right]^T,$$

$$A = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_N(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_N) & \varphi_2(x_N) & \cdots & \varphi_N(x_N) \end{bmatrix},$$

$$w_{x_i}^{(l)} = \left[w_{i1}^{(l)} w_{i2}^{(l)} \cdots w_{iN}^{(l)}\right]^T.$$

The matrix $A$ is (conditionally) positive definite for RBFs, so the nonsingularity $A$ is guaranteed on distinct nodes and the corresponding coefficients are denoted by

$$w_{x_i}^{(l)} = A^{-1}\Psi_{x_i}^{(l)}. \tag{14}$$

For finding the weight coefficients corresponding to each node, we should solve a linear algebraic system as above. But since the matrix $A$ in (14) is unchanged for various nodes, we use the LU factorization only once and use this factorization to find the weight coefficients corresponding to each node.

### 2.2. Globally radial basis functions (GRBFs) method

In the GRBFs method, the function $u(x)$ on a finite set $X = \{x_1, \ldots, x_N\}$ is directly approximated using the elements of space [48]

$$S_m(X) := \mathcal{P}_m^d + \left\{\sum_{x_j\in X} c_j\varphi_j(.) : X \text{ is } \mathcal{P}_m^d\text{-unisolvent}, \right. \\ \left. \mathbf{c} = (c_1, \ldots, c_N) \in M\right\}, \tag{15}$$

where

$$M := \left\{\mathbf{c} \in R^N : \sum_{j=1}^{N} c_j p(x_j) = 0 \text{ for all } p \in \mathcal{P}_m^d\right\},$$

and $\mathcal{P}_m^d$ are $d$-variate real-valued polynomials of order at most $m$ and radial function $\varphi$ is a fixed conditionally positive definite function of order $m$. Hence

$$u(x) \simeq \sum_{j=1}^{N}\lambda_j\varphi_j(x) + \sum_{j=N+1}^{N+Q}\lambda_j p_{j-N}(x), \tag{16}$$

where $N$ is the total number of interpolation points $x_i$ ($N-2$ interior points and 2 boundary points) and set $\{p_1(x), \ldots, p_Q(x)\}$ is a base for polynomial space $\mathcal{P}_m^d$. The coefficients $(\lambda_1, \lambda_2, \ldots, \lambda_{N+Q})$ are unknown. To determine these coefficients, it is necessary to apply the following additional conditions [48]

$$\sum_{j=1}^{N}\lambda_j p_i(x_j) = 0, \quad 1 \leq i \leq Q. \tag{17}$$

With collocation (16)–(17) at center points, we can write these relations in the matrix form as

$$[u] = \mathcal{A}[\lambda], \tag{18}$$

where $[u] = [u_1, \ldots, u_N, 0, \ldots, 0]$, $[\lambda] = [\lambda_1, \ldots, \lambda_{N+Q}]$ and

$$\mathcal{A} = \begin{pmatrix} A & P \\ P^T & 0_{Q\times Q} \end{pmatrix}, \qquad A = (\varphi_j(x_i))_{1\le i,j\le N},$$

$$P = (p_i(x_j))_{1\le i\le Q,\ 1\le j\le N}.$$

The solution of system (18) gives the coefficients $\lambda_j$. We can find the values of derivatives $u^{(l)}(x)$ and $\bar{u}^{(l)}(x)$ of the approximate solution (16) as

$$u^{(l)}(x) \simeq \sum_{j=1}^{N} \lambda_j \varphi_j^{(l)}(x) + \sum_{j=N+1}^{N+Q} \lambda_j p_{j-N}^{(l)}(x), \tag{19}$$

$$\bar{u}^{(l)}(x) \simeq \sum_{j=1}^{N} \bar{\lambda}_j \varphi_j^{(l)}(x) + \sum_{j=N+1}^{N+Q} \bar{\lambda}_j p_{j-N}^{(l)}(x). \tag{20}$$

## 3. Implementation of the numerical methods

First, we present the time discretization for Eqs. (1).

We can rewrite the expression $(|u|^2)_{xx}$ as follows:

$$(|u|^2)_{xx} = (u\bar{u})_{xx} = (u_x\bar{u} + u\bar{u}_x)_x = u_{xx}\bar{u} + 2u_x\bar{u}_x + u\bar{u}_{xx}, \tag{21}$$

where $\bar{u}$ is the conjugate of complex number $u$. Noting the above relation we discretize (1) according to the Crank–Nicolson type scheme

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{(\delta t)^2} - \frac{(u_{xx})^{n+1} + (u_{xx})^n}{2} + \frac{u^{n+1} + u^n}{2} + u^n v^n + |u^n|^2 u^n = 0, \tag{22}$$

$$\frac{v^{n+1} - 2v^n + v^{n-1}}{(\delta t)^2} - \frac{(v_{xx})^{n+1} + (v_{xx})^n}{2} = u_{xx}^{n+1}\bar{u}^n + 2u_x^{n+1}\bar{u}_x^{n+1} + u^n\bar{u}_{xx}^{n+1}, \tag{23}$$

where $u^{n+1} = u(x, t^{n+1})$, $t^{n+1} = t^n + \delta t$ and $\delta t$ is the time step size.

With rearranging (22) and (23), we obtain

$$\left(1 + \frac{(\delta t)^2}{2}\right) u^{n+1} - \frac{(\delta t)^2}{2}(u_{xx})^{n+1} = \left(2 - \frac{(\delta t)^2}{2}\right) u^n - u^{n-1} + \frac{(\delta t)^2}{2}(u_{xx})^n - (\delta t)^2 u^n v^n - (\delta t)^2 |u^n|^2 u^n, \tag{24}$$

$$v^{n+1} - \frac{(\delta t)^2}{2}(v_{xx})^{n+1} = 2v^n - v^{n-1} + \frac{(\delta t)^2}{2}(v_{xx})^n + (\delta t)^2(u_{xx}^{n+1}\bar{u}^n + 2u_x^{n+1}\bar{u}_x^{n+1} + u^n\bar{u}_{xx}^{n+1}). \tag{25}$$

Now, we apply the methods introduced in the previous section for the space discretization of Eqs. (24)–(25).

### 3.1. The differential quadrature (DQ) method

Implementation of CDQ and RBF-DQ methods for solving KGZ equations is the same and the only difference between them is the choice of basis functions to determine the weight coefficients. So substitution of Eq. (5) into discretized Eqs. (24)–(25) yields

$$(24) \Longrightarrow \left(1 + \frac{(\delta t)^2}{2}(1 - w_{ii}^{(2)})\right) u_i^{n+1} - \sum_{j=0, j\ne i}^{N} \frac{(\delta t)^2}{2} w_{ij}^{(2)} u_j^{n+1} = \left(2 - \frac{(\delta t)^2}{2}\right) u_i^n - u_i^{n-1} + \frac{(\delta t)^2}{2}(Bu)_i^n - (\delta t)^2 u_i^n v_i^n - (\delta t)^2 |u_i^n|^2 u_i^n,$$

$$(25) \Longrightarrow \left(1 - \frac{(\delta t)^2}{2} w_{ii}^{(2)}\right) v_i^{n+1} - \sum_{j=0, j\ne i}^{N} \frac{(\delta t)^2}{2} w_{ij}^{(2)} v_j^{n+1} = 2v_i^n - v_i^{n-1} + \frac{(\delta t)^2}{2}(Bv)_i^n + (\delta t)^2((Bu)_i^{n+1}\bar{u}_i^n + 2(Au)_i^{n+1}(\bar{A}u)_i^{n+1} + u_i^n(\bar{B}u)_i^{n+1}),$$

where $Au_i^n = \sum_{j=0}^{N} w_{ij}^{(1)} u_j^n$, $Bu_i^n = \sum_{j=0}^{N} w_{ij}^{(2)} u_j^n$, $Bv_i^n = \sum_{j=0}^{N} w_{ij}^{(2)} v_j^n$.

After imposing the boundary conditions, we can write the above relations in the matrix notation as

$$\mathbf{Du} = \mathbf{f}, \qquad \mathbf{Cv} = \mathbf{g}, \tag{26}$$

where the elements of $\mathbf{D}$, $\mathbf{C}$, $\mathbf{f}$ and $\mathbf{g}$ are

$$d_{ii} = 1 + \frac{(\delta t)^2}{2}(1 - w_{ii}^{(2)}), \qquad d_{ij} = -\frac{(\delta t)^2}{2} w_{ij}^{(2)}, \quad i \ne j,$$

$$c_{ii} = 1 - \frac{(\delta t)^2}{2} w_{ii}^{(2)}, \qquad c_{ij} = -\frac{(\delta t)^2}{2} w_{ij}^{(2)}, \quad i \ne j,$$

$$f_i^n = \left(2 - \frac{(\delta t)^2}{2}\right) u_i^n - u_i^{n-1} + \frac{(\delta t)^2}{2}(Bu)_i^n - (\delta t)^2 u_i^n v_i^n - (\delta t)^2 |u_i^n|^2 u_i^n + \frac{(\delta t)^2}{2}(w_{i0}^{(2)} u_0^n + w_{iN}^{(2)} u_N^n),$$

and

$$g_i^n = 2v_i^n - v_i^{n-1} + \frac{(\delta t)^2}{2}(Bv)_i^n + (\delta t)^2((Bu)_i^{n+1}\bar{u}_i^n + 2(Au)_i^{n+1}(\bar{A}u)_i^{n+1} + u_i^n(\bar{B}u)_i^{n+1}) + \frac{(\delta t)^2}{2}(w_{i0}^{(2)} v_0^n + w_{iN}^{(2)} v_N^n),$$

respectively.

Due to values of the function $\cot y$ at the boundaries 0 and $\pi$, we cannot compute the weighting coefficient $w_{i0}^{(2)}$ using Eq. (12). If the boundary conditions are zero, we do not need to calculate $w_{i0}^{(2)}$ and otherwise take them as zero to have a solvable system [46].

