A LOCAL RADIAL BASIS FUNCTIONS BASED APPROACH FOR SOLVING KURAMOTO-SIVASHINSKY (KS) EQUATIONS USING DIFFERENTIAL QUADRATURE

Aqib Zafar[1*], Aimel Zafar[2]

1. College of Mathematics and System Science, Xinjiang University, Urumqi, China.
2. Department of Statistics, University of Peshawar, Pakistan.

ABSTRACT

In this article, we have constructed a scheme to calculate the numerical solutions of 1-D nonlinear Kuramoto-Sivashinsky equations. The scheme is created on local radial basis functions (LRBFs) using the differential quadrature (DQ) method. The space derivative is approximated through the RBFs using a central scheme in the neighborhood of a collocation node. Neighborhoods have seven, nine, eleven and fifteen points used in this regard. The LRBFDQ method converts the PDEs to a system of ODEs and is then evaluated by the typical RK-4 method. Some examples are taken to clarify the accuracy and implementation of the planned scheme. The numerical solutions are compared with some existing results; for the accuracy, validity and stability of the proposed scheme, we use error norms L_2 , L_∞ and Global Relative Error (GRE).

Key words: Radius, quadrature, neighborhood, evaluated, accuracy.

1. Introduction

In mathematics, PDEs are differential equations containing indefinite several variable functions and their partial derivatives; PDEs are used to develop problems, including functions of multi-variables and which are used to generate a computer model. PDEs can also describe various phenomena such as sound, electrostatics, fluid dynamics, heat, diffusion, elasticity and quantum mechanics. Many problems in biological, chemical, mechanical, and electrical science are usually represented by PDEs. Since PDEs are used in every discipline of science as well as engineering. Various numerical methodologies have been proposed in the literature for solving PDEs. In 1990, Kansa [1] extended the global RBF technique for the solution of different types of PDEs. RBF based methods are indeed mesh-less methods as these circumvent mesh production, which is the main issue in mesh-centered approaches. Also, these are infinitely differentiable, flexible with respect to geometry, easily extendable to higher dimensional problems, highly accurate, and exponentially convergent. The RBF constructed techniques were further extended for approximate solution of various PDEs containing unsteady Naiver–Stokes equations [2], RLW equation [3], and reaction-diffusion brusselator system involving two-dimensional coupled equations [4]. Many nonlinear PDEs cannot solve analytically; to overcome this difficulty, mesh-free methods are used.

Mesh-less methods are newly approximated algorithms for the pretend of physical consequence. In recent years, the mesh-less method has widely applied to many solid mechanics, electrostatics and fluid dynamic problems. Some traditional algorithms include the Finite Difference Method (FDM) [5-6]. The Finite Element Method (FEM) [7-8], and the Finite Volume Method (FVM) [9]. These methodologies have gained popularity in many branches of engineering and sciences [10,11,12]. Due to their extensive use, examples of many types of mesh-less techniques can be seen in the literature [13-14]. One of the presently existing concepts includes a category of mesh-less methods that focus on using RBFs [15], which is especially helpful for solving PDEs.

The RBFs method was firstly presented by Hardy in 1971 [16]. He introduced the multiquadric type of RBF. Richard Franke 1982 introduced many methods such as Foley's method, global basis function method, finite element basis function method, etc. He evaluated methods based on different parameters such as accuracy, storage and time reserved through the methods. On this basis, he encounters that the RBF method is one of the best methods among these methods [17]. RBFs have been applied to explain the problems in engineering and other sciences. The multiquadric RBF approach was acquired by Roland Hardy [18] in 1971. $\psi(\mathrm{r}) = \sqrt{r^2 + c^2}, \mathrm{c} > 0$, where c is MQ shape parameter that impact form of the surface. The MQ is one of the best spread data interpolation methods based on their accuracy, stability, validity and implementation. A farther interpolation matrix designed for multiquadric is invertible. Kansa, in 1990, adapted multiquadric for approximate solutions of elliptic, parabolic and hyperbolic kind PDEs. RBF has been used to solve PDEs numerically. The precision of MQ is determined by selecting a well-distinct parameter c, called the shape parameter. Golberg [19] and Hon [16] applied the method of cross support for the shape parameter to acquire the best results.

Over the past two decades, RBFs have played an important role in meshless methods for solving PDEs. This one is significant to remark that modifying the interpolation matrix in the totally maintained RBFs collocation approach is necessary. Those methods that use RBFs at each interpolation node in the domain are known as the global RBFs interpolation method. Such methods lead to a full or dense interpolation matrix. The RBFs interpolation methods converge as the number of the nodes but cause ill-conditioning of the interpolation matrix and huge computations, which further lead to instability of the RBF interpolation method. This issue of ill-conditioning and instability of these methods can be avoided by using RBF interpolation in a sub-domain in the local support of a node. This technique is known as the local RBF interpolation method. In 2006, Vertnik and Sarler introduced the local RBF scheme for diffusion and convective- diffusive solid liquid phase change problems [29-30]. Recently, Zheng et al. [31] used this for calculating band formations of the two-dimensional dense fluid and solid liquid phonic crystals. Shen calculated the numerical solution of the Storn-Liouville problem with the LRBFDQ collocation method [35]. In 2010, Shu worked on the upwind LRBFDQ method to simulate inviscid compressible flows [36]. The nodes in a sub-domain in local support nodes are manipulated to construct the Radial Basis Functions.

2. Kuramoto –Sivashinsky (KS) Equations

The KS equation was firstly introduced by Yoshiki-Kuramoto and Gregory-Sivashinsky in 1976 [16]. Nonlinear KS equations were derived in the condition of plasma unreliability, the appearance of agitation in the reaction-diffusion system and flame propagation [17]. KS equations are also used in the acquisition of long waves on the overlap in the middle of two viscid liquid and are well known for their disorder manner; KS equations has a solution like traveling waves, which can move with no change of configuration in spatiality content [22-23].

The KS - equations are

$$u_t + uu_x + \alpha u_{xx} + \sigma u_{xxx} + \beta u_{xxxx} = 0, \quad (1.1)$$

$$u_t + uu_x + \alpha u_{xx} + \beta u_{xxxx} = 0. \quad (1.2)$$

In KS equations arbitrary constants are used such as α, β and σ. When α and β are positive, its linear expressions exhibit counterbalance among extended wave instability and small wave stability with the nonlinear expressions provide a pattern for energy conversion among wave modality. For $\sigma = 0$ Eq. (1.1) is called Kuramoto-Sivashinsky equations that is a recognized nonlinear evolution equation increasing in different diversity of physical context. For $\alpha = \beta = 1$ and $\sigma = 0$ shows simulations of pattern formation on unsteady flame propagationThe KS equation has immense in different natural processes such as reaction-diffusion systems and thin hydro dynamic films. KS equation has been solved by some methods such as discontinuous Galerkin Method [24], Septic B-spline Collocation technique [25], Cubic B-Spline method [26], Exponential cubic B-spline method [27] and Kudryashov Method [28].