### 3.2. The globally RBFs (GRBFs) method

By using the notations in Section 2.2, we approximate the $n$th step solutions $u(x, t^n)$ and $v(x, t^n)$ as [30,35,38]

$$u^n(x) \simeq \sum_{j=1}^{N} \lambda_j^n \varphi_j(x) + \sum_{j=N+1}^{N+Q} \lambda_j^n p_{j-N}(x), \tag{27}$$

$$v^n(x) \simeq \sum_{j=1}^{N} \gamma_j^n \varphi_j(x) + \sum_{j=N+1}^{N+Q} \gamma_j^n p_{j-N}(x). \tag{28}$$

Also, the numerical derivative values (27)–(28) are computed by relations (19)–(20).

Substituting Eqs. (27)–(28) into Eqs. (24)–(25) and applying the collocation technique in the points $x_i$, $i = 1, 2, \ldots, N$, we obtain

$$
\begin{aligned}
(24) \Longrightarrow\ & \left(1+\frac{(\delta t)^2}{2}\right)\left(\sum_{j=1}^{N}\lambda_j^{n+1}\varphi(r_{ij})+\sum_{j=N+1}^{N+Q}\lambda_j^{n+1}p_{j-N}(x_i)\right)\\
&-\frac{(\delta t)^2}{2}\left(\sum_{j=1}^{N}\lambda_j^{n+1}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\lambda_j^{n+1}p''_{j-N}(x_i)\right)=\left(2-\frac{(\delta t)^2}{2}\right)u_i^n\\
&-u_i^{n-1}+\frac{(\delta t)^2}{2}\left(\sum_{j=1}^{N}\lambda_j^{n}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\lambda_j^{n}p''_{j-N}(x)\right)-(\delta t)^2u_i^nv_i^n\\
&-(\delta t)^2|u_i^n|^2u_i^n, \qquad (29)\\
(25) \Longrightarrow\ & \sum_{j=1}^{N}\gamma_j^{n+1}\varphi(r_{ij})+\sum_{j=N+1}^{N+Q}\gamma_j^{n+1}p_{j-N}(x)\\
&-\frac{(\delta t)^2}{2}\left(\sum_{j=1}^{N}\gamma_j^{n+1}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\gamma_j^{n+1}p''_{j-N}(x)\right)=2v_i^n-v_i^{n-1}\\
&+\frac{(\delta t)^2}{2}\left(\sum_{j=1}^{N}\gamma_j^{n}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\gamma_j^{n}p''_{j-N}(x)\right)\\
&+(\delta t)^2\left(\left(\sum_{j=1}^{N}\lambda_j^{n+1}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\lambda_j^{n+1}p''_{j-N}(x_i)\right)\bar{u}_i^n\right.\\
&+2\left(\sum_{j=1}^{N}\lambda_j^{n+1}\varphi'(r_{ij})+\sum_{j=N+1}^{N+Q}\lambda_j^{n+1}p'_{j-N}(x_i)\right)\left(\sum_{j=1}^{N}\bar{\lambda}_j^{n+1}\varphi'(r_{ij})+\sum_{j=N+1}^{N+Q}\bar{\lambda}_j^{n+1}p'_{j-N}(x_i)\right)\\
&\left.+u_i^n\left(\sum_{j=1}^{N}\bar{\lambda}_j^{n+1}\varphi''(r_{ij})+\sum_{j=N+1}^{N+Q}\bar{\lambda}_j^{n+1}p''_{j-N}(x_i)\right)\right), \qquad (30)
\end{aligned}
$$

and the boundary conditions

$$
\begin{aligned}
&\sum_{j=1}^{N}\lambda_j^{n}\varphi\left(r_{ij}\right)+\sum_{j=N+1}^{N+Q}\lambda_j^{n}p_{j-N}(x_i)=u\left(x_i,t^{n+1}\right),\\
&i=1,N, \qquad (31)\\
&\sum_{j=1}^{N}\gamma_j^{n}\varphi\left(r_{ij}\right)+\sum_{j=N+1}^{N+Q}\gamma_j^{n}p_{j-N}(x_i)=v\left(x_i,t^{n+1}\right),\\
&i=1,N, \qquad (32)
\end{aligned}
$$

where $r_{ij} = \|x_i - x_j\|$. Eqs. (29) and (31) together with the additional conditions

$$\sum_{j=1}^{N}\lambda_j p_i(x_j)=0,\quad 1\le i\le Q,$$

and Eqs. (30) and (32) together with the additional conditions

$$\sum_{j=1}^{N}\gamma_j p_i(x_j)=0,\quad 1\le i\le Q,$$

form two systems of linear algebraic equations $(N+Q)\times(N+Q)$ in the unknowns $\lambda_j^{n+1}$ and $\gamma_j^{n+1}$, respectively. Since the coefficient matrices of these systems are unchanged in time steps, we use the LU factorization only once and use this factorization to find solutions in each time step. After computing the unknowns $\lambda_j^{n+1}$ and $\gamma_j^{n+1}$, we can find the approximate solutions at each time step using Eqs. (27)–(28) by

$$
\begin{aligned}
&[u]^n=\mathcal{A}[\lambda]^n,\\
&[v]^n=\mathcal{A}[\gamma]^n,
\end{aligned}
$$

where

$$
\begin{aligned}
&[u]^n=[u_1^n,\ldots,u_N^n,0,\ldots,0],\\
&[v]^n=[v_1^n,\ldots,v_N^n,0,\ldots,0],\\
&[\lambda]^n=[\lambda_1^n,\ldots,\lambda_{N+Q}^n],\\
&[\gamma]^n=[\gamma_1^n,\ldots,\gamma_{N+Q}^n].
\end{aligned}
$$

Also, the values $[\lambda]^0$ and $[\gamma]^0$ are computed using the initial conditions

$$
\begin{aligned}
&[\lambda]^0=\mathcal{A}^{-1}[u]^0,\\
&[\gamma]^0=\mathcal{A}^{-1}[v]^0.
\end{aligned}
$$

## 4. New bases for spanned space by positive definite RBFs

In implementation of the GRBFs method, for obtaining the values $\lambda_j^0$ and $\gamma_j^0$, we need the inverse of matrix $\mathcal{A}$. But as we know if interpolation on a finite set $X = \{x_1, \ldots, x_N\}$ of centers is done by translates $\varphi(\|x - x_j\|)$ for $x \in X$ of a fixed radial function $\varphi$, then the basis spanned by this function often is badly conditioned. So far two different classes of schemes have been applied to eliminate the ill-conditioning of interpolation matrix RBFs in numerical solution of PDEs. The first class includes LU decomposition schemes, good matrix preconditioners [49], domain decomposition [50,51], the compactly supported RBFs [52,53] and locally supported RBFs [54,55]. In some of these techniques (e.g. LU decomposition) the bases of the spanned space by positive definite RBFs do not need to be changed. In the other set of these techniques (e.g. compactly supported RBFs) not only the bases are changed but the new bases span the new approximation space, which is different with the principal approximation space (spanned space by positive definite RBFs). But the second set of techniques is based on the basis transformation from a very ill-conditioned one (made of the translates of the radial function) to a much better conditioned one, which spans the same exact space. Making such a basis change is exactly what we do when we go from the ill-conditioned monomials $1, x, x^2, \ldots, x^n$ on $[-1, 1]$ to the much better conditioned Chebyshev basis $T_0(x), T_1(x), T_2(x), \ldots, T_n(x)$ [56]. Authors of

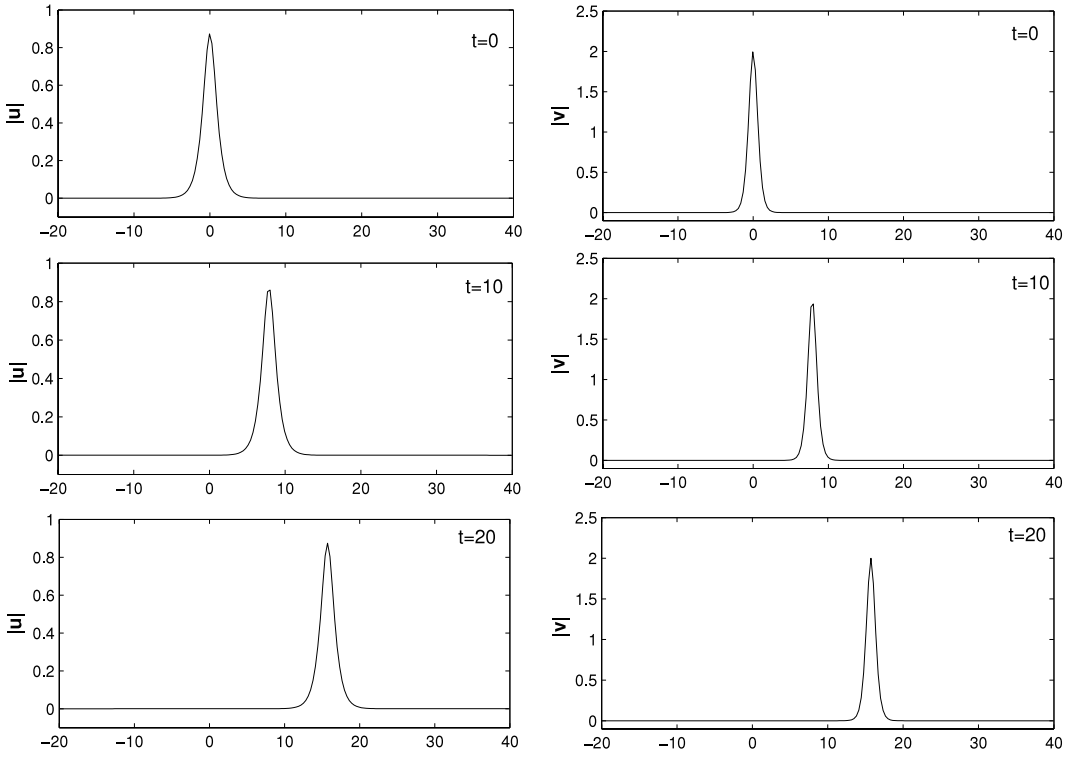


**Fig. 1.** Exact solution for the single soliton at different times.

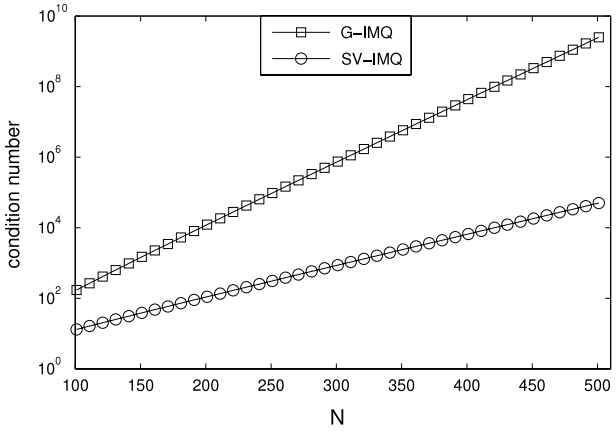


**Fig. 2.** Condition number of interpolation matrix in G-IMQ and SV-IMQ methods.

[41,42] combined series expansions of the Gaussian radial function with a QR decomposition of the interpolation matrix and obtained an algorithm named the RBF-QR algorithm. Recently, using Cholesky and SVD decompositions of the interpolation matrix of positive definite radial basis functions, Pazouki and Schaback [44] obtained bases for spanned spaces by these functions. In numerical examples we use bases produced by SVD and Cholesky decompositions of the interpolation matrix of positive definite Inverse Multiquadrics (IMQ). We denote these bases by SV-IMQ and CH-IMQ, respectively.

### 4.1. Cholesky and SVD decompositions

Here we follow the approach presented in [44]. According to the linear algebra, if $\Phi = \{\phi(\|x-x_1\|), \phi(\|x-x_2\|), \ldots, \phi(\|x-x_N\|)\}$ is a base for $N$-dimensional finite space $S_N$ then for each $N \times N$ nonsingular matrix $C$, the function

$$\Psi = \Phi \cdot C, \tag{33}$$

is also a base for $S_N$. As suggested in [44], we call matrix $C$ as a construction matrix and denote it by $C_\Psi$. Therefore the new bases $\Psi$ are produced with various choices of $C_\Psi$.

As introduced in Section 2.2, to approximate function $u$ by positive definite radial basis function $\phi$ we should solve the linear algebraic system

$$V_\Phi \lambda = u, \tag{34}$$

where $V_\Phi = (\phi(\|x_i - x_j\|))_{1\le i,j\le N,}$. If, instead of base $\Phi$, we use base $\Psi$ then the system (34) is updated as $V_\Psi \lambda = u$ where $V_\Psi$ (see relation (33)) is defined as

$$V_\Psi = V_\Phi \cdot C_\Psi. \tag{35}$$

We call matrices $V_\Phi$ and $V_\Psi$ as the value matrices of bases $\Phi$ and $\Psi$, respectively. So the key point is determination of the construction matrix $C_\Psi$ so that the new value matrix $V_\Psi$ is more well-conditioned than the original value matrix $V_\Phi$. One way is introduced by Cholesky decomposition of matrix $V_\Phi$ [44]. If the Cholesky decomposition of matrix $V_\Phi$ is $V_\Phi = LL^T$ where $L$ is a nonsingular lower triangular matrix, then with setting $C_\Psi = (L^T)^{-1}$, the value matrix of the new base is $V_\Psi = L$. So for obtaining the values $\lambda_j^0$ and $\gamma_j^0$ in the first step of the GRBFs algorithm, we need to have the inverse of matrix $V_\Psi = L$, which is much better conditioned than the original matrix $V_\Phi$ [44].

The other case is induced by singular value decomposition (SVD) in the form $V_\Phi = Q^T \cdot \sum^2 \cdot Q$ with an orthogonal matrix $Q$ and a diagonal matrix $\sum$ having the eigenvalues of $V_\Phi$ on its diagonal [44]. With assumption $C_\Psi = Q^T \cdot \sum^{-1}$ as the construction matrix, the value matrix is obtained from relation (35) as $V_\Psi = Q^T \cdot \sum$. So the inverse of matrix $V_\Psi$ is $V_\Psi^{-1} = \sum^{-1} \cdot Q$, indeed for obtaining the values $\lambda_j^0$ and $\gamma_j^0$ we need to have the inverse of the diagonal matrix $\sum$.

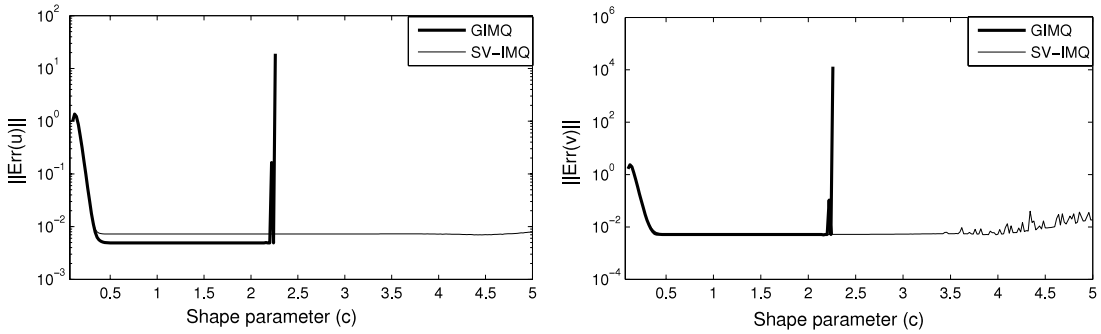


**Fig. 3.** Comparison of optimal interval of shape parameter in G-IMQ and SV-IMQ methods.

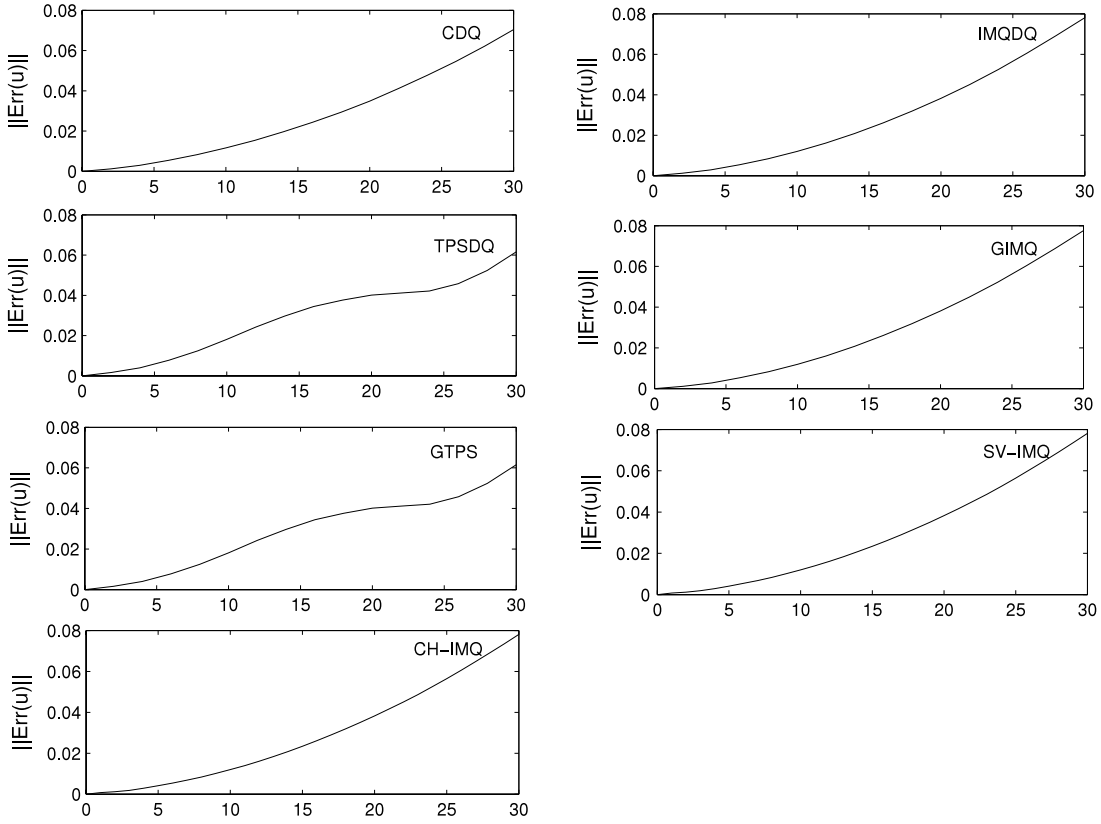


**Fig. 4.** Stability of different methods for the single soliton at different times with $x \in [-20, 30]$, $h = 0.2$ and $\delta t = 0.001$.

## 5. Numerical results

In this section we will study the solitary wave solutions (solitons) of system (1)–(3). A soliton can be defined as a solution of a nonlinear partial differential equation that exhibits the following properties [15,16]:

(i) the solution should demonstrate a wave of permanent form,
(ii) the solution is localized, which means that the solution either decays exponentially to zero such as the solitons provided by the KdV equation, or converges to a constant at infinity such as the solitons given by the Sine-Gordon equation,
(iii) the soliton interacts with other solitons preserving its character.

To compare the accuracy and efficiency of the numerical methods introduced in the current paper, we solve two test examples. We use $L_\infty$ error norm which is defined by

$$L_\infty = \max_{1\le i\le N} \|u_{\text{exact}}(i) - u_{\text{app}}(i)\|,$$

to measure the accuracy.