The Kuramoto-Sivashinsky equations are

$$u_t + uu_x + \alpha u_{xx} + \sigma u_{xxx} + \beta u_{xxxx} = 0 \tag{1}$$

$$u_t + uu_x + \alpha u_{xx} + \beta u_{xxxx} = 0 \tag{2}$$

Initial conditions are defined below

$$u(x, 0) = h(x), \quad a \leq x \leq b \tag{3}$$

And the boundary conditions are given by

$$u(a, t) = r_1(t)\,, \;\; u(b, t) = r_2(t), \qquad t > 0 \tag{4}$$

We choose N different knots $x_1, x_2, x_3, \ldots, x_N$. In LRBFDQ interpolation the values of derivatives of $u(x)$ is calculated by the method of DQ, that is the derivative of $u(x)$ at the centre x_i are approximate by the function values in the sub-domain in local support of x_i,$\{x_{i_1}, x_{i_2}, \ldots, x_{i_{ni}}\}$containing in$\{x_1, x_2, x_3, \ldots, x_N\}$, where n_i is a number of knots in the sub domain of ith knot and equated with N. The value of n_i is very small.

Let

$$u^{(m)}(x_i) \approx \sum_{j=1}^{n_i} \xi_{i_j}^{(m)}\, u\left(x_{i_j}\right), \quad i = 1,2,3,4, \ldots, N. \tag{5}$$

The weighted coefficients $\xi_j^{(m)}$ can be achieved by putting the RBF $\psi(\|x - x_k\|)$ in Eq. (2.5) as follows

$$\psi^{(m)}(\|x_i - x_k\|) = \sum_{j=1}^{n_i} \xi_{i_j}^{(m)} \psi\left(\left\|x_{i_j} - x_k\right\|\right), \quad k = i_1, i_2, i_3 \dots \dots i_{n_i}. \tag{6}$$

Eq. (2.6) can be justified as in matrix arrangements

$$\begin{bmatrix} \psi_{i_1}^{(m)}(x_i) \\ \psi_{i_2}^{(m)}(x_i) \\ \vdots \\ \psi_{i_{ni}}^{(m)}(x_i) \end{bmatrix} = \begin{bmatrix} \psi_{i_1}(x_{i_1}) & \psi_{i_2}(x_{i_1}) & \cdots & \psi_{i_{ni}}(x_{i_1}) \\ \psi_{i_1}(x_{i_2}) & \psi_{i_2}(x_{i_2}) & \cdots & \psi_{i_{ni}}(x_{i_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{i_1}(x_{i_{ni}}) & \psi_{i_2}(x_{i_{ni}}) & \cdots & \psi_{i_{ni}}(x_{i_{ni}}) \end{bmatrix} \begin{bmatrix} \xi_{i_1}^{(m)} \\ \xi_{i_2}^{(m)} \\ \vdots \\ \xi_{i_{ni}}^{(m)} \end{bmatrix}, \tag{7}$$

where

$$\psi_k(x_j) = \psi(\|x_j - x_k\|). \tag{8}$$

Therefore

$$\boldsymbol{\psi}_{n_i}^{(m)} = \mathbf{A}_{n_i} \boldsymbol{\xi}_{n_i}^{(m)} \tag{9}$$

$$\boldsymbol{\psi}_{n_i}^{(m)} = \left[\psi_{i_1}^{(m)}(x_i), \psi_{i_2}^{(m)}(x_i), \dots, \psi_{i_{n_i}}^{(m)}(x_i)\right]^T,$$

$$\boldsymbol{\xi}_{n_i}^{(m)} = [\xi_{i_1}^{(m)}, \xi_{i_2}^{(m)}, \dots, \xi_{i_{n_i}}^{(m)}]^T,$$

$$\mathbf{A}_{n_i} = \begin{bmatrix} \psi_{i_1}(x_{i_1}) & \psi_{i_2}(x_{i_1}) & \cdots & \psi_{i_{ni}}(x_{i_1}) \\ \psi_{i_1}(x_{i_2}) & \psi_{i_2}(x_{i_2}) & \cdots & \psi_{i_{ni}}(x_{i_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{i_1}(x_{i_{ni}}) & \psi_{i_2}(x_{i_{ni}}) & \cdots & \psi_{i_{ni}}(x_{i_{ni}}) \end{bmatrix}.$$

So the corresponding weighting coefficients can be written as,

$$\xi_{n_i}^{(m)} = A_{n_i}^{-1} \psi_{n_i}^{(m)}. \tag{10}$$

Now putting Eq. (2.9) in Eq. (2.5) hence the new form will be,

$$u^{(m)}(x_i) = (\xi_{n_i}^{(m)})^T \overline{\mathbf{u}}_{n_i}, \tag{11}$$

where $\overline{\mathbf{u}}_{n_i} = \left[u\left(x_{i_1}\right), u\left(x_{i_2}\right), u\left(x_{i_3}\right), \ldots, u\left(x_{i_{ni}}\right)\right]^T$.

By using LRBFDQ approach to Eq. (2.1) we develop the form,

$$\frac{du_i}{dt} + u(\xi_{n_i}^{(1)})^T\overline{\mathbf{u}}_{n_i} + \alpha(\xi_{n_i}^{(2)})^T\overline{\mathbf{u}}_{n_i} + \sigma(\xi_{n_i}^{(3)})^T\overline{\mathbf{u}}_{n_i} + \beta(\xi_{n_i}^{(4)})^T\overline{\mathbf{u}}_{n_i} = 0,$$

$$\frac{du_i}{dt} = -\left[u * \left(\xi_{n_i}^{(1)}\right)^T\overline{\mathbf{u}}_{n_i} + \alpha\left(\xi_{n_i}^{(2)}\right)^T\overline{\mathbf{u}}_{n_i} + \sigma\left(\xi_{n_i}^{(3)}\right)^T\overline{\mathbf{u}}_{n_i} + \beta\left(\xi_{n_i}^{(4)}\right)^T\overline{\mathbf{u}}_{n_i}\right]. \tag{12}$$

The matrix form of Eq. (2.12) is,

$$\frac{d\mathbf{u}}{dt} = -\left[u * \left(\boldsymbol{\xi}^{(1)}\mathbf{u}\right) + \alpha\left(\boldsymbol{\xi}^{(2)}\mathbf{u}\right) + \sigma\left(\boldsymbol{\xi}^{(3)}\mathbf{u}\right) + \beta\left(\boldsymbol{\xi}^{(4)}\mathbf{u}\right)\right]. \tag{13}$$

The sign * shows component wise multiplication of two vectors,

Let $\frac{d\mathbf{u}}{dt} = F(\mathbf{u})$,

where $F(\mathbf{u}) = -\left[u * \left(\boldsymbol{\xi}^{(1)}\mathbf{u}\right) + \alpha\left(\boldsymbol{\xi}^{(2)}\mathbf{u}\right) + \sigma\left(\boldsymbol{\xi}^{(3)}\mathbf{u}\right) + \beta\left(\boldsymbol{\xi}^{(4)}\mathbf{u}\right)\right]$

and $\boldsymbol{\xi}^{(1)} = [\xi_j^{(1)}]_{N\times N}$, $\boldsymbol{\xi}^{(2)} = [\xi_j^{(2)}]_{N\times N}$,

$\boldsymbol{\xi}^{(3)} = [\xi_j^{(3)}]_{N\times N}$, $\boldsymbol{\xi}^{(4)} = [\xi_j^{(4)}]_{N\times N}$.