In the RBF-DQ and GRBFs methods we concentrate on Inverse Multiquadric (IMQ) and second-order Thin Plate Spline (TPS) radial basis functions. IMQ is a positive definite function and the second-order TPS is a conditionally positive definite function with $m = 3$ (conditionally order). So the base of polynomial space in (15) for the second-order TPS consists of $\{1, x, x^2\}$. Also, we apply SVD and Cholesky decompositions for the produced matrix by the IMQ function and obtain the new bases for spanned space by this function. We denote the result of bases by SV-IMQ and CH-IMQ, respectively. Using these two new bases in the GRBFs method, we compare them with IMQ in two different properties: the condition

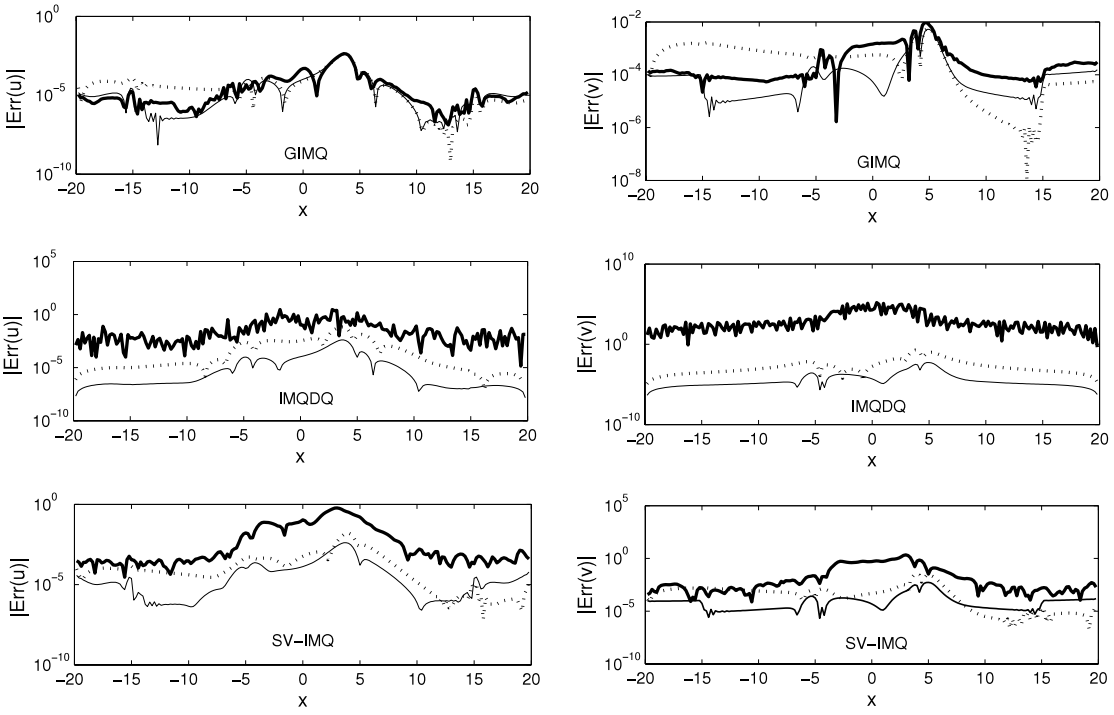


**Fig. 5.** Error distribution for the single soliton at different shape parameters, solid line: constant shape parameter, dot line: exponential shape parameter, bold solid line: random shape parameter with $x \in [-20, 20]$, $h = 0.2$, $T = 5$, $\delta t = 0.001$, $c_{\min} = 0.3$ and $c_{\max} = 0.7$.

number of the interpolation matrix and the optimal interval of the shape parameter. The exact solution of the KGZ equations (1) for the single soliton is [13]

$$\begin{cases} u(x,t) = \mu\ \operatorname{sech}(\rho(x - \vartheta t))\exp(i(-\kappa x + \omega t)), \\ v(x,t) = \xi\ \operatorname{sech}^2(\rho(x - \vartheta t)), \end{cases} \tag{36}$$

where $\mu$ and $\xi$ are the amplitudes of the $u$ and $v$ solitons respectively. Also, $\rho$ is the inverse width of the soliton and $\vartheta$ is the soliton velocity. Note that $-\kappa x + \omega t$ represents the phase of the soliton with $\kappa$ as the soliton frequency and $\omega$ as the soliton wave number.

The following relations are satisfied between $\mu, \xi, \rho, \vartheta, \omega$ and $\kappa$ [13]:

$$\begin{aligned} &\mu = \sqrt{\xi(\vartheta^2 - 1)}, \qquad \rho = \pm\sqrt{\frac{\xi + \mu^2}{2(\vartheta^2 - 1)}}, \\ &\omega = \pm\sqrt{\frac{2 + \xi + \mu^2}{2(1 - \vartheta^2)}}, \qquad \kappa = \pm\vartheta\omega. \end{aligned} \tag{37}$$

### 5.1. Single solitary wave

In this test with the choice $\mu = \frac{\sqrt{10}-\sqrt{2}}{2}$, $\vartheta = \sqrt{\frac{\sqrt{5}-1}{2}}$ in (36) and the negative value of $\omega$ and positive values for $\rho$ and $\kappa$ in (37), we obtain [57,58]

$$\begin{cases} u(x,t) = \dfrac{\sqrt{10}-\sqrt{2}}{2}\operatorname{sech}\left(\sqrt{\dfrac{1+\sqrt{5}}{2}}x - t\right) \\ \quad \times \exp\left[i\left(\sqrt{\dfrac{2}{1+\sqrt{5}}}x - t\right)\right], \\ v(x,t) = -2\operatorname{sech}^2\left(\sqrt{\dfrac{1+\sqrt{5}}{2}}x - t\right), \end{cases}$$

with the initial conditions

$$\begin{cases} u_0(x) = \dfrac{\sqrt{10}-\sqrt{2}}{2}\operatorname{sech}\left(\sqrt{\dfrac{1+\sqrt{5}}{2}}x\right) \\ \quad \times \exp\left[i\left(\sqrt{\dfrac{2}{1+\sqrt{5}}}x\right)\right], \\ v_0(x) = -2\operatorname{sech}^2\left(\sqrt{\dfrac{1+\sqrt{5}}{2}}x\right). \end{cases}$$

Fig. 1 shows the graph of the exact solutions at several times. The $L_\infty$ error in approximation solutions for the methods introduced in the current paper is obtained in Tables 2–4 for $T = 1$, 5 and 10, respectively. In each table we consider three different time steps $\delta t = 0.01$, 0.001 and 0.0005. As is observed, when the time step is refined, the accuracy of the methods increases. Fig. 6 displays the space–time numerical solution at different time levels up to $t = 12$. It is clear that the waves move towards the right with the increase of time. By Table 5 we can see the present methods conserve the conservation quantity (4) by almost two decimal places. From Fig. 2 and Table 7 it can be seen that the condition number of the interpolation matrix of the GIMQ method is almost two times the condition number of the interpolation matrix of the SV-IMQ method. Hence, as is illustrated in Fig. 3, the optimal interval of the shape parameter ($c$) in SV-IMQ is larger than the GIMQ method. Also, unlike the GRBFs method, the produced matrix by the collocation technique in Section 3 for DQ methods (CDQ and RBF-DQ) is well-conditioned (see Table 8). Fig. 4 presents the numerical errors at different times for methods in the time interval [0, 30]. The errors show that among the introduced methods, the TPS-DQ and GTPS techniques give somewhat accurate stable solutions of KGZ equations for large intervals, in comparison to other methods.

Another important point about the IMQ radial function is determination of the shape parameter. This parameter controls

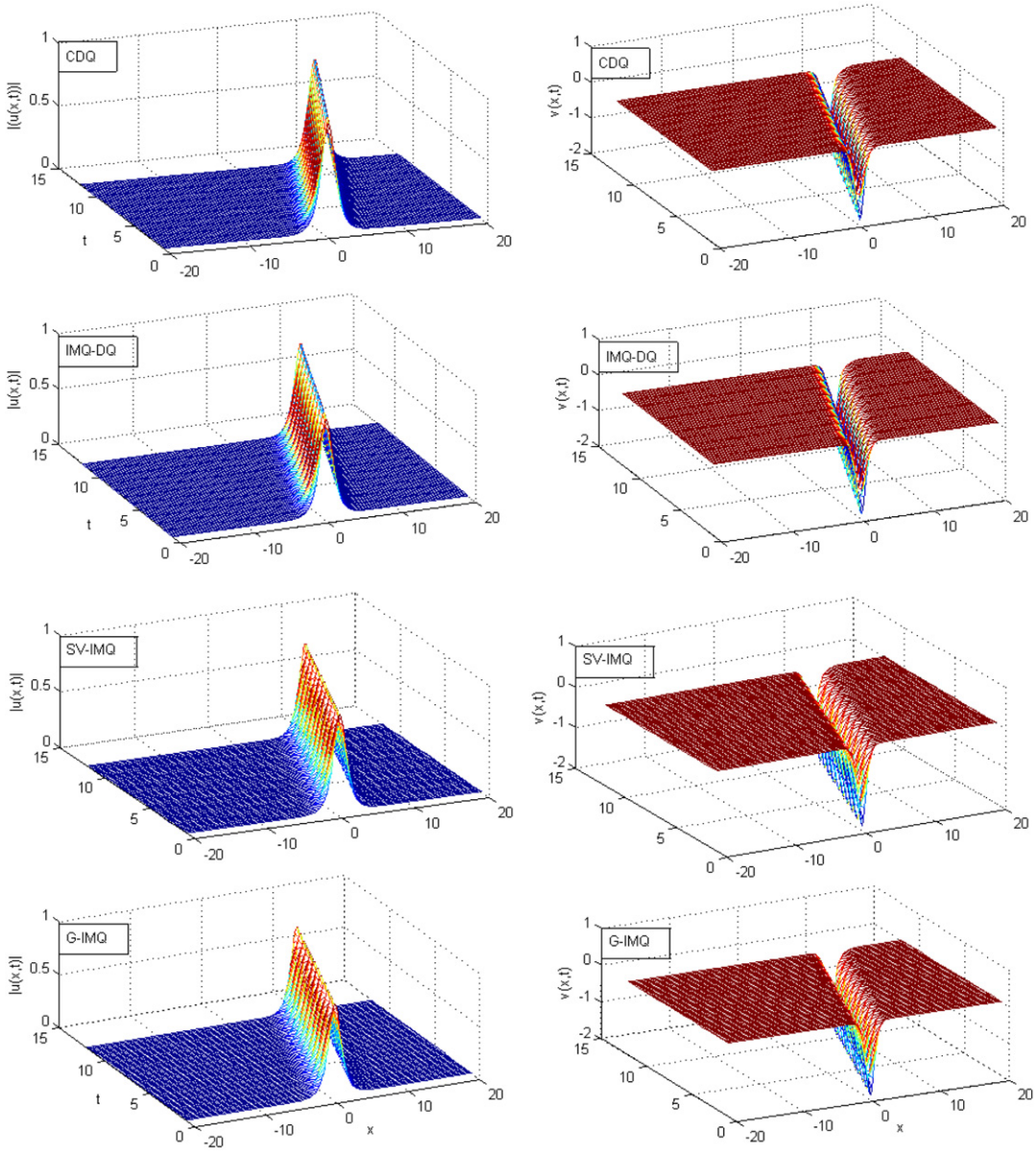


**Fig. 6.** Space–time graph of numerical solution of single soliton up to $t = 12$.

**Table 2**
$L_\infty \times 10^3$ error of different methods in interval $[-20, 20]$ and $T = 1$ with $\delta t = 0.01/0.001/0.0005$ and $h = 0.2$.

| Methods | $\|Err\{u\}\|_\infty$ | | | $\|Err\{v\}\|_\infty$ | | |
|---|---|---|---|---|---|---|
| | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ |
| CDQ | 7.24 | 0.73 | 0.37 | 5.23 | 0.53 | 0.26 |
| IMQ-DQ | 7.24 | 0.73 | 0.37 | 5.29 | 0.61 | 0.34 |
| TPS-DQ | 7.17 | 1.05 | 0.79 | 7.08 | 2.99 | 2.87 |
| G-IMQ | 5.69 | 0.59 | 0.30 | 5.15 | 0.53 | 0.29 |
| G-TPS | 5.69 | 0.97 | 0.77 | 5.99 | 2.96 | 2.90 |
| SV-IMQ | 7.23 | 0.73 | 0.40 | 5.29 | 0.61 | 0.34 |
| CH-IMQ | 7.23 | 0.73 | 0.40 | 5.29 | 0.61 | 0.34 |

**Table 3**
$L_\infty \times 10^2$ error of different methods in interval $[-20, 20]$ and $T = 5$ with $\delta t = 0.01/0.001/0.0005$ and $h = 0.2$.

| Methods | $\|Err\{u\}\|_\infty$ | | | $\|Err\{v\}\|_\infty$ | | |
|---|---|---|---|---|---|---|
| | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ |
| CDQ | 3.99 | 0.41 | 0.20 | 4.92 | 0.52 | 0.26 |
| IMQ-DQ | 3.99 | 0.41 | 0.20 | 4.94 | 0.54 | 0.28 |
| TPS-DQ | 4.07 | 0.58 | 0.41 | 4.99 | 1.44 | 1.35 |
| G-IMQ | 3.92 | 0.40 | 0.20 | 4.83 | 0.53 | 0.28 |
| G-TPS | 4.01 | 0.57 | 0.41 | 5.03 | 1.48 | 1.34 |
| SV-IMQ | 3.99 | 0.41 | 0.20 | 4.94 | 0.54 | 0.28 |
| CH-IMQ | 3.99 | 0.41 | 0.20 | 4.94 | 0.54 | 0.28 |

the shape of the radial function. In the above results, we use the constant shape parameter $c = 0.5$ for the IMQ function. Another technique for the choice of the shape parameter is the use of a variable shape parameter, i.e. applying a different shape value at each center. Between possible variable shape parameter strategies we use the exponentially varying shape defined as [38]

$$c_j = \left[ c_{\min}^2 \left( \frac{c_{\max}^2}{c_{\min}^2} \right)^{\frac{j-1}{N-1}} \right]^{1/2}, \quad j = 1, \ldots, N,$$

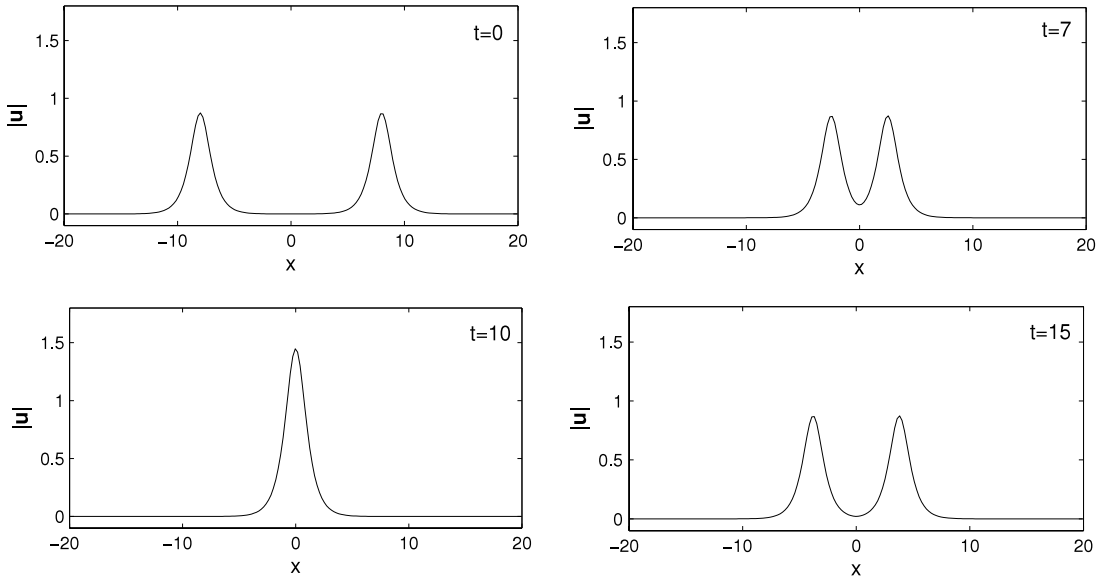


**Fig. 7.** Interaction of two symmetric solitons at different times with $x_1 = -8$ and $x_2 = 8$.

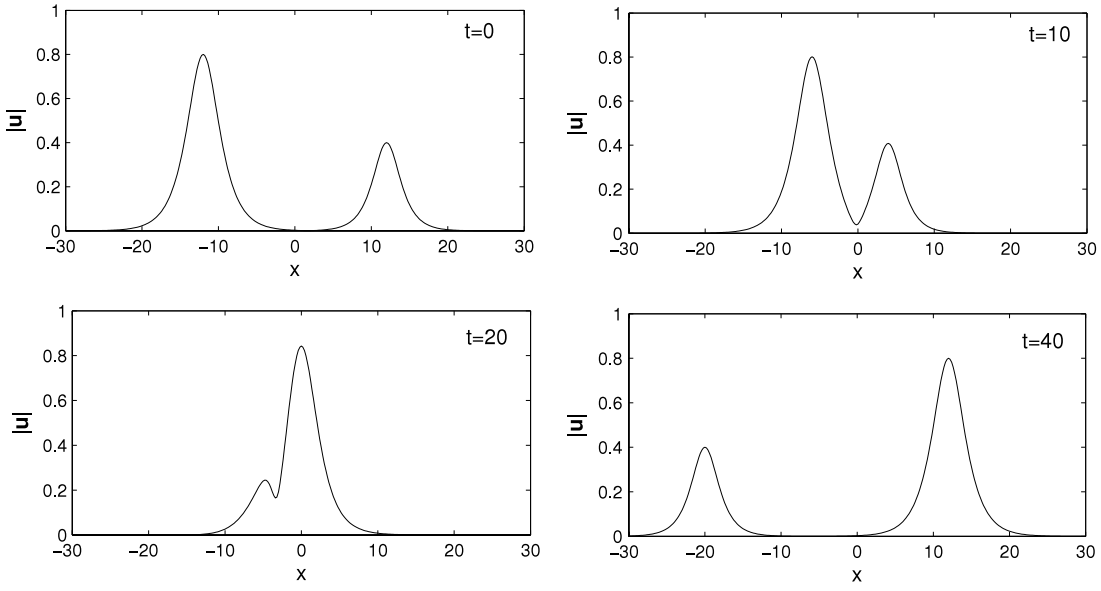


**Fig. 8.** Interaction of two non-symmetric solitons at different times with $x_1 = -12$, $x_2 = 12$, $\mu_1 = 0.8$, $\mu_2 = 0.4$, $\vartheta_1 = 0.6$ and $\vartheta_2 = -0.8$.

**Table 4**
$L_\infty \times 10^1$ error of different methods in interval $[-20, 20]$ and $T = 10$ with $\delta t = 0.01/0.001/0.0005$ and $h = 0.2$.

| Methods | $\|Err\{u\}\|_\infty$ | | | $\|Err\{v\}\|_\infty$ | | |
|---|---|---|---|---|---|---|
| | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ | $\delta t = 0.01$ | $\delta t/10$ | $\delta t/20$ |
| CDQ | 1.10 | 0.12 | 0.06 | 1.16 | 0.12 | 0.06 |
| IMQ-DQ | 1.10 | 0.12 | 0.06 | 1.15 | 0.12 | 0.06 |
| TPS-DQ | 1.15 | 0.19 | 0.14 | 1.09 | 0.22 | 0.17 |
| G-IMQ | 1.10 | 0.12 | 0.06 | 1.14 | 0.12 | 0.06 |
| G-TPS | 1.15 | 0.19 | 0.14 | 1.09 | 0.22 | 0.17 |
| SV-IMQ | 1.10 | 0.12 | 0.06 | 1.15 | 0.12 | 0.06 |
| CH-IMQ | 1.10 | 0.12 | 0.06 | 1.15 | 0.12 | 0.06 |

**Table 5**
Conservative law of different methods in interval $[-20, 20]$ and different times with $\delta t = 0.001$ and $h = 0.2$.