Now RK-4 method is,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{1}{6}\delta t(k_1 + 2k_2 + 2k_3 + k_4)$$

In which $\mathbf{k}_1 = \delta t F(\mathbf{u})$,

$$\mathbf{k}_2 = \delta t F\left(\mathbf{u}^n + \frac{1}{2}\delta t\mathbf{k}_1\right),$$

$$\mathbf{k}_3 = \delta t F\left(\mathbf{u}^n + \frac{1}{2}\delta t\mathbf{k}_2\right),$$

$$\mathbf{k}_4 = \delta t F(\mathbf{u}^n + \delta t\mathbf{k}_3).$$

3. Numerical Results and Discussion

In this section, we implement the proposed scheme to determine the approximate solutions of the family of Kuramoto-Sivashinsky equations and equate the present results with existing results Method of Lines (MOL) [32], Mesh Free Colocation Method (MFCM) [34] and Lattice Boltzman Method (LBM) [33]. To elaborate, the accuracy and validity of the present method is organized by calculating L_2 , L_∞ and Global Relative error (GRE) that are given below.

$$L_2 = \|u_{ex} - u_{ap}\|_2 = \sqrt{h \sum_{j=1}^{N} (u_{ex} - u_{ap})^2}\ , \qquad (14)$$

$$L_\infty = \|u_{ex} - u_{ap}\|_\infty = \max_j |u_{ex} - u_{ap}|$$

$$L_\infty = \|u_{ex} - u_{ap}\|_\infty = \max_j |u_{ex} - u_{ap}| \quad , \qquad (15)$$

$$GRE = \frac{\sum_J |u_{ap} - u_{ex}|}{\sum_j |u_{ex}|}\ , \qquad (16)$$

where u_{ex} and u_{ap} denote the exact and approximate solutions correspondingly, and h is the spatial step size. Now we consider three examples of Kuramoto-Sivashinsky equations with exact solutions and investigate the numerical solutions.

Example.1 Consider the Kuramoto-Sivashinsky equation (1).

For the values $\alpha = 1,\ \sigma = 0,$ and $\beta = 1$

with initial conditions

$$u(x,0) = C + \frac{15}{19}\sqrt{\frac{11}{19}}\left[11 \tanh^3(k(x - x_0)) - 9 \tanh(k(x - x_0))\right]. \qquad (17)$$

Exact solution of (2.1) for above defined problem is given below

$$u(x,t) = C + \frac{15}{19}\sqrt{\frac{11}{19}}\left[11 \tanh^3(k(x - Ct - x_0)) - 9 \tanh(k(x - Ct - x_0))\right]. \qquad (18)$$

The other values of parameters are given as;

$\mathrm{C} = 0.1, \mathrm{K} = 0.5\sqrt{\frac{11}{19}}$, $\mathrm{x}_0 = -10, \mathrm{dt} = 0.001, \mathrm{h} = 0.5, [\mathrm{a}, \mathrm{b}] = [-30{,}30]$

The numerical calculations of Eq. (1) are calculated by using the proposed scheme (13) for different points that is N=9, 11, 15 at different time level t=1, 2, 3, 4, in Tables from 1 to 6 using shape parameter c=2.7, 0.009, 5.5, 0.1, 4.7, 0.19 for both RBFs MQ and GA. When the present method results of error norms, particularly GRE, are equated with MOL [32] and LBM [33], we conclude that the present results are better than the existing results of MOL and LBM. While, in Tables 7 to 12 using time level t=0.1, 0.3, 0.5, 0.7, 1.0 and the same points and same shape parameters for both MQ and GA, when present results of error norms that is max error and L_2 are compared with MFCM [34] the results are as good as and comparable.

Table.1 MQ c=2.7, N=9, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(MQ)[52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	2.4103e-04	3.6931e-04	3.0881e-05	2.0528e-04	6.7923e-04
2	3.6577e-04	7.3172e-04	5.2225e-05	3.7496e-04	1.1503e-03
3	4.5622e-04	1.1000e-03	6.9962e-05	5.6925e-04	1.5941e-03
4	5.0663e-04	1.4000e-03	8.5807e-05	8.0226e-04	2.0075e-03

Table.2 GA c=0.09 N=9, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(GA)[52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	1.1879e-04	5.8621e-04	2.0275e-05	7.4687e-03	6.7923e-04
2	1.0922e-04	1.2000e-03	3.3109e-05	1.2227e-02	1.1503e-03
3	1.3573e-04	1.7000e-03	4.5000e-05	1.7828e-02	1.5941e-03
4	1.4906e-04	2.3000e-03	5.5442e-05	2.4541e-02	2.0075e-03

Table.3 MQ c=5.5, N=11, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(MQ) [52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	2.4070e-04	1.0585e-04	3.0414e-05	2.0528e-04	6.7923e-04
2	3.4917e-04	2.1528e-04	5.4775e-05	3.7496e-04	1.1503e-03
3	4.0586e-04	3.1210e-04	8.3071e-05	5.6925e-04	1.5941e-03
4	4.3207e-04	3.3407e-04	1.0864e-04	8.0226e-04	2.0075e-03

Table.4 GA c=0.1 N=11, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(GA) [52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	1.9865e-05	1.4676e-04	2.9475e-06	7.4687e-03	6.7923e-03
2	2.2929e-05	2.8815e-04	4.7455e-06	1.2227e-02	1.1503e-03
3	2.5525e-05	4.2626e-04	6.3935e-06	1.7828e-02	1.5941e-03
4	3.4872e-05	5.6426e-04	7.8858e-06	2.4541e-02	2.0075e-03

Table.5 MQ c=4.7, N=15, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(MQ)[52]	LBM[53]
T	Max error	L_2	GRE	GRE	GRE
1	1.5963e-04	8.0719e-05	2.3865e-05	2.0528e-04	6.7923e-04
2	2.5857e-04	1.5649e-04	4.3301e-05	3.7496e-04	1.1503e-03
3	3.2328e-04	1.8561e-04	6.3275e-05	5.6925e-04	1.5941e-03
4	3.5537e-04	2.1289e-04	8.1939e-05	8.0226e-04	2.0075e-03

Table.6 GA c=0.19, N=15, $dt = 0.001$ and error norms for Example 1

	Present Method			MOL(GA) [52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	1.7149e-05	8.8866e-06	4.1084e-06	7.4687e-3	6.7923e-04
2	3.0647e-05	2.1986e-05	7.9768e-06	1.2227e-02	1.1503e-03
3	4.1266e-05	4.5065e-05	1.1576e-05	1.7828e-02	1.5941e-03
4	4.9290e-05	8.2398e-05	1.4747e-05	2.4541e-02	2.0075e-03