| Methods | $T = 1$ | $T = 5$ | $T = 10$ |
|---|---|---|---|
| Exact | 6.2892 | 6.2892 | 6.2892 |
| CDQ | 6.2904 | 6.2727 | 6.2506 |
| IMQ-DQ | 6.2901 | 6.2711 | 6.2476 |
| TPS-DQ | 6.2884 | 6.2635 | 6.2337 |
| G-IMQ | 6.2903 | 6.2713 | 6.2478 |
| G-TPS | 6.2883 | 6.2628 | 6.2325 |
| SV-IMQ | 6.2903 | 6.2713 | 6.2478 |
| CH-IMQ | 6.2903 | 6.2713 | 6.2478 |

and the random shape strategy [59]

$$\mathbf{c} = c_{\min} + (c_{\max} - c_{\min}) \times \mathrm{rand}(1, N),$$

for the IMQ function and compare their results with the constant shape parameter $c = 0.5$. Fig. 5 displays the absolute value of the point-wise error of different methods for the constant shape parameter of 0.5 (solid line), the exponential shape parameter (dot line) and the random shape parameter (bold solid line). Unlike the variable exponentially and random shape parameters, using the constant shape parameter achieves better results. Now, for comparison of methods introduced in the current paper with the method presented in [13], we consider $\mu = 0.5$ and $\vartheta = 0.3$ at the

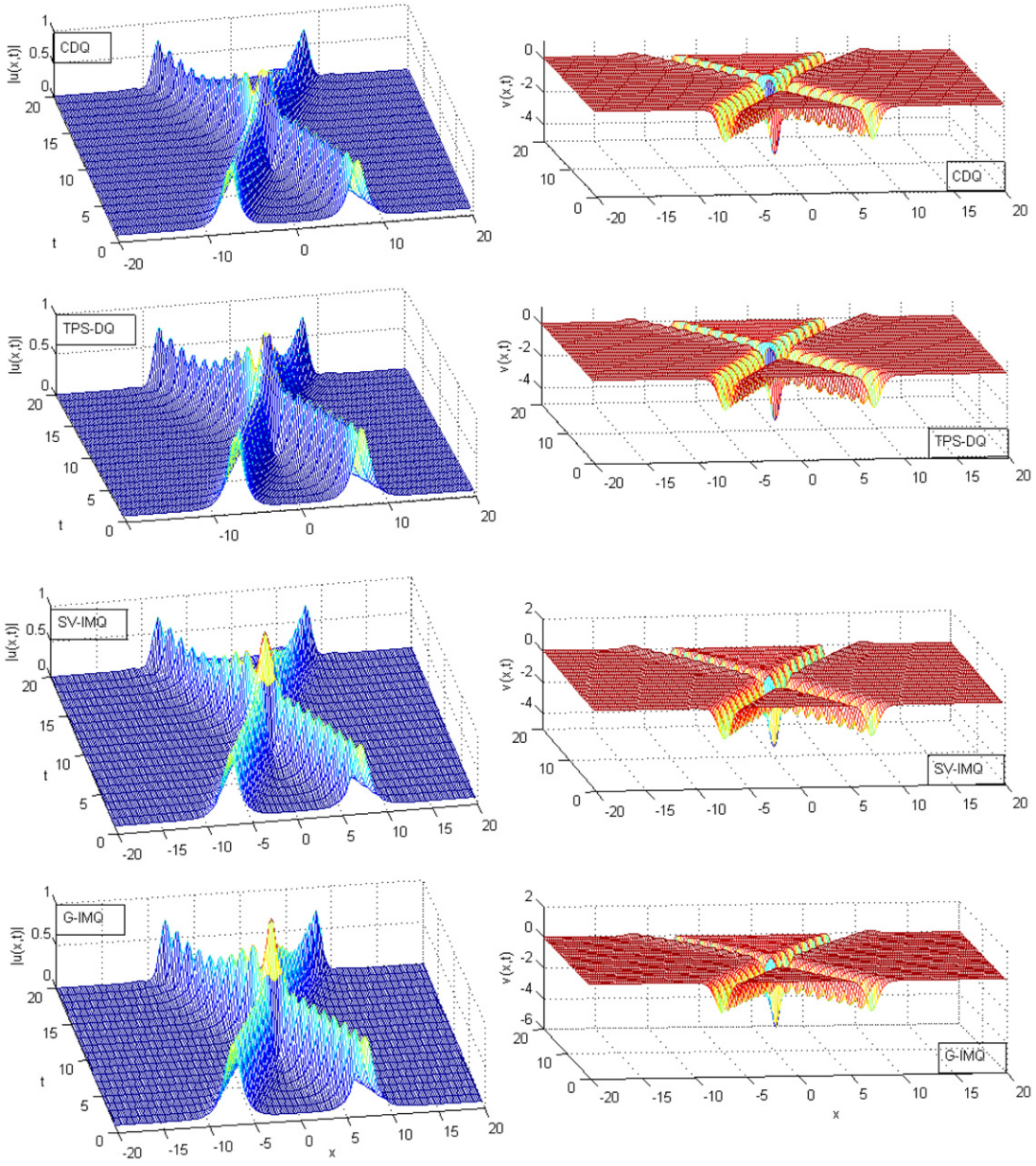


**Fig. 9.** Surface plot of interaction of two symmetric solitons at $t = 10$.

**Table 6**
$L_\infty$ error of different methods for function $u(x, t)$ in interval $[-40, 40]$ with $\delta t = 0.01$ and $h = 0.1$.

| Method | $T = 5$ | $2T$ | $3T$ | $4T$ |
|---|---|---|---|---|
| CDQ | 0.00798 | 0.01953 | 0.03529 | 0.05580 |
| IMQ-DQ | 0.02532 | 0.03053 | 0.03676 | 0.05603 |
| G-IMQ | 0.00794 | 0.01935 | 0.03531 | 0.05574 |
| SV-IMQ | 0.00798 | 0.01953 | 0.03529 | 0.05580 |
| CH-IMQ | 0.00798 | 0.01953 | 0.03529 | 0.05580 |
| [13] | 0.00953 | 0.01091 | 0.00831 | 0.01693 |

**Table 7**
Condition number of interpolation matrix for different methods in interval $[-20, 20]$ with $h = 0.2$.

| Method | Condition number |
|---|---|
| IMQ-DQ | 1.2145E+004 |
| TPS-DQ | 2.4412E+011 |
| G-IMQ | 1.2145E+004 |
| G-TPS | 6.0339E+014 |
| SV-IMQ | 1.1020E+002 |
| CH-IMQ | 1.1020E+002 |

interval $x \in [-40, 40]$. The comparison of the max-error between these methods is shown in Table 6. Due to severe ill-conditioning of the interpolation matrix, TPS-DQ and GTPS methods are instable.

### 5.2. The interaction of two solitary waves

Now, we turn to the interaction of two solitons with equal amplitudes and opposite velocities. The corresponding initial values are given as [60]

$$\begin{cases} u_0(x) = u(x - x_1, 0, \mu_1, \vartheta_1) + u(x - x_2, 0, \mu_2, \vartheta_2), \\ v_0(x) = v(x - x_1, 0, \xi_1, \vartheta_1) + v(x - x_2, 0, \xi_2, \vartheta_2), \\ u_1(x) = \{u_t(x - x_1, t, \mu_1, \vartheta_1) + u_t(x - x_2, t, \mu_2, \vartheta_2)\}|_{t=0}, \\ v_1(x) = \{v_t(x - x_1, t, \xi_1, \vartheta_1) + v_t(x - x_2, t, \xi_2, \vartheta_2)\}|_{t=0}. \end{cases}$$

Fig. 7 shows the interaction process of two symmetric solitons at different times with $x_1 = -8, x_2 = 8, \mu_1 = \mu_2 = \frac{\sqrt{10}-\sqrt{2}}{2}$

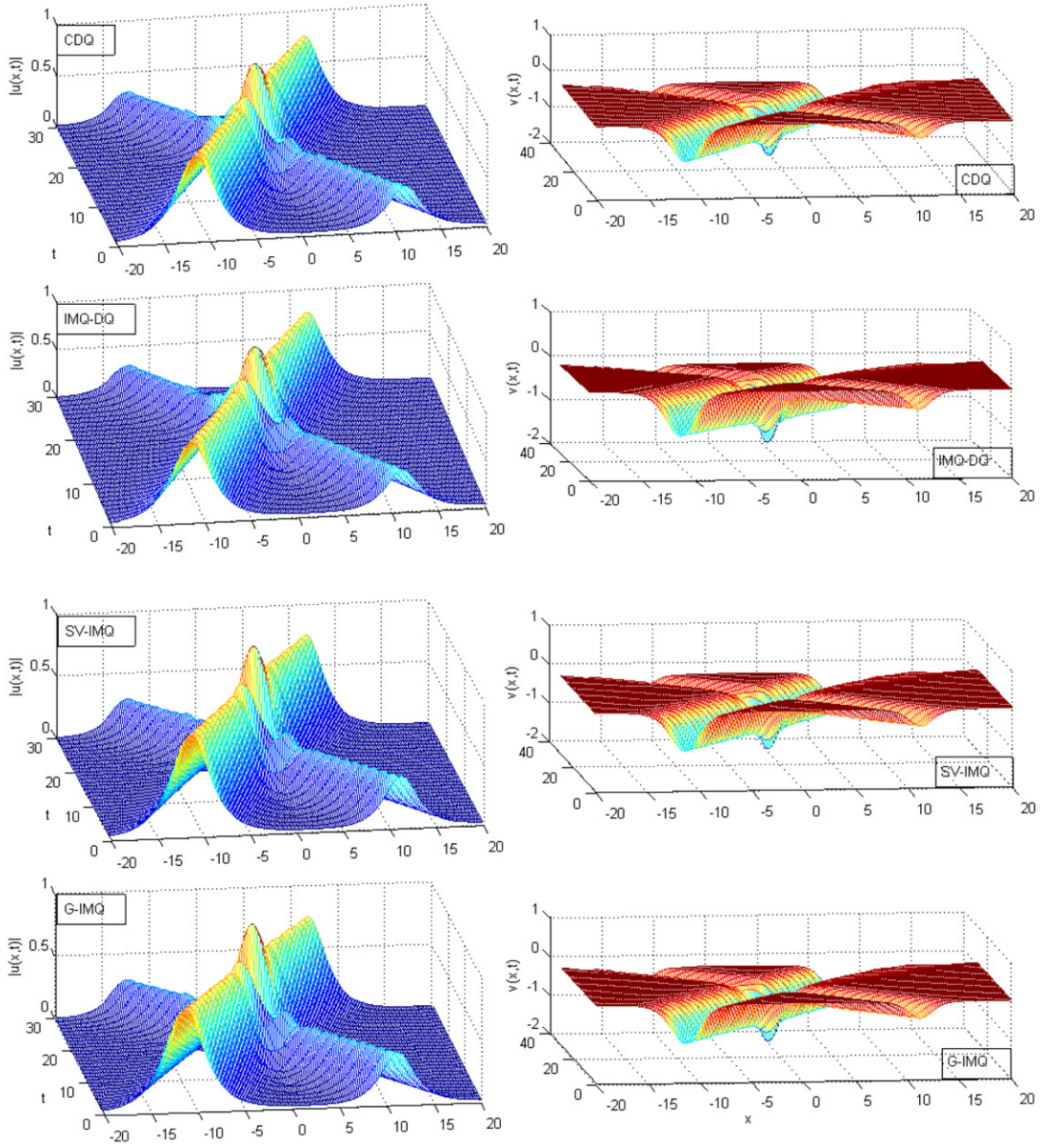


**Fig. 10.** Surface plot of interaction of two non-symmetric solitons up to $t = 30$.

**Table 8**
Condition number of the produced matrices by collocation at centers in Section 3 in interval $[-20, 20]$ with $h = 0.2$.

| Method | Condition number |
|---|---|
| CDQ | 1.0001E+000 |
| IMQ-DQ | 1.0001E+000 |
| TPS-DQ | 1.0001E+000 |
| G-IMQ | 1.2132E+004 |
| G-TPS | 1.1833E+015 |
| SV-IMQ | 1.1009E+002 |
| CH-IMQ | 1.1009E+002 |

and $\vartheta_1 = -\vartheta_2 = \sqrt{\frac{\sqrt{5}-1}{2}}$. The solitons move in the opposite directions and collide at $t = 10$. Fig. 9 illustrates this collision by the presented methods. It is clear that the solitons preserve their character after interaction. In Table 9, the results are shown in the time interval [0, 10] with $\delta t = 0.001$ and $\delta t = 0.0005$.

The last two columns show the CPU time of all methods. In the CDQ method the weight coefficients can be explicitly calculated, thus the CDQ method is faster than the RBFDQ method. The linear algebraic system arising in the GRBFs method is ill-conditioned. Therefore we use the LU decomposition procedure with pivoting for solving this system. Hence the GIMQ method is much slower than the SV-IMQ and CH-IMQ techniques.

Also the exact shape of two nonsymmetric solitons with parameters $\mu_1 = 0.8$, $\mu_2 = 0.4$, $\vartheta_1 = 0.6$ and $\vartheta_2 = -0.8$ is illustrated at the times $t = 0$, 10, 20 and 40 in Fig. 8. In this figure we study the behavior of function $u(x, t)$. It is observed that the larger soliton wave is positioned to the left of the smaller soliton wave and peak positions of them are located at $x = -12$ and 12. The interaction of waves takes place at about time $t = 10$ and passes completely from each other at about time $t = 30$. After time $t = 30$, both waves began to take their original shapes. The peak points of the larger soliton wave and smaller wave at time $t = 30$ are located at $x = 6.2$ and $x = -11.8$, respectively. The interaction between nonsymmetric solitons is numerically shown in Fig. 10. Calculations are carried out with given parameters, time steps $\delta t = 0.001, 0.0005$, mesh size $h = 0.2$ over the solution

**Table 9**
$L_\infty \times 10^2$ error of different methods in interval $[-20, 20]$ and $T = 1, 5, 10$ with $\delta t = 0.001/0.0005$, $h = 0.2, x_1 = -8$ and $x_2 = 8$.

| Methods | $\|Err\{u\}\|_\infty$ | | $\|Err\{v\}\|_\infty$ | | Time (s) | |
|---|---|---|---|---|---|---|
| | $\delta t = 0.001$ | $\delta t/2$ | $\delta t = 0.001$ | $\delta t/2$ | $\delta t = 0.001$ | $\delta t/2$ |
| $T = 1$ | | | | | | |
| CDQ | 0.07 | 0.04 | 0.05 | 0.03 | 1.9 | 3.6 |
| IMQ-DQ | 0.07 | 0.04 | 0.06 | 0.03 | 3.0 | 4.0 |
| TPS-DQ | 0.11 | 0.08 | 0.30 | 0.29 | 3.0 | 4.0 |
| G-IMQ | 0.06 | 0.03 | 0.05 | 0.03 | 12.8 | 21.4 |
| G-TPS | 0.10 | 0.08 | 0.30 | 0.30 | 12.8 | 21.4 |
| SV-IMQ | 0.07 | 0.04 | 0.06 | 0.03 | 1.1 | 1.7 |
| CH-IMQ | 0.07 | 0.04 | 0.06 | 0.03 | 1.1 | 1.7 |
| $T = 5$ | | | | | | |
| CDQ | 0.41 | 0.21 | 0.51 | 0.26 | 5.9 | 11.5 |
| IMQ-DQ | 0.41 | 0.20 | 0.54 | 0.28 | 7.0 | 14.1 |
| TPS-DQ | 0.58 | 0.41 | 1.46 | 1.33 | 7.0 | 14.1 |
| G-IMQ | 0.40 | 0.20 | 0.53 | 0.29 | 67.0 | 113.9 |
| G-TPS | 0.57 | 0.41 | 1.48 | 1.35 | 67.0 | 113.9 |
| SV-IMQ | 0.41 | 0.20 | 0.54 | 0.28 | 5.4 | 7.1 |
| CH-IMQ | 0.41 | 0.20 | 0.54 | 0.28 | 5.4 | 7.1 |
| $T = 10$ | | | | | | |
| CDQ | 2.51 | 2.65 | 52.64 | 52.62 | 11.7 | 23.2 |
| IMQ-DQ | 2.53 | 2.66 | 52.84 | 52.83 | 14.2 | 27.8 |
| TPS-DQ | 2.95 | 3.02 | 53.49 | 53.47 | 14.2 | 27.8 |
| G-IMQ | 2.53 | 2.66 | 52.83 | 52.82 | 135.9 | 167.6 |
| G-TPS | 2.95 | 3.03 | 53.53 | 53.53 | 135.9 | 167.6 |
| SV-IMQ | 2.53 | 2.66 | 52.84 | 52.84 | 8.9 | 15.7 |
| CH-IMQ | 2.53 | 2.66 | 52.84 | 52.84 | 8.9 | 15.7 |

**Table 10**
$L_\infty \times 10^2$ error of different methods in interval $[-30, 30]$ and $T = 10, 20$ with $\delta t = 0.001/0.0005$, $h = 0.2, x_1 = -12$ and $x_2 = 12$.

| Methods | $\|Err\{u\}\|_\infty$ | | $\|Err\{v\}\|_\infty$ | |
|---|---|---|---|---|
| | $\delta t = 0.001$ | $\delta t/2$ | $\delta t = 0.001$ | $\delta t/2$ |
| $T = 10$ | | | | |
| CDQ | 0.79 | 0.40 | 0.50 | 0.25 |
| IMQ-DQ | 0.87 | 0.49 | 0.55 | 0.31 |
| TPS-DQ | 4.23 | 4.29 | 16.64 | 17.43 |
| G-IMQ | 0.87 | 0.49 | 0.55 | 0.31 |
| G-TPS | 0.89 | 0.51 | 0.55 | 0.32 |
| SV-IMQ | 0.87 | 0.49 | 0.55 | 0.31 |
| CH-IMQ | 0.87 | 0.49 | 0.55 | 0.31 |
| $T = 20$ | | | | |
| CDQ | 11.98 | 10.83 | 10.29 | 9.61 |
| IMQ-DQ | 12.72 | 11.19 | 9.98 | 9.31 |
| TPS-DQ | 2079.56 | 2919.63 | 42 794.80 | 84 455.57 |
| G-IMQ | 12.72 | 11.19 | 9.98 | 9.31 |
| G-TPS | 13.04 | 11.52 | 9.69 | 9.02 |
| SV-IMQ | 12.72 | 11.19 | 9.98 | 9.31 |
| CH-IMQ | 12.72 | 11.19 | 9.98 | 9.31 |

domain $-30 \leq x \leq 30$ and $L_\infty$ errors are presented in Table 10. As is observed, among the methods, the TPS-DQ method has lower accuracy in comparison with other methods.

We refer the interested reader to [16] for an overview of the theory of solitons and the applications [15,61].

## 6. Conclusion

The DQ method with cosine expansion basis (CDQ) and radial basis functions (RBF-DQ) together with the globally radial basis functions (GRBFs) method are used to find the numerical solutions of the coupled Klein–Gordon–Zakharov (KGZ) equations. In the DQ method, the derivative value of the function with respect to a point is directly approximated by a linear combination of all functional values in the global domain while the GRBFs method directly substitutes the expression of the function approximation by RBFs into the partial differential equation. In the numerical examples we used the Inverse Multiquadric (IMQ) and the second-order TPS radial basis functions. The results show that the accuracy of methods is almost the same, but because of the direct approximation of derivatives, implementation of the DQ method is easier than the GRBFs method. The comparison of the CPU time of methods shows the GRBFs method is much slower with respect to other methods. Also, unlike the GRBFs method, the produced matrix by collocation technique in the DQ methods (CDQ and RBF-DQ) is well-conditioned. We used the bases introduced in [44] (SV and Cholesky decomposition cases) for avoiding the ill-conditioning problem of the interpolation matrix in the GRBFs method. The numerical results show the condition number of the produced interpolation matrices by SV-IMQ and CH-IMQ functions is half of the condition number of the produced interpolation matrix by the IMQ function. Also the numerical results show that in comparison with the variable shape parameter (exponentially and random), using the constant shape parameter for the IMQ function achieves better results.