Table.7 MQ c=2.7, N=9, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	8.5378e-05	3.7523e-05	7.9512e-06	1.02912e-04	1.72557e-05
0.3	1.2547e-04	1.1204e-04	1.3651e-05	1.85165e-04	3.85943e-04
0.5	1.6319e-04	1.8609e-04	1.8069e-05	2.89155e-04	7.29738e-04
0.7	1.9568e-04	2.5937e-04	2.3220e-05	3.85161e-04	7.13340e-04
1.0	2.4103e-04	3.6931e-04	3.0881e-05	5.22599e-04	1.51203e-03

Table.8 GA c=0.09 N=9, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	1.1503e-04	5.9203e-05	7.2933e-06	4.00703e-02	7.09072e-03
0.3	1.1611e-04	1.7716e-04	1.0608e-05	8.14516e-02	1.84731e-02
0.5	1.1731e-04	2.9461e-04	1.3500e-05	1.09220e-01	2.64874e-02
0.7	1.1857e-04	4.1099e-04	1.6270e-05	1.37528e-01	3.23056e-02
1.0	1.1879e-04	5.8621e-04	2.0275e-05	1.75154e-01	3.65761e-02

Table.9 MQ c=5.5, N=11, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	5.1132e-05	1.0746e-05	4.6842e-06	1.02912e-04	1.72557e-05
0.3	1.0859e-04	3.2119e-05	1.1573e-05	1.85165e-04	3.85943e-04
0.5	1.5372e-04	5.3386e-05	1.7632e-05	2.89155e-04	7.29738e-04
0.7	1.9201e-04	7.4418e-05	2.3011e-05	3.85161-e04	7.13340e-04
1.0	2.4070e-04	1.0585e-04	3.0414e-05	5.22599e-04	1.51203e-03

Table.10 GA c=0.1 N=11, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	2.1084e-05	1.4961e-05	1.1918e-06	4.00703e-02	7.09072e-03
0.3	2.1504e-05	4.4602e-05	1.6191e-06	8.14516e-02	1.84731e-02
0.5	2.1397e-05	7.4016e-05	2.0043e-06	1.09220e-01	2.64874e-02
0.7	2.0948e-05	1.0310e-04	2.3881e-06	1.37528e-01	3.23056e-02
1.0	1.9865e-05	1.4676e-04	2.9475e-06	1.75154e-01	3.65761e-02

Table.11 MQ c=4.7, N=15, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	3.1402e-05	8.0883e-06	3.1257e-06	1.02912e-04	1.72557e-05
0.3	6.8120e-05	2.4229e-05	8.1688e-06	1.85165e-04	3.85943e-04
0.5	9.6979e-05	4.0271e-05	1.2846e-05	2.89155e-04	7.29738e-04

0.7	1.2289e-04	5.6170e-05	1.7406e-05	3.85161e-04	7.13340e-04
1.0	1.5963e-04	8.0719e-05	2.3865e-05	5.22599e-04	1.51203e-03

Table.12 GA c=0.19, N=15, $dt = 0.001$ and error norms for Example 1

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	2.3286e-06	5.0508e-07	4.5092e-07	4.00703e-02	7.09072e-03
0.3	6.0184e-06	1.9730e-06	1.2921e-06	8.14516e-02	1.84731e-02
0.5	9.3787e-06	3.7339e-06	2.1182e-06	1.09220e-01	2.64874e-02
0.7	1.2559e-05	5.6856e-06	2.9184e-06	1.37528e-01	3.23056e-02
1.0	1.7149e-05	8.8866e-06	4.1084e-06	1.75154e-01	3.65761e-02

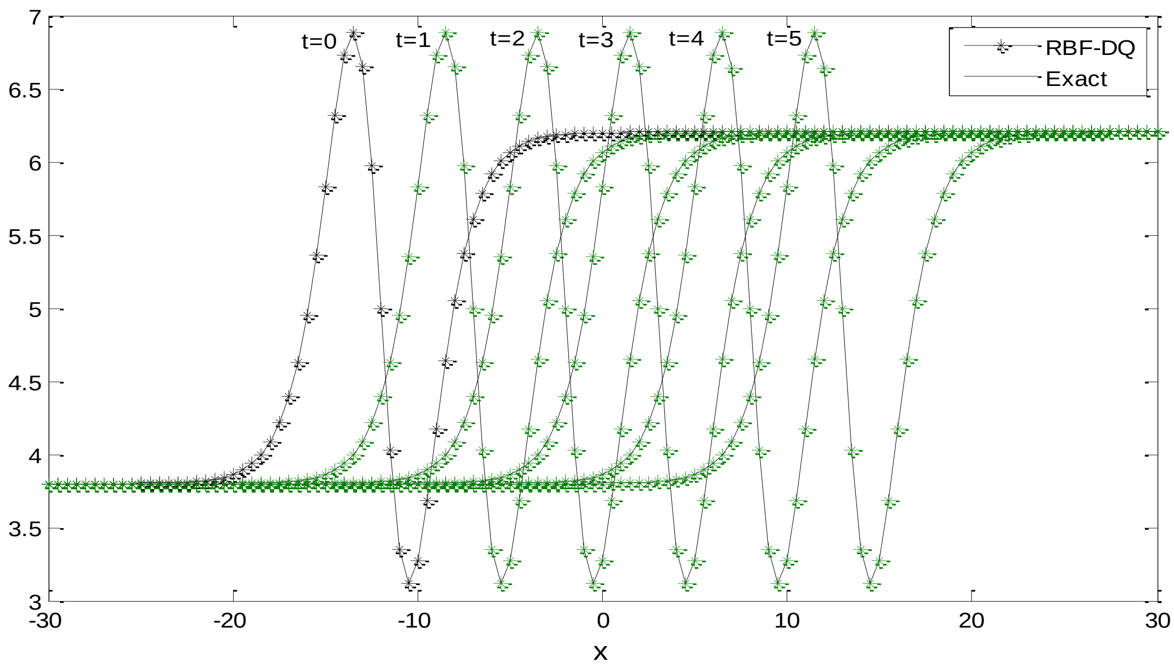


Fig.1 Shock wave propagation of KS- equations, in this figure solid line represents exact solutions while dotted lines exhibit numerical solutions

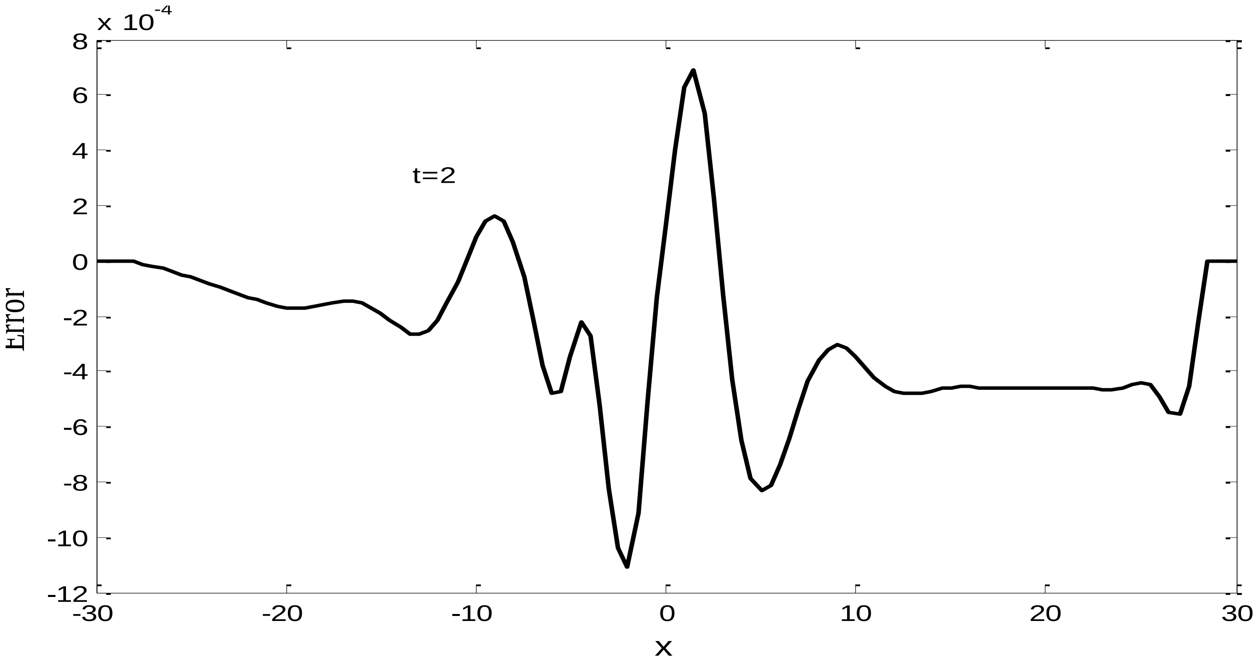


Fig.2 Error graph for example 1 for t=2

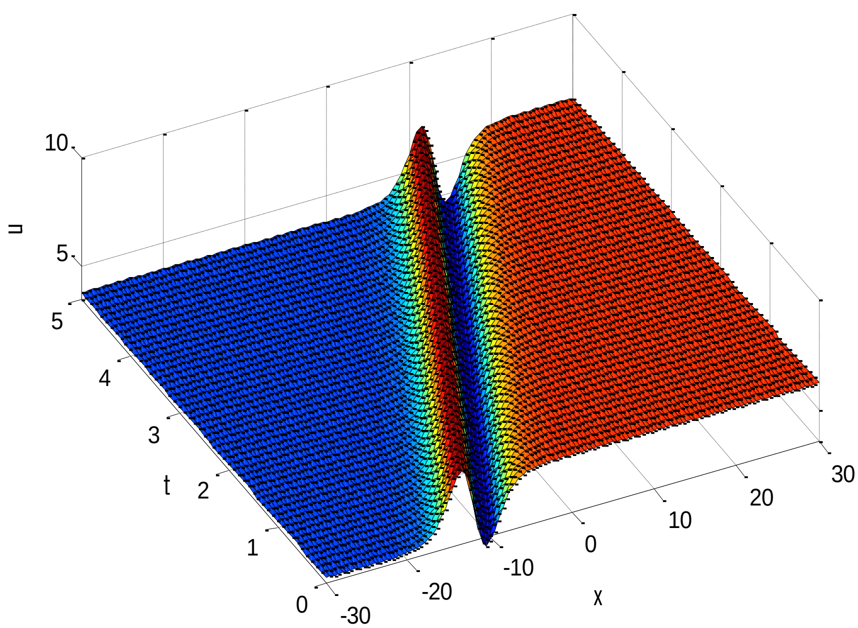


Fig.3 Solution for u over time interval [0, 5] and space interval [-30, 30]

Example.2 Let suppose again the Kuramoto- Sivashinsky equation (1).

Exact solution of above example with $\alpha = 1, \sigma = 4, \beta = 1$ is,

$$u(x,t) = 11 + 15\tanh\theta - 15\tan^3\theta - 15\tan^3\theta \quad \text{with } \theta = -0.5x + t \tag{19}$$

The other parameters are defined as follows;

$h = 0.1, dt = 0.001, [a, b] = [-10,10]$.

The numerical computations of equation (1) are achieved through planned scheme (13) for points that are N=7, 9 and time level t=1, 2, 3, 4 using shape parameters c= 6, 0.1, 0.065 for both RBFs MQ and GA, In Tables from 13 to 16 When present results of error norms GRE at this stage is compared with MOL [32] and LBM [33], present method results are very stable and better than the existing results of mentioned consequences. While taking time level t=0.1, 0.3, 0.5, 0.7, 1.0 same points and same shape parameters in Tables 17 to 20 the present results of error norms, that is max error and L_2 are comparable and stable with existing results of MFCM [34].

Table.13 MQ c=6, N=9, $dt = 0.001$ and error norms for example 2

	Present Method			MOL(GA) [52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	4.3000e-03	4.5290e-04	2.2788e-04	1.2184e-06	2.5945e-02
2	1.1100e-03	6.3000e-03	4.1329e-04	2.4441e-06	2.7959e-02
3	1.9900e-02	6.9029e-04	8.1103e-04	2.6723e-06	2.6701e-02
4	9.1000e-02	3.8500e-04	6.9123e-04	5.2730e-06	3.5172e-02

Table.14 GA c=0.1, N=9, $dt = 0.001$ and error norms for example 2

	Present Method			MOL(GA)[52]	LBM[53]
T	Max error	L_2	GRE	GRE	GRE
1	6.7000e-03	7.0000e-03	2.9000e-04	5.5060e-07	2.5945e-02
2	1.3200e-02	1.0000e-03	4.5519e-04	5.5059e-07	2.7959e-02
3	2.2200e-02	7.10000e-03	7.9458e-04	5.5060e-07	2.6701e-02
4	9.0000e-03	5.9400e-02	5.4791e-04	5.5058e-07	3.5172e-02

Table.15 MQ c=6, N=7, $dt = 0.001$ and error norms for example 2

	Present Method			MOL(MQ)[52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	4.4800e-02	6.1000e-03	2.0000e-03	1.2184e-06	2.5945e-02
2	9.5800e-02	3.1200e-02	3.6000e-03	2.4441e-06	2.7959e-02
3	1.3890e-01	1.0340e-01	6.6000e-03	2.6723e-06	2.6701e-02
4	8.2900e-02	4.8540e-01	3.9000e-03	5.2730e-06	3.5172e-02

Table.16 GA c=0.065, N=7, $dt = 0.001$ and error norms for example 2

	Present Method			MOL(GA)[52]	LBM [53]
T	Max error	L_2	GRE	GRE	GRE
1	5.999e-02	2.1903e-04	2.9000e-03	5.5060e-07	2.5945e-02
2	2.225e-01	7.2800e-02	8.2000e-03	5.5059e-07	2.7959e-02
3	4.377e-01	1.6000e-02	1.6800e-02	5.5060e-07	2.6701e-02
4	1.876e-01	1.2700e-02	1.0200e-02	5.5058e-07	3.5172e-02