## Acknowledgments

The authors thank the reviewers for their valuable suggestions.

## References

[1] R.O. Dendy, Plasma Dynamics, Oxford University Press, Oxford, 1990.
[2] B. Texier, WKB asymptotics for the Euler–Maxwell equations, Asymptot. Anal. 42 (2005) 211–250.
[3] T. Wang, J. Chen, L. Zhang, Conservative difference methods for the Klein–Gordon–Zakharov equations, J. Comput. Appl. Math. 205 (2007) 430–452.
[4] G. Boling, Y. Guangwei, Global smooth solution for the Klein–Gordon–Zakharov equations, J. Math. Phys. 36 (1995) 4119–4124.
[5] C. Lin, Orbital stability of solitary waves for the Klein–Gordon–Zakharov equations, Acta Math. Appl. Sin. 15 (1999) 54–64.
[6] K. Tsutaya, Global existence of small amplitude solutions for the Klein–Gordon–Zakharov equations, Nonlinear Anal. 27 (12) (1996) 1373–1380.
[7] J. Zhang, Z. Gan, Sharp conditions of global existence for Klein–Gordon–Zakharov equations in thre space dimensions, Adv. Math. 34 (2) (2005) 241–244.
[8] J. Zhang, Z. Gan, B. Guo, Stability of the standing waves for a class of coupled nonlinear Klein–Gordon equations, Acta Math. Appl. Sin. 26 (2010) 427–442.

[9] M. Dehghan, Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices, Math. Comput. Simul. 71 (2006) 16–30.
[10] J. Chen, L. Zhang, Numerical simulation for the initial-boundary value problem of the Klein–Gordon–Zakharov equations, Acta Math. Appl. Sin. 28 (2012) 325–336.
[11] M. Ghoreishi, A.I.B.Md. Ismail, A. Rashidy, Numerical solution of Klein–Gordon–Zakharov equations using Chebyshev Cardinal functions, J. Comput. Anal. Appl. 14 (2012) 574–582.
[12] G. Ebadi, E.V. Krishnan, A. Biswas, Solitons and cnoidal waves of the Klein–Gordon–Zakharov equation in plasmas, Pramana—J. Phys. 79 (2012) 185–198.
[13] M.S. Ismail, A. Biswas, 1-Soliton solution of the Klein–Gordon–Zakharov equation with power law nonlinearity, Appl. Math. Comput. 217 (2010) 4186–4196.
[14] J. Li, Exact explicit travelling wave solutions for $(n+1)$-dimensional Klein–Gordon–Zakharov equations, Chaos Solitons Fractals 34 (2007) 867–871.
[15] A.M. Wazwaz, Partial Differential Equations and Solitary Waves Theory, Springer, New York, 2009.
[16] M.A. Helal, Soliton solution of some nonlinear partial differential equations and its application in fluid mechanics, Chaos Solitons Fractals 13 (2002) 1917–1929.
[17] D.J. Korteweg, G. de Vries, On the change of form of long waves advancing in a rectangular canal and on a new type of long stationary waves, Phil. Mag. 39 (5) (1985) 422–443.
[18] R.M. Sorensen, Basic Coastal Engineering, third ed., Springer Science, New York, 2006.
[19] L. Munteanu, S. Donescu, Introduction To Soliton Theory: Applications To Mechanic, Springer Science, 2005.
[20] R. Bellman, B.G. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear differential equations, J. Comput. Phys. 10 (1972) 40–52.
[21] M.R. Hashemi, F. Hatam, Unsteady seepage analysis using local radial basis function-based differential quadrature method, Appl. Math. Model. 35 (2011) 4934–4950.
[22] R. Jiwari, S. Pandit, R.C. Mittal, Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method, Comput. Phys. Comm. 183 (2012) 600–616.
[23] B. Pekmen, M. Tezer-Sezgin, Differential quadrature solution of nonlinear Klein–Gordon and sine-Gordon equations, Comput. Phys. Comm. 183 (2012) 1702–1713.
[24] C. Shu, Differential Quadrature and its Application in Engineering, Springer-Verlag, London, 2000.
[25] K.M. Liew, Y.Q. Huang, J.N. Reddy, Moving least squares differential quadrature method and its application to the analysis of shear deformable plates, Internat. J. Numer. Methods Engrg. 56 (2003) 2331–2351.
[26] K.M. Liew, Y.Q. Huang, J.N. Reddy, A hybrid moving least squares and differential quadrature meshfree method, Int. J. Comput. Eng. Sci. 3 (2002) 1–12.
[27] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, IMA J. Numer. Anal. 32 (2012) 983–1000.
[28] H. Zhong, Spline-based differential quadrature for fourth-order differential equations and its application to Kirchhoff plates, Appl. Math. Model. 28 (2004) 353–366.
[29] C. Shu, H. Ding, K.S. Yeo, Solution of partial differential equations by a global radial basis function-based differential quadrature method, Eng. Anal. Bound. Elem. 28 (2004) 1217–1226.
[30] M. Dehghan, A. Shokri, Numerical solution of the Klein–Gordon equation using radial basis functions, J. Comput. Appl. Math. 230 (2009) 400–410.
[31] M. Dehghan, A. Shokri, A numerical method for solution of the two-dimensional sine-Gordon equation using the radial basis functions, Math. Comput. Simul. 79 (2008) 700–715.
[32] J. Li, Y.C. Hon, C.S. Chen, Numerical comparisons of two meshless methods using radial basis functions, Eng. Anal. Bound. Elem. 26 (2002) 205–225.
[33] A. Mohebbi, M. Abbaszadeh, M. Dehghan, The use of a meshless technique based on collocation and radial basis functions for solving the time fractional nonlinear Schrödinger equation arising in quantum mechanics, Eng. Anal. Bound. Elem. 37 (2013) 475–485.
[34] S.A. Sarra, Adaptive radial basis function methods for time dependent partial differential equations, Appl. Numer. Math. 24 (2005) 79–94.
[35] A. Shokri, M. Dehghan, A Not-a-Knot meshless method using radial basis functions and predictor–corrector scheme to the numerical solution of improved Boussinesq equation, Comput. Phys. Comm. 181 (2010) 1990–2000.
[36] M. Tatari, M. Dehghan, On the solution of the non-local parabolic partial differential equations via radial basis functions, Appl. Math. Model. 33 (2009) 1729–1738.
[37] M. Tatari, M. Dehghan, A method for solving partial differential equations via radial basis functions: application to the heat equation, Eng. Anal. Bound. Elem. 34 (2010) 206–212.
[38] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics-I, Comput. Math. Appl. 19 (1990) 127–145.
[39] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics-II, Comput. Math. Appl. 19 (1990) 147–161.
[40] S. Marchi, R. Schaback, Stability of kernel-based interpolation, Adv. Comput. Math. 32 (2010) 155–161.
[41] B. Fornberg, C. Piret, A stable algorithm for at radial basis functions on a sphere, SIAM J. Sci. Comput. 30 (2007) 60–80.
[42] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, SIAM J. Sci. Comput. 33 (2011) 869–892.
[43] C. Piret, E. Hanert, A radial basis functions method for fractional diffusion equations, J. Comput. Phys. 238 (2013) 71–81.
[44] M. Pazouki, R. Schaback, Bases for kernel-based spaces, J. Comput. Appl. Math. 236 (2011) 575–588.
[45] M. Pazouki, R. Schaback, Bases for conditionally positive definite kernels, J. Comput. Appl. Math. 243 (2013) 152–163.
[46] I. Dag, A. Korkmaz, B. Saka, Cosine expansion-based differential quadrature algorithm for numerical solution of the RLW equation, Numer. Methods Partial Differential Equations 26 (2010) 544–560.
[47] A. Korkmaz, I. Dag, Crank–Nicolson-differential quadrature algorithms for the Kawahara equation, Chaos Solitons Fractals 42 (2009) 65–73.
[48] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2005.
[49] E.J. Kansa, 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.
[50] E.J. Kansa, J. Geiser, Numerical solution to time-dependent 4D inviscid Burgers' equations, Eng. Anal. Bound. Elem. 37 (2013) 637–645.
[51] J.C. Li, Y.C. Hon, Domain decomposition for radial basis meshless methods, Numer. Methods Partial Differential Equations 20 (2004) 450–462.
[52] E.E. Hart, S.J. Cox, K. Djidjeli, Compact RBF meshless methods for photonic crystal modelling, J. Comput. Phys. 230 (2011) 4910–4921.
[53] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1995) 389–396.
[54] E.F. Bollig, N. Flyer, G. Erlebacher, Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs, J. Comput. Phys. 231 (2012) 7133–7151.
[55] G. Yao, B. Sarler, C.S. Chen, A comparison of three explicit local meshless methods using radial basis functions, Eng. Anal. Bound. Elem. 35 (2011) 600–609.
[56] C. Piret, Analytical and numerical advances in radial basis functions, Ph.D. Thesis, University of Colorado, 2007.
[57] S.K. Liu, Z.T. Fu, The periodic solutions for a class of coupled nonlinear Klein–Gordon equations, Phys. Lett. A 323 (2004) 415–420.
[58] S.K. Liu, Z.T. Fu, Jacobi elliptic function expansion method and periodic wave solutions of nonlinear wave equations, Phys. Lett. A 289 (2001) 69–74.
[59] S.A. Sarra, D. Sturgill, A random variable shape parameter strategy for radial basis function approximation methods, Eng. Anal. Bound. Elem. 33 (2009) 1239–1245.
[60] M. Dehghan, A. Taleei, Numerical solution of the Yukawa-coupled Klein–Gordon–Schrödinger equations via a Chebyshev pseudospectral multidomain method, Appl. Math. Model. 36 (2012) 2340–2349.
[61] M. Dehghan, R. Salehi, A meshless based numerical technique for traveling solitary wave solution of Boussinesq equation, Appl. Math. Model. 36 (2012) 1939–1956.