Table.17 MQ c=6, N=9, $dt = 0.001$ and error norms for example 2

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	3.6000e-03	7.8083e-05	2.5243e-04	1.64470e-03	8.49417e-05
0.3	4.0000e-03	3.5000e-03	2.4237e-04	7.83221e-03	2.63959e-04
0.5	4.5000e-03	1.9000e-03	2.4751e-04	5.25919e-03	4.09262e-04
0.7	4.6000e-03	8.0397e-04	2.4001e-04	1.22430e-02	6.32613e-04
1.0	4.3000e-03	4.5290e-04	2.2788e-04	1.20928e-02	7.81989e-04

Table.18 GA c=0.1, N=9, $dt = 0.001$ and error norms for example 2

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	5.9000e-03	3.8951e-04	1.9594e-04	1.57721e-02	6.62754e-04
0.3	5.8000e-03	1.5000e-03	2.2748e-04	3.54322e-02	6.54807e-04

0.5	5.5000e-03	3.5000e-03	2.5525e-04	5.45042e-02	8.45945e-04
0.7	6.3000e-03	5.5000e-03	2.6046e-04	1.19586e-01	1.92659e-04
1.0	6.7000e-03	7.0000e-03	2.9000e-04	1.14401e-01	1.14018e-03

Table.19 MQ c=6, N=7, $dt = 0.001$ and error norms for example 2

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	2.2600e-02	8.7268e-04	1.3000e-03	1.64470e-03	8.49417e-05
0.3	3.2000e-02	1.5500e-02	1.7000e-03	7.83221e-03	2.63959e-04
0.5	3.5100e-02	2.8300e-02	1.9000e-03	5.25919e-03	4.09262e-04
0.7	3.9300e-02	1.7400e-02	1.9000e-03	1.22430e-02	6.32613e-04
1.0	4.4800e-02	6.1000e-03	2.0000e-03	120928e-02	7.81989e-04

Table.20 GA c=0.065, N=7, $dt = 0.001$ and error norms for example 2

	Present Method			MFCM[54]	MFCM[54]
T	Max error	L_2	GRE	Max error	L_2
0.1	4.2300e-03	1.1000e-03	1.7000e-03	1.57721e-002	6.62754e-004
0.3	4.5900e-03	1.4500e-02	1.9000e-03	3.54322e-02	6.54807e-04
0.5	5.1900e-03	9.7203e-04	2.1000e-03	5.45042e-02	8.45945e-04

0.7	5.1900e-03	8.4000e-03	2.3000e-03	1.19586e-01	1.92659e-04
1.0	5.9900e-03	2.1903e-04	2.9000e-03	1.14401e-01	1.14018e-03

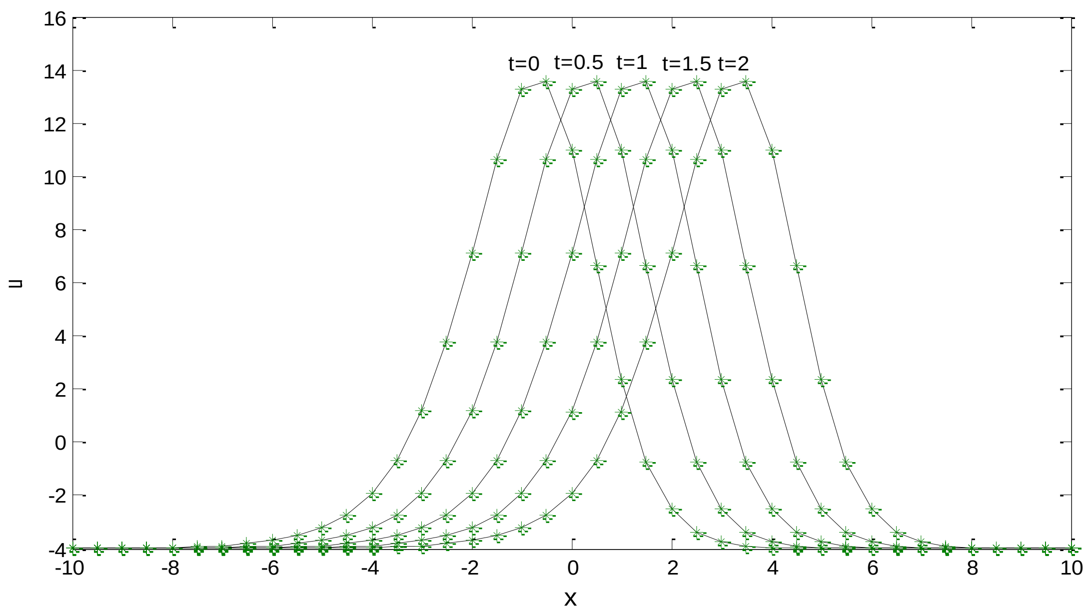


Fig.4 This figure shows solitary wave propagation. Solid lines represent exact solutions while dotted lines showing numerical solutions for example 1

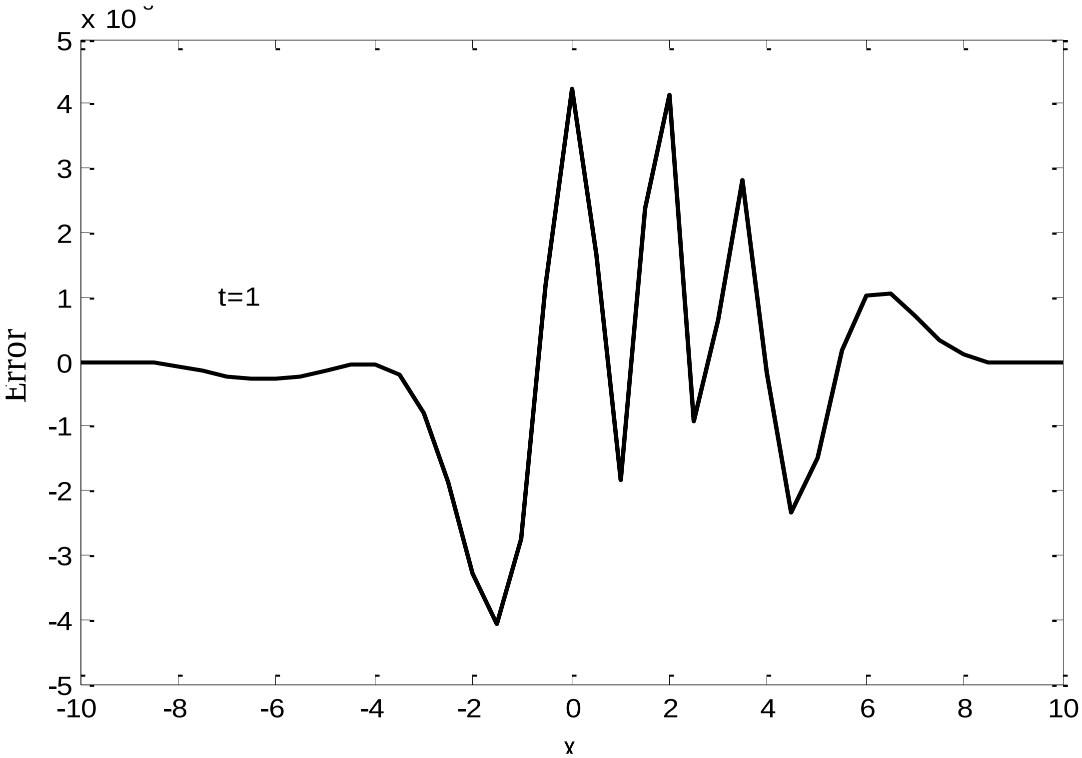


Fig.5 Error plot corresponding to for example 2 and for t=1

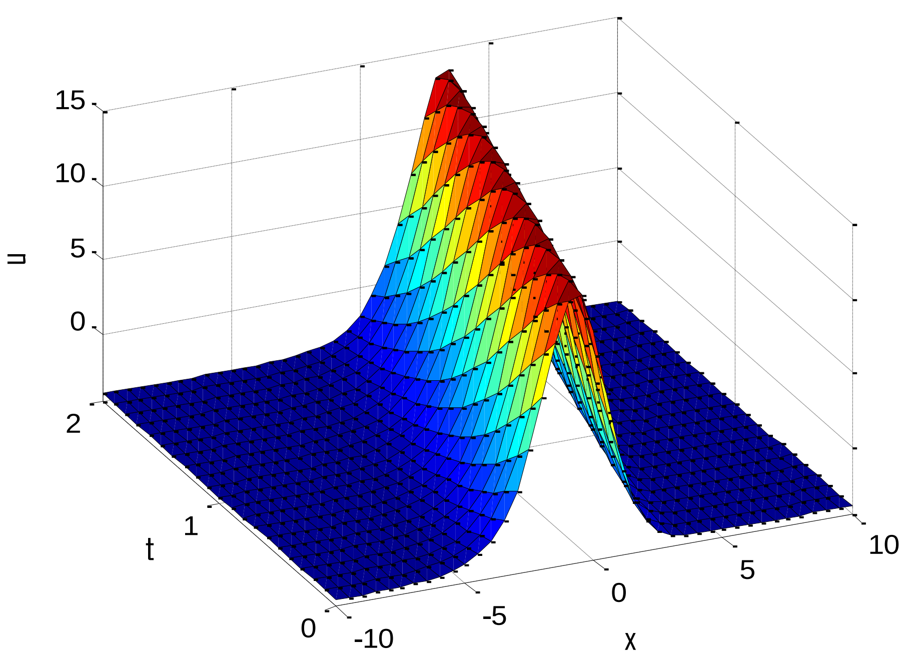


Fig.6 The solutions of u over time interval [0, 2] and space interval [-10, 10] for example 2

Example.3 Consider another type of KS equations

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0\,, \tag{20}$$

which is simple nonlinear differential equation showing the chaotic manners in whole domain.

Consider the Gaussian initial condition

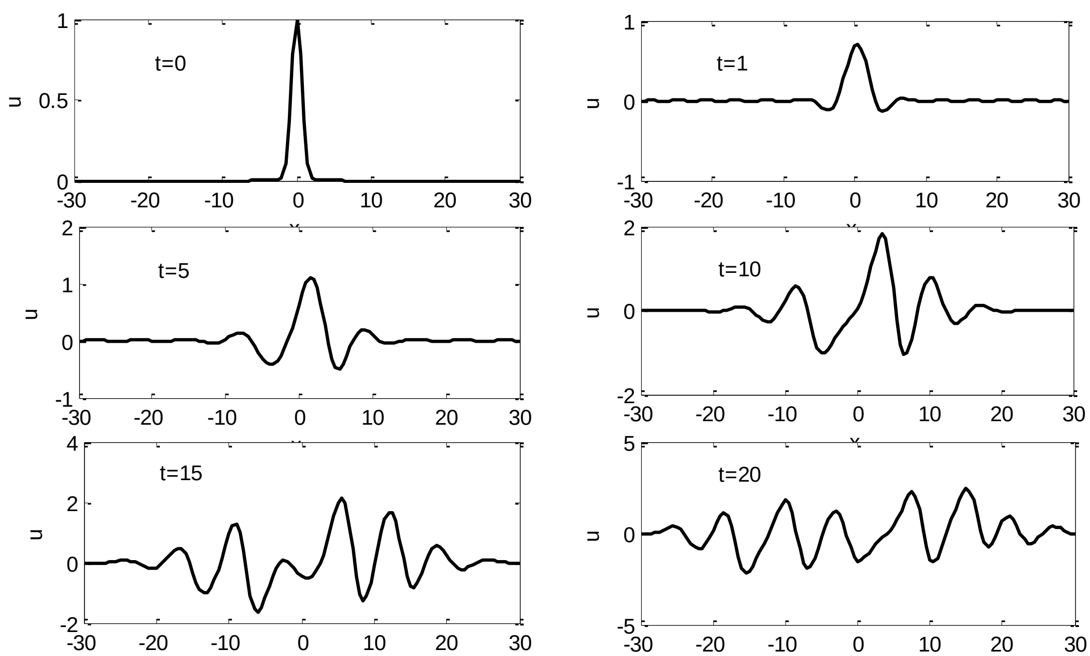
$$u(x,0) = \exp\left(-x^2\right)$$

with the boundary conditions

$$u(a,t) = 0 \quad , \quad u(b,t) = 0.$$

The numerical results are shown in fig. 7 we can detect that the numerical results are convergent for every chaotic behavior.

convergent for every chaotic behavior.



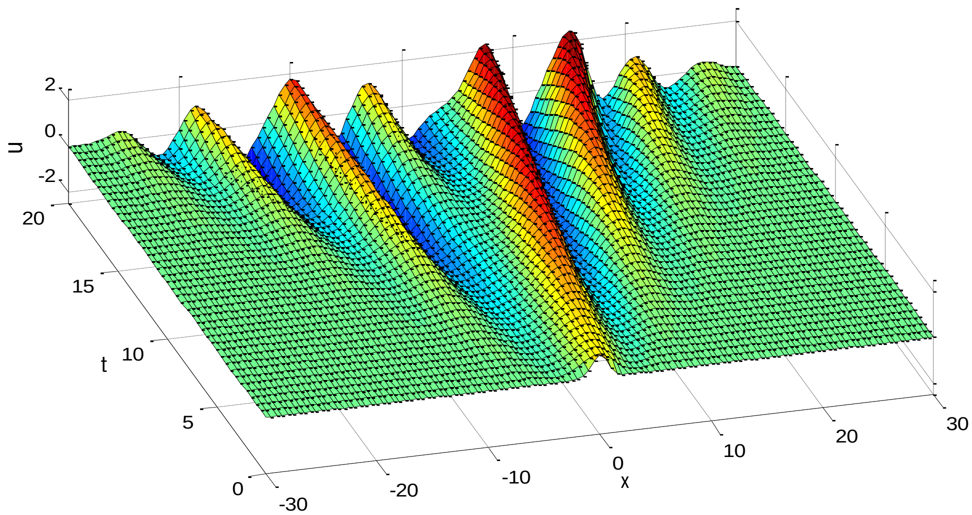


Fig.7 The chaotic solution of KS equation for example 3 with GA initial conditions and boundary conditions over space interval [-30, 30]

4. Conclusion

In this work, a new scheme is planned for the approximate solution of 1-D nonlinear Kuramoto-Sivashinsky equations, which is identified as the KS equation. The method is constructed by using the differential quadrature collocation method along with the Local Radial Basis Function. Some functions are taken, that is, MQ and GA as RBFs. For the accurateness of the newly established method, we have calculated error norms L_2, L_∞ and GRE. The acquired approach is investigated through some examples. Comparison of the new method results is correspondingly made with existing results of Method of Lines (MOL), Mesh Free Collocation Method (MFCM) and Lattice Boltzman Method (LBM). From comparing, we concluded that the results of our new method are better than the mentioned approaches so that this method can be prolonged to other PDEs.

Data Availability Statement

Data will be available on demand.

Acknowledgments

In accordance with the National Research Program for Universities, we are grateful to the Higher Education Commission of Pakistan for financing this initiative.

Conflict of Interest

There is no conflict of interest among authors.

References

[1] E.J. Kansa, Multiquadric scattered data approximation scheme with applications to Computational fluid-dynamics I, surface approximations and partial derivative estimates. Comput. Math. Appl.,19, 127-145, 1990.

[2] Z.H. Zin and T.Y. Ji,Meshless schemes for unsteady Navier–Stokes equations invorticity formulation using radial basis functions. J. Comput. Appl. Math., 192, 328-338, 2006.

[3] Siraj ul Islam, S. Haq and A. Ali, A meshfree method for the numerical solution of RLW equation, J. Comp. Appl. Math., 223, 997-1012, 2012.

[4] B. Sarler and R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems. Comput. Math. Appl.,51, 1269-1282, 2006.

[5] A.R. Mitchell and D.F. Griffiths the Finite Difference Method in partial Differential Equations. John Wiley and Sons Ltd, PP.233, 1980.

[6] L. Wo and Y. K. Kwork. A front-fixing finite difference method for the evaluation of American options Comput. Math. Appl.,6, 83, 1997.

[7] J.M. Melenk and I. Babuska, the partition of unity finite element method, Basic theory and applications. Comput. Math. Appl. Mech.Eng., 139,289-314, 1996.

[8] D. Braess, Finite Elements. Theory, Fast solvers and applications in solid Mechanics, Cambridge University Press, 2001.

[9] R. Eymard, T. Gallouet, and R. Herbin, Finite volume methods. In P.G. Ciarlet and J. L. Lions, editors, Techniques of scientific computing, part 3, Handbook of Numerical Analysis, part 7, 713-1020. North- Holland, Amster dam,2000.

[10] S. N. Atluri and S. Shen, the meshless method. Tech. Sciences press, Forsyth, 2002.

[11] G.E. Fasshauer and G. liu, mesh-free methods, moving beyond the finite element method CR press, Boca Raton, 2007.

[12] N. Mai-Duy and T. Tran-Cong. Mesh-free radial Basis functions network methods with domain decomposition for approximation of functions and numerical solution of poisson's equation. Engineering Analysis with Boundary Elements, 26, 133-156,2002.

[13] S.N. Atluri and S. Shen. The Mesh-less Local petrov-Galerkin (MLPG) Method, Tech. Science Press, Encino, 2002.

[14] G.R. Liv, K.Y. Dai, and Y.T. lim, K.M. Lim, K.M. Gu. A point interpolation mesh-free method for static and frequency analysis of two-dimensional piezoelectric structures, Comput. Mech., 29,510-519,2002.

[15] M.D. Buhmann, Radial Basis Functions, Theory and implementations Cambridge university press, 2003.

[16] R.L. Hardy multiquadric equations of topography and other irregular surfaces. J. Geo phys, Res, 76, 1971.

[17] R. Franke scattered data interpolation, Tests of some methods, Math. Compute, 38, 181-200, 1982.

[18] P.W. Partidge, C.A. Brebbia, and L.C. Wrobel, the Dual Reciprocity Boundary Element method, CMP/ Elsenier,1992.

[19] E. Gregory Fasshauer, solving partial differential equations by collocation with radial basis functions. In surface fitting and multi resolution methods, A.Le. M. Chaut'e, C. Rabut and L.L. schumaker (eds.), Vanderbilt, 131-138,1997.

[20] W. Chen, Z.J. Shen, L.J. Shen and G.W. yuan. General solutions and fundamental solutions varied orders to the vibrational thin, the berger, and the winkler, plates. Engineering Analysis with Boundary Elements, 29, 699-702,2005.

[21] S.C. Brenner and L.R. Scott, The Mathematical Theory of Finite Element Method, Second edition, Springer,2002.

[22] A.P. Hooper and R.S. Grim,Non-linear instability at the interface between two viscous fluids. Phys.Fluids,28, 37-45, 1985.

[23] G.I. Sivashinsky, Instabilities, pattern-formation, and turbulance in flames, Ann. Rev. Fluid Mech., 15, 179–199, 1983.

[24] G.I. Sivashinsky, Instabilities, pattern-formation, and turbulance in flames, Ann. Rev. Fluid Mech., 15, 179–199, 1983.

[25] M. Zarebnia, R. Parvaz, Septic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation. Int. J. Comput. Math. Sci. Eng.,2(1):55–61,2013.

[26] M. Lakestania and M. Dehghan, Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions. Appl. Math. Model., 36(2), 605–17, 2012.

[27] O. Ersoy and I. Dag, the Exponential Cubic B-Spline Collocation Method for the Kuramoto-Sivashinsky Equation, Filomat, 30(3), 853-861, 2016.

[28] A. Kilicman and R. Silambarasan, Modified Kudryashov Method to Solve Generalized Kuramoto-Sivashinsky Equation, Symmetry, 10, 527, 2018.

[29] R. Vertnik and B. Sarler, Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. Int. J. Num. Meth. Heat Fluid Flow,16, 617-640, 2006.

[30] Siraj-ul-Islam, B. Sarler and R. Vertnik, Local radial basis function collocation method along with explicit time stepping for hyperbolic partial differential equations, Appl. Numer. Math.,67, 136-151, 2013.

[31] H. Zheng, C. Zhang, Y. Wang, W. Chen, J. Sladek and V. Sladek, A local RBF collocation method for band structure computations of 2D solid/fluid and solid/fluid phononic crystals, Int. J. Num. Methods Eng., 110, 467-500, 2017.

[32] S. Haq, N. Bibi, S.I.A. Tirmizi and M. Usman, Meshless method of lines for the numerical solution of generalized Kuramoto- Sivashinsky equation, Comput. Math. Appl., 217, 2404–2413, 2010.

[33] H. Lai and M. Changfeng, Lattice Boltzmann method for the generalized Kuramoto-Sivashinsky equation, Physica A, 388,1405–1412,2009.

[35] Q. Shen, Numerical solution of the Sturm-Liouville problem with local RBF-based differential quadrature collocation method. Int. J. Comput.Math 88(2), 285-295,2011.

[36] C. Shu., H. Ding, H.T. Chen and T.G. Wang, An upwind local RBF-DQ method for simulation of inviscid compressible flows. Comput.Math. Appl.Mech.Eng.194, 2001- 2017, 2005